J2EE de la teoria a la pràctica: Enquestes de satisfacció de la docència

Olga Vendrell i Balcells
Enginyeria Informàtica

Verònica Peña Pastor

15 de gener de 2007

Resum i paraules clau

Resum

J2EE de la teoria a la pràctica: “Enquestes de satisfacció de la docència”.

Aquest projecte final de carrera pertany a l’àrea J2EE (Java 2 Enterprise Edition), una plataforma de desenvolupament d’aplicacions empresarials distribuïdes i bàsicament en format web.

El seu principal objectiu, tal i com indica el seu títol, és ampliar els coneixements sobre aquesta àrea tant a nivell teòric com pràctic. Per fer-ho es divideix en 3 parts clarament diferenciades però dependents cadascuna de l’anterior.

En primer lloc cal fer un estudi de diferents frameworks web que hi ha actualment al mercat. D’entre un gran ventall de possibilitats se n’analitzen 3 amb detall (MyFaces, Spring MVC i Struts) i se n’escull un per a ampliar-lo: MyFaces.

En segon lloc, s’aprofundeix en el framework seleccionat i es desenvolupen nous components que permeten ampliar les seves funcionalitats.

Finalment, es desenvolupa una petita aplicació de mostra, “Enquestes de satisfacció de la docència” que utilitza MyFaces com a framework web de la capa de presentació i incorpora els components implementats.

Paraules clau

J2EE

Framework web

Capa presentació

MyFaces

Components JSF

Índex

Índex de continguts

Resum i paraules clau 3
Resum 3
Paraules clau 3
Índex 5
Índex de continguts 5
Índex de figures 6
Índex de taules 7
1. Introducció 8
1.1. Justificació i context en el qual es desenvolupa 8
1.2. Objectius 8
1.3. Enfocament i mètode seguit 10
1.4. Planificació 11
1.5. Productes obtinguts 17
1.6. Organització de la memòria 17
2. Estudi de frameworks de la capa de presentació 19
2.1. Introducció 19
2.2. Relació de frameworks existents al mercat 20
2.3. MyFaces 22
2.4. Spring MVC 25
2.5. Struts 27
2.6. Estudi comparatiu 30
2.7. Conclusions 32
2.8. Framework a ampliar: MyFaces 34
3. Ampliació d'un framework: MyFaces 36
3.1. Anàlisi 36
3.2. Disseny 38
3.3. Implementació 43
3.4. Dificultats 44
4. Desenvolupament d'una aplicació de mostra 46
4.1. Anàlisi 46
4.2. Disseny 49
4.3. Base de dades 55
4.4. Implementació 56
4.5. Dificultats 57
5. Seguiment i valoració econòmica 59
5.1. Seguiment 59
5.2. Valoració econòmica 60
6. Conclusions 62
7. Línies de futur 64
8. Glossari 65

9. Bibliografia ... 67
9.1. Llibres ... 67
9.2. Documents electrònics ... 68
9.3. Llocs web ... 68
Annexos ... 71
Annex A: Guia d'instal·lació i execució de l'aplicació ... 71
Annex B: Model d'enquesta de satisfacció de la docència ... 73
Annex C: Disseny físic de la base de dades ... 74

Índex de figures

Figura 1.1:Mapa conceptual del Procés Unificat de Rational ... 11
Figura 1.2:Diagrama de Gantt ... 16
Figura 2.1:Model MVC que implementa MyFaces ... 22
Figura 2.2:Model MVC que implementa Spring MVC ... 25
Figura 2.3:Model MVC que implementa Struts ... 27
Figura 3.1:Ampliació de MyFaces ... 36
Figura 3.2:Mostra del component *LlistaDoble* ... 37
Figura 3.3:Mostra del component *TaulaDades* ... 37
Figura 3.4:Mostra del component *Opcions* ... 37
Figura 3.5:Estructura de classes d'un component de MyFaces ... 38
Figura 3.6:Jerarquia del component *LlistaDoble* ... 39
Figura 3.7:Jerarquia del component *TaulaDades* ... 39
Figura 3.8:Jerarquia del component *Opcions* ... 40
Figura 3.9:Estructura del component *LlistaDoble* ... 40
Figura 3.10:Estructura del component *TaulaDades* ... 40
Figura 3.11:Estructura del component *Opcions* ... 41
Figura 3.12:Elements del component *LlistaDoble* ... 41
Figura 3.13:Elements del component *Opcions* ... 42
Figura 4.1:Diagrama del concepte de negoci ... 47
Figura 4.2:Diagrama de casos d'ús del mòdul de Gestió de la informació ... 48
Figura 4.3:Diagrama de casos d'ús del mòdul de Realització d'enquestes ... 48
Figura 4.4:Arquitectura hardware ... 49
Figura 4.5:Arquitectura software ... 50
Figura 4.6:Diagrama de flux de les pantalles del mòdul de Gestió de la informació ... 51
Figura 4.7:Estructura bàsica de totes les pantalles del mòdul Gestió de la informació ... 51
Figura 4.8:Pantalla d'identificació de l'usuari ... 52
Figura 4.9:Pantalla de gestió de dades ... 52
Figura 4.10:Pantalla d'alta/modificació de dades ... 53
Figura 4.11:Diagrama de flux de pantalles del mòdul Realització d'enquestes ... 53
Figura 4.12:Pantalla de selecció de l'enquesta ... 54
Figura 4.13:Pantalla de realització de l'enquesta ... 54
Figura 4.14:Model de dades Entitat-Relació ... 55
Figura 4.15:Codi exemple d'utilització d'un component en un document JSP ... 56

Índex de taules

Taula 1.1:Temporització prevista 14
Taula 2.1:Taula resum estudi frameworks 30
Taula 2.2:Taula comparativa frameworks 32
Taula 3.1:Components a estendre per a desenvolupar els propis 39
Taula 3.2:Controls HTML del component *LlistaDoble* 43
Taula 3.3:Controls HTML del component *Opcions* 44
Taula 4.1:Distribució dels components a l'aplicació de mostra 57
Taula 5.1:Temporització prevista vs temporització real 60
Taula 5.2:Valoració econòmica 61

1. Introducció

1.1. Justificació i context en el qual es desenvolupa

Aquest projecte final de carrera pertany a l'àrea J2EE (Java 2 Enterprise Edition) i, més concretament, està definit sota la proposta de fer un disseny i implementació d'un marc de treball (framework) de presentació per a aplicacions J2EE.

J2EE és una plataforma de desenvolupament d'aplicacions distribuïdes per a entorns empresarials. És força complexa i molt completa, en part, perquè utilitza diverses i variades tecnologies com servlets, JSP, XML, EJB... La gran majoria d'aplicacions implementades actualment amb J2EE tenen format web, cosa que facilita la seva distribució entre els clients (entre d'altres coses).

Per a obtenir un major rendiment tant en el desenvolupament com en l'explotació de les aplicacions, J2EE es basa en una arquitectura estructurada en capes relacionades entre sí però que mantenen certa independència. Aquestes són: capa de presentació (interfície usuari-aplicació), capa de negoci (contingut de la lògica de negoci) i capa d'accés a dades (connexió amb els sistemes d'informació).

Degut a l'extens abast d'aquest tema, s'ha decidit acotar el projecte i centrar-lo fonamentalment en una d'aquestes capes, la de presentació, una de les més importants i a la que a priori els desenvolupadors hi dediquen més esforços.

És precisament per ajudar als desenvolupadors del programari, que han anat sorgint diferents propostes de frameworks que faciliten la implementació de la capa on l'usuari interacciona amb l'aplicació.

Aquest projecte vol fer un estudi de les opcions més conegudes i utilitzades actualment. Vol veure quines possibilitats ofereix el mercat i analitzar amb un cert detall aquelles que poden ser més interessants per als desenvolupadors.

I no pretén quedar-se aquí, sinó que vol anar un pas més enllà. Després de seleccionar un dels marcs estudiats, vol fer-ne una extensió, afegint alguna funcionalitat que es detecti que no té i que pot ser d'utilitat a l'hora de desenvolupar aplicacions web. Per fer-ho cal aprofundir en l'estructura i el funcionament el framework en qüestió, tant a nivell teòric com pràctic.

Finalment, per tal de mostrar el seu funcionament, es procedirà a desenvolupar una aplicació d'exemple, titulada "Enquestes de satisfacció de la docència".

Aquesta aplicació està pensada per a resoldre un problema real d'una empresa concreta. Es tracta d'una Escola Universitària on, cada final de semestre, es passen enquestes als alumnes perquè valorin tant les assignatures com els professors. A partir de les dades recollides, s'elaboren diverses estadístiques i se n'extrauen conclusions. Actualment, aquest procés es fa de forma manual. El que aquí es planteja és desenvolupar una aplicació que permeti gestionar les enquestes, que l'alumne les pugui respondre des d'un laboratori del centre o des de casa i que automàticament s'enregistrin els resultats a una base de dades.

1.2. Objectius

Bàsicament, l'objectiu principal d'aquest projecte és ampliar els coneixements sobre la plataforma J2EE i les tecnologies que la formen, adquirint al mateix temps certa agilitat a l'hora de portar aquests coneixements a la pràctica.

Però, com s'ha dit abans, la plataforma J2EE és molt complexa i extensa, per la qual cosa, es pretén centrar el projecte en un dels seus aspectes més importants i de més ressò actualment: la capa de presentació de les aplicacions web.

L'objectiu marcat es pot aconseguir de moltes maneres. La proposta feta en aquest cas es pot concretar definint 3 grans objectius fonamentals més específics:

- **Objectiu 1:** Fer un estudi dels frameworks orientats a aplicacions web de la capa de presentació existents actualment.

  Això implica, per una banda, conèixer quines propostes hi ha al mercat. I per una altra, fer una valoració de les més conegudes i utilitzades. Aquesta valoració passa per descobrir les seves característiques, funcionalitats... i fer alguna prova a nivell pràctic (instal·lació, tutorial o exemple...).

- **Objectiu 2:** Dissenyar i implementar noves funcionalitats per a un marc de treball de la capa de presentació.

  Es tracta de triar justificadament un dels frameworks estudiats i, veient les seves mancances, estendre'l. Per això s'afegiran noves funcionalitats que facilitin la tasca del desenvolupador, facin més robustes les aplicacions o aportin qualsevol altre avantatge.

- **Objectiu 3:** Desenvolupar una aplicació que mostri com funciona el nou framework obtingut.

  L'aplicació escollida és "Enquestes de satisfacció de la docència". És fonamental que quedi clar com s'utilitza i funciona el marc generat.

Cadascun d'aquests objectius també es pot desglossar en altres més específics, per tal de definir amb més detall i claredat allò que es vol aconseguir en aquest projecte.

Però cal tenir present que el projecte s'ha de portar a terme en un temps determinat, contemplat al capítol de planificació. Per aquest motiu es deixa oberta la possibilitat d'aprofundir més o menys en algun dels subojectius (alguns d'ells poc concrets) per tal de poder complir els 3 objectius generals en les dates establertes i aconseguir així, concloure amb èxit el projecte.

A continuació es mostra un esquema de tots els objectius que es proposen:

- Ampliar els coneixements sobre la plataforma J2EE i les tecnologies que la formen.
  - Fer un estudi dels frameworks orientats a aplicacions web de la capa de presentació existents actualment.
    - Cercar i recopilar informació sobre el tema en els mitjans i formats disponibles (llibres, manuals, tutorials, articles...). De tota ella, seleccionar-ne les fonts més útils.
    - Tractar la informació recollida (llegir-la, entendre-la...).
    - Sintetitzar aquesta informació i extraure'n conclusions.
    - Escollir un framework per a treballar-hi amb més profunditat.
    - Documentar aquest estudi.
  - Dissenyar i implementar noves funcionalitats per a un marc de treball de la capa de presentació.
    - Analitzar un framework existent i detectar-ne alguna mancança.
    - Ampliar aquest framework dissenyant i implementant alguna nova funcionalitat.
  - Desenvolupar una aplicació que mostri com funciona el nou framework obtingut.
    - Fer l'anàlisi, disseny i implementació d'una aplicació amb J2EE amb accés a bases de dades que utilitzi el framework ampliat.
    - Integrar les noves funcionalitats implementades en aquesta aplicació.
- Portar a terme el projecte complint el calendari previst.

1.3. Enfocament i mètode seguit

El mètode seguit per a portar a terme aquest projecte consisteix, en un primer terme, a dividir-lo en fases:

- Primera fase

  Elaboració del pla de treball. Descripció del projecte, definició d'objectius, planificació temporal...

- Segona fase

  Realització de l'estudi dels frameworks web de la capa de presentació amb l'elaboració de l'informe corresponent.

- Tercera fase

  Ampliació del framework escollit. Anàlisi, disseny i implementació de noves funcionalitats.

- Quarta fase

  Desenvolupament d'una aplicació de mostra on s'utilitzi el framework ampliat: "Enquestes de satisfacció de la docència".

- Cinquena fase

  Elaboració de la documentació del projecte: memòria tècnica i presentació.

El seu enfocament té dues vessants clarament diferenciades: una teòrica i una altra pràctica.

La part teòrica inclou tota una tasca de recerca de l'entorn J2EE centrat en la capa de presentació. L'estudi s'orienta en dos sentits:

- Cap als frameworks existents al mercat que afavoreixen el desenvolupament de la interfície amb l'usuari d'una aplicació empresarial en un entorn web.
- Cap al funcionament, estructura i organització d'un d'aquests frameworks per tal de poder-lo ampliar.

La part pràctica permet aplicar els coneixements adquirits en la vessant anterior, mitjançant el desenvolupament i la implementació concreta de dos elements:

- Un o més components que ampliïn la funcionalitat del marc de treball escollit.
- Una senzilla aplicació de mostra on s'utilitzi el framework estès i on es vegi la utilització dels components implementats.

La metodologia utilitzada per a portar a terme el projecte es basa en el sentit comú i així es pretén reflectir en l'apartat següent (la planificació). En aquest aspecte també es poden distingir les dues vessants comentades anteriorment: la teòrica i la pràctica.

En la part teòrica, el mètode a seguir consisteix a:

1. Cercar informació: llibres, internet...
2. Seleccionar aquelles fonts més interessants.
3. Treballar aquesta documentació: llegir-la, entendre-la...
4. Extraure les conclusions pertinents i sintetitzar el coneixement adquirit.

En la part pràctica, s'utilitza una metodologia basada en el Procés Unificat de Rational, de la qual es mostra un esquema-resum a continuació.

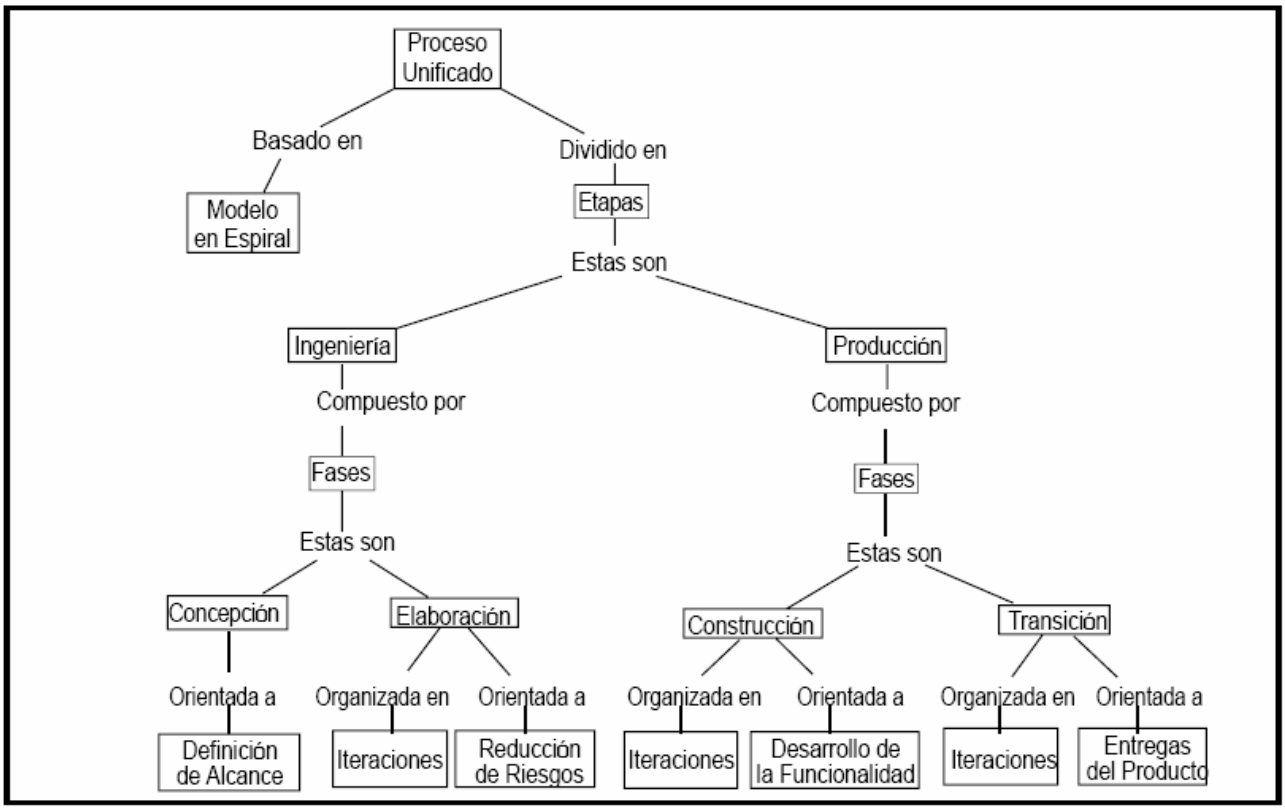


Figura 1.1:Mapa conceptual del Procés Unificat de Rational
(diagrama extret d'un llibre esmentat a la bibliografia)

Cal destacar que en la part pràctica s'ha donat més importància a les fases de desenvolupament del programari en sí, que no pas al producte obtingut com a resultat final. Això s'explica amb més detall al capítol de conclusions.

1.4. Planificació

A l'hora de planificar un projecte, cal tenir molt clar què s'ha de fer (definició de tasques) i els terminis que cal complir (definició de fites).

1.4.1. Tasques

En primer lloc es farà una relació de les tasques que es portaran a terme per a realitzar aquest projecte i a continuació es farà una breu descripció de cadascuna d'elles.

1. Familiarització J2EE i entorn de desenvolupament
   1.1. Instal·lació i configuració eines
      1.1.1. J2EE
      1.1.2. Eclipse
   1.2. Tutorials
      1.2.1. J2EE
      1.2.2. Eclipse
2. Elaboració pla de treball
3. Estudi frameworks web capa presentació
   3.1. Cerca frameworks web

   3.2. Estudi detallat de cadascun
   3.3. Elaboració estudi
4. Desenvolupament framework web propi
   4.1. Anàlisi
   4.2. Disseny
   4.3. Implementació
   4.4. Proves
5. Desenvolupament aplicació: “Enquestes de satisfacció de la docència”
   5.1. Anàlisi de l’aplicació
   5.2. Disseny
   5.3. Implementació
   5.4. Proves
   5.5. Implantació en entorn real
6. Documentació memòria PFC
7. Preparació presentació virtual

Tasca 1: Familiarització J2EE i entorn de desenvolupament

Preparació del lloc de treball i presa de contacte amb la plataforma J2EE i l’entorn de desenvolupament Eclipse. Instal·lació i configuració de les eines. Realització d’alguns tutorials per familiaritzar-se amb l’entorn i la plataforma que es faran servir al llarg del projecte.

Tasca 2: Elaboració del pla de treball

Planificació del projecte. Descripció a tall de presentació del què serà el projecte. Definició dels objectius a assolir. Enumeració de les tasques a realitzar i breu descripció de cadascuna d’elles.

Producte resultant: Pla de treball (el present document).

Lliurament: PAC1

Tasca 3: Estudi frameworks web capa presentació

Recopilació d’informació de diferents marcs de treball (frameworks) orientats a la capa de presentació per a entorns web. Estudi detallat de cadascun d’ells. Dins d’aquest estudi s’inclourà: lectura de la documentació corresponent, instal·lació del framework, cerca d’algun exemple que l’utilitzi, instal·lació i prova d’aquest exemple, avaluació del marc i extracció de conclusions. Valoració comparativa dels diferents frameworks estudiats.

Producte resultant: Estudi comparatiu dels frameworks.

Lliurament: PAC2

Tasca 4: Desenvolupament framework web propi

Selecció d’un dels frameworks estudiats i ampliació de les seves funcionalitats per tal de cobrir alguna de les mancances detectades. Aquesta tasca inclou les fases d’anàlisi, disseny, implementació i prova del software a desenvolupar.

Producte resultant: Framework propi.

Lliurament: PAC3

Tasca 5: Desenvolupament aplicació: “Enquestes de satisfacció de la docència”

Desenvolupament de l’aplicació “Enquestes de satisfacció de la docència” com a mostra de funcionament del marc de treball generat a la tasca anterior. Aquest punt inclou les fases d’anàlisi, disseny, implementació, prova i implantació en l’entorn real de l’aplicació creada.

Producte resultant: Aplicació de mostra.

Lliurament: PAC3

Tasca 6: Documentació memòria PFC

Documentació i síntesi del treball realitzat al llarg del desenvolupament d’aquest projecte final de carrera. Elaboració de la memòria del PFC.

Producte resultant: Memòria del PFC.

Lliurament: Lliurament final del PFC

Tasca 7: Preparació presentació virtual

Resum dels punts més importants del projecte presentats visualment en una presentació virtual.

Producte resultant: presentació virtual.

Lliurament: Lliurament final del PFC

1.4.2. Fites

Les dates límit per als lliuraments són les següents:

- 25/09/2006: Inici del PFC
- 10/10/2006: PAC1: pla de treball
- 30/10/2006: PAC2: estudis dels frameworks
- 18/12/2006: PAC3: lliurament del software desenvolupat
- 15/01/2007: Lliurament final: lliurament memòria i presentació virtual

1.4.3. Temporització

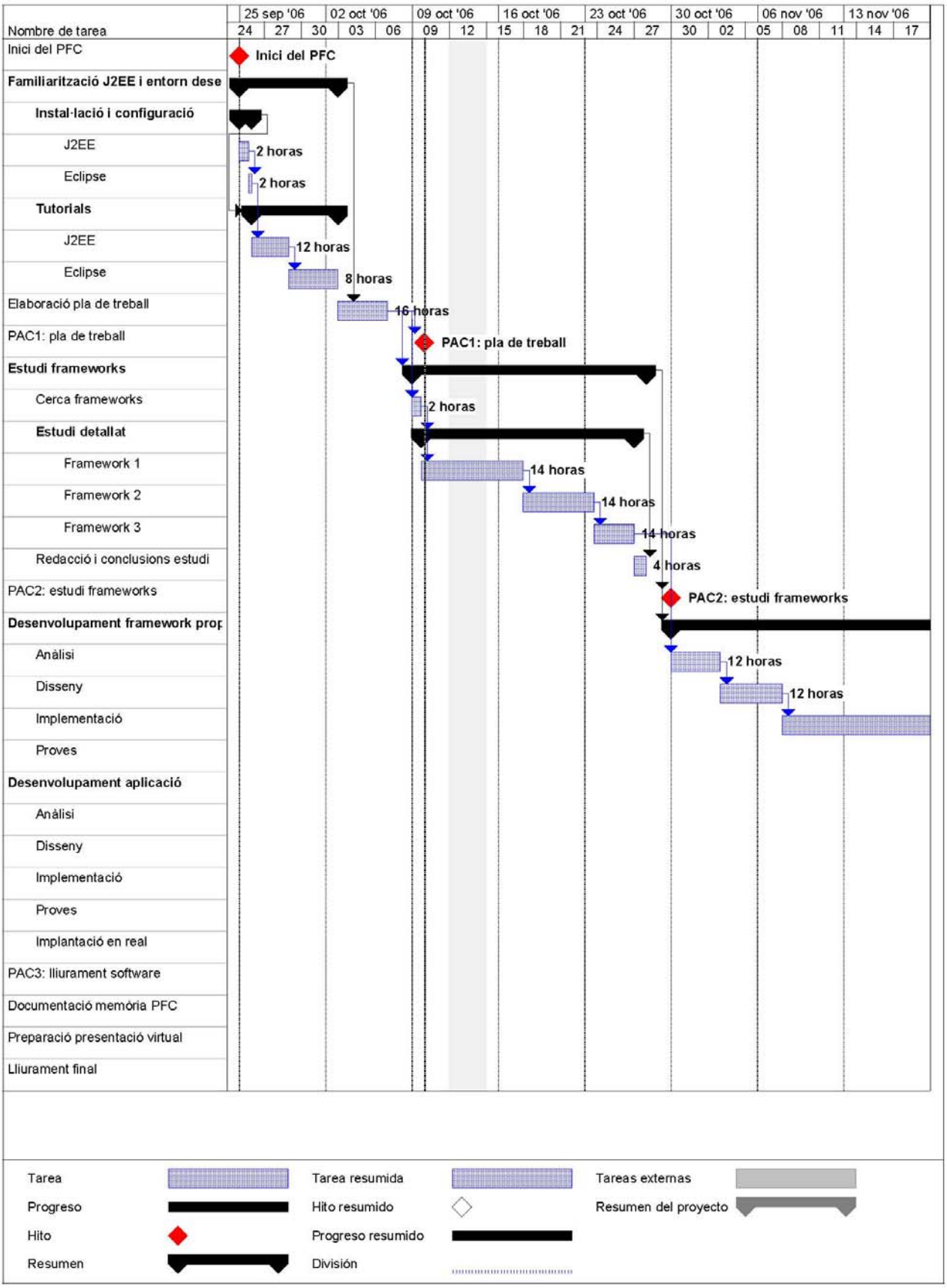
A continuació es mostra el diagrama de Gantt amb la previsió temporal per a les tasques i fites definides anteriorment.

Aquesta planificació està feta pensant en un horari de dedicació de 20 hores setmanals.

TASCA	DURADA	INICI	FI
Inici del PFC	0h	lu 25/09/06	lu 25/09/06
Familiarització J2EE i entorn desenvolupament	6d	lu 25/09/06	lu 02/10/06
Instal·lació i configuració	1d	lu 25/09/06	lu 25/09/06
J2EE	2h	lu 25/09/06	lu 25/09/06
Eclipse	2h	lu 25/09/06	lu 25/09/06
Tutorials	5d	ma 26/09/06	lu 02/10/06
J2EE	12h	ma 26/09/06	ju 28/09/06
Eclipse	8h	vi 29/09/06	lu 02/10/06
Elaboració pla de treball	16h	ma 03/10/06	vi 06/10/06
PAC1: pla de treball	0h	ma 10/10/06	ma 10/10/06
Estudi frameworks	12d	lu 09/10/06	vi 27/10/06
Cerca frameworks	2h	lu 09/10/06	lu 09/10/06
Estudi detallat	10,5d	lu 09/10/06	ju 26/10/06
Framework 1	14h	lu 09/10/06	ma 17/10/06
Framework 2	14h	mi 18/10/06	lu 23/10/06
Framework 3	14h	lu 23/10/06	ju 26/10/06
Redacció i conclusions estudi	4h	vi 27/10/06	vi 27/10/06
PAC2: estudi frameworks	0h	lu 30/10/06	lu 30/10/06
Desenvolupament framework propi	18d	lu 30/10/06	ju 23/11/06
Anàlisi	12h	lu 30/10/06	ju 02/11/06
Disseny	12h	vi 03/11/06	ma 07/11/06
Implementació	40h	mi 08/11/06	ma 21/11/06
Proves	8h	mi 22/11/06	ju 23/11/06
Desenvolupament aplicació	17d	vi 24/11/06	vi 15/12/06
Anàlisi	12h	vi 24/11/06	ma 28/11/06
Disseny	12h	mi 29/11/06	vi 01/12/06
Implementació	32h	vi 01/12/06	ma 12/12/06
Proves	8h	mi 13/12/06	ju 14/12/06
Implantació en real	4h	vi 15/12/06	vi 15/12/06
PAC3: lliurament software	0h	lu 18/12/06	lu 18/12/06
Documentació memòria PFC	20h	lu 18/12/06	vi 22/12/06
Preparació presentació virtual	8h	ma 02/01/07	mi 03/01/07
Lliurament memòria i presentació	0h	mi 03/01/07	mi 03/01/07

Taula 1.1:Temporització prevista

1.4.4. Diagrama de Gantt



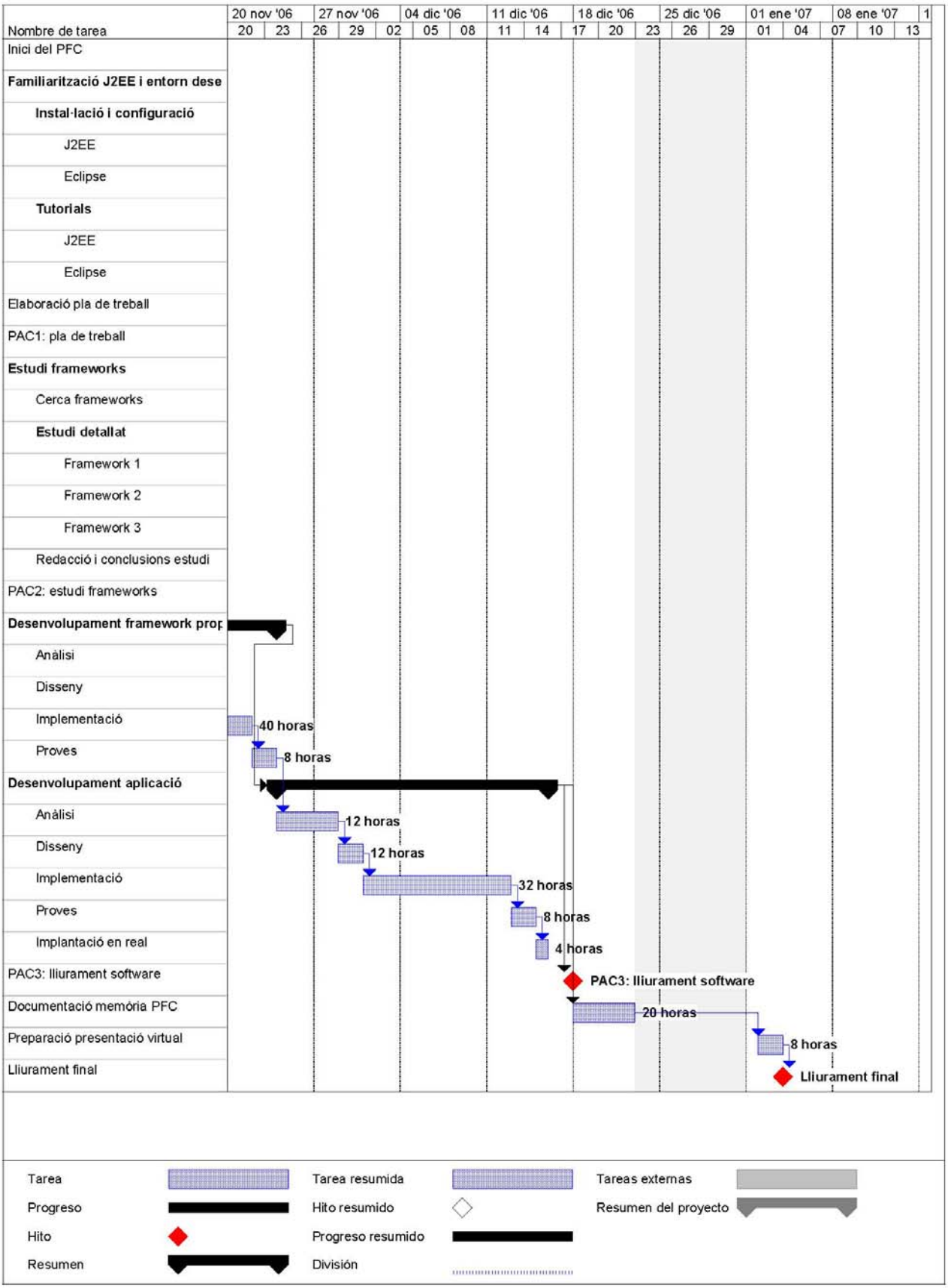


Figura 1.2:Diagrama de Gantt

1.5. Productes obtinguts

Els productes obtinguts en aquest projecte són els següents:

- Memòria tècnica (ovendrell_memoria.pdf)

  És el present document i inclou un resum de tot allò que es fa durant el desenvolupament del projecte. També recull informació important fruit de la tasca de recerca que s'espera d'un projecte d'aquest tipus.

- Productes de programari desenvolupats (ovendrell_producte.zip)
  - Components desenvolupats

    Es proporciona el codi font dels components desenvolupats.

    - *LlistaDoble.zip*: projecte del component LlistaDoble comprimit.

      Inclou un fitxer eclipse-build.xml a l'arrel que permet compilar i distribuir el projecte a l'aplicació que convingui (LlistaDoble.jar).

    - *Opcions.zip*: projecte del component Opcions comprimit.

      Inclou un fitxer eclipse-build.xml a l'arrel que permet compilar i distribuir el projecte a l'aplicació que convingui (Opcions.jar)

  - Aplicació de mostra

    S'adjunta el codi font de la petita aplicació de mostra. Es fa en dues parts: la primera gestiona la informació necessària per a poder fer les enquestes i la segona simula la generació d'aquestes enquestes.

    - *PFC_UOC.zip*: projecte de l'aplicació de gestió.

      Inclou un fitxer eclipse-build.xml a l'arrel que permet compilar i distribuir el projecte al servidor JBoss (enquestesAdmin.ear).

    - *Enquestes.zip*: projecte de l'aplicació de generació d'enquestes.

      Inclou un fitxer eclipse-build.xml a l'arrel que permet compilar i distribuir el projecte al servidor JBoss (enquestes.war).

    - *BD.zip*: scripts necessaris per a crear la BD en un servidor SQL server 2000.

      Inclou un fitxer amb la cadena de connexió que cal indicar a la configuració del servidor JBoss.

- Presentació virtual (ovendrell_presentacio.pps)

  És la síntesi de tot el projecte en format visual a mode d'exposició o presentació.

1.6. Organització de la memòria

Aquesta memòria pretén incloure tot allò que es fa durant el desenvolupament del projecte. És el que quedarà de la feina feta durant un semestre juntament amb els productes de software desenvolupats.

Està organitzada de la següent manera:

- Comença amb una introducció on es presenta el projecte: descripció, abast, objectius, planificació...
- A continuació dedica un capítol a cadascuna de les 3 fases en que es divideix el projecte:

  1. Estudi de frameworks de la capa de presentació

2. Ampliació d'un framework
3. Desenvolupament d'una aplicació de mostra

En cadascun d'ells es tracta amb detall allò que s'hi fa i s'hi inclou la documentació específica de cada part: dissenys, diagrames, esquemes, proves... És la part central i més tècnica de la memòria.

- Segueix amb un capítol sobre l'evolució del projecte i el seu seguiment (en relació amb la planificació inicial prevista). Inclou una valoració econòmica.
- Abans d'acabar, dedica un capítol a documentar les conclusions en funció dels objectius marcats a l'inici. Justifica la redefinició d'alguns objectius feta a mig projecte.
- Tanca amb diversos annexos que recullen documentació auxiliar com manuals o guies d'instal·lació i utilització del programari desenvolupat.

2. Estudi de frameworks de la capa de presentació

2.1. Introducció

El vertiginós creixement d'internet en els darrers anys és del tot inqüestionable. Amb ell s'han produït un seguit d'esdeveniments en cadena que afecten molt directament la manera com avui es planteja una web qualsevol.

Cada cop més, els llocs web s'interpreten no només com un aparador d'una organització, sinó com una eina que els clients o usuaris utilitzen per a portar a terme determinades gestions, processos... És a dir, s'han deixat enrera les webs totalment estàtiques i s'ha donat pas a aplicacions completes que aporten dinamisme i interacció amb l'usuari a molt alt nivell.

Això ha provocat una evolució espectacular en el desenvolupament d'aquests llocs webs, ara entesos com a aplicacions. La seva implementació pot arribar a ser molt costosa ja que les noves tecnologies que van apareixent són cada vegada més complexes, per poder cobrir les necessitats que se'ls demana.

Però les pressions del propi mercat en quant a temps, costos i esforços per a obtenir uns resultats de qualitat ràpidament han obligat als desenvolupadors de programari a reunir esforços i enginyar-se-les per tal de facilitar-se la feina.

En aquest punt és on apareixen els frameworks o marcs de treball. La seva principal raó de ser és agilitar el desenvolupament d'aplicacions per a obtenir resultats de qualitat en el menor temps possible, aprofitant els coneixements, esforços i experiències d'altres.

En l'entorn que ocupa aquest projecte, J2EE per a aplicacions web, hi ha una llista inacabable de projectes que es dediquen a construir frameworks que ajuden al seu desenvolupament. A partir de l'arquitectura per capes d'aquestes aplicacions, s'implementen marcs de treball més o menys genèrics. Alguns es centren en una sola capa (presentació, lògica de negoci o accés a dades) però d'altres són més ambiciosos i ofereixen solucions a tots els nivells.

Concretant una mica més, aquest projecte es vol centrar en els frameworks de la capa de presentació d'una aplicació web J2EE.

Que un framework estigui orientat a la capa de presentació no vol dir que no implementi cap funcionalitat de la capa de negoci o de la capa de dades (tot i que en molts casos sigui així), sinó que aporta notables beneficis a la primera.

Actualment, existeixen gran quantitat de marcs de treball d'aquest tipus. Alguns s'han estès notablement entre els desenvolupadors d'aquesta mena d'aplicacions convertint-se gairebé en estàndards i d'altres s'utilitzen en cercles molt puntuals. Però tots ells aporten un valor afegit a aquells qui els fan servir.

La implementació d'un nou framework és una tasca complexa i suposa un esforç important. Per això és habitual que molts projectes es dediquin a ampliar les opcions i funcionalitats que ofereix un entorn de treball ja desenvolupat per a obtenir un altre framework millorat i/o més potent. En aquest punt és on pren especial importància el codi lliure.

Tal i com es comentava a la descripció inicial d'aquest projecte, fer un estudi dels frameworks de la capa de presentació és el primer objectiu. Aquest estudi inclourà les següents activitats:

1. Buscar les diferents opcions existents actualment al mercat i fer-ne una relació.
2. Seleccionar alguns dels frameworks i fer-ne un estudi detallat a nivell teòric i pràctic.
3. Fer un estudi comparatiu dels frameworks avaluats.
4. Extraure conclusions.
5. Seleccionar un framework amb el que centrar la següent fase del projecte i justificar aquesta tria.

2.2. Relació de frameworks existents al mercat

A continuació es mostra una llista d'alguns marcs de treball que existeixen avui dia en el mercat per a entorns J2EE. Està dividida segons si el framewok és de codi lliure o propietari i està ordenada alfabèticament.

2.2.1. Codi lliure o OpenSource

- ADF - Oracle Application Development Framework (http://www.oracle.com/technology/products/jdev/htdocs/partners/addins/exchange/jsf/index.html) d'Oracle.
- Anvil (http://njet.org/) de Njet Communications Ltd.
- Aranea Web Framework (http://www.araneaframework.org/) d'Aranea Team, Webmedia Ltd.
- Aurora (http://www.auroramvc.org/aurora-web/index.htm) d'AuroraMVC.
- Barracuda (http://barracudamvc.org/Barracuda/) de BarracudaMVC.
- Brill (http://www.brillframework.org/fwwebapp/Home.html) de Brill Software Limited.
- Caramba (http://www.carambacomponents.org) de Systemworks.
- Chrysalis (http://chrysalis.sourceforge.net/) de Paul Strack.
- Click (http://click.sourceforge.net) de Malcom Edgar i altres.
- Cocoon (http://cocoon.apache.org/) d'Apache Software Fundation.
- DWR – Direct Web Remoting (http://www.getahead.ltd.uk/dwr/) de Getahead.
- Echo (http://www.nextapp.com/products/echo/) de NextApp.
- Expresso (http://www.jcorporate.com) de Jcorporate.
- Google Web Toolkit (http://code.google.com/webtoolkit/) de Google.
- Helma (http://helma.org/) d'Helma.
- Jaffa - Java Application Framework For All (http://jaffa.sourceforge.net/)
- Japple (http://www.japple.org/about.html) de Japple.org.
- JAT – Java Application Template (http://www.myjavaserver.com/~jatopensource/) de STF.
- javaserverfaces (http://java.sun.com/javaee/javaserverfaces/) de Sun.
- JOSSO – Java Open Single SignOn (http://www.josso.org) de Novascope S.A. i JOSSO Team.
- JPublish (http://www.jpublish.org/) d'Aetrion LLC.
- JSPWidget (http://edu.uuu.com.tw/jspwidget/default.jsp)
- Jucas (http://jucas.sourceforge.net/)
- Maverich (http://mav.sourceforge.net/)
- Millstone (http://www.millstone.org) d'IT Mill Ltd.
- MyFaces (http://www.myfaces.org/) d'Apache Software Fundation.
- OpenXava (http://www.openxava.org)
- Restlet (http://www.restlet.org) de Jerome Louvel.
- RIFE (http://rifers.org/) de RIFE Team.
- RSF (http://www2.caret.cam.ac.uk/rsfwiki/) de la Univertitat de Cambridge.
- SOFIA (http://www.salmonllc.com/website/Jsp/vanity/Sofia.jsp) de SalmonLLC.

- Spring MVC (http://www.springframework.org/)
- Strecks (http://strecks.sourceforge.net/) de Realsolve Solutions.
- Stripes (http://stripes.mc4j.org)
- Struts (http://jakarta.apache.org/struts/) d'Apache Software Fundation.
- Swinglets (http://www.javelinsoft.com/swinglets/) de Javelin Software.
- SwingWeb (http://swingweb.sourceforge.net)
- Tapestry (http://jakarta.apache.org/tapestry/) d'Apache Software Fundation.
- ThinWire (http://www.thinwire.com/) de Custom Credit Systems.
- Turbine (http://jakarta.apache.org/turbine/) d'Apache Software Fundation.
- Verge (http://www.inversoft.com) d'Inversoft LLC.
- VRaptor (http://www.vraptor.org/)
- WebOnSwing (http://webonswing.sourceforge.net/) de XOOPS.
- WebWork (http://www.opensymphony.com/webwork/) d'OpenSympony Group.
- Wicket (http://wicket.sourceforge.net/) de Wicket developers.
- wingS (http://www.j-wings.org)
- ZK (http://zk1.sourceforge.net) de Potix Corporation.

2.2.2. Codi propietari

- Backbase Ajax Java Edition (http://www.backbase.com/) de Backbase.
- WebGalieoFaces (http://www.jscape.com/webgalileofaces/index.html) de jscape

2.2.3. Selecció dels frameworks a estudiar

De tots ells, aquest projecte es dedicarà a estudiar-ne amb detall únicament 3. Degut a la gran quantitat de frameworks existents i al temps disponible per a l'estudi, no se'n poden abastar més, tot i que n'hi ha d'altres tan interessants com els seleccionats en aquest cas.

Els criteris utilitzats per a triar-los són:

- La seva extensió al mercat
- La documentació que hi ha al respecte
- La seva utilització a nivell laboral o professional actual i amb visió de futur.

A priori, els que s'ha cregut que compleixen millor aquestes expectatives són: MyFaces, Spring MVC i Struts.

De cada marc de treball es pretén fer un estudi a dos nivells:

- Teòric: característiques, avantatges i inconvenients...
- Pràctic: instal·lació, configuració (si cal), seguiment d'algun tutorial i prova d'algun exemple disponible.

2.2.4. Entorn de proves

A continuació es defineix l'entorn on es portarà a terme l'estudi:

- PC: *Pentium IV a 2,4 GHZ amb 1,5 Gb de RAM i 250 Gb de disc dur.*

- SO: *Windows XP professional (SP2)*
- Servidor web: *Apache-Tomcat 5.5.20*
- JVM: *J2SE_5_06*
- *J2EE 1.4 SDK*
- Entorn de desenvolupament: *Eclipse 3.2 (amb els plugins per a J2EE)*.

2.3. MyFaces

MyFaces és un framework OpenSource basat en Java per al desenvolupament d'aplicacions web. Es distribueix sota la llicència Apache 2.0.

És una implementació feta per l'Apache Software Fundation de l'especificació JSF (Java Server Faces). Val a dir que no és l'única. Sun, per exemple, té el seu propi framework sobre la mateixa especificació anomenat javaserverfaces.

S'estudiarà la distribució MyFaces Core 1.1.4 (implementada sobre l'especificació JSR 127).

2.3.1. Característiques

- És un framework que segueix el patró de disseny MVC (Model-Vista-Controlador), és a dir, separa clarament la gestió del workflow de l'aplicació (controlador), del model d'objectes de negoci (model) i de la generació de la interfície amb l'usuari (vista).
- La seva arquitectura, basada en l'especificació de JavaServer Faces, es mostra a continuació:

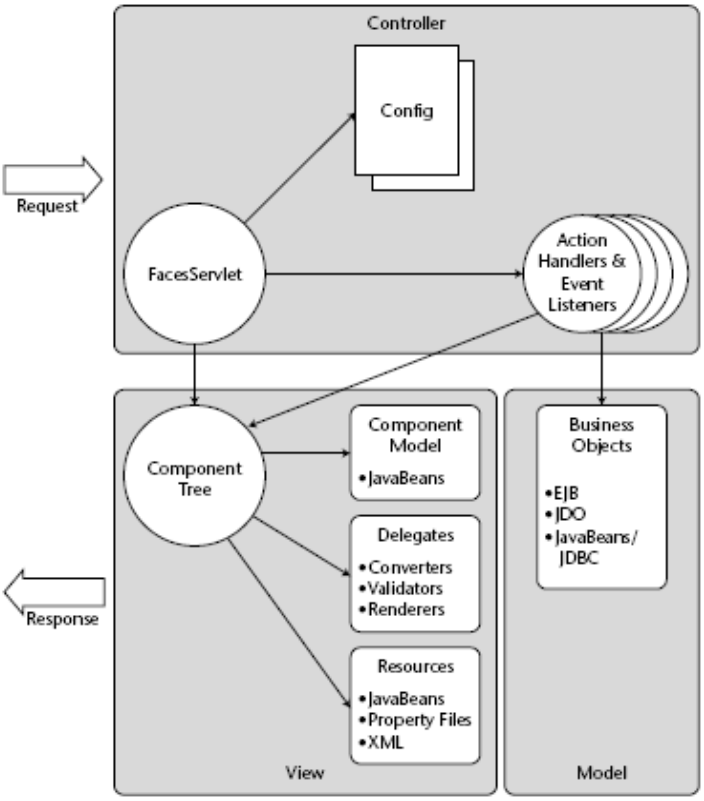


Figura 2.1:Model MVC que implementa MyFaces
(diagrama extret d'un llibre citat a la bibliografia)

- Separa totalment el disseny de la interfície d'usuari, del codi a executar per l'aplicació, alhora que facilita la seva connexió.
- Simplifica notablement el desenvolupament de les interfícies d'usuari.

- La programació de la interfície amb l'usuari es fa per mitjà de components (marc orientat al component) i es basa en events.
- La navegació segueix un model declaratiu, és a dir, es defineixen regles en el fitxer de configuració XML (faces-config.xml).
- Les regles de navegació són independents dels mètodes de la lògica de negoci.
- Utilitza JSP (Java Server Pages) com a tecnologia de visualització habitual, tot i que en pot fer servir d'altres com Velocity, Tiles...
- Els mètodes de la lògica de negoci i accés a dades estan directament implementats amb POJO (Plain Old Java Objects), classes típiques de Java.
- Inclou un conjunt d'APIs (Application Programming Interface) per representar components d'interfície d'usuari i administrar el seu estat, gestionar events, validar dades introduïdes, definir un esquema de navegació entre pàgines...
- Inclou un parell de llibreries de marques (tags) que es poden personalitzar. La primera (HTML Render Kit Library) defineix totes les etiquetes dels components de la interfície d'usuari estàndard. La segona (Core Tag Library) defineix les marques que representen la conversió de tipus, la validació de dades, els gestors d'events...
- La seva arquitectura permet mostrar els components de presentació de diferents maneres per poder-se adaptar a gran varietat de clients sense un cost addicional.
- Permet a l'usuari crear els seus propis components i ampliar el framework.
- Incorpora un sistema per a validar dades d'entrada per part de l'usuari.
- Dóna suport per a desenvolupar aplicacions multi-idiomes (i18n).
- S'adapta a nombrosos entorns de desenvolupament: eclipse, Oracle JDeveloper, Borland JBuilder, NetBeans, Exadel Studio...

2.3.2. Instal·lació

La seva instal·lació consisteix en 4 passos bàsics:

1. Baixar-se el paquet http://www.apache.org/dyn/closer.cgi/myfaces/binaries/myfaces-core-1.1.4-bin.zipde la web oficial (http://myfaces.apache.org/download.html)
2. Descomprimir-lo.
3. Copiar totes les llibreries *myfaces-core-1.1.4\lib*.jar* a la carpeta *WEB-INF\lib* de l'aplicació a desenvolupar.
4. Configurar el fitxer *WEB-INF\web.xml* de l'aplicació a desenvolupar.

2.3.3. Avantatges

- Es basa en una especificació, concretament de JSF, la qual és molt probable que esdevingui un estàndard oficial a curt termini. Per aquest fet, té un gran suport de la indústria i dels fabricants.
- És força senzill de comprendre i fàcil d'utilitzar (en relació a altres frameworks).
- És molt flexible en tots els seus aspectes.
- Desacobla completament la fase de presentació del controlador, fet que permet al desenvolupador generar les seves pròpies eines de presentació per construir components totalment personalitzats.
- Proporciona independència del dispositiu client.
- Pot integrar diverses tecnologies de presentació: JSPs, Velocity Templates, Tiles...

- És fàcilment extensible. El desenvolupador pot crear els seus propis components d'interfície d'usuari. Actualment hi ha moltes ampliacions disponibles per internet.
- Suporta la internacionalització de les aplicacions.
- Instal·lació senzilla i ràpida.
- Nombrosos IDEs el suporten: Eclipse, Sun Java Studio, Oracle JDeveloper, Borland JBuilder, NetBeans, Exadel Studio...
- Aplicacions independents de la plataforma i que funcionen sobre qualsevol servidor web J2EE: Tomcat, BEA WebLogic, JBoss...
- Té molt bones perspectives de futur (hi ha nombrosos projectes que hi estan dedicant grans esforços) degut a que el seu disseny s'ha fet a consciència i amb l'experiència d'altres frameworks més antics (com Struts).

2.3.4. Inconvenients

- No té validació de dades a la banda del client, només validadors a la part del servidor.
- És un marc poc madur encara.
- La documentació específica sobre MyFaces que hi ha és limitada i bàsicament en anglès. Sobre JSF ja és una altra cosa.

2.3.5. Referències

Llibres

- DUDNEY, Bill... *Mastering JavaServer Faces*. John Wiley & Sons, 2004.

  ISBN: 0471462071
- GEARY, David... *Core JavaServer Faces*. Sun Microsystem Press Java Series. Prentice Hall, 2004.

  ISBN: 0131463055
- MANN, Kito. *JavaServer Faces in action*. Manning Publications, 2005.

  ISBN: 1932394125

Webs

- Web oficial de MyFaces.
  http://www.myfaces.org
- Tutorial de JSF i MyFaces.
  http://www.coreservlets.com/JSF-Tutorial/
- Recursos diversos de JSF i MyFaces.
  http://www.jsftutorials.net/
- Classes d'API de MyFaces
  http://myfaces.apache.org/api/apidocs/index.html
- Classes generals de MyFaces
  http://myfaces.apache.org/commons/apidocs/index.html
- Tutorials diversos en espanyol.
  http://www.adictosaltrabajo.com/indexg.php?noimages=SI&pagina=listacat&id=32
- Tutorials diversos.
  http://www.lalilluna.de/javaserver-faces-tutorials.html

2.4. Spring MVC

Spring MVC és un framework de codi lliure per a crear aplicacions web basades en Java que es distribueix sota la llicència Apache 2.0.

Forma part d'Spring, un framework molt complet que inclou components per a les tres capes d'aplicació: presentació, lògica de negoci i accés a dades. Per això aquest projecte es centrarà en la plataforma Spring MVC, que és la part que correspon precisament al paradigma MVC – Model-Vista-Controlador - (tal i com el seu nom indica) i que està orientada a la capa de presentació.

S'estudiarà l'última versió estable i ja disponible per entorns de producció, recentment publicada: Spring 2.0.

2.4.1. Característiques

- Com el seu nom indica, es basa en el patró Model-Vista-Controlador. La separació entre les tres capes és molt clara i ben diferenciada.
- La seva arquitectura es mostra a continuació:

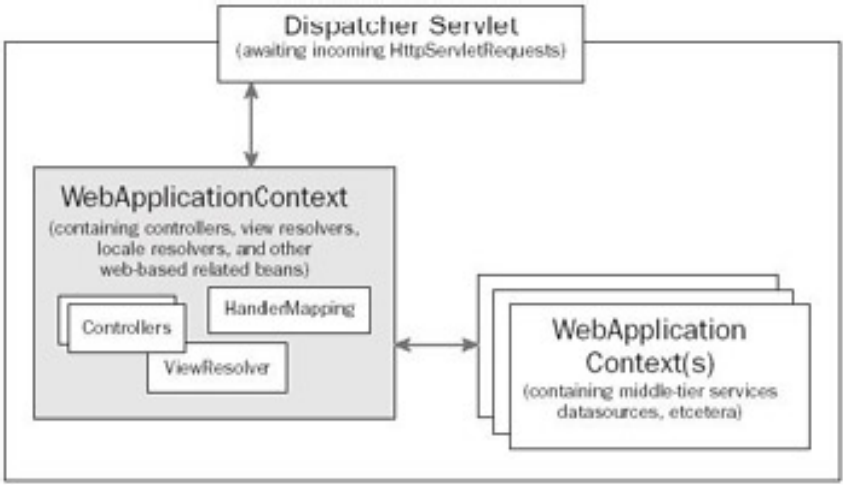


Figura 2.2:Model MVC que implementa Spring MVC
(diagrama extret d'un llibre esmentat a la bibliografia)

- Es basa en peticions-respostes.
- Complementa la programació orientada a objectes (OOP) amb la programació orientada a l'aspecte (AOP).
- Es basa totalment en interfícies.
- Habitualment utilitza JSP com a tecnologia de presentació, tot i que en pot fer servir moltes altres com Velocity, XLST o fins i tot un mecanisme propi personalitzat.
- La interfície bàsica que implementa el controlador és molt abstracta. Per això Spring ofereix nombroses implementacions que resolen situacions específiques.
- A més dels controladors proporciona interceptors, fet que facilita la planificació del comportament davant de nombroses peticions de clients.
- Permet la definició i utilització de temes per al disseny de l'aspecte de l'aplicació.
- Dóna suport per a desenvolupar aplicacions multi-idiomes (i18n).
- S'adapta a nombrosos entorns de desenvolupament: eclipse, NetBeans, Exadel Studio...

2.4.2. Instal·lació

Per instal·lar-lo cal seguir les següents passes:

1. Baixar-se el paquet de la web oficial (http://www.springframework.org/download).
2. Descomprimir-lo.
3. Copiar totes les llibreries *spring-framework-2.0\dist\spring*.jar* a la carpeta *WEB-INF\lib* de l'aplicació a desenvolupar.
4. Configurar el fitxer *WEB-INF\web.xml* de l'aplicació a desenvolupar.

2.4.3. Avantatges

- Està integrat dins del framework Spring, el qual abasta totes les capes d'una aplicació web: presentació, lògica de negoci i accés a dades.
- És fàcilment ampliable, fent extensions de les interfícies.
- És molt flexible gràcies a que es basa tot en interfícies.
- Pot integrar diverses tecnologies de presentació: JSPs, Velocity Templates, Tiles...
- Es pot utilitzar com a plataforma de molt diversos tipus de projectes.
- Incorpora un sistema personalitzable per a la comprovació de les dades introduïdes per l'usuari en formularis i la gestió d'errors corresponents.
- Proporciona independència del dispositiu client.
- Suporta la internacionalització de les aplicacions.
- S'estan dedicant molts esforços a millorar aquest framework i adaptar-lo a les noves necessitats que van sorgint. Té molt bones expectatives de futur.
- Existeixen força eines i plugins per a IDEs que faciliten el desenvolupament i configuració d'una aplicació implementada amb Spring.
- Aplicacions independents de la plataforma i que funcionen sobre qualsevol servidor web J2EE: Tomcat, BEA WebLogic, JBoss...
- La comunitat Spring és extensa i molt activa.

2.4.4. Inconvenients

- La major part de la documentació existent és de la versió anterior i està fonamentalment en anglès. De l'actual encara no s'ha escrit gaire i no hi ha massa exemples.
- No implementa un model de resposta a events.
- No té nocions de components a nivell de servidor, per la qual cosa no pot salvar ni recuperar l'estat d'un component.
- Es basa en peticions-respostes.
- La navegació entre pàgines es determina mitjançant codi encapsulat dins dels controladors.

2.4.5. Referències

Llibres

- JOHNSON, Rod. *Professional Java Development with the Spring Framework*. Wrox Press, 2005.

  ISBN: 0764574833

- WALLS, Craig... *Spring in action*. Manning Publications, 2005.

  ISBN: 1932394354

Webs

- Web oficial d'Spring MVC
  http://www.springframework.org/docs/reference/mvc.html
- Tutorial d'Spring MVC
  http://www.springframework.org/docs/MVC-step-by-step/Spring-MVC-step-by-step.html
- Web oficial d'Spring
  http://www.springframework.org/
- Recursos diversos en espanyol sobre Spring
  http://www.springhispano.org
- Introducció a Spring
  http://www.theserverside.com/tt/articles/article.tss?l=SpringFramework
- Classes de referència d'Spring
  http://static.springframework.org/spring/docs/2.0-m3/api/index.html?overview-summary.html

2.5. Struts

Struts és un framework de codi lliure per a crear aplicacions web basades en Java. S'emmarca sota la llicència Apache 2.0.

S'avaluarà Struts 1.3.5, versió estable més recent en el moment de portar a terme l'estudi. Struts 2 tot just acaba de ser publicat en una versió beta.

2.5.1. Característiques

- Es basa en el patró MVC (Model-Vista-Controlador).
- La seva arquitectura es mostra a continuació:

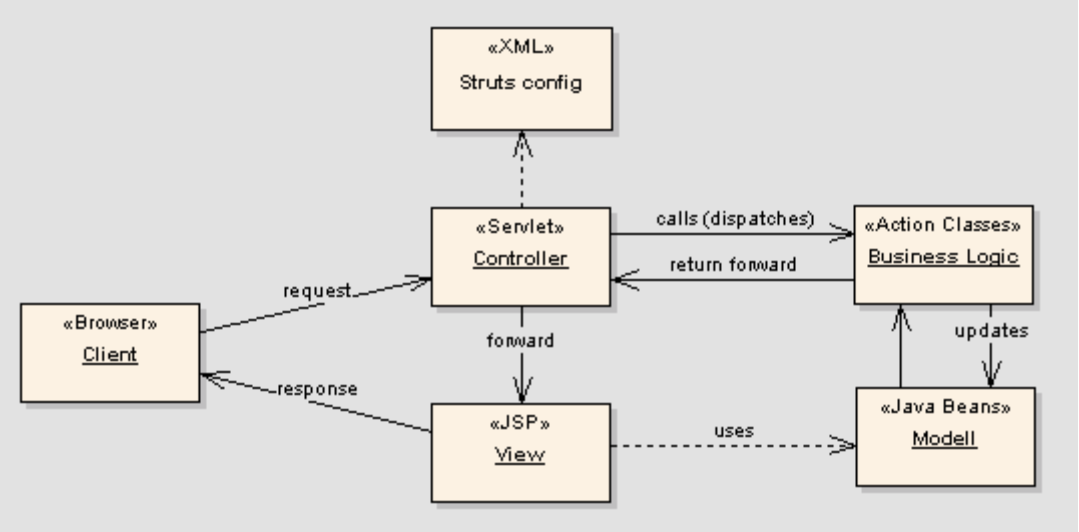


Figura 2.3:Model MVC que implementa Struts
(diagrama extret d'una web citada a la bibliografia)

- Es basa en peticions-respostes.
- La vista s'implementa fonamentalment amb JSP i a base d'etiquetes o tags predefinides.

- El controlador s'implementa amb un servlet i un fitxer de configuració XML. És l'encarregat de rebre les peticions de l'usuari, executar les accions del model que corresponguin i llençar el JSP necessari perquè generi la resposta final.
- El model s'implementa amb Beans i Actions, els quals s'encarreguen de recuperar la informació d'entrada (formularis, sessions...), fer crides als objectes de negoci necessaris, analitzar els resultats i generar la sortida corresponent.
- La navegació és declarativa i les regles es defineixen en un fitxer de configuració XML (struts-config.xml).
- Dóna suport per a desenvolupar aplicacions multi-idiomes (i18n).
- S'adapta a nombrosos entorns de desenvolupament: eclipse, IDEA, Exadel Studio, emacs... Molts d'ells disposen de plug-ins que faciliten la seva utilització.

2.5.2. Instal·lació

La seva instal·lació consisteix a :

1. Baixar-se el paquet de la web (http://struts.apache.org/downloads.html).
2. Descomprimir-lo.
3. Copiar les llibreries necessàries *struts-1.3.5/lib/*.jar* a la carpeta *WEB-INF\lib* de l'aplicació a desenvolupar.
4. Configurar el fitxer *WEB-INF\web.xml* de l'aplicació a desenvolupar.

2.5.3. Avantatges

- És un dels frameworks més estesos actualment. Hi ha innombrables aplicacions desenvolupades que l'utilitzen i això implica que hi hagi gran quantitat d'informació, de documentació i d'exemples al respecte.
- És una tecnologia molt madura, que ha esdevingut estàndard de facto, tot i que avui ha quedat una mica endarrerida (cal esperar les novetats que aporta la nova versió).
- Incorpora un sistema molt potent per a la comprovació de les dades introduïdes per l'usuari en formularis i la gestió d'errors corresponents.
- Ofereix moltes possibilitats d'extensió i adaptació a les necessitats de cada desenvolupador. Actualment hi ha incomptables ampliacions disponibles a internet que incorporen i/o milloren les funcionalitats que struts ja disposa.
- Suporta la internacionalització de les aplicacions.
- Hi ha moltíssima documentació, tant en anglès com en espanyol. És molt variada: llibres, tutorials on-line, fòrums, exemples...
- Instal·lació senzilla.
- Existeixen força eines i plugins per a IDEs que faciliten el desenvolupament i configuració d'una aplicació implementada amb struts.
- Aplicacions independents de la plataforma i que funcionen sobre qualsevol servidor web J2EE: Tomcat, BEA WebLogic, JBoss...

2.5.4. Inconvenients

- Requereix d'un temps previ d'aprenentatge a tenir en compte, potser superior a altres tecnologies. No només cal aprendre les possibilitats que ofereix, sinó que cal entendre la seva filosofia.

- És poc apropiat per a utilitzar en paradigmes que no siguin el MVC.
- No implementa un model de resposta a events.
- El codi (scripts) només es pot escriure dins de classes Action de Java.
- No té nocions de components a nivell de servidor, per la qual cosa no pot salvar ni recuperar l'estat d'un component.
- La separació entre vista i model no és total.
- Es basa en peticions-respostes.
- Tot i que s'estan dedicant molts esforços a millorar aquest framework i adaptar-lo a les noves necessitats que van sorgint, no té bones expectatives de futur.

2.5.5. Referències

Llibres

- GOODWILL, James... *Professional Jakarta Struts*. Wrox Press, 2003.

  ISBN: 0764544373

Webs

- Web oficial d'Apache Struts.
  http://struts.apache.org/index.html
- Web d'un projecte per implementar aplicacions utilitzant Struts: exemples, utilitats...
  http://struts.sourceforge.net/
- Directori de recursos diversos sobre Struts
  http://www.strutscentral.net
- Tutorial bàsic d'Struts en espanyol
  http://www.programacion.com/java/tutorial/joa_struts/
- Tutorials diversos en espanyol
  http://www.adictosaltrabajo.com/indexg.php?noimages=SI&pagina=listacat&id=15
- Introducció a Struts
  http://www.coreservlets.com/Apache-Struts-Tutorial/Understanding-Struts.html
- Article d'introducció a Struts
  http://www-128.ibm.com/developerworks/ibm/library/j-struts/
- Implementacions que amplien les funcionalitats d'Struts
  http://wiki.apache.org/struts/StrutsExtensions
- Classes de referència d'Struts
  http://struts.apache.org/1.3.5/struts-core/apidocs/index.html

2.6. Estudi comparatiu

Aquest estudi comparatiu consta de dues parts.

En primer lloc es mostra una taula resumint les característiques més importants de cada framework.

2.6.1. Taula resum

	MyFaces	Spring MVC	Struts
Versió estudiada	MyFaces Core 1.1.4	Spring 2.0	Struts 1.3.5
Procés d'instal·lació	Baixar-se el paquet i copiar llibreries (*.jar) a la carpeta */WEB-INF/lib* de l'aplicació.	Baixar-se el paquet i copiar llibreries (*.jar) a la carpeta */WEB-INF/lib* de l'aplicació.	Baixar-se el paquet i copiar llibreries (*.jar) a la carpeta */WEB-INF/lib* de l'aplicació.
Adaptació a IDEs	Eclipse, Sun Java Studio, Oracle JDeveolper, Borland JBuilder...	Eclipse, Oracle JDeveloper, IDEA, Borland JBuilder...	Eclipse, IDEA, emacs...
Paradigma	Patró MVC (Model-Vista-Controlador).	Patró MVC (Model-Vista-Controlador).	Patró MVC (Model-Vista-Controlador).
Orientació de la programació	Programació orientada a components i basada en events.	Programació basada en peticions i respostes. Complementa la programació orientada a objectes amb la programació orientada a l'aspecte.	Programació basada en peticions i respostes.
Implementació vista	Tecnologies diverses: JSPs amb tags (habitualment), Velocity, Tiles...	Tecnologies diverses: JSPs amb tags (habitualment), Velocity, Tiles...	JSPs amb tags.
Implementació controladors	Components implementats amb POJO (classes típiques de Java).	Interfície de serveis.	Servlets.
Implementació model (scripts)	Components Bean implementats amb POJO (classes típiques de Java).	Components Bean implementats en llenguatges diversos: Java, JRuby...	JavaBeans amb Action.
Navegació	Descriptiva, configurada en un fitxer XML (faces-config.xml).	Mitjançant codi encapsulat en els controladors.	Descriptiva, configurada en un fitxer XML (struts-config.xml).
Validació de dades d'entrada	Es basa en afegir i/o modificar les etiquetes corresponents en els fitxers JSP. Té alguns validadors predefinits.	Es basa en un fitxer XML de validació (validation.xml) i en una llibreria d'etiquetes de formulari.	Es basa en un fitxer XML de validació (validation.xml) i en una llibreria d'etiquetes de formulari. Hi ha un framework addicional de validació.

Taula 2.1:Taula resum estudi frameworks

En segon lloc es descriuen i valoren els criteris o indicadors utilitzats per a fer l'estudi comparatiu entre els 3 frameworks:

- **Separació MVC:** nivell d'independència entre model, vista i controlador que implementa el framework.
  Valors possibles: baixa, mitja, alta.
- **Diversitat tecnològica:** possibilitat d'utilitzar diverses tecnologies a la vista a part de les habituals JSPs.
  Valors possibles: sí, no.
- **Independència del client:** capacitat del framework per generar una capa vista independent del dispositiu del client.
  Valors possibles: sí, no.
- **Suport multi-idioma:** facilitat del framework per a desenvolupar aplicacions en diferents llengües. S'indica si ofereix alguna eina o recurs per a implementar aquesta característica.
  Valors possibles: sí, no.
- **Validació de dades:** capacitat del framework a convertir i validar dades d'entrada tant a la banda del servidor com a la del client.
  Valors possibles: només servidor, només client, servidor i client.
- **Model d'events:** implementació i gestió d'un model de resposta a events en la interfície amb l'usuari per part del framework.
  Valors possibles: sí, no.
- **Extensió i creació de components:** possibilitat d'estendre un conjunt estàndard de components i crear-ne de totalment nous.
  Valors possibles: sí, no.
- **Gestió de sessions:** possibilitat per part del framework de gestionar l'estat de la sessió de l'aplicació a la part del servidor.
  Valors possibles: automàtica, manual.
- **Documentació disponible:** quantitat, varietat de format i qualitat de la documentació disponible (llibres, tutorials, articles, fòrums...).
  Valors possibles: poca, mitja, molta.
- **Instal·lació:** dificultat del procés i temps per a portar-la a terme.
  Valors possibles: senzilla, mitja, complexa.
- **IDE:** disponibilitat d'entorns de desenvolupament integrats que faciliten l'elaboració d'aplicacions basades en el framework.
  Valors possibles: cap, un, més d'un.
- **Demanda laboral:** quantitat d'ofertes laborals que es troben a diverses borses de treball per internet (Infojobs, Laboris, Monster...) a nivell de tot l'estat espanyol. Es calcula un percentatge en funció de diverses consultes realitzades al llarg d'aquest estudi. Cal tenir en compte que la cerca s'ha fet amb JFS i MyFaces, Spring (en lloc d'Spring MVC) i Struts. Per tant els resultats no són tan significatius com es desitjaria.
  Valors possibles: quantitat en mitjana i percentatge.
- **Visió de futur:** tendència de la utilització dels frameworks en aplicacions de nova construcció. Aquest criteri és una mica subjectiu ja que sorgeix de la lectura de diversos articles i fòrums i de la consulta a professionals del ram, però no hi ha cap indicador que permeti valorar-ho de forma objectiva.

2.6.2. Taula comparativa

Indicador	MyFaces	Spring MVC	Struts
Separació MVC	Alta	Alta	Mitja
Diversitat tecnològica	Sí	Sí	No
Independència del client	Sí	Sí	No
Suport multi-idioma	Sí	Sí	Sí
Validació de dades	Només servidor	Servidor i client	Servidor i client
Model d'events	Sí	No	No
Extensió i creació de components	Sí	No	No
Gestió de sessions	Automàtica	Manual	Manual
Documentació disponible	Mitja (bàsicament en anglès)	Poca (bàsicament en anglès)	Molta (en anglès i espanyol)
Instal·lació	Senzilla	Senzilla	Senzilla
IDE	Més d'un	Més d'un	Més d'un
Demanda laboral	33 ofertes de mitjana suposen el 7%	Spring: 80 ofertes de mitjana suposen el 17%	364 ofertes de mitjana suposen el 76%
Visió de futur	Moltes aplicacions de nova construcció es basen en JSF (especificació implementada per MyFaces).	Té futur en aquelles aplicacions que volen un sol framework per a totes les capes.	Les aplicacions noves no es comencen amb Struts.

Taula 2.2:Taula comparativa frameworks

2.7. Conclusions

Per finalitzar aquest estudi falta exposar les conclusions que se'n poden extreure:

- A nivell d'entorns i aplicacions web.
- A nivell de frameworks web en general.
- A nivell dels frameworks estudiats.
- I a nivell personal.

Actualment hi ha diverses evidències pel que fa al desenvolupament d'aplicacions web en entorns empresarials:

- Els entorns web ja no són un contenidor d'informació estàtica, sinó que han passat a ser aplicacions dinàmiques amb moltes possibilitats.
- A la majoria dels casos, el desenvolupament de les aplicacions es basa en frameworks. Això permet reduir el temps d'implementació, estructurar l'aplicació en mòduls molt ben definits,

aconseguir major flexibilitat davant de canvis i ampliacions, aprofitar funcionalitats ja implementades, centrar la feina del desenvolupador en la lògica de negoci (ocultant-li certa complexitat tecnològica)...

Pel que fa als frameworks, també es poden destacar diversos punts:

- S'utilitzen principalment frameworks de codi lliure o OpenSource. Els motius són diversos: es redueixen costos, s'afavoreix la portabilitat de les aplicacions, es concentren esforços i s'orienten cap a una causa comuna, es facilita la reutilització de codi...
- Destaquen els marcs desenvolupats sobre J2EE.
- Fer un framework des de zero és molt costós, per això és habitual agafar-ne un d'existent, el que millor s'adapti a les necessitats del moment, i afegir-hi aquelles funcionalitats que no té.
- Avui en dia n'hi ha molts disponibles, però només són uns pocs els utilitzats per la majoria.

De l'estudi detallat dels 3 frameworks seleccionats (MyFaces, Spring MVC i Struts) es pot deduir que tenen força semblances a nivell general, però que individualment cadascun té els seus aspectes diferencials.

Alguns punts en comú són:

- Estan orientats a la capa de presentació (tot i que Spring MVC s'integri dins d'Spring, un framework que abasta també la capa de lògica de negoci i la d'accés a dades).
- Permeten desenvolupar aplicacions web per a entorns empresarials sota el paradigma MVC (Model-Vista-Controlador), fet que afavoreix la separació entre les diferents capes (en alguns casos millor aconseguit que en uns altres).
- Per a utilitzar-los n'hi ha prou amb baixar-se el paquet corresponent d'internet i copiar les llibreries necessàries al directori */WEB-INF/lib* de l'aplicació.
- Habitualment, es basen en JSPs com a tecnologia de la vista.
- Ofereixen suport per a desenvolupar aplicacions en diferents idiomes de manera simple. Es configura mitjançant un fitxer XML i es defineix un fitxer de recursos per a cada llengua a implementar.
- Disposen d'elements per a convertir i validar les dades d'entrada a l'aplicació per la interfície d'usuari.
- Les aplicacions resultants són fàcilment portables i independents de la plataforma. L'únic requisit és que el servidor web incorpori la tecnologia J2EE.
- Estan àmpliament suportats pels entorns de desenvolupament integrats. Hi ha un gran nombre d'IDEs que incorporen ampliacions o plug-ins per a facilitar la utilització dels frameworks.

Les principals diferències radiquen en:

- La implementació. Mentre que MyFaces es basa en components UI (d'interfície d'usuari), Spring MVC es basa en interfícies i Struts en peticions-respostes.
- La gestió de la sessió. Només MyFaces gestiona automàticament l'estat de la sessió de l'aplicació. Spring MVC i Struts no incorporen cap element específic per a fer-ho.
- La independència del dispositiu client. MyFaces i Spring MVC són capaços de mostrar els elements de presentació de diferent manera per adaptar-se a la varietat de clients existent. Struts no ho pot fer.
- L'extensibilitat. MyFaces pot estendre fàcilment les seves funcionalitats per mitjà de la personalització i creació de nous components UI. Spring MVC i Struts no implementen components.

- La flexibilitat tecnològica. Els 3 frameworks implementen la vista de manera habitual amb JSPs, però MyFaces i Spring MVC permeten, a més a més, utilitzar altres tecnologies com Velocity, Tiles...
- La maduresa de la tecnologia i l'experiència que hi ha al darrera. Una conseqüència directa d'aquesta maduresa és la quantitat de documentació que s'ha escrit sobre el tema. Així d'Struts es pot trobar material i informació de tot tipus i en idiomes diversos. De MyFaces hi ha menys varietat, tot i que es troba moltíssima cosa de JavaServer Faces, l'especificació en la que està basada. Per últim, de la nova versió d'Spring MVC gairebé no hi ha res donat que tot just acaba de sortir.
- El tipus d'aplicació on s'utilitza. Spring MVC per sí sol no té massa sentit (s'acostuma a fer servir integrat en Spring). Struts va ser estàndard de facto fa un temps i per això hi ha moltes aplicacions que es van començar en el seu dia i que l'utilitzen, però les noves prefereixen escollir una tecnologia més recent i una arquitectura més pensada. Aquest és el cas de MyFaces (amb JSF al darrera), cada dia més acceptat.
- La visió de futur. Alguns factors que influeixen en la visió de futur són el tipus d'aplicació on s'utilitza, el disseny del framework, la tecnologia en què es basa, l'acceptació en la comunitat de desenvolupadors d'aplicacions web, el suport de grans empreses, els interessos del mercat... I tenint en compte tots aquests elements, el framework que té millors expectatives és MyFaces com a tal (o JSF més en general).

A nivell personal també s'han tret algunes conclusions:

- El fet de desenvolupar una aplicació web amb frameworks és una metodologia molt potent, però també molt complexa.
- Cal un període previ d'aprenentatge força important. No només s'ha d'aprendre un nou llenguatge, un nou entorn o una nova API, sinó que cal canviar de mentalitat i entendre una nova filosofia de treball.
- La documentació que hi ha sobre el tema és tan extensa que dificulta agafar una primera impressió general.
- La tria d'un framework a l'hora de desenvolupar una aplicació és complicada. Cal documentar-se bé per saber quines opcions hi ha, conèixer les possibilitats que ofereix cadascuna d'elles, provar-les.. i això requereix molta dedicació. En aquest sentit, l'experiència és un factor molt important.

2.8. Framework a ampliar: MyFaces

Un cop finalitzat l'estudi s'està en disposició de seleccionar un framework per a ampliar en la següent fase d'aquest projecte, i aquest és **MyFaces**.

Els criteris als que s'ha donat més importància a l'hora de prendre la decisió són:

- **Actualitat tecnològica.** S'ha preferit una tecnologia més recent i puntera que una més estable però ja una mica obsoleta. MyFaces es basa en una especificació que ha esdevingut estàndard oficial i que segueix evolucionant segons les necessitats del moment. És una tecnologia molt viva.
- **Possibilitat d'extensió del framework.** L'objectiu de la següent fase del projecte és dotar al framework d'alguna funcionalitat que no implementa. Per això interessa que es pugui estendre fàcilment. En aquest sentit, MyFaces té una part bona i una altra no tant. La primera és que es pot estendre còmodament implementant nous components UI. La segona és que al basar-se en una especificació com JSF és molt complet i costarà trobar quina funcionalitat afegir.
- **Flexibilitat.** La flexibilitat en un entorn és un aspecte desitjable sempre i quan no impliqui una excessiva complicació per a aconseguir-la. MyFaces és independent del dispositiu del client.

- **Documentació disponible.** MyFaces no és dels frameworks millor documentats, però sí que ho és l'especificació JSF. Per tant, no hi ha d'haver cap problema a aconseguir la informació necessària per a poder-se familiaritzar amb el framework.
- **Visió de futur.** Aquest indicador té una part subjectiva a tenir en compte. Es basa en l'opinió de diversos professionals i experts en el tema que han publicat els seus articles exposant el seu punt de vista sobre la tendència d'aquestes plataformes. En aquest sentit, sembla que hi ha una clara preferència per JSF, cosa que indirectament afavoreix a MyFaces.

3. Ampliació d'un framework: MyFaces

El segon gran objectiu d'aquest projecte és el desenvolupament d'un framework web propi. Per a ser més exactes, el que es pretén en aquest punt és ampliar les funcionalitats d'un marc de treball ja existent, concretament de MyFaces. En cap cas es vol implementar un nou framework partint de zero. Per aquest motiu s'ha decidit canviar la descripció de l'objectiu donada a la introducció d'aquesta memòria (desenvolupament d'un framework web propi) per la que apareix com a títol d'aquest capítol (ampliació d'un framework: MyFaces).

3.1. Anàlisi

Després de l'estudi realitzat sobre MyFaces en el capítol anterior, cal aprofundir més en la seva estructura, organització i funcionament per poder-lo ampliar.

MyFaces és un marc orientat al component, és a dir, que la programació de la interfície amb l'usuari es fa per mitjà de components. Això és un dels seus punts forts, ja que facilita la seva ampliació i la reutilització de codi.

Així doncs, el que cal fer per a afegir noves funcionalitats a MyFaces és implementar nous components que les portin a terme. I més en concret, el que interessa aquí és crear nous components d'interfície d'usuari (UIComponent). Com a component d'interfície d'usuari es pot entendre un control que s'aplica sobre una pantalla d'una aplicació (d'escriptori, web...) i amb el que habitualment l'usuari hi pot interaccionar (i pot respondre a events).

Aquests components tampoc no es desenvolupen partint de zero, sinó que sempre que sigui possible, s'estén algun dels que ja existeixen en l'especificació JSF i s'han implementat per a MyFaces. De fet, n'hi ha uns quants que es poden considerar com a primaris perquè serveixen per a incorporar les propietats mínimes fonamentals de qualsevol component i que són una bona base per a qualsevol altre.

Ara cal veure quines mancances té MyFaces, per tal de poder-ne implementar alguna. Aquest framework és molt complet i, a més, hi ha nombrosos projectes de codi lliure que treballen per ampliar-lo i millorar-lo. Per això és difícil trobar algun component que no existeixi ja.

Per a resoldre aquest problema, se li ha donat la volta al plantejament dels objectius implicats. El segon objectiu és ampliar un framework ja existent i el tercer és desenvolupar una aplicació on es mostri el seu funcionament.

El que s'ha fet és decidir quina aplicació es vol desenvolupar i, a partir d'aquí, pensar quins components poden ser útils i que MyFaces no inclou, o si més no, no inclou amb totes les funcionalitats desitjades.

Les opcions són diverses, però les que cobren més força són: *LlistaDoble*, *TaulaDades* i *Opcions*.

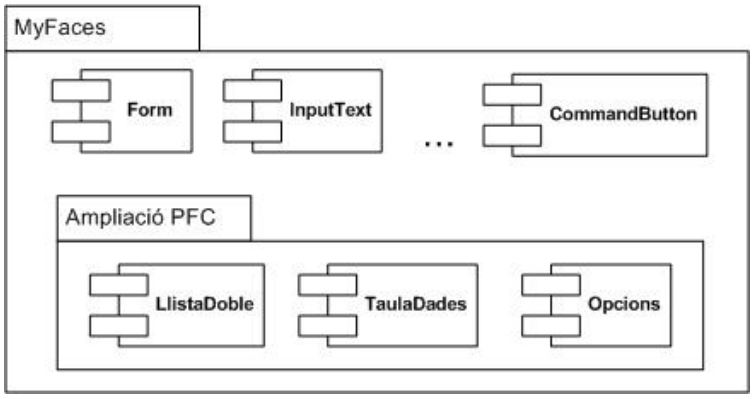


Figura 3.1:Ampliació de MyFaces

A continuació es descriu breument cadascun d'ells:

- *LlistaDoble*: component que permet seleccionar diversos elements, passant-ne un o més d'una llista a una altra, i ordenar-los.

  Funcionalitats:
  - Seleccionar/deseleccionar una o més opcions d'una llista.
  - Seleccionar/deseleccionar tots els elements d'una llista.
  - Desplaçar un o més elements de la llista seleccionada amunt i avall per a ordenar-la.

  A l'aplicació de mostra, al fer la gestió de les enquestes, pot ser útil per a triar les preguntes que ha d'incloure una enquesta.

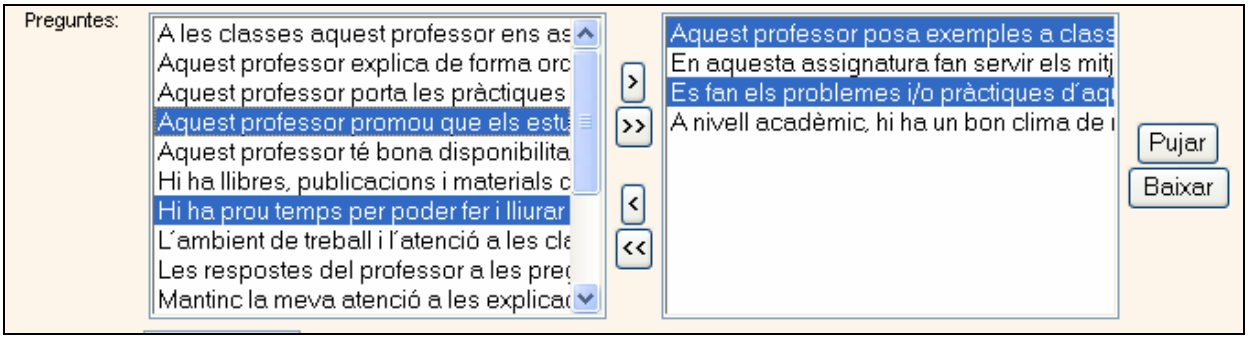
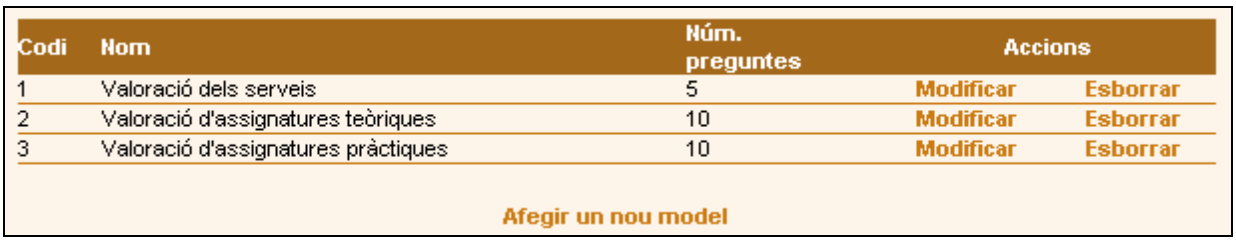


Figura 3.2:Mostra del component *LlistaDoble*

- *TaulaDades*: component que correspon a una taula de dades, els registres de la qual es poden modificar, esborrar, ordenar...

  Funcionalitats:
  - Paginar el llistat.
  - Ordenar els registres a partir d'un camp determinat.
  - Seleccionar un registre per a consultar-lo i/o modificar-lo.
  - Seleccionar un registre per a esborrar-lo.
  - Afegir un nou registre.

  A l'aplicació de mostra, al fer la gestió de les enquestes, pot ser útil per a tenir una visió general dels models definits i fer-ne el manteniment (altes, baixes, consultes i modificacions).



Codi	Nom	Núm. preguntes	Accions	
1	Valoració dels serveis	5	Modificar	Esborrar
2	Valoració d'assignatures teòriques	10	Modificar	Esborrar
3	Valoració d'assignatures pràctiques	10	Modificar	Esborrar

Afegir un nou model

Figura 3.3:Mostra del component *TaulaDades*

- *Opcions*: component per a mostrar un text i triar un valor entre 0 i un valor màxim. Aquesta ampliació es decideix fer-la a mig projecte (veure apartat de dificultats dins d'aquest mateix cap

  Funcionalitats:
  - Triar un valor entre els possibles.

  A l'aplicació de mostra, al generar una enquesta, pot ser útil per a representar cadascuna de les preguntes que ha de contestar l'usuari.

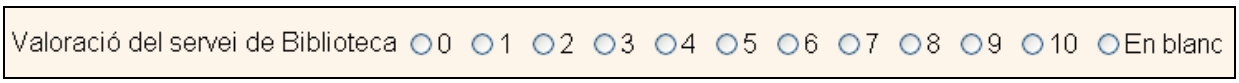


Figura 3.4:Mostra del component *Opcions*

3.2. Disseny

Un component de MyFaces està format per diversos elements:

- **Classes del component:**
  - **Classe UIComponent:** classe de Java derivada d'un altre component que l'estén i inclou la lògica o el cos del component.
  - **Classe Tag:** classe de manipulació del Tag (marca o etiqueta) del component. Gestiona les seves propietats i atributs.
  - **Classe Renderer:** classe de representació del component. Genera el codi necessari per a mostrar el component en el client i en recupera el valor. En pot tenir més d'una si es vol reproduir el component de diferent manera segons els client que l'utilitzi. No és necessària però sí recomanable si es vol aconseguir la independència del client.
  - **Classe Converter:** classe que converteix el valor del component al tipus de dades oportú. Si no s'implementa una classe de representació (renderer), no és necessària ja que la conversió la porta a terme el propi component en la fase de validacions de procés del cicle de vida.
  - **Classe Validator:** classe que permet validar determinats aspectes del component. No és necessària si no es vol fer cap comprovació específica d'algun atribut o propietat.
- **Descriptor del Tag (TLD)**: fitxer XML que descriu les marques necessàries per a utilitzar el component en documents JSPs. Associa les etiquetes a les classes que manipulen els seus atributs i propietats.
- **Descriptor de desplegament del component:** fitxer XML que enregistra, entre d'altres dades, la definició del component i del renderer. Per defecte s'anomena faces-config.xml.

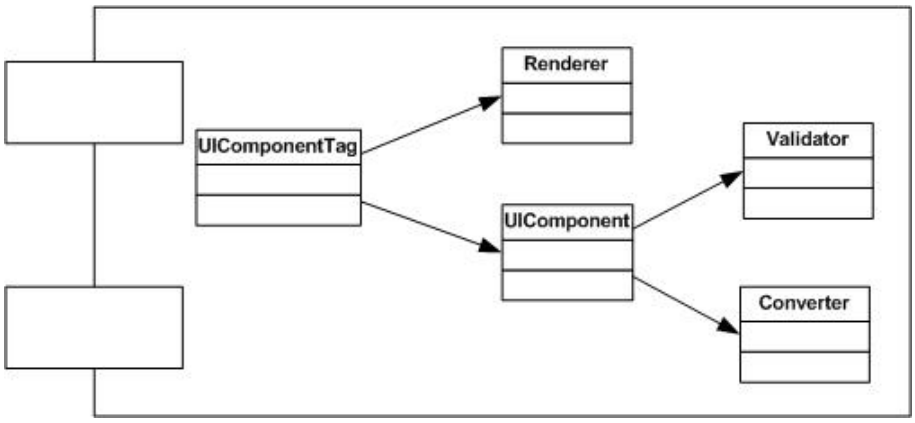


Figura 3.5:Estructura de classes d'un component de MyFaces

3.2.1. Decisions de disseny

El disseny dels components a implementar es vol fer de manera estructurada i preparada per a si en futures versions es volen ampliar les seves funcionalitats o presentacions per a diferents clients (tot i que en aquests moments no sigui necessari). Per això, es separen Tag i Renderer del propi component.

Cada component s'ha de desenvolupar individualment i independentment de la resta. Això permetrà poder crear un objecte per a cadascun d'ells que es pugui distribuir allà on convingui.

En principi, aquests components s'han de poder utilitzar en pàgines JSPs, ja que l'aplicació està pensada per a clients amb navegadors web.

La forma més habitual de desenvolupar un nou component per a MyFaces és estenent-ne un altre d'existent que serveixi com a base del seu comportament. Per això cal veure la jerarquia de components JSF i quines característiques tenen cadascun d'ells.

A continuació es detallen quins seran els components a partir del qual es desenvoluparan *LlistaDoble*, *TaulaDades* i *Opcions*:

Component	Classe a estendre	Propietats a aprofitar
LlistaDoble	UISelectMany	Retorn de múltiples elements seleccionats
TaulaDades	UIData	Estructura de taula, amb registres i camps
Opcions	UIInput	Retorn d'un valor numèric

Taula 3.1:Components a estendre per a desenvolupar els propis

Els diagrames que es mostren tot seguit permet veure la jerarquia de components dels 3 casos estudiats:

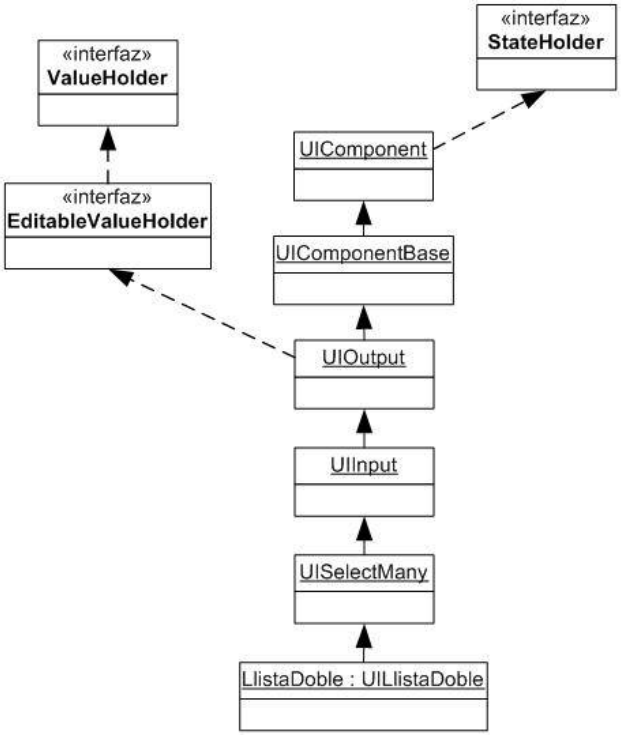


Figura 3.6:Jerarquia del component *LlistaDoble*

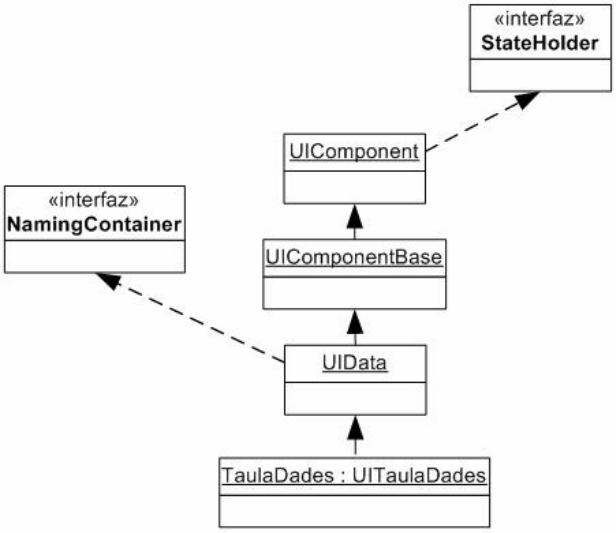


Figura 3.7:Jerarquia del component *TaulaDades*

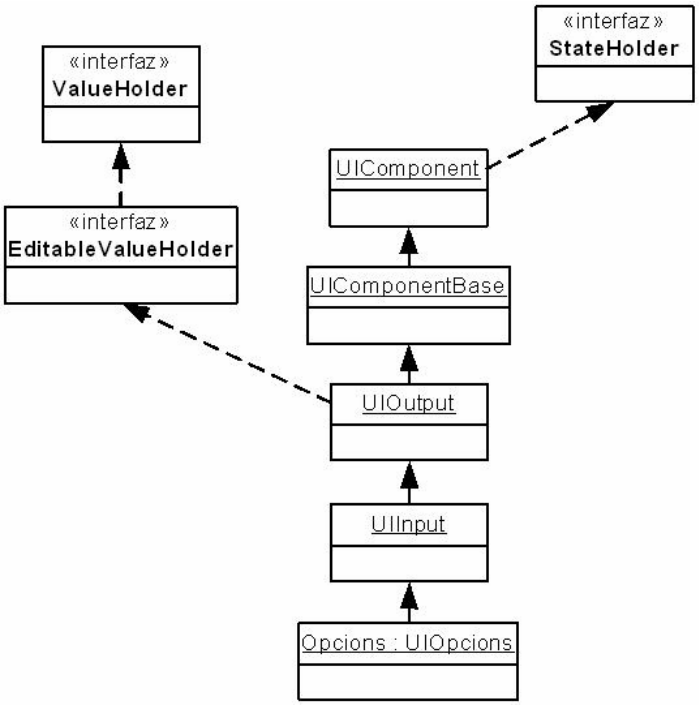


Figura 3.8:Jerarquia del component *Opcions*

3.2.2. Estructura dels components

En aquest punt ja es pot veure l'estructura dels components: quins elements els formen i de quins altres components són descendents.

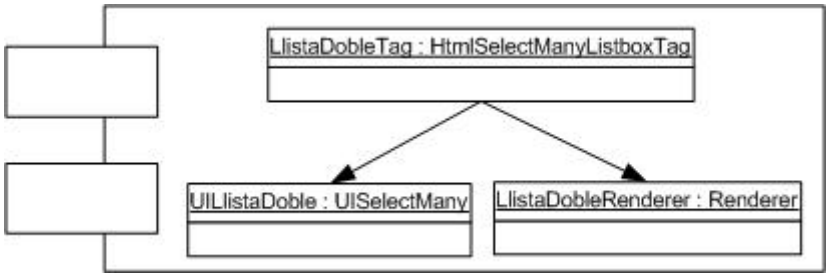


Figura 3.9:Estructura del component *LlistaDoble*

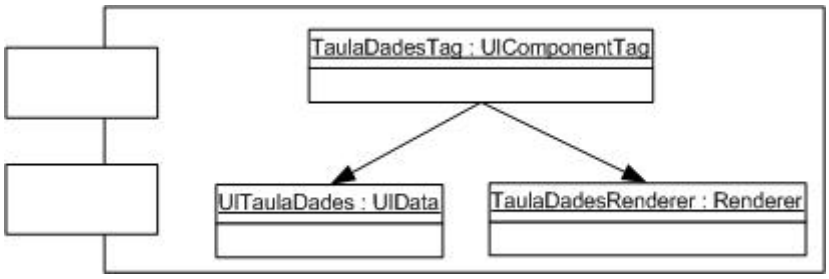


Figura 3.10:Estructura del component *TaulaDades*

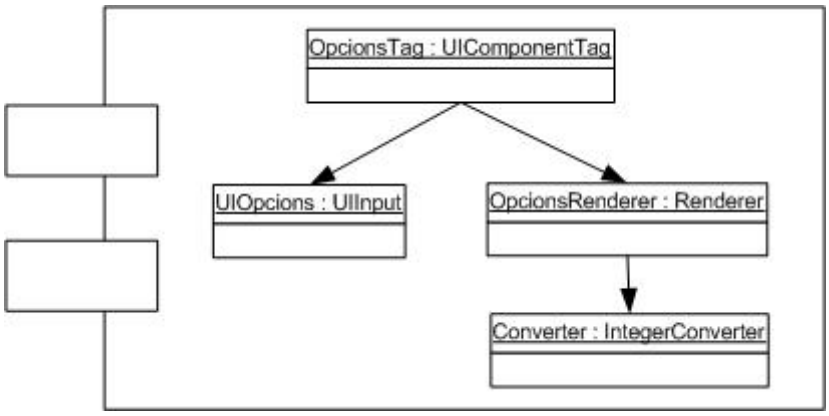


Figura 3.11:Estructura del component *Opcions*

3.2.3. Component LlistaDoble

El disseny de la interfície d'usuari del component es basa en format HTML. Està compost per diversos controls:

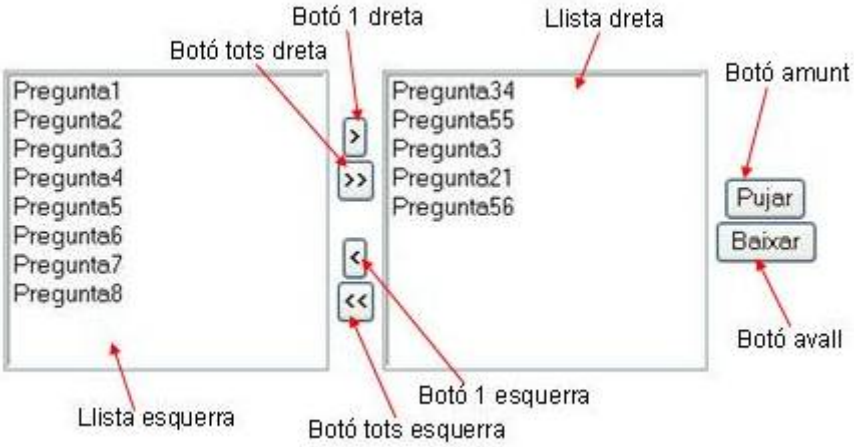


Figura 3.12:Elements del component *LlistaDoble*

- **Llista esquerra:** llista d'elements a seleccionar.
- **Llista dreta:** llista d'elements seleccionats. És el valor que torna el component com a resultat. Es poden ordenar els seus elements.
- **Botó 1 dreta:** botó per a moure de la llista de l'esquerra a la de la dreta un o més elements, aquells que estiguin marcats. És a dir, permet seleccionar una o més opcions.
- **Botó tots dreta:** botó per a moure tots els elements de la llista de l'esquerra a la de la dreta. És a dir, permet seleccionar tots els elements.
- **Botó 1 esquerra:** botó per a moure de la llista de la dreta a la de l'esquerra un o més elements, aquells que estiguin marcats. És a dir, permet treure de la selecció una o més opcions.
- **Botó tots esquerra:** botó per a moure tots els elements de la llista de la dreta a la de l'esquerra. És a dir, permet deseleccionar tots els elements.
- **Botó amunt:** botó per a moure els elements marcats de la llista de la dreta una posició cap amunt.
- **Botó avall:** botó per a moure els elements marcats de la llista de la dreta una posició cap avall.

Aquest component està pensat per a ser utilitzat en pàgines JSP, per la qual cosa necessita tenir definit un Tag (*dobleLlista*). Els seus atributs i propietats són els següents:

- **id:** nom identificador del component dins del document JSP.
- **size:** llargada de les llistes. Número d'elements visibles sense scroll. Si no s'especifica, per defecte són 10.
- **width:** amplada de les llistes (en píxels). Si no s'indica, per defecte són 300px.
- **llistaOrigen:** elements de la llista esquerra. Elements disponibles per a ser seleccionats. És un atribut obligatori.
- **llistaDesti:** elements de la llista dreta. Elements seleccionats.
- **value:** llista d'elements seleccionats.

La sintaxi d'utilització del component en un JSP és la següent:

```
<%@ taglib uri="http://components/llistaDoble" prefix="nou" %>

<nou:dobleLlista id=".."
                 size=".."
                 width=".."
                 llistaDesti=".."
                 llistaOrigen=".."
                 value=".." />
```

3.2.4. Component Opcions

El disseny de la interfície d'usuari del component es basa en format HTML. Està compost per diversos controls:

Figura 3.13:Elements del component *Opcions*

- **Text:** Text o descripció.
- **Opcions:** Cadascun dels possibles valors que pot retornar el camp. Inclou una opció extra per a deixar la resposta en blanc.

Aquest component està pensat per a ser utilitzat en pàgines JSP, per la qual cosa necessita tenir definit un Tag (*opcions*). Els seus atributs i propietats són els següents:

- **id:** nom identificador del component dins del document JSP.
- **text:** text o descripció que acompanya les opcions.
- **quants:** Número de possibles opcions a triar. Si no s'indica, per defecte són 10.
- **value:** valor marcat dins de les opcions disponibles. És un atribut obligatori.

La sintaxi d'utilització del component en un JSP és la següent:

```
<%@ taglib uri="http://components/opcions" prefix="nou" %>

<nou:opcions id=".."
             text=".."
             quants=".."
             value=".." />
```

3.3. Implementació

La implementació dels components s'ha portat a terme per separat. S'ha creat un projecte per a cadascun d'ells i s'ha obtingut una llibreria en cada cas, a punt per a ser desplegades en l'aplicació que interessi.

3.3.1. Procés

Les passes a seguir per a implementar un nou component són:

1. Crear la classe del component, estenent aquella que sigui més apropiada.
    a. Indicar el renderer a utilitzar.
    b. Especificar el convertidor associat.
2. Definir la classe de renderització o presentació del component en el client.
    a. Descriure l'aspecte que ha de tenir el component.
    b. Detallar el seu funcionament.
    c. Indicar el tractament del valor a recuperar.
3. Implementar la classe de manipulació del Tag.
    a. Gestionar els valors de propietats i atributs.
4. Crear el descriptor de marques (fitxer TLD).
    a. Definir el punt de referència a partir del qual es podrà utilitzar el component en els JSPs.
    b. Indicar els tags necessaris.
5. Configurar el descriptor de desplegament del component (faces-config.xml).
    a. Enregistrar el component i el seu renderer.
6. Desplegar el component.
    a. Crear l'estructura de directoris apropiada.
    b. Compilar les classes.
    c. Compactar les classes compilades per a obtenir un component desplegable (.jar).
    d. Desplegar el component a l'aplicació oportuna (directori /WEB-INF/lib).

3.3.2. Component LlistaDoble

La composició interna de *LlistaDoble* per a la presentació dissenyada es mostra en la taula següent:

Component	Representació	Control HTML
Llista esquerra	Llista de selecció múltiple	<select><option>...</select>
Llista dreta	Llista de selecció múltiple	<select><option>...</select>
Botó 1 dreta	Botó	<input type="button">
Botó tots dreta	Botó	<input type="button">
Botó 1 esquerra	Botó	<input type="button">
Botó tots esquerra	Botó	<input type="button">
Botó amunt	Botó	<input type="button">
Botó avall	Botó	<input type="button">

Taula 3.2:Controls HTML del component *LlistaDoble*

El funcionament dels botons es fa per mitjà de funcions en Javascript. Si totes les dades es carreguen inicialment a les llistes, no té sentit anar a cada acció de selecció o ordenació al servidor.

Internament, el valor es controla amb un camp HTML ocult (<input type="hidden">), que guarda en tot moment la llista d'elements seleccionats (aquells que estan a la llista de la dreta).

3.3.3. Component Opcions

La composició interna de *LlistaDoble* per a la presentació dissenyada es mostra en la taula següent:

Component	Representació	Control HTML
Text	Text	<label>
Opcions	Opcions de selecció	<input type="radio">

Taula 3.3:Controls HTML del component *Opcions*

Internament, el valor es controla directament a partir dels camps HTML de les opcions. Es tracta com a valor numèric (enter).

3.3.4. Entorns de desenvolupament i proves

Tots dos entorns són el mateix i es caracteritzen per:

- En un sol equip físic hi ha el client (navegador web) i el servidor d'aplicacions J2EE (que inclou el servidor web).
- Configuració de l'equip:
  - Sistema operatiu: Windows XP professional
  - Entorn de desenvolupament: Exadel 4.1 sobre Eclipse 3.2
  - Suport compilació: Apache Ant 1.6.5
  - Servidor J2EE d'aplicacions: JBoss 4.0.4 GA
  - Servidor web: Apache Tomcat 5.5.17

3.4. Dificultats

Al llarg del desenvolupament de la fase d'ampliació de MyFaces, han sorgit nombrosos problemes, alguns fruits de la inexperiència amb la plataforma J2EE i d'altres típics de qualsevol projecte de software.

En un moment determinat, això ha provocat haver-se de fer alguns replantejaments i poder, d'aquesta manera, assolir els objectius principals del projecte.

La majoria d'aquests entrebancs han aparegut durant la implementació dels components. Sobre el paper s'ha pogut fer l'anàlisi i el disseny tal i com s'havien previst inicialment.

3.4.1. Problemes i solucions adoptades

A continuació es descriuen els problemes més importants que s'han trobat al llarg d'aquesta ampliació del framework i que han provocat una major desviació en la planificació del projecte. També es comenten les solucions que s'han adoptat, tot i que algunes no hagin donat el resultat esperat.

1. Context: Desconeixement previ de la plataforma J2EE, de la seva filosofia i manera de treballar.

   Problema: Amb aquest panorama és força complicat introduir-se en el tema començant per la teulada (creant uns components propis), en lloc de preparar els fonaments (provant exemples i fent petits projectes inicialment senzills).

   Solucions: Es busquen i es segueixen exemples i tutorials diversos per a una primera presa de contacte.

2. Problema: Els tutorials i exemples que es poden trobar per internet no sempre són de bona qualitat i funcionen. Fins i tot algun trobat en llibres i manuals conté errors difícils de trobar per a un principiant o bé no són prou exhaustius per a seguir-los completament.

   Solucions: Es dediquen més hores a fer proves, consultar dubtes a la tutora i a algun fòrum d'internet, buscar solucions...

3. Problema: L'entorn de desenvolupament escollit per a treballar, Eclipse amb plugins per a MyFaces, és molt inestable. Peta sovint i de vegades dóna errors en execucions successives sense haver canviat codi.

   Solucions: Es busca alguna eina alternativa. Es troba Exadel sobre Eclipse amb molt bones referències i un entorn més amigable per a entrar en contacte amb el desenvolupament sobre MyFaces. S'instal·la i es fan les proves pertinents.

4. Context: Es comença a implementar el component LlistaDoble. Es creen tots els elements necessaris i es fan proves.

   Problema: El seu funcionament en pantalla és l'esperat però no retorna cap valor.

   Solucions: Es fan canvis diversos en la implementació, es demana opinió a la tutora, es busquen problemes similars a diferents llistes d'internet, es rellegeixen manuals... A hores d'ara encara no s'ha pogut resoldre aquest problema.

5. Problema: La gran quantitat d'hores dedicades al component LlistaDoble (sense èxit perquè funcioni correctament) comencen a passar factura a la planificació prevista. La desviació de la temporització real respecte a la prevista és força important, tant que perilla l'èxit general del projecte.

   Solució: S'augmenta el número d'hores de dedicació setmanals al projecte (es passa d'un calendari de 20h a un de 30h aproximadament). Es replanteja algun objectiu. L'ampliació del framework consisteix a afegir alguna funcionalitat. Sembla del tot impossible poder implementar el component *TaulaDades* en els terminis disponibles. Es veu llavors una altra necessitat a l'aplicació de proves (que s'ha començat a implementar en paral·lel): el component *Opcions*. Es decideix, doncs, implementar aquest i tancar el segon a la fase de disseny.

6. Context: La implementació es porta a terme, com s'ha comentat abans, amb l'entorn d'Exadel sobre Eclipse. Com a servidor web s'utilitza Apache Tomcat. Les proves de funcionament del component LlistaDoble es fan amb BackingBeans com a suport per a les dades. És a dir, només es treballa amb la capa de presentació de l'aplicació, sense accés a cap base de dades. A l'haver d'utilitzar EJB per a desenvolupar l'aplicació de mostra (que es va fent en paral·lel per arribar a temps), cal instal·lar JBoss com a servidor d'aplicacions.

   Problema: El component i totes les opcions fetes fins llavors de l'aplicació deixen de funcionar. El sistema dóna l'error "Context Faces not found".

   Solucions: Es busca per internet i es proven algunes de les solucions trobades. Cap funciona. Es consulta de nou a la tutora. Es canvien les llibreries que utilitza JBoss com a suport a JSF i s'afegeixen les pròpies de MyFaces als projectes en desenvolupament. Funciona !

4. Desenvolupament d'una aplicació de mostra

El tercer gran objectiu del projecte consisteix a desenvolupar una aplicació on es mostri el funcionament del framework ampliat. És a dir, fer una aplicació on s'utilitzin els components implementats sobre MyFaces.

4.1. Anàlisi

Tal i com diu el propi títol del projecte, l'aplicació de mostra és "Enquestes de satisfacció de la docència".

Aquesta aplicació sorgeix per a cobrir una necessitat real en una empresa determinada. Es tracta d'una Escola Universitària on, cada final de semestre, es passen enquestes als alumnes perquè valorin les assignatures, els professors, els serveis del centre... A partir de les dades recollides, s'elaboren diverses estadístiques i se n'extrauen conclusions.

Actualment, aquest procés es fa manualment. És a dir, els alumnes omplen les enquestes en format paper a classe. Hi ha una persona encarregada d'introduir aquestes dades a un full de càlcul. I a partir d'aquí hi ha un responsable que genera les estadístiques necessàries.

Aquesta manera de fer comporta diversos inconvenients:

- Gran quantitat de feina per introduir totes les respostes de les enquestes. Això implica un cost temporal i econòmic considerable, especialment quant més gran és el número d'enquestes que es passen i major el volum d'alumnes que l'han respost.
- Possibilitat de caure en errors a l'introduir manualment els resultats a un full de càlcul.
- Dades emmagatzemades de manera poc pràctica per a poder-ne fer consultes diverses.
- Despesa innecessària de paper (cal protegir el mediambient).

Per tot això es proposa desenvolupar una aplicació que permeti gestionar les enquestes informàticament, que l'alumne les pugui respondre des d'un laboratori del centre o des de casa via web i que automàticament s'enregistrin els resultats a una base de dades.

"Enquestes de satisfacció de la docència" té 3 mòduls ben diferenciats:

- Gestió de la informació necessària per a poder fer les enquestes.
- Generació i realització de les enquestes.
- Càlcul d'estadístiques a partir dels resultats.

A causa de la limitació de temps per a desenvolupar aquest projecte, no es pretén desenvolupar l'aplicació completa, sinó que només es portaran a terme algunes opcions que permetin fer-se una idea general del conjunt.

Bàsicament es tractaran les dues primeres parts. Tant en la gestió de la informació com en la realització d'enquestes es podran utilitzar diversos formularis que permetin mostrar el funcionament dels components descrits al capítol anterior.

El subsistema de les estadístiques queda al marge d'aquest projecte perquè és la que té menys rellevància a nivell de presentació en entorns web. El realment interessant aquí és la manera de calcular les diferents estadístiques i això correspon a la lògica de negoci.

4.1.1. Concepte de negoci

Els conceptes bàsics implicats en l'aplicació de mostra són els següents:

- Model d'enquesta: es pot entendre com una plantilla de definició d'enquestes amb una descripció, el número de preguntes que ha de tenir...
- Pregunta: bàsicament és l'enunciat de les preguntes que s'assignaran a les diferents enquestes.
- Professor: algunes enquestes es fan per valorar la docència del professorat, per tant, cal tenir en compte certa informació relacionada.
- Assignatura: altres enquestes serveixen per valorar el funcionament, l'organització... de les assignatures, per tant, cal tenir present aquest concepte.
- Enquesta: representa la unió d'un conjunt de preguntes que es basen en un model d'enquesta i que valoren un professor, una assignatura...
- Resultat: correspon als valors donats pels alumnes al respondre les enquestes.

Val a dir que els conceptes de professor i assignatura es tracten aquí de retruc (només per a poder-ne fer referència), ja que la seva gestió pertany a una altra aplicació que el centre ja té implementada: "Gestió acadèmica".

A continuació es mostren aquests conceptes i la relació entre ells.

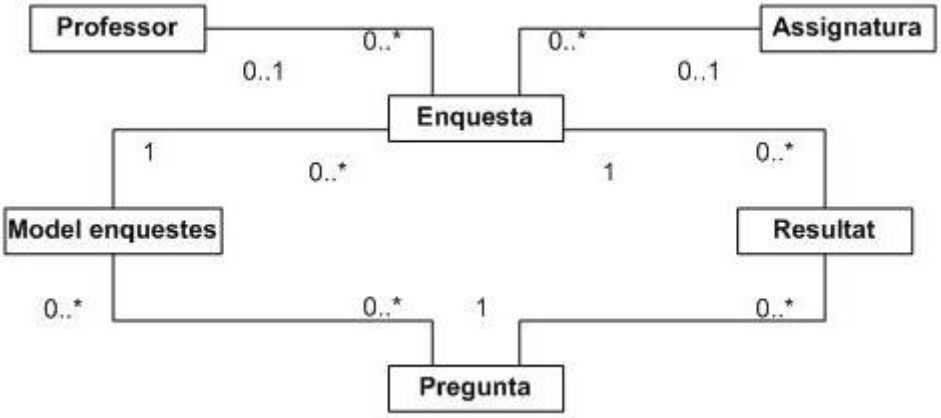


Figura 4.1:Diagrama del concepte de negoci

4.1.2. Casos d'ús

Per a completar l'anàlisi funcional de l'aplicació de mostra, cal determinar quins tipus d'usuari hi ha i quines interaccions tenen amb el sistema.

Aquesta informació es pot sintetitzar en el diagrama de casos d'ús, obtenint així una visió general de les funcionalitats de l'aplicació i qui pot portar-les a terme.

En aquest cas es mostra un diagrama de casos d'ús per a cadascun dels mòduls a implementar:

- Mòdul 1: Gestió de la informació
- Mòdul 2: Realització d'enquestes

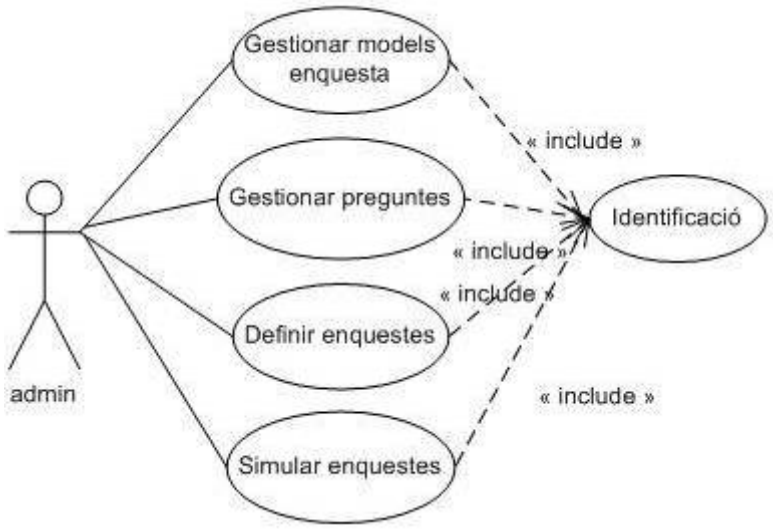


Figura 4.2:Diagrama de casos d'ús del mòdul de Gestió de la informació

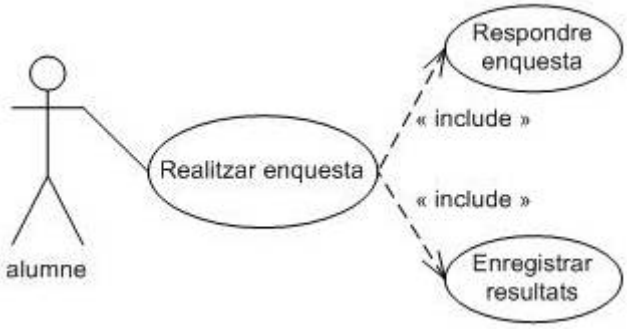


Figura 4.3:Diagrama de casos d'ús del mòdul de Realització d'enquestes

Els diferents tipus d'usuari (segons el rol que desenvolupen) són:

- **Admin o administrador:**

  Responsable de gestionar les enquestes. Aquest paper el representa habitualment el cap d'estudis del centre.

- **Alumne:**

  Qualsevol alumne matriculat al centre.

Les funcionalitats que ofereix l'aplicació, representades com a casos d'ús en el diagrama, es detallen a continuació:

- **Identificació:**

  Permet a l'usuari que fa d'administrador identificar-se al sistema.

- **Gestionar models enquesta:**

  Correspon a una típica gestió de dades, és a dir, a donar d'alta, de baixa, consultar i modificar models d'enquesta.

- **Gestionar preguntes:**

  Correspon a una típica gestió de dades, és a dir, a donar d'alta, de baixa, consultar i modificar preguntes.

- **Definir enquestes:**

  Correspon a una típica gestió de dades, és a dir, a donar d'alta, de baixa, consultar i modificar enquestes.

- **Simular enquestes:**

  Permet a l'usuari que fa d'administrador generar enquestes simulant que és un alumne. En aquest cas no s'enregistren les respostes.

- **Realitzar enquesta:**

  Inclou tot el procés de generació i resposta d'una enquesta per part d'un alumne, amb el conseqüent registre dels resultats introduïts.

- **Respondre enquesta:**

  Permet a l'alumne generar una enquesta (segons uns patrons determinats) en format de formulari web i respondre-la.

- **Enregistrar resultats:**

  Correspon a l'enregistrament dels valors introduïts per un alumne en una enquesta.

4.2. Disseny

4.2.1. Decisions de disseny

Les decisions de disseny de l'aplicació de mostra estan condicionades per un dels dos fets següents:

- L'aplicació està pensada per a ser utilitzada en un entorn real, el qual imposa certes restriccions.
- L'aplicació ha de servir per demostrar que s'han adquirit uns coneixements de J2EE i que se sap com portar-los a la pràctica.

L'aplicació es desenvolupa per a un entorn web. És una aplicació "thin client", és a dir, el pes principal de la qual l'executa el servidor (i no el client).

La informació amb la que treballa el sistema s'emmagatzema en una base de dades relacional. Així, l'aplicació ha de ser capaç de comunicar-s'hi per a mantenir i consultar les seves dades.

L'arquitectura hardware és distribuïda. En concret està formada per 3 màquines que es corresponen amb les 3 capes físiques:

- **Equips clients.** Han de complir 2 requisits: estar connectats al servidor (ja sigui a través d'una xarxa interna o bé per internet) i tenir instal·lat un navegador web (Internet Explorer, Firefox...).
- **Servidor J2EE.** En aquest equip hi ha el servidor J2EE d'aplicacions (JBoss), que incorpora un servidor web (Tomcat).
- **Servidor de bases de dades.** Aquesta màquina té el SGBD o Sistema Gestor de Bases de Dades (Ms SQL Server 2000) que emmagatzema tota la informació necessària per al funcionament de l'aplicació.

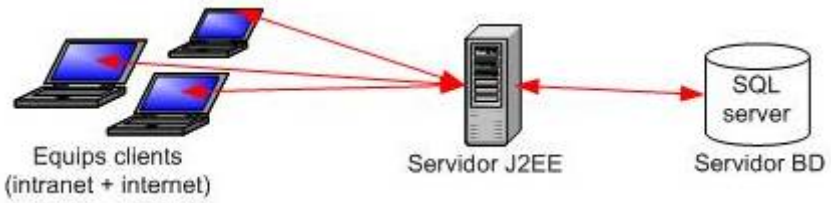


Figura 4.4:Arquitectura hardware

Sobre aquest entorn, l'arquitectura software s'hi adapta de la següent manera:

- Equips clients. Executen codi HTML en els seus navegadors. Envien peticions HTTP i reben respostes en format de plana web.

- Servidor J2EE.
  - Contenidor web: inclou els servlets i JSPs.
    - Servlet: per atendre les peticions d'usuari i fer de façana entre l'usuari i els EJB.
    - JSP: per a generar els resultats en format de pàgines web.
  - Contenidor EJB: inclou els EJB d'entitat i els de sessió.
    - EJB d'entitat: per a establir connexió amb les dades persistents de la BD
    - EJB de sessió: per a fer de façana entre els EJB d'entitat i les peticions del servlet i per a implementar la lògica de negoci.
  - Classes Java (POJO): desenvolupa part de la lògica de negoci.
- Servidor BD. Conté la BD relacional.

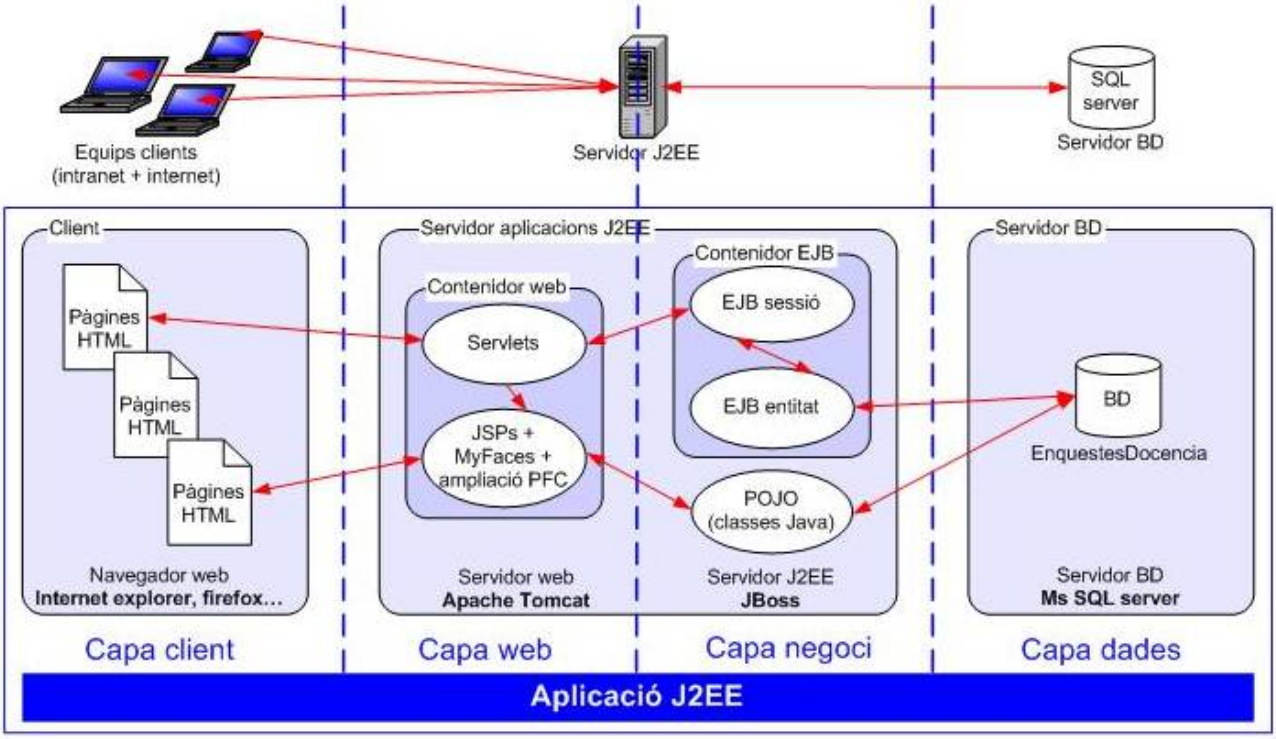


Figura 4.5:Arquitectura software

Com s'ha dit durant l'anàlisi, a nivell funcional l'aplicació es divideix en 3 mòduls ben diferenciats, dels quals en aquest projecte n'abasta 2. Cadascun d'ells s'ha de desenvolupar per separat, ja que a l'hora de passar-los a l'entorn real, aniran en contextos molt diferents.

- Mòdul 1: Gestió de la informació necessària per a poder fer les enquestes.

  A l'entorn real ha d'anar a la zona de la intranet reservada a Personal de l'Escola Universitària i sota identificació explícita.

- Mòdul 2: Realització de les enquestes i emmagatzemament dels resultats.

  A l'entorn real ha d'anar a la zona de la intranet accessible als alumnes. I dins d'aquesta se li ha de fer un enllaç des dels punts on interessi (web de les assignatures, per exemple).

4.2.2. Disseny de les interfícies d'usuari

La interfície d'usuari de l'aplicació es vol dissenyar de manera que sigui còmoda, pràctica i intuïtiva per a l'usuari. És una aplicació que es fa servir molt puntualment i, tot i que el perfil dels usuaris és bàsicament tècnic, no han de perdre temps recordant el seu funcionament i navegant entre les opcions.

En aquest sentit, el disseny de tota l'aplicació manté una uniformitat, sigui el mòdul que sigui i encara que cadascun vagi encaminat a un perfil d'usuari diferent.

Mòdul de Gestió de la informació

En primer lloc es presenta el diagrama de flux de les diferents pantalles d'aquest subsistema.

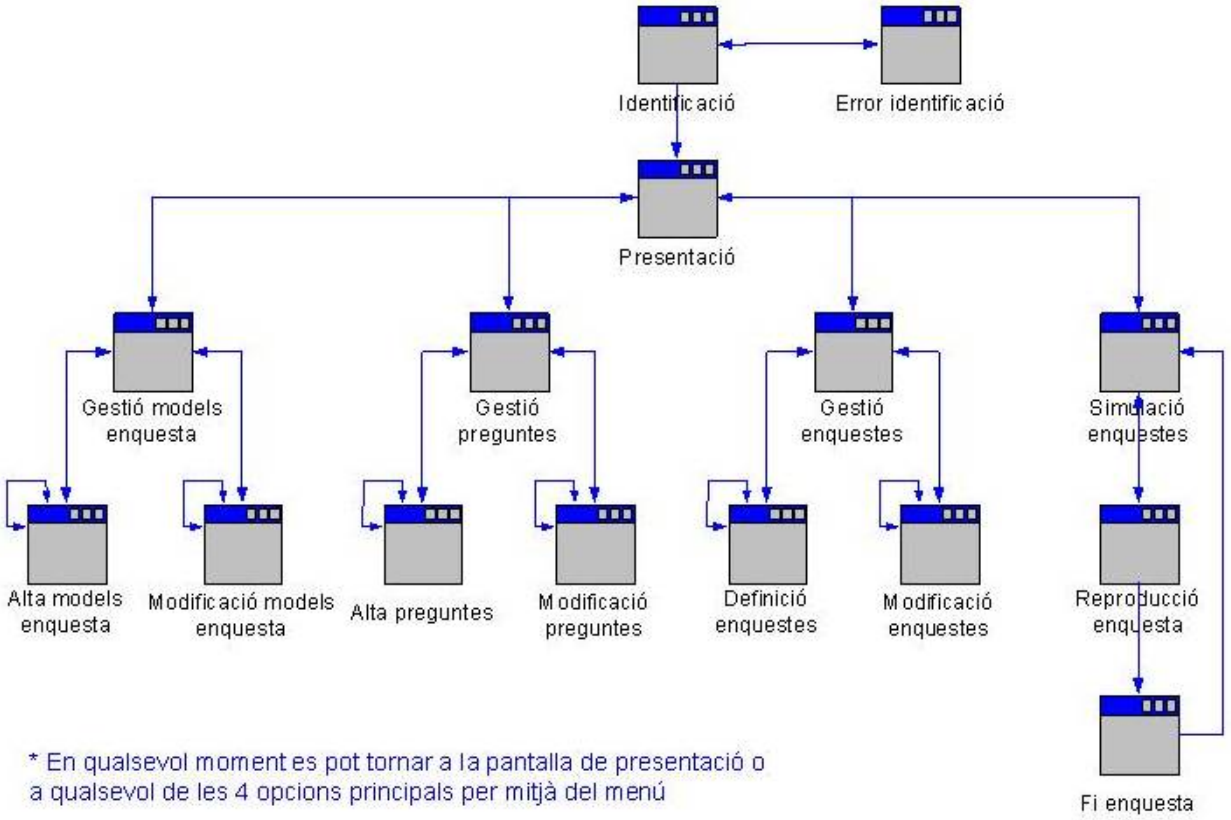


Figura 4.6:Diagrama de flux de les pantalles del mòdul de Gestió de la informació

Per tal d'afavorir la navegabilitat indicada en el diagrama anterior, totes i cadascuna d'elles tenen una estructura molt clara dividida en un parell de zones: un menú permanent (a la banda esquerra) i la zona de treball que varia en funció de l'opció seleccionada (tota la resta).

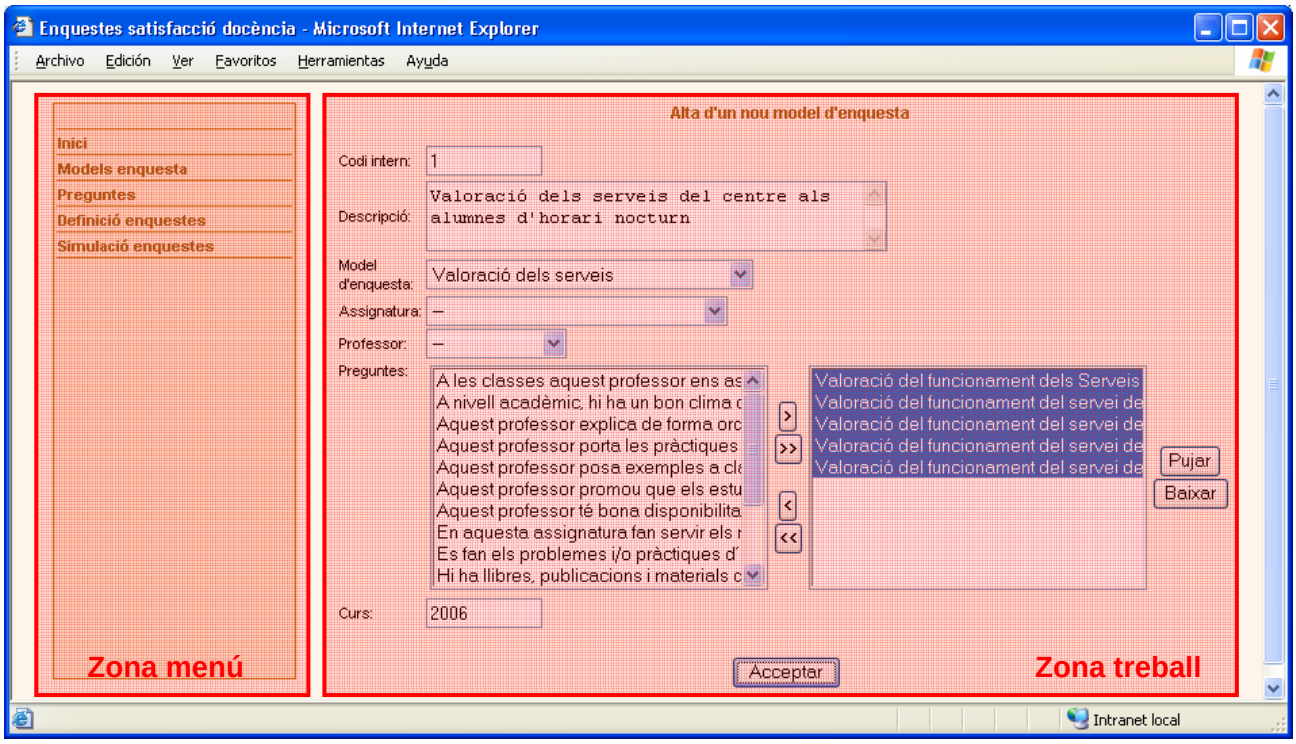


Figura 4.7:Estructura bàsica de totes les pantalles del mòdul Gestió de la informació

La zona de treball també manté el format en aquelles opcions que són similars, com per exemple, totes les gestions (preguntes, enquestes...), les altes, les modificacions...

A continuació es mostra el disseny d'una pantalla de cada tipus.

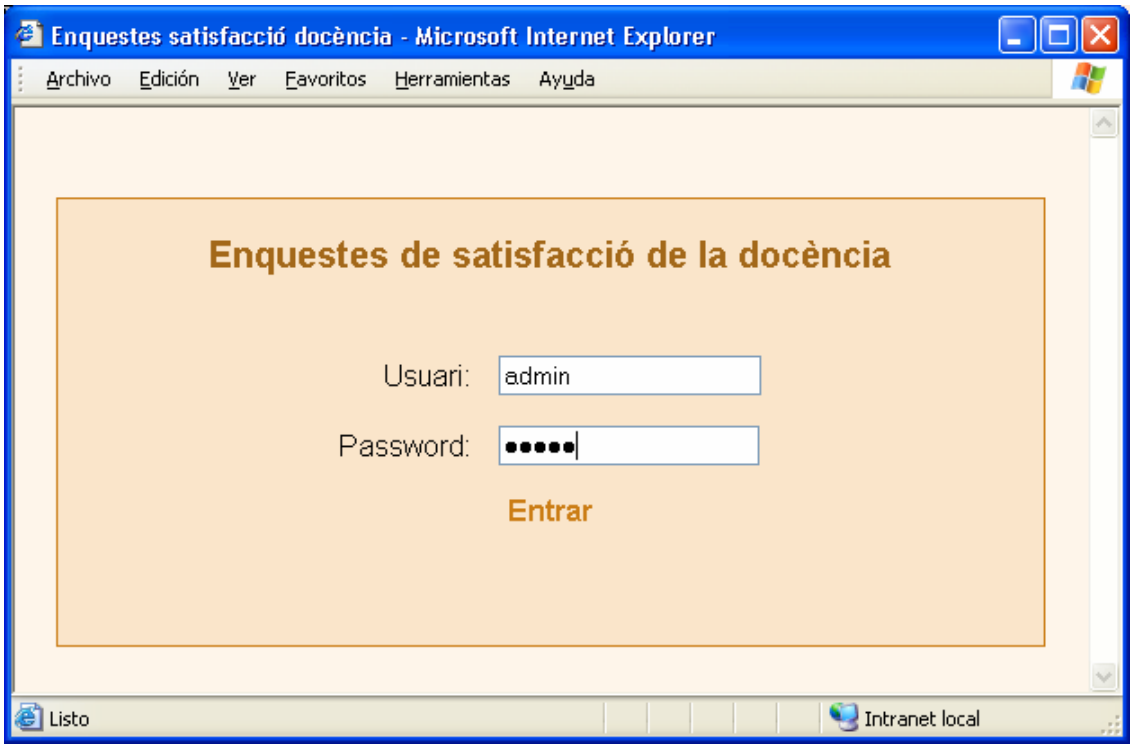


Figura 4.8:Pantalla d'identificació de l'usuari

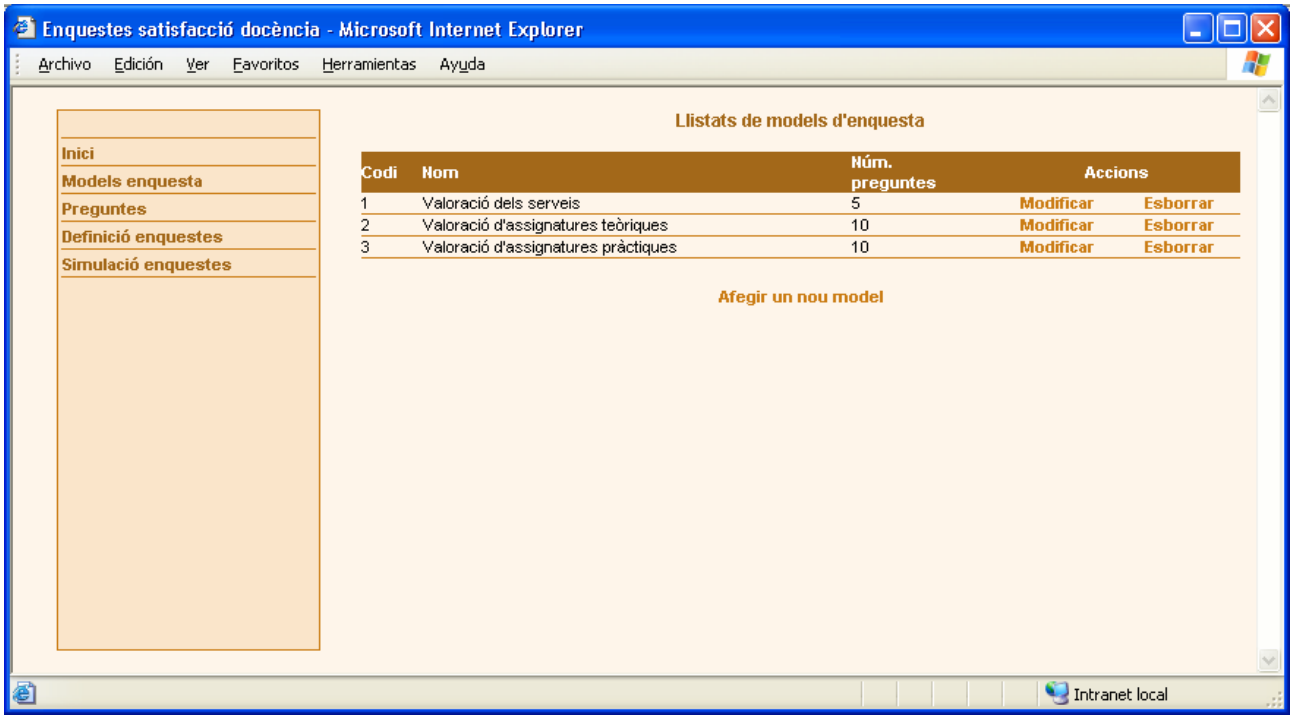


Figura 4.9:Pantalla de gestió de dades

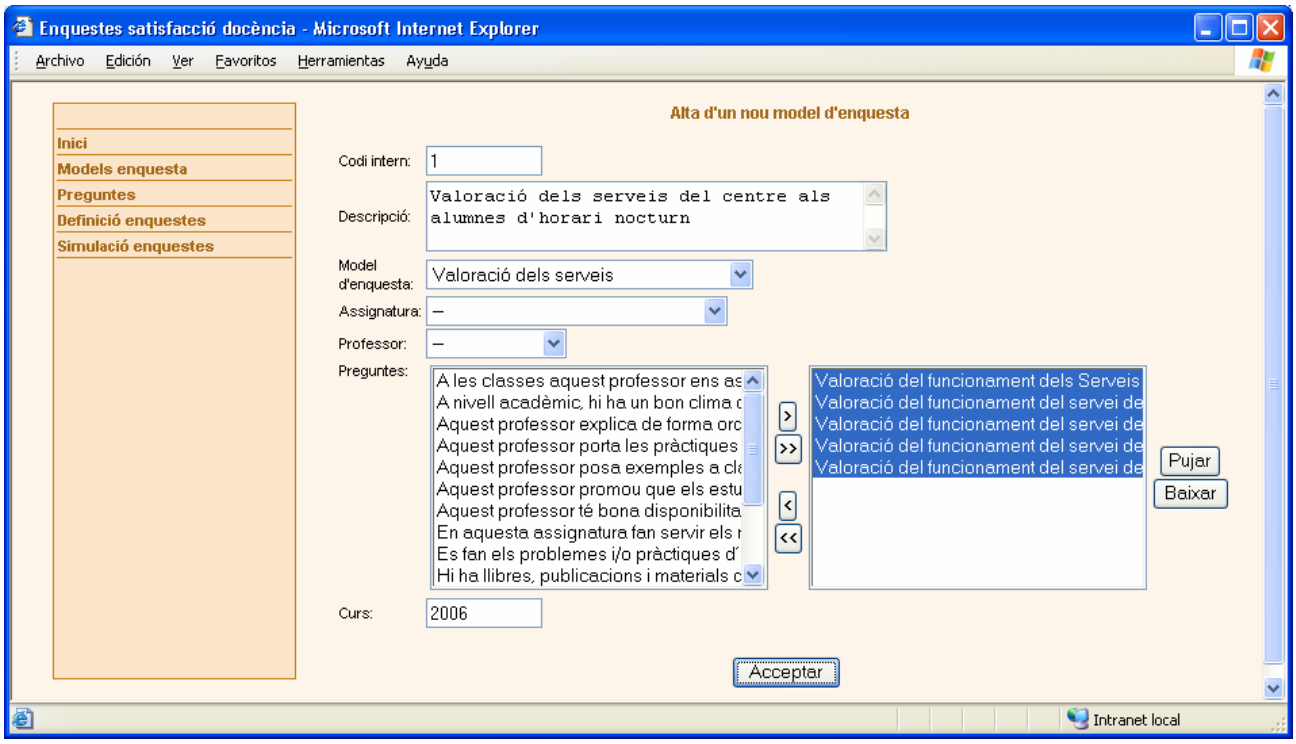


Figura 4.10:Pantalla d'alta/modificació de dades

Mòdul de Realització d'enquestes

En aquest subsistema, el diagrama de flux de les pantalles és força simple. De fet, quan s'implementi en l'entorn real, la pantalla de selecció d'enquestes desapareixerà ja que aquesta tria ja vindrà donada per l'entorn. L'alumne s'identificarà a la intranet de la Escola Universitària i en el lloc corresponent hi trobarà un enllaç que directament el conduirà a la generació de l'enquesta oportuna.

Figura 4.11:Diagrama de flux de pantalles del mòdul Realització d'enquestes

A continuació es mostra el disseny de les 2 primeres pantalles. La tercera simplement inclou un missatge de notificació de finalització de l'enquesta.

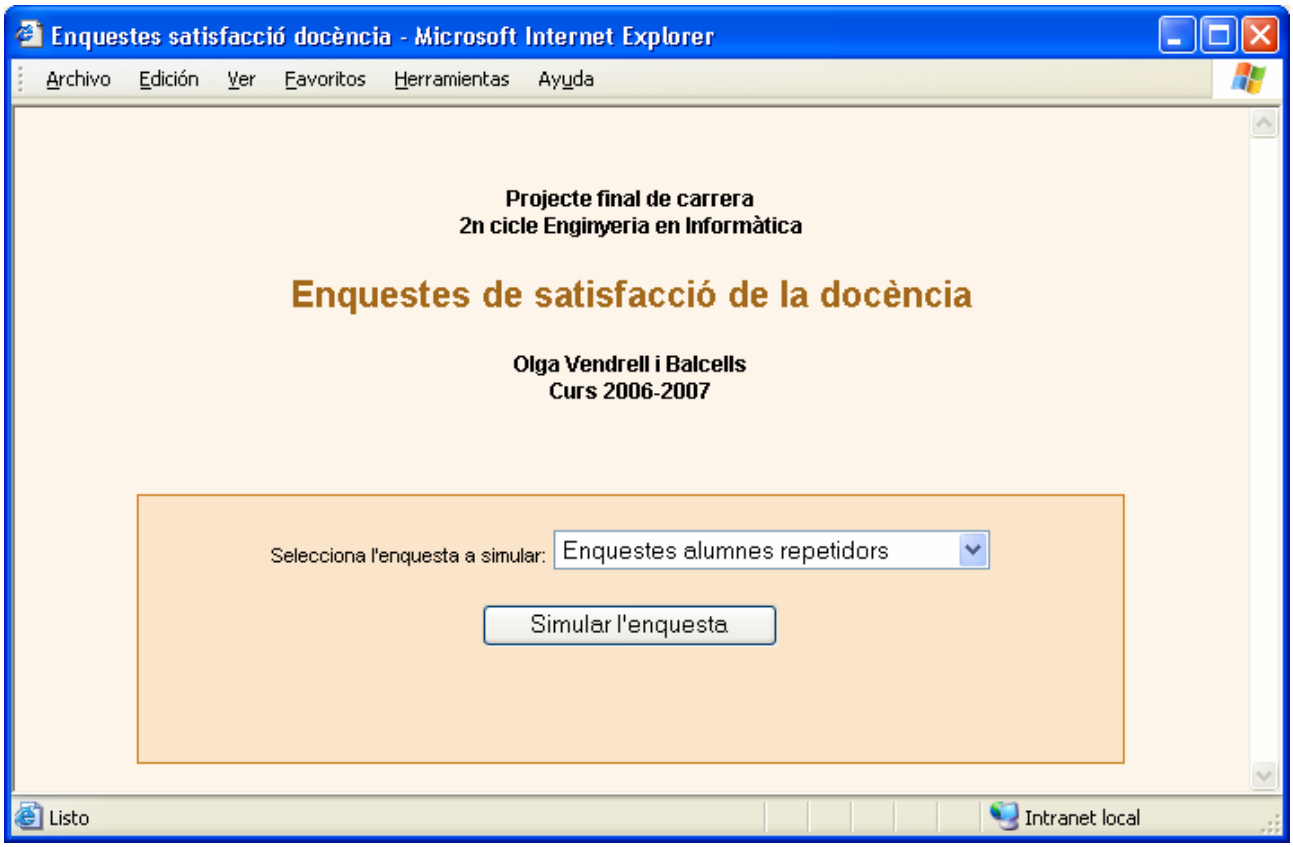


Figura 4.12:Pantalla de selecció de l'enquesta

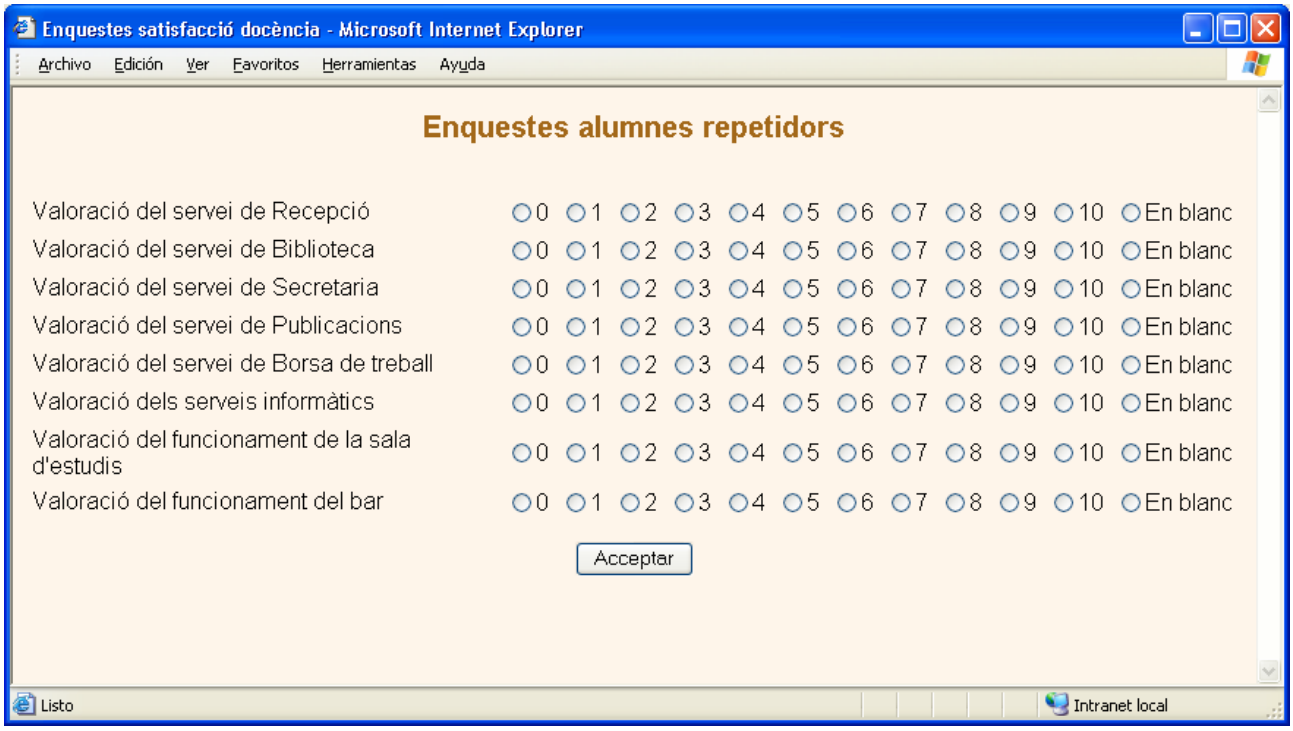


Figura 4.13:Pantalla de realització de l'enquesta

4.3. Base de dades

Per tal que el sistema "Enquestes de satisfacció de la docència" tingui sentit, cal emmagatzemar certa informació de manera persistent en una BD. En concret, cal guardar tot allò important en el concepte de negoci.

A continuació es mostra un diagrama on es recullen totes aquelles dades rellevants (i les seves relacions) per al bon funcionament de l'aplicació.

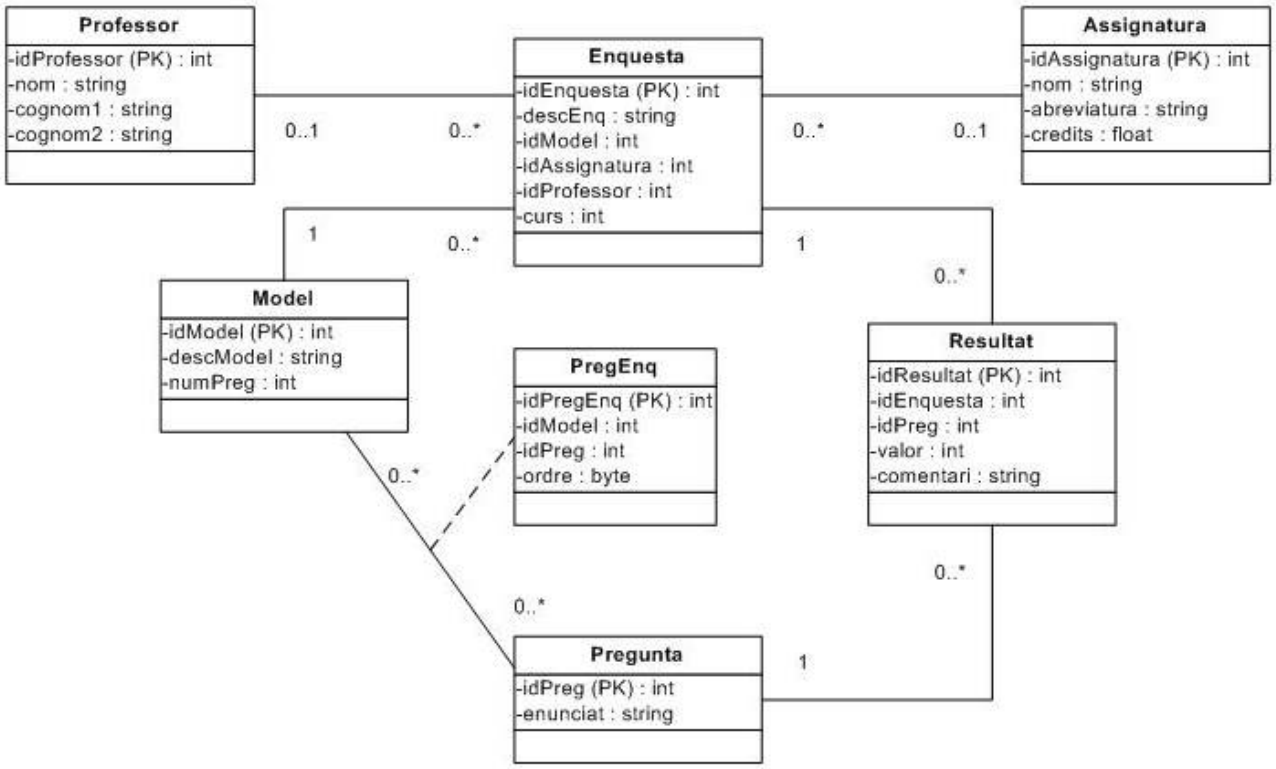


Figura 4.14:Model de dades Entitat-Relació

4.3.1. Disseny lògic

La base de dades definida per a aquesta aplicació està estretament lligada amb una altra que ja existeix en l'entorn on s'ha de desplegar. De fet, n'ha de passar a formar part i per això, hi ha algunes dades que venen donades per aquesta relació, com per exemple les de professors i assignatures. Pel mateix motiu, la BD serà de tipus relacional, és a dir, organitzada en taules, registres i camps.

A continuació es defineix el disseny lògic, a partir de les dades recollides en el model Entitat-Relació.

```
PROFESSOR (idProfessor, nom, cognom1, cognom2)

ASSIGNATURA (idAssignatura, nom, abreviatura, credits)

PREGUNTA (idPreg, enunciat)

MODEL (idModel, descModel, numPreg)

PREGENQ (idPregEnq, idModel, idPreg, ordre)
on (idModel, idPreg) és clau única,
on (idModel, ordre) és clau única,
on (idModel) fa referència a MODEL,
on (idPreg) fa referència a PREGUNTA
```

```
ENQUESTA (idEnquesta, descEnq, idModel, idAssignatura, idProfessor, curs)
on (idModel) fa referència a MODEL,
on (idAssignatura) fa referència a ASSIGNATURA,
on (idProfessor) fa referència a PROFESSOR

RESULTAT (idResultat, idEnquesta, idPreg, valor)
on (idEnquesta) fa referència a ENQUESTA,
on (idPreg) fa referència a PREGUNTA
```

4.3.2. Disseny físic

La BD es defineix sobre un servidor Ms SQL server 2000.

L'aplicació de mostra accedeix al servidor BD mitjançant JDBC (Java DataBase Connectivity), una interfície que fa transparent al desenvolupador el SGBD físic que hi ha al darrera. En cas de canviar de tipus de servidor BD, n'hi ha prou amb canviar les llibreries del servidor d'aplicacions (drivers) i modificar la cadena de connexió en el codi.

El disseny físic de la BD es pot consultar en l'Annex C d'aquesta mateixa memòria.

4.4. Implementació

La implementació de l'aplicació s'ha portat a terme separant els dos mòduls: gestió de la informació i realització d'enquestes. S'ha creat un projecte per a cadascun d'ells i s'ha obtingut una aplicació compactada i a punt per a ser desplegada en el servidor J2EE d'aplicacions.

4.4.1. Integració dels components a l'aplicació

Per a integrar a l'aplicació un component dels implementats com a ampliació de MyFaces, cal seguir les següents instruccions:

1. Crear el projecte de l'aplicació amb suport a MyFaces. Importar les llibreries de MyFaces a /WEB-INF/lib si és necessari (depèn del servidor J2EE d'aplicacions utilitzat).
2. Importar a /WEB-INF/lib el component ja compilat i compactat, és a dir, en format .jar.
3. Incloure les referències oportunes en el codi dels JSPs on s'hagi d'utilitzar.

```
<%@ taglib uri="http://components/llistaDoble" prefix="nou" %>
<html>
<head>
   <%@ include file="includes/head.inc"%>
</head>
<body>
   <f:view>
   <h:form>
   ...
         <nou:dobleLlista id="preg"
                          size="10"
                          width="250"
                          llistaOrigen="#{dadesBD.llistaPreguntes}"
                          llistaDesti="#{dadesBD.llistaPreguntesMarcades}"
                          value="#{dadesBD.preguntesMarcades}">
         </nou:dobleLlista>
   ...
   </f:view>
</body>
</html>
```

Figura 4.15:Codi exemple d'utilització d'un component en un document JSP

A continuació es mostra una relació dels mòduls on s'utilitzen cadascun dels components dissenyats:

Component	Mòdul
LlistaDoble	Gestió de la informació
TaulaDades[1]	Gestió de la informació
Opcions	Realització d'enquestes

Taula 4.1:Distribució dels components a l'aplicació de mostra

4.4.2. Entorns de desenvolupament, proves i execució

Els entorns de desenvolupament i proves són el mateix. Es caracteritzen per:

- En un sol equip físic hi ha el client (navegador web), el servidor d'aplicacions J2EE (que inclou el servidor web) i el servidor de bases de dades.
- Configuració de l'equip:
  - Sistema operatiu: Windows XP professional
  - Entorn de desenvolupament: Exadel 4.1 sobre Eclipse 3.2
  - Suport compilació: Apache Ant 1.6.5
  - Servidor J2EE d'aplicacions: JBoss 4.0.4
  - Servidor web: Apache Tomcat 5.5.17 (inclòs al servidor J2EE)
  - Servidor de bases de dades: Ms SQL server 2000

L'entorn d'execució o producció (entorn real) és diferent dels anteriors. Es caracteritza per:

- Ve donat per l'empresa per a qui es desenvolupa l'aplicació.
- Té un equip físic per a cada capa de l'aplicació:
  - Equips dels clients amb els navegadors web corresponents: Internet Explorer, Firefox... (el sistema operatiu no té cap rellevància).
  - Servidor J2EE d'aplicacions JBoss 4.0.4 amb l'Apache Tomcat 5.5.17 com a servidor web. El sistema operatiu és linux (distribució Debian).
  - Servidor de bases de dades Ms SQL server 2000. En aquest cas el SO és Windows 2000 Server.

4.5. Dificultats

Al llarg de la fase de desenvolupament de l'aplicació de mostra, també han sorgit nombrosos problemes, alguns dels quals ja s'han introduït al capítol anterior (estan directament relacionats amb l'ampliació del framework).

De nou, en un moment donat, ha sigut necessari redefinir algunes tasques per tal de complir els objectius generals del projecte.

Un altra vegada, la majoria d'aquests entrebancs han aparegut durant la implementació de l'aplicació. L'anàlisi i el disseny s'han portat a terme com estava previst.

[1] Quan estigui implementat.

4.5.1. Problemes i solucions adoptades

A continuació es descriuen els problemes més importants que s'han trobat al llarg d'aquesta etapa i que han provocat una major desviació en la planificació del projecte. També es comenten les solucions que s'han adoptat, tot i que algunes no hagin donat el resultat esperat.

1. Problema: En molts casos, els missatges d'error que mostra el sistema durant el desenvolupament d'una aplicació J2EE són poc explícits. Si no es té un bon rodatge, s'ha de dedicar força temps a trobar-lo i saber perquè s'ha produït. Tot això implica una pèrdua de temps considerable.

   Solucions: Es busquen errors similars a fòrums i llistes d'internet per si en algun cas diu a què és degut l'error en qüestió.

2. Context: S'utilitza Eclipse com a entorn de desenvolupament. La compilació i l'execució del codi es fan directament des d'aquest entorn.

   Problema: No sempre es reflecteixen a la web els canvis realitzats en el codi. Si un JSP n'inclou un altre i només es modifica el segon, el primer no el torna a carregar actualitzat.

   Solucions: En un primer moment es modifiquen sempre tots els documents afectats i es reinicia el servidor web (amb la conseqüent pèrdua de temps). En una segon fase, es crea un fitxer de compilació amb Ant i es genera cada vegada un nou fitxer desplegable (.jar) que es copia al servidor.

3. Context: Es comença a implementar el mòdul de Gestió de la informació per la capa de presentació.

   Problema: Al passar a desenvolupar les capes de lògica de negoci i d'accés a dades es veu que falta el servidor d'aplicacions JBoss. S'instal·la i es configura. Llavors es detecta que la capa de presentació no funciona.

   Solucions: Es canvien les llibreries que utilitza JBoss per defecte com a suport a JSF i s'afegeixen les pròpies de MyFaces al projecte en desenvolupament.

4. Context: La implementació del mòdul de Gestió de la informació parteix de la base de la pràctica tutoritzada publicada per la tutora. Es crea un servlet que gestiona les peticions dels usuaris.

   Problema: A l'afegir el servlet propi, els elements de la capa de presentació que es basen en MyFaces deixen de funcionar correctament (els camps dels formularis no validen, per exemple). Sembla com si el servlet implementat en el PFC impedeixi l'execució del que incorpora MyFaces (tot i que al fitxer de desplegament web.xml es carreguen tots dos).

   Solucions: Es busquen respostes a fòrums i llistes d'internet. Es consulta a la tutora. Es fan canvis diversos i proves de tot tipus. A hores d'ara, aquest problema encara no està resolt. El temps dedicat a intentar solucionar aquest tema torna a afectar de nou al bon desenvolupament del projecte. Per això es decideix implementar només algunes de les opcions de l'aplicació, en concret, aquelles que utilitzen els components implementats.

5. Seguiment i valoració econòmica

5.1. Seguiment

El desenvolupament d'aquest projecte ha patit una important variació respecte la planificació inicial, tant pel que fa a objectius definits com pel que fa a la temporització establerta (o millor dit, a la dedicació prevista). El principal motiu d'aquesta desviació ha estat el desconeixement inicial de la tecnologia a utilitzar.

La part més teòrica del projecte ha avançat i s'ha portat a terme segons la previsió inicial. El que s'ha hagut de modificar a causa de la manca de temps (tot i haver-hi dedicat més hores de les previstes) ha estat la part més pràctica.

Abans de començar aquest projecte es tenien coneixements de java, servlets i JSP, però no es coneixia ni s'havia treballat mai amb J2EE. Això ha suposat una feina extra de recerca i aprenentatge molt més important del previst. Lògicament, han anat sorgint els típics problemes causats per la manca de rodatge en el tema. És a dir, aquells entrebancs que apareixen al començar a treballar amb una nova tecnologia i un nou entorn de desenvolupament i això ha provocat haver-se de replantejar gran part de la vessant pràctica del projecte.

Inicialment es va preveure un calendari de treball de 20 hores setmanals. Arribats al mes de novembre i amb vistes a la part pràctica, aquest calendari es va ampliar a unes 30 hores setmanals.

A continuació es mostra un resum de la dedicació prevista, en comparació amb la real. Es pot veure clarament que la major diferència està en les tasques d'implementació.

TASCA	DURADA PREVISTA	DURADA REAL	DESVIACIÓ
Inici del PFC	0 h	0 h	
Familiarització J2EE i entorn desenvolupament			
Instal·lació i configuració			
J2EE	2 h	2 h	
Eclipse	2 h	2 h	
Tutorials			
J2EE	12 h	18 h	6 h
Eclipse	8 h	10 h	2 h
Elaboració pla de treball	16 h	16 h	
PAC1: pla de treball	0 h	0 h	
Estudi frameworks			
Cerca frameworks	2 h	4 h	2 h
Estudi detallat			
Framework 1	14 h	16 h	2 h
Framework 2	14 h	16 h	2 h
Framework 3	14 h	16 h	2 h
Redacció i conclusions estudi	4 h	8 h	4 h
PAC2: estudi frameworks	0h	0h	
Desenvolupament framework propi			
Anàlisi	12 h	12 h	
Disseny	12 h	16 h	4 h
Implementació	40 h	78 h	38 h
Familiarització Exadel		6 h	6 h
Exemples JSF i MyFaces		18 h	18 h

Proves	8 h	12 h	4 h
Desenvolupament aplicació			
Anàlisi	12 h	14 h	2 h
Disseny	12 h	16 h	4 h
Implementació	32 h	60 h	28 h
Proves	8 h	8 h	
Implantació en real	4 h	0 h	-4 h
PAC3: lliurament software	0 h	0 h	
Documentació memòria PFC	20 h	30 h	10 h
Preparació presentació virtual	8 h	8 h	
Lliurament memòria i presentació	0 h	0 h	
TOTAL	**256 h**	**386 h**	**130 h**

Taula 5.1:Temporització prevista vs temporització real

En la taula anterior també es pot apreciar que hi ha dues tasques noves respecte la planificació inicial: familiarització Exadel i Exemples JSF i MyFaces. Tot just iniciar la implementació dels nous components de MyFaces, es va veure la necessitat de buscar un entorn de desenvolupament més amigable (com és el cas d'Exadel) que facilités la primera presa de contacte. També es va creure oportú fer alguns tutorials i exemples de JSF i MyFaces per aclarir conceptes i veure les coses més clares.

5.2. Valoració econòmica

La valoració econòmica d'un projecte d'aquest tipus pot incloure diversos factors:

- Cost de llicències de programari.
- Cost del sistema físic on es posarà en producció l'aplicació.
- Cost de desenvolupament del projecte.

En aquest cas particular, cal tenir en compte alguns aspectes per entendre els càlculs que es presenten més endavant:

- Tot el programari utilitzat per a la implementació pràctica és programari lliure, tant el que es fa servir per al desenvolupament com el que es necessita en l'entorn real d'explotació. Així doncs, el cost d'aquesta partida és zero.
- El sistema físic on es desplegarà l'aplicació ja existeix i compleix els requeriments per a poder-la suportar. Per tant, no cal fer una inversió extra i per aquest motiu també es considera zero aquesta partida.
- El cost de desenvolupament del projecte es calcula en número d'hores de dedicació, indicades a l'apartat anterior. Es poden distingir dos rols (analista i programador) ja que el preu per hora en el mercat de cadascun d'ells és diferent.
- Dins del cost de desenvolupament del projecte s'inclou tota la feina feta. No es té en compte només la implementació de l'aplicació de mostra perquè es considera que, tot i que només sigui aquesta la part que veu el client, tota la resta també és necessària. En el supòsit que aquest fos un projecte de tants que s'han de portar a terme, els costos de l'estudi dels frameworks, per exemple, s'hauria de repartir entre tots ells, ja que és una tasca que no es repetiria per a cada cas.
- No entra dins de l'àmbit d'aquest projecte fer un pressupost econòmic excessivament detallat que tingui present aspectes com les despeses fixes que intervenen en un projecte informàtic (electricitat, desgast d'ordinador...) o d'altres que tampoc no venen al cas.
- Tampoc no es pretén en cap cas saber a quin preu es pot vendre l'aplicació de mostra (fent un estudi de mercat, per exemple).

A continuació es mostra un quadre que resumeix el càlcul de la valoració econòmica del projecte. El preu/hora considerat és simplement orientatiu.

Concepte	Recursos dedicats	Cost recursos	Import
Llicències de programari			0 €
Sistema físic d'explotació			0 €
Desenvolupament del projecte	386 hores	40 €/hora	15.440 €
TOTAL			15.440 €

Taula 5.2:Valoració econòmica

6. Conclusions

Després de desenvolupar aquest projecte, es poden extraure conclusions des de diferents punts de vista:

- Assoliment d'objectius.
- Tecnologia utilitzada.
- Impressions personals.

En termes generals, els objectius definits a l'inici d'aquest PFC s'han assolit, tal i com es resumeix a continuació:

- S'han ampliat els coneixements sobre la plataforma J2EE i les tecnologies que la formen. Especialment a nivell teòric, s'ha fet un gran estudi i aprofundiment.
  - S'ha fet l'estudi dels frameworks orientats a aplicacions web de la capa de presentació existents actualment.
    - S'ha cercat i recopilat informació sobre el tema en els mitjans i formats disponibles (llibres, manuals, tutorials, articles...). De tota ella, s'han escollit les fonts que han semblat més interessants i que han resultat ser força útils.
    - S'ha treballat la informació recollida (llegit, analitzat, comprès, instal·lat, provat...).
    - S'ha sintetitzat tota aquesta informació i se n'han extret conclusions.
    - S'ha pogut escollir un framework per a treballar-hi amb més profunditat en base a l'estudi realitzat.
    - S'ha documentat amb detall aquest estudi.
  - S'han dissenyat i implementat noves funcionalitats per a un marc de treball de la capa de presentació: MyFaces.
    - S'ha fet l'anàlisi de MyFaces (a nivell d'estructura i funcionament) i se li han detectat algunes mancances pel que fa a funcionalitats.
    - S'ha ampliat aquest framework dissenyant nous components i implementant-los, tot i que aquesta última fase no ha donat els resultats esperats.
  - S'ha desenvolupat una aplicació que mostra com s'utilitzen les noves funcionalitats afegides al framework.
    - S'ha fet l'anàlisi i disseny d'una aplicació amb J2EE amb accés a bases de dades que utilitza el framework ampliat.
    - S'ha implementat part de l'aplicació, de manera que es pot veure com integrar els nous components creats.
  - S'ha complert el calendari previst per al projecte (s'han fet els lliuraments d'acord amb les fites preestablertes), tot i que s'han hagut d'augmentar les hores de dedicació i deixar en fase de construcció algunes opcions de l'aplicació.

Pel que fa a la plataforma utilitzada, J2EE, i totes les tecnologies que aquesta ofereix, es poden treure algunes conclusions:

- J2EE, com el seu nom indica, està pensat per a aplicacions empresarials. Això es fa evident després de veure les dificultats trobades durant la fase d'implementació tant dels components de MyFaces com de l'aplicació de mostra. No és eficient resoldre un petit problema amb una plataforma d'aquesta complexitat.

- J2EE incorpora moltes i diverses tecnologies, fet que permet adaptar-se a nombrosos tipus de projectes, aprofitar molt millor la seva potència, però que també requereix un major esforç d'aprenentatge inicial.
- En el mercat hi ha moltes eines de desenvolupament (IDE) que donen suport a J2EE en general, i a alguns frameworks en particular. L'entorn més amigable, la configuració més senzilla, la possibilitat de disposar d'elements diversos que simplifiquen una tasca o altra (compilació, disseny de formularis...) faciliten el procés de desenvolupament als programadors. Actualment hi ha una gran oferta d'aquest tipus d'entorns en codi lliure.
- La utilització dels frameworks (que suposen una capa d'abstracció entre el sistema i els desenvolupadors d'aplicacions) realment faciliten la feina dels desenvolupadors, tot i que inicialment hi hagi un període d'adaptació a la nova manera de treballar gens menyspreable.
- Actualment es poden trobar nombrosos frameworks de codi lliure per a les diferents capes d'una aplicació J2EE. Això fa que hi hagi moltes persones al darrera que col·laboren en el seu desenvolupament, testeig... afavorint així la seva evolució i millora de la qualitat. Pel mateix motiu, també es poden afegir noves funcionalitats, personalitzar les existents o reutilitzar-ne d'altres per a una millor integració en el projecte que s'estigui treballant.

A nivell personal, un cop arribats a aquest punt, es destaquen alguns aspectes:

- És molt difícil poder fer una bona planificació d'un projecte que s'ha de desenvolupar amb una tecnologia que no es coneix. El marge d'error és molt alt.
- La corba d'aprenentatge per aconseguir un nivell acceptable en la utilització de les diferents tecnologies de J2EE (frameworks inclosos) és realment gran. Hi ha una gran quantitat de detalls que compliquen la familiarització amb l'entorn de desenvolupament i la comprensió de la seva filosofia de treball.
- Finalment, dir que fer un projecte com aquest, amb una vessant pràctica tan important, en un període de temps tant limitat (4 mesos aproximadament) i sense tenir per la mà la tecnologia a utilitzar, no és la millor opció.

7. Línies de futur

Les línies de futur d'aquest projecte passen, la gran majoria, per aprofundir en la part pràctica de J2EE, és a dir, en la implementació. Degut a la manca d'experiència amb J2EE i la temporització tan ajustada del projecte, no s'han pogut portar a la pràctica tots aquells coneixements que s'haurien volgut.

Per aquest motiu es té previst:

- Buscar solucions o alternatives als problemes trobats i encara no resolts de la implementació (component *LlistaDoble* i combinació de servlets propi amb MyFaces).
- Implementar el component *TaulaDades* per a poder-lo incorporar a l'aplicació.
- Acabar d'implementar les opcions que falten de l'aplicació "Enquestes de satisfacció de la docència".
- Desenvolupar el mòdul d'estadístiques de la mateixa aplicació.
- Desplegar l'aplicació de mostra en l'entorn real durant aquest curs 2006-2007, de manera que es pugui fer una prova pilot al final del 2n semestre.
- Millorar especialment la capa d'accés a dades, fent servir algun framework existent (com pot ser Hibernate).

Amb totes aquestes propostes es vol aconseguir adquirir un bon nivell de desenvolupament d'aplicacions sobre la plataforma J2EE, cosa que no ha estat possible durant aquest projecte per tenir un calendari tan ajustat (4 mesos encara no).

8. Glossari

- **AOP** (Aspect-Oriented Programming): paradigma de programació orientada a aspectes que pretén aconseguir una modularització apropiada i una millor separació de conceptes, que disminueixi o elimini les dependències inherents entre diferents mòduls.
- **API** (Application Programming Interface): conjunt d'especificacions de comunicació entre components de programari que proporciona abstracció entre els nivells inferiors i superiors d'una aplicació.
- **BD** (Base de Dades): és un conjunt de dades relacionades entre sí i que pertanyen al mateix context. S'organitzen d'una manera determinada per a poder ser consultades i modificades amb més comoditat.
- **EJB** (Enterprise Java Bean): API per a J2EE implementada per Sun Microsystems que ofereix un model de components del costat del servidor per a desenvolupar la lògica de negoci en aplicacions distribuïdes. Proporciona una capa d'abstracció per sobre de problemes com la concurrència, les transaccions, la seguretat...
- **Framework**: marc o entorn de treball orientat a la reutilització de codi per tal d'agilitzar el desenvolupament d'aplicacions. Extensió d'un llenguatge per mitjà de jerarquies de classes que implementen funcionalitats que també poden ser exteses.
- **IDE** (Integrated Development Environment): entorn integrat de desenvolupament que incorpora diferents eines (editor, compilador....) que faciliten la tasca del programador.
- **IoC** (Inversion of Control): principi de la programació orientada a objectes pel qual es redueix l'acoblament inherent en els programes informàtics. El control es passa de l'aplicació al framework.
- **J2EE** (Java 2 Enterprise Edition): estàndard basat en Java orientat al desenvolupament d'aplicacions empresarials distribuïdes, creat per Sun Microsystems.
- **JDBC** (Java DataBase Connectivity): API de Java que indica al client com s'ha de connectar a una base de dades relacional.
- **JFS** (Java Server Faces): framework basat en Java per al desenvolupament d'aplicacions web que simplifica el desenvolupament de les interfícies d'usuari. Estàndard desenvolupat per Sun Microsystems.
- **JSP** (Java Server Pages): tecnologia basada en Java que permet generar pàgines web de forma dinàmica en el servidor.
- **MVC** (Model-View-Controller): patró de disseny de programari que pretén separar clarament la lògica de control –Controller- (sap què s'ha de fer però no com), la lògica de negoci –Model- (sap com s'han de fer les coses) i la lògica de presentació –View- (sap com interactuar amb l'usuari).
- **MyFaces:** framework web de la capa de presentació basat en Java que implementa l'especificació JSF.
- **OOP** (Object-Oriented Programming): paradigma de programació que defineix els programes en termes de classes d'objectes i en ells encapsula estat (dades), comportament (mètodes i prodeciments) i identitat.
- **POJO** (Plain Old Java Object): classe definida en Java que no és de cap tipus especial (EJBs), ni implementa cap interfície específica.
- **Tag**: Etiqueta o marca d'un llenguatge basat en XML (HTML per exemple) que defineix un element i permet associar-li propietats i atributs que configuren el seu comportament.

- **TLD** (Tag Library Descriptor): document XML que defineix els tags d'un UIComponent. És un descriptor que mapeja les marques utilitzades en fitxers JSP a les classes corresponents que manipulen les seves propietats i atributs.
- **UIComponent** (User Interface Component): component o control que forma part de la interfície d'usuari d'una aplicació.
- **XML** (eXtensible Markup Language): metallenguatge extensible que permet definir la gramàtica de llenguatges específics. Estàndard per a l'intercanvi d'informació estructurada entre diferents plataformes.

9. Bibliografia

A causa de la gran tasca de recerca que s'ha hagut de fer al llarg de tot el projecte, la bibliografia és força extensa. A continuació es citen les referències més transcendents i freqüentades. Tot i així, el volum és gran, de manera que es destaquen (amb un asterisc vermell) aquelles que es creuen més útils i interessants en l'entorn treballat.

9.1. Llibres

- ALLAMARAJU, Subrahmanyam... *Profesional Programación Java Server con J2EE Edición 1.3.* Wrox. Ediciones Anaya Multimedia, 2005.

  ISBN 8441513589

- ALUR, Deepak... *Core J2EE Patterns. Best practices and design strategies.* 2ª ed. Sun Microsystems Press. Prentice Hall, 2003.

  ISBN 0131422464

- * BODOF, Stephanie... *The J2EE™ Tutorial*. 2a ed. Addison Wesley Professional, 2004.

  ISBN 0321245755

- DUDNEY, Bill... *Mastering JavaServer Faces*. John Wiley & Sons, 2004.

  ISBN: 0471462071

- * GEARY, David... *Core JavaServer Faces*. Sun Microsystem Press Java Series. Prentice Hall, 2004.

  ISBN: 0131463055

- GOODWILL, James... *Professional Jakarta Struts*. Wrox Press, 2003.

  ISBN: 0764544373

- JOHNSON, Rod. *Expert One-on-One J2EE design and development*. Wrox Press, 2002.

  ISBN: 0764543857

- JOHNSON, Rod. *Professional Java Development with the Spring Framework*. Wrox Press, 2005.

  ISBN: 0764574833

- JOHNSON, Mark... *Designing Enterprise Applications with the J2EE Platform*. 2ª ed. Addison-Wesley, 2002.

  ISBN: 0201787903

- KEOGH, Jim. *J2EE: Manual de referencia*. McGraw-Hill, 2003.

  ISBN: 8448139801

- * MANN, Kito. *JavaServer Faces in action*. Manning Publications, 2005.

  ISBN: 1932394125

- WALLS, Craig... *Spring in action*. Manning Publications, 2005.

  ISBN: 1932394354

9.2. Documents electrònics

- ÁLVAREZ RODRÍGUEZ, José María. *Desarrollo ágil en J2EE con herramientas OpenSource*.
  Adreça URL: http://www.asturlinux.org/jornadas2006/opensource-j2ee.php
- * ARMSTRONG, Eric. *The J2EE 1.4 Tutorial. For Sun Java System Application Server Platform Edition 8.2*. California, 2006.
  Adreça URL: http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html
- GONZÁLEZ, Cristóbal. *Mi primera web Struts*.
  Adreça URL: http://www.adictosaltrabajo.com/tutoriales/pdfs/miPrimeraWebStruts.pdf
- GOUJIE, Jackwind Li. UI Development with JavaServer Faces.
  Adreça URL: https://www6.software.ibm.com/developerworks/education/j-jsf/j-jsf-a4.pdf
- GUTIÉRREZ, Javier J. *¿Qué es un framework web?*
  Adreça URL: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- JOHNSON, Rod... *Spring. Java/J2EE application framework*.
  Adreça URL: http://static.springframework.org/spring/docs/2.0.x/spring-reference.pdf
- LABRA GAYO, Jose Emilio. Tecnologías Web.
  Adreça URL: http://www.di.uniovi.es/~labra/cursos/Doc05UPSAM/PPS/Doc05Arquitecturas.pps
- PÉREZ GARCÍA, Alejandro. *JSF – Java Server Faces (y su comparación con Struts)*.
  Adreça URL: http://www.adictosaltrabajo.com/tutoriales/pdfs/jsf.pdf
- RAIBLE, Matt. *Comparing Web Frameworks.*
  Adreça URL: https://equinox.dev.java.net/framework-comparison/WebFrameworks.pdf
- ROS, Thornton. *Creating Container-managed Entity Beans with JBoss.*
  Adreça URL: http://www.devx.com/Java/Article/17663
- ROSADO, Fernando. *Spring framework*.
  Adreça URL: http://www.mygnet.com/zip/man/3f7c1c0e2785a9ac93377d3e9462db3f.zip
- WOLSKI, Sascha. *First Java Server Faces Tutorial*.
  Adreça URL: http://www.lalilluna.de/assets/tutorials/first-java-server-faces-tutorial-en.pdf

9.3. Llocs web

- Akadia AG (última consulta: desembre 2006). *Connecting to SQL Server 2000 from Unix/Windows using JDBC*
  Adreça URL: http://www.akadia.com/services/sqlsrv_jdbc.html
- * Apache Software Fundation (última consulta: gener 2007). *MyFaces - Wellcome to the Apache MyFaces Project.*
  Adreça URL: http://myfaces.apache.org/
- Autentia (última consulta: desembre 2006). *Formación y desarrollo (Java, J2EE, UML, XML, etc.). Tutoriales sobre nuevas tecnologías.*
  Adreça URL: http://www.adictosaltrabajo.com/index.php?pagina=home
- CSIC (última consulta: octubre 2006). *Área de TIC. CSIC*.
  Adreça URL: http://tic.csic.es/documentos.htm
- Exadel (última consulta: novembre 2006). *Exadel – ExadelStudio.*
  Adreça URL: http://www.exadel.com/web/portal/products/ExadelStudio
- * Exadel (última consulta: desembre 2006). *Exadel – Tutorials*.
  Adreça URL: http://www.exadel.com/web/portal/products/Tutorials

- Exadel (última consulta: desembre 2006). *Exadel Studio 4.0 Guide and Features*.
  Adreça URL: http://www.exadel.com/exadelstudio/help/index.html
- * IBM (última consulta: gener 2007). JSF for nonbelievers. JSF component development.
  Adreça URL: http://www-128.ibm.com/developerworks/java/library/j-jsf4/
- Irian (última consulta: novembre 2006). *My Faces – the free JSF implementation.*
  Adreça URL: http://www.irian.at/myfaces/home.jsf f
- Java Enterprise Community (última consulta: octubre 2006). *Equinox: frameworks-comparison.*
  Adreça URL: https://equinox.dev.java.net/framework-comparison/
- javapassion.com (última consulta: octubre 2006). *Advanced J2EE Programming Class Syllabus*.
  Adreça URL: http://www.javapassion.com/j2eeadvanced/index0.html
- Java-Source.net (última consulta: octubre 2006). *Open Source Web Frameworks in Java*.
  Adreça URL: http://java-source.net/open-source/web-frameworks
- * Jax Systems LLC (última consulta: gener 2007). *Docjar. All packages from org.apache.myfaces.*
  Adreça URL: http://www.docjar.com/docs/api/org/apache/myfaces/overview-summary.html
- JBoss, Inc (última consulta: desembre 2006). *The JBoss 4 Application Server Guide.*
  Adreça URL: http://docs.jboss.org/jbossas/jboss4guide/r4/html/index.html
- JSF-COMP (última consulta: novembre 2006). *Welcome to JSF-COMP*.
  Adreça URL: http://jsf-comp.sourceforge.net/
- jsfTutorials.net (última consulta: novembre 2006). JavaServer Faces (JSF) Tutorial Net.
  Adreça URL: http://www.jsftutorials.net/
- Lalilunа (última consulta: novembre 2006). *JavaServer Faces Tutorials*.
  Adreça URL: http://www.lalilunа.de/javaserver-faces-tutorials.html
- Marty Hall (última consulta: novembre 2006). *JSF Tutorial from Marty Hall and coreservlets.com.*
  Adreça URL: http://www.coreservlets.com/JSF-Tutorial/
- MyFaces (última consulta: novembre 2006). *MyFaces Wiki*.
  Adreça URL: http://wiki.apache.org/myfaces/
- MyFaces Team (última consulta: novembre 2006). MyFaces: homepage.
  Adreça URL: http://www.marinschek.com/myfaces/tiki/tiki-index.php
- MyFaces Team (última consulta: desembre 2006). *Hibernate and MyFaces*.
  Adreça URL: http://wiki.apache.org/myfaces/Hibernate_And_MyFaces
- OpenSymphony (última consulta: octubre 2006). *WebWork - WebWork.*
  Adreça URL: http://www.opensymphony.com/webwork/
- O'Reilly Media (última consulta: gener 2007). *OnJava.com – The independent source for Enterprise Java.*
  Adreça URL: http://www.onjava.com/topics/java/J2EE
- Programación en castellano, S.L. (última consulta: novembre 2006). *Java en castellano. Cursos. J2EE java jsp.*
  Adreça URL: http://www.programacion.com/java/tutoriales/J2EE/
- Red Hat Middleware, LLc (última consulta: desembre 2006). *JBoss.com – Wiki – JBossFaces*.
  Adreça URL: http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossFaces
- SourceForge (última consulta: novembre 2006). *Welcome to SourceForge.net*.
  Adreça URL: http://sourceforge.net/
- Springframework (última consulta: octubre 2006). *Springframework.org*.
  Adreça URL: http://www.springframework.org/

- Struts (última consulta: octubre 2006). *StrutsExtensions – Wiki Struts.*
  Adreça URL: http://wiki.apache.org/struts/StrutsExtensions
- * Sun Microsystems (última consulta: desembre 2006). *Creating and Using a Custom Render Kit*.
  Adreça URL: http://java.sun.com/javaee/javaserverfaces/reference/docs/customRenderKit.html
- Sun Microsystems (última consulta: gener 2007). *J2EE 1.4 Platform*.
  Adreça URL: http://java.sun.com/j2ee/1.4/index.jsp
- Sun Microsystems (última consulta: desembre 2006). *JDBC Overview*.
  Adreça URL: http://java.sun.com/products/jdbc/overview.html
- Sun Microsystems (última consulta: gener 2007). *Java Technology Forums*.
  Adreça URL: http://forum.java.sun.com/index.jspa
- Sun Microsystems (última consulta: gener 2007). *java.net: Creating JSF custom components*.
  Adreça URL: http://today.java.net/pub/a/today/2004/07/16/jsfcustom.html
- Virtua, Inc. (Última consulta: novembre 2006). *JSF Central – Your JavaServer Faces Community*.
  Adreça URL: http://www.jsfcentral.com/
- Wikipedia (Última consulta: gener 2007). *Wikipedia*.
  Adreça URL http://www.wikipedia.org/

Nota: veure també la bibliografia del capítol “Estudi de frameworks de la capa de presentació”.

Annexos

Annex A: Guia d'instal·lació i execució de l'aplicació

9.3.1. Configuració del servidor de BD

1. Definir la BD al servidor SQL server.

   S'aporta un script per a crear les taules necessàries, assignar els permisos oportuns i insertar algunes dades de mostra.

2. Baixar-se les llibreries necessàries de http://www.akadia.com/services/sqlsrv_jdbc.html
3. Configurar el servidor JBoss perquè accedeixi a la BD.

   S'adjunta el fitxer de configuració adaptat C:\jboss-4.0.4.GA\server\default\deploy\mssql-ds.xml.

9.3.2. Configuració i desplegament de les aplicacions

1. Crear un nou projecte per a cada mòdul de l'aplicació.
2. Importar tota l'estructura de directoris i fitxers corresponent a cada mòdul.
3. Baixar-se les llibreries de MyFaces necessàries:
   - commons-beanutils-1.7.0.jar
   - commons-codec-1.3.jar
   - commons-collections-3.1.jar
   - commons-digester-1.6.jar
   - commons-el-1.0.jar
   - commons-fileupload.jar
   - commons-lang-2.1.jar
   - commons-logging-1.0.4.jar
   - jstl-1.1.0.jar
   - myfaces-api-1.1.4.jar
   - myfaces-impl-1.1.4.jar
   - tomahawk-1.1.3.jar
4. Baixar-se les llibreries de JBDC per a Ms SQL server necessàries:
   - msbase.jar
   - mssqlserver.jar
   - msutil.jar
5. Posar totes les llibreries necessàries (incloses les dels components implementats) al directori /WEB-INF/lib del projecte.
6. Si es fa amb eclipse, a l'arrel de cada projecte hi ha un descriptor anomenat eclipse-build.xml que permet compilar i obtenir el desplegament de l'aplicació.

9.3.3. Execució de les aplicacions

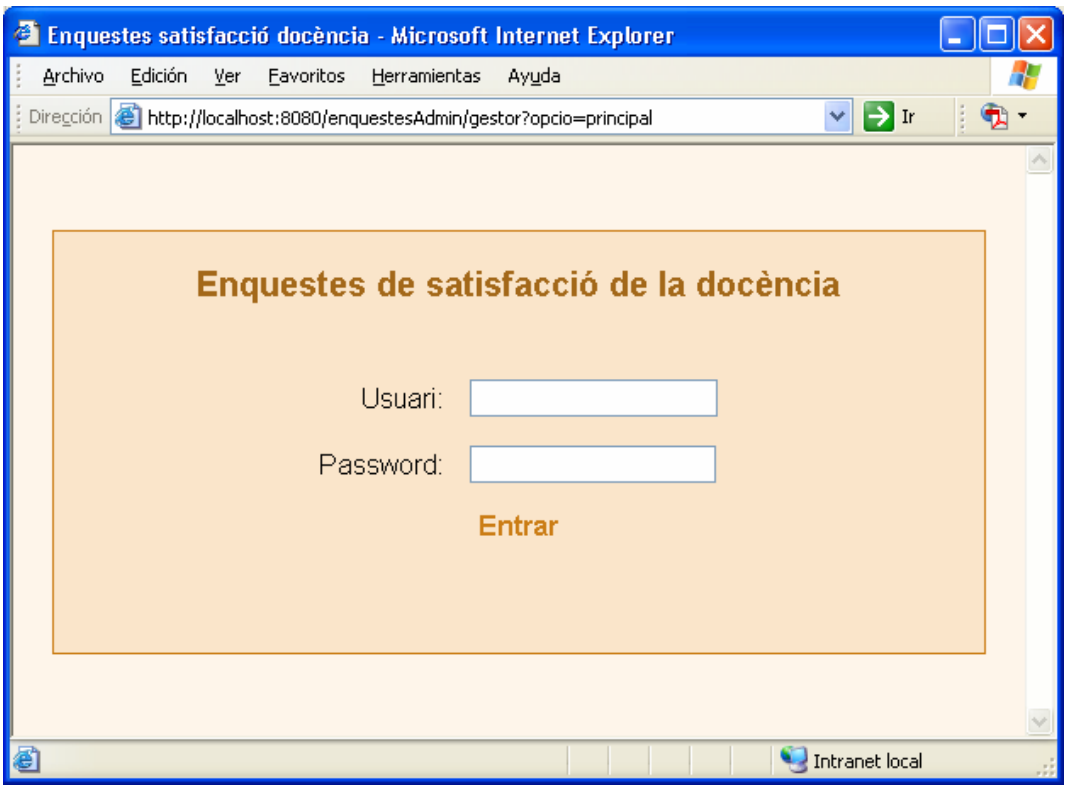
1. Copiar els fitxers enquestesAdmin.ear i enquestes.war al servidor d'aplicacions C:\jboss-4.0.4.GA\server\default\deploy

   Si es fa la compilació i desplegament a partir del descriptor, ja es copien els fitxers necessaris als llocs corresponents.

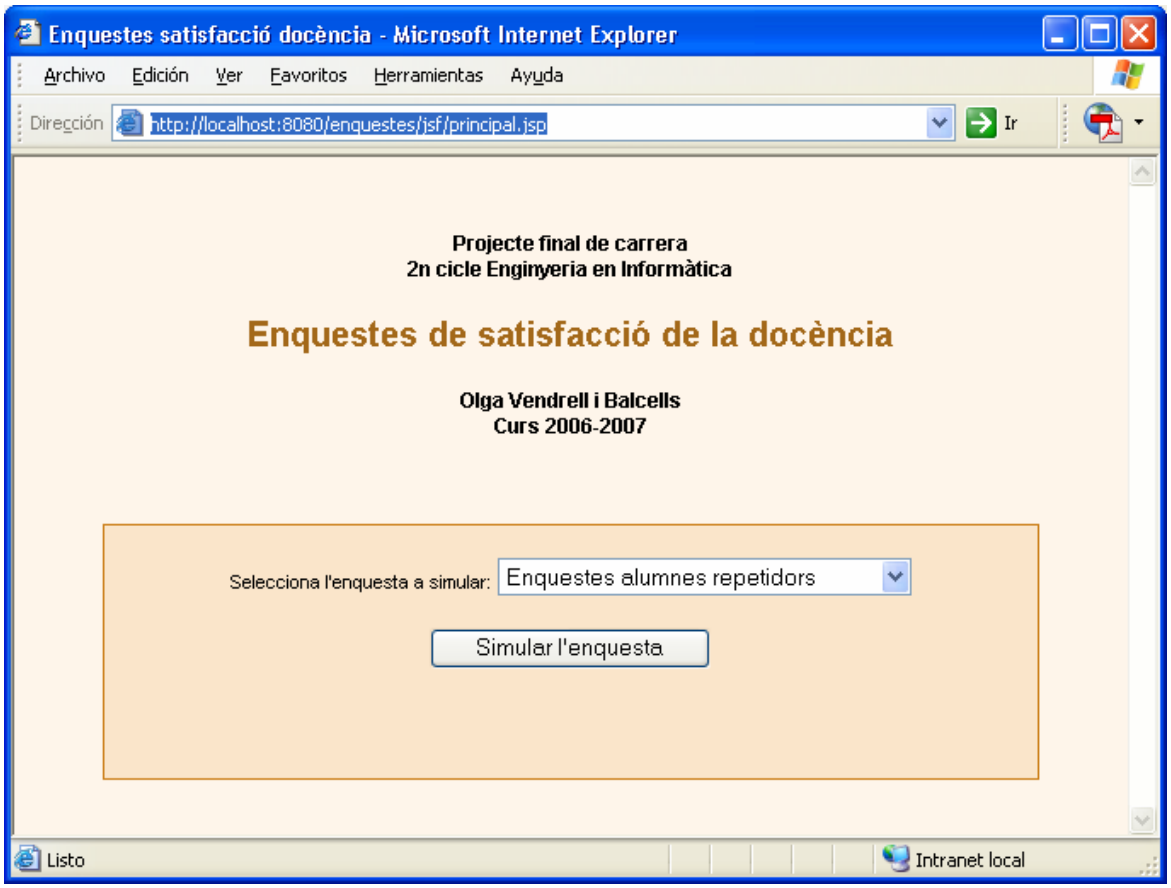
2. Iniciar el servidor JBoss.

3. Connectar-se a cadascun dels mòduls mitjançant les següents adreces:
   - http://localhost:8080/enquestesAdmin/gestor?opcio=principal

     En aquest cas demana identificació (Usuari: admin – Password: admin).



   - http://localhost:8080/enquestes/jsf/principal.jsp



Annex B: Model d'enquesta de satisfacció de la docència

El següent model és una mostra real d'una enquesta realitzada als alumnes per a avaluar la seva satisfacció respecte una assignatura en concret donada per un professor determinat.

1r semestre 2006-2007 **Avaluació part teòrica**

Fes una valoració de cada ítem del 0 al 10 (10 = totalment d'acord ; 0 = totalment en desacord).

1. Normalment arribo a classe amb puntualitat.	10 9 8 7 6 5 4 3 2 1 0
2. Aquest professor posa exemples a classe que ens motiven i ajuden a comprendre millor l'assignatura.	10 9 8 7 6 5 4 3 2 1 0
3. A nivell acadèmic, hi ha un bon clima de relació professor-estudiant.	10 9 8 7 6 5 4 3 2 1 0
4. Hi ha llibres, publicacions i materials de suport per fer un seguiment adequat d'aquesta assignatura.	10 9 8 7 6 5 4 3 2 1 0
5. Mantinc la meva atenció a les explicacions del professor al llarg de la classe.	10 9 8 7 6 5 4 3 2 1 0
6. En aquesta assignatura fan servir els mitjans auxiliars i complementaris (gràfics, audiovisuals, multimèdia...) adients com a suport.	10 9 8 7 6 5 4 3 2 1 0
7. Aquest professor explica de forma ordenada i amb claredat.	10 9 8 7 6 5 4 3 2 1 0
8. Porto fets els exercicis i problemes proposats pel professor.	10 9 8 7 6 5 4 3 2 1 0
9. En les classes d'aquest professor l'ambient general i d'atenció és bo.	10 9 8 7 6 5 4 3 2 1 0
10. Aquest professor porta la classe preparada.	10 9 8 7 6 5 4 3 2 1 0
11. Aquest professor té bona disponibilitat per atendre els seus alumnes.	10 9 8 7 6 5 4 3 2 1 0
12. Hi ha prou controls periòdics.	10 9 8 7 6 5 4 3 2 1 0
13. He assistit a totes les classes d'aquesta assignatura.	10 9 8 7 6 5 4 3 2 1 0
14. Les respostes del professor a les preguntes que li fem els estudiants sobre la matèria que explica són aclaridores.	10 9 8 7 6 5 4 3 2 1 0
15. El ritme d'aquesta assignatura és l'adequat.	10 9 8 7 6 5 4 3 2 1 0

GRÀCIES.

Annex C: Disseny físic de la base de dades

```
-- Professors
CREATE TABLE Professor
(
   -- Codi intern (identificador)
   idProfessor int NOT NULL,
   -- Nom
   nom nvarchar(100) NOT NULL,
   -- Primer cognom
   cognom1 nvarchar(100) NOT NULL,
   -- Segon cognom
   cognom2 nvarchar(100),
   --
   CONSTRAINT PK_Professor PRIMARY KEY (idProfessor)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Professor TO admin

-- Assignatures
CREATE TABLE Assignatura
(
   -- Codi intern (identificador)
   idAssignatura int NOT NULL,
   -- Nom complet
   nom nvarchar(100) NOT NULL,
   -- Abreviatura
   abreviatura nvarchar(5),
   -- Número de crèdits
   credits real,
   --
   CONSTRAINT PK_Assignatura PRIMARY KEY (idAssignatura)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Assignatura TO admin

-- Preguntes
CREATE TABLE Pregunta
(
   -- Codi intern (identificador)
   idPreg int NOT NULL,
   -- Enunciat de la pregunta
   enunciat nvarchar (3000) NOT NULL,
   --
   CONSTRAINT PK_Pregunta PRIMARY KEY (idPreg)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Pregunta TO admin

-- Models d'enquesta
CREATE TABLE Model
(
   -- Codi intern (identificador)
   idModel int NOT NULL,
   -- Descripcio
   descModel nvarchar(200) NOT NULL,
   -- Quantitat de preguntes
   numPreg int NOT NULL,
   --
   CONSTRAINT PK_Model PRIMARY KEY (idModel)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Model TO admin

-- Preguntes de cada enquesta
CREATE TABLE PregEnq
```

```
(
   idPregEnq int NOT NULL,
   -- Codi intern del model
   idModel int NOT NULL,
   -- Codi intern de la pregunta
   idPreg int NOT NULL,
   -- Ordre de la pregunta dins de l'enquesta
   ordre tinyint,
   --
   CONSTRAINT PK_PregEnq PRIMARY KEY (idPregEnq),
   CONSTRAINT UN1_PregEnq UNIQUE (idModel, idPreg),
   CONSTRAINT UN2_PregEnq UNIQUE (idModel, ordre),
   CONSTRAINT FK1_PregEnq FOREIGN KEY (IdModel) REFERENCES Model (idModel),
   CONSTRAINT FK2_PregEnq FOREIGN KEY (IdPreg) REFERENCES Pregunta (idPreg)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON PregEnq TO admin

-- Enquestes
CREATE TABLE Enquesta
(
   -- Codi intern (identificador)
   idEnquesta int NOT NULL,
   -- Breu descripció
   descEnq nvarchar(200) NOT NULL,
   -- Identificador del model d'enquesta
   idModel int NOT NULL,
   -- Codi de l'assignatura
   idAssignatura int,
   -- Codi del professor
   idProfessor int,
   -- Curs o any acadèmic
   curs int,
   --
   CONSTRAINT PK_Enquesta PRIMARY KEY (idEnquesta),
   CONSTRAINT FK1_Enquesta FOREIGN KEY (idModel) REFERENCES Model (idModel),
   CONSTRAINT FK2_Enquesta FOREIGN KEY (idAssignatura) REFERENCES Assignatura
(idAssignatura),
   CONSTRAINT FK3_Enquesta FOREIGN KEY (idProfessor) REFERENCES Professor
(idProfessor)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Enquesta TO admin

-- Resultats obtinguts a les enquestes
CREATE TABLE Resultat
(
   -- Codi intern (identificador)
   idResultat int NOT NULL,
   -- Codi intern de l'enquesta
   idEnquesta int NOT NULL,
   -- Codi intern de la pregunta
   idPreg int NOT NULL,
   -- Valor contestat a l'enquesta
   valor int,
   -- Comentaris respecte la pregunta
   comentari nvarchar(500),
   --
   CONSTRAINT PK_Resultat PRIMARY KEY (idResultat),
   CONSTRAINT FK1_Resultat FOREIGN KEY (idEnquesta) REFERENCES Enquesta
(idEnquesta),
   CONSTRAINT FK2_Resultat FOREIGN KEY (idPreg) REFERENCES Pregunta (idPreg)
)
GRANT INSERT, UPDATE, SELECT, DELETE ON Resultat TO admin
```