

Llicència

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons.

Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada.

La llicència completa es pot consultar en

<http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>

Anàlisi, disseny i implementació d'una CMDB amb Java EE

Miquel Moragues Mas
ETIS

Xavier Escudero i Sabadell

14 de Gener de 2009

Resum del projecte final de carrera

Aquest projecte final de carrera consisteix en l'anàlisi, disseny i implementació d'una **CMDB** (Configuration Management Database) que es considera la pedra angular del conjunt de bones pràctiques **ITIL** per a la gestió de serveis TI.

Es tracta, per tant, d'un projecte amb aplicacions reals en el camp de la gestió de serveis TI on, als darrers anys, ITIL s'ha convertit en l'estàndard de facto de la indústria.

Com es detalla en els diferents capítols d'aquesta memòria, l'anàlisi, disseny i implementació d'aquest projecte s'inscriu dins el paradigma de l'orientació a objectes.

La solució presentada està basada en un **arquitectura Java EE** on s'integren les tecnologies, frameworks i patrons que hem considerant més adients per a donar resposta als requeriments demanats.

L'ús de la plataforma Java EE ens ha permès implementar una aplicació multicapa escalable, d'alta disponibilitat, segura i eficient, característiques imprescindibles en qualsevol aplicació empresarial.

Com a framework de desenvolupament s'ha fet servir **Struts**, que és un framework que implementa el patró **MVC** (Model, Vista, Controlador). Juntament amb Struts, s'ha fet servir el framework **Tiles** per a facilitar la utilització de plantilles a la capa de presentació

La utilització conjunta d'aquest dos frameworks ens ha permès tenir una clara separació entre les diferents capes de l'aplicació (presentació, lògica de negoci i accés a les dades).

Amb la utilització dels components d'internalització (**i18n**) que incorpora Struts, aquesta aplicació l'hem dissenyada per a que es pugui traduir de forma senzilla a altres idiomes tot i que inicialment només està en català.

A la capa de presentació s'ha fet servir pàgines **JSP** per crear el contingut dinàmic amb la combinació d'**AJAX** per millorar l'experiència de l'usuari.

La lògica de negoci està basada en diferents **EJB 3.0** que s'encarregen d'establir les regles de negoci que s'han de complir.

Les dades resideixen en una BBDD **MySQL** i la persistència està gestionada amb la implementació que fa Hibernate del **JPA** (Java Persistence API) que facilita el mapeig objecte / relacional.

Índex de continguts

1.	INTRODUCCIÓ	7
1.1.	JUSTIFICACIÓ DEL TFC I CONTEXT EN EL QUAL ES DESENVOLUPA	7
1.2.	OBJECTIUS DEL TFC	8
1.3.	ENFOCAMENT I MÈTODE SEGUIT	8
1.4.	PLANIFICACIÓ DEL PROJECTE	9
1.4.1.	<i>Precedències de les activitats</i>	9
1.4.2.	<i>Punts de control del projecte</i>	10
1.4.3.	<i>Diagrama Gantt de la planificació</i>	11
1.5.	PRODUCTES OBTINGUTS	12
1.6.	BREU DESCRIPCIÓ DELS ALTRES CAPÍTOLS DE LA MEMÒRIA	12
2.	ESPECIFICACIÓ I ANÀLISIS DELS REQUERIMENTS	13
2.1.	INTRODUCCIÓ	13
2.2.	IDENTIFICACIÓ DELS ACTORS DEL SISTEMA	13
2.3.	IDENTIFICACIÓ DELS CASOS D'ÚS DELS ACTORS	14
2.3.1.	<i>Manteniments d'elements de la CMDB</i>	14
2.3.2.	<i>Acceptar Canvis</i>	18
2.3.3.	<i>Realitzar Canvis</i>	21
2.3.4.	<i>Solucionar incidències</i>	23
2.4.	ESPECIFICACIÓ DE LES CLASSES D'ANÀLISIS	25
2.4.1.	<i>Classes d'entitats</i>	25
2.4.2.	<i>Classes de frontera</i>	28
2.4.3.	<i>Classes de control</i>	29
2.5.	ANÀLISIS DELS USUARIS DEL SISTEMA	30
2.5.1.	<i>Perfils dels usuaris</i>	30
2.5.2.	<i>Requisits d'usabilitat</i>	30
3.	DISSENY DE L'APLICACIÓ	31
3.1.	INTRODUCCIÓ	31
3.2.	VISIÓ GENERAL DE L'ARQUITECTURA DE L'APLICACIÓ	31
3.2.1.	<i>Capa de client</i>	33
3.2.2.	<i>Capa de presentació</i>	33
3.2.3.	<i>Capa de la Lògica de Negoci</i>	33
3.2.4.	<i>Capa de Sistemes d'Informació Empresarial EIS</i>	33
3.3.	DISSENY DELS CASOS D'ÚS	34
3.4.	DIAGRAMA ESTÀTIC DE DISSENY	36
3.5.	DIAGRAMA D'ESTATS	36
3.5.1.	<i>Estats d'una incidència</i>	36
3.5.2.	<i>Estats d'un RFC</i>	37
3.5.3.	<i>Estats de HW i SW</i>	37
3.5.4.	<i>Estats dels Equips de Connexió, Servidors i Equips TC</i>	37
3.5.5.	<i>Estats de les Funcionalitats TI i les Funcionalitats TC</i>	38
3.5.6.	<i>Estats dels Serveis</i>	38
3.6.	DISSENY DE LA PERSISTÈNCIA	38
3.6.1.	<i>Model ER</i>	39
3.6.2.	<i>Atributs de les entitats i relacions del model ER</i>	39
3.7.	DIAGRAMA DE COMPONENTS	42
3.7.1.	<i>Capa web</i>	43
3.7.2.	<i>Capa EJB</i>	43
3.8.	DIAGRAMA DE DESPLEGAMENT	43
3.9.	DESCRIPCIÓ DEL PROGRAMARI NECESSARI	44
4.	CONCLUSIONS	45

5. GLOSSARI.....	46
6. BIBLIOGRAFIA	48

Índex de figures

Figura 1. Relació entre els diferents processos ITIL	8
Figura 2. Diagrama Gantt de la planificació.....	11
Figura 3. Diagrama cas d'ús Manteniment de RFCs	15
Figura 4. Diagrama cas d'ús Manteniment d'Incidències	15
Figura 5. Diagrama cas d'ús Manteniment d'Usuaris	15
Figura 6. Diagrama cas d'ús Manteniment CIs, Funcionaliats i Serveis	15
Figura 7. Pantalla per buscar o crear un element de la CMDB	17
Figura 8. Pantalla per crear o modificar un element de la CMDB	18
Figura 9. Diagrama cas d'ús Acceptar RFC.....	18
Figura 10. Pantalla resum canvis pendents d'acceptar.....	20
Figura 11. Pantalla detall canvi pendent d'acceptar.....	20
Figura 12. Diagrama cas d'us Realitzar Canvis	21
Figura 13. Pantalla de realització de canvis	22
Figura 14. Diagrama cas d'ús Solucionar Incidències	23
Figura 15. Pantalla solucionar incidències	24
Figura 16. Classe Usuaris	25
Figura 17. Classe Cis i les seves subclasses	26
Figura 18. Classes FuncionalitatTI, FuncionalitatTC i Serveis.....	27
Figura 19. Classe RFC.....	28
Figura 20. Classe Incidències	28
Figura 21. Model multicapa de l'arquitectura JEE.....	31
Figura 22. Funcionament d'Struts.....	32
Figura 23. Diagrama de seqüències del cas d'ús Modificació d'Elements de la CMDB	34
Figura 24. Diagrama de seqüències del cas d'ús Solucionar Incidències	35
Figura 25. Diagrama estàtic de disseny	36
Figura 26. Diagrama d'estats d'una incidència	36
Figura 27. Diagrama d'estats d'un RFC	37
Figura 28. Diagrama d'estats de HW i SW.....	37
Figura 29. Diagrama d'estats Equips de Connexió, Servidors i Equips TC	37
Figura 30. Diagrama d'estats de les Funcionalitats TI i les Funcionalitats TC	38
Figura 31. Diagrama d'estats dels Serveis	38
Figura 32. Model Entitat Relació	39
Figura 33. Diagrama de components de l'aplicació.....	43
Figura 34. Diagrama de desplegament de l'aplicació	44

1. Introducció

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

Les companyies depenen cada cop més de la provisió de serveis TI per donar resposta a les necessitats dels seus clients. Això implica la necessitat de serveis TI d'alta qualitat i en concordança amb les necessitats del negoci i els requeriments dels clients.

La metodologia ITIL (IT Infrastructure Library) és l'aproximació més globalment acceptada per a la gestió de serveis TI a tot el món.

ITIL proporciona un conjunt cohesionat de bones pràctiques, construïdes a partir de les experiències recollides dels sectors públic i privat a nivell internacional i suportades per un esquema global de capacitació, organitzacions acreditades per donar formació i eines d'avaluació i implementació.

L'aproximació basada en ITIL es centra en definir els processos adequats pels serveis que proveeix l'organització, proporcionant un esquema base per a la seva definició.

Els diferents processos que ITIL defineix per a la gestió de serveis TI els podem dividir en dos gran blocs:

- **Suport del Servei**, que inclou:
 - Gestió de la Configuració
 - Gestió de Incidències
 - Gestió de Problemes
 - Gestió de Versions
 - Gestió de Canvis

- **Provisió del Servei**, que inclou:
 - Gestió del nivell de Servei
 - Gestió Financera de TI
 - Gestió de Capacitat
 - Gestió de la Disponibilitat
 - Gestió de la Continuïtat dels Servei TI
 - Gestió de la Seguretat dels Serveis TI

Pel correcte funcionament de tots aquests processos és necessari que l'organització disposi d'un repositori únic amb la informació actualitzada i fiable de tots els elements del inventari en ús.

En el context d'ITIL, aquest repositori únic és la CMDB (Configuration Management Database) on es guarden totes les dades requerides per l'organització per a la producció de bens i serveis, incloent la descripció dels elements i de la seva interconnexió.

Com es veu a la figura següent, el procés de Gestió de la Configuració és el responsable de la creació i manteniment de la CMDB, assegurant així, un enllaç directe amb la resta de processos de la gestió de serveis TI.

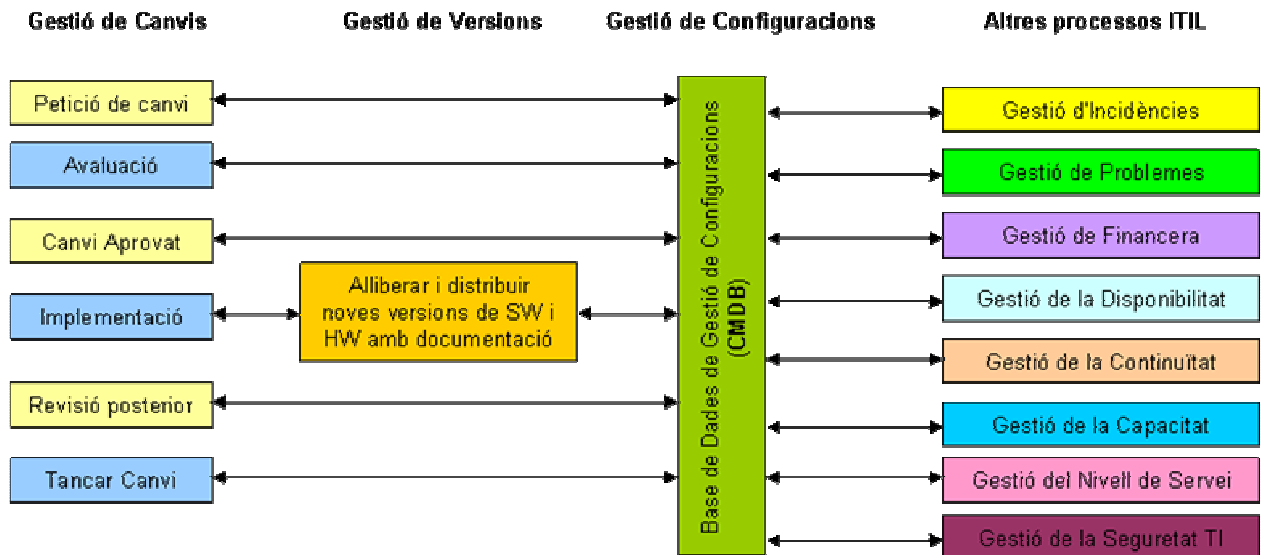


Figura 1. Relació entre els diferents processos ITIL

1.2. Objectius del TFC

Aquest TFC consisteix amb l'anàlisi, disseny i implementació d'una CMDB que doni suport als processos relacionats amb la Gestió de Configuracions, Gestió de Canvis, Gestió d'Incidències i Gestió de Versions que es defineixen a la metodologia ITIL.

Per aconseguir això, primerament s'haurà de construir una BBDD que permeti emmagatzemar els diferents elements de configuració (en anglès Configuration Item o CI) i la relació que hi ha entre ells per, seguidament, implementar el conjunt de funcionalitats que es detallen al següent apartat.

1.3. Enfocament i mètode seguit

L'enfocament utilitzat en el desenvolupament d'aquest TFC ha estat separar el projecte en dues grans fases.

A la primera fase s'ha treballat en paral·lel l'anàlisi i el disseny de l'aplicació a la vegada que s'avançava amb la corba d'aprenentatge de les tecnologies que es farien servir a la segona fase.

A la segona fase, s'ha focalitzat tots els esforços en desenvolupar l'aplicació amb les tecnologies apreses ajustant-se al disseny realitzat a la primera fase.

1.4. Planificació del projecte

A la figura 2 es pot veure el diagrama de Gantt de la planificació però abans, s'enumeren les diferents activitats, l'estimació de dies que requeriran cada una d'elles així com les seves precedències.

1.4.1. Precedències de les activitats

Codi Activitat	Nom Activitat	Estimació (dies)	Precedències
01	Inici del Projecte	0	
02	Estudi d'Oportunitat	9	
02.1	Recollida de Requeriments	6	01
02.2	Planificació del Projecte	3	02.1
03	Lliurament PAC-1	0	02.2
04	Corba d'Aprenentatge	30	
04.1	Aprenentatge Tecnologies J2EE	30	03
04.2	Instal·lació software de Base	10	04.1 + 7 dies
05	Anàlisi de Requeriments	15	
05.1	Identificació Actors i Casos d'ús	3	02.2
05.2	Identificació classes d'Entitat	6	05.1
05.3	Identificació classes Frontera	6	05.1
05.4	Identificació classes de Control	6	05.1
05.5	Especificació formal dels casos d'ús	3	05.2; 05.3; 05.4
05.6	Anàlisi Interfície d'Usuari	3	05.5
06	Disseny	20	
06.1	Disseny dels casos d'ús	5	05.6
06.2	Diagrama estàtic de disseny	5	06.1
06.3	Disseny de la persistència	5	06.2
06.4	Disseny interfície gràfica d'usuari	5	06.3
07	Lliurament PAC-2	0	06.4
08	Implementació i Proves	42	
08.1	Implementació Mòduls Bàsics	14	04.1; 04.2; 07
08.2	Integració i proves Mòduls Bàsics	5	08.1
08.3	Versió Pre-Alfa	0	08.2
08.4	Implementació Mòduls Secundaris	14	08.3
08.5	Integració i proves Mòduls Secundaris	5	08.4
08.6	Versió Alfa	0	08.5
08.7	Proves i modificacions Versió Alfa	4	08.6
08.8	Versió Beta	0	08.7
09	Lliurament PAC-3	0	08.8
10	Transició	28	
10.1	Revisió Global i modificacions finals	14	08.8
10.2	Versió definitiva	0	10.1
10.3	Elaboració presentació final	14	10.2
11	Lliurament Final	0	10.3

1.4.2. Punts de control del projecte

Aquest projecte té una durada de 18 setmanes i és possible que es produeixin desviacions sobre la planificació.

Per tal de poder mesurar aquestes possibles desviacions i prendre les mesures oportunes per corregir-les, s'han definit els següents punts de control.

Data	Punt de Control	Observacions
22/9/08	Inici Projecte	Kick off del projecte
1/10/08	PAC-1	Lliurament de la planificació del projecte
5/11/08	PAC-2	Lliurament de l'Anàlisi i del Disseny del projecte
23/11/08	Versió Pre-Alfa	Versió que només tindrà funcionalitats bàsiques
12/12/08	Versió Alfa	Versió no estable però amb al que es podran fer proves
16/12/08	Versió Beta	Versió amb totes les funcionalitats operatives
17/12/08	PAC-3	Lliurament versió Beta i documentació provisional
30/12/08	Versió Definitiva	Versió totalment operativa i comprovada
14/1/09	Lliurament Final	Lliurament versió definitiva i documentació final

1.4.3. Diagrama Gantt de la planificació

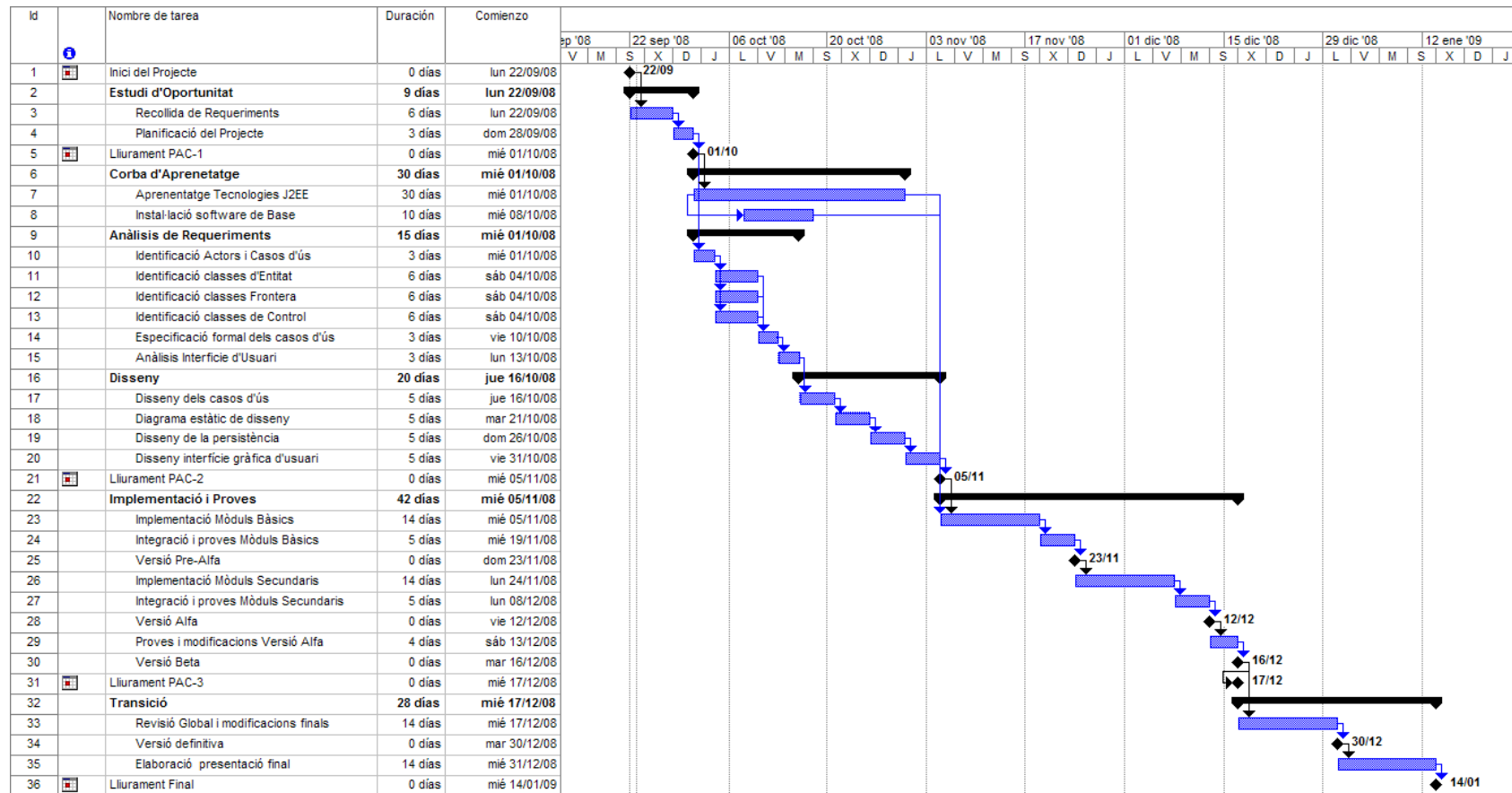


Figura 2. Diagrama Gantt de la planificació

1.5. Productes obtinguts

El producte obtingut en la implementació d'aquest projecte ha estat una aplicació web realitzada amb tecnologia JEE.

Es descompon en els següent elements:

- Fitxer de distribució .ear que inclou el .war de la capa web i el .jar de les classes de la lògica de negoci
- El codi font de totes les classes utilitzades
- Documentació detallada i normalitzada en format javadoc de totes les classes
- El fitxers de creació de les taules i la inserció de dades necessàries a la BD a utilitzar
- Manual d'instal·lació i execució de l'aplicació
- Memòria del TFC
- Presentació en PPT del TFC

1.6. Breu descripció dels altres capítols de la memòria

En els següents capítols es detallaran les diferents fases del desenvolupament d'aquest TFC (anàlisi, disseny i implementació) i acabarem amb un capítol amb les conclusions més importants que hem extret de la realització d'aquest projecte.

2. Especificació i anàlisi dels requeriments

2.1. Introducció

A la fase de recollida de requeriments s'han establert uns requeriments que ens han servit per elaborar el pla de treball però, al estar expressats d'una forma poc formalitzada, no són una bona base per construir l'aplicació.

A la fase d'anàlisi de requeriments traduirem els requeriments de la fase anterior a un llenguatge més formal fent servir els models i diagrames d'UML.

A més, identificarem les classes fonamentals que seran la base de la implementació del programari i expressarem els diferents casos d'ús en termes d'aquestes classes

2.2. Identificació dels actors del sistema

La finalitat de qualsevol programari és proporcionar i/o rebre informació d'un conjunt d'entitats exteriors (persones, màquines, programaris de l'exterior, etc.)

Un actor del sistema és un conjunt de papers d'una entitat exterior en relació amb el sistema de programari considerat.

A partir d'aquesta definició, s'han detectat sis tipus diferents d'actors que poden interactuar amb la nostra aplicació:

- **Administrador del sistema:** L'administrador del sistema és l'encarregat de gestionar els usuaris de l'aplicació i els seus privilegis.
La seva interacció amb el sistema consisteix en donar d'alta, baixa i modificar les dades i permisos dels usuaris de l'aplicació
- **Gestor de Configuracions:** La seva responsabilitat és assegurar-se que l'inventari dels elements de configuració està actualitzada i que les dades són correctes.
La seva interacció amb el sistema consisteix donar d'alta, baixa i modificar dades administratives (nom, nº de sèrie, dates d'expiració de les llicències, etc...) dels diferents elements de la CMDB
- **Gestor de Canvis:** La seva responsabilitat és tenir control de tots els canvis tècnics que es realitzen i obtenir l'autorització dels responsables dels serveis que es vegin afectats per aquests canvis.
La seva interacció amb el sistema consisteix en obrir i tancar les peticions de canvis o RFC (Request For Change), informar als diferents responsables dels serveis afectats i obtenir la seva autorització

- **Gestor d'Incidències:** La seva responsabilitat és determinar l'afectació sobre els processos de negoci de les incidències que es produeixin.
La seva interacció amb el sistema consisteix en obrir i tancar incidències i informar als diferents responsables dels serveis afectats
- **Tècnic:** La seva responsabilitat és realitzar els canvis aprovats i solucionar les incidències que vagin sorgint
La seva interacció amb el sistema consisteix en informar de les tasques que va realitzant per implementar els canvis o solucionar les incidències.
- **Responsable del Servei:** Aquest actor té la visió de negoci dels serveis dels quals és responsable. És l'encarregat d'assegurar que els canvis afecten el menys possible al negoci i té potestat per modificar les finestres d'actuació proposades pel gestor de canvis.
La seva interacció amb el sistema consisteix amb acceptar o denegar els canvis que afecten al seus serveis i tenir constància de les incidències en el hardware o software que en un moment donat poden estar afectant als seus serveis

2.3. Identificació dels casos d'ús dels actors

Un cas d'ús documenta una interacció entre el programari i un actor o més. En principi, la interacció en qüestió ha de ser una funció autònoma dins el programa.

A continuació es mostren els diferents diagrames de casos d'ús, la seva fitxa i la pantalla associada a cada cas.

Aquests diagrames i fitxes ens ajudaran a mostrar les funcions del sistema des del punt de vista de les seves interaccions amb l'exterior però sense entrar ni en la descripció detallada ni en la implementació d'aquestes funcions.

2.3.1. Manteniments d'elements de la CMDB

Per la seva similitud hem agrupat tots els casos d'ús de manteniment d'elements de la CMDB en un únic apartat.

Aquests elements són RFCs, Incidències, Usuaris, Configuration Items (CIs), Funcionalitats i Serveis.

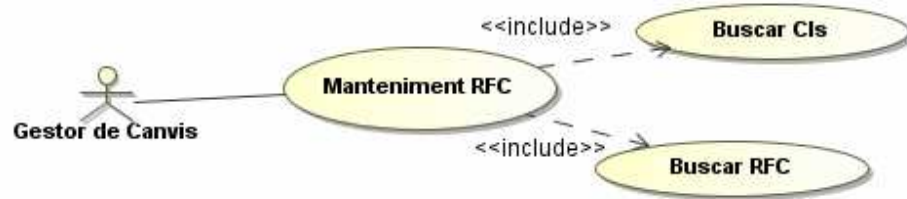


Figura 3. Diagrama cas d'ús Manteniment de RFCs

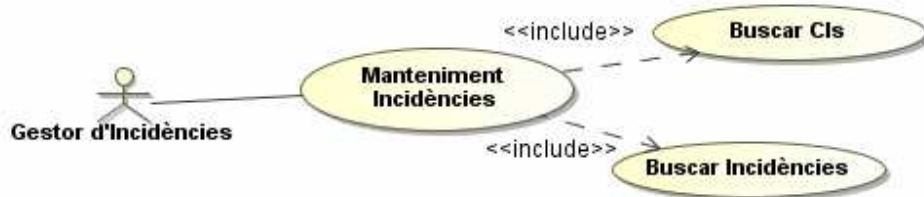


Figura 4. Diagrama cas d'ús Manteniment d'Incidències



Figura 5. Diagrama cas d'ús Manteniment d'Usuaris

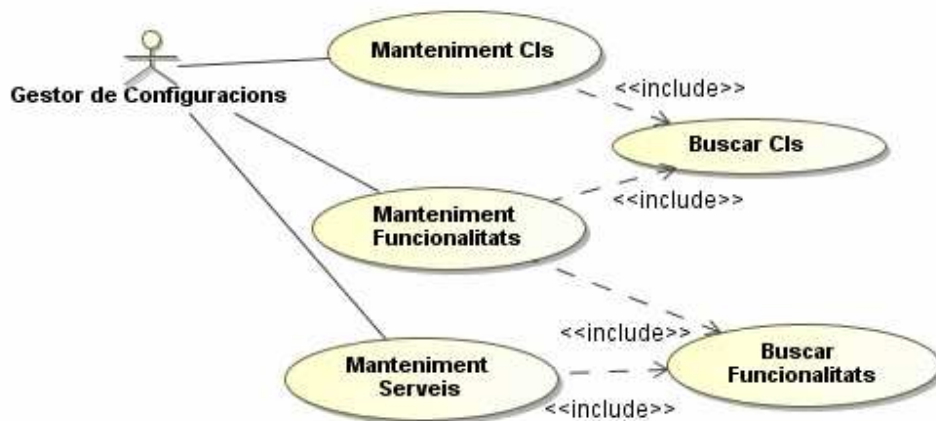


Figura 6. Diagrama cas d'ús Manteniment Cls, Funcionalitats i Serveis

CAS D'ÚS		Manteniment d'elements de la CMDB	
Versió	1.0	Data	Octubre 2008
Autors	MM1		
Descripció	Els actors podran donar d'alta, baixa i modificar els diferents elements de la CMDB		
Actors	Gestor de Canvis, Gestor d'Incidències, Gestor de Configuracions i Administrador del Sistema		
Precondició	L'usuari ha d'estar loginat a l'aplicació i tenir el rol amb permisos per fer el manteniment de l'element que pertoqui. Els elements relacionats amb l'element que s'està fent el manteniment han d'estar donats d'alta a la CMDB		
Flux principal	El cas comença quan l'usuari selecciona el rol oportú del menú principal. El sistema dona l'opció a l'usuari de buscar un element a partir d'algun dels seus atributs o bé de crear-ne un de nou.		
Subfluxos	<p><u>Alta</u></p> <ol style="list-style-type: none"> 1. L'usuari selecciona l'opció de crear un nou element 2. El sistema mostra un formulari on se li demanen les dades administratives pròpies de l'element que dona d'alta així com les dades d'altres elements amb els que pot estar relacionat que són: <ul style="list-style-type: none"> • CIs en el cas del manteniment de RFCs, manteniment d'Incidències, manteniment de CIs i manteniment de Funcionalitats • Funcionalitats en el cas de manteniment de Serveis i manteniment de Funcionalitats • Serveis en el cas de manteniment d'Usuaris 3. L'usuari omple les dades necessàries per crear l'element. Una vegada acabat, l'usuari prem "Crear" 4. El sistema enregistra l'element creat i les seves relacions <p><u>Modificació</u></p> <ol style="list-style-type: none"> 1. L'usuari introdueix la dada amb la que vol buscar l'element i prem el boto "Buscar" 2. El sistema busca i mostra els elements que compleixen les condicions del punt anterior. Si no n'existeix cap es mostra un missatge d'error 3. Si la cerca ha retornat un únic element, es mostren les dades d'aquest element. Si n'ha retorna més d'un, es mostra el codi identificatiu i la descripció de cada 		

CAS D'ÚS	Manteniment d'elements de la CMDB
	<p>un dels elements. L'usuari prem sobre un d'aquests elements i el sistema mostra les seves dades</p> <ol style="list-style-type: none"> L'usuari pot modificar les dades de l'element igual com ho fa al cas d'ús d'Alta. Una vegada modificats, l'usuari prem "Acceptar" El sistema enregistra l'element modificat <p><u>Cancel·lar</u></p> <p>En qualsevol moment l'usuari pot cancel·lar les accions realitzades. En aquest cas el sistema no guarda cap acció realitzada</p>
Fluxos alternatius	<p><u>Element no trobat</u></p> <p>Si el subflux "Modificació" no troba cap element es mostra un missatge d'error. L'usuari l'accepta i el cas d'ús es reprèn</p>
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

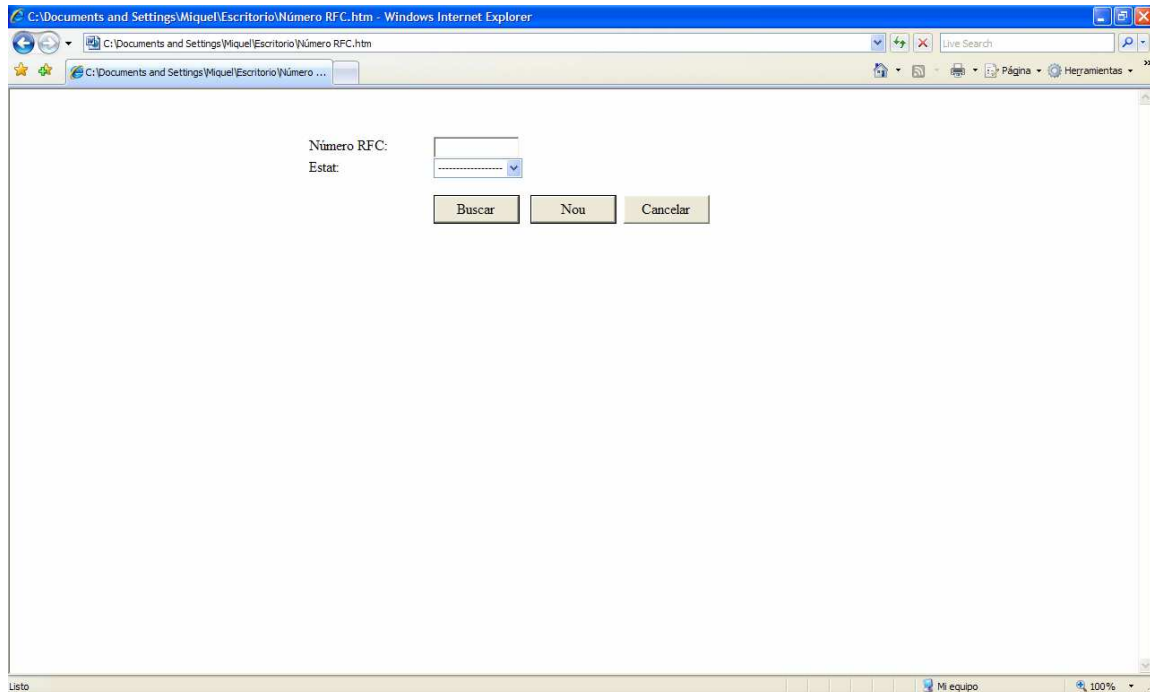


Figura 7. Pantalla per buscar o crear un element de la CMDB

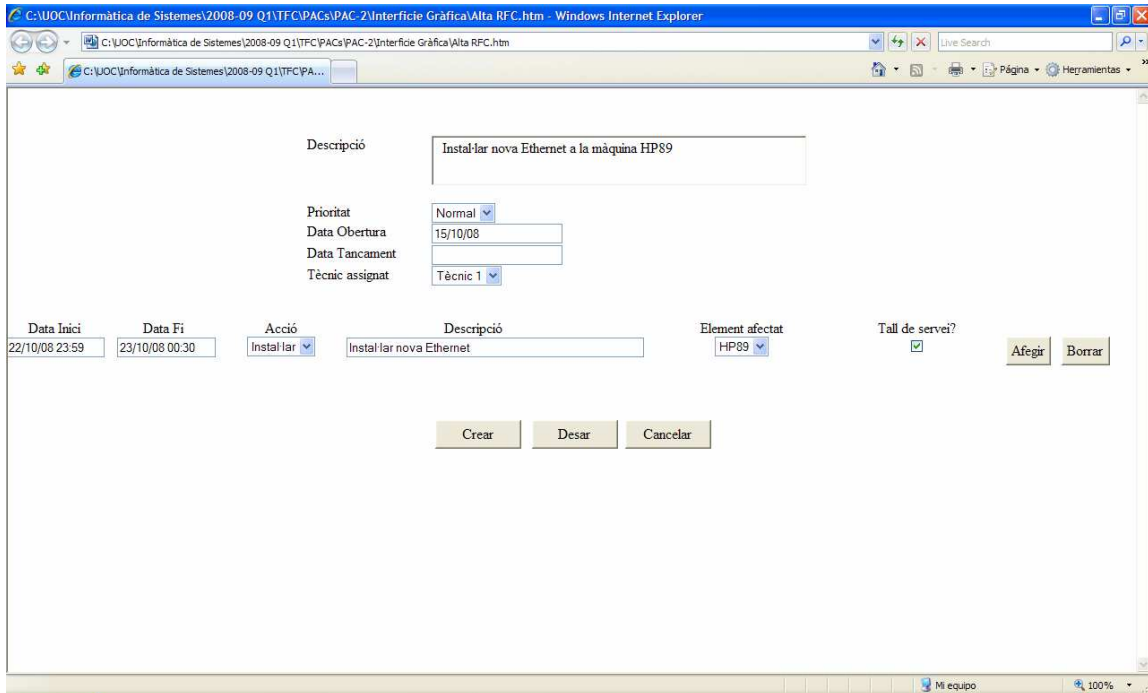


Figura 8. Pantalla per crear o modificar un element de la CMDB

2.3.2. Acceptar Canvis



Figura 9. Diagrama cas d'ús Acceptar RFC

CAS D'ÚS		Acceptar canvis	
Versió	1.0	Data	Octubre 2008
Autors	MM1		
Descripció	El Responsable del Servei podrà acceptar o denegar un RFC		
Actors	Responsable del Servei		
Precondició	L'usuari s'ha d'estar loginat a l'aplicació i tenir el rol de Responsable del Servei		
Flux principal	El cas comença quan l'usuari selecciona el rol oportú del menú principal. Per a cada un dels RFCs que l'usuari té pendents d'acceptar, el sistema li mostra el nº del RFC, la descripció, l'hora d'inici i fi i l'afectació que tindrà aquest canvi. Al costat de cada RFC hi haurà dos botons, un per acceptar i un per denegar el canvi.		

CAS D'ÚS	Acceptar canvis
Subfluxos	<p><u>Acceptar</u></p> <ol style="list-style-type: none"> 1. L'usuari prem el botó d'acceptar el RFC 2. El sistema mostra un espai per si l'usuari vol escriure alguna anotació (és opcional) 3. L'usuari prem el botó d'acceptar 4. El sistema guarda el RFC amb l'estat Acceptat <p><u>Denegar</u></p> <ol style="list-style-type: none"> 1. L'usuari prem el botó de denegar el canvi 2. El sistema mostra un espai perquè l'usuari escrigui el motiu de la denegació (és obligatori) 3. Una vegada ha escrit el motiu de la denegació, l'usuari prem el botó d'acceptar 4. El sistema guarda el RFC amb l'estat denegat <p><u>Veure detalls del RFC</u></p> <ol style="list-style-type: none"> 1. L'usuari prem sobre un dels RFCs 2. El sistema mostra els detalls del RFC 3. L'usuari prem el botó OK <p><u>Cancel·lar</u></p> <p>En qualsevol moment l'usuari pot cancel·lar les accions realitzades. En aquest cas el sistema no guarda cap acció realitzada</p>
Fluxos alternatius	
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

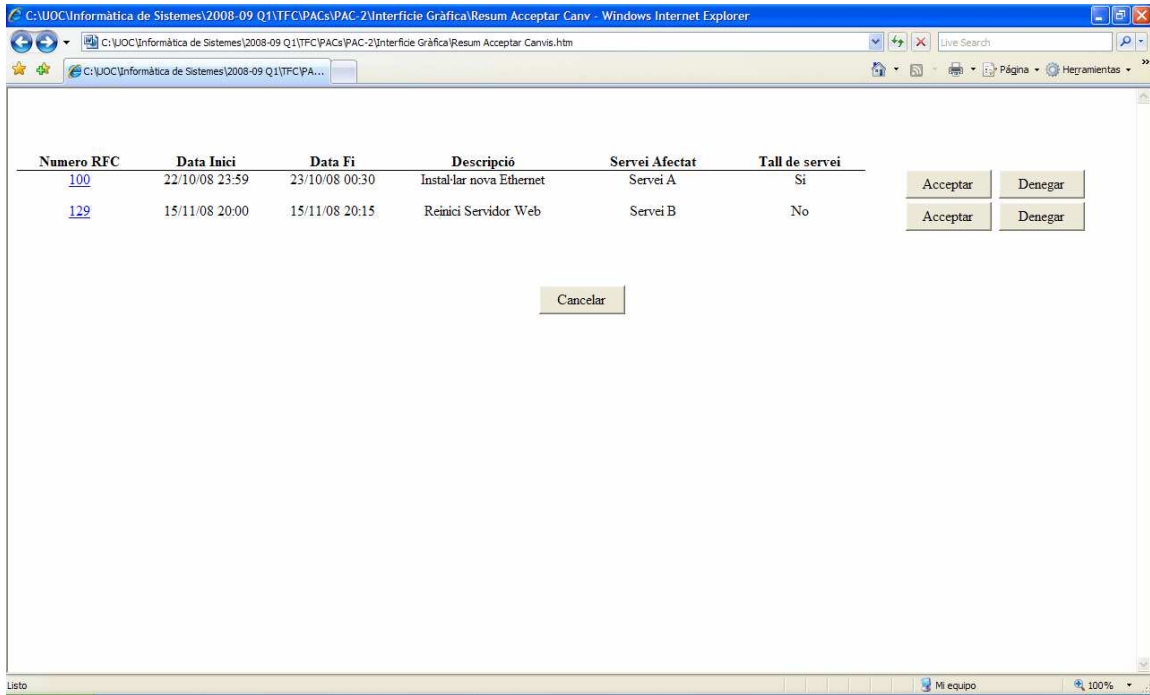


Figura 10. Pantalla resum canvis pendents d'acceptar

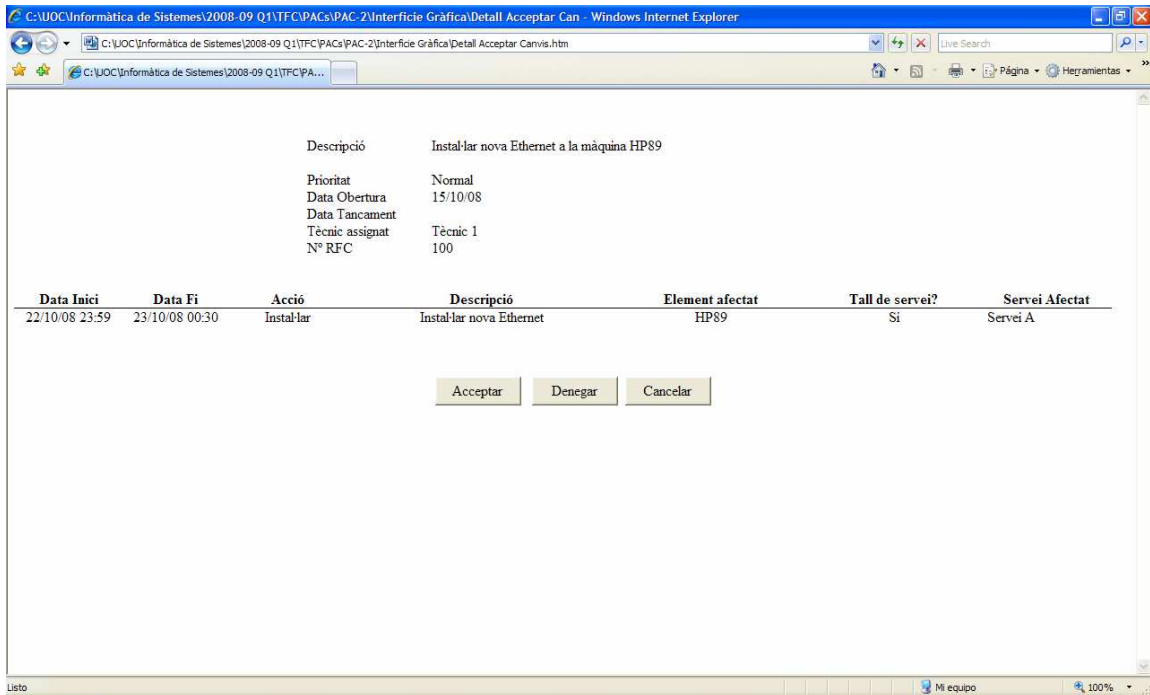


Figura 11. Pantalla detall canvi pendent d'acceptar

2.3.3. Realitzar Canvis



Figura 12. Diagrama cas d'us Realitzar Canvis

CAS D'ÚS		Realitzar canvis	
Versió	1.0	Data	Octubre 2008
Autors	MM1		
Descripció	El Tècnic realitza les tasques que indica el RFC i anota l'hora d'inici i fi reals		
Actors	Tècnic		
Precondició	L'usuari s'ha d'estar loginat a l'aplicació i tenir el rol deTècnic		
Flux principal	<p>El cas comença quan l'usuari selecciona el rol oportú del menú principal. Per a cada un dels RFCs que l'usuari té pendents de realitzar, el sistema li mostra el nº del RFC, la descripció, l'hora d'inici i fi i l'afectació que tindrà aquest canvi.</p> <p>L'usuari prem sobre un dels RFCs per veure el detall de les tasques a realitzar. Al costat de cada tasca hi ha una casella per marca que s'ha realitzat i una altra per posar el dia i hora que s'ha realitzat</p>		
Subfluxos	<p><u>Realitzar</u></p> <ol style="list-style-type: none"> 1. L'usuari va marcant les accions que va realitzant i indica en quin dia i hora l'ha realitzat 2. Una vegada ha realitzat totes les tasques, l'usuari prem al botó "Fet" 3. Si alguna tasca no s'ha marcat com a realitzada, es mostra un missatge d'error que l'usuari ha d'acceptar. Si totes les tasques s'han realitzat, el sistema guarda el RFC amb l'estat realitzat <p><u>Fer marxa enrere</u></p> <ol style="list-style-type: none"> 1. Si per alguna raó l'usuari no pot realitzar el canvi tal i com s'ha determinat al RFC, desfà totes les actuacions que ha fet fins llavors i prem el botó "Marxa Enerera" 2. El sistema guarda el RFC amb l'estat marxa enrere 		

CAS D'ÚS	Realitzar canvis
	<p><u>Cancel·lar</u></p> <p>En qualsevol moment l'usuari pot cancel·lar les accions realitzades. En aquest cas el sistema no guarda cap acció realitzada</p>
Fluxos alternatius	<p><u>RFC no acabat</u></p> <p>Si en el subflux Realitzar l'usuari dona per acabat el RFC sense haver marcat totes les tasques com a realitzades, surt un missatge avisant d'aquest fet. L'usuari l'haurà d'acceptar i el cas d'ús es reprèn.</p>
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

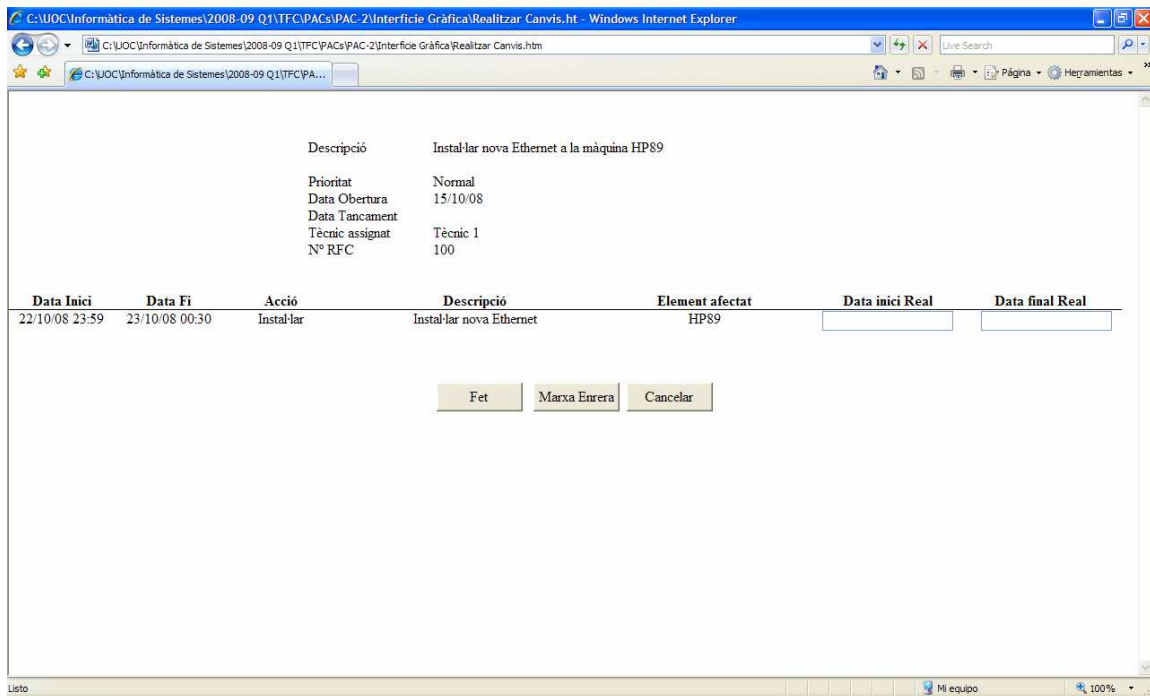


Figura 13. Pantalla de realització de canvis

2.3.4. Solucionar incidències



Figura 14. Diagrama cas d'ús Solucionar Incidències

CAS D'ÚS		Solucionar Incidències	
Versió	1.0	Data	Octubre 2008
Autors	MM1		
Descripció	El Tècnic anota les accions que va realitzant per solucionar les incidències que tracta		
Actors	Tècnic		
Precondició	L'usuari s'ha d'estar loginat a l'aplicació i tenir el rol deTècnic		
Flux principal	<p>El cas comença quan l'usuari selecciona el rol oportú del menú principal. Per a cada una de les incidències que hi ha obertes, el sistema mostra el seu nº, la descripció, l'hora d'obertura, la prioritat i quina afectació té.</p> <p>L'usuari prem sobre una de les incidències per veure el detall de la incidència i l'històric d'accions realitzades fins al moment</p>		
Subfluxos	<p><u>Afegir acció</u></p> <ol style="list-style-type: none"> L'usuari escriu al l'acció realitzada a la incidència i prem el botó "Afegeix" El sistema afegeix el dia i hora actuals, guarda l'acció i refresca l'històric d'accions que es veu per pantalla <p><u>Tancar incidència</u></p> <ol style="list-style-type: none"> L'usuari escriu al quadre de text d'accions com ha solucionat la incidència i prem el botó "Tancar". Si prem el botó tancar sense explicar com l'ha solucionada surt un missatge d'error que l'usuari ha d'acceptar El sistema afegeix el dia i hora actuals, guarda la solució i guarda la incidència amb l'estat solucionat. 		

CAS D'ÚS	Solucionar Incidències
	<p><u>Sortir</u></p> <p>En qualsevol moment l'usuari pot sortir de la incidència.</p>
Fluxos alternatius	<p><u>Incidència no solucionada</u></p> <p>Si en el subflux Tancar Incidència l'usuari no indica com l'ha solucionat, surt un missatge avisant d'aquest fet. L'usuari l'haurà d'acceptar i el cas d'ús es reprèn.</p>
Postcondició	
Requeriments no funcionals	
Prioritat	Alta
Comentaris	

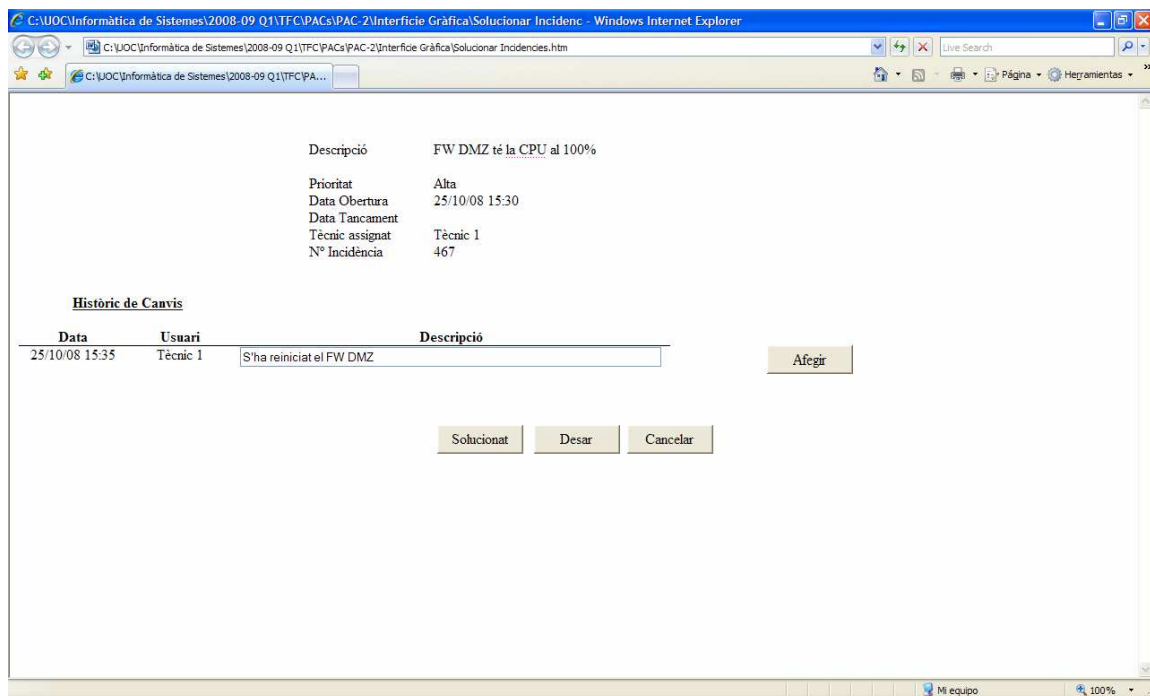


Figura 15. Pantalla solucionar incidències

2.4. Especificació de les classes d'anàlisis

Es consideren els tres tipus de classes d'anàlisis que es detallen a continuació:

- **Classes d'Entitats:** Són classes que modelen entitats o esdeveniments del món real dels quals el programari ha de fer servir informació pel que, molts d'aquests objectes, hauran de ser persistents
- **Classes de Frontera:** Aquestes classes representen objectes gràfics complexos com finestres, diàlegs per pantalla i menús. En aquesta etapa no es pretén descriure els detalls del format d'aquests objectes
- **Classes de Control:** Corresponen a objectes interns del programari i no persistents. Les operacions d'aquests tipus de classes contenen la lògica del negoci.

2.4.1. Classes d'entitats

- **Usuaris:** Són els usuaris de l'aplicació. Ens interessa guardar el seu nom, cognoms, NIF, login, password i si són usuaris actius o no. Els usuaris poden tenir diferents rols a la vegada (administrador, gestor de canvis, gestor d'incidències, etc..)

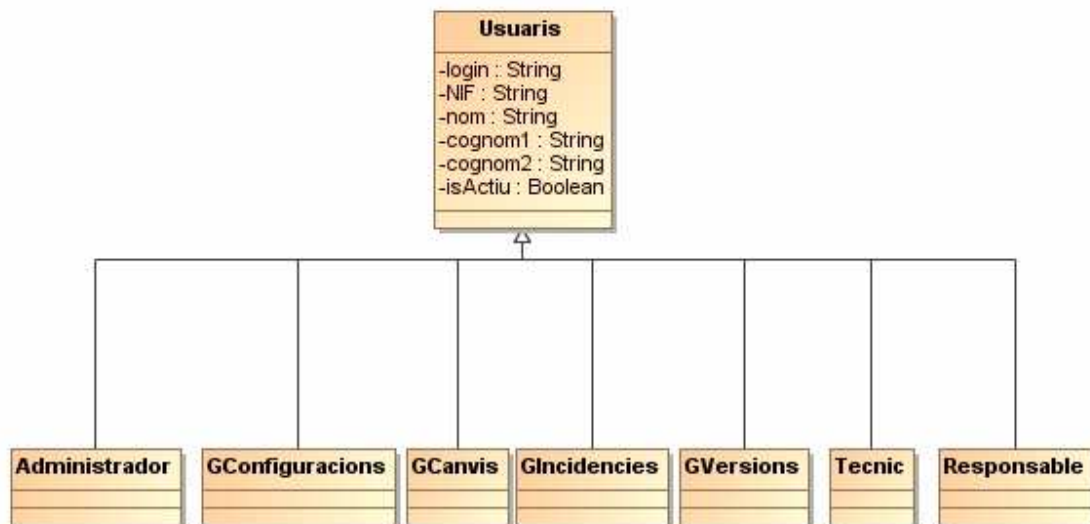


Figura 16. Classe Usuaris

- **CI:** Un CI (Configuration Item), és qualsevol element susceptible de ser controlat i configurat. Són elements tangibles (servidors, routers, commutadors, aplicacions...). S'han identificat 5 subclasses de CIs:

- **Servidors:** Són els servidors TI (servidors webs, servidors de BBDD, servidors d'aplicacions, etc...). Aquests servidors poden tenir backups
- **Equips TC:** Són els equips de comunicacions de nivell 3 o superiors de la capa OSI (routers, balancejadors, encriptadors SSL, etc...). Aquests equips poden tenir backups
- **Equips de Connexió:** Són els equips de nivell 1 i 2 de la capa OSI (hubs, commutadors, etc...). Són els equips on estaran connectats els servidors
- **HW:** Són els components hardware que formen els diferents equips (CPU, memòria, disc dur, etc...). Els components HW poden tenir backups
- **SW:** Són els components software que gestionen els diferents equips (sistema operatiu, sw de servidors, etc...)

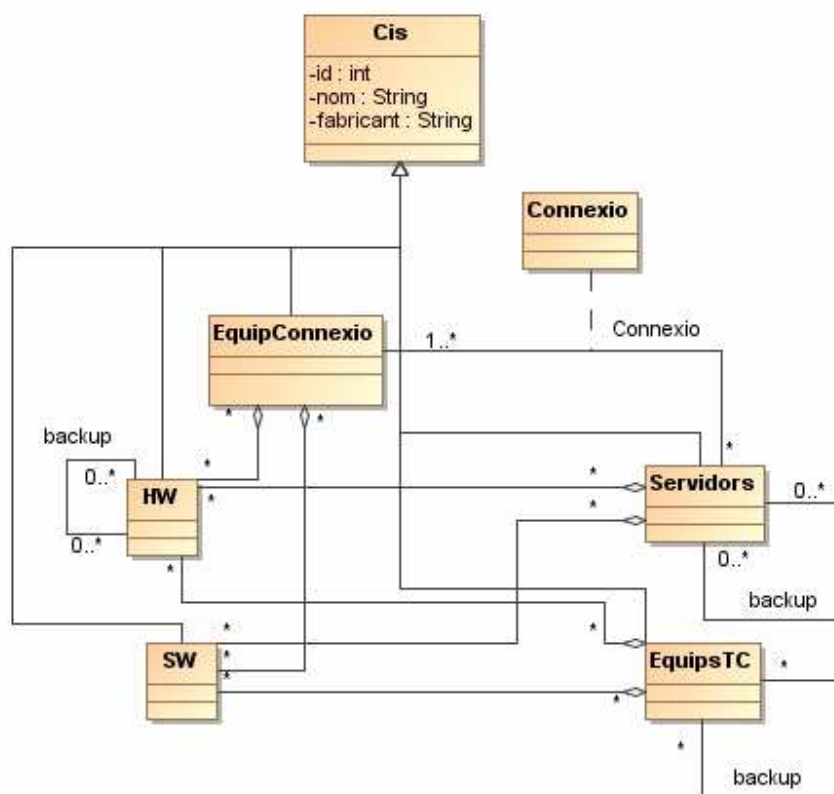


Figura 17. Classe Cis i les seves subclasses

- **Funcionalitats TI:** Són agrupacions lògiques de servidors TI que fan la mateixa funció. Per exemple, un conjunt de servidors web poden formar la funcionalitat de Hosting Internet que és on estaran hostatjades totes les webs de l'organització

- **Funcionalitats TC:** Són agrupacions lògiques d'equips de comunicacions que fan la mateixa funció. Per exemple, un conjunt de FWs poden formar la funcionalitat de FW de navegació que protegeix la navegació dels usuaris dels atacs externs. Les funcionalitats TC donen serveis a les funcionalitats TI i a alguns serveis
- **Serveis:** Són els lliurables d'una organització de serveis TI tal i com són percebuts pel client. Un servei en concret pot estar relacionat amb altres serveis. Per exemple, el servei d'Internet és necessari perquè el servei de correu i el de publicació de les webs de l'empresa funcionin correctament. A la seva vegada, els serveis reben servei de les funcionalitats TI i TC

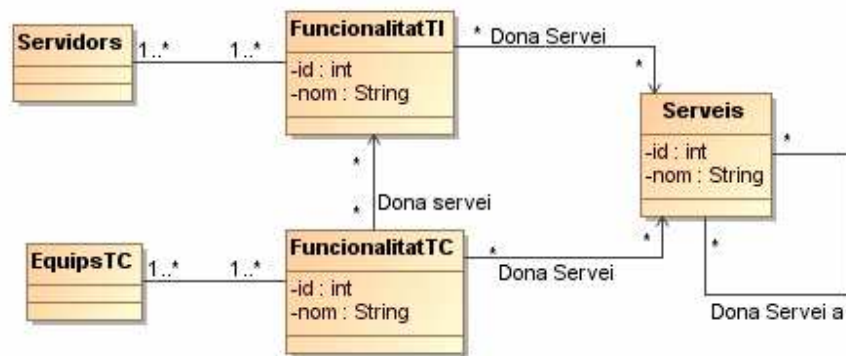


Figura 18. Classes FuncionalitatTI, FuncionalitatTC i Serveis

- **RFCs:** Són els registres dels detalls d'una petició de canvi (Request For Change) sobre qualsevol element de configuració dins una infraestructura. Els RFCs estan compostats per un conjunt de canvis a realitzar. Hem de guardar l'històric d'accions que els usuaris vagin realitzant sobre aquest RFC

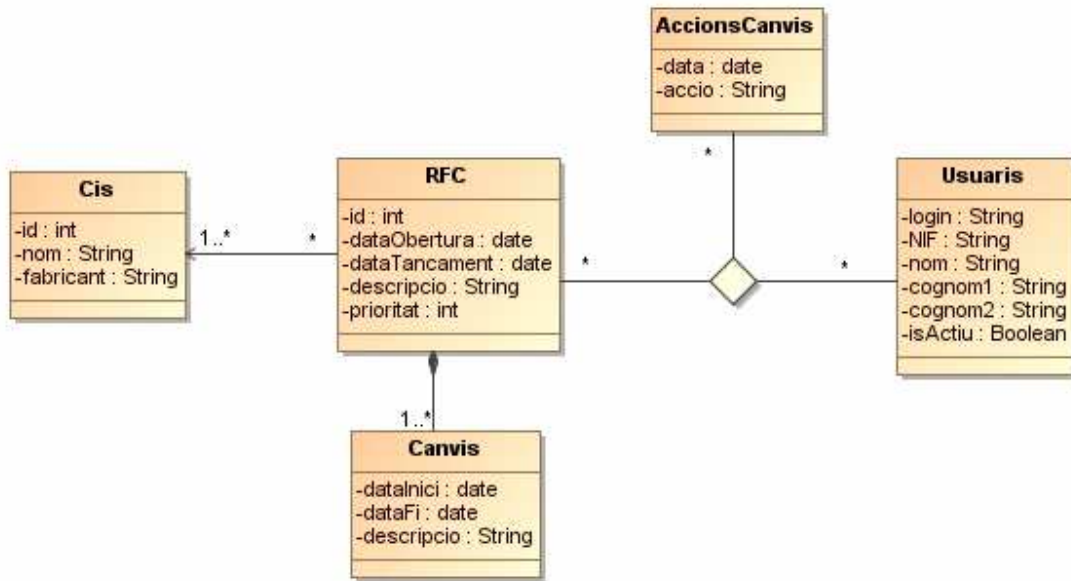


Figura 19. Classe RFC

- **Incidències:** Són errors, fallades o mal funcionaments d'un element de configuració. Hem de guardar l'històric d'accions que els usuaris vagin realitzant sobre aquestes incidències

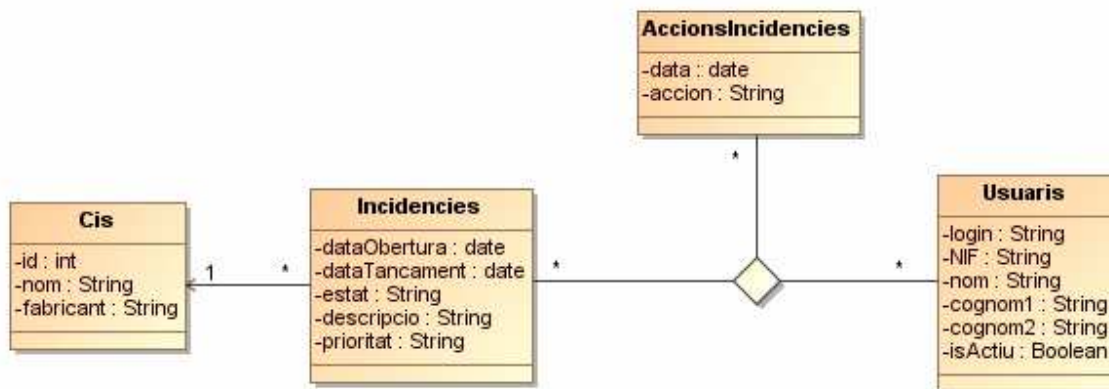


Figura 20. Classe Incidències

2.4.2. Classes de frontera

- **Pantalla d'entrada a l'aplicació:** En aquesta pantalla es demanarà el login i password de l'usuari per entrar a l'aplicació. Si un usuari intenta anar a qualsevol altre pantalla quan encara no està loginat, se'l reenviarà cap a aquesta pantalla

- **Pantalles de manteniment d'elements de la CMDB:** Aquest conjunt de pantalles agrupa les pantalles necessàries per fer el manteniment dels diferents elements de la CMDB. Per a cada element necessitem tres tipus de pantalles:
 - **Pantalles d'alta i cerca d'elements:** Aquestes pantalles permeten donar d'alta un nou element o buscar-ne a partir de certs atributs
 - **Pantalles d'elecció d'elements:** Aquestes pantalles s'utilitzen quan la cerca d'elements ha donat un resultat múltiple i mostren les dades identificatives de cada element perquè l'usuari en pugui triar un
 - **Pantalles de modificació d'elements:** En aquestes pantalles es mostren les dades dels elements en diferents quadres de text. Els quadres estaran buits si l'element s'ha creat nou o plens si l'element ja existeix. L'usuari els podrà modificar a la seva voluntat
- **Pantalles per acceptar canvis:** Aquest conjunt de pantalles agrupa les pantalles necessàries per acceptar o denegar canvis. Es necessiten dos tipus de pantalles:
 - **Pantalla resum de canvis:** Aquesta pantalla mostra el resum dels canvis pendents d'acceptar per l'usuari i permet acceptar o denegar els canvis
 - **Pantalla detall del canvi:** Aquesta pantalla mostra el detall d'un canvi i permet denegar-lo o acceptar-lo
- **Pantalla per realitzar canvis:** Aquesta pantalla permet que l'usuari vagi anotant la data d'inici i final reals de cada una de les accions que s'han planificat en el canvi en qüestió. També permet fer marxa enrere del canvi
- **Pantalla per solucionar incidències:** Aquesta pantalla permet que l'usuari vagi apuntat les accions que va realitzant per solucionar una incidència fins donar-la per tancada

2.4.3. Classes de control

- **ControladorUsuaris:** És la classe encarregada de controlar la lògica per donar d'alta, baixa o modificar els usuaris de l'aplicació
- **ControladorCIs:** És la classe encarregada de controlar la lògica per donar d'alta, baixa o modificar els elements de configuració
- **ControladorRFCs:** És la classe encarregada d'obrir, tancar i modificar les dades i l'estat dels RFCs. També determinarà l'afectació que tindran aquests RFCs sobre els serveis

- **ControladorIncidencies:** És la classe encarregada d'obrir, tancar i modificar les dades i l'estat de les incidències. També determinarà l'afectació que tindran aquestes incidències sobre els serveis

2.5. Anàlisi dels usuaris del sistema

Una de les claus perquè el sistema es percebi com a d'alta qualitat és que els usuaris es sentin còmodes amb ell i no els hi sigui complicat de fer servir. Per aquest motiu, en aquest apartat fem un petit anàlisi de com són els usuaris i com esperen que sigui l'aplicació

2.5.1. Perfils dels usuaris

Tots els usuaris coneixen a fons la seva feina, estan acostumats amb la nomenclatura que es fa servir i a treballar amb aplicacions amb interfícies webs.

L'únic rol que és nou pels usuaris és el d'administrador del sistema, ja que mai havien fet una tasca similar.

Totes les funcions de l'aplicació es faran servir de forma molt sovint.

2.5.2. Requisits d'usabilitat

La interfície d'usuari ha de ser senzilla i permetre una visió clara de totes les tasques que pot realitzar un usuari (cal recordar que un usuari pot tenir més d'un rol diferent on cada rol pot fer tasques molt diferents)

Els gestors de configuracions, gestors de canvis, gestors d'incidències, gestors de versions, responsables de servei i tècnics han de ser capaços de fer servir l'aplicació després d'una hora d'explicació i pràctica.

Els administradors han de poder fer servir el sistema després de dues hores d'explicació i pràctica.

3. Disseny de l'Aplicació

3.1. Introducció

A la fase d'anàlisi hem formalitzat el requeriments recollits inicialment, cosa que ens ha permès descriure les funcions que ha de fer el programari i els requeriments no funcionals que ha de complir.

Tot i que a la fase anterior s'ha plantejat formalment el problema que volem resoldre amb el programari, encara no és té la base adequada per emprendre directament la implementació de l'aplicació, ja que no s'ha expressat en termes de la tecnologia que es farà servir en el projecte.

Com es veurà en els següents apartats, proposem una solució basada en un arquitectura Java EE on integrarem les tecnologies, frameworks i patrons més adients per donar resposta als requeriments demanats.

3.2. Visió general de l'arquitectura de l'aplicació

La nostra aplicació estarà basada en la plataforma de programació Java EE (Java Platform, Enterprise Edition).

Aquesta plataforma ens permet implementar aplicacions multicapa escalables, d'alta disponibilitat, segures i eficients.

A la figura 2.1 es mostra el model multicapa de l'arquitectura JEE.

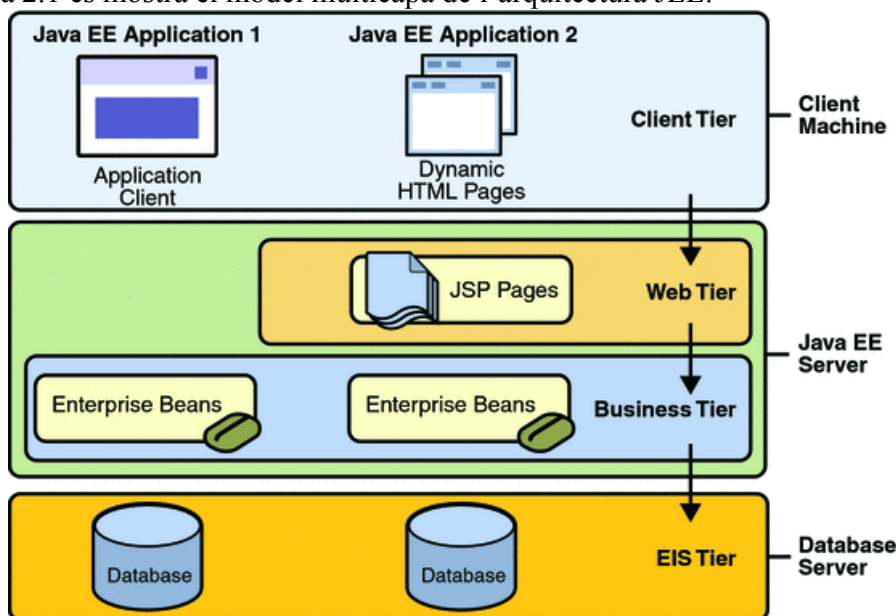


Figura 21. Model multicapa de l'arquitectura JEE

Com a framework de desenvolupament farem servir Struts, que és un framework que implementa el patró MVC (Model, Vista, Controlador)

Aquest patró defineix tres components lògics amb la finalitat de desacoblar l'accés a les dades, la lògica de negoci i la presentació dels resultats. Aquestes tres subunitats són:

- **Model:** És el component que conté una o més classes i interfícies que són les responsables de la lògica de negoci. S'ocupa de la integritat de les dades i permet derivar-ne de noves
- **Vista:** És el component responsable de presentar les dades amb un format adequat per interactuar-hi. El component vista pot recuperar informació del model però no pot modificar-la (aquestes peticions es dirigeixen al component controlador)
- **Controlador:** Aquest component és el responsable del control de flux. Respon a events, usualment accions de l'usuari, i invoca canvis al model i a la vista

A la figura 2.2 podem veure la implementació que fa Struts del patró MVC i com es relacionen els diferents components

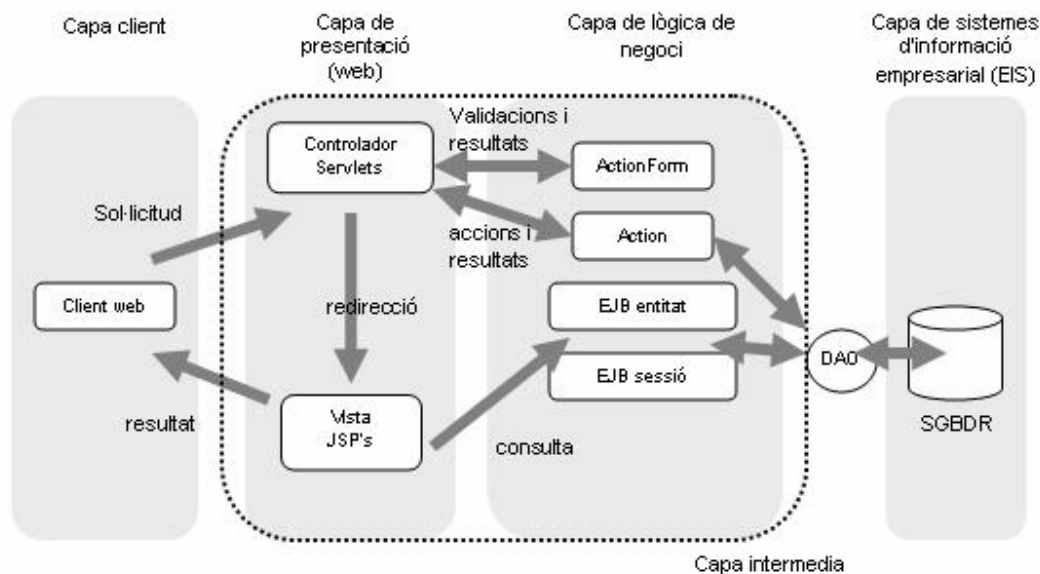


Figura 22. Funcionament d'Struts

Amb l'ús d'Struts, serem capaços de separar de forma clara la lògica del negoci, la presentació de les dades i la gestió de l'intercanvi d'informació entre capes.

Aquesta separació ens permetrà reduir el temps de desenvolupament, augmentar la qualitat de l'aplicació i reduir els errors.

A continuació detallem quines tecnologies farem servir a cada una de les capes de l'arquitectura JEE i quina relació tenen amb el model MVC que implementa Struts.

3.2.1. Capa de client

La capa de client és l'encarregada de proporcionar a l'usuari la interfície d'accés a l'aplicació.

De les diferents opcions que contempla l'arquitectura JEE per a la capa de client, hem optat per accedir a l'aplicació a través d'un navegador web.

Aquest navegador haurà de complir les especificacions HTML 4.01 i permetre l'execució de codi Javascript.

3.2.2. Capa de presentació

La capa de presentació és l'encarregada de controlar la generació del contingut a l'usuari final. Per aquest motiu, es comunica amb la capa de la lògica de negoci per obtenir les dades necessàries.

En aquesta capa, implementarem dues components del patró MVC:

- Vista: Utilitzarem pàgines JSP per crear el contingut dinàmic amb la combinació d'AJAX per millorar l'experiència de l'usuari
- Controlador: Com a controlador es farà servir el Servlet que proporciona el framework Struts

3.2.3. Capa de la Lògica de Negoci

La capa de la lògica de negoci és l'encarregada d'establir les regles que s'han de complir per implementar els processos i el tractament de les dades relacionades amb el negoci de l'empresa.

En aquesta capa implementarem el Model del patró MVC utilitzant EJB 3.0. Aquests EJB rebran les dades introduïdes pels clients, les processaran i les enviaran a la capa de sistemes d'informació empresarial per a que es guardin.

Els EJB també poden recuperar dades anteriorment guardades, processar-les i enviar-les cap al client per a la seva presentació.

3.2.4. Capa de Sistemes d'Informació Empresarials EIS

La capa EIS és l'encarregada de facilitar l'accés a la informació guardada a les BBDD.

L'estratègia que farem servir per a gestionar la persistència es basarà en el JPA (Java Persistence API) que facilita el mapeig objecte / relacional.

JPA es basa en classes d'entitat (Entity Class). Cada classe d'entitat representa una taula del model ER i cada instància de la classe correspon a una fila d'aquesta taula.

Per la seva senzillesa, estabilitat i rapidesa, s'ha optat per fer servir MySQL com a sistema gestor de la BBDD.

3.3. Disseny dels casos d'ús

En aquest apartat mostrarem els diagrames de seqüències de dos dels quatre casos d'ús de l'aplicació (modificació d'elements de la CMDB i solucionar incidències).

Els altres dos casos d'ús analitzats (acceptar canvis i realitzar canvis) són molt similars a solucionar incidències i el seus diagrames de seqüències no aportarien res de nou.

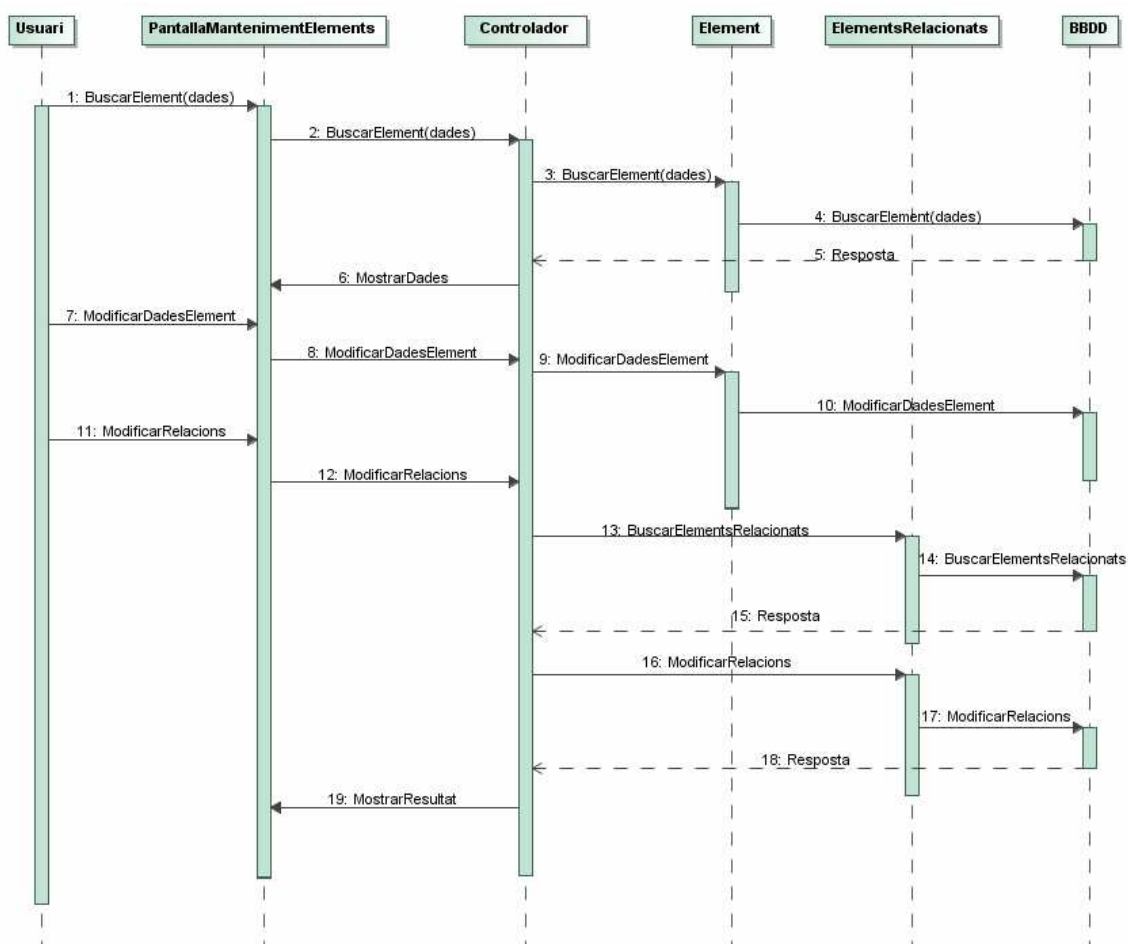


Figura 23. Diagrama de seqüències del cas d'ús Modificació d'Elements de la CMDB

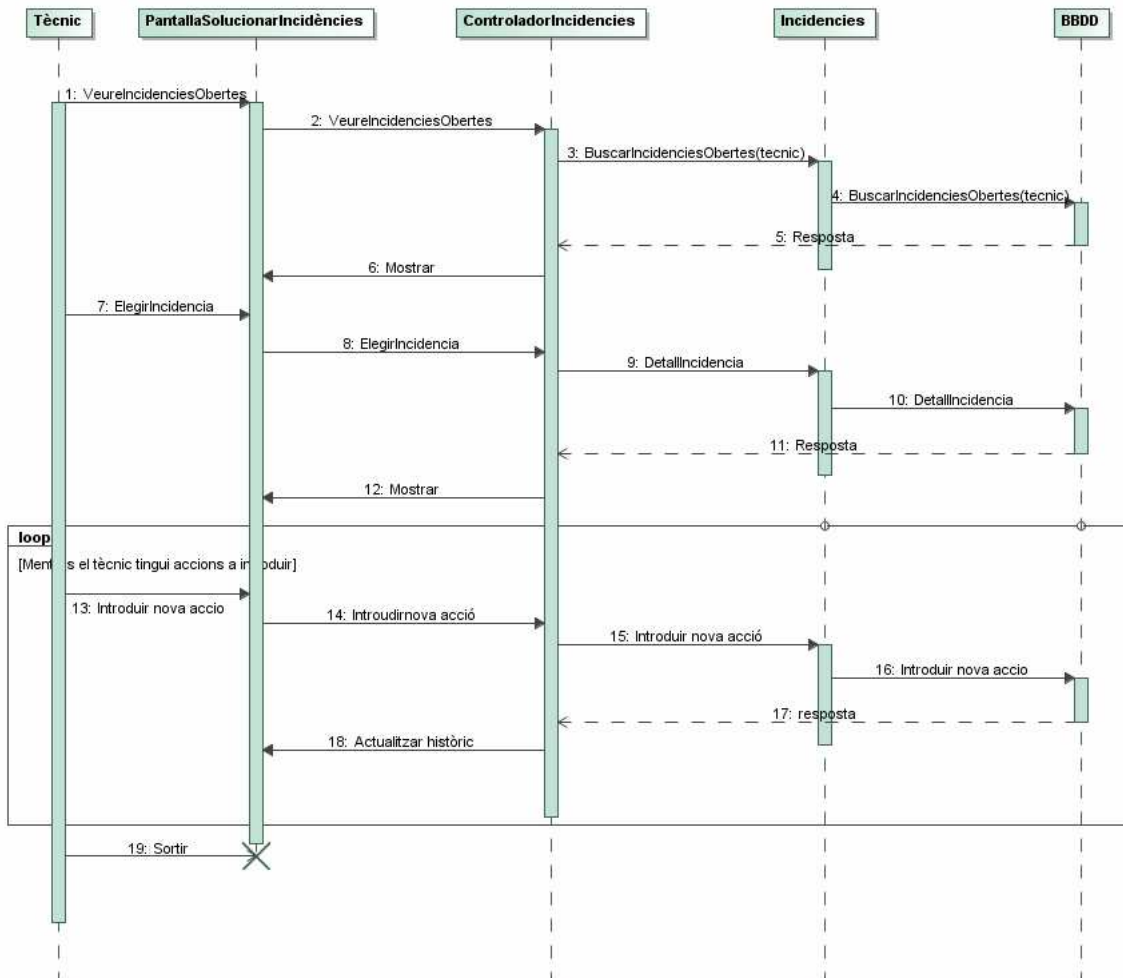


Figura 24. Diagrama de seqüències del cas d'ús Solucionar Incidències

3.4. Diagrama estàtic de disseny

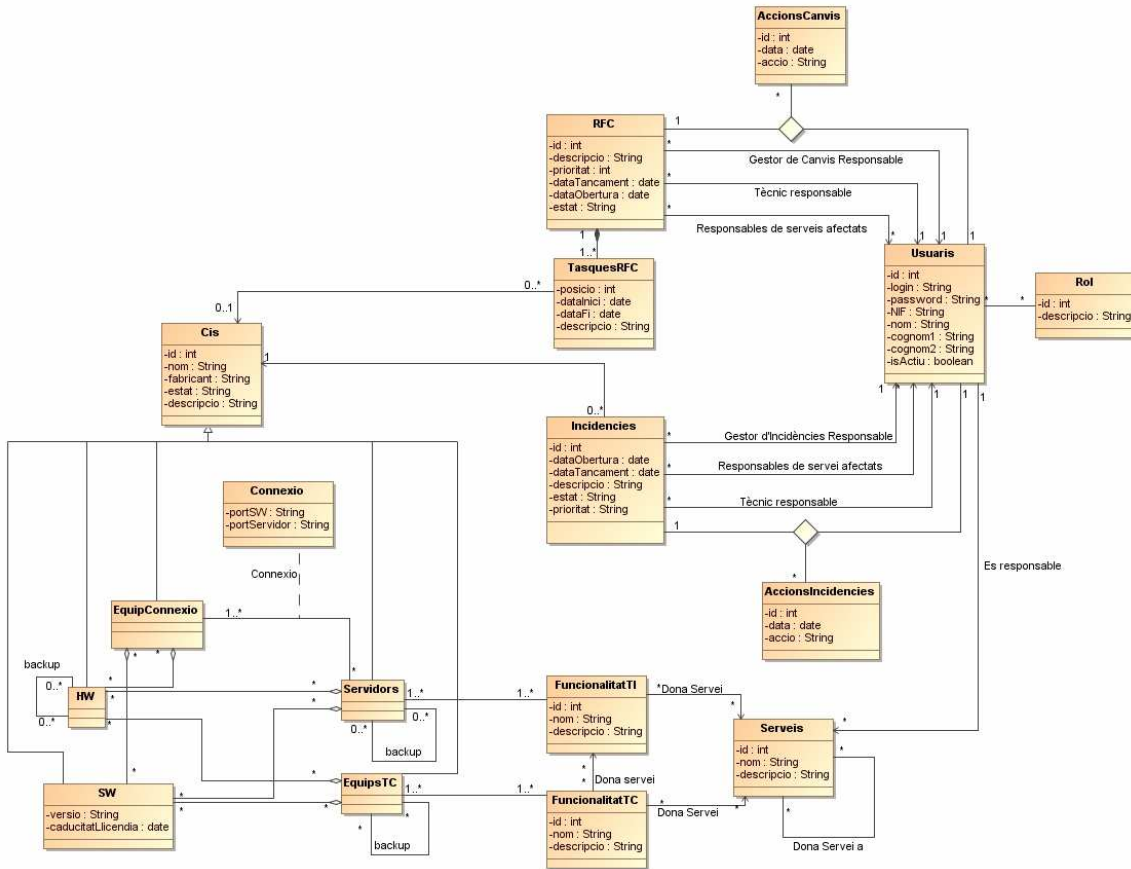


Figura 25. Diagrama estàtic de disseny

3.5. Diagrama d'estats

A continuació es mostren els diagrames d'estat dels elements més significatius de l'aplicació.

3.5.1. Estats d'una incidència

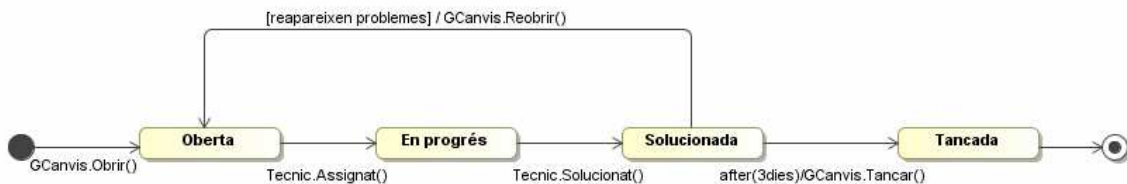


Figura 26. Diagrama d'estats d'una incidència

3.5.2. Estats d'un RFC

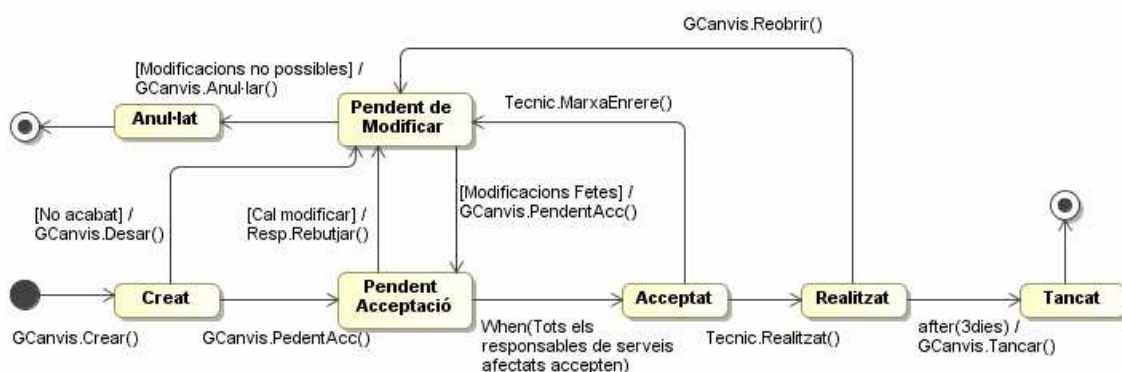


Figura 27. Diagrama d'estats d'un RFC

3.5.3. Estats de HW i SW

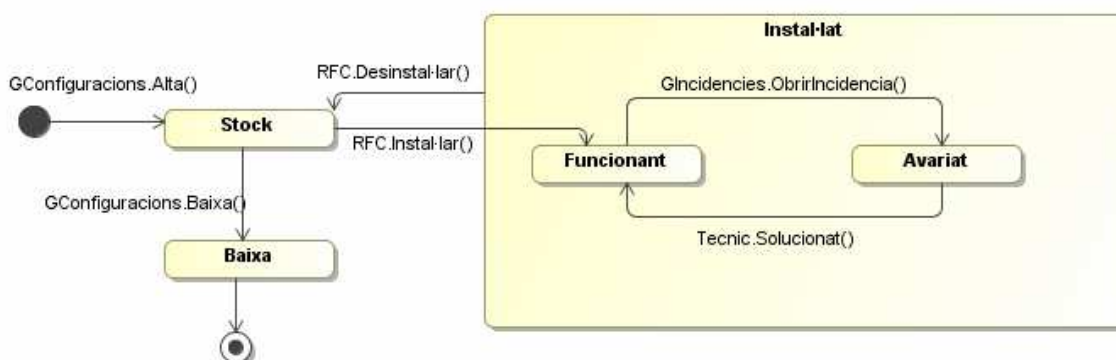


Figura 28. Diagrama d'estats de HW i SW

3.5.4. Estats dels Equips de Connexió, Servidors i Equips TC

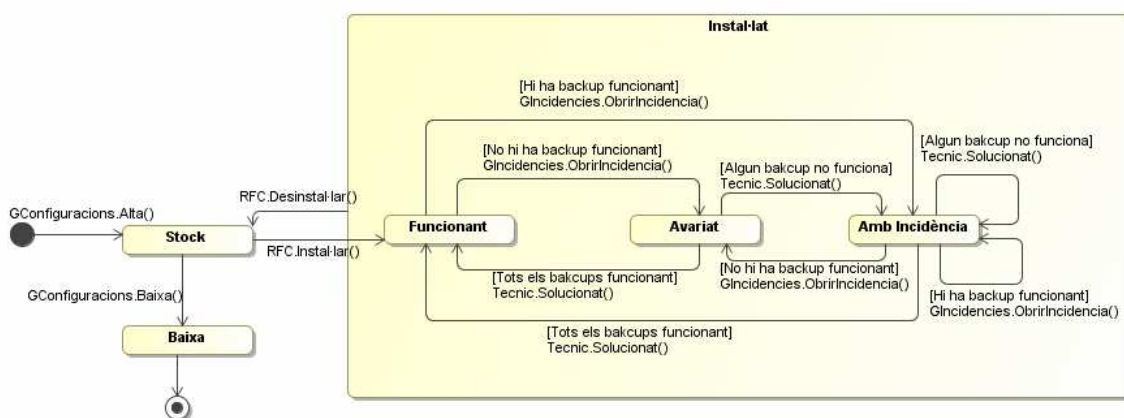


Figura 29. Diagrama d'estats Equips de Connexió, Servidors i Equips TC

3.5.5. Estats de les Funcionalitats TI i les Funcionalitats TC

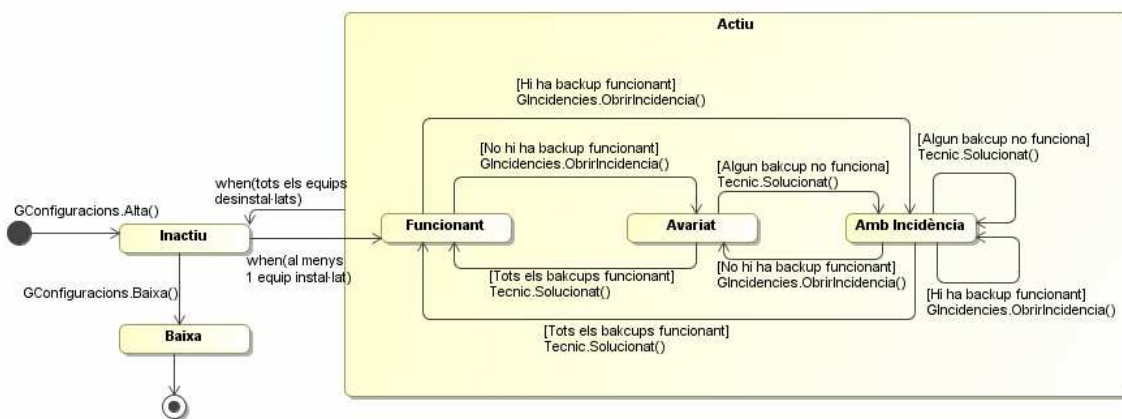


Figura 30. Diagrama d'estats de les Funcionalitats TI i les Funcionalitats TC

3.5.6. Estats dels Serveis

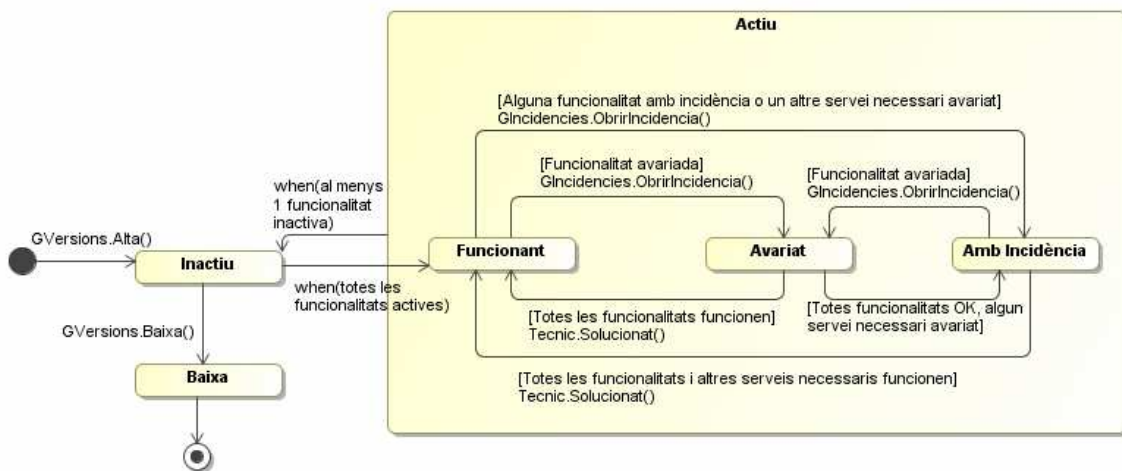


Figura 31. Diagrama d'estats dels Serveis

3.6. Disseny de la persistència

La base de partida per obtenir la definició de l'estructura de la base de dades relacional que farà servir l'aplicació és el diagrama estàtic de disseny que hem presentat a l'apartat 4.

En el primer subapartat transformarem aquest model estàtic en un model entitat-relació (model ER) i en el segon es detallarà els atributs de les diferents taules que tindrà l'aplicació.

3.6.1. Model ER

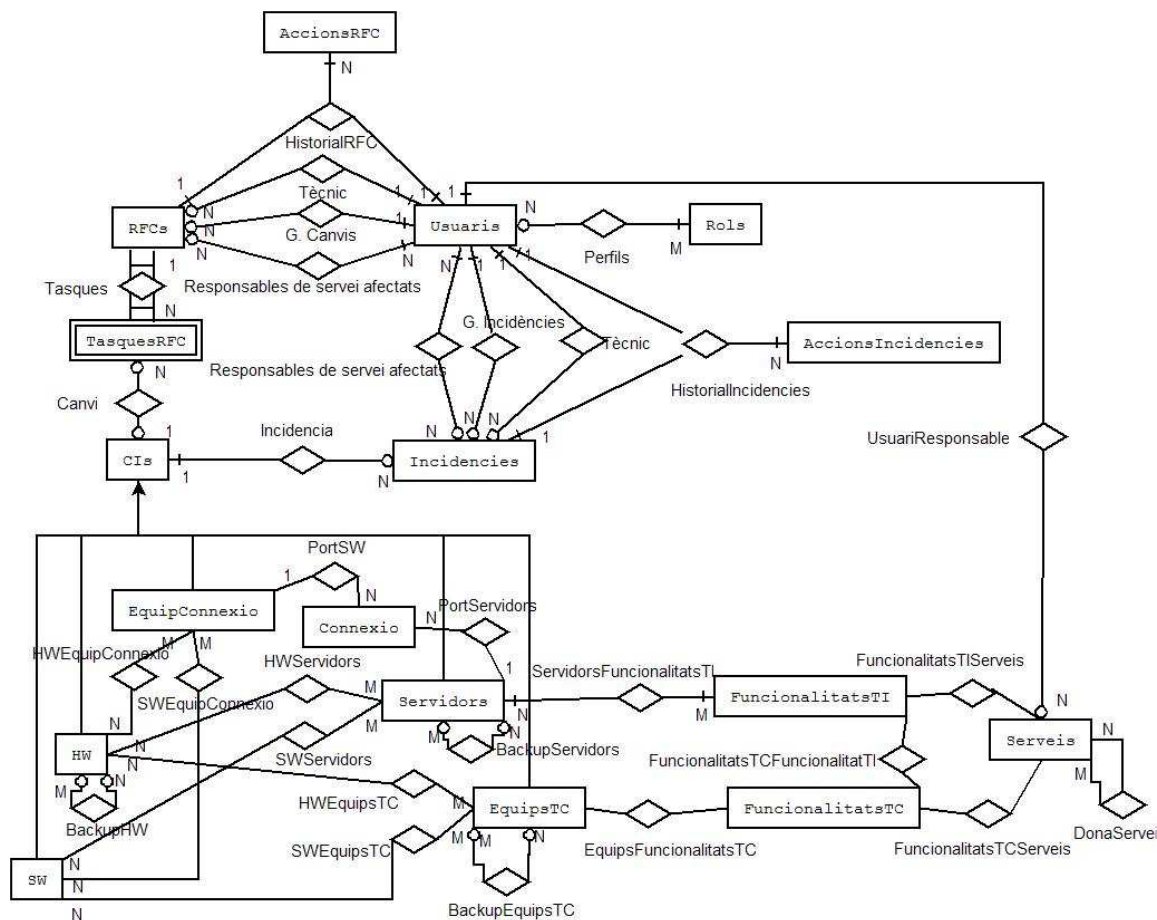


Figura 32. Model Entitat Relació

3.6.2. Atributs de les entitats i relacions del model ER

A continuació es mostren els atributs de les diferents taules que tindrà la base dades. Les claus primàries de cada taula estan subratllades.

ROLS

id, descripcio

USUARIS

id, login, password, NIF, nom, cognom1, cognom2, e-mail, isActiu

PERFILS

idUsuari, idRol

{idUsuari és clau forana de USUARIS i idRol ho és de ROLS}

ACCIONSINCIDENCIES

id, idIncidencia, idUsuari, data, accio
{idIncidencia és clau forana de INCIDENCIES i idUsuari ho és de USUARIS}

INCIDENCIES

id, idCI, dataObertura, dataTancament, descripcio, estat, prioritat,
idGestorIncidencies, idTecnic
{idCI és clau forana de CIS i idGestorIncidencies i idTecnic ho són de USUARIS}

RESPONSABLESAFECTATSCANVIS

idIncidencia, idResponsable
{idIncidencia és clau forana de INCIDENCIES i idResponsable ho és d'USUARIS}

CIS

id, nom, fabricant, estat, descripcio

RFC

id, dataObertura, dataTancament, descripcio, estat, prioritat, idGestorCanvis,
idTecnic
{idGestorCanvis i idTecnic són claus foranes de USUARIS}

RESPONSABLESAFECTATSCANVIS

idRFC, idResponsable
{idRFC és clau forana de RFC i idResponsable ho és de USUARIS}

TASQUESRFC

idRFC, posicio, idCI, dataInici, dataFi, descripcio
{idRFC és clau forana de RFC i idCI ho és de CIS}

ACCIONSCANVIS

id, idRFC, idUsuari, data, accio
{idRFC és clau forana de RFC i idUsuari ho és de USUARIS}

HW

id
{id és clau forana de CIS}

BACKUPHW

idHW, idBackup
{idHW i idBackup són claus foranes de HW}

SW

Id, versio, caducitatLlicencia
{id és clau forana de CIS}

EQUIPCONNEXIO

id
{id és clau forana de CIS}

SWEQUIPCONNEXIO

idSW, idEquipConnexio

{idSW és clau forana de SW i idEquipConnexio ho és de EQUIPCONNEXIO}

HWEQUIPCONNEXIO

idHW, idEquipConnexio

{idHW és clau forana de HW i idEquipConnexio ho és de EQUIPCONNEXIO}

SERVIDORS

id

{id és clau forana de CIS}

BACKUPSERVIDORS

idServidor, idBackup

{idServidor i idBackup són claus foranes de SERVIDORS}

SWSERVIDOR

idSW, idServidor

{idSW és clau forana de SW i idServidor ho és de SERVIDORS}

HWSERVIDOR

idHW, idServidor

{idHW és clau forana de HW i idServidor ho és de SERVIDORS}

EQUIPSTC

id

{id és clau forana de CIS}

BACKUPEQUIPSTC

idEquipTC, idBackup

{idEquipTC i idBackup són claus foranes de EQUIPSTC}

SWEQUIPSTC

idSW, idEquipTC

{idSW és clau forana de SW i idEquipTC ho és de EQUIPSTC}

HWEQUIPSTC

idHW, idEquipTC

{idHW és clau forana de HW idEquipTC ho és de EQUIPSTC}

CONNEXIO

id, idSwitch, idServidor, portSW, portServidor

{idSwitch és clau forana de EQUIPCONNEXIO i idServidor ho és de SERVIDORS}

FUNCIONALITATSTI

id, nom, descripció

SERVIDORSFUNCIONALITATSTI

idServidor, idFuncionalitatTI

{idServidor és clau forana de SERVIDORS i idFuncionalitatTI ho és de FUNCIONALITATSTI}

FUNCIONALITATSTC

id, nom, descripció

EQUIPSSFUNCIONALITATSTC

idEquipTC, idFuncionalitatTC

{idEquipTC és clau forana de EQUIPSTC i idFuncionalitatTC ho és de FUNCIONALITATSTC}

FUNCIONALITATTCFUNCIONALITATI

IdFuncionalitatTC, idFuncionalitatTI

{IdFuncionalitatTC és clau forana de FUNCIONALITATSTC i idFuncionalitatTI ho és de FUNCIONALITATSTI}

SERVEIS

id, nom, descripció

DONASERVEI

idServeiServidor, idServeiClient

{idServeiServidor i idServeiClient són clau forana de SERVEIS}

FUNCIONALITATSTISERVEIS

idFuncionalitatTI, idServei

{idFuncionalitatTI és clau forana de FUNCIONALITATSTI i idServei ho és de SERVEIS}

FUNCIONALITATSTCSERVEIS

idFuncionalitatTC, idServei

{idFuncionalitatTC és clau forana de FUNCIONALITATSTC i idServei ho és de SERVEIS}

USUARIRESPONSABLE

idUsuari, idServei

{idUsuari és clau forana de USUARIS i idServei ho és de SERVEIS}

3.7. Diagrama de components

A la figura 7.1 podem veure el diagrama de components de l'aplicació. En aquest diagrama podem observar dues parts, la capa Web, on s'agrupen les vistes i el controlador, i la capa EJB, on tenim el model i les regles de negoci.

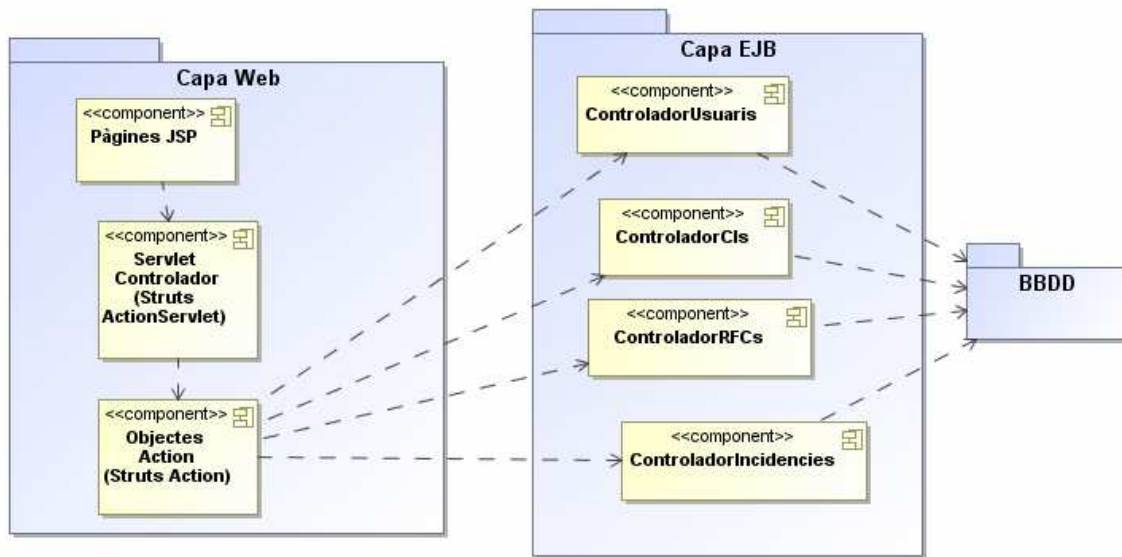


Figura 33. Diagrama de components de l'aplicació

3.7.1. Capa web

En aquesta capa s'ofereix la porta d'entrada a l'aplicació. Quan el controlador rep una petició, instancia un objecte Action que implementarà un cas d'ús en concret.

Aquest Action es comunicarà amb el controlador de l'element corresponent

3.7.2. Capa EJB

En aquesta capa es defineixen els diferents EJBs que emmagatzemaran les regles de negoci de l'aplicació. Aquests serveis tindran accés tant a components de càlcul com a les dades de la base de dades

3.8. Diagrama de desplegament

A la figura 8.1 podem veure el diagrama de desplegament de l'aplicació. Com es pot observar, tenim dos nodes, el client i el servidor.

El node de client té un navegador web que es comunica amb el servidor d'aplicacions del servidor i aquest es comunica amb la BBDD que també resideix al mateix servidor

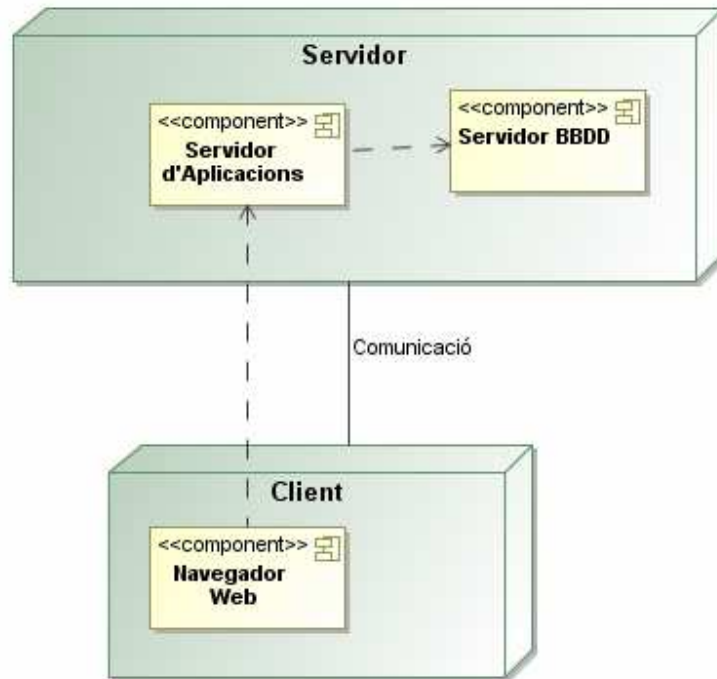


Figura 34. Diagrama de desplegament de l'aplicació

3.9. Descripció del programari necessari

Per implementar i provar el sistema utilitzarem el següent programari:

- IDE: Netbeans 6.1
- Servidor d'aplicacions: GlassFish V2 UR1
- SGBD: MySQL Server 5.0
- Framework MVC: Apache Struts 1.2.9
- Navegadors web: Microsoft Internet Explorer 7 i Mozilla Firefox 3

4. Conclusions

Com s'ha explicat a llarg d'aquesta memòria, **ITIL** s'ha convertit en l'estàndard de facto de la indústria per a la gestió de serveis TI.

En aquest TFC hem realitzat una aplicació web que permet a una empresa del sector TIC introduir-se en l'ús de les bones pràctiques que descriu ITIL.

L'ús de la plataforma **Java EE** i frameworks com ara **Struts** i **Tiles** ens ha permès obtenir una aplicació robusta, escalable i segura.

L'elecció d'aquesta arquitectura ens ha permès tenir una clara separació de funcionalitats entre components, desacoblant així el codi d'accés a les dades (on hem utilitzat **JPA**), el codi de la lògica de negoci (implementada amb **EJB 3.0**) i el codi de presentació (amb **JSP** i **AJAX**).

Des d'un punt de vista més personal, la realització d'aquest TFC ha estat per a mi un experiència molt enriquidora i que m'ha permès aprofundir en tecnologies que fa uns mesos desconeixia per complet.

Amb aquest treball he pogut aplicar en un projecte de construcció de programari els coneixements adquirits durant la carrera, en especial tots els relacionats amb la programació orientada a objectes, enginyeria del programari i BBDD.

Si bé la corba d'aprenentatge de les tecnologies que he fet servir ha estat molt dura, ara que miro cap endarrere, penso que ha valgut la pena i estic orgullós del treball realitzat.

5. Glossari

- **CI:** (Configuration Item) Qualsevol element tangible (servidors, routers, commutadors, aplicacions...) susceptible de ser controlat i configurat
- **CMDB:** Base de dades de gestió de configuracions. Base de dades que conté totes les dades rellevants de cada element de configuració (CI) i els detalls de les interconnexions i relacions existents entre ells
- **Gestor d'incidències:** És la persona responsable de determinar l'afectació sobre els processos de negoci de les incidències que es produeixin
- **Gestor de canvis:** És la persona responsable de tenir control de tots els canvis tècnics que es realitzen i obtenir l'autorització dels responsables dels serveis que es vegin afectats per aquests canvis
- **Gestor de configuracions:** És la persona responsable que l'inventari dels elements de configuració estigui actualitzada i que les dades siguin correctes
- **Funcionalitats TC:** Són agrupacions lògiques d'equips de comunicacions que fan la mateixa funció. Per exemple, un conjunt de FWs poden formar la funcionalitat de FW de navegació que protegeix la navegació dels usuaris dels atacs externs. Les funcionalitats TC donen serveis a les funcionalitats TI i a alguns serveis
- **Funcionalitats TI:** Són agrupacions lògiques de servidors TI que fan la mateixa funció. Per exemple, un conjunt de servidors web poden formar la funcionalitat de Hosting Internet que és on estaran hostatjades totes les webs de l'organització
- **Incidències:** Són errors, fallades o mal funcionaments d'un element de configuració i que causa o pot causar una interrupció del servei o una reducció de la seva qualitat
- **ITIL:** Biblioteca d'infraestructures de TI de la OGC – Conjunt de guies sobre la gestió i provisió de serveis de TI. Actualment és l'estàndard global de facto a l'àrea de gestió de serveis TI
- **Responsable de Servei:** És la persona que té la visió de negoci dels serveis dels quals és responsable. És l'encarregat d'assegurar que els canvis afecten el menys possible al negoci i té potestat per modificar les finestres d'actuació proposades pel gestor de canvis
- **RFC:** Són els registres dels detalls d'una petició de canvi (Request For Change) sobre qualsevol element de configuració dins una infraestructura
- **Serveis:** Són els lliurables d'una organització de serveis TI tal i com són percebuts pel client. Un servei en concret pot estar relacionat amb altres serveis. Per exemple,

el servei d'Internet és necessari perquè el servei de correu i el de publicació de les webs de l'empresa funcionin correctament. A la seva vegada, els serveis reben servei de les funcionalitats TI i TC

6. Bibliografia

Java EE:

The Java EE 5 Tutorial. <http://java.sun.com/javaee/5/docs/tutorial/doc/index.html> (darrera consulta 15/12/08)

Netbeans IDE – Java EE Development. <http://www.netbeans.org/features/web/java-ee.html> (darrera consulta 30/12/08)

AJAX:

Crane, Dave; Pascarella, Eric. (2006). “Ajax in action” (1ª ed.). Greenwich: Manning Publications

Project jMaki. <https://ajax.dev.java.net> www.jmaki.org (darrera consulta 30/12/08)

Ajax Ya. <http://www.ajaxya.com.ar> (darrera consulta 28/11/08)

MySQL:

MySQL. <http://www.mysql.com> (darrera consulta 3/11/08)

MySQL Developer Zone. <http://dev.mysql.com> (darrera consulta 17/12/08)

UML:

Martin, Robert C. (2004). “UML para programadores Java” (1ª ed.). Madrid: Person Educación.

JSP:

Trigos García, Esteban. (2001). “JSP” (1ª ed.) (“Colección Guías Prácticas”). Madrid: Anaya Multimedia

JSP Tutorial. <http://www.jsptut.com> (darrera consulta 30/10/08)

ITIL:

itSMF Internacional. <http://www.itsmfi.org> (darrera consulta 6/10/08)

ITIL V2 Fundamentos. <http://itil.wikidot.com> (darrera consulta 15/10/08)

Struts:

The Apache Software Foundation. <http://struts.apache.org> (darrera consulta 25/10/08)

Cavaness, Chuck. (2002). “Programming Jakarta Struts” (1^a ed.). Sebastopol: O'Reilly & Associates

Altres:

Programación en castellano. <http://www.programacion.com> (darrera consulta 30/10/08)

Wikipedia. <http://www.wikipedia.org> (darrera consulta 15/12/08)

La web del programador. <http://www.lawebdelprogramador.com> (darrera consulta 1/12/08)

Adictos al trabajo. <http://www.adictosaltrabajo.com> (darrera consulta 29/12/08)

Netbeans Docs & Support. <http://www.netbeans.org/kb/> (darrera consulta 21/12/08)

Forums de SUN. <http://forums.sun.com> (darrera consulta 22/12/08)

Java Passion. <http://www.javapassion.com/> (darrera consulta 20/11/08)