

Euromaster-Quality

Nom Estudiant: Oriol Santacruz Alba

ETIG / ETIS

Nom Consultor: Jose Juan Rodriguez

Data Lliurament: 16/01/2012

1. Resum

Aquest projecte pretén cobrir una necessitat concreta d'empresa, gestionar el departament de qualitat, facilitant eines per a una correcta administració d'aquesta secció, permet la gestió d'usuaris, l'administració d'informes de no conformitat i d'accions correctives i/o preventives, a més posa a disposició un quadre de control d'activitats dintre del departament de SHEQ.

El projecte s'ha desenvolupat utilitzant diverses eines de programació que han solucionat totes les exigències que s'ha presentat al llarg de l'evolució d'aquest, el treball s'ha centrat en la tecnologia J2EE, una plataforma de programació que s'executa sobre un servidor d'aplicacions, i s'ha recolzat en diferents frameworks com Hibernate 3.2.5 per resoldre la capa de persistència, Spring 3.0.2 per a gestionar la capa de negoci i JSF 2.1 per l'administració de la capa vista, també s'ha utilitzat el framework Spring Security 3.0.5 per al control d'accés i autenticació d'usuaris a l'aplicació. Paral·lelament per a la gestió y construcció, el projecte està recolzat per l'eina de software Maven 3.0 conjuntament amb NetBeans 7.0.1. Per a la gestió de la base de dades el projecte es recolza en el sistema MySQL 5.2. Com a servidor web d'aplicacions s'utilitza Apache Tomcat 7.0.23.

© (Oriol Santacruz Alba)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

2. Índex de continguts

1. Resum.....	2
2. Índex de continguts	3
3. Índex de figures.	5
4 Introducció.	7
4.1. Justificació del TFC i context en el qual es desenvolupa.	7
4.2. Objectius del TFC.....	8
4.3. Enfocament i mètode seguit.....	9
4.4. Planificació del projecte.	10
4.5. Productes obtinguts	12
4.6. Breu descripció dels altres capítols de la memòria.....	13
5. Anàlisi.....	14
5.1 Actors	14
6. Casos d'ús	15
6.1. Casos d'ús Subsistema Manteniment	16
6.2. Casos d'ús Subsistema Funcions	17
6.3. Casos d'ús Subsistema de Connexió	22
7. Disseny	23
7.1. Arquitectura; Vista, Negoci, Persistència	23
7.2. Frameworks	28
7.3. Accés a dades.....	32
8. Diagrames de classes	34
8.1. Model DAO	34
8.2. Diagrama model DAO.....	34

9. Interfície gràfica.....	36
9.1. Funcionalitats Subsistema de Manteniment	36
9.2. Funcionalitats Subsistema de Funcions.....	39
9.3. Funcionalitats Subsistema Connexió.....	47
9.4. Decisions de Disseny i Implementació	49
9.5. Relació de funcionalitats implementades	50
9.6. Eines utilitzades per al desenvolupament de l'aplicació.....	51
10. Conclusions finals.....	52
11. Glossari.....	53
12. Bibliografia.....	54
13. Annex	55
13.1. Creació de la Base de Dades	55
13.2. Manual d'Instal·lació	60

3. Índex de figures.

<i>Figura 1: Diagrama General casos d'ús.....</i>	<i>15</i>
<i>Figura 2: Casos d'ús Subsistema Manteniment.....</i>	<i>16</i>
<i>Figura 3: Casos d'ús Subsistema Funcions.....</i>	<i>17</i>
<i>Figura 4: Casos d'ús Subsistema Connexió.....</i>	<i>22</i>
<i>Figura 5: MVC.....</i>	<i>24</i>
<i>Figura 6: Model E/R.....</i>	<i>32</i>
<i>Figura 7: Diagrama Classes.....</i>	<i>35</i>
<i>Figura 8: Diagrama DAO.....</i>	<i>35</i>
<i>Figura 9: Alta Usuari.....</i>	<i>36</i>
<i>Figura 10: Modificar Usuari Llistat.....</i>	<i>37</i>
<i>Figura 11: Modificar Usuari.....</i>	<i>37</i>
<i>Figura 12: Eliminar Usuari.....</i>	<i>38</i>
<i>Figura 13: Alta INC.....</i>	<i>39</i>
<i>Figura 14: Tancar INC.....</i>	<i>40</i>
<i>Figura 15: Tancar INC Genera ACP.....</i>	<i>40</i>
<i>Figura 16: Llistat INC</i>	<i>41</i>
<i>Figura 17: Alta ACP.....</i>	<i>42</i>
<i>Figura 18: Tancar ACP Llistat.....</i>	<i>43</i>
<i>Figura 19: Tancar ACP</i>	<i>43</i>
<i>Figura 20: Llistat ACP.....</i>	<i>44</i>

<i>Figura 21: Control Activitats.....</i>	<i>45</i>
<i>Figura 22: Informació.....</i>	<i>46</i>
<i>Figura 23: Inici Sessió.....</i>	<i>47</i>
<i>Figura 24: Tancar Sessió.....</i>	<i>48</i>
<i>Figura 25: Obrir Sessió.....</i>	<i>48</i>

4 Introducció.

4.1. Justificació del TFC i context en el qual es desenvolupa.

Inicialment la idea de desenvolupar aquest projecte va ser suggerida per un familiar que tenia una necessitat per a la seva empresa, en principi pensava fer una aplicació en J2SE però més endavant vaig decidir-me per utilitzar J2EE ja que era una tecnologia nova per a mi i podria aprofitar per aprendre nous conceptes, a part és més adient per al tipus d'aplicació que se m'havia demanat.

Dintre de l'entorn J2EE hi ha frameworks molt utilitzats per al desenvolupament i gestió de projectes que m'han semblat de molt interès per a futures aplicacions i a l'hora d'especialitzar-me en un camp concret dintre del món de l'informàtica m'ha semblat que és un camí molt fructífer.

4.2. Objectius del TFC.

Els objectius d'aquest TFC com s'ha comentat anteriorment, són primerament cobrir la necessitat d'una empresa, punt important ja que a partir d'aquest projecte es poden desenvolupar aplicacions semblants per a la distribució dins del món empresarial.

Com a segon punt, l'aprenentatge d'una nova tecnologia, molt utilitzada avui dia i en constant progrés i demanda, la majoria d'ofertes de treball que es poden trobar ara mateix són referides a aquest camp, motivació suficient per a decidir-se per aquesta disciplina.

4.3. Enfocament i mètode seguit.

Inicialment com ja es disposava d'una idea clara per al projecte i recolzament per part de l'empresa interessada la definició del projecte ha sigut una tasca força assequible, en canvi tota la feina estalviada en aquest punt, ha servit per reforçar el temps d'aprenentatge de les tecnologies utilitzades, els diferents frameworks utilitzats han requerit de molt temps per a la total comprensió de la funcionalitat que disposen. Per tant l'etapa d'aprenentatge ha sigut la més dura, juntament amb l'implementació, quasi paral·leles.

El projecte estructurat en 3 PACs i memòria facilita la distribució de la feina i la planificació d'aquesta, ja que marca uns temps i s'han de seguir per poder acabar el projecte satisfactòriament.

4.4. Planificació del projecte.

Per a la planificació del projecte s'ha tingut en compte l'aprenentatge de les tecnologies utilitzades, la majoria de les fites han sigut assolides en el temps establert, gràcies a les recomanacions del tutor i al suport de l'empresa interessada

Codi Tasca	Nom tasca	Duració (en dies)	Data Inici	Data fi
1	Fita 1: Especificació i Planificació	13	23/09/2011	05/10/2011
2	Cercar recursos	2	23/09/2011	25/09/2011
3	Elaboració proposta de descripció del projecte	2	26/09/2011	28/09/2011
4	Elaboració proposta de la planificació	1	29/09/2011	29/09/2011
5	Elaboració de les especificacions de cada subsistema	4	30/10/2011	03/10/2011
6	Elaboració del document final a entregar	1	04/10/2011	04/10/2011
7	Entrega del document d'Especificació	1	05/10/2011	05/10/2011
8	Fita 2: Disseny	35	06/10/2011	10/11/2011
9	Aprenentatge de la tecnologia J2EE	30	06/10/2011	04/11/2011
10	Muntar entorn de desenvolupament	30	06/10/2011	04/11/2011
11	Aprenentatge i estudi frameworks	30	06/10/2011	04/11/2011
12	Disseny del Model de Dades	4	07/10/2011	10/10/2011
13	Elaboració del document de persistència	2	11/10/2011	12/10/2011
14	Elaboració de les descripcions de les Funcionalitats	4	17/10/2011	20/10/2011
15	Elaboració de diagrames	5	21/10/2011	25/10/2011
16	Elaboració del redisseny de les Interfícies	4	26/10/2011	29/10/2011
17	Revisió amb el consultor	1	30/10/2011	30/10/2011
18	Elaboració del document final a entregar	3	31/10/2011	02/11/2011
19	Últims retocs als subsistemes	2	03/11/2011	05/11/2011
20	Entrega del document de disseny	1	06/11/2011	06/11/2011

21	Fita 3: Implementació	39	10/11/2011	19/12/2011
22	Instal·lació de l'entorn de desenvolupament	1	10/11/2011	10/11/2011
23	Instal·lació de l'entorn d'execució	1	11/11/2011	11/11/2011
24	Instal·lació SGBD i creació BBDD	2	12/11/2011	13/11/2011
25	Desenvolupament cas d'ús senzill	1	14/11/2011	14/11/2011
26	Perfeccionar cas d'ús	1	15/11/2011	15/11/2011
27	Replicar procediment als demés casos d'ús	21	16/11/2011	06/12/2011
28	Implementació subsistema Connexió	7	16/11/2011	22/11/2011
29	Implementació subsistema Manteniment	7	23/11/2011	29/11/2011
30	Implementació subsistema Funcions	7	30/11/2011	06/12/2011
31	Revisió amb el consultor	1	07/12/2011	07/12/2011
32	Elaboració del document final a entregar	2	08/12/2011	09/12/2011
33	Últims retocs als sistemes	2	10/12/2011	12/12/2011
34	Entrega del document d'implementació	1	13/12/2011	13/12/2011
35	Fita 4: Memòria	27	20/12/2011	16/01/2012
36	Elaboració del manual d'instal·lació	10	20/12/2011	29/12/2011
37	Elaboració del manual d'utilització	10	30/12/2011	08/01/2012
38	Revisió amb el consultor	1	09/01/2012	09/01/2012
39	Elaboració de la memòria	4	10/01/2012	13/01/2012
40	Revisió de la memòria	2	14/01/2012	15/01/2012
41	Entrega del document	1	16/01/2012	16/01/2012

4.5. Productes obtinguts

El resultat del projecte és una aplicació J2EE per a la gestió del departament de qualitat d'una empresa concreta 'Euromaster', encara que és una aplicació fàcilment adaptable a qualsevol empresa.

Per poder utilitzar l'aplicació s'ha d'estar registrat, és a dir s'ha de pertànyer a l'empresa que l'utilitza, i dins de l'aplicació hi ha dos rols, el d'administrador que podrà accedir a totes les funcionalitats de l'aplicació i l'únic que podrà gestionar els usuaris, i el rol d'empleat que podrà utilitzar les funcionalitats de treball però no podrà gestionar usuaris ni tancar els informes generats.

L'aplicació corre sobre un servidor d'aplicacions i estarà allotjada al servidor de l'empresa.

4.6. Breu descripció dels altres capítols de la memòria.

En els següents capítols es mostra com s'han anat assolit els diferents objectius de cada entrega, descripció de funcionalitats, pel que fa a l'especificació.

Referent al disseny, s'exposaran els casos d'ús, els frameworks utilitzats, i la capa model i accés a dades.

Relacionat amb l' implementació, es representaran els manuals, tant el manual d'instal·lació com el manual d'usuari, les decisions de disseny i Implementació i la relació de funcionalitats implementades.

5. Anàlisi

5.1 Actors

Hi ha dos tipus d'usuaris en la present aplicació, cadascun amb una sèrie de característiques ben definides:

- Administrador: Té funcions de cap administratiu central. Pot fer el manteniment d'usuaris i té accés a totes les funcions de tots els departaments, és l'encarregat de tancar els Informes de No Conformitat i gestionar les Accions Correctives/Preventives.

- Empleat: Aquest tipus d'usuari podrà generar Informes de No Conformitat, Accions Correctives/Preventives i consultar els llistats d'INC i d'ACP, però no podrà tancar cap informe o acció.

6. Casos d'ús

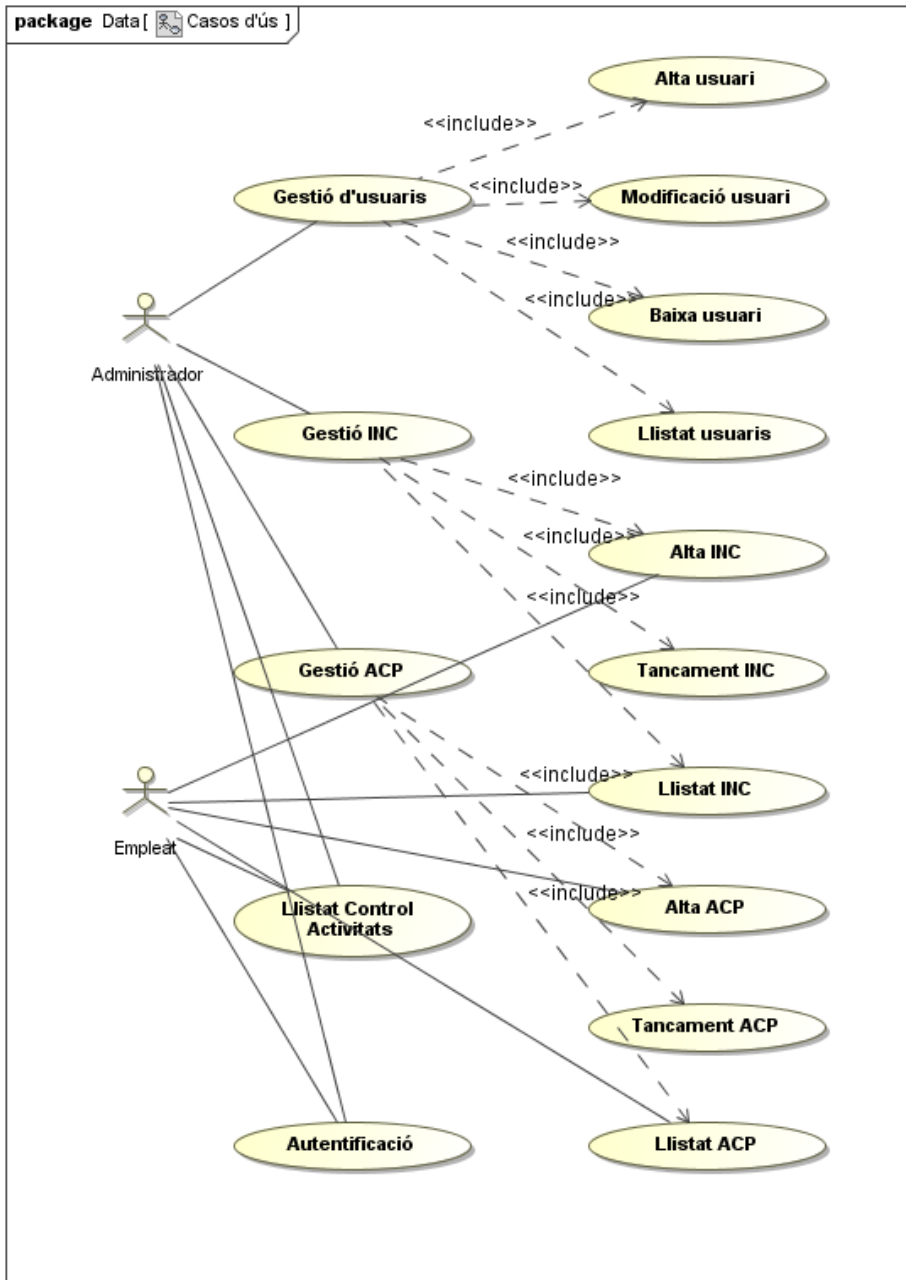


Figura 1: Diagrama General casos d'ús

6.1. Casos d'ús Subsistema Manteniment

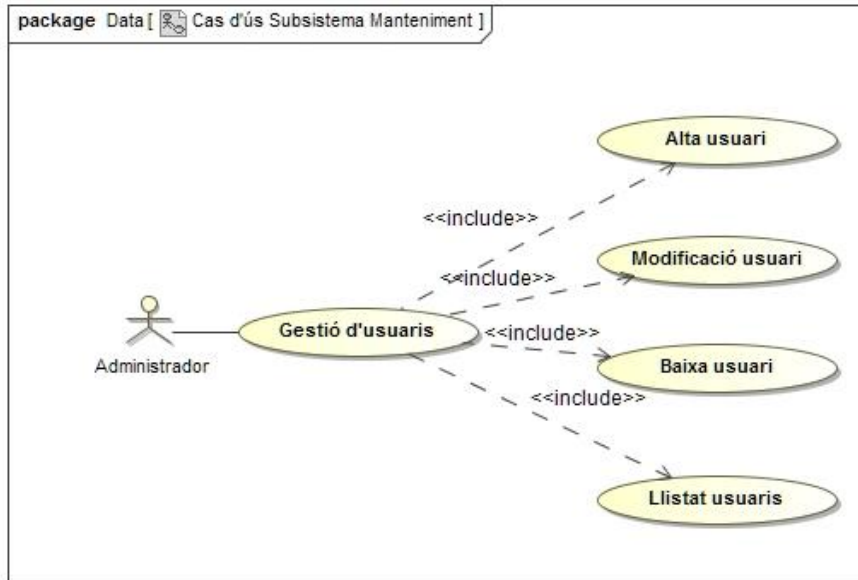


Figura 2: Casos d'ús Subsistema Manteniment

6.1.1 Cas d'ús Gestió Usuaris

Resum de la funcionalitat general	Permet als administradors del sistema donar d'alta, modificar, eliminar i llistar als usuaris que tindran accés a l'aplicació.
Descripció	És el cas d'us principal on l'administrador du a terme les tasques de gestió.
Actors	Administrador.
Precondició	L'usuari és l'Administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Manteniment.
Flux d'events	-L'usuari es valida com a administrador. -L'usuari pot accedir a les funcions de gestió d'usuaris.
Flux Alternatiu	-L'usuari no es valida correctament, no pot accedir a les funcions. -L'usuari no introdueix les dades correctament, no es validarà la gestió.

6.2. Casos d'ús Subsistema Funcions

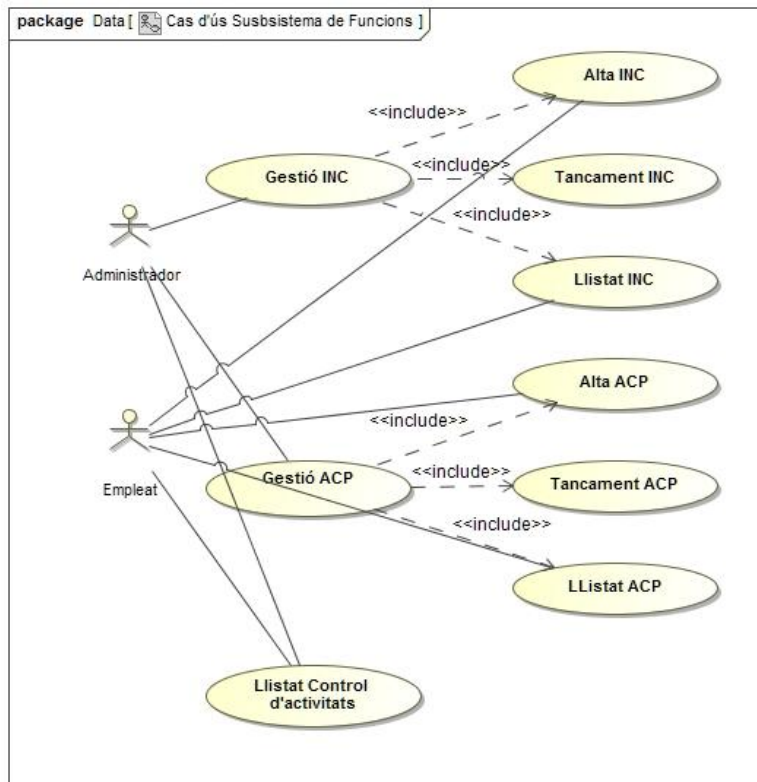


Figura 3: Casos d'ús Subsistema Funcions

6.2.1. Casos d'ús de la Gestió d'Informes de No Conformitat

Resum de la funcionalitat general	Permet als administradors del sistema donar d'alta, tancar i llistar els INC.
Descripció	És el cas d'us principal on l'administrador du a terme les tasques de gestió.
Actors	Administrador.
Precondició	L'usuari és l'Administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	-L'usuari es valida com a administrador. -L'usuari pot accedir a les funcions de gestió d'INC.
Flux Alternatiu	-L'usuari no es valida correctament, no pot accedir a les funcions. -L'usuari no introdueix les dades correctament, no es validarà la gestió.

6.2.2. Cas d'ús alta Informe de No Conformitat

Resum de la funcionalitat general	Permet als usuaris donar d'alta un INC.
Descripció	Els usuaris creen els INC per què l'administrador el processi.
Actors	Empleat, administrador
Precondició	L'usuari és un s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida al sistema -L'usuari pot accedir a la funció d'alta INC. -S'obre una finestra amb caselles per omplir les dades d'alta INC. -Un cop validades les dades es modifica la BBDD.
Flux Alternatiu	<ul style="list-style-type: none"> -L'usuari no es valida correctament, no pot accedir a la funció alta INC. -L'usuari no introdueix les dades correctament, no es validarà l'alta INC.

6.2.3. Cas d'ús llistar Informes de No Conformitat

Resum de la funcionalitat general	Permet als usuaris llistar els INC.
Descripció	Els usuaris poden veure el llistat d'INC.
Actors	Empleat, administrador.
Precondició	L'usuari s'ha validat al sistema de forma correcte.
Postcondició	-
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida. -L'usuari pot accedir a la funció llistat INC.
Flux Alternatiu	-L'usuari no es valida correctament, no pot accedir a la funció llistar INC.

6.2.4. Cas d'ús tancar Informe de No Conformitat

Resum de la funcionalitat general	Permet als administradors del sistema tancar els INC.
Descripció	L'administrador accedeix a un INC ja generat i el tanca.
Actors	Administrador.
Precondició	L'usuari és l'Administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida com a administrador. -Accedeix a la funció tancar INC. -S'obre una finestra amb la llista d'INC pendents de tancar l'administrador seleccionarà una. -S'obre una finestra amb la INC per completar. -Un cop validades les dades es modificarà a la BBDD.
Flux Alternatiu	<ul style="list-style-type: none"> -L'usuari no es valida correctament, no pot accedir a la funció tancar INC. -L'usuari no introdueix les dades correctament, no es validarà el tancament d'INC.

6.2.5. Cas d'ús de la Gestió d'Accions Correctives/Preventives

Resum de la funcionalitat general	Permet als administradors del sistema donar d'alta, tancar i llistar les ACP.
Descripció	És el cas d'us principal on l'administrador du a terme les tasques de gestió.
Actors	Administrador.
Precondició	L'usuari és l'Administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida com a administrador. -L'usuari pot accedir a les funcions de gestió d'ACP.
Flux Alternatiu	<ul style="list-style-type: none"> -L'usuari no es valida correctament, no pot accedir a les funcions. -L'usuari no introdueix les dades correctament, no es validarà la gestió.

6.2.6. Cas d'ús alta Acció Correctiva/Preventiva

Resum de la funcionalitat general	Permet a l'administrador donar d'alta una ACP.
Descripció	L'administrador crea les ACP per processar-les mes endavant.
Actors	Administrador
Precondició	L'usuari és un administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida al sistema -L'usuari pot accedir a la funció d'alta ACP. -S'obre una finestra amb caselles per omplir les dades d'alta ACP. -Un cop validades les dades es modifica la BBDD.
Flux Alternatiu	<ul style="list-style-type: none"> -L'usuari no es valida correctament, no pot accedir a la funció alta ACP. -L'usuari no introdueix les dades correctament, no es validará l'alta ACP.

6.2.7. Cas d'ús llistar Accions Correctives/Preventives

Resum de la funcionalitat general	Permet als usuaris llistar les ACP.
Descripció	Els usuaris poden veure el llistat d'ACP.
Actors	Empleat, administrador.
Precondició	L'usuari s'ha validat al sistema de forma correcte.
Postcondició	-
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida. -L'usuari pot accedir a la funció llistat ACP.
Flux Alternatiu	-L'usuari no es valida correctament, no pot accedir a la funció llistar ACP.

6.2.8. Cas d'ús tancar Acció Correctiva/Preventiva

Resum de la funcionalitat general	Permet als administradors del sistema tancar les ACP.
Descripció	L'administrador accedeix a una ACP ja generada i la tanca.
Actors	Administrador.
Precondició	L'usuari és l'Administrador i s'ha validat al sistema de forma correcte.
Postcondició	Si es compleixen les validacions dels sistema, els canvis realitzats es transmeten a la base de dades.
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida com a administrador. -Accedeix a la funció tancar ACP. -S'obre una finestra amb la llista d'ACP pendents de tancar l'administrador seleccionarà una. -S'obre una finestra amb la ACP per completar. -Un cop validades les dades es modificarà a la BBDD.
Flux Alternatiu	<ul style="list-style-type: none"> -L'usuari no es valida correctament, no pot accedir a la funció tancar ACP. -L'usuari no introdueix les dades correctament, no es validarà el tancament de l'ACP.

6.2.9. Cas d'ús llistar Control d'activitats

Resum de la funcionalitat general	Permet als usuaris llistar el Planning de Control d'activitats
Descripció	Els usuaris poden veure el llistat de Control d'Activitats.
Actors	Empleat, administrador.
Precondició	L'usuari s'ha validat al sistema de forma correcte.
Postcondició	-
Àrea	Subsistema de Funcions.
Flux d'events	<ul style="list-style-type: none"> -L'usuari es valida. -L'usuari pot accedir a la funció llistat de Control d'Activitats.
Flux Alternatiu	-L'usuari no es valida correctament, no pot accedir a la funció llistar Planning de Control d'Activitats.

6.3. Casos d'ús Subsistema de Connexió

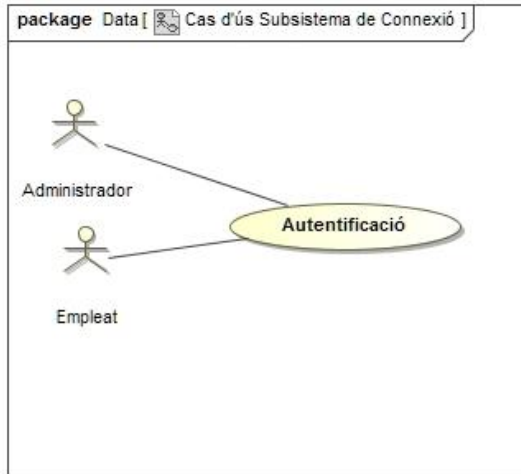


Figura 4: Casos d'ús Subsistema Connexió

6.3.1. Cas d'ús Autenticació

Resum de la funcionalitat general	Permet als usuaris autenticar-se per entrar al sistema
Descripció	L'usuari introdueix el nom i la clau de pas per accedir al sistema, un cop autenticat pot accedir al sistema.
Actors	Empleat, administrador.
Precondició	-
Postcondició	Si es compleixen les validacions dels sistema, l'usuari accedir al sistema.
Àrea	Subsistema de Connexió.
Flux d'events	-S'obre la finestra d'autenticació. -L'usuari introdueix el nom d'usuari i la clau de pas. -Si les dades son correctes l'usuari podrà entrar al sistema.
Flux Alternatiu	-L'usuari no introdueix les dades correctes. -L'usuari no podrà accedir al sistema.

7. Disseny

7.1. Arquitectura; Vista, Negoci, Persistència

Per al desenvolupament d'aquesta aplicació s'utilitza la plataforma J2EE per tal de poder llençar l'aplicació a la intranet, de tal manera que els empleats puguin accedir des del seu lloc de treball, utilitzarem la lògica de programació per capes. Per tal de tenir ben diferenciats els elements de l'aplicació, utilitzarem el patró de disseny MVC indica que s'han d'establir tres components o capes en l'arquitectura, i que cadascuna d'aquestes, només es comunica amb l'adjacent, el concepte es basa en separar el model de dades de l'aplicació de la seva representació per a l'usuari i de la interacció d'aquest amb l'aplicació, mitjançant la divisió de l'aplicació en tres parts fonamentals:

El model, que conté la lògica de negoci de l'aplicació.

La vista, que mostra a l'usuari la informació que aquest necessita.

El controlador, que rep i interpreta la interacció de l'usuari, actuant sobre model i vista de manera adequada per provocar canvis d'estat en la representació interna de les dades, així com en la seva visualització.

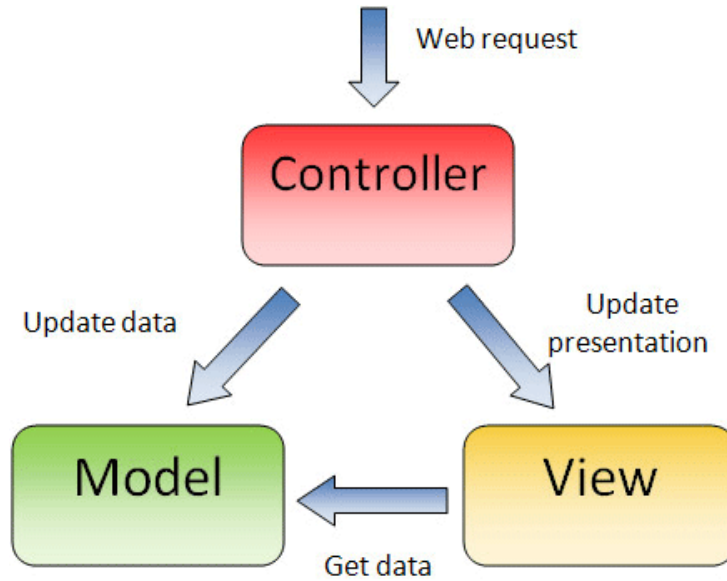


Figura 5: MVC

Presentació : La capa de Presentació està implementada amb: Java Server Pages (JSP) com la tecnologia que permet fer el desplegament de les pàgines, CSS per a la representació d'estils, javascript per a la millora d'interfície de l'usuari, jstl quatre biblioteques que amplien l'especificació de JSP.

Exemple CSS utilitzat per al disseny de la mostra dels errors:

```
<style type="text/css">
    #error { color: red; border: black; background: yellow;}
</style>
```


Exemple Javascript utilitzat per avisar a l'usuari que falten alguns camps a l'hora de fer el login:

```
<script language=JavaScript>

    function valido(form){

        if (form.j_username.value == "")

            { alert ("Introdueix un nom d'usuari"); form.j_username.value="";

              form.j_username.focus();   return false;}

        else if ((form.j_password.value == ""))

            { alert ("Introdueix una clau"); form.j_password.value="";

              return false;}

        else    return true;

    }

</script>
```

Exemple ús de les biblioteques de jstl dintre de les JSP;

```
<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>

<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
```

Controladors: Definit a dispatcherServlet.xml, per la capa de control s'utilitzarà Spring, l'objectiu central d'Spring és permetre que objectes de negoci i d'accés a dades siguin reutilitzables, no lligats a serveis J2EE específics.

Exemple viewResolver ens permetrà redirigir a la vista;

```
<bean id="viewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver"
      p:prefix="/WEB-INF/jsp/"
      p:suffix=".jsp" />
```

Exemple definició beans per l' integració entre capes;

```
<bean id="usuariBo" class="com.bo.UsuariBO">
  <property name="usuariDAO" ref="usuariDao"/>
</bean>

<bean id="usuariDao" class="com.dao.UsuariDAO">
  <property name="sessionFactory" ref="sessionFactory"/>
</bean>
```

Persistència: Bàsicament conté la lògica de negoci real, el domini de l'aplicació (VO: Value Object) amb les classes get i set, i els objectes d'accés a dades (DAO) que implementin les operacions CRU (Create, Read, Update , Delete). Aquesta capa interactua o bé directament (per exemple, mitjançant jdbc) o per mitjà de la capa de persistència, amb servidors de bases de dades. Per gestionar aquesta capa s'ha decidit usar Hibernate, és un potent mapeador objecte / relacional i servei de consultes per a Java. És la solució ORM (Object-Relational Mapping) més popular de Java.

Exemple obtenció llista d'Informes de no conformitat;

@Override

```
public List<Inc> llistaInc() {  
    return getHibernateTemplate().find("from Inc");  
}
```

7.2. Frameworks

7.2.1. Java Server Faces

JSF és un marc de treball per crear aplicacions java J2EE basades en el patró MVC de tipus 1. JSF té com característiques principals: Utilitza pàgines JSP per generar les vistes, afegint una biblioteca pròpia d'etiquetes per crear els elements dels formularis HTML. Associa cada vista amb formularis, un conjunt d'objectes java manipulats pel controlador (managed beans) que faciliten la recollida, manipulació i visualització dels valors en els diferents elements dels formularis. Introdueix una sèrie d'etapes en el processament de la petició, com per exemple la de validació, reconstrucció de la vista, recuperació dels valors dels elements, etc.

Utilitza un senzill fitxer de configuració per al controlador en format xml. És extensible, podent crear nous elements de la interfície o modificar els ja existents. I el que és més important: forma part de l'estàndard J2EE. En efecte, hi ha moltes alternatives per crear la capa de presentació i control d'una aplicació web Java, com Struts i altres frameworks, però només JSP forma part de l'estàndard.

7.2.2. Spring

Spring és un framework d'aplicacions Java/J2EE que proporciona una potent gestió de configuració basada en JavaBeans, aplicant els principis d'Inversió de Control (IOC). Aquesta factoria de beans pot ser usada en qualsevol entorn, des d'applets fins contenidors J2EE. Aquestes definicions de beans es realitzen en el que s'anomena el context d'aplicació.

Una capa genèrica d'abstracció per a la gestió de transaccions fent senzilla la demarcació de transaccions sense tractar a baix nivell. S'inclouen estratègies genèriques per JTA i un únic JDBC datasource. En contrast amb el JTA simple o EJB CMT, el suport de transaccions de Spring no està lligat a entorns J2EE.

Una capa d'abstracció JDBC que ofereix una significativa jerarquia d'excepcions (evitant la necessitat d'obtenir de SQLException els codis que cada gestor de base de dades assigna als errors), simplifica el maneig d'errors, i redueix considerablement la quantitat de codi necessari .

Integració amb Hibernate, JDO i iBatis SQL Maps en termes de suport a implementacions DAO i estratègies amb transaccions. Especial suport a Hibernate afegint convenients característiques de IOC, i solucionant molts dels comuns problemes d'integració de Hibernate. Tot això complint amb les transaccions genèriques de Spring i la jerarquia d'excepcions DAO.

Un framework MVC (Model-View-Controller), construït sobre el nucli de Spring. Aquest framework és altament configurable via interfícies i permet l'ús de múltiples tecnologies per a la capa vista com poden ser JSP, Velocity, Tiles, iText o POI. De qualsevol manera una capa model realitzada amb Spring pot ser fàcilment utilitzada amb una capa web basat en qualsevol altre framework MVC, com Struts, Webwork o Tapestry.

Tota aquesta característica es pot usar en qualsevol servidor J2EE, i la majoria ni tan sols requereix el seu ús. L'objectiu central de Spring és permetre que objectes de negoci i d'accés a dades siguin reutilitzables, no lligats a serveis J2EE específics. Aquests objectes poden ser reutilitzats tant en entorns J2EE (web o EJB), aplicacions standalone, entorns de proves, ... sense cap problema.

7.2.3. Hibernate

Hibernate és un potent mapeador objecte / relacional i servei de consultes per a Java. És la solució ORM (Object-Relational Mapping) més popular al món Java.

Hibernate permet desenvolupar classes persistents a partir de classes comuns, incloent associació, herència, polimorfisme, composició i col·leccions d'objectes. El llenguatge de consultes d'Hibernate HQL (Hibernate Query Language), dissenyat com una mínima extensió orientada a objectes de SQL, proporciona un pont elegant entre els mons objectual i relacional. Hibernate també permet expressar consultes utilitzant SQL natiu o consultes basades en criteris.

Suporta tots els sistemes gestors de bases de dades SQL i s'integra de manera elegant i sense restriccions amb els més populars servidors d'aplicacions J2EE i contenidors web.

7.2.4. Spring Security

El Servei de Seguretat té com a propòsit principal gestionar l'autenticació i l'autorització dels usuaris de les nostres aplicacions. L'objectiu de l'autenticació és comprovar que l'usuari és qui diu ser, mentre que l'autorització s'encarrega de comprovar que realment té accés al recurs sol·licitat. Per tal d'instal·lar el mòdul de seguretat s'afegeix manualment en el pom.xml de l'aplicació.

Per a configurar la font d'autorització mitjançant base de dades és necessari:

- Configurar l'arxiu de propietats **security.properties**.
- Configurar el proveïdor de seguretat dins de la configuració de seguretat de Spring.

Els dos arxius es generen i configuren de manera automàtica mitjançant l'eina de desenvolupament.

7.2.5. Entorn de desenvolupament

Per al desenvolupament de l'aplicació Euromaster-Quality, he optat per utilitzar l'IDE Netbeans 7.0.1, La plataforma NetBeans permet que les aplicacions siguin desenvolupades a partir d'un conjunt de components de programari anomenats mòduls, amb l'ajut de Maven, que és una eina per gestionar les dependències d'un projecte i per automatitzar algunes tasques del seu seguiment, Maven utilitza un Project Object Model (POM) per descriure el projecte de programari a construir, les seves dependències d'altres mòduls i components externs, i l'ordre de construcció dels elements. Ve amb objectius predefinits per realitzar certes tasques clarament definides, com la compilació del codi i l'empaquetat.

Una característica clau de Maven és que està llest per a usar en xarxa. El motor inclòs en el nucli pot dinàmicament descarregar plugins d'un dipòsit, el mateix arxiu que proveeix accés a moltes versions de diferents projectes Open Source en Java, d'Apache i altres organitzacions i desenvolupadors

7.2.6. Entorn d'execució

Pel que fa al servidor de l'aplicació utilitzaré Tomcat, un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, que compila JSPs convertint-les en servlets. El motor de servlets de Tomcat sovint es presenta en combinació amb el servidor web Apache, que serà la utilitzada.

7.3. Accés a dades

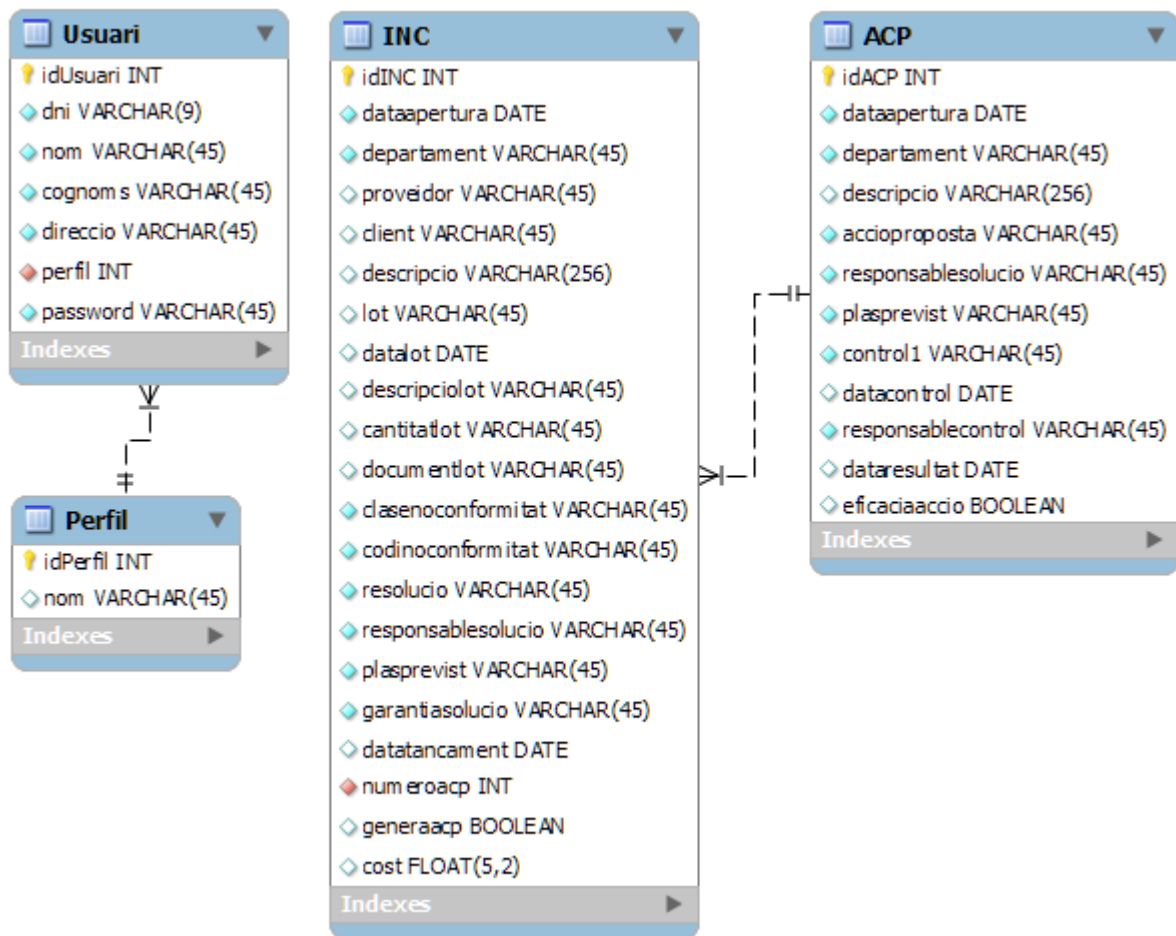


Figura 6: Model E/R

Taula usuari; conté els diferents usuaris que podran accedir a l'aplicació, aquests estan dividits en dos rols ROLE_ADMIN i ROLE_USER, el primer tindrà accés complet a tota l'aplicació actua com a administrador, el segon només podrà accedir a part de les funcionalitats de l'aplicació.

Taula acp; conté les dades emmagatzemades sobre les Accions Correctives/Preventives, tots els usuaris en podran crear, però només els usuaris amb rol administrador podran tancar-les.

Taula inc; conté les dades emmagatzemades sobre els Informes de No Conformitat, tots els usuaris en podran crear, però només els usuaris amb rol administrador podran tancar-les.

Taula perfil; llistat dels diferents perfils que poden tenir els usuaris en aquest cas, ROLE_ADMIN i ROLE_USER.

Taula garantiasol; llistat dels diferents tipus de garanties de solució que poden tenir les incs o les acps.

Taula departament; llistat de departaments que consten a l'empresa.

Taula control; conté una taula amb els diferents punts de control de qualitat que l'empresa ha d'avaluar anualment.

Taula codinoconf; llistat dels codis de no conformitat dels que disposa l'empresa.

Taula clasnoconf; llistat de les classes de no conformitat que disposa l'empresa.

8. Diagrames de classes

8.1. Model DAO

El patró de disseny Data Access Object (DAO), s'encarrega d'encapsular la interacció d'una aplicació amb la base de dades, d'aquesta manera quan la capa de lògica de negoci necessiti interactuar amb la base de dades, aquesta utilitzarà els mètodes oferts per DAO. Generalment la classe d'operacions que ofereix la capa de dades se li coneix com CRU (Create, Read, Update i Delete).

DTO s (Data Transfer Object), els quals són utilitzats per DAO per transportar les dades des de la base de dades fins a la capa de lògica de negoci o viceversa, es a dir un DTO és un objecte que té com atributs els mateixos atributs del model, amb els seus corresponents accessos (Setters i Getters).

8.2. Diagrama model DAO

En els següent diagrama es mostra la distribució de classes en diferents paquets;

- Lògica de negoci: hi hauran les classes que treballen sobre la lògica de negoci.
- DAO: inclou els objectes d'accés a la base de dades.
- DTO: Objecte intermedi que encapsula una unitat d'informació de la font de dades, es com la representació d'una taula de la base de dades.
- Base de dades: representació de la base de dades.

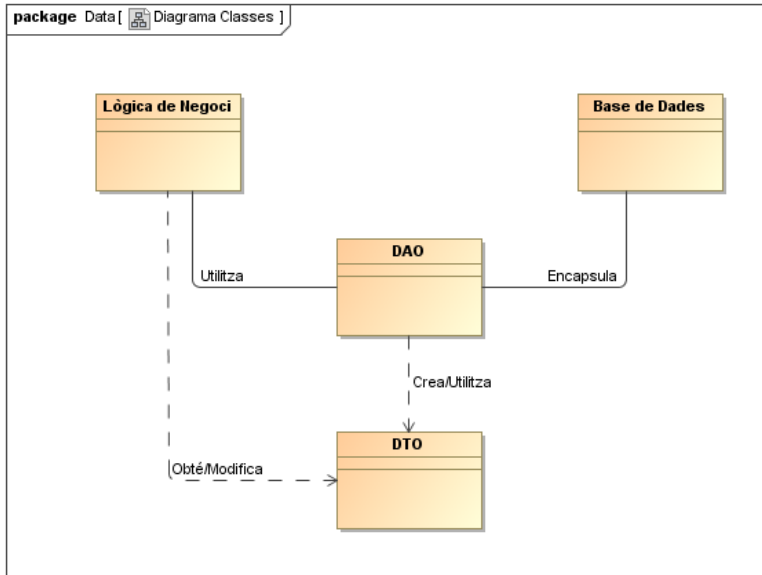


Figura 7: Diagrama Classes

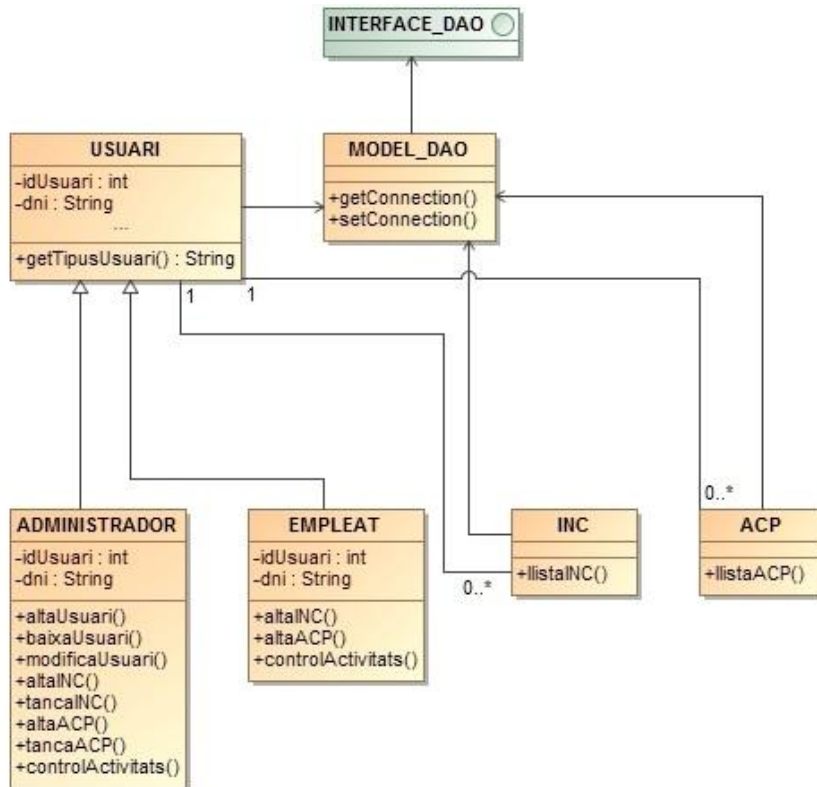


Figura 8: Diagrama DAO

9. Interfície gràfica

9.1. Funcionalitats Subsistema de Manteniment

9.1.1. Gestió d'usuaris del sistema

Alta: El procés d'alta d'un usuari al sistema constarà d'una pantalla on l'Administrador ha d'entrar les dades requerides: NIF, nom, cognoms, adreça, perfil, contrasenya. El NIF serà el camp que identificarà l'usuari al sistema de forma inequívoca, i el nom juntament amb la contrasenya seran els que serviran per a fer el 'login' al sistema. Quan l'encarregat de les altes finalitzi, haurà de prémer el botó 'Desar' per tal de desar les dades a la base de dades.

EUROMASTER

Inici Qualitat Control Activitats Control Usuaris Informació Tancar Sessió

Nov Usuari
Modifica Usuari
Esborra Usuari

Afegir Usuari

DNI (11111111A):

Nom: Cognoms:

Direcció:

Perfil: ROLE_ADMIN ▾ Clau:

Alta

Figura 9: Alta Usuari

Modificació: La Modificació d'un usuari consta dels mateixos paràmetres d'entrada que el procés d'alta d'usuaris que apareixeran inhabilitats. La pantalla de modificació de dades, inicialment mostra una llista d'usuaris, permetent seleccionar l'usuari que es desitja modificar, redireccionant a la pagina d'Alta Usuari, quan l'usuari vulgui desar les modificacions haurà de prémer el botó 'Desar' perquè els canvis tinguin efecte.



Figura 10: Modificar Usuari Llistat

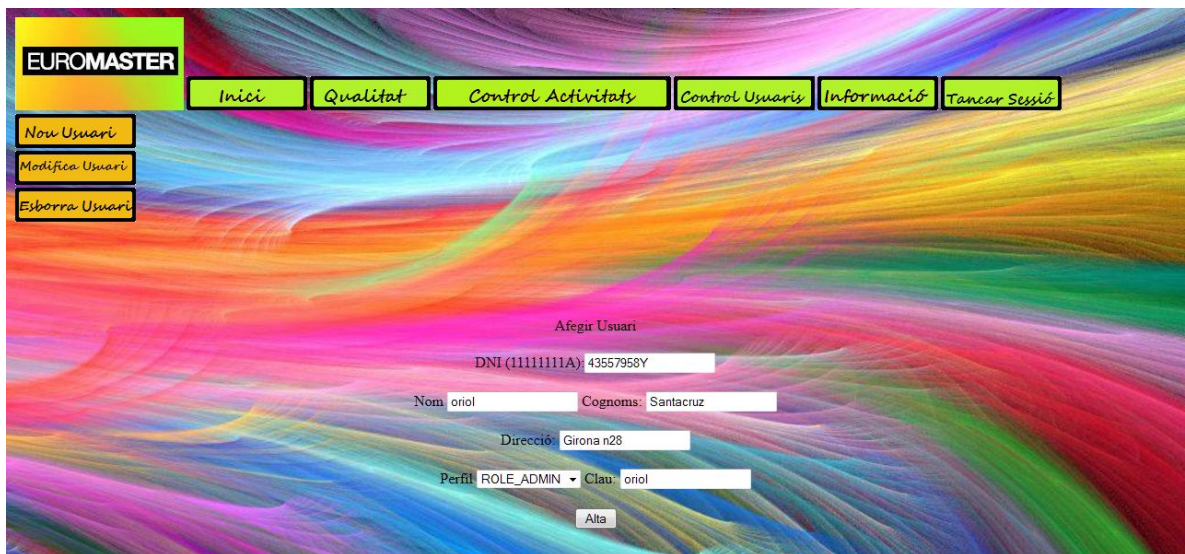


Figura 11: Modificar Usuari

Baixa: El procés de baixa d'un usuari consta en esborrar les dades de la base de dades. Inicialment es mostra una llista de la qual podem eliminar l'usuari que vulguem només pitjant el botó d'eliminar. Observem que es mostren força dades perquè l'usuari tingui el màxim d'informació a l'esborrar les dades.

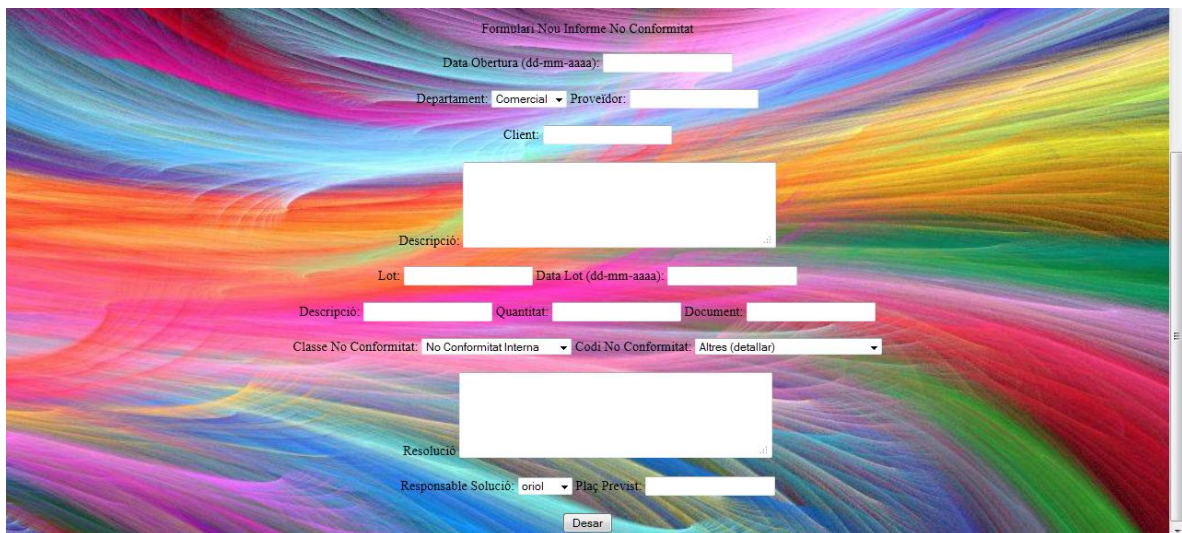


Figura 12: Eliminar Usuari

9.2. Funcionalitats Subsistema de Funcions

9.2.1. Gestió d'Informes de No Conformitat

Alta: Aquesta opció li permetrà a qualsevol usuari generar un INC, disposa d'un calendari desplegable i una llista amb els departaments a escollir, un cop s'omplen les dades es pitja el botó 'Entrar' i l'informe quedarà desat a la base de dades.



The image shows a web form titled "Formulari Nou Informe No Conformitat" set against a colorful, abstract background. The form contains the following fields and controls:

- Data Obertura (dd-mm-aaaa):** A text input field.
- Departament:** A dropdown menu with "Comercial" selected.
- Proveïdor:** A text input field.
- Client:** A text input field.
- Descripció:** A large text area.
- Lot:** A text input field.
- Data Lot (dd-mm-aaaa):** A text input field.
- Descripció:** A text input field.
- Quantitat:** A text input field.
- Document:** A text input field.
- Classe No Conformitat:** A dropdown menu with "No Conformitat Interna" selected.
- Codi No Conformitat:** A dropdown menu with "Altres (detallar)" selected.
- Resolució:** A large text area.
- Responsable Solució:** A dropdown menu with "oriol" selected.
- Plaç Previst:** A text input field.
- Desar:** A button at the bottom center.

Figura 13: Alta INC

Tancar: Un Informe de No Conformitat al ser donat d’alta només es donen part de les dades, ja que un responsable amb rol administrador ha de repassar l’INC i tancar-lo amb les dades restants, per tant per tancar un INC primer s’ha d’anar a Llistat d’INC’s on apareix una llista amb les INC que hi ha on se’n pot escollir una i redirecciona a la pàgina de tancament d’INC’s, en cas que l’Informe de No Conformitat generi una Acció Correctiva/Preventiva, al acabar de tancar l’INC hi ha l’opció de Generar ACP, la qual obre una nova pestanya amb la funció d’alta d’ACP.

Figura 14: Tancar INC

Figura 15: Tancar INC Genera ACP

Llistat: Aquesta opció permetrà a tots els usuaris veure el llistat de totes les INC que s'han anat generant.



The screenshot displays the EUROMASTER web application interface. The background is a colorful, abstract pattern. The top navigation bar includes the logo 'EUROMASTER' and several menu items: 'Inici', 'Qualitat', 'Control Activitats', 'Control Usuaris', 'Informació', and 'Tancar Sessió'. On the left side, there is a vertical menu with options: 'INC', 'Nova INC', 'Tancar INC', and 'Llistat INC'. The main content area is titled 'Formulari Nou Informe No Conformitat' and contains a table with the following data:

Id	Data Obertura	Departament	Selecció
2	2011-05-15	Comercial	Tancar
3	2011-05-15	Comercial	Tancar
4	2011-05-15	Comercial	Tancar
5	2011-05-15	Comercial	Tancar
6	2011-05-15	Comercial	Tancar
7	0021-11-01	Comercial	Tancar
8	2011-05-15	Comercial	Tancar
9	2011-05-15	Comercial	Tancar

Figura 16: Llistat INC

9.2.2. Gestió d'Accions Correctives/Preventives

Alta: Aquesta opció permetrà a qualsevol usuari introduir les dades necessàries per generar l'ACP i veure els llistats dels diferents informes que s'han generat, el tancament però només el podrà fer el responsable de qualitat.

ACP
Nov ACP
Tancar ACP
Llistat ACP

Formulari Nou Informe Acció Correctiva Preventiva

Data Obertura (dd-mm-aaaa):

Departament: Comercial ▾

Descripció:

Acció proposta:

Entrar

Figura 17: Alta ACP

Tancar: Aquesta opció li permetrà al responsable de qualitat revisar i tancar una Acció Correctiva/Preventiva, per al seu emmagatzemament a la base de dades, inicialment apareixerà un llistat on seleccionar l'ACP que es vol tancar.

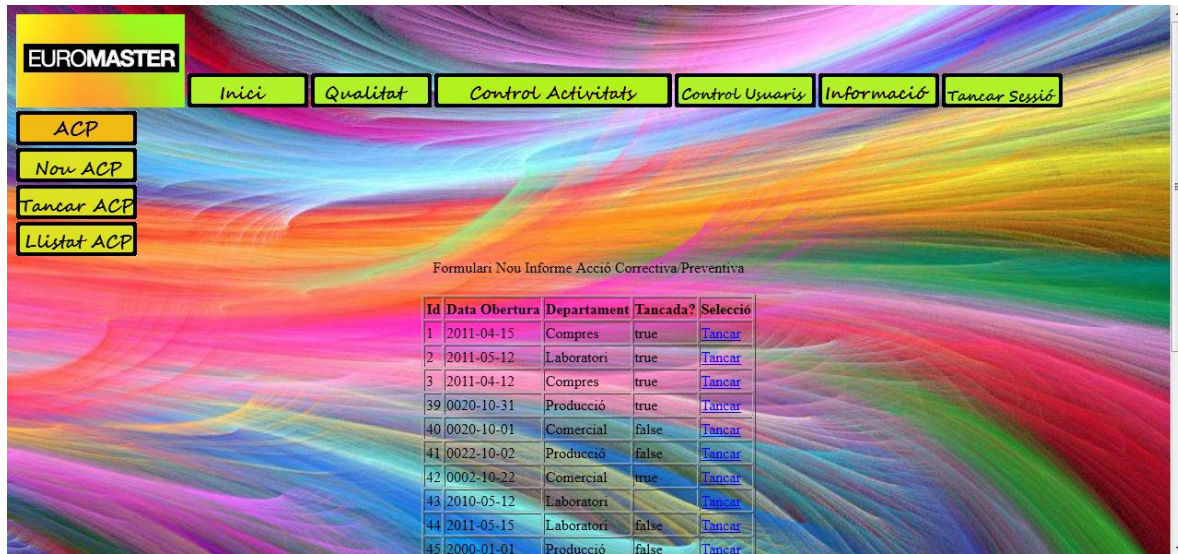


Figura 18: Tancar ACP Llistat

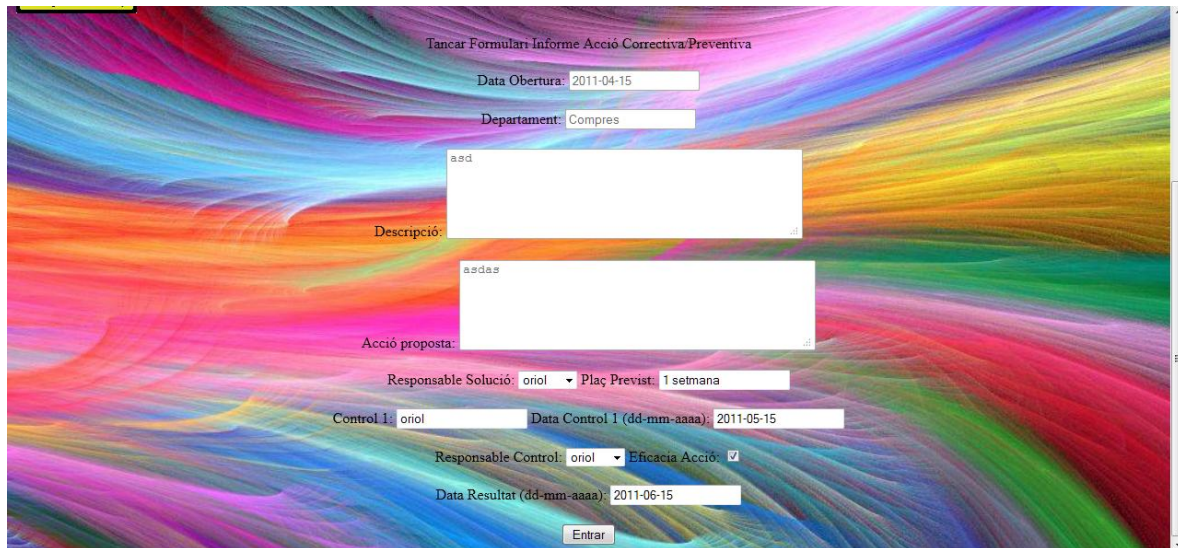


Figura 19: Tancar ACP

Llistat: Aquesta opció permetrà a tots els usuaris veure el llistat de totes les ACP que s'han anat generant.



Figura 20: Llistat ACP

9.2.3. Control d'activitats

Amb aquesta opció qualsevol usuari pot consultar la taula de control d'activitats que conté les dates en que les diferents activitats de control de l'empresa s'han de dur a terme.

The screenshot shows the EUROMASTER web application interface. At the top, there is a navigation menu with buttons for 'Inici', 'Qualitat', 'Control Activitats', 'Control Usuaris', 'Informació', and 'Tancar Sessió'. Below the menu, the title 'Taula Control Activitats De Qualitat' is displayed. The main content is a table with 13 columns representing months from January to December and 15 rows representing different quality control activities. Each cell in the table contains a small calendar icon, indicating the scheduled month for that activity.

Activitat	Gener	Febrer	Març	Abril	Maig	Juny	Juliol	Agost	Setembre	Octubre	Novembre	Desembre
Accions Correctives Preventives												
Auditoria Externa												
Auditoria Interna												
Avaluació Proveïdors												
Avaluació Satisfacció Clients												
Control de Documents												
Control de Manteniment												
Control de Producció												
Control de Registres												
Control Formació												
Control Indicadors												
Control Stock Seguretat												
Informes No Conformitat												
Revisió Controls Departament												
Revisió del Sistema												

Figura 21: Control Activitats

9.2.4. Informació

Aquesta pantalla permet a l'usuari veure l'autor del programa la versió, la data de l'última actualització...



Figura 22: Informació

9.3. Funcionalitats Subsistema Connexió

És el subsistema que executa qualsevol usuari per tal d'accedir a l'aplicació. Els usuaris s'identificaran, i en funció del seu perfil el sistema els mostrarà les opcions adequades.

Entrar: L'usuari s'identifica al sistema mitjançant un nom d'usuari i una contrasenya, enregistrats anteriorment per l'administrador.

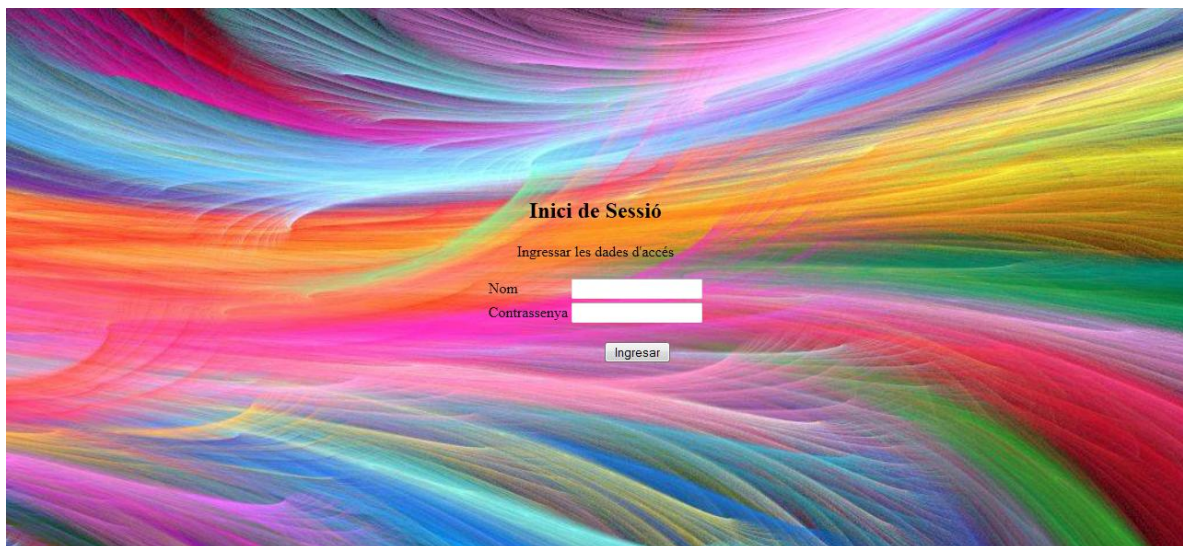


Figura 23: Inici Sessió

Sortir: Botó que permet a qualsevol usuari tancar la sessió des de qualsevol plana i tornar a la pàgina inicial on es demana el nom d'usuari i la contrasenya.



Figura 24: Tancar Sessió

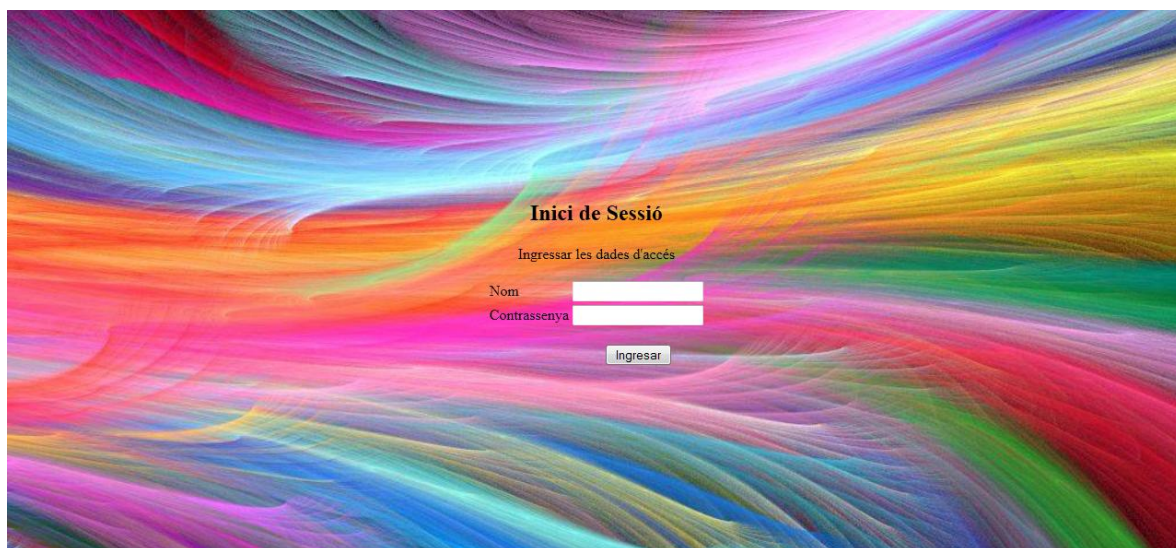


Figura 25: Obrir Sessió

9.4. Decisions de Disseny i Implementació

Per implementar el projecte he optat per crear varies carpetes amb classes que distribueixen el tipus d'accés de les dades, aquestes carpetes són;

Web; (Vista) Podem trobar els arxius jsp que descriuen el que es mostra a l'usuari i intercanvia informació amb les classes bean.

Bean: [Presentació] Carpeta que conté les classes que comuniquen amb la capa de presentació i criden a les classes bo.

Bo; (Controlador) Carpeta contenidora de les classes de la lògica de negoci, és el marc de treball d'Spring, crida a la classe dao i treballa amb les classes pojo.

Dao; [Persistència] Carpeta on es troben les classes d'accés a dades, a través d'hibernate.

/(Model) Al paquet principal es troben les classes xml que treballen directament amb hibernate, resolent la comunicació amb la base de dades.

Pojo; [Negoci] Carpeta on hi ha les classes que descriuen els objectes tant per a la base de dades com per al programa.

9.5. Relació de funcionalitats implementades

9.5.1. Subsistema de Manteniment

- **Gestió d'usuaris;**
 - **Alta, Baixa, Modificació.**

9.5.2. Subsistema de Funcions

- **Gestió Informes de No Conformitat;**
 - **Alta, Tancar, Llistat.**
- **Gestió Accions Correctives/Preventives;**
 - **Alta, Tancar, Llistat.**
- **Control d'Activitats.**
- **Informació**

9.5.3. Subsistema de Connexió

- **Autenticació**
- **Tancar sessió**

9.6. Eines utilitzades per al desenvolupament de l'aplicació

IDE: Netbeans 7.0.1 + Maven 2.0

Servidor: Tomcat 7.0

SGBD: MySQL 5.0

Navegador: Firefox 8.0.1

Capa Presentació: JSP, HTML, javascript, CSS, jstl.

Comunicació Capa de Negoci: Spring

Comunicació Capa de Dades: Hibernate

10. Conclusions finals

La creació d'aquest projecte ha sigut una recerca constant ja que la tecnologia utilitzada era totalment desconeguda per a mi, inicialment la meva intenció era desenvolupar el projecte com a aplicació d'escriptori, però finalment he decidit ampliar horitzons i aprofitar l'estudi d'aquest camp de l'informàtica.

L' utilització i integració de frameworks ha sigut difícil, però es un recurs que segur requeriré més endavant a la meva vida professional.

Finalment l'aplicació creada, encara que resten alguns punts a perfeccionar desitjo que pugui ser aprofitada, i distribuïda per al major nombre d'empreses.

11. Glossari.

Actor: Entitat externa relacionada amb un sistema. Amb unes funcionalitats concretes.

ACP: Acció Correctiva/Preventiva

BBDD: Base de Dades.

Cas d'ús: Tècnica de captura de requisits. Mostra com interactua el sistema amb l'usuari per aconseguir un objectiu concret.

Css: (Cascading style sheets) llenguatge formal utilitzat per a definir l'estil d'un document html o xml.

Dao: (Data Access Object) Patró de disseny utilitzat per a la capa de persistència.

Framework: Estructura de suport creada per a crear i organitzar altres aplicacions.

HTML: (HyperText Markup Language), llenguatge que serveix per a modelar textos i agregar funcionalitats especials, és la base de la creació de webs tradicionals.

IDE: (Integrated Development Environment), Entorn de Desenvolupament Integrat.

INC: Informe de No Conformitat.

J2EE: (Java 2 Platform, Enterprise Edition), Plataforma de Programació per desenvolupar i executar programari escrit amb Java.

JSP: (Java Server Pages) tecnologia Java que permet generar contingut dinàmic per a la web, en forma de documents HTML.

Mvc: (Model Vista Controlador) Patró de disseny de software basat en la separació de les capes vista, dades i lògica.

Servlet: És un programa que s'executa en un servidor.

SGBD: Sistema de Gestió de Bases de Dades.

Tomcat: És un contenidor de servlets i Java Server Pages.

12. Bibliografia.

<http://www.springsource.org/>

<http://www.springhispano.org/>

<http://www.hibernate.org/>

<http://www.jsptut.com/>

<http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

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

<http://www.desarrolloweb.com/html/>

<http://www.adictosaltrabajo.com/>

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

13. Annex

13.1. Creació de la Base de Dades

```
CREATE DATABASE `euromaster`;  
USE `euromaster`;
```

```
DROP TABLE IF EXISTS `clasnoconf`;  
CREATE TABLE `clasnoconf` (  
  `Clasnoconf` varchar(45) NOT NULL,  
  PRIMARY KEY (`Clasnoconf`),  
  UNIQUE KEY `ClasseNoConfrmitat_UNIQUE` (`Clasnoconf`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `codinoconf`;  
CREATE TABLE `codinoconf` (  
  `Codinoconf` varchar(45) NOT NULL,  
  PRIMARY KEY (`Codinoconf`),  
  UNIQUE KEY `descripcio_UNIQUE` (`Codinoconf`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `control`;  
CREATE TABLE `control` (  
  `Activitat` varchar(45) NOT NULL,  
  `Gener` tinyint(1) DEFAULT NULL,  
  `Febrer` tinyint(1) DEFAULT NULL,  
  `Mar` tinyint(1) DEFAULT NULL,  
  `Abril` tinyint(1) DEFAULT NULL,  
  `Maig` tinyint(1) DEFAULT NULL,  
  `Juny` tinyint(1) DEFAULT NULL,  
  `Juliol` tinyint(1) DEFAULT NULL,  
  `Agost` tinyint(1) DEFAULT NULL,  
  `Setembre` tinyint(1) DEFAULT NULL,  
  `Octubre` tinyint(1) DEFAULT NULL,  
  `Novembre` tinyint(1) DEFAULT NULL,  
  `Desembre` tinyint(1) DEFAULT NULL,  
  PRIMARY KEY (`Activitat`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `departament`;
CREATE TABLE `departament` (
  `departament` varchar(45) NOT NULL,
  PRIMARY KEY (`departament`),
  UNIQUE KEY `departament_UNIQUE` (`departament`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `garantiasol`;
CREATE TABLE `garantiasol` (
  `idgarantiasol` varchar(5) NOT NULL,
  `descripcio` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idgarantiasol`),
  UNIQUE KEY `idgarantiasol_UNIQUE` (`idgarantiasol`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `perfil`;
CREATE TABLE `perfil` (
  `perfil` varchar(45) NOT NULL,
  PRIMARY KEY (`perfil`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `usuari`;
CREATE TABLE `usuari` (
  `idUsuari` int(11) NOT NULL AUTO_INCREMENT,
  `dni` varchar(9) NOT NULL,
  `nom` varchar(45) NOT NULL,
  `cognoms` varchar(45) NOT NULL,
  `direccio` varchar(45) NOT NULL,
  `perfil` varchar(45) NOT NULL,
  `password` varchar(45) NOT NULL,
  PRIMARY KEY (`idUsuari`),
  UNIQUE KEY `idUsuari_UNIQUE` (`idUsuari`),
  UNIQUE KEY `dni_UNIQUE` (`dni`),
  KEY `perfil` (`perfil`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8;
```



```

DROP TABLE IF EXISTS `acp`;
CREATE TABLE `acp` (
  `idACP` int(11) NOT NULL AUTO_INCREMENT,
  `dataapertura` date NOT NULL,
  `departament` varchar(45) NOT NULL,
  `descripcio` varchar(256) DEFAULT NULL,
  `accioproposta` varchar(45) NOT NULL,
  `responsablesolucio` varchar(45) DEFAULT NULL,
  `plasprevist` varchar(45) DEFAULT NULL,
  `control1` varchar(45) DEFAULT NULL,
  `datacontrol` date DEFAULT NULL,
  `responsablecontrol` varchar(45) DEFAULT NULL,
  `dataresultat` date DEFAULT NULL,
  `eficaciaaccio` tinyint(1) DEFAULT NULL,
  `tancament` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`idACP`),
  UNIQUE KEY `idACP_UNIQUE` (`idACP`),
  KEY `departament` (`departament`),
  CONSTRAINT `departament` FOREIGN KEY (`departament`) REFERENCES `departament`
(`departament`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE=InnoDB DEFAULT
CHARSET=utf8;

```

```

DROP TABLE IF EXISTS `inc`;
CREATE TABLE `inc` (
  `idINC` int(11) NOT NULL AUTO_INCREMENT,
  `dataapertura` date NOT NULL,
  `departament` varchar(45) NOT NULL,
  `proveidor` varchar(45) DEFAULT NULL,
  `client` varchar(45) DEFAULT NULL,
  `descripcio` varchar(256) DEFAULT NULL,
  `lot` varchar(45) DEFAULT NULL,
  `datalot` date DEFAULT NULL,
  `descripciolot` varchar(45) DEFAULT NULL,
  `cantitatlot` varchar(45) DEFAULT NULL,
  `documentlot` varchar(45) DEFAULT NULL,
  `clasenoonconformitat` varchar(45) NOT NULL,
  `codinoconformitat` varchar(45) NOT NULL,
  `resolucio` varchar(45) NOT NULL,
  `responsablesolucio` varchar(45) NOT NULL,
  `plasprevist` varchar(45) NOT NULL,
  `garantiasolucio` varchar(45) DEFAULT NULL,
  `datatancament` date DEFAULT NULL,
  `numeroacp` int(11) DEFAULT NULL,
  `generaacp` tinyint(1) DEFAULT NULL,

```

```

`cost` float(5,2) DEFAULT NULL,
`tancament` tinyint(1) DEFAULT NULL,
PRIMARY KEY (`idINC`),
UNIQUE KEY `idINC_UNIQUE` (`idINC`),
KEY `key2` (`numeroacp`),
KEY `key1` (`departament`),
CONSTRAINT `` FOREIGN KEY (`departament`) REFERENCES `departament` (`departament`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `key2` FOREIGN KEY (`numeroacp`) REFERENCES `acp` (`idACP`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;

```

```

#--Registres perfil

```

```

INSERT INTO `euromaster`.`perfil` VALUES ('ROLE_ADMIN'),('ROLE_USER');

```

```

#--Registres garantiasol

```

```

INSERT INTO `euromaster`.`garantiasol` (`idgarantiasol`, `descripcio`) VALUES ('A',
'Segurament');
INSERT INTO `euromaster`.`garantiasol` (`idgarantiasol`, `descripcio`) VALUES ('B',
'Possiblement');
INSERT INTO `euromaster`.`garantiasol` (`idgarantiasol`, `descripcio`) VALUES ('C', 'Mai');

```

```

#--Registres departament

```

```

INSERT INTO `euromaster`.`departament` VALUES
('Comercial'),('Compres'),('Laboratori'),('Producció'),('Qualitat');

```

```

#--Registres codinoconf

```

```

INSERT INTO `euromaster`.`codinoconf` VALUES ('Altres(detallar)'), ('Assessorament tècnic
inadequat'), ('Averies'), ('Desquadrament stock'), ('Desviació de procés'), ('Desviació del color'),
('Documentació obsoleta'), ('Error de facturació'), ('Error de quantitat'), ('Error de referència'), ('Error
en qualitat del producte'), ('Error int. formula'), ('Format gra inadequat'), ('Incompliment plaç
entrega'), ('Mal estat embalatge'), ('Material contaminat'), ('Material fora de zona
establerta'), ('Material mullat'), ('No utilització de registres'), ('Reclamació
improcedent'), ('Rendiment inadequat'), ('Retard en certificats'), ('Trencament stock');

```

--Registre clasnoconf

```
INSERT INTO `euomaster`.`clasnoconf` VALUES ('No Conformitat Interna'),('No Conformitat Proveïdor'),('Reclamació Client');
```

--Resgístres usuari

```
INSERT INTO `euomaster`.`usuari` VALUES  
(1,'43557958Y','oriol','Santacruz','Giro28','ROLE_ADMIN','oriol'),  
(2,'16589965R','jordi','Santacruz','Euro','ROLE_ADMIN','jordi'),  
(3,'43557958','pep','Santacruz','girona','ROLE_ADMIN','pep'),  
(4,'53285539G','josep','alba','girona','ROLE_USER','josep');
```

13.2. Manual d'Instal·lació

13.2.1. Instal·lació programari

- Instal·lar Servidor Tomcat V7.0

- Instal·lar MySQL 5.5
 - L'usuari i clau de la sessió de MySQL ha de ser (Usuari: root, Clau: 650002830), o bé canviar l'arxiu jdbc.properties i l'arxiu hibernate.cfg.xml amb l'usuari i clau que hi hagi configurats a la BBDD que s'utilitzi.

- Generació de la base de dades
 - A la carpeta hi haurà els fitxers que contenen els scripts per: Crear la Base de Dades, l'estructura de taules i les claus, anomenat DDL-QualityEuromaster.sql
 - Inserir la informació mínima necessària per garantir l'execució de l'aplicació anomenat DML-QualityEuromaster.sql.

- Instal·lació de l'aplicació

- S'ha de copiar el war a la carpeta webapp del Tomcat.
- Un cop arranca el Tomcat, al reconèixer que hi ha un fitxer war a la carpeta webapp, el desplega i crea el conjunt de carpetes corresponents.
- Dins la carpeta conf/Catalina/localhost de Tomcat s'ha de crear un arxiu xml amb el mateix nom que l'aplicació, en aquest arxiu s'ha de copiar el que hi ha a l'arxiu applicationContext.xml de l'aplicació euromaster.

- Executar l'aplicació

- Si s'obre el manager de Tomcat, introduint la direcció següent al navegador: <http://localhost:8080/manager/html>, després d'iniciar sessió, es pot veure que s'ha creat un nou projecte amb el nom euromasterV7-1.0.war, fer doble clic, i ja es pot provar.