

Gestor d'un Departament de Formació

Eduard Capell Brufau

ETIG

Director: Jordi Ceballos Vilach

Gener de 2006

Agraïments

Vull agrair, per començar, la col·laboració del meu Director de projecte, Jordi Ceballos Vilach, els seus consells han estat els més encertats i la seva resposta sempre ha estat ràpida i acurada.

Aquest projecte de final de carrera és el punt i final d'un camí que va començar fa uns cinc anys, en part com a entreteniment, en part com a formació complementària per la meva feina.

Després d'aquest temps puc dir que l'experiència ha estat més que profitosa, l'experiència i els coneixements adquirits han estat molt valuosos, i no puc fer una altra cosa que donar les gràcies a la persona que em va empènyer a emprendre aquest camí i que m'ha donat tot el suport en els moments difícils del mateix. Gràcies, Pilar.

Barcelona, Gener de 2006

Índex de continguts

1	Resum del projecte	7
2	Introducció.....	8
2.1	Funcionament general de l'aplicació	8
2.2	Objectius de l'aplicació.....	8
2.3	Enfocament i mètode seguit	8
2.4	Planificació del projecte	9
2.4.1	PAC 1 – Pla de treball (26/9/2005)	9
2.4.2	Anàlisi i Prototip (17/10/2005).....	9
2.4.3	PAC 2 – Disseny (7/11/2005).....	9
2.4.4	PAC 3 – Implementació (9/12/2005).....	9
2.4.5	Lliurament final, Memòria i Presentació (9/1/2006).....	10
2.4.6	Diagrama de Gantt.....	10
2.5	Productes obtinguts	10
2.6	Capítols següents:	11
3	Anàlisi de requeriments	12
3.1	Composició del programari	12
3.1.1	Subsistema d'administració.....	12
3.1.2	Subsistema de gestió de cursos.....	12
3.1.3	Subsistema de gestió d'alumnes	12
3.1.4	Subsistema d'entrada / sortida.....	12
3.2	Requeriments funcionals	12
4	Arquitectura de l'aplicació	14
4.1	El patró MVC	14
5	Model de dades.....	16
5.1	Detall de les taules.....	16
5.1.1	Taula 'Usuari'	16
5.1.2	Taula 'Alumne'	17
5.1.3	Taula 'Formador'	18
5.1.4	Taula 'Administrador'	18
5.1.5	Taula 'Curs'	18
5.1.6	Taula 'AssistentCurs'	19
5.1.7	Taula 'FormadorCurs'	19
5.1.8	Taula 'InscritCurs'	19
5.1.9	Taula 'Materia'	19

5.1.10	Taula 'MateriaCurs'	20
5.1.11	Taula 'Recurs'	20
5.1.12	Taula 'TipusRecurs'	20
5.1.13	Taula 'RecursTipus'	20
5.1.14	Taula 'ValoracioCurs'	21
6	Disseny de classes	22
6.1	Classes del model	22
6.2	Classes d'accés a dades	22
6.3	Classes del <i>framework</i> Struts	23
6.3.1	Classes <i>Action</i>	24
6.3.1.1	MantenimentUsuarisAction	25
6.3.1.2	MantenimentRecursosAction	25
6.3.1.3	GestioAlumnesAction	25
6.3.1.4	MantenimentMateriesAction	25
6.3.1.5	MantenimentCursAction	25
6.3.1.6	GestioCursosAction	26
6.3.1.7	LoginAction	26
6.3.1.8	LogoutAction	26
6.3.2	Classes <i>ActionForm</i>	26
7	Arquitectura de la Interfície (Vista)	28
7.1	Benvinguda (index.jsp)	28
7.2	Pantalla principal (main.jsp)	28
7.3	Pàgina de capçalera (capcalera.jsp)	29
7.4	Pàgina de benvinguda (benvinguda.jsp)	29
7.5	Usuari estàndard (estandard.jsp)	30
7.6	Usuari formador (formador.jsp)	30
7.7	Usuari administrador (administrador.jsp)	30
7.8	Formulari de cursos (form_curs.jsp)	30
7.9	Formulari de matèries (form_materia.jsp)	31
7.10	Formulari de recursos (form_recurs.jsp)	31
7.11	Formulari de persones (form_persona.jsp)	32
7.12	Baixa de persones (baixa_persona.jsp)	32
7.13	Modificació de persones (modifica_persona.jsp)	33
7.14	Bloqueig d'usuaris (bloqueig_persona.jsp)	33
7.15	Llista de cursos (llista_cursos.jsp)	33

8	Consideracions tècniques	35
8.1	Programari	35
8.1.1	Pàgines web	35
8.1.2	Servidor d'aplicacions	35
8.1.3	Servidor de base de dades.....	35
8.1.4	Sistema operatiu	35
8.2	Maquinari	35
9	Conclusions	37
9.1	Línies futures de treball	37
10	Glossari.....	38
11	Bibliografia.....	40
12	Annexos	41

Índex de figures

Diagrama de Gantt del projecte	10
Struts i MVC.....	14
Diagrama E/R	16
Classes del model	22
Classes d'accés a dades	23
Classes <i>Action</i>	24
Classes <i>Form</i>	27
Pàgina de benvinguda (<i>index.jsp</i>).....	28
Pàgina de benvinguda (<i>benvinguda.jsp</i>).....	30
Pàgina de l'usuari estàndard (<i>estandard.jsp</i>).....	30
Pàgina de l'usuari formador (<i>formador.jsp</i>).....	30
Pàgina de l'usuari administrador (<i>administrador.jsp</i>).....	30
Pàgina de detalls del curs (<i>form_curs.jsp</i>).....	31
Formulari de matèries (<i>form_materia.jsp</i>).....	31
Formulari de recursos (<i>form_rekurs.jsp</i>).....	32
Formulari de persones (<i>form_persona.jsp</i>).....	32
Formulari de baixa de persones (<i>baixa_persona.jsp</i>).....	32
Formulari de modificació de persones (<i>modifica_persona.jsp</i>)	33
Formulari de bloqueig d'usuaris (<i>bloqueig_persona.jsp</i>)	33
Llista de cursos (<i>llista_cursos.jsp</i>)	33

1 Resum del projecte

La tecnologia J2EE és una de les més presents en la majoria d'entorns empresarials, sobre tot quan es tracta d'aplicacions en un entorn web i aplicacions client – servidor.

És per aquest motiu que he decidit que aquesta havia de ser la tecnologia que triaria per al projecte. A més, he fet ús d'un framework, Jakarta Struts, que em permet d'aprofundir en l'arquitectura Model – Vista – Controlador (MVC), que és el fonament de moltes aplicacions client – servidor actuals.

Part del projecte també ha consistit en el procés de preparació, anàlisi, disseny, implementació, documentació, etc. Això és el cicle de vida d'un producte de software des de la seva concepció com a idea de solució a un problema fins al moment de la implantació d'aquesta solució. És evident que hi falta una fase addicional, molt important, la de producció en un entorn real, amb el seguiment del producte, dels comentaris dels usuaris, dels desenvolupadors que el mantenen, etc.

El producte busca la facilitat d'ús per tal que qualsevol usuari d'una empresa el pugui fer servir sense cap mena de problema, esperant d'aquesta manera que aquest sigui el portal de formació dels empleats d'una empresa genèrica.

Per tota l'aplicació s'han utilitzat tecnologies de codi obert, des del servidor d'aplicacions, fins a la base de dades, passant pel framework Struts i fins i tot pel disseny de l'aspecte visual de les pàgines web.

2 Introducció

Aquest projecte s'emmarca en la gestió integral d'una empresa a través de la intranet.

En aquesta intranet s'hi troben multitud d'aplicacions de gestió, entre les quals hi haurà la del Departament de Formació.

La formació és un aspecte cada cop més rellevant en les empreses d'avui en dia, ja que els empleats acostumen a desenvolupar les seves funcions en un entorn cada cop més canviant, i que requereix una labor de formació (presencial o no) cada cop més intensa.

Per tal de poder gestionar el volum d'informació cada vegada més gran que tindrem com a conseqüència d'una activitat formativa creixent, no n'hi ha prou amb els sistemes tradicionals, i hem de passar a un entorn de control de la informació el més exhaustiu i precís possible, a més de ser en un entorn ben conegut per a tots els usuaris del mateix.

Aquesta és la causa de la creació d'una aplicació de gestió integral de l'acció formativa, en un entorn web.

2.1 Funcionament general de l'aplicació

L'aplicació consisteix en una interfície web on hi tenen accés tots els empleats d'una empresa. Aquests empleats han estat donats d'alta prèviament, i poden fer una sèrie d'accions en funció del seu perfil d'usuari (Alumne / Formador / Administrador).

Les accions bàsiques dels usuaris es basen en els perfils descrits:

- **Alumne:** pot accedir a la llista de cursos disponibles, i s'hi pot inscriure, si és que són del seu interès.
- **Formador:** pot veure la informació dels cursos que se li han assignat, i pot reservar recursos per un curs concret i gestionar les inscripcions d'alumnes.
- **Administrador:** pot crear cursos i gestionar-ne l'assignació d'alumnes. També s'ocupa de crear i mantenir la llista de persones que poden utilitzar l'aplicació.

2.2 Objectius de l'aplicació

Els objectius de l'aplicació són els següents:

- Centralitzar la informació referent a les accions de formació de tota l'empresa.
- Ser un punt de referència sobre l'acció formativa per tot el personal de l'empresa, fent que sempre que es parli de formació, el punt d'informació bàsic (únic) sigui aquest lloc web.
- Ser un punt de trobada de tots els alumnes pel que fa a formació, podent desenvolupar mòduls addicionals a tal efecte (fòrums, bústia de suggeriments, etc.)

2.3 Enfocament i mètode seguit

Com qualsevol aplicació web amb accés a base de dades, calia estructurar l'aplicació en tres parts o capes ben diferenciades:

- La capa de la interfície gràfica d'usuari, constituïda per un conjunt de pàgines web, que en el nostre cas són pàgines JSP.

- La capa del model de l'aplicació, que és un conjunt de classes Java més la base de dades on es guarda tota la informació de l'aplicació.
- La capa del controlador de l'aplicació, que és un conjunt de classes Java que s'utilitzen per a gestionar els fluxos de peticions del client, accessos a la base de dades, etc.

Aquesta estructura en capes ben diferenciades ens permet, de manera fàcil, de modificar-ne una d'elles sense afectar de manera més que residual les altres dues. Per exemple, si es vol canviar de motor de base de dades, només caldrà modificar el *Driver* d'accés a la base de dades, i potser algun fitxer de configuració, res més.

2.4 Planificació del projecte

Donat que el projecte s'emmarca en l'assignatura de TFC de la Universitat Oberta de Catalunya, els entregables del mateix tenien assignada una temporalitat fixada en forma de "PACs", com es fa en la resta d'assignatures de la Universitat. És per això que les fites més importants del projecte estaven clarament definides ja d'entrada, i eren les següents, segons el Pla Docent de l'assignatura TFC:

2.4.1 PAC 1 – Pla de treball (26/9/2005)

Aquest document consistia en una breu presentació del que hauria d'acabar sent el projecte, i constava dels següents elements:

- Requeriments funcionals
- Requeriments tècnics
- Planificació temporal

2.4.2 Anàlisi i Prototip (17/10/2005)

Aquest document era una ampliació dels requeriments funcionals especificats al Pla de Treball, de manera que quedés ben clara la funcionalitat de l'aplicació, especificant correctament cada cas d'ús.

El prototip era una maqueta de l'aplicació, per mostrar clarament quin seria el funcionament en un entorn semblant al real.

2.4.3 PAC 2 – Disseny (7/11/2005)

En aquest document ja s'especificava de manera definitiva com s'implementaria allò que fins ara només era una especificació en forma de casos d'ús. En aquest moment es baixava ja a nivell de tecnologia d'implementació (J2EE), i també es definia clarament el model de base de dades en el qual s'havia de recolzar l'aplicació.

2.4.4 PAC 3 – Implementació (9/12/2005)

Aquest lliurament consistia en una aplicació completament funcional i que s'adaptés als requisits indicats al document d'anàlisi, i implementada segons el definit al document de disseny.

2.4.5 Lliurament final, Memòria i Presentació (9/1/2006)

Finalment, el lliurament final consisteix en aquest document, més una presentació de diapositives sobre l'aplicació. També s'hi inclou la implementació definitiva i el manual d'instal·lació de l'aplicació.

2.4.6 Diagrama de Gantt

Al següent diagrama s'hi pot observar la planificació de tot el semestre, amb les diferents fases del projecte:

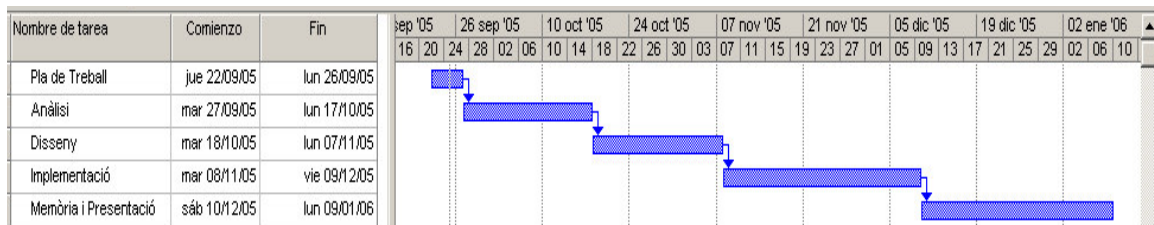


Diagrama de Gantt del projecte

2.5 Productes obtinguts

Durant la realització del projecte s'han anat obtenint documents i/o productes com a resultat de la feina feta. Els productes acostumaven a tenir relació amb l'entrega que tocava en aquell moment, i es poden resumir en la següent relació:

- Pla de Treball: document amb la declaració d'intencions inicial, esbós dels requeriments, i planificació pel semestre.
- Anàlisi de requeriments: document on s'hi feien constar els requeriments funcionals de l'aplicació i que caldria posteriorment implementar.
- Prototip del projecte: aplicació web estàtica que era una "demo" del que hauria de ser l'aplicació definitiva. És una entrega fonamental per tal que l'usuari final pugui fer-se una idea del *look and feel* de l'aplicació, i perquè hi posi objeccions o hi faci suggeriments.
- Document de disseny: el document que especifica com s'implementarà allò que s'ha dissenyat prèviament. Es baixa a nivell de classes, arquitectura del sistema, disseny del model relacional subjacent, etc.
- El propi projecte *Gestor d'un Departament de Formació*: consistia en un conjunt de fitxers que composaven l'aplicació web en el seu conjunt. El contingut era el següent:
 - Fitxers de configuració del servidor d'aplicacions.
 - Fitxers de configuració de la pròpia aplicació.
 - Fitxers de biblioteques necessàries pel bon funcionament de l'aplicació.
 - Codi font de totes les pàgines JSP i de totes les classes Java.
 - Classes Java compilades.
 - Fitxer *.sql* amb l'script que permet de crear l'estructura de la base de dades.
 - Fitxer amb unes breus instruccions d'instal·lació.

- Memòria: és el present document, que ve a ser un resum – explicació del projecte.
- Manual d'instal·lació de l'aplicació: document que indica com s'ha de posar en marxa l'aplicació i n'especifica els requisits.
- Presentació en MS –Powerpoint.

2.6 Capítols següents:

Els capítols següents resumiran el cicle de vida del desenvolupament de l'aplicació, i s'estructuren de la següent manera:

- Al capítol 3 (Anàlisi de requeriments) parlem de l'anàlisi dels requeriments de l'aplicació.
- Al capítol 4 (Arquitectura de l'aplicació) es tracta d'explicar el disseny a alt nivell de l'aplicació, com s'ha utilitzat en el cas que ens ocupa i per quin motiu.
- Al capítol 5 (Model de dades) s'hi detalla l'estructura de la base de dades relacional de l'aplicació.
- Al capítol 6 (Disseny de classes) es parla del model de classes en el qual es recolza l'aplicació, amb diagrames UML que mostren l'estructura de la implementació de l'aplicació.
- Al capítol 7 (Arquitectura de la interfície) s'hi mostren les pantalles de l'aplicació, explicant-ne la funcionalitat.
- Al capítol 8 (Consideracions tècniques) s'hi expliquen els detalls tècnics de la implementació i els requeriments de maquinari i programari.
- Al capítol 9 (Conclusions) es comenten les conclusions del projecte i es parla de les línies futures de treball que se'n podrien derivar del mateix.

3 Anàlisi de requeriments

Aquest pas és fonamental. Sense una bona anàlisi de requeriments, una aplicació no serà mai acceptable, ja que no estarà ben definit el que ha de fer. Si l'anàlisi de requeriments no és exhaustiu fins al darrer detall, el Client sempre dubtarà dels resultats, i el Desenvolupador mai no estarà segur de la correctesa de la feina realitzada.

L'anàlisi de requeriments sol constar d'una descripció breu del projecte, identificació dels actors que intervindran en l'ús de l'aplicació, la composició del programari i la descripció de les tasques que cada actor realitzarà dins de l'aplicació. A més, també se sol fer un primer esbós de la interfície gràfica d'usuari.

3.1 Composició del programari

Funcionalment, l'aplicació consta de tres mòduls separats clarament, més un mòdul addicional d'entrada/sortida del sistema (login/logout).

3.1.1 Subsistema d'administració

Aquest subsistema s'ocupa de les funcions de manteniment del propi sistema: manteniment de persones, manteniment de cursos, manteniment de recursos, matèries, etc.

Només el gestionen els usuaris amb el perfil *Administrador*.

3.1.2 Subsistema de gestió de cursos

Aquest subsistema s'ocupa de les funcions de manteniment dels cursos un cop que aquests ja han estat creats i se'ls ha assignat un formador. Consisteix en l'assignació de recursos, acceptació de preinscripcions, etc.

El gestionen els usuaris administradors i/o formadors.

3.1.3 Subsistema de gestió d'alumnes

Aquest subsistema és el comú a tots els usuaris (qualsevol usuari és, com a mínim, Alumne). Un alumne pot veure els cursos disponibles, preinscriure-s'hi, i veure el seu propi calendari de cursos, a més de valorar aquells als quals assisteix.

3.1.4 Subsistema d'entrada / sortida

Aquest subsistema és necessari per tal que els usuaris s'identifiquin a l'aplicació. En qualsevol petició que es faci, es comprova que hi hagi un usuari actiu en la sessió, prèviament identificat.

3.2 Requeriments funcionals

Segons la informació proporcionada pel client, i consensuada amb ell, els requeriments funcionals bàsics de l'aplicació són els següents:

- Cal una aplicació accessible des de qualsevol punt de l'empresa, i fins i tot remotament (via extranet o altres).
- L'aplicació ha de suportar múltiples plataformes (les actuals, o les que es decideixi d'implantar en el futur).

- L'aplicació no ha de dependre d'un client únic (aplicació d'un sistema operatiu, per exemple).
- Ha de ser una aplicació de fàcil ús i intuïtiva.
- Cal que a l'aplicació hi hagi gestió d'usuaris.
- Els perfils d'usuaris són 3: Alumnes, Formadors, i Administradors. Els Formadors també són al mateix temps Alumnes, i els Administradors també són al mateix temps Formadors.
- Només es mostraran les funcions disponibles pel perfil de l'usuari que es connecti en aquell moment (un Administrador ho podrà veure tot, però un Alumne només la part corresponent als Alumnes).
- Els Administradors s'ocupen de la gestió de persones, podent-les crear i eliminar, excepte aquelles persones que són super-usuàries de l'aplicació, que no poden ser eliminades en cap cas (aquestes últimes només es poden modificar directament a través de l'accés a la Base de Dades).
- Els Administradors s'ocupen del manteniment dels Cursos i de les entitats que hi estan relacionades (Matèries, Recursos, assignació de Formadors).
- Els Formadors s'ocupen de gestionar aquells cursos que els han estat assignats, podent-hi reservar recursos i gestionar les inscripcions dels alumnes.
- Els Alumnes poden inscriure's a qualsevol curs (el Formador validarà aquesta inscripció amb posterioritat) i poden veure el Calendari complert de cursos de tota l'empresa.
- Els Alumnes poden valorar un curs segons uns criteris de valoració establerts.

4 Arquitectura de l'aplicació

En aquest apartat expliquem el perquè de l'elecció de l'arquitectura de l'aplicació, i en veurem els detalls.

4.1 El patró MVC

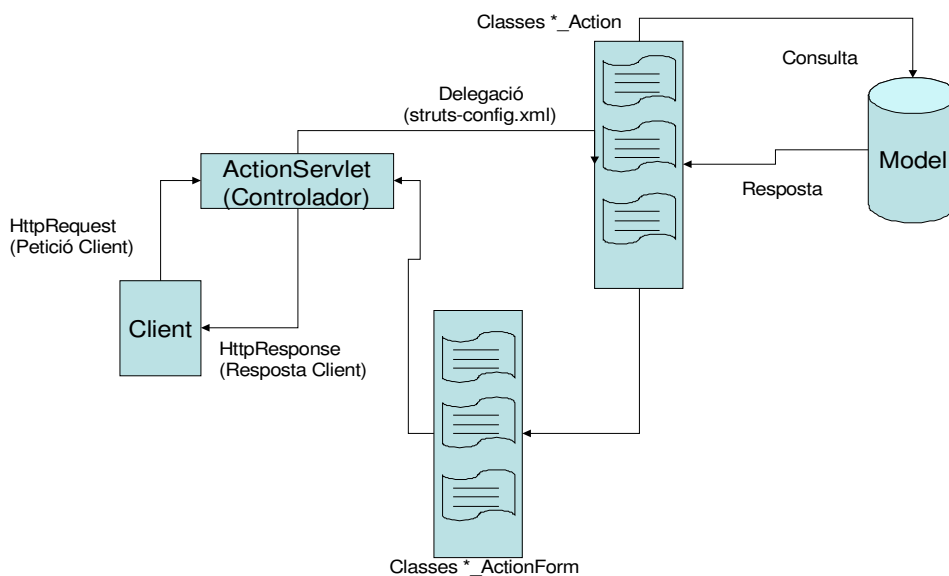
El patró de disseny MVC és un clàssic en les aplicacions web d'avui dia. La idea en la qual es fonamenta és ben simple: cal independitzar les capes d'una aplicació que tenen responsabilitats diferents, de manera que sigui fàcil de canviar-ne una d'elles sense afectar gaire o gens la resta del sistema.

Amb aquesta filosofia, s'ha acceptat que les capes amb responsabilitats diferents són la Vista (la part que "es veu", que correspon a la interfície gràfica d'usuari); el Model (la part que modela la realitat, bàsicament classes Java d'entitat i la Base de Dades); el Controlador (la part que gestiona totes les peticions, les encamina a les classes del model que correspongui, i indica la vista que cal mostrar a l'usuari en funció de la petició i del resultat d'aquesta).

Si en una aplicació MVC canviéssim alguna de les parts del sistema, per exemple la Vista, passant d'una interfície Web a un Applet Java, per exemple, no caldria fer grans canvis en la resta de l'aplicació, només caldria fer un Applet que s'adaptés mínimament al controlador, que és amb qui es relaciona la Vista.

És evident que l'element central del patró de disseny MVC és el controlador, que és l'única capa que està en contacte amb les altres dues. Per tal d'utilitzar un controlador el més eficient possible, he optat per escollir un *framework* que ja em dóna un Controlador implementat i amb un gran nombre de funcionalitats molt interessants, aquesta implementació és *Jakarta Struts* (<http://struts.apache.org>). En l'aplicació del projecte he utilitzat la versió 1.1 de Struts. Aquest *framework*, configurat a través de fitxers xml de configuració, gestionarà totes les peticions dels usuaris i les reenviarà a les classes del Model que desitgem. En funció de les respostes d'aquestes classes del Model, el Controlador redirigirà la petició cap a una vista o una altra.

A continuació es pot veure un diagrama sobre Struts i MVC que comentarem:



Struts i MVC

En aquest cas es pot veure com el Client fa peticions, a través de la Vista, i aquestes peticions arriben al Controlador (el servlet *ActionServlet* de Struts).

L'*ActionServlet*, en funció de la configuració del fitxer *struts-config.xml*, definit pel desenvolupador, li delegarà les instruccions específiques a la classe del Model que correspongui segons la petició feta pel client. El Model, representat per una classe *_Action*, acudirà si cal a la Base de Dades (part del Model, conceptualment), i retornarà una resposta a l'*ActionServlet*. Aquest, en funció d'aquesta resposta, i fent ús de les classes *_ActionForm*¹, redirigirà la petició a la pantalla que calgui.

Anem a veure aquest procés amb un exemple, ben simple, com ara la pantalla de Login.

- L'usuari introdueix el seu usuari i password a un formulari (Vista) JSP i fa click al botó "Enviar".
- La petició arriba a l'*ActionServlet* (Controlador), que la processa, i, combinant el patró de la URL de la petició amb el fitxer *struts-config.xml*, arriba a la conclusió que cal delegar l'acció a la classe *LoginAction*.
- La classe *LoginAction* (Controlador) instanciarà un objecte de tipus Usuari (Model) i aquest acudirà a la Base de Dades per tal de comprovar si la combinació usuari / password és acceptable.
- Quan la classe Usuari tingui una resposta de la base de dades, li donarà aquesta resposta a la classe *LoginAction*, que redirigirà la vista cap a la pantalla de benvinguda (login amb èxit) o altre cop a la pantalla de login amb algun missatge d'avertència (login sense èxit).

La resta de peticions de l'aplicació es gestionaran de manera anàloga.

¹ És discutible si les classes hereves d'*ActionForm* pertanyen al Controlador o a la Vista. En la meua opinió, són part de la Vista, ja que són un objecte Java que representa un formulari de la pàgina web.

5 Model de dades

En aquest apartat s'especifica el model relacional en el qual es basa el funcionament de l'aplicació.

El diagrama E/R de l'aplicació és el següent:

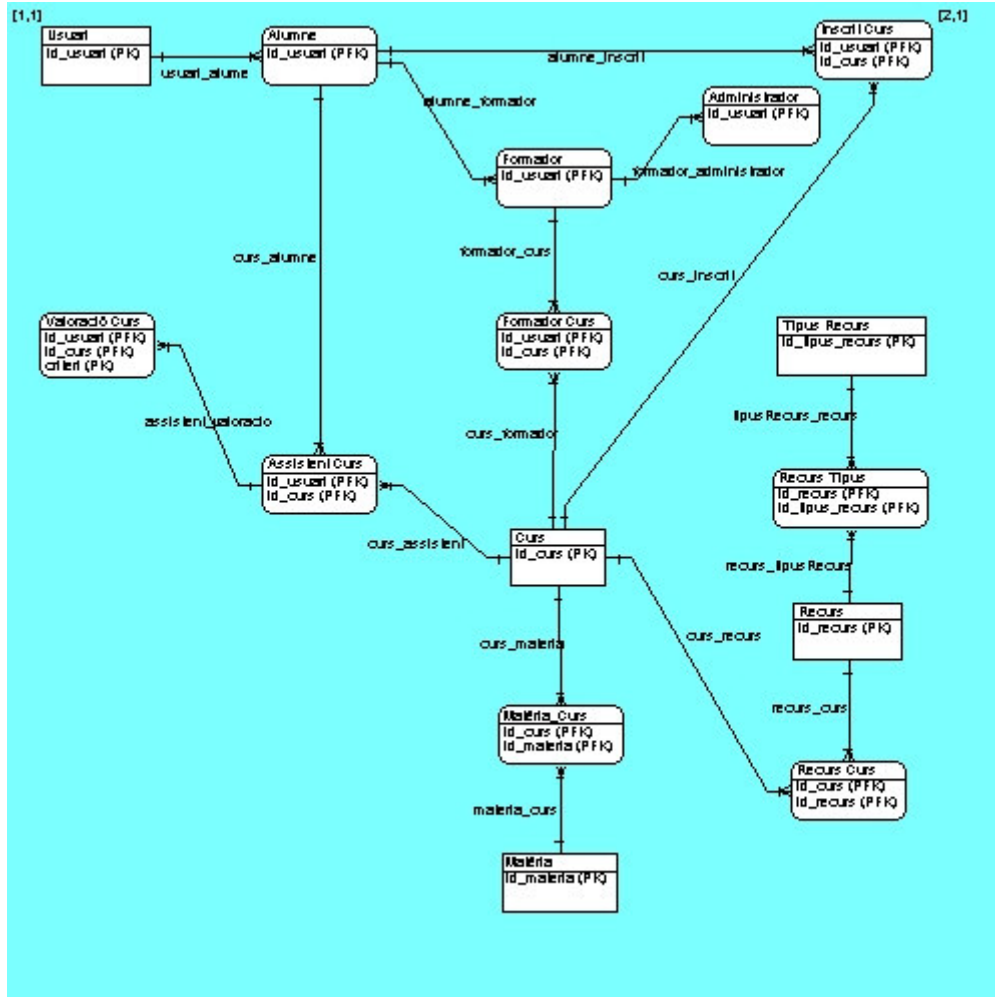


Diagrama E/R

5.1 Detall de les taules

A continuació es mostrarà el detall al complet de cada taula, amb tots els camps, tipus de dades, claus primàries, i comentaris que corresponguin. Els camps de tipus TEXT són camps amb caràcters, amb capacitat fins a 64K.

5.1.1 Taula 'Usuari'

Aquesta taula guarda la informació de qualsevol persona que pot utilitzar el sistema. D'aquest usuari n'hereten tots els usuaris del sistema, siguin del perfil (Alumne, Formador, Administrador) que siguin. Cal que, per començar a fer funcionar el sistema, hi hagi almenys un usuari creat de tipus Administrador, ja que, si no fos així, no es podria començar a donar d'alta d'altres usuaris.

Camp	Tipus	Clau	Descripció	NULL
------	-------	------	------------	------

id_usuari	INTEGER	Primària	Codi d'usuari	NO
nom	VARCHAR(50)		Nom de l'usuari	NO
cognom1	VARCHAR(50)		Primer cognom de l'usuari	NO
cognom2	VARCHAR(50)		Segon cognom de l'usuari	
data_naixement	DATE		Data de naixement de l'usuari	
adreca	VARCHAR(255)		Adreça postal de l'usuari	
login	VARCHAR(20)	Única	Identificador de l'usari per a entrar al sistema	NO
pass	VARCHAR(32)		Hash MD5 de la contrassenya de l'usuari	NO
bloquejat	ENUM('S', 'N')		Indicador d'usuari bloquejat (default 'N')	NO
baixa	ENUM('S', 'N')		Indicador d'usuari de baixa (default 'N')	NO
observacions	TEXT		Observacions sobre la persona	
reintents_ko	TINYINT UNSIGNED		Nombre de cops que l'usuari ha intentat d'entrar sense èxit. En arribar a 3 l'usuari es bloquejarà. (default 0).	NO

5.1.2 Taula 'Alumne'

En aquesta taula hi ha el codi d'usuari de totes aquelles persones que tenen el perfil d'alumne. Tota la seva informació és igualment a la taula d'usuari, i l'únic que fa que siguin en aquesta taula és el seu perfil.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Usuaris(id_usuari)	Codi d'usuari de l'alumne	NO

5.1.3 Taula 'Formador'

En aquesta taula hi ha el codi d'usuari de totes aquelles persones que tenen el perfil de formador. Tota la seva informació és igualment a la taula d'usuari, i l'únic que fa que siguin en aquesta taula és el seu perfil.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Alumne(id_usuari)	Codi d'usuari del Formador	NO

5.1.4 Taula 'Administrador'

En aquesta taula hi ha el codi d'usuari de totes aquelles persones que tenen el perfil d'administrador. Tota la seva informació és igualment a la taula d'usuari, i l'únic que fa que siguin en aquesta taula és el seu perfil.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Formador(id_usuari)	Codi d'usuari de l'Administrador	NO
super_usuari	ENUM('S', 'N')		Indica si l'usuari és un super administrador, que no pot ser esborrat excepte per accés directe a la BD (default 'N')	N

5.1.5 Taula 'Curs'

Aquesta taula conté informació sobre tots els cursos que imparteix el Departament de Formació.

Camp	Tipus	Clau	Descripció	NULL
id_curs	INTEGER	Primària	Codi del curs	NO
nom	VARCHAR(50)		Nom del curs	NO
descripcio	VARCHAR(255)		Descripció llarga del curs	NO
data_inici	DATE		Data en què comença a impartir-se el curs	NO
data_final	DATE		Data en què acaba el curs	NO
numero_hores	INT UNSIGNED		Nombre d'hores que dura el curs (default 0)	NO

5.1.6 Taula 'AssistentCurs'

Aquesta taula mostra els alumnes que van a un curs.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Alumne(id_usuari)	Usuari que assisteix al curs	NO
id_curs	INTEGER	Primària i Forana, Curs (id_curs)	Curs al qual assisteix l'alumne	NO

5.1.7 Taula 'FormadorCurs'

Aquesta taula mostra els formadors d'un curs.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Formador(id_usuari)	Formador que imparteix en el curs en qüestió	NO
id_curs	INTEGER	Primària i Forana, Curs (id_curs)	Curs impartit	NO

5.1.8 Taula 'InscritCurs'

És la taula que conté les inscripcions fetes pels alumnes, que posteriorment seran o no confirmades.

Camp	Tipus	Clau	Descripció	NULL
id_usuari	INTEGER	Primària i Forana, Alumne(id_usuari)	Usuari que s'inscriu al curs	NO
id_curs	INTEGER	Primària i Forana, Curs (id_curs)	Curs al qual s'inscriu l'usuari	NO
data_inscripcio	DATE		Data en què es formalitza la inscripció	NO

5.1.9 Taula 'Materia'

És la taula que conté les possibles matèries sobre les quals pot tractar algun curs.

Camp	Tipus	Clau	Descripció	NULL
id_materia	INTEGER	Primària	Codi de la matèria	NO
nom	VARCHAR(100)		Nom o descripció	NO

			de la matèria	
--	--	--	---------------	--

5.1.10 Taula 'MateriaCurs'

Aquesta taula contè les associacions entre cada curs i les matèries sobre les quals tracta.

Camp	Tipus	Clau	Descripció	NULL
id_curs	INTEGER	Primària i Forana, Curs (id_curs)	Codi del curs	NO
id_materia	INTEGER	Primària i Forana, Materia (id_materia)	Nom o descripció de la matèria	NO

5.1.11 Taula 'Recurs'

Aquesta taula contè els recursos amb què es compta per a poder impartir cursos.

Camp	Tipus	Clau	Descripció	NULL
id_recurs	INTEGER	Primària	Codi del recurs	NO
nom	VARCHAR (100)		Nom o descripció del recurs	NO

5.1.12 Taula 'TipusRecurs'

Aquesta taula contè els possibles tipus de recursos que té l'aplicació.

Camp	Tipus	Clau	Descripció	NULL
id_tipus_recurs	INTEGER	Primària	Codi del tipus de recurs	NO
nom	VARCHAR (100)		Nom o descripció del tipus de recurs	NO

5.1.13 Taula 'RecursTipus'

Aquesta taula contè les associacions entre recursos i tipus de recursos, de manera que indica de quin tipus és cada un dels recursos.

Camp	Tipus	Clau	Descripció	NULL
id_recurs	INTEGER	Primària i Forana, Recurs (id_recurs)	Codi del recurs	NO
id_tipus_recurs	INTEGER	Primària i Forana, TipusRecurs (id_tipus_recurs)	Codi del tipus de recurs	NO

5.1.14 Taula 'ValoracioCurs'

Aquesta taula contè les valoracions que els alumnes fan sobre els diferents aspectes (criteris) d'un curs.

Camp	Tipus	Clau	Descripció	NULL
id_usuari*	INTEGER	Primària	Codi de l'alumne que valora	NO
id_curs*	INTEGER	Primària	Codi del curs que es valora	NO
materials	TINYINT UNSIGNED		Nota pels materials (default 0)	NO
professor	TINYINT UNSIGNED		Nota pel professor (default 0)	NO
convocatoria	TINYINT UNSIGNED		Nota per la convocatòria (default 0)	NO
instal·lacions	TINYINT UNSIGNED		Nota per les instal·lacions (default 0)	NO
observacions	TEXT		Observacions generals sobre el curs	

* Aquests dos camps (*id_usuari* + *id_curs*), formen una clau forana, que referencia els camps *id_usuari* + *id_curs* de la taula *AssistentCurs*.

6 Disseny de classes

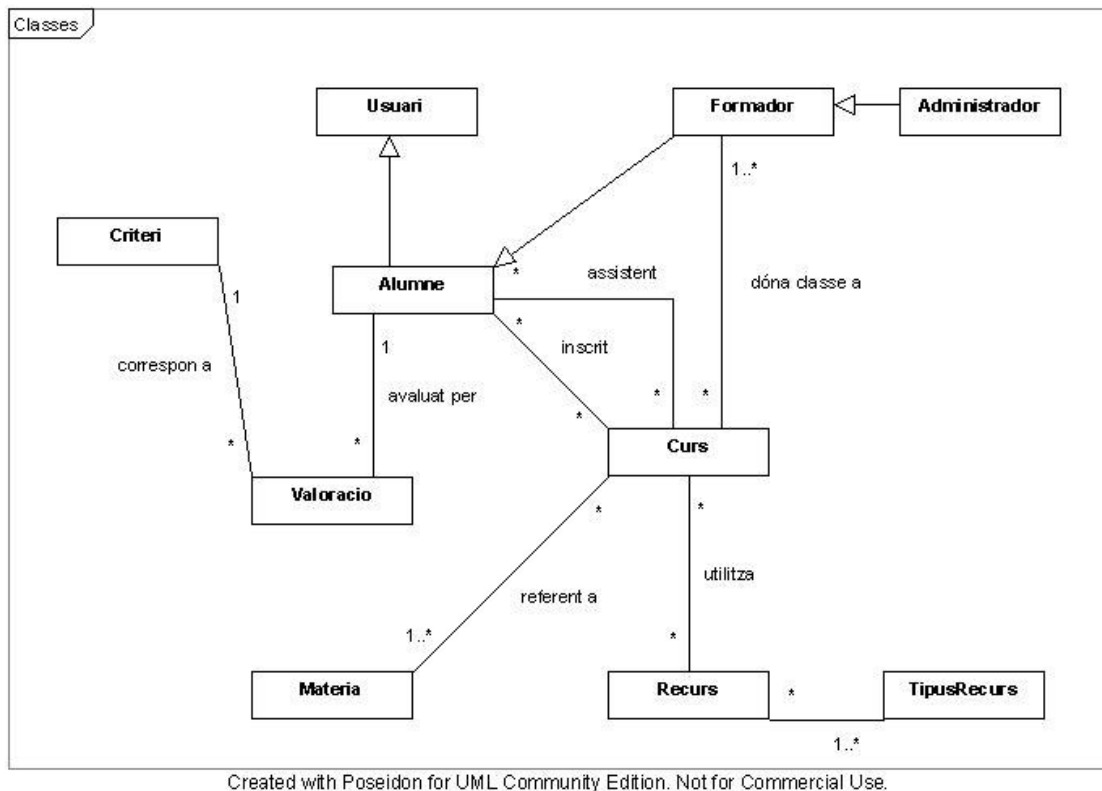
En el disseny de classes, diferenciarem entre diferents part del sistema, per tal de veure un millor detall del disseny proposat.

Per començar, veurem les classes del model, aquelles que fan referència a objectes del món real, que l'aplicació representa.

A continuació veurem les classes d'accés a dades.

Tot seguit veurem les classes de l'arquitectura Struts superposada ja a la nostra aplicació.

6.1 Classes del model

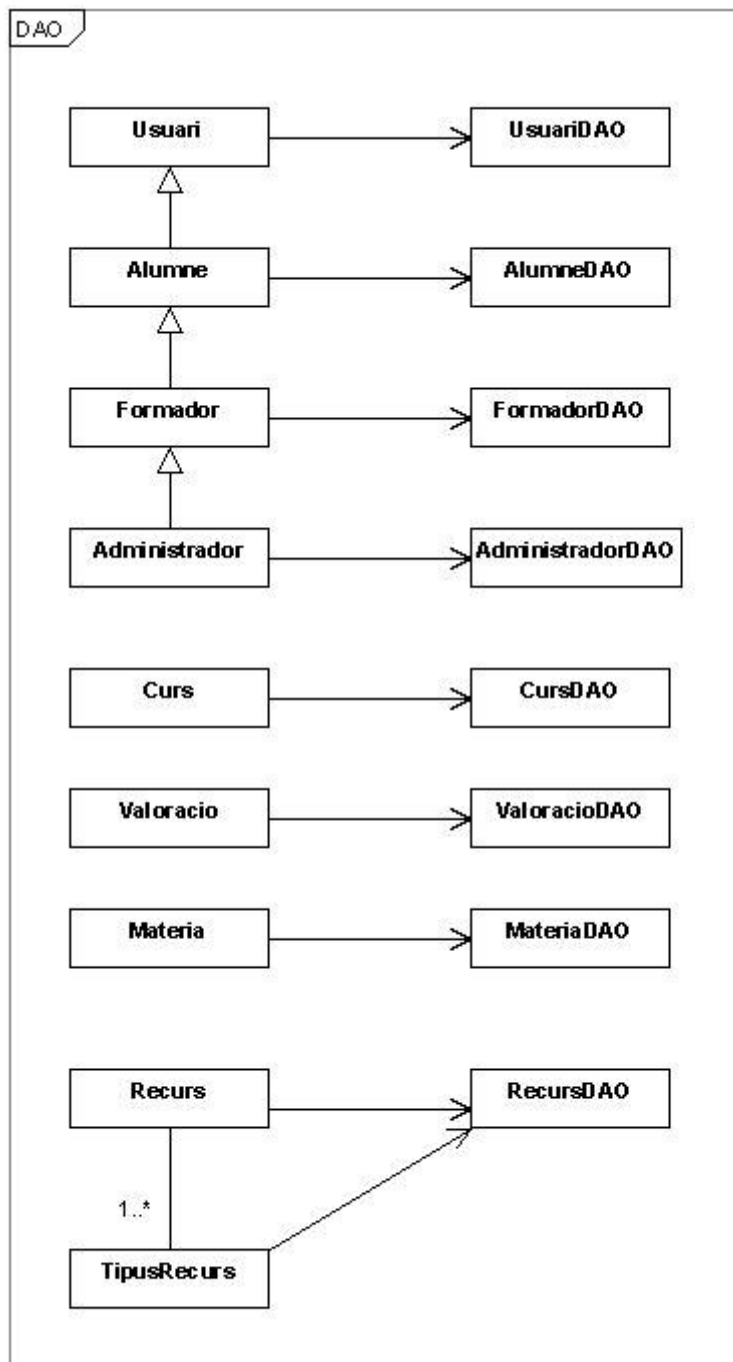


Classes del model

6.2 Classes d'accés a dades

Cada classe del model té la seva corresponent classe d'accés a dades. En el moment en què s'ha de fer qualsevol operació d'accés a dades, sigui del tipus que sigui (consulta, inserció, esborrat, modificació), els objectes del model acudirán a aquestes classes per a anar a tocar la base de dades físicament. És a dir, les classes del model són independents de la implementació de base de dades que es triï. Si mai s'opta per canviar de la base de dades, en el pitjor dels casos només s'hauran de tocar les classes d'aquest subapartat.

El diagrama es mostra a continuació.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Classes d'accés a dades

6.3 Classes del *framework* Struts

En aquest punt, inserim al nostre model de classes les classes que són part del *framework* Struts. Hi indiquem les classes de les quals heretem, i les classes que utilitzem.

Subdividim el conjunt de classes del *framework* Struts en diferents apartats, en funció de la seva funcionalitat:

- Accions (classes **Action*): Són aquelles classes que crida l'*ActionServlet*, i que nosaltres hem definit al fitxer *struts-config.xml*.

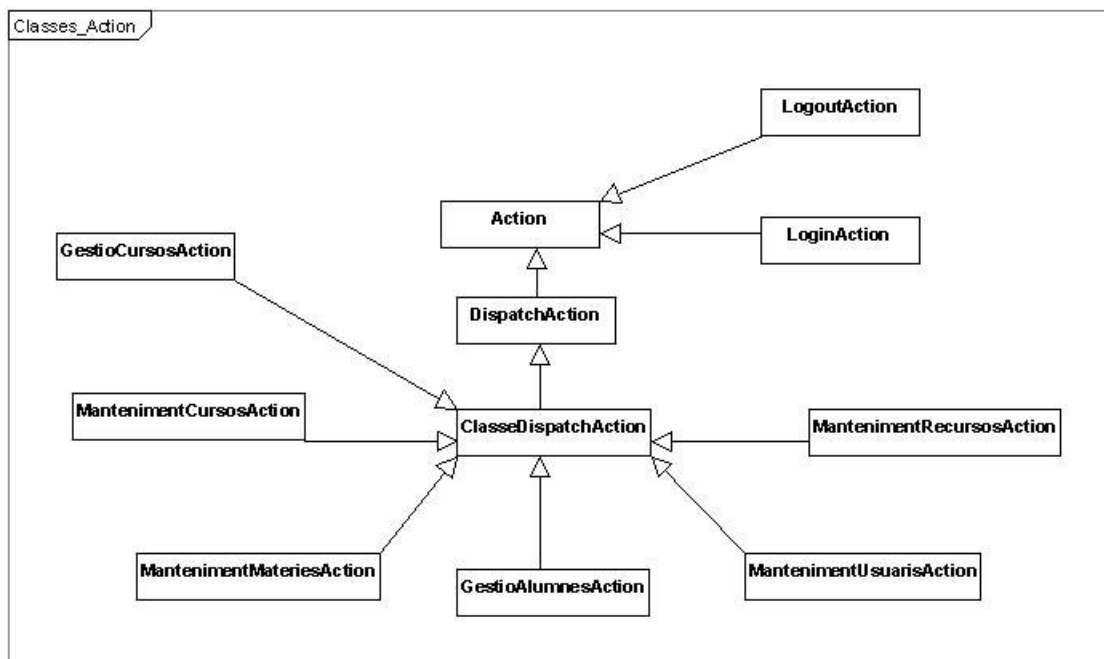
6.3.1 Classes *Action*

Les classes *Action* són, com hem dit, les que crida l'*ActionServlet*, com a resposta a peticions HTTP de l'usuari. Les nostres classes *Action* heretaran d'aquesta, però d'una manera indirecta, és a dir, a través de la classe *DispatchAction* d'Struts. Aquesta classe que fa redireccions, en funció de la URL sol·licitada, als mètodes corresponents, amb un comportament parametritzat en *struts-config.xml*. Per exemple, si el nostre fitxer *struts-config.xml* té una entrada com aquesta:

```
<action
    path="/mant_users"
    type="org.apache.struts.actions.DispatchAction"
    name="manteniment_usuaris"
    scope="request"
    parameter="metode"/>
```

Llavors, això vol dir que la nostra classe de tipus *DispatchAction*, i cridarà el mètode indicat al paràmetre mètode (http://xxx/app/mant_users.do?metode=afegir voldria dir que s'executarà el mètode *afegir()* de la nostra classe).

Amb això present, anem a presentar l'estructura de classes *Action* del sistema.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Classes *Action*

La classe *ClasseDispatchAction* s'ha implementat amb la única intenció d'agrupar algun tipus de funcionalitat, per exemple, control d'accés.

Totes aquestes classes han d'interactuar amb les classes del model, per tal de dur a terme les seves funcionalitats.

S'hi pot veure que la majoria de classes *Action* hereten, com hem dit, de *DispatchAction*. No té tant de sentit de fer això amb dues classes d'acció una mica especials, que són *LoginAction* i *LogoutAction*.

A continuació exposem els mètodes més rellevants d'aquestes classes, i els comentem.

6.3.1.1 MantenimentUsuarisAction

- **afegir()** Afegeix un usuari al sistema, a partir del formulari d'entrada proporcionat per l'Administrador. El formulari el recollirà Struts i n'obtindrà els paràmetres necessaris per a instanciar l'objecte de tipus *ActionForm* necessari (vegeu l'apartat sobre classes *ActionForm*).
- **eliminar()** Elimina l'usuari seleccionat.
- **modificar()** L'usuari triat és modificat segons el nou formulari de dades proporcionat per l'Administrador.
- **bloquejar()** L'usuari és bloquejat per l'Administrador.

6.3.1.2 MantenimentRecursosAction

- **afegirTipus()** S'afegeix un nou tipus de recurs.
- **afegir()** S'afegeix un nou recurs al sistema.
- **modificar()** Es canvia un recurs d'acord a les dades introduïdes.
- **eliminar()** Es treu del sistema el recurs.
- **bloquejar()** Es bloqueja el recurs.

6.3.1.3 GestioAlumnesAction

- **cercarCursos()** Es cercaran tots els cursos que compleixen amb les condicions de cerca imposades per l'usuari. Aquestes condicions de cerca estaran al formulari d'introducció de dades.
- **canviarPass()** L'usuari modificarà la seva contrassenya.
- **inscriure()** L'usuari actiu s'inscriu a un dels cursos.
- **desferInscripcio()** L'usuari s'esborra d'una inscripció.
- **veureDetallsCurs()** L'usuari veu els detalls d'un curs.
- **veureInscripcions()** L'usuari veu els cursos on s'ha inscrit.
- **valorarCurs()** L'usuari pot valorar un curs al qual ha assistit.

6.3.1.4 MantenimentMateriesAction

- **afegir()** S'afegeix una matèria a la base de dades.
- **eliminar()** S'elimina una matèria de la base de dades.

6.3.1.5 MantenimentCursAction

- **afegir()** S'afegeix un curs a la base de dades. Els detalls del curs venen del formulari d'entrada de dades, a través d'un objecte de tipus *ActionForm*.
- **modificar()** Es modifiquen detalls d'un curs, a través d'un formulari similar al d'alta de curs, vist al mètode anterior.
- **esborrar()** S'esborra un curs de la base de dades.

6.3.1.6 GestioCursosAction

- **assignar()** S'assigna un alumne a un curs, per tal de comptar-lo ja com a assistent al curs.
- **eliminarAssignacio()** S'esborra l'assignació d'un alumne a un curs.
- **reservar()** Es fa una reserva d'algun recurs, de manera que ara aquest recurs queda bloquejat, i lligat a aquest curs pel període de durada del curs.
- **eliminarReserva()** Es desfà la reserva del recurs indicat al mètode anterior.
- **veureDetalls()** Permet de consultar tots els detalls d'un curs.
- **veureValoracions()** Permet de veure les valoracions que els alumnes assistents hi han fet.
- **veureRecursos()** Permet de veure els recursos assignats al curs.
- **veureAssistents()** Permet de veure els alumnes que assisteixen o que han assistit al curs.
- **veureInscripcions()** Permet de veure els alumnes que s'hi han inscrit.
- **acceptarInscripcio()** Accepta una de les inscripcions, passant l'alumne inscrit a alumne assistent.
- **denegarInscripcio()** Denega una petició d'inscripció.

6.3.1.7 LoginAction

Aquesta classe no hereta de *DispatchAction*, i, per tant, el mètode bàsic és *execute()* (de la classe *Action*).

- **execute()** Aquest mètode realitza les funcions d'entrada al sistema.

6.3.1.8 LogoutAction

Igual que *LoginAction*, aquesta classe tampoc no hereta de *DispatchAction*, sinó d'*Action*, per la qual cosa el mètode clau és *execute()*.

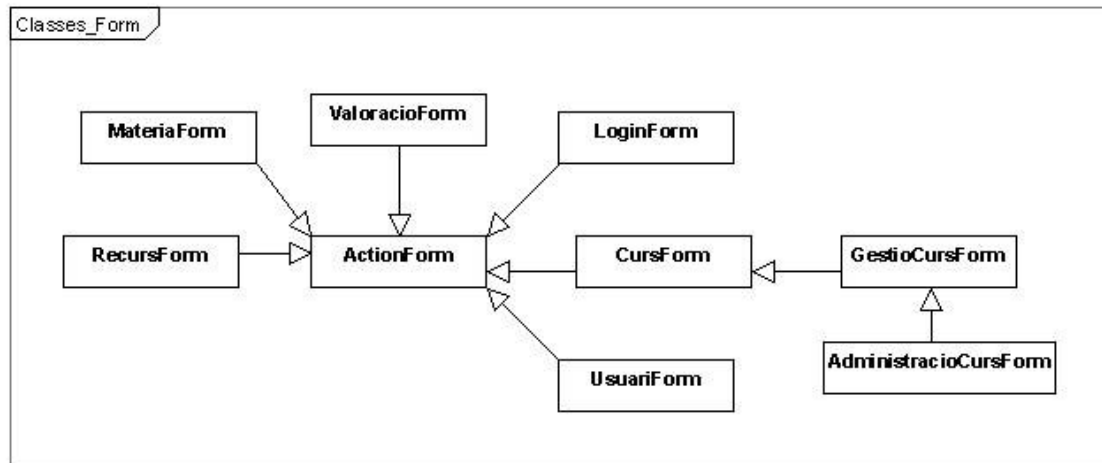
- **execute()** Aquest mètode realitza les funcions de sortida del sistema.

6.3.2 Classes *ActionForm*

Les classes *ActionForm* de Struts són unes classes que actuen de correspondència entre pantalles i classes Java. Això és una manera de representar la Vista en classes Java. Hi ha opinions diverses sobre si aquestes classes són part de la "C" o de la "V" del model MVC. La meua opinió és que són part de la Vista ("V"), ja que el que fan és un mapeig entre la pantalla (formulari) i una classe Java. Les classes *Action* obtenen, gràcies a Struts, una instància de la classe *ActionForm* corresponent (la correspondència entre una classe *Action* i la seva *ActionForm* s'ha definit prèviament al fitxer *struts-config.xml*). Per tant, són una manera que tenen les classes *Action* d'accedir als formularis que han introduït els usuaris, però d'una manera molt més "Object-Oriented" que a través dels paràmetres de l'objecte *HttpServletRequest*.

A més, les classes *ActionForm* tenen dos mètodes fonamentals, *reset()* i *validate()*, que són molt útils per a poder controlar les dades entrades pels usuaris. En el nostre cas, aquests mètodes s'implementaran **sempre**.

A continuació veiem un diagrama de les classes **Form* del nostre sistema.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Classes *Form*

Podem veure que *GestioCursForm* hereta de *CursForm*. Això és perquè *CursForm* és el formulari d'alumnes, mentre que *GestioCursForm* és el de Formadors. Passa el mateix entre *GestioCursForm* i *AdministracioCursForm*. Això es pot fer perquè cada vista és tan sols una extensió de la vista anterior, no canvia res, només afegeix comportaments.

7 Arquitectura de la Interfície (Vista)

Fins ara hem vist el model i el controlador, encara que una part del controlador jo opino que té un fort component de la vista (les classes *ActionForm* de Struts).

A partir d'ara ens centrem en la part de la vista, els formularis que componen la part de la interfície d'usuari, amb la qual interactuaran els destinataris de l'aplicació.

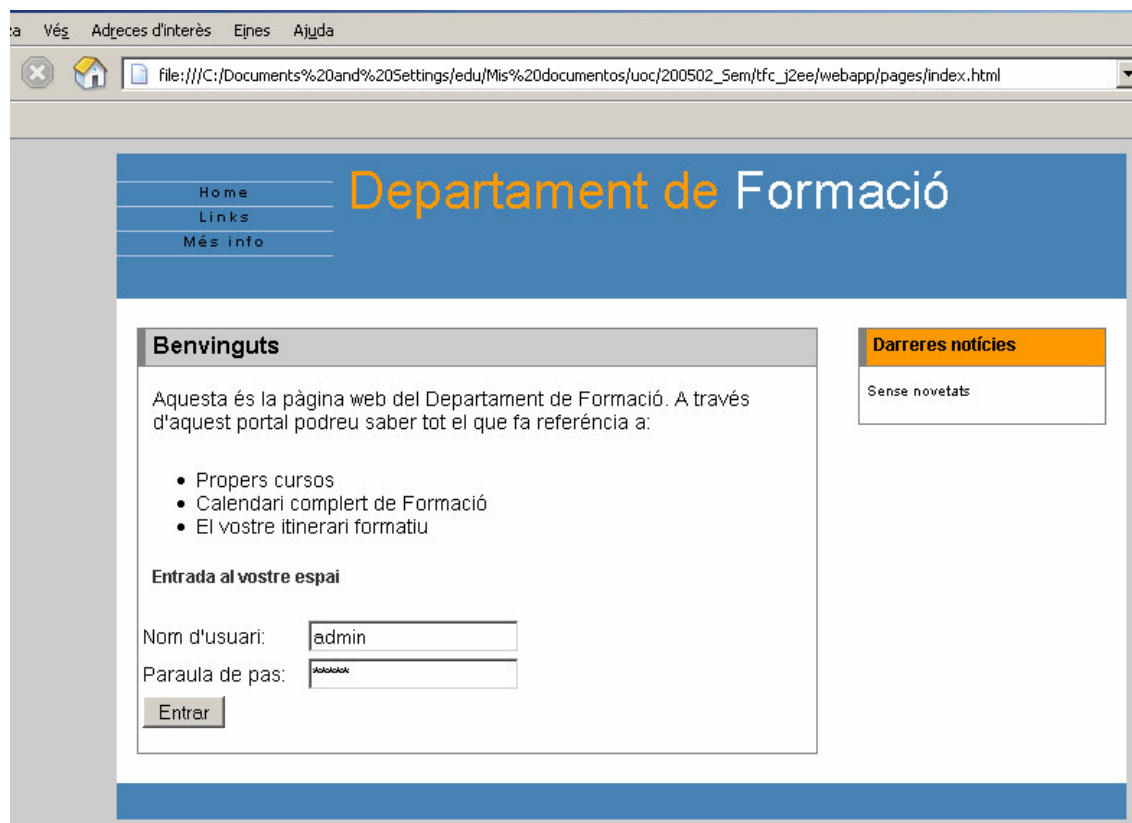
La vista es dissenya en JSP, fent ús, ocasionalment, de *custom tags* per a representar parts de la informació.

A continuació veiem les pantalles i les comentem.

7.1 Benvinguda (*index.jsp*)

Aquesta pantalla és la pantalla inicial. Podria no contenir codi JSP (i ser, consegüentment, de tipus *html*), però l'he dissenyada ja de tipus JSP per tal de tenir en compte si l'usuari introdueix combinacions usuari/contrassenya incorrectes, i llavors apareixerà un comentari avisant l'usuari de l'error.

El formulari que conté aquesta pantalla està relacionat amb la classe *LoginForm* indicada a l'apartat de les classes *ActionForm*.



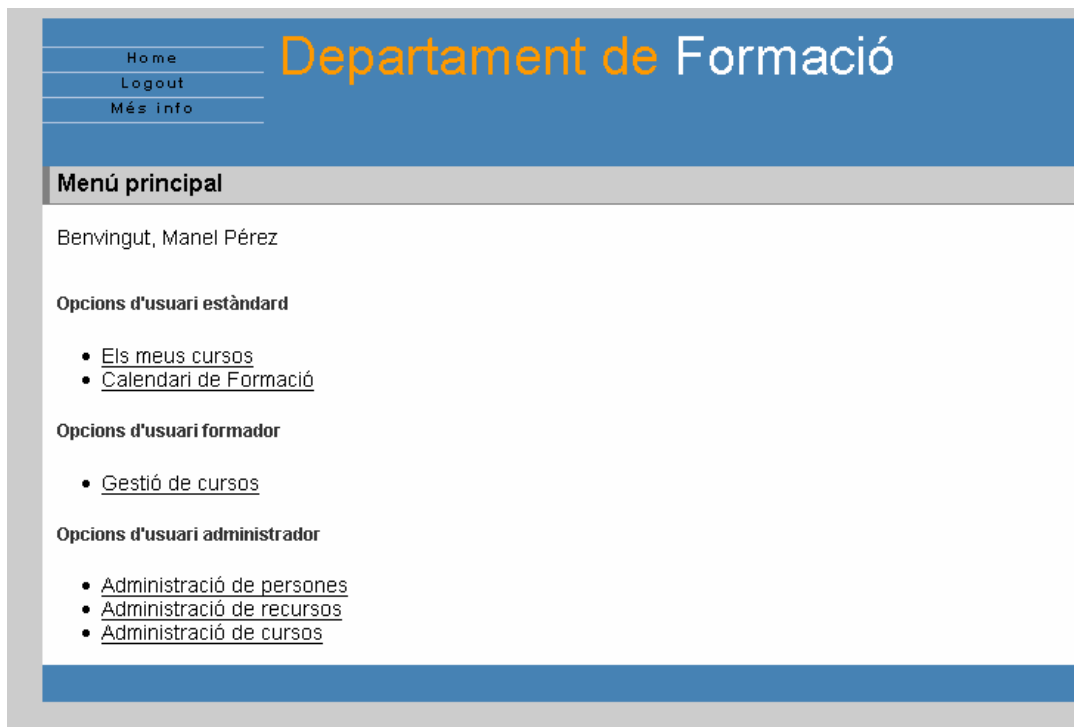
Pàgina de benvinguda (*index.jsp*)

7.2 Pantalla principal (*main.jsp*)

Aquesta pantalla és la que veu l'usuari després d'entrar al sistema. Com la resta de l'aplicació, conté dues àrees ben diferenciades, que són la capçalera, i la part de treball.

La capçalera és un *include* (el tag `<jsp:include>` de JSP) de la pàgina *capçalera.jsp*. La resta de la pàgina conté l'àrea de treball. En aquest cas, la pàgina mostrada és la pàgina

benvinguda.jsp. La resta de l'aplicació s'estructura exactament igual, en capçalera i àrea de treball, per la qual cosa, a partir d'ara, passarem a descriure tan sols l'àrea de treball.



Pàgina principal (*main.jsp*)

7.3 Pàgina de capçalera (*capcalera.jsp*)

Aquesta pàgina no es veurà mai per ella mateixa, sinó únicament en combinació amb alguna altra pàgina a l'àrea de treball. La capçalera inclourà un enllaç a l'acció de Logout, per tal de sortir del sistema. No té *ActionForm* associat, ja que no hi ha un formulari d'entrada de dades, i té poc sentit l'ús d'*ActionForms* amb pàgines sense formularis, tot i que es podria fer igualment.



Pàgina de capçalera (*capcalera.jsp*)

7.4 Pàgina de benvinguda (*benvinguda.jsp*)

Aquesta és la pàgina on arriba l'usuari després d'entrar. No té *ActionForm* associat, i tots els links són enllaços a noves funcionalitats. Cada una de les subzones ("usuari estàndard", "usuari formador" i "usuari administrador") són *includes* de pàgines diferents que veurem a continuació.

Benvingut, Manel Pérez

Opcions d'usuari estàndard

- [Els meus cursos](#)
- [Calendari de Formació](#)

Opcions d'usuari formador

- [Gestió de cursos](#)

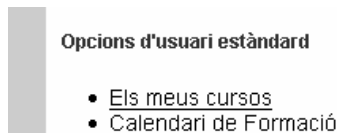
Opcions d'usuari administrador

- [Administració de persones](#)
- [Administració de recursos](#)
- [Administració de cursos](#)

Pàgina de benvinguda (*benvinguda.jsp*)

7.5 Usuari estàndard (estandard.jsp)

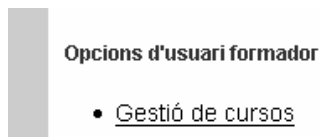
Aquesta és una subpantalla que contè tan sols els links a les funcionalitats de l'usuari estàndard. És la primera part de la pantalla de benvinguda. Tothom veu aquesta part.



Pàgina de l'usuari estàndard (*estandard.jsp*)

7.6 Usuari formador (formador.jsp)

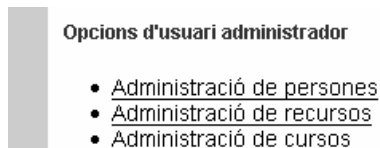
Aquesta és una subpantalla que contè els links a les funcionalitats de l'usuari formador. És la segona part de la pantalla de benvinguda. Mitjançant tags condicionals de Struts (tags *<logic>*) decidirem si la mostrem o no en funció del perfil de l'usuari actiu.



Pàgina de l'usuari formador (*formador.jsp*)

7.7 Usuari administrador (administrador.jsp)

Aquesta és una subpantalla que contè els links a les funcionalitats de l'usuari formador. És la tercera part de la pantalla de benvinguda. Mitjançant tags condicionals de Struts (tags *<logic>*) decidirem si la mostrem o no en funció del perfil de l'usuari actiu.



Pàgina de l'usuari administrador (*administrador.jsp*)

7.8 Formulari de cursos (form_curs.jsp)

Aquesta pantalla, lligada a la classe *AdministracioCursForm*, és la que permet que un Administrador doni d'alta un curs al sistema, o bé que en modifiqui detalls. Els formadors també el poden utilitzar per les inscripcions provisionals.

Detalls del curs					
Codi de curs	<input type="text" value="12321"/>				
Descripció	<input type="text" value="Introducció a l'economia"/>			Matèria	<input type="text"/>
Inici	<input type="text" value="11/11/2005"/>	Final	<input type="text" value="11/11/2005"/>	Hores	<input type="text" value="8"/>
Detalls ampliats del curs					
Formadors		Inscrits		Recursos	
Rafel Martos	✗	Maria Puertas	✗	Projector	✗
Albert Garcia	✗	Albert Ràfols	✗	Aula 2 - Oficines centrals	✗
				Micròfon	✗
<u>Afegir formador...</u>		<u>Afegir alumne...</u>		<u>Afegir recurs...</u>	
Inscripcions provisionals					
Jordi Garcia			✓	⊖	
Manel Sabaté Pérez			✓	⊖	

Pàgina de detalls del curs (*form_curs.jsp*)

Quan l'Administrador vol modificar els detalls d'un curs, veurà aquesta mateixa pantalla, amb els detalls omplerts, i que el podrà canviar.

Si és l'alumne el que hi entra, no pot veure certes opcions, que són les d'afegir o eliminar formadors, alumnes inscrits i recursos, i tampoc no veurà les inscripcions provisionals.

En canvi, els formadors no poden afegir formadors ni eliminar-los, i, per tant, tampoc no veuran aquestes opcions. El fet de mostrar aquestes opcions es controlarà a través dels tags condicionals de Struts, ja que no considero que valgui la pena de crear una pàgina a incloure tan sols per aquestes opcions tan simples.

7.9 Formulari de matèries (*form_materia.jsp*)

Aquesta pantalla serveix per a donar d'alta o modificar matèries existents. L'*ActionForm* associat és *MateriaForm*.

Alta de nova matèria	
Codi de la matèria	<input type="text"/>
Nom de la matèria	<input type="text"/>
<input type="button" value="Acceptar"/> <input type="button" value="Restaurar"/>	

Formulari de matèries (*form_materia.jsp*)

7.10 Formulari de recursos (*form_rekurs.jsp*)

Aquesta pantalla, lligada a la classe *RecursForm*, servirà per a donar d'alta i/o modificar recursos.

Alta de nou recurs

Codi del recurs

Nom del recurs

Bloquejat?

Formulari de recursos (*form_rekurs.jsp*)

7.11 Formulari de persones (*form_persona.jsp*)

Aquesta pantalla és la que s'utilitza per a donar d'alta o modificar dades de les persones. La classe relacionada és *UsuariForm*.

Alta de nova persona

Codi d'usuari

Nom

Cognoms

Perfil

Adreça

Departament

Observacions

Formulari de persones (*form_persona.jsp*)

7.12 Baixa de persones (*baixa_persona.jsp*)

Aquesta pantalla no és estrictament un formulari, sinó simplement un llistat de persones, amb l'opció de treure-les del sistema.

Baixa de persona

Usuaris disponibles

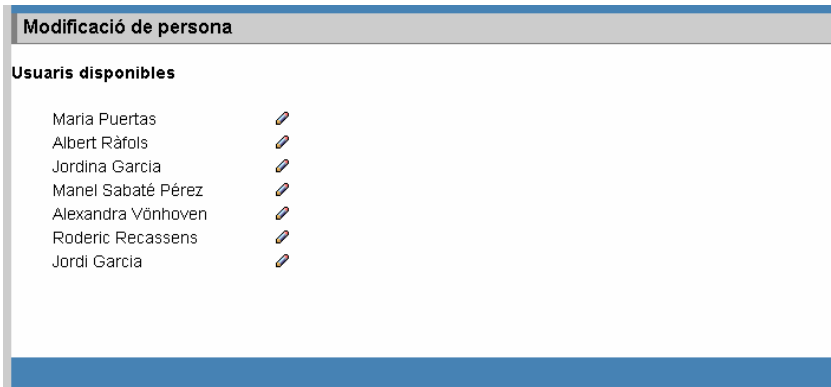
Maria Puertas	×
Albert Ràfols	×
Jordina Garcia	×
Manel Sabatè Pérez	×
Alexandra Vónhoven	×
Roderic Recassens	×
Jordi Garcia	×

Formulari de baixa de persones (*baixa_persona.jsp*)

La petició de baixa es farà sobre la classe *MantenimentUsuarisAction*, que serà la que avaluarà la petició i la traslladarà als objectes del model, que, al seu torn, utilitzaran les classes DAO per tal de fer el canvi a la Base de Dades.








7.13 Modificació de persones (modifica_persona.jsp)

Aquesta pantalla és com l'anterior, amb la diferència que el click sobre la icona de modificació es limita a fer una redirecció cap als detalls de la persona, mostrant el formulari de persones (*form_persona.jsp*).



Modificació de persona

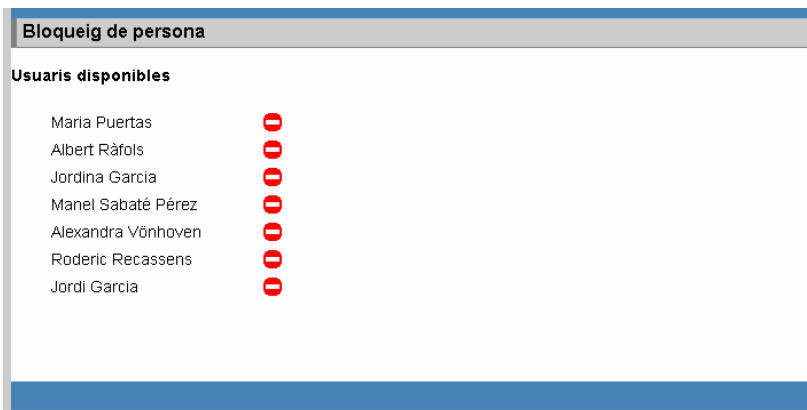
Usuaris disponibles

Maria Puertas	
Albert Ràfols	
Jordina Garcia	
Manel Sabatè Pérez	
Alexandra Vónhoven	
Roderic Recassens	
Jordi Garcia	

Formulari de modificació de persones (*modifica_persona.jsp*)








7.14 Bloqueig d'usuari (bloqueig_persona.jsp)

Aquesta pantalla també és com l'anterior, però ara la icona és diferent, i l'efecte és el de fer que una persona ja no pugui tornar a entrar al sistema.



Bloqueig de persona

Usuaris disponibles

Maria Puertas	
Albert Ràfols	
Jordina Garcia	
Manel Sabatè Pérez	
Alexandra Vónhoven	
Roderic Recassens	
Jordi Garcia	

Formulari de bloqueig d'usuari (*bloqueig_persona.jsp*)

7.15 Llista de cursos (llista_cursos.jsp)

Aquest llistat de cursos permet de veure tots els cursos. Tothom el pot fer servir, però amb propòsits diferents, segons els seus perfils.

Els meus cursos - Manel Pérez							
Codi	Nom	Inici	Final	Hores	Tema	Inscrit	Obert
12321	Introducció a l'economia	10/10/2005	10/10/2005	8	General	No	Sí
23213	Aprendre a parlar en públic	20/10/2005	27/10/2005	40	General	Sí	No
23221	Introducció als productes d'assegurances	31/10/2005	31/10/2005	8	General	No	No
33211	Les noves tecnologies	04/11/2005	04/11/2005	8	General	No	No

Llista de cursos (*llista_cursos.jsp*)

Quan l'usuari fa click a un dels cursos, pot veure'n els detalls, i, en funció del seu perfil, podrà fer-hi més o menys coses (inscriure-s'hi, afegir/eliminar alumnes, reservar recursos, etc.).

8 Consideracions tècniques

Un dels motius que m'han mogut a realitzar el projecte en la plataforma J2EE ha estat la motivació que tinc per realitzar aplicacions multiplataforma, que utilitzin el màxim de components *opensource* possible, i que no lliguin el client a un sistema operatiu i/o maquinari específics. És ben sabut que Java el trobem en multitud de sistemes i arquitectures, per la qual cosa penso que és l'entorn més apropiat.

A continuació s'exposen les característiques tècniques de l'aplicació, juntament amb els requisits de maquinari i/o programari, indicant la plataforma on s'ha dut a terme el desenvolupament de l'aplicació.

8.1 Programari

L'aplicació és una aplicació de tipus web, per la qual cosa serà possible de ser utilitzada des de qualsevol entorn amb un navegador que respecti els estàndards.

8.1.1 Pàgines web

Les pàgines web validen totes elles l'estàndard *XHTML 1.0 Transitional*.

Seguint amb la meua filosofia de reutilització de components *opensource*, el disseny visual ha estat extret del lloc web <http://www.oswd.org>, i han estat realitzades per KC R (<http://www.digital-web-designs.com>).

8.1.2 Servidor d'aplicacions

Per al servidor d'aplicacions s'ha utilitzat la versió 5.0.28 de Jakarta Tomcat (<http://tomcat.apache.org/tomcat>).

8.1.3 Servidor de base de dades

El servidor de base de dades que s'ha utilitzat és MySQL 5.0 (<http://www.mysql.com>). Aquesta és la darrera versió d'aquest conegut motor de bases de dades *opensource*, i incorpora novetats tan necessàries com els *triggers*, *stored procedures*, i d'altres eines que ja fan que MySQL sigui una solució perfectament professional.

8.1.4 Sistema operatiu

El sistema operatiu on pot córrer la plataforma és molt variable, a l'elecció del client, en funció de les seves disponibilitats de maquinari i/o llicències de programari. Tots els components funcionaran correctament en qualsevol sistema operatiu on hi puguem instal·lar els components prèviament indicats. Per il·lustrar aquesta circumstància, només direm que el desenvolupament s'ha realitzat en diferents màquines obtenint en totes elles un sistema perfectament funcional (Windows XP Home, Debian Linux amb kernel 2.6.10, FreeBSD 6.0). Personalment, m'inclino per qualsevol versió estable de Debian Linux (<http://www.debian.org>).

La plataforma Java de desenvolupament ha estat NetBeans 4.1 i NetBeans 5.0 Beta 1 i 2 (<http://www.netbeans.org>).

8.2 Maquinari

El maquinari necessari és el que calgui per executar sense problemes els sistemes operatius indicats prèviament. Si hi ha prevista una càrrega d'usuaris notable (a partir de

50 sessions simultànies) seria important de tenir una quantitat notable de memòria RAM (no inferior a 2GB).

9 Conclusions

Com a tancament d'aquesta memòria només indicarè que el Treball de Fi de Carrera que ha suposat aquest projecte ha estat molt interessant, ja que he pogut experimentar en totes les fases del cicle de vida del desenvolupament d'un producte de programari. Normalment, en la vida "real", quan he participat en projectes similars, sempre m'havia ocupat d'alguna de les fases del projecte, però mai de tot, des del principi fins al final. Penso que és molt important de fer tot el conjunt de tasques per tal d'adonar-se'n de la complexitat que suposa qualsevol projecte de programari, per petit que sigui.

9.1 Línies futures de treball

En el futur l'aplicació és ampliable de moltes maneres, en funció de les necessitats del client. Jo proposaria els següents àmbits:

- Inserció de fòrums / xats
- Enllaç amb l'aplicació ERP de l'empresa
- Ampliació dels perfils d'usuaris
- Parametrització màxima de l'aplicació, de manera que es pugui vendre l'aplicació a d'altres empreses amb un mínim esforç de parametrització.

10 Glossari

Administració de cursos: eina que tenen els administradors de l'aplicació per a poder crear / modificar cursos.

Administració de persones: eina que tenen els administradors de l'aplicació per tal d'efectuar el manteniment de les persones que utilitzen l'aplicació.

Administració de recursos: eina que tenen els administradors de l'aplicació per tal de fer el manteniment dels recursos de l'aplicació.

Administrador: usuari de l'aplicació amb tots els permisos, pot crear i modificar qualsevol entitat de l'aplicació. Al mateix temps, un Administrador també és un Alumne i un Formador.

Alumne: usuari de l'aplicació que pot assistir a cursos. És l'usuari per al qual està pensada l'aplicació. Pot veure els calendaris de cursos, inscriure-s'hi, etc.

Anàlisi: fase del desenvolupament de l'aplicació que consisteix en la identificació dels requeriments funcionals del sistema.

Bloqueig: funcionalitat que tenen els administradors per a evitar que un usuari segueixi entrant. També es pot activar el bloqueig d'un usuari de manera automàtica quan s'excedeix el nombre de reintents fallits màxim (3).

Client: l'empresa que vol implementar la solució de programari desenvolupada, per tal de fer més àgil el seu departament de formació.

Curs: l'entitat bàsica de l'aplicació, un curs té unes matèries, uns recursos, uns formadors, uns alumnes, unes dates, uns temes, etc.

Desenvolupador: la persona que ha dut a terme la implementació de l'aplicació.

Diagrama de Gantt: diagrama que permet de veure a "vista d'ocell" la temporització del projecte, amb les fases ben identificades i les dependències entre elles, si és el cas.

Disseny: fase del desenvolupament de l'aplicació en la qual s'explicita com es resoldrà la implementació dels requeriments indicats al document d'anàlisi.

Formació: activitat bàsica en tota empresa actual, sense la qual es perd competitivitat, i que, per tant, cal reforçar en tots els aspectes, també en la gestió de cursos, que és el que pretèn aquesta aplicació.

Formador: usuari del sistema que imparteix la formació. És al mateix temps un usuari de tipus "Alumne". Pot bloquejar recursos i admetre o no les pre-inscripcions dels alumnes. És l'Administrador de l'aplicació el qui decideix quins formadors s'assignen a un curs.

Framework: estructura de programari que s'integra en un entorn de producció (el servidor d'aplicacions en el nostre cas) i que resol un problema concret d'una manera ben provada i documentada (en el cas que ens ocupa, la part del Controlador en un sistema amb arquitectura MVC).

Gestió de cursos: és l'activitat bàsica que duen a terme els usuaris Formadors.

Implementació: la fase del projecte en la qual es fan efectius aquells requisits que s'han indicat al document de Disseny.

Interfície gràfica d'usuari: allò que l'usuari veu, les pàgines web en el cas de la nostra aplicació.

J2EE: plataforma Java per al desenvolupament d'aplicacions distribuïdes, molt adequada per a les solucions client – servidor basades en web, com és el nostre cas.

MVC: patró de disseny molt útil per a la realització de solucions de programari com la que ens ocupa. Distingeix entre capes (Model – Vista – Controlador) independents i poc lligades (la terminologia anglosaxona ho anomena “*loose coupling*”), de manera que es pot substituir algun element del sistema sense afectar gaire o gens els altres.

MySQL: servidor de bases de dades *opensource* altament professional, que assegura unes prestacions excel·lents en un entorn web.

Prototip: demostració no funcional de l'aplicació. Ve a ser una maqueta que ajuda a fer-se una idea del *look & feel* que tindrà el sistema un cop desenvolupat. Ajuda al client a refinar els requeriments, i al desenvolupador a tenir més clars els objectius desitjats.

Recurs: qualsevol recurs que pugui ser necessari en una acció de formador (aules, projectors, materials, etc.)-

Reintents fallits: intents d'accès al sistema sense èxit per part d'un usuari.

Struts: *framework* de tipus *opensource* que resol satisfactòriament la part del Controlador en una arquitectura MVC.

SuperUsuari: usuari del sistema que és un Administrador amb uns permissos especials, que impliquen que aquest usuari mai no pot ser esborrat, si no és accedint directament a la Base de Dades.

Tipus de recurs: tipologia d'un recurs (aula, per exemple). Un recurs pot ser de diferents tipus al mateix temps.

Tomcat: servidor d'aplicacions J2EE *opensource* sobre el que corre l'aplicació.

11 Bibliografia

STELTING, Stephen i MAASSEN, Olav. *Applied Java Patterns*. Palo Alto, California, The Sun Microsystems Press Java Series, 2002

HUSTED, Ted, *Struts in Action*. Greenwich, CT, Manning Publications Co, 2002

FALKNER, Jayson et al., *Desarrollo Web con JSP*. Madrid, Anaya Multimedia, 2002

ALLAMARAJU, Subrahmanyam, et al., *Professional Java Server Programming J2EE 1.3 Edition*. Birmingham, UK. Wrox Press Ltd., 2001

AKIF, Mohammad, et al., *Java y XML*. Madrid. Anaya Multimedia, 2003

BRADENBAUGH; Jerry, *Aplicaciones JavaScript*. Madrid. Anaya Multimedia, 2000

GILFILLAN, Ian, *Mastering MySQL 4*. Alameda, California. Sybex, 2003

12 Annexos

A aquest document de memòria s'hi adjunten els següents documents annexos:

- Arxiu .zip amb el codi font de les classes java (*src.zip*) i el fitxer de muntatge amb l'eina ANT (*build.xml*).
- Arxiu .zip amb el *JavaDoc* del codi font de l'aplicació (*formacio_docs.tar.gz*).
- Arxiu .sql amb l'script de creació de les taules de la base de dades (*formacio.sql*).
- Arxiu .war amb l'aplicació web per instal·lar-la a un servidor d'aplicacions J2EE-compliant (*formacio.war*).
- Arxiu .txt amb el manual d'instal·lació (*formacio_install.txt*)
- Fitxer .jar amb el *Driver* d'accés al MySQL (*mysql-connector-java-3.1.12-bin.jar*)
- Fitxer .ppt amb la presentació de diapositives del projecte (*ecapell_presentacio.ppt*).