



PARLEM

Memòria de Projecte Final de Grau/Màster

Màster Universitari en Enginyeria Informàtica

Desenvolupament d'aplicacions web

Autor: JONATAN NUALART SÁNCHEZ

Consultor: Joan Giner Miguelez

Professor: César Pablo Córcoles Briongos

Data de lliurament



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Dedicatòria/Cita

Per tu Papa.

Abstract

Aquest projecte neix en virtut del que estableix la Llei Orgànica 3/2007, de 22 de març, per a la igualtat efectiva de dones i homes, i veien que al departament de prevenció de Riscos Laborals els hi van arribant notificacions per a la gestió de conflictes entre companys. El departament malgrat els intents no disposa d'eines ni recursos per fer front a aquests tipus de situacions entre companys, creant un Comitè de Seguretat i Salut [CSS].

El comitè de Seguretat i Salut [CSS] es gesta la creació d'una comissió de gestió de conflictes en l'entorn laboral, la CAM, un equip multidisciplinar que englobi protocols de Assetjament, agent d'igualtat i conflictes entre professionals.

El resultat d'una enquesta realitzada a l'empresa al juny del 2015, evidencia la percepció que suposa el risc dels conflicte relacionals on el 41% dels enquestats puntua negativament el recolzament en cas de agressions verbals, i s'aposta per un nou model de gestió del conflicte.

Paraules clau: Conflicte, Pau, Resolució, Parlem, Mediator, Memòria, Treball de Fi de Grau/Màster.

Abstract (english version)

This project is born under the provisions of Organic Law 3/2007, of March 22, for the effective equality of women and men, and they saw that in the Department of Prevention of Labor Risks they received notifications for the management of conflicts between colleagues. The department despite the attempts does not have tools or resources to deal with these types of situations among colleagues, creating a Health and Safety Committee [CSS].

The Health and Safety Committee [CSS] is developing the creation of a conflicts management committee in the work environment, the CAM, a multidisciplinary team that encompass Assessment protocols, an agent of equality and conflicts between professionals.

The result of a survey carried out in the company in June 2015 reveals the perception that the risk of relational conflicts represents, where 41% of the respondents negatively endorse support in case of verbal aggression, and strives for a new model of conflict management.

Keywords: Conflict, Peace, Resolution, Speech, Mediator, Memory, Final Project / Master's Degree.

ÍNDIX DE CONTINGUTS

1. INTRODUCCIÓ	8
1.1 CONTEXT I JUSTIFICACIÓ DEL TREBALL	8
1.2 - OBJECTIUS DEL TREBALL	9
1.2 .1- JUSTIFICACIÓ	9
1.3 - ENFOCAMENT I MÈTODE SEGUIT.	10
1.4 - PLANIFICACIÓ DEL PROJECTE	11
1.4.1 FITES I DATES RELLEVANTS:	11
1.4.2 DIAGRAMA DE GANNT	12
2. ANÀLISI (ACTORS, CASOS D'ÚS, DESCRIPCIÓ TEXTUAL CASOS D'ÚS)	13
2.1 ACTORS	13
2.2 CASOS D'ÚS	13
2.3 DESCRIPCIÓ TEXTUAL CASOS D'ÚS	14
3. PROTOTIPS	19
4. USABILITAT/UX	21
5. DISSENY (ARQUITECTURA, DISSENYSS BBDD...)	22
5.1- Arquitectura del Sistema	22
5.2. MVC i Frameworks per J2EE	24
5.3 STRUTS	28
5.4 Hibernate	30
5.3 Disseny de la base de dades	31
5.4 Diagrama de classes	31
6. IMPLEMENTACIÓ (REQUERIMENTS DEL SOFTWARE, ESTRUCTURA...)	32
6.1 - Requeriments del software	32
6.2 - Els requeriments del software d'aquest projecte són els següents:	32
6.3. Estructura del projecte	33
6.4. Decisions	34
6.4. Consideracions a l'hora d'instal·lar	35
6.4.1. - Instal·lació de Java (jdk)	35
6.4.2 - Servidor d'aplicacions web (WildFly)	36
6.4.3 - Servidor base de dades (MySQL)	36
6.4.4 – Executar l'aplicació	36
7. VALORACIÓ ECONÒMICA.	37
8. CONCLUSIONS	37
9. GLOSSARI	38

10. BIBLIOGRAFIA	39
11. ANNEXA	40

ÍNDIX DE FIGURES

<i>II·lustració 1: Calendari</i>	12
<i>II·lustració 2: Diagrama de GANNT</i>	12
<i>II·lustració 3: Casos D'ús</i>	13
<i>II·lustració 4: Pantalla login</i>	19
<i>II·lustració 5: Pantalla menú</i>	19
<i>II·lustració 6: Pantalla formulari</i>	20
<i>II·lustració 7: Pantalla cens treball</i>	20
<i>II·lustració 8: Pantalla Edició Notificació</i>	21
<i>II·lustració 9: Pantalla Edició Notificació seguiment</i>	21
<i>II·lustració 9: Arquitectura J2EE</i>	23
<i>II·lustració 10: MVC</i>	25
<i>II·lustració 11: FrameWorks</i>	27
<i>II·lustració 12: Entitat relació</i>	31
<i>II·lustració 13: Diagrama de classes</i>	31
<i>II·lustració 14 Estructura del projecte</i>	33
<i>II·lustració 15 Cens treball</i>	35
<i>II·lustració 16 Missatge Sessió Caducada</i>	35
<i>II·lustració 17 Missatge dades guardades correctament</i>	35

1. INTRODUCCIÓ

1.1 CONTEXT I JUSTIFICACIÓ DEL TREBALL

Aquest treball neix amb motiu de la finalització del Màster Universitari en Enginyeria Informàtica de l'àrea de treball de Desenvolupament d'aplicacions web.

L'interès d'aquest projecte ve determinat per motivacions tipus personal i professional envers la resolució de conflictes entre professionals.

L'objectiu principal del projecte és la realització d'una aplicació web i el nom escollit és PARLEM. Es tracta d'una eina per la resolució de conflictes entre professionals que està gestionada per la figura del facilitador.

El sistema de gestió de les discrepàncies i els conflictes recau sobre la figura del facilitador, que té les competències, les habilitats, les eines i els coneixements en gestió alternativa de conflictes que els permet prevenir, contenir i gestionar en positiu les discrepàncies i els conflictes entre professionals.

L'aplicació també satisfà la LOPD en el seu capítol IV, que preveu la creació de codis de conducta, que seran vinculants per qui vulgui adherir al mateix. A més anima a la creació i ús de mecanismes de resolució extraoficial de conflictes, que seran promoguts per les associacions i altres organismes representatius de categories de responsables o encarregats del tractament.

Les avantatges queden evidenciades tant per el professional afectat ja que té una forma senzilla d'intentar arreglar ràpidament i satisfactori mitjançant un acord consensuat. I pe l'empresa suposa deixar clar que està complint amb la RGPD de forma proactiva evitant l'obertura d'un expedient per part de l'autoritat amb una possible sanció deixant una mala imatge pública.

Molts conceptes estudiats al llarg dels estudis són utilitzats per al desenvolupament d'aquest treball. També ha estat necessari estudiar diferents frameworks per desenvolupar la tecnologia J2EE: Struts2, Tiles, Hibernate.

1.2 - OBJECTIUS DEL TREBALL

La tecnologia Java i l'arquitectura J2EE s'han convertit en un referent en el món de l'indústria pel desenvolupament distribuït d'aplicacions empresarials a Internet. A nivell personal, escollint com a temàtica J2EE pel TFM, m'agradaria profunditzar en aquesta matèria assolint més coneixements i així oferir més qualitat i professionalitat al món laboral.

- Aprofundir en el patró MVC (Model View Controller) per organitzar els fluxos de dades en les aplicacions, construint sistemes robustos, fàcils de mantenir i de créixer.
- Estudiar els FrameWorks i estàndards definits a l'arquitectura J2EE com ara JavaServer Pages, Servlets o Enterprise JavaBeans, Struts 2, JBoss, Hibernate. ...
- Dissenyar una interfície web pràctica alhora intuïtiva i perfilant els aspectes d'usabilitat.

1.2 .1- JUSTIFICACIÓ

Les raons per la qual he escollit Java com a llenguatge de desenvolupament pel projecte són les següents:

Java és un dels llenguatges de programació més utilitzat del món.

Java és multiplataforma, pots desenvolupar una aplicació que podrà funcionar en qualsevol plataforma ja sigui Windows, Mac o Linux, i a més no s'ha de pagar cap llicències ja que és completament gratuït utilitzar aquesta tecnologia.

La comunitat de Java té disponible un gran suport i sobretot documentació per continuar aprenent i resolent tots els dubtes i problemes que poden sorgir a l'hora de desenvolupar les aplicacions.

MySQL és un Sistema Gestor de Base de Dades (SGBD) relacional, multifil i multiusuari utilitzat per la majoria de les pàgines web actuals. A més es el més usat en aplicacions creades com software lliure.

S'ofereix sota la GNU GPL encara que també és possible adquirir una llicència per empreses que ho volguessin incorporar en productes privats (sobretot des de la compra per part d'ORACLE s'està orientant en aquest àmbit empresarial)

Les principals avantatges d'aquest sistema gestor de base de dades són:

- Facilitat d'ús i gran rendiment.
- Facilitat per la instal·lació i configuració.
- Suport multiplataforma.
- Suport SSL

Però com a desavantatge és la escalabilitat, es a dir no treballa de manera eficient amb base de dades molt grans que superen una determinada mida.

1.3 - ENFOCAMENT I MÈTODE SEGUIT.

Al llarg dels estudis realitzats a l'Enginyeria i el Màster he après dividir un projecte amb subsistemes facilitant molt la feina. Per aquest motiu he decidit dividir el projecte amb quatre subsistemes:

Subsistema Notificació conflicte:

Aquest subsistema serà l'eix central dels treballadors sent els encarregats d'enregistrar les dades per notificar el conflicte. L'usuari serà qualsevol treballador del centre.

Subsistema llistat de treball:

Aquest subsistema serà l'encarregat de mostrar un cens amb els registres introduïts al subsistema notificació de conflictes. Existirà uns criteris de cerca que seran personalitzats per l'usuari.

Subsistema Seguiment del cas:

Aquest subsistema serà l'encarregat de fer un seguiment del registre introduït en el subsistema notificació conflicte. Aquest subsistema ampliarà la informació ja introduïda en el primer subsistema L'usuari recau en la figura del facilitador.

Subsistema Connexió:

Quan un usuari vulgui accedir a l'aplicació s'haurà d'autenticar mitjançant un nom d'usuari i una clau. Si no és correcte el sistema mostrarà un missatge indicant l'error. Si les dades d'identificació són correctes el sistema permetrà que l'usuari accedeixi a les dades. Existirà accessos restringits ja que hi haurà dos tipus de perfils: treballadors i facilitadors.

Subsistema traçabilitat

Serà l'encarregat de enregistrar tots els accessos a l'aplicació per part dels usuaris. L'objectiu és augmentar la protecció a les persones davant el tractament i circulació de les dades personals cobrint la normativa del nou reglament general de protecció de dades.

1.4 - PLANIFICACIÓ DEL PROJECTE

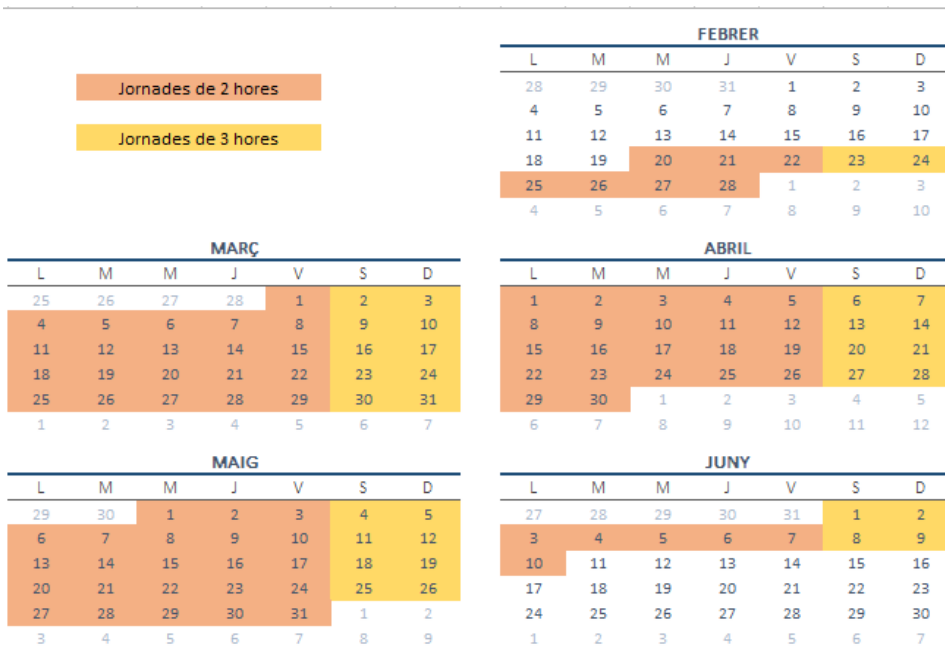
1.4.1 FITES I DATES RELLEVANTS:

El treball s'inicia el 20 de febrer i finalitza el 10 de Juny amb el lliurament de la memòria

Les dates claus són:

- Elaboració Pla de treball. (PAC1) 20/02/2019 - 05/03/2019
- Primera fase de l'execució del pla de treball. (PAC2) 06/03/2019 - 03/04/2019
- Segona fase de l'execució del pla de treball (PAC3) 04/04/2019 - 12/05/2019
- Lliurament memòria 13/05/2019 – 10/06/2019
- Defensa Virtual – 17/06/2019 - 21/06/2019

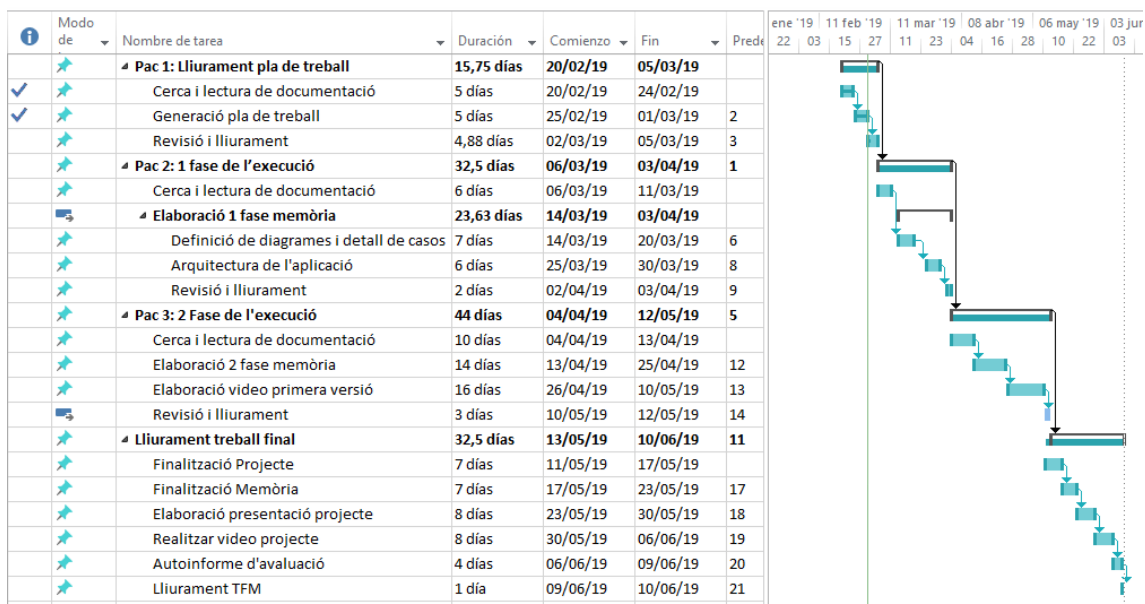
Es preveu un dedicació de 254 hores distribuïdes de la següent manera:



Il·lustració 1: Calendari

1.4.2 DIAGRAMA DE GANNT

L'eina gràfica de diagrama de GANNT es mostra la relació de fites i les seves tasques així com la relació de dependència entre elles. GANNT ens mostra una visió de la planificació general i la planificació de cada tasca:



Il·lustració 2: Diagrama de GANNT

2. ANÀLISI (ACTORS, CASOS D'ÚS, DESCRIPCIÓ TEXTUAL CASOS D'ÚS)

2.1 ACTORS

Existeixen tres tipus d'actors:

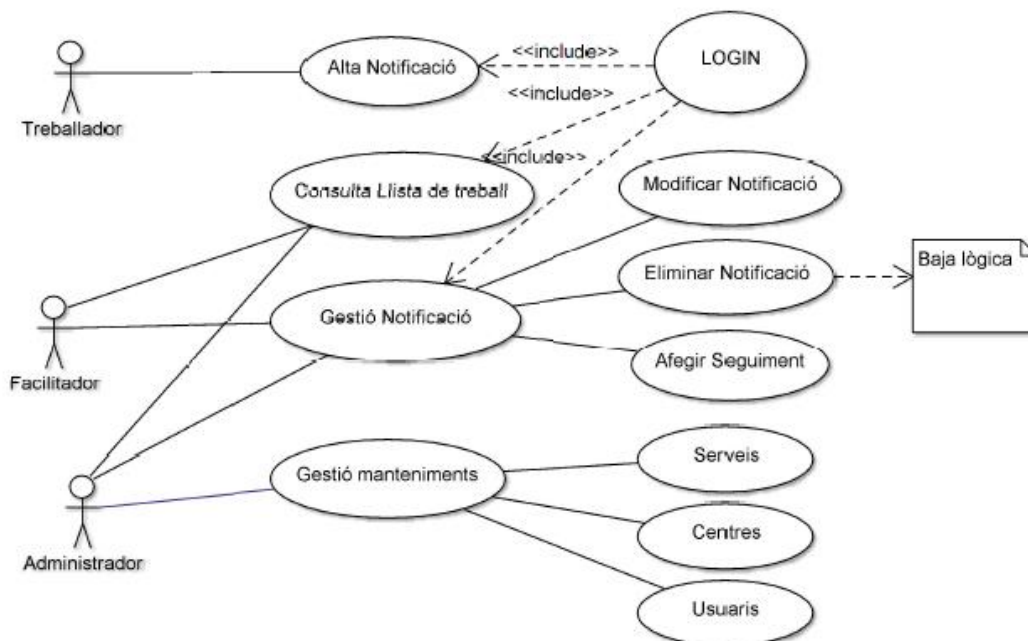
Facilitador: Aquest usuari podrà accedir a crear, modificar i esborrar les notificacions. Aquest usuari estarà enregistrat a la base de dades i per accedir-hi s'haurà d'identificar.

Administrador: Tindrà els mateixos permisos que el facilitador però a més tindrà accés a la gestió dels manteniments.

Treballador: S'identificarà per accedir a l'aplicació i podrà fer una notificació.

2.2 CASOS D'ÚS

El següent diagrama es pot visualitzar la funcionalitat del sistema, cada actor té les seves funcions i depenent del rol cadascú podrà realitzar diferents tasques o accés a diferents mòduls del sistema.



Il·lustració 3: Casos D'ús.

2.3 DESCRIPCIÓ TEXTUAL CASOS D'ÚS

CAS D'ÚS: LOGIN USUARI	
Resum de la funcionalitat	Valida si l'usuari està registra
Actors	Treballador, facilitador
Casos d'ús relacionats	Cap
Precondició	L'usuari es troba registrat a l'aplicació
Postcondició	L'usuari accedeix a l'aplicació
Cas	L'usuari introdueix el seu nom usuari i la clau. Si és correcte accedeix a l'aplicació.
Alternativa	L'usuari introduït no existeix a la base de dades o la clau no és correcte, no accedeix a l'aplicació.

CAS D'ÚS: ALTA NOTIFICACIÓ	
Resum de la funcionalitat	Mostra una pantalla per omplir les dades de la notificació.
Actors	Treballador, facilitador, administrador
Casos d'ús relacionats	Login
Precondició	No existeix notificació.
Postcondició	Crea una notificació a la base de dades
Cas	El treballador introdueix les dades de la notificació
Alternativa	Error al introduir un registre i mostra error.

CAS D'ÚS: CONSULTA LLISTA DE TREBALL	
Resum de la funcionalitat	Mostra un llistat amb tots els pacients enregistrats.
Actors	Facilitador, administrador
Casos d'ús relacionats	Modificar, eliminar, afegir notificació.
Precondició	Existeixen notificacions a la BBDD

Postcondició	Es recuperen les notificacions existents a la base de dades seguint els criteris de selecció escollits.
Cas	El facilitador consulta segons un criteris la llista de notificacions. Des d'aquesta llista es podrà accedir a la gestió de la notificació.
Alternativa	No recuperar cap registre amb els criteris de selecció escollits.

CAS D'ÚS: GESTIÓ NOTIFICACIONS

Resum de la funcionalitat	Mostra un llistat amb tots els pacients enregistrats.
Actors	Facilitador
Casos d'ús relacionats	Modificar, eliminar, afegir notificació.
Precondició	L'usuari s'ha identificat com a Facilitador
Postcondició	El facilitador escull l'opció.
Cas	El facilitador depenent l'opció que escull accedeix.
Alternativa	

CAS D'ÚS: MODIFICACIÓ NOTIFICACIÓ

Resum de la funcionalitat	Mostra una pantalla per modificar les dades de la notificació
Actors	Facilitador, administrador
Casos d'ús relacionats	Gestió de notificacions
Precondició	Existeixen la notificació a la base de dades
Postcondició	Modifica les dades de la notificació.
Cas	El facilitador accedeix a la pantalla de la notificació i podrà modificar-la.
Alternativa	Es produeix un error al modificació les dades de la notificació i es mostrar un missatge.

CAS D'ÚS: ELIMINACIÓ NOTIFICACIÓ

Resum de la funcionalitat	Mostra una opció per eliminar la notificació
Actors	Facilitador, administrador
Casos d'ús relacionats	Gestió de notificacions
Precondició	Existeixen la notificació a la base de dades
Postcondició	S'esborra de manera lògica la notificació del treballador
Cas	El facilitador mitjançant una opció podrà eliminar una notificació.
Alternativa	Es produeix un error a l'eliminar la notificació i es mostrar un missatge.

CAS D'ÚS: AFEGIR SEGUIMENT

Resum de la funcionalitat	Mostra una opció per afegir un seguiment
Actors	Facilitador, administrador
Casos d'ús relacionats	Gestió de notificacions
Precondició	Existeixen la notificació a la base de dades
Postcondició	S'afegeix un seguiment
Cas	El facilitador mitjançant una opció podrà afegir un seguiment.
Alternativa	Es produeix un error a afegir un seguiment i es mostrar un missatge.

CAS D'ÚS: GESTIÓ DE MANTENIMENTS

Resum de la funcionalitat	Menú principal per accedir a diferents manteniments
Actors	Administrador
Casos d'ús relacionats	Login
Precondició	L'usuari s'ha identificat com a administrador i ha accedit al menú gestió manteniments
Postcondició	L'administrador escull un manteniment

Cas	L'usuari després de prémer el botó manteniments accedeix a un llistat de manteniments .
Alternativa	

CAS D'ÚS: MANTENIMENT SERVEIS

Resum de la funcionalitat	L'administrador podrà crear, modificar i eliminar serveis.
Actors	Administrador
Casos d'ús relacionats	Gestió manteniments
Precondició	A seleccionat l'opció manteniment de servies
Postcondició	L'administrador ha creat, modificat o eliminat serveis
Cas	L'administrador podrà afegir, modificar o eliminar serveis
Alternativa	Si s'afegeix un servei existent mostrarà missatge indicant-ho.

CAS D'ÚS: MANTENIMENT USUARIS

Resum de la funcionalitat	L'administrador podrà crear, modificar i eliminar usuaris.
Actors	Administrador
Casos d'ús relacionats	Gestió manteniments
Precondició	A seleccionat l'opció manteniment d'usuaris
Postcondició	L'administrador ha creat, modificat o eliminat usuaris
Cas	L'administrador podrà afegir, modificar o eliminar usuaris
Alternativa	Si s'afegeix un usuari existent mostrarà missatge indicant-ho.

CAS D'ÚS: MANTENIMENT CENTRES

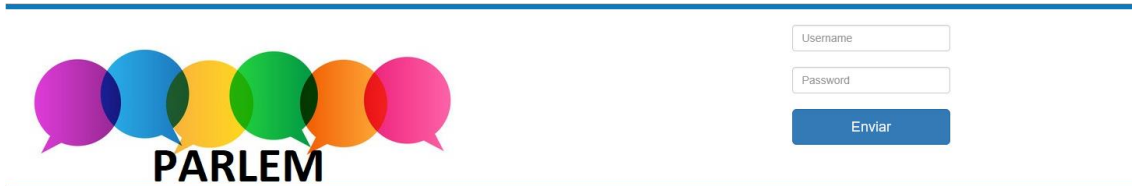
Resum de la funcionalitat	L'administrador podrà crear, modificar i eliminar centres.
Actors	Administrador

Casos d'ús relacionats	Gestió manteniments
Precondició	A seleccionat l'opció manteniment de centres
Postcondició	L'administrador ha creat, modificat o eliminat centres
Cas	L'administrador podrà afegir, modificar o eliminar centres
Alternativa	Si s'afegeix un centre existent mostrarà missatge indicant-ho.

3. PROTOTIPS

A continuació es presenten els prototips de les principals pantalles.

L'aplicació tindrà accés restringit, per tant a la primera pantalla s'haurà d'accedir mitjançant un usuari i una clau.



Il·lustració 4: Pantalla login

Un cop s'ha identificat l'usuari s'accedirà a l'aplicació i tindrà una sèrie d'opcions depenent el perfil de l'usuari.



PARLEM

Il·lustració 5: Pantalla menú

Si s'escull nou registre s'accedirà a un formulari per introduir les dades mínimes per poder fer un seguiment del cas:

Nou Registre Gestió Manteniment Sortir

Nova Notificació

Notificant
Notificant
Data acc/inc
Correu
@Correu
Centre
Centre 1
Servei
Servei 1
Horari/Torn
Torn 1
Telèfon
Telèfon

Descriu la situació:
Has parlat amb el teu comandament d'aquest cas?
SI No
Si és així, quina ha estat la seva actuació al respecte?
Quines expectatives tens de la nostra intervenció:
Les altres persones implicades saben que t'has adreçat a l'agora?
SI No
Altres informació d'interès:

Enviar

Il·lustració 6: Pantalla formulari

Des de la pantalla de menú si s'accedeix a la gestió apareix un cens amb tots els casos registres filtrats des de principi d'any fins la data actual.

Nou Registre Gestió Manteniment Sortir

Cerca

Des de 20-03-2019 Fins 02-04-2019 Cerca

Ref.M...	Nom	Data	Servei	Centre	Descripció Situació	Icon
28	✎
45	✎
18	✎

Rows per page: 10 1-3 of 3

Il·lustració 7: Pantalla cens treball

En la pantalla de l'edició de la notificació el facilitador podrà completar la notificació que ha fet l'usuari.

I

Nou Registre Gestió Manteniment ▾ Sortir

Notificació: 28
Jonatan Nualart

Telèfon: 627929752 Data acc/inc: 16/04/19 0:00:00.000 Correu: jnualart@hotmail.com Centre: parlem.centre Servei: Servei 6

Descriu la situació: M'he barallat amb tothom Has parlat amb el teu comandament daquest cas? Si Actuació: NO ha fewt res	global.comu.parlat.expectatives Quines expectatives tens de la nostra intervenció: Sluvsdf Les altres persones implicades saben que thas adreçat a làgora? Si	Altre informació d'interès ssdfg
---	---	--

Tipus Contacte:

Tipus Conflict:

Facilitador:

Estat:

Part 2

Nom:

Correu:

Telèfon:

Seguiment

Data acc/inc:

Seguiment:

[Submit](#)

Sequiments de la notificació: **28**

II-Il·lustració 8: Pantalla Edició Notificació

Aquest serà un seguiment evolutiu de la gestió / edició de la notificació.

Sequiments de la notificació: **28**

Ref.Id	Descripció Situació
17/04/2019	Primera prova de seguiment,
29/04/2019	prov. aparo vsf sadf sa
18/04/2019	prova control seguement
15/04/2019	prova seguimet 3
18/04/2019	preova defginaf
15/04/2019	un dia després de l'aniversari de Roger
14/04/2019	Aniversari de Roger
24/04/2019	prova seguiment 5
17/04/2019	prova 2

Page **1** of **1** **10**

Displaying records 1 - 9 of 9

II-Il·lustració 9: Pantalla Edició Notificació seguiment

4. USABILITAT/UX

La finalitat amb la Usabilitat és que un usuari mitjà pugui navegar i aconseguir el que desitja a l'aplicació.

Els aspectes més importants a tenir en compte per millor la usabilitat web de l'aplicació són:

Temps de càrrega: El temps de resposta ha de ser el mínim possible.

Disseny responsive: Ha d'estar adaptada a qualsevol tipus de dispositiu, PCs, mòbils o tableta.

UI SCROLL La informació imprescindible estarà a la part superior de la pàgina, sense que l'usuari hagi de desplaçar-se cap avall i fer scroll.

Mínim de clicks: Quan menys clicks hi hagi a la web millor. L'objectiu serà minimitzar l'ús de clicks.

Disseny de qualitat: El disseny és molt important i la qualitat donarà confiança a l'usuari.

Enllaços interns: Els enllaços a altres apartats de la pàgina fomentarà la navegació i la usabilitat.

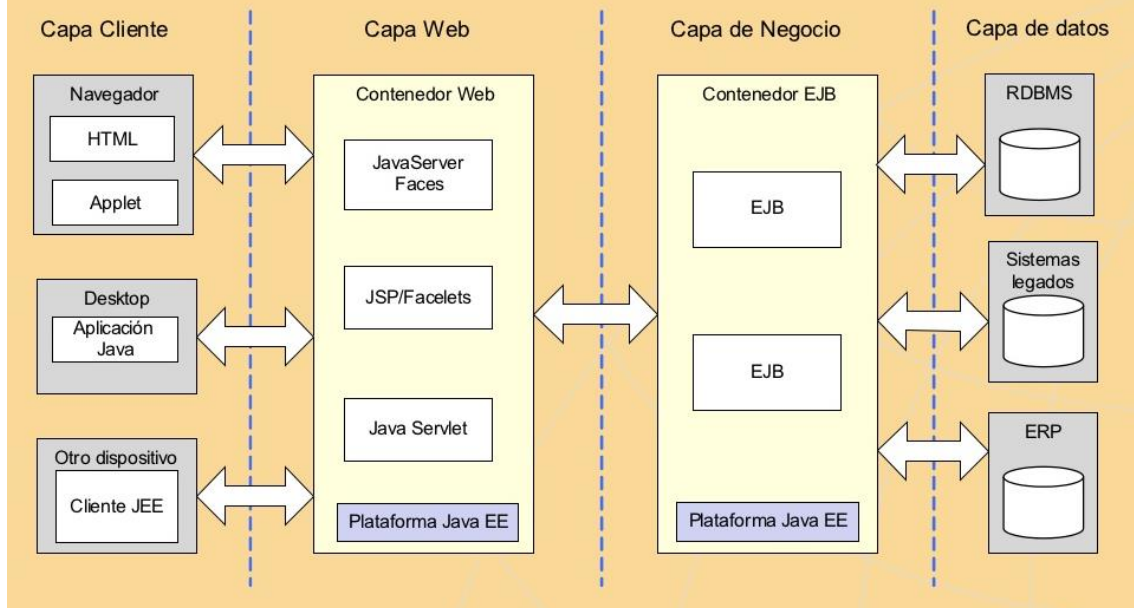
5. DISSENY (ARQUITECTURA, DISSENY BBDD...)

5.1- Arquitectura del Sistema

L'arquitectura del sistema és J2EE, aquesta ha estat escollida com a plataforma de programació. Java EE i està definida en un conjunt de serveis, API i protocols per al desenvolupament d'aplicacions de múltiples capes. Es recolza en components de software modulars.

La plataforma J2EE implica una forma d'implementar i desplegar aplicacions empresarials. Està basada amb el llenguatge JAVA, és oberta i està suportada per diferents plataformes que tinguin instal·lat el Java Virtual Machine. Amb això ens assegurem tenir portabilitat i robustesa. L'arquitectura J2EE implica un model d'aplicacions distribuïdes en diferents capes o nivells.

Arquitectura de Java EE



Il·lustració 9: Arquitectura J2EE.

Capa client: La funció bàsica d'aquesta capa és la interacció amb l'usuari i la presentació de les dades. Admet diferents tipus de clients (HTML, XHTML, Applet, aplicacions d'escriptoris Java GUI, etc.)

Capa web: Suporta la lògica de presentació Java EE. Totes les tecnologies (jsp, jsf, servlets etc...) que es desenvolupen en el contenidor van orientats al model de petició resposta (request-response).

Capa negoci: Suporta la lògica de negoci de les aplicacions, es representen tots els processos de l'empresa per poder reutilitzar els components creats a varies aplicacions i poder reutilitzar el codi. En aquesta capa/contenidor es troben dos tipus de components, els EJBs i els serveis web.

Capa de dades: Suporta les persistència de les dades, es pot entendre com un magatzem d'informació (EIS Enterprise Information System).

Cal destacar el concepte “components” que no és res més que un entorn d'execució estandarditzat que ofereix servies mitjançant components. El components externs tenen una forma estàndard d'accedir als serveis amb independència del fabricant.

Aquests components, que estan en contenidors, inclouen descriptors de desplegament que són arxius XML, els quals ens serveixen per configurar l'entorn d'execució.

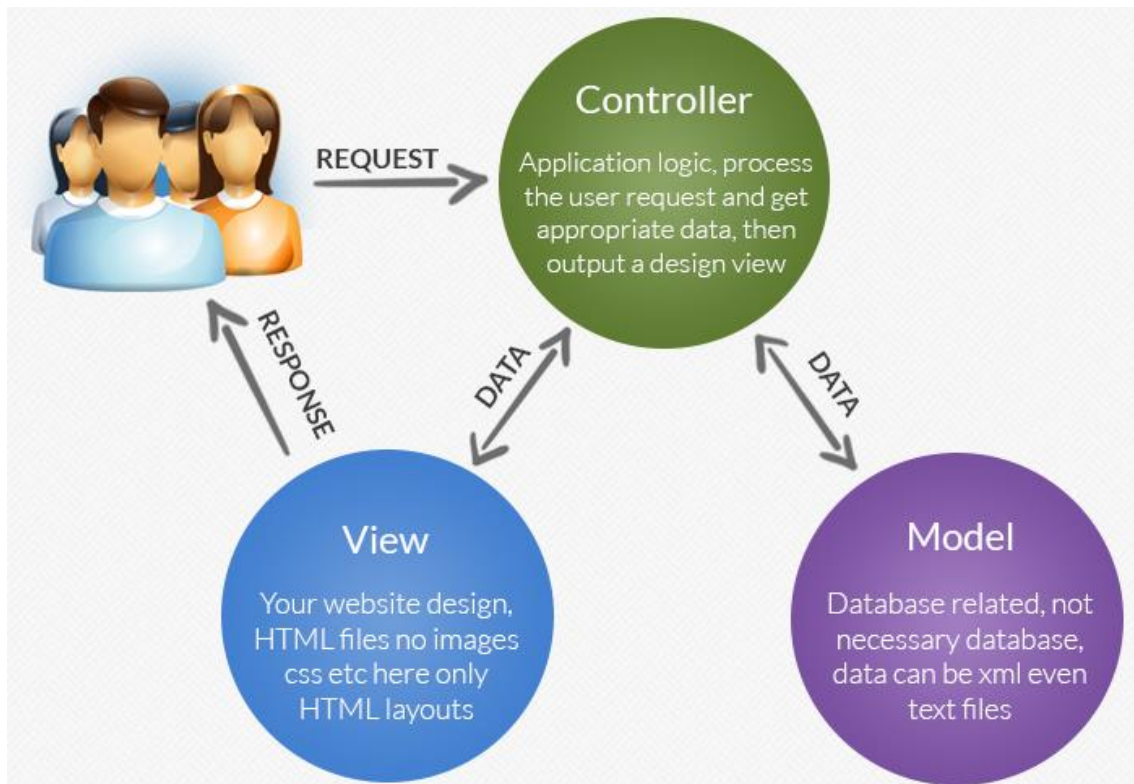
Per dur a terme el projecte necessitem un servidor d'aplicacions J2EE, entre la varietat que existeix s'ha escollit JBOSS que és de lliure distribució.

Entre els principals avantatges que trobem per utilitzar l'arquitectura multicapa se'n destaquen els següents

- **Manteniment:** Una modificació en una capa no implica modificar les altres capes.
- **Re-usabilitat:** Les dades i el model es defineixen uns sola vegada, permetent així que altres aplicacions les puguin utilitzar sense transgredir cap regla del sistema.
- **Escalabilitat:** El sistema és flexible per dividir-se físicament quan els requeriments de l'aplicació canvien.
- **Portabilitat:** Els usuaris es poden connectar a l'aplicació sense importar el sistema operatiu i beneficiar-se de les millores en el sistema sense la necessitat de descarregar cap software adicional

5.2. MVC i Frameworks per J2EE

El patró Model-Vista-Controlador és una guia pel disseny d'arquitectures d'aplicacions que ofereix una forta interactivitat amb els usuaris. Aquest patró organitza l'aplicació en tres models separats: el model, la vista i el controlador



Il·lustració 10: MVC.

Controlador

És el responsable de rebre les peticions dels clients i és responsable de:

- Processar les peticions determinant quina classe és la que ha de respondre.
- Modificar el model en funció dels paràmetres que rep executant la lògica de negoci que el desenvolupador hagi definit.
- Redirigir a la vista en funció de la lògica de negoci.

El controlador permet l'ús de tecnologies estàndards com per exemple Java Filters, Java Beans, ResourceBundles, XML, etc.

Model

És la part responsable de la gestió de la informació. En aquest punt s'inclouen les classes, eines, llibreries, etc., permetent l'accés a les dades.

Algunes de les tecnologies que s'utilitzen són JDBC, EJB, Hibernate o JPA, entre altres.

Vista

És la responsable de la percepció que tenen els usuaris finals de l'aplicació. S'inclouen les pàgines Html, JS, etc., i en general tots els elements de visualització de l'aplicació. S'alimenta de la informació que el controlador ha captat del model per pintar les pàgines finals

Algunes de les tecnologies són: HTML, JSP, XSTL, PDF, Templates, etc.

Estructura del patró MVC

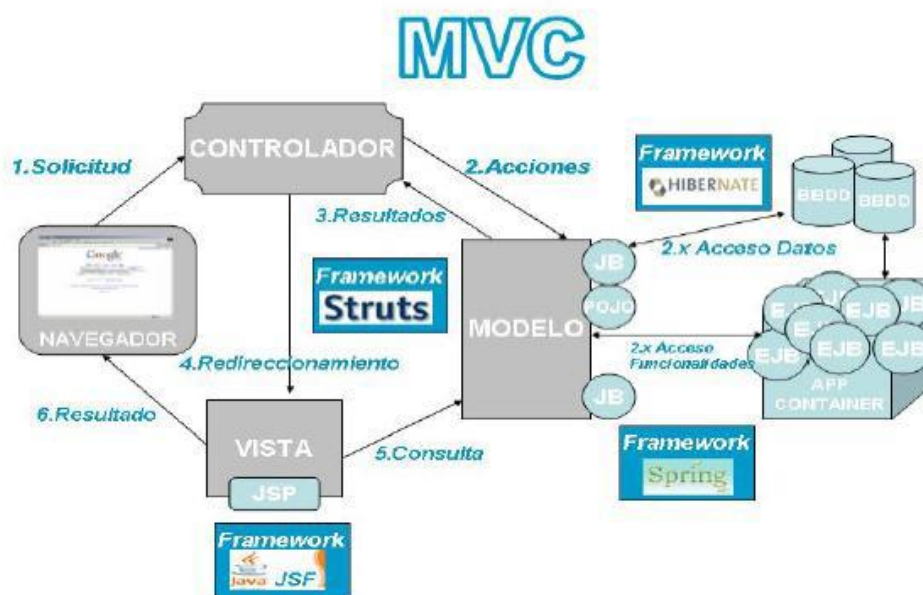
La majoria dels Frameworks per Web implementen aquest patró. Una versió d'aquest patró per programació Web J2EE és el que es coneix amb el nom d'arquitectura model 2. Aquesta arquitectura consisteix en la utilització de servlets per processar les peticions (controladors) i les pàgines JSP per mostrar la interfície d'usuari (vistes); tot implementant la part del model mitjançant JavaBeans. A l'arquitectura MVC, l'usuari llença peticions utilitzant un sol controlador a través de les diferents pàgines que són controlades per un conjunt d'accions prèviament registrades al controlador. Un únic servlet és cridat des del client, quedant visible un sol punt d'entrada al controlador. Aquest és on la lògica de control de l'aplicació (accions que s'han de cridar, vistes que s'han de generar i control de navegació entre pàgines) contrasta amb la primera on va integrat en la capa de presentació.

Frameworks

En el desenvolupament del software un Framework és una estructura de suport definida en el qual un projecte pot ser organitzat i desenvolupat. Un Framework pot incloure suport a programes, les biblioteques i el llenguatge de scriping per ajudar a desenvolupar i unir els diferents components del projecte. A més, proveeix d'una estructura i d'una metodologia de treball.

Els objectius principals que persegueix un Framework són: accelerar el procés de desenvolupament, reutilitzar el codi existent i promoure bones pràctiques de desenvolupament com ara la utilització de patrons. Un Framework Web, per tant, podem definir-lo com un conjunt de components (classes Java, descriptors, arxius de configuració XML) que componen un disseny reutilitzable que facilita i agilitza el desenvolupament de sistemes Web.

Per facilitar el desenvolupament de les aplicacions J2EE existeixen diferents Frameworks, ara veurem uns exemples.



Il·lustració 11: FrameWorks

Java Server Faces: Orientat a la creació d'interfícies d'usuari.

Hibernate: El seu objectiu és facilitar la persistència d'objectes Java amb base de dades relacionals i, al mateix temps, consultar aquesta base de dades per obtenir objectes.

Spring: Té com a objectiu facilitar la configuració dels Java Beans dins d'una aplicació. La seva fita és aconseguir separar els accessos a les dades i als aspectes relacionats amb les transaccions, per permetre objectes de la capa de negoci reutilitzables que no depenguin de cap estratègia d'accés a les dades o transaccions.

Struts: Orientat a la part de control d'events. Aquesta serà utilitzada per la nostra aplicació i explicada amb detall més endavant.

Aquests Frameworks poden ser utilitzats entre ells per poder donar solucions a diferents problemes. És possible una combinació entre els quatre, depenent de les necessitats de l'aplicació. **Aquest projecte utilitzarà Struts i Hibernate ja que inclou més Frameworks incrementaria la carga de treball més enllà del estipulat.**

5.3 STRUTS

L'STRUTS 2 és la nova versió del framework de desenvolupament web Java apache Struts. Però l'STRUTS 2 no està basat amb el codi de l'STRUTS 1.x, sinó en el framework de desenvolupament web Java WebWork.

El framework ens permet reduir el temps de desenvolupament d'aplicacions web. És de software lliure i és compatible amb qualsevol plataforma que disposi de Java.

L'STRUTS 2 està basat amb el patró MVC sota la plataforma J2EE.

Funcionament Struts

Quan es programen aplicacions Web amb el patró MVC sorgeix el dubte d'usar un sol o varis controladors. Si decidim utilitzar un sol controlador, per tenir tota la nostra lògica al mateix lloc, ens trobem amb el problema de convertir el

nostre controlador amb el que es coneix com a “fat controller”, és a dir, un controlador saturat de peticions.

A continuació es podrà veure un cicle de vida d’una petició:

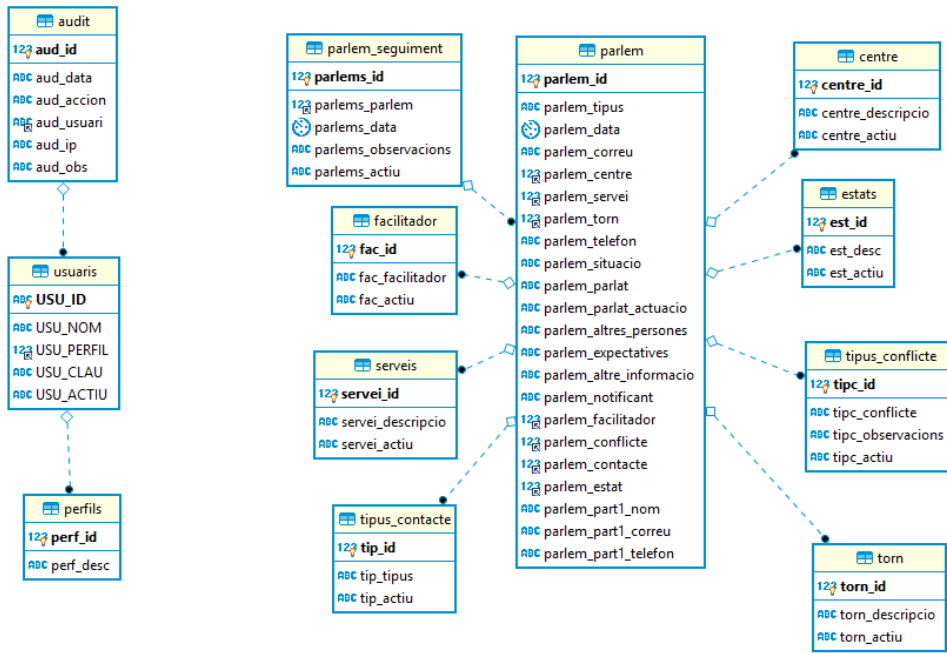
- **Un usuari envia una petició.** Un usuari fa una petició d’un recurs dins del servidor.
- **El FilterDispatcher determina l’acció que haurà de respondre.** El Framework disposa d’elements requerits perquè el dispatcher sigui capaç de determinar quin action és el responsable de rebre la petició i processar-la.
- **S’apliquen els interceptors definits.** Existeixen diferents interceptors que es poden configurar perquè s’executin diferents funcionalitats com ara workflows, validacions, upload de fitxers, etc.
- **S’executa l’Action.** Després d’executar-se els diferents interceptors el mètode específic del Action és executat, realitzant-se aquelles operacions i accions que s’hagin definit. L’Action acaba retornant un resultat que s’utilitza per determinar la pàgina on retornar.
- **Sortida.** Després de l’execució de l’Action es determina quina és la pàgina a retornar i s’executa un forward a la pàgina.
- **Es retorna la petició.** Per poder fer la devolució s’executen els interceptors que corresponen i es procedeix a retornar la petició al client. D’aquesta manera és possible afegir la lògica externa als servidors, i també en la devolució.
- **Es mostra el resultat al client final.** Finalment el control és retornat al client, qui podrà visualitzar el resultat en el seu navegador.

5.4 Hibernate

Com accés a les dades utilitzarem el motor de persistència de codi obert Hibernate. Aquest permetrà guardar un objecte a la base de dades simplement amb `session.save(objecte)` o eliminant-ho fent `session.delete (objecte)`.

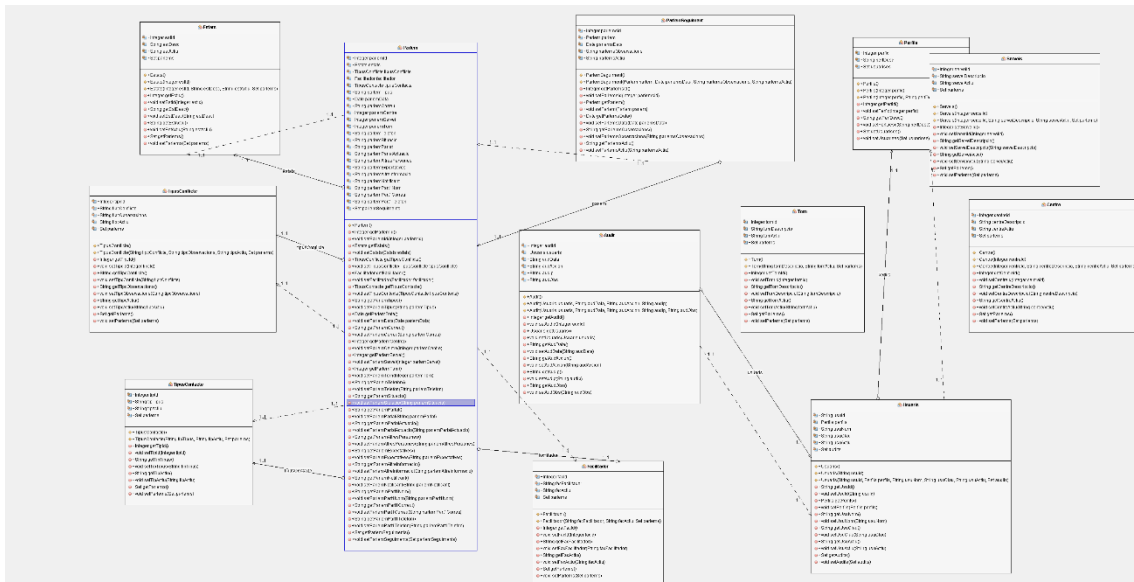
Haurem de tenir un arxiu de configuració XML (`hibernate.cfg.xml`), un conjunt de JavaBeans (seran les classes on cada camp estarà associat amb una columna de la base de dades) i un arxiu XML per cada una d'aquestes classes (`nomclasse.hbm.xml`) que serà el mapping dels objectes.

5.3 Disseny de la base de dades



Il·lustració 12: Entitat relació

5.4 Diagrama de classes



Il·lustració 13: Diagrama de classes.

6. IMPLEMENTACIÓ (REQUERIMENTS DEL SOFTWARE, ESTRUCTURA...)

6.1 - Requeriments del software

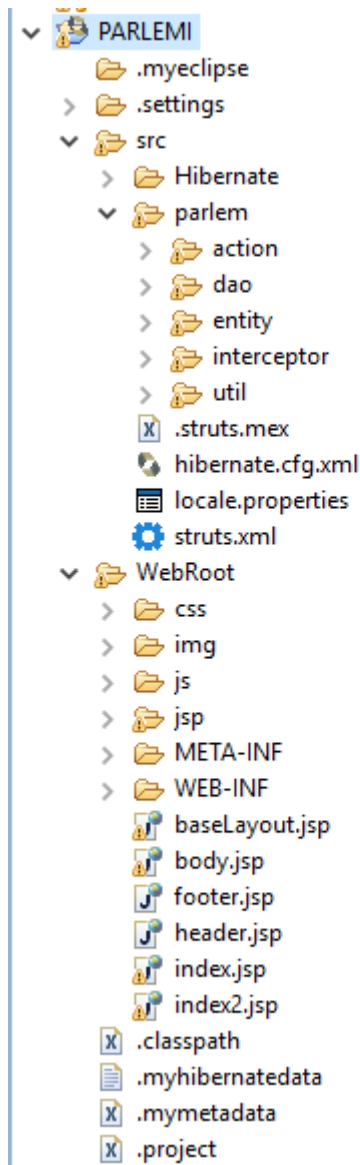
Es proposa que per a la part del servidor s'utilitzi un servidor dedicat amb el Sistema Operatiu Linux i per la resta d'equips clients el Sistema operatiu Windows, ja que és el sistema operatiu més extens entre els usuaris.

6.2 - Els requeriments del software d'aquest projecte són els següents:

- IDE per el desenvolupament Eclipse 2018-09
- Navegador Internet Explorer 11.
- Java Runtime Environment 1.6 i Java Development Kit 1.6 (JRE6 i JDK6) o superior per al desenvolupament i execució del projecte.
- JBOSS com a servidor d'aplicacions J2EE.
- Struts 2 com a framework per implementar el patró MVC
- Hibernate 4.1 com a framework per implementar la persistència de les dades.
- Sistema Gestor de Base de Dades MySQL 5
- Llibreries de la JDBC per a les connexions a la BD (incloses amb l'actual projecte).
- MySQL Administrador 1.1.9 per l'administració de la base de dades
- MySQL workbench 5.2.35
- Tiles com a eina de plantilles.

6.3. Estructura del projecte

L'estructura del projecte es pot veure en la següent imatge:



Il·lustració 14: Estructura del projecte.

A continuació una breu explicació del contingut de les carpetes més rellevants.

La carpeta “src” conté les classes Java i els recursos de l’aplicació: Actions, POJO, DAO, etc. Un dels arxius importants és el fitxer de configuració struts.xml, que és utilitzat per iniciar els recursos del projecte com poden ser els interceptors, actions, results i d’altres.

“src”: Dins d’aquesta carpeta hi ha les classes Java i els recursos de l’aplicació

- **Hibernate**: En aquesta carpeta hi ha tots els mapping de les classes i taules relacionals de la base de dades.

- parlem: En aquesta carpeta hi ha totes les classes Java del projecte. Depenent de la funcionalitat hi ha diferents ubicacions: “action”, “dao”, “entity”, “utils”.
- Hibernate.cfg.xml: Arxiu de configuració dels paràmetres de connexió entre el projecte i la base de dades. Es configura la connexió JDBC indicant, entre d’altres, la base de dades, el port, l’usuari i la contrasenya; i s’indiquen els mapping existents de les classes i taules relacionals.
- Locale.properties: Arxiu dels missatges del projecte. Per mostrar un missatge ho farem de la següent manera
- Struts.xml: Un dels arxius més importants que és utilitzat per iniciar els recursos del projecte, com poden ser els interceptors, actions, results i d’altres



WEBCONTENT

- “css”: És la fulla d’estil que utilitzaran les diferents .jsp de l’aplicació.
- “img”: Carpeta on estaran ubicades les imatges del projecte.
- “js”: Carpeta on estaran ubicats els arxius javascript
- “jsp”: Tots els arxius .jsp que seran els que mostrarà el navegador d’internet.
- “META-INF”
- “WEB-INF”:
 - o “lib”: carpeta on hi ha totes les llibreries utilitzades per l’aplicació.
 - o Web.xml: Arxiu de configuració de l’espai web.
 - o tiles.xml: Arxiu de configuració de plantilles.

6.4. Decisions

El disseny ha estat bastant fidel al prototip inicial.

Al cens de treball s’ha afegit una nova columna indicant l’estat de la notificació. L’he reforçat amb colors per poder identificar l’estat més visualment.

Ref.Id..	Nom	Data	Servei	Centre	Descripció Situació	Estat	Edit
36	pepito de los palotes	12/05/2019	Servei 11	Centre3	asdfsasf	Nou	
40	Jonatan Nualart Sanchez	10/05/2019	Servei 2	Centre 1	El company no para de posar-se amb mi	Obert	
37	Jonatan Nualart Sanchez	09/05/2019	Servei 2	Centre 1	El meu no pare de faltar-me el respecte	Tancat	
38	Jonatan Nualart	08/05/2019	Servei 3	Centre 1	El meu company m'ha perdut el respecte	Nou	
39	Jonatan Nualart Sanchez	08/05/2019	Servei 43	Centre4	He tingut problemes amb el company perquè para de posar-se	Nou	

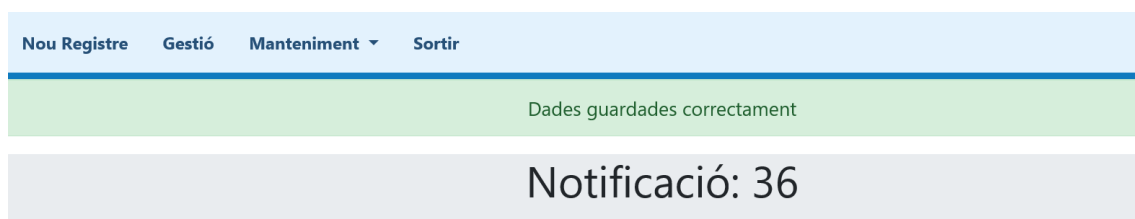
Il·lustració 15: Cens treball columna estat.

He afegit un control de la sessió, quan es caduca surt un missatge indicant-ho:



Il·lustració 16 Missatge Sessió Caducada

S'ha afegit un missatge indicant que les dades s'han guardat correctament. No ho vaig tenir en compte i quan s'enviaven les dades a guardar no hi havia cap senyal que indiqués que s'havia fet correctament transmetent una mica d'inseguretat.



Il·lustració 17 Missatge dades guardades correctament

6.4. Consideracions a l'hora d'instal·lar

6.4.1. - Instal·lació de Java (jdk)

Descarregar la plataforma Java actual, en concret Java Development Kit de la ruta:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Executar l'arxiu descarregat, i continuar amb la configuració per defecte.

6.4.2 - Servidor d'aplicacions web (WildFly)

Es necessita el servidor d'aplicacions WildFly que es pot trobar en el següent enllaç:

<https://wildfly.org/downloads/>

Per engegar el servei s'ha d'executar l'arxiu **standalone.bat** que està dins de la carpeta "**bin/**"

Perquè funcioni correctament el projecte s'ha de configurar les variable d'entorn de l'equip en aquest cas:

- JBOSS_HOME: C:/WildFly
- JAVA_HOME: C:\Program Files\Java\jdk1.8.0_73
- CLASSPATH: C:\Program Files\Java\jdk1.8.0_73/lib

6.4.3 - Servidor base de dades (MySQL)

1. Descarregar el servidor de la ruta :

<https://dev.mysql.com/downloads/installer/>

2. Executar l'arxiu mysql-installer-community-8.0.15.0.msi i continuar amb la configuració per defecte.

3. Instal·lar el programa: MySQL GUI Tools Bundle que inclou MySQL Administrator, MySQL Query Browser:

6.4.4 – Executar l'aplicació

Si s'obre un navegador web i a la direcció posem:

<http://localhost:8080/PARLEMI> s'obrirà el projecte i es podrà començar a treballar.

7. VALORACIÓ ECONÒMICA.

A continuació una proposta de valoració econòmica, cal dir que en la proposta he comptat el preu hora a 25€ sense tenir en compte el perfil del professional en cada fase del projecte.

Descripció	Quantitat(hores)	Preu(hora/€)	Total(€)
Anàlisis i recollida requeriments	35	25,00	875,00
Arquitectura del projecte	35	25,00	875,00
Disseny	75	25,00	1875,00
Desenvolupament	75	25,00	1875,00
Implementació	34	25,00	850,00

Subtotal	6350,00
IVA 21%	1333,50
TOTAL:	7683,50

8. CONCLUSIONS

El Treball de Final de Màster (TFM) planteja l'avaluació com una eina per saber la percepció de l'estudiant sobre la seva formació i quin grau d'aprenentatge ha assolit al llarg de la seva formació. Cal destacar que pel desenvolupament de TFM ha calgut familiaritzar-se amb noves tecnologies que no s'han vist durant els estudis i ha suposat un sobreesforç.

En un principi l'objectiu principal del projecte és la realització d'una aplicació web i convertir-la en una eina per la resolució de conflictes entre professionals gestionada per la figura del facilitador.

Pel desenvolupament del projecte em vaig marcar tres objectius ben diferenciats dividits per les tasques que calguessin per assolir-los. Tenia clar que primer havia de fer la part de la notificació per part del treballador. Un cop aquesta part estigués acabada el segon objectiu seria la part de la gestió per part del facilitador. L'ordre era important ja que part de la feina que havia fet per la introducció de la notificació sabia que la podia aprofitar a l'hora de desenvolupar la part de la gestió. I per últim els manteniment de les taules auxiliars.

En la notificació del conflicte m'he permès afegir preguntes que no són tant del propi conflicte sinó més de caire estadístic i d'interès del departament que els gestiona. L'objectiu principal era fer un formulari senzill, intuïtiu on l'usuari es sentís còmode a l'hora de fer-ho. Per complir amb aquest objectiu amb marca fites, intentava que fossin setmanals, com ara primera fita formulari fet. Segona fita accés a BBDD i així fins tenir-ho acabat.

En la gestió tenia he pensat que era millor que la part que havia introduït el treballador no es pogués modificar per tenir sempre la part original intacte escrita per l'usuari. La primera complicació ha estat com encabir tota la informació a la pantalla, era un requeriment que tot estigués visible en una pantalla, però al final crec que ho he aconseguit. De la mateixa manera que em marcava fites en la notificació també em vaig marcar fites per la gestió. Cada fita intentava lliurar-la setmanalment, a vegades disminuïa el temps per poder fer més interaccions per tal de controlar el temps d'entrega. En aquesta ocasió vaig tenir que modificar el plantejament inicial per tal de poder registrar accions que es podien generar en diferents moments. És per això que vaig tenir que afegir un petit formulari on es fa dues preguntes: una data i uns comentaris.

El tercer objectiu ha estat el més fàcil, suposo que per l'experiència

Sobre la planificació cal dir que estic satisfet de la temporització que vaig fer ja que la he pogut seguir-la sense problemes. El fet de marcar-me fites i fer entregues setmanals m'ha servit com indicador de la situació real del projecte. Si detectava alguna desviació en el temps global que podia perjudicar el projecte tenia la tranquil·litat de rectificar i ajustar-me a la planificació perquè aquesta es complís. Penso que és una metodologia que faré servir en els projectes que em puguin sorgir.

M'he trobat molt còmode treballant amb STRUTS2 i Hibernate. He de reconèixer que el meu fort no és el disseny i cal dir que el Framework de bootstrap m'ha facilitat molt aquesta part.

He optat per delegar la part d'explotació de dades al departament de Business Intelligence (BI) però penso que potser m'ha faltat afegir la possibilitat de poder realitzar un informe final a la pantalla de la gestió.

Penso que afegint més estats (actualment només existeixen 3) es podria generar fluxos de treballs interessants. Per exemple es podria afegir un estat que fos pendent de revisió cap superior on aquest revés un avís (per exemple via correu electrònic) on pogués accedir a l'aplicació i afegir els seus comentaris.

Poder adjuntar documents es una funcionalitat que actualment no existeix però podria ser molt útil per reforçar alguns arguments.

Finalment crec que és un eina que està enfocada a qualsevol tipus d'empresa de qualsevol sector i per aquest motiu penso que el ventall de possibilitats al mercat és molt ampli.

9. GLOSSARI

Administrador: Usuari del sistema amb privilegis especials.

BBDD: Són les inicials de base de dades

Framework: És un terme adoptat de l'anglès i equival a “entorn de treball” o “capa de treball”. Inclou tots aquells recursos, metodologia, programari de suport, llibreries de codi, llenguatges de programació, i programari extra que ajudi a desenvolupar i integrar un projecte de programari

J2EE: Java Enterprise Edition, plataforma de desenvolupament de Java amb components distribuïbles amb N capes desenvolupada en components modulars que s'executa mitjançant programari de servidors d'aplicacions.

JSP: Java Server Pages és una tecnologia de l'empresa Sun Microsystems que permet la creació de pàgines web dinàmiques que es modelen segons les necessitats del moment.

MVC: Model-View-Controller o Model-Vista-Controlador és un patró de disseny d'aplicacions que divideix la mateix en tres parts independents que són la de model, la de vista i la de control. És el que millora la reutilització i el manteniment.

Servidor d'aplicacions: Dispositiu, normalment de software, que permet l'execució i la gestió de certes parts (o la seva totalitat) d'un programari desenvolupat expressament per aquest tipus de entorns

Servlet: Dintre del context de l'arquitectura J2EE, és una classe que s'executa dins d'un contenidor web i que compleix un contracte que el facilita per atendre peticions HTTP.

Struts: Eina de suport al desenvolupament d'aplicacions que permet d'una manera fàcil i ràpida la creació d'aplicacions seguint el patró MVC.

UML: UML o llenguatge unificat de modelatge és un llenguatge gràfic estàndard per visualitzar, especificar, construir i documentar un sistema.

Usuari: Qualsevol persona que s'autentifiqui contra el sistema i interaccioni amb ell.

10. BIBLIOGRAFIA

Keogh, Jim (2003). Manual de referencia. J2EE. Madrid (Espanya): McGraw-Hill

Roughley , Ian (2006). Starting Struts 2. C4Media, Publisher of InfoQ.com.

Roughley , Ian (2007). Practical Apache Struts2 Web 2.0 Projects. Apress.

Brown ,Donald; Chad, Michael Davis (2007). Struts2 in action. MEAP Edition Manning Early Access Program.

Oracle Corporation (2010). The Java EE 5 Tutorial. [en línia].
<http://download.oracle.com/javaee/5/tutorial/doc/bnagx.html>

Oracle Corporation (2010). Java EE Tutorials. [en línia].
<http://www.oracle.com/technetwork/java/javaee/documentation/tutorials-137605.html>

Pons Ricardo (2010). Trabajando con Hibernate y Eclipse Helios. Parte I (Preparando el Entorno). [En línia].
<http://ricardopons.wordpress.com/2010/09/24/trabajando-conhibernate-y-eclipse-helios-parte-i-preparando-el-entorno>

Pons Ricardo (2010). Trabajando con Hibernate y Eclipse Helios. Parte I (Preparando el Entorno). [En línia].
<http://ricardopons.wordpress.com/2010/09/27/trabajando-conhibernate-y-eclipse-helios-parte-ii-configurando-hibernate/>

11. ANNEXA.

Exemple del codi d'una entitat, en aquest cas la entitat TORN:

```
package parlem.entity;
import java.util.HashSet;
import java.util.Set;
public class Torn implements java.io.Serializable {
    // Fields
    private Integer tornId;
    private String tornDescripcio;
    private String tornActiu;
    private Set parlems = new HashSet(0);
    // Constructors
    /** default constructor */
    public Torn() {
```

```

}
/** full constructor */
public Torn(String tornDescripcio, String tornActiu, Set parlems) {
    this.tornDescripcio = tornDescripcio;
    this.tornActiu = tornActiu;
    this.parlems = parlems;
}
// Property accessors
public Integer getTornId() {
    return this.tornId;
}
public void setTornId(Integer tornId) {
    this.tornId = tornId;
}
public String getTornDescripcio() {
    return this.tornDescripcio;
}
public void setTornDescripcio(String tornDescripcio) {
    this.tornDescripcio = tornDescripcio;
}
public String getTornActiu() {
    return this.tornActiu;
}
public void setTornActiu(String tornActiu) {
    this.tornActiu = tornActiu;
}
public Set getParlems() {
    return this.parlems;
}
public void setParlems(Set parlems) {
    this.parlems = parlems;
}

```

```
    }  
}
```

Exemple del codi d'una classe DAO (Data access object) Interface d'accés a les dades emmagatzemades:

```
package parlem.dao;  
  
import java.util.List;  
import java.util.Set;  
import org.apache.commons.logging.Log;  
import org.apache.commons.logging.LogFactory;  
import org.hibernate.LockMode;  
import org.hibernate.Query;  
import org.hibernate.criterion.Example;  
  
import parlem.entity.Torn;  
  
import Hibernate.BaseHibernateDAO;  
  
/**  
 * A data access object (DAO) providing persistence and search support for Torn  
 * entities. Transaction control of the save(), update() and delete() operations  
 * can directly support Spring container-managed transactions or they can be  
 * augmented to handle user-managed Spring transactions. Each of these methods  
 * provides additional information for how to configure it for the desired type  
 * of transaction control.  
 *  
 * @see Hibernate.Torn  
 * @author MyEclipse Persistence Tools  
 */  
public class TornDAO extends BaseHibernateDAO {  
    private static final Log log = LogFactory.getLog(TornDAO.class);
```

```

// property constants
public static final String TORN_DESCRIPCIO = "tornDescripcio";
public static final String TORN_ACTIU = "tornActiu";

public void save(Torn transientInstance) {
    log.debug("saving Torn instance");
    try {
        getSession().save(transientInstance);
        log.debug("save successful");
    } catch (RuntimeException re) {
        log.error("save failed", re);
        throw re;
    }
}

public void delete(Torn persistentInstance) {
    log.debug("deleting Torn instance");
    try {
        getSession().delete(persistentInstance);
        log.debug("delete successful");
    } catch (RuntimeException re) {
        log.error("delete failed", re);
        throw re;
    }
}

public Torn findById(java.lang.Integer id) {
    log.debug("getting Torn instance with id: " + id);
    try {
        Torn instance = (Torn) getSession().get("parlem.entity.Torn", id);
        return instance;
    }
}

```

```

    } catch (RuntimeException re) {
        log.error("get failed", re);
        throw re;
    }
}

public List findByExample(Torn instance) {
    log.debug("finding Torn instance by example");
    try {
        List results = getSession().createCriteria("parlem.entity.Torn")
            .add(Example.create(instance)).list();
        log.debug("find by example successful, result size: "
            + results.size());
        return results;
    } catch (RuntimeException re) {
        log.error("find by example failed", re);
        throw re;
    }
}

public List findByProperty(String propertyName, Object value) {
    log.debug("finding Torn instance with property: " + propertyName
        + ", value: " + value);
    try {
        String queryString = "from Torn as model where model."
            + propertyName + "= ?";
        Query queryObject = getSession().createQuery(queryString);
        queryObject.setParameter(0, value);
        return queryObject.list();
    } catch (RuntimeException re) {
        log.error("find by property name failed", re);
    }
}

```

```

        throw re;
    }
}

public List findByTornDescripcio(Object tornDescripcio) {
    return findByProperty(TORN_DESCRIPCIO, tornDescripcio);
}

public List findByTornActiu(Object tornActiu) {
    return findByProperty(TORN_ACTIU, tornActiu);
}

public List findAll() {
    log.debug("finding all Torn instances");
    try {
        String queryString = "from Torn";
        Query queryObject = getSession().createQuery(queryString);
        return queryObject.list();
    } catch (RuntimeException re) {
        log.error("find all failed", re);
        throw re;
    }
}

public Torn merge(Torn detachedInstance) {
    log.debug("merging Torn instance");
    try {
        Torn result = (Torn) getSession().merge(detachedInstance);
        log.debug("merge successful");
        return result;
    } catch (RuntimeException re) {

```

```

        log.error("merge failed", re);
        throw re;
    }
}

public void attachDirty(Torn instance) {
    log.debug("attaching dirty Torn instance");
    try {
        getSession().saveOrUpdate(instance);
        log.debug("attach successful");
    } catch (RuntimeException re) {
        log.error("attach failed", re);
        throw re;
    }
}

public void attachClean(Torn instance) {
    log.debug("attaching clean Torn instance");
    try {
        getSession().lock(instance, LockMode.NONE);
        log.debug("attach successful");
    } catch (RuntimeException re) {
        log.error("attach failed", re);
        throw re;
    }
}
}
}

```

Les accions en STRUTS son classes simples de JAVA (POJOS) que estenen de la classe ActionSuport i s'encarreguen de la lògica del negoci, exemple d'ACTION :

```
package parlem.action;
```

```

import java.awt.datatransfer.SystemFlavorMap;
import java.util.Hashtable;
import java.util.Map;
import javax.naming.AuthenticationException;
import javax.naming.Context;
import javax.naming.NamingEnumeration;
import javax.naming.NamingException;
import javax.naming.directory.Attributes;
import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;
import javax.naming.directory.SearchControls;
import javax.naming.directory.SearchResult;
import javax.naming.ldap.LdapContext;
import javax.xml.registry.infomodel.AuditableEvent;
import parlem.dao.AuditDAO;
import parlem.dao.UsuarisDAO;
import parlem.entity.Audit;
import parlem.entity.Usuaris;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
public class loginAction extends ActionSupport{

    private String usuari;
    private String password;
    private boolean loggin;
    private String accio;
    private String session;
    public String getSession() {
        return session;
    }
    public void setSession(String session) {

```



```

        this.session = session;
    }
    public String getUsuari() {
        return usuari;
    }
    public void setUsuari(String usuari) {
        this.usuari = usuari;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public boolean isLoggin() {
        return loggin;
    }
    public void setLoggin(boolean loggin) {
        this.loggin = loggin;
    }
    /**
     * @return the accio
     */
    public String getAccio() {
        return accio;
    }
    /**
     * @param accio the accio to set
     */
    public void setAccio(String accio) {
        this.accio = accio;
    }

```

```

}
public String execute(){
    boolean logat=false;
    String perfil ="";
    Usuaris usuaris = new Usuaris();
    UsuarisDAO usuarisDAO = new UsuarisDAO();
    try {
        Map session = ActionContext.getContext().getSession();
        if (accio!=null){
            if (accio.equals("exit")){

                session.remove("logged");
                session.remove("perfil");
                session.remove("usuari");
                return "input";
            }
            if (accio.equals("login")){
                perfil = usuarisDAO.getUsuariLogat(this.usuari, this.password);
                if(perfil!=null){
                    session.put("logged", "true");
                    session.put("perfil",perfil);
                    session.put("usuari",usuari);
                    logat= true;
                }
            }
            if(logat){
                usuaris = usuarisDAO.findByUsu(usuari);
                AuditDAO auditDAO = new AuditDAO();
                auditDAO.DesarAudit(accio, usuaris.getUsuld(), "Login pantalla
inicial");

                return "success";
            }
        }
    }
}

```

```

        }else{
            session.remove("logged");
            session.remove("perfil");
            session.remove("usuari");
            return "exit";
        }

    }

} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

return SUCCESS;
}
}

```

Amb el fitxer struts.xml li estem indicant al STRUTS que existeix un action i a més es defineix com és el mapeig les pàgines resultants.

Exemple struts.xml:

```

<action name="LoginAction" class="parLem.action.LoginAction">
    <param name="accio" >${accio}</param>
    <result name="success" type="redirectAction">LlistatTreball</result>
    <result name="exit">index.jsp</result>
</action>

```

Amb el tag `<%@ taglib uri="/struts-tags" prefix="s" %>` podrem accedir a les seves funcionalitats.

Exemple de jsp:

```

<%@ taglib uri="/struts-tags" prefix="s" %>

```

```

</head>
<body>
<div class="container-fluid">
  <div class="jumbotron text-center" style="padding:0ex;margin-bottom:0px;">
    <h1>
      <s:text name="global.gestio.notificacio"></s:text>
      <s:text name="parLem.parLemId"/>
    </h1>
    <h2><u><s:text name="parLem.parLemNotificant"></s:text></u></h2>
    <b><s:text name="global.gestio.telefon"></s:text></b>
    <s:text name="parLem.parLemTelefon"></s:text>
    &nbsp;
    <b><s:text name="global.gestio.data"></s:text></b>
    <s:text name="parLem.parLemData"></s:text>
    &nbsp;
    <b><s:text name="global.gestio.correu"></s:text></b>
    <s:text name="parLem.parLemCorreu"></s:text>
    &nbsp;
    <b><s:text name="global.gestio.centre"></s:text></b>
    <s:text name="parLem.centre"></s:text>
    &nbsp;
    <b><s:text name="global.gestio.servei"></s:text></b>
    <s:text name="parLem.servei"></s:text>
  </div>
  <div class="row" style="border: //1px solid;padding: 10px;">
    <div class="col-sm-4" style="border: 1px solid;border-color:#0080C0;">
      <b><s:text name="global.comu.descriu.situacio"/></b>
      <br>
      <s:text name="parLem.parLemSituacio"></s:text>
      <br><b><s:text name="global.comu.parLat.comandament"/></b>
      <s:text name="parLem.parLemParLatDesc"></s:text>
      <br><b><s:text name="global.actuacio"></s:text></b>
      <br><s:text name="parLem.parLemParLatActuacio"/>
    </div>
  </div>

```

L'arxiu web.xml proporciona informació sobre la configuració i desplegament dels components web que componen l'aplicació web.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name></display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <context-param>
    <param-name>org.apache.tiles.impl.BasicTilesContainer.DEFINITIONS_CONFIG</param-name>
    <param-value>/WEB-INF/tiles.xml</param-value>
  </context-param>
  <listener>
    <listener-class>org.apache.struts2.tiles.StrutsTilesListener</listener-class>
  </listener>
  <filter>
    <filter-name>struts</filter-name>
    <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts</filter-name>
    <url-pattern>*/url-pattern</url-pattern>
  </filter-mapping>
  <session-config>
    <session-timeout>60</session-timeout>
  </session-config>
</web-app>

```

El fitxer de configuració general de Hibernate on es posa la informació sobre la base de dades que es treballant. En aquest fitxer definim la connexió, el mapeig de les classes, el dialect...

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.zeroDateTimeBehavior">
      convertToNull
    </property>
    <property name="connection.datasource">java:/MySQLparlem</property>
    <mapping resource="Hibernate/Audit.hbm.xml" />
    <mapping resource="Hibernate/Centre.hbm.xml" />
    <mapping resource="Hibernate/ParLem.hbm.xml" />
    <mapping resource="Hibernate/Perfils.hbm.xml" />
    <mapping resource="Hibernate/Torn.hbm.xml" />
    <mapping resource="Hibernate/Usuaris.hbm.xml" />
    <mapping resource="Hibernate/Serveis.hbm.xml" />
    <mapping resource="Hibernate/TipusContacte.hbm.xml" />
    <mapping resource="Hibernate/TipusConflicte.hbm.xml" />
    <mapping resource="Hibernate/ParLemSeguiment.hbm.xml" />
    <mapping resource="Hibernate/Facilitador.hbm.xml" />
    <mapping resource="Hibernate/Estats.hbm.xml" />
  </session-factory>
</hibernate-configuration>

```

DataSource on es defineix la connexió a la base de dades:

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.zeroDateTimeBehavior">
      convertToNull
    </property>
    <property name="connection.datasource">java:/MySQLparlem</property>
    <mapping resource="Hibernate/Audit.hbm.xml" />
    <mapping resource="Hibernate/Centre.hbm.xml" />
    <mapping resource="Hibernate/ParLem.hbm.xml" />
    <mapping resource="Hibernate/Perfils.hbm.xml" />
    <mapping resource="Hibernate/Torn.hbm.xml" />
    <mapping resource="Hibernate/Usuaris.hbm.xml" />
    <mapping resource="Hibernate/Serveis.hbm.xml" />
    <mapping resource="Hibernate/TipusContacte.hbm.xml" />
    <mapping resource="Hibernate/TipusConflicte.hbm.xml" />
    <mapping resource="Hibernate/ParLemSeguiment.hbm.xml" />
    <mapping resource="Hibernate/Facilitador.hbm.xml" />
    <mapping resource="Hibernate/Estats.hbm.xml" />
  </session-factory>
</hibernate-configuration>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>MySQLparlem</jndi-name>

    <connection-url>jdbc:mysql://172.16.1.88:3306/parlem</connection-url>
    <driver-class>org.gjt.mm.mysql.Driver</driver-class>
    <user-name>root</user-name>
    <password>unset</password>
    <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter</exception-sorter-class-name>

    <min-pool-size>1</min-pool-size>
    <max-pool-size>5</max-pool-size>

  <metadata>
    <type-mapping>MySQLparlem</type-mapping>
  </metadata>
</local-tx-datasource>
</datasources>

```

JAVABEAN:

```

package parlem.entity;
import java.util.HashSet;
import java.util.Set;
public class Usuaris implements java.io.Serializable {
    // Fields
    private String usuId;
    private Perfils perfils;
    private String usuNom;
    private String usuServei;
    private String usuActiu;
    private Set audits = new HashSet(0);
    // Constructors
    /** default constructor */
    public Usuaris() {
    }
    /** minimal constructor */
    public Usuaris(String usuId) {
        this.usuId = usuId;
    }
    /** full constructor */
    public Usuaris(String usuId, Perfils perfils, String usuNom,
        String usuServei, String usuActiu, Set audits) {
        this.usuId = usuId;
        this.perfils = perfils;
        this.usuNom = usuNom;
        this.usuServei = usuServei;
        this.usuActiu = usuActiu;
        this.audits = audits;
    }
    // Property accessors
    public String getUsuId() {
        return this.usuId;
    }
    public void setUsuId(String usuId) {
        this.usuId = usuId;
    }
    public Perfils getPerfils() {
        return this.perfils;
    }
    public void setPerfils(Perfils perfils) {
        this.perfils = perfils;
    }
    public String getUsuNom() {
        return this.usuNom;
    }
    public void setUsuNom(String usuNom) {
        this.usuNom = usuNom;
    }
    public String getUsuServei() {
        return this.usuServei;
    }
    public void setUsuServei(String usuServei) {
        this.usuServei = usuServei;
    }
    public String getUsuActiu() {
        return this.usuActiu;
    }
    public void setUsuActiu(String usuActiu) {

```

Fitxer mapping hibernate corresponent:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!--
Mapping file autogenerated by MyEclipse Persistence Tools
-->
<hibernate-mapping>
  <class name="parlem.entity.Usuaris" table="usuaris" catalog="parLem">
    <id name="usuId" type="java.lang.String">
      <column name="USU_ID" length="100" />
      <generator class="assigned" />
    </id>
    <many-to-one name="perfils" class="parlem.entity.Perfils" fetch="select">
      <column name="USU_PERFIL" />
    </many-to-one>
    <property name="usuNom" type="java.lang.String">
      <column name="USU_NOM" length="50" />
    </property>
    <property name="usuServei" type="java.lang.String">
      <column name="USU_SERVEI" length="11" />
    </property>
    <property name="usuActiu" type="java.lang.String">
      <column name="USU_ACTIU" length="1" />
    </property>
    <set name="audits" inverse="true">
      <key>
        <column name="aud_usuari" length="100" not-null="true" />
      </key>
      <one-to-many class="parlem.entity.Audit" />
    </set>
  </class>
</hibernate-mapping>
```

Com es pot veure l'etiqueta class s'utilitza per associar la nostra classe amb la taula que li correspon a la base de dades. Les etiquetes de més avall en serveixen per definir la relació entre les propietats de la classe i les columnes de la base de dades.

Fitxer tiles.xml on es defineix la configuració del FrameWork Apache Tiles:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
"-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"
"http://tiles.apache.org/dtds/tiles-config_2_0.dtd">
<tiles-definitions>
  <definition name="baseLayout" template="/baseLayout.jsp">
    <put-attribute name="header" value="/header.jsp"/>
    <put-attribute name="body" value="/body.jsp"/>
    <put-attribute name="footer" value="/footer.jsp"/>
    <put-attribute name="sessionexpired" value="/sessionexpired.jsp"/>
  </definition>
  <definition name="LlistatTreball" extends="baseLayout">
    <put-attribute name="title" value="PARLEM"/>
    <put-attribute name="body" value="/jsp/LlistatTreball.jsp"/>
    <put-attribute name="footer" value="/footer.jsp"/>
  </definition>
  <definition name="nouRegistre" extends="baseLayout">
    <put-attribute name="title" value="Nou Registre"/>
    <put-attribute name="body" value="/jsp/nouRegistre.jsp?accio=nou"/>
    <put-attribute name="footer" value="/footer.jsp"/>
  </definition>
  <definition name="gestio" extends="baseLayout">
    <put-attribute name="title" value="Gestio"/>
    <put-attribute name="body" value="/jsp/gestio.jsp"/>
    <put-attribute name="footer" value="/footer.jsp"/>
  </definition>
</tiles-definitions>

```