Desenvolupament d'aplicacions per a Internet amb Tecnologia JAVA.
Projecte *Xhelp OutsourcingService*

José Quiles Garcia
ETIG

Albert Grau Perise

12 de Juny de 2004

Contingut del Document

En el present document hi trobareu un resum sobre el meu Projecte Final de Carrera, amb reflexions del perquè del projecte i perquè J2EE com a eina de desenvolupament.

Després d'aquest resum entrarem en detall sobre els aspectes més crucials del procés de desenvolupament, de les fases d'anàlisi, disseny i desenvolupament que s'han anat aplicant de manera iterativa durant el projecte.

Una vegada explicat el mètode de control i gestió del projecte es dgna una explicació sobre les principals tecnologies aplicades en el desenvolupament del projecte: J2EE i Apache Struts.

Hi trobareu també, un capítol sencer amb tota la documentació generada (la documentació final, una vegada depurades totes les iteracions). Allà hi trobareu el disseny dels casos d'us, el disseny de l'aplicació, un diagrama de la base de dades i el model genèric de l'arquitectura utilitzada.

Índex general

Introducció. **6**
Objectius del TFC 7
Enfocament i mètode seguit 8
Planificació 9
Producte obtingut 11
Contingut específic de la memòria 12

L'aplicació Xhelp OutsourcingService. **13**
Requisits funcionals 13
Glossari de Termes. 15
Definició d'actors i casos d'us 16
Classificació dels Actors del sistema. 18

Casos d'us dels actors. **19**
Casos d'us del Responsable del Client 19
Casos d'us del cap de departament 20
Casos d'us de l'usuari bàsic 20
Casos d'us del responsable d'outsourcing 21
Casos d'us del tècnic del CAU 21
Casos d'us del tècnic 21

Descripció dels casos d'Us. **21**
Comunicació de peticions al servei de HelpDesk. 21
Consulta de peticions. 21
Assignació de peticions. 21
Gestió d'usuaris. 21
Configuració de Departaments. 21
Tancament de Peticions. 21
Valoració de Peticions. 21
Aprovació de Valoracions de Peticions. 21
Realització d'accions sobre peticions. 21
Proposta de Nivells de Servei. 21
Aprovació de Nivells de Servei. 21
Informe SLR Mensual. 21

Especificació dels casos d'Us. **21**
Diagrames de seqüències. 21
Diagrama de seqüències de: Consulta de Peticions. 21
Diagrama de seqüències de: Gestió d'usuaris. 21
Diagrama de seqüències de: Tancament de peticions. 21
Diagrama de seqüències de: Accions sobre peticions. 21
Diagrama de seqüències de: Avaluació de peticions. 21
Diagrama de seqüències de: Realització de peticions. 21
Diagrama de seqüències de: Gestió de departaments. 21

Disseny tècnic del sistema **21**
Entitats derivades de l'anàlisi. 21
Diagrama de Classes. 21

Arquitectura de la Persistència i ús d'EJB d'entitat. **21**
Diagrames d'estat 21
Diagrama d'Estats d'Incidència 21
Diagrama d'Estats d'Evolutiu 21
Diagrama d'estats de Consulta 21

Diagrama d'estats de Tasca Periòdica 21

Disseny de l'arquitectura **21**
El patró de disseny MVC (Model view controller). Apache Struts. 21
Apache Struts. 21
J2EE i el patró MVC 21
Disseny del control d'accés i d'usuaris 21
Distribució del programari en paquets. 21

Índex d'il·lustracions

Il·lustració 1. Planificació final del projecte 9
Il·lustració 2. Gantt de Planificació del projecte 10
Il·lustració 3. Actors de XHelp OutsourcingService 16
Il·lustració 4. Classificació dels casos d'us per prioritats 17
Il·lustració 5. Casos d'us del Responsable del client 19
Il·lustració 6. Casos d'us d'un cap de departament 20
Il·lustració 7.Casos d'us d'un usuari bàsic 20
Il·lustració 8. Casos d'us del Responsable d'outsourcing 21
Il·lustració 9. Casos d'us del Tècnic del C.A.U. 21
Il·lustració 10. Casos d'us del Tècnic Resolutor 21
Il·lustració 11. Diagrama de seqüències **Consulta de Peticions** 21
Il·lustració 12. Diagrama de seqüències **Gestió d'usuaris**. 21
Il·lustració 13. Diagrama de seqüències **Tancament de peticions.** 21
Il·lustració 14. Diagrama de seqüències **Actuacions sobre peticions.** 21
Il·lustració 15. Diagrama de seqüències **Avaluació de peticions.** 21
Il·lustració 16. Diagrama de seqüències **Realització de peticions.** 21
Il·lustració 17. Diagrama de seqüències **Gestió de departaments.** 21
Il·lustració 18. Diagrama de classes 21
Il·lustració 19. Diagrama de classes EJB i disseny de la persistència. 21
Il·lustració 20. Diagrama d'estats de **Consulta**. 21
Il·lustració 21. Diagrama d'estats d'**Evolutiu**. 21
Il·lustració 22. Diagrama d'estats d'**Incidència**. 21
Il·lustració 23. Diagrama d'estats de **Tasca periòdica**. 21
Il·lustració 24. Exemple d'arquitectura basat en Apache Struts. 21
Il·lustració 25. Arquitectura general de Xhelp OutsourcingService 21
Il·lustració 26. Disseny del control d'accés. 21
Il·lustració 27. Plana d'accés a Xhelp OutsourcingService 21
Il·lustració 28. Menú principal de l'Administrador 21
Il·lustració 29. Manteniment de departaments del client / grups funcionals 21
Il·lustració 30. Configuració dels nivells de servei 21
Il·lustració 31. Manteniment d'usuaris 21
Il·lustració 32. Creació d'una petició al sistema 21
Il·lustració 33. Assignació a un grup de tècnics 21
Il·lustració 34. Actuació sobre una petició 21
Il·lustració 35. Consulta de peticions, amb l'historial d'actuacions. 21
Il·lustració 36. Informe SLR 21

Introducció.

En la meva vida professional em dedico al desenvolupament d'aplicacions Java per a Internet, tot i que sovint no he fet servir la tecnologia J2EE per aquest tipus de projectes.

Per aquest motiu em vaig sentir interessat per desenvolupar aquest projecte, i també, per que l'àmbit d'aquest desenvolupament es pot ajustar a la meva feina diària. Soc responsable del manteniment d'algunes aplicacions per a Internet, al departament d'Application Management d'una coneguda consultora, i sovint, hi trobo a faltar alguna eina de gestió com la que he desenvolupat.

L'entorn de l'aplicació que he desenvolupat es dona en una relació client-consultor per a un determinat servei (l'aplicació en suporta més d'un). Parlem de servei, en el sentit del manteniment evolutiu i correctiu d'una aplicació informàtica en explotació.

En un servei d'aquest tipus, les peticions que realitza el client per modificar o corregir aspectes de l'aplicació objecte del servei, han de seguir un flux de controls i validacions, des del moment de la petició fins la resolució, que assegurin una mínima qualitat en tot el procés. Això implica enregistrar i documentar totes les actuacions que es facin per resoldre una determinada petició.

Amb aquesta idea vaig pensar en desenvolupar *Xhelp OutsourcingService*, d'una banda utilitzar una tecnologia moderna i professional, com J2EE i d'altra, fer-ho per desenvolupar una eina d'utilitat en el mon real.

L'abast de les especificacions inicials del projecte s'han complert, tot i alguns problemes de darrera hora amb la planificació. Al final, el producte desenvolupat compleix amb les necessitats especificades al document inicial del projecte, amb les necessitats tècniques i amb el disseny original.

Evidentment, durant el desenvolupament del projecte han aparegut situacions que m'han forçat a aplicar petits canvis de disseny en l'aplicació, com ara, el desenvolupament d'un petit motor d'estats dirigit per base de dades, que gestiona, ara, el nucli de l'aplicació, i que, en el seu moment, es va dissenyar com una lògica de negoci més. Amb aquest canvi s'aconsegueix més flexibilitat de configuració del producte i més facilitat per ampliar i escalar l'aplicació.

Finalment, voldria dir que em sento satisfet per la feina feta, pel coneixement sobre J2EE i altres tecnologies, com Struts, Eclipse, JBOSS, MySQL, etc... que he anat adquirint a mida que portava a terme el desenvolupament, i també pels coneixements de gestió i seguiment de projectes que m'han ajudat a prendre decisions sobre el rumb de l'aplicació que m'han portat a bon fi.

Objectius del TFC

Com a objectiu principal del TFC m'he proposat aprendre sobre l'arquitectura J2EE i sobre tècniques modernes de desenvolupament de programari, més adaptades a les noves tecnologies que fem servir com ara JAVA i Internet.

El següent objectiu del TFC ha estat desenvolupar una eina de gestió de projectes de manteniment de programari, que donés la possibilitat de enregistrar i controlar la qualitat del servei que es dona al client d'una empresa consultora de programari.

Enfocament i mètode seguit

Com tot projecte de desenvolupament de programari, el procés de construcció s'ha dividit en diverses fases ben diferenciades i amb objectius molt diferents. Les fases utilitzades han estat les habituals: Presa de requisits, anàlisi d'aquests requisits, disseny, desenvolupament, proves i desplegament.

Els projectes de construcció de programari es resolen millor si s'apliquen iteracions a aquestes fases. El fet d'aplicar iteracions permet detectar mancances i indefinicions en els requisits que es poden anar resolent a mida que avança el projecte.

Fer servir aquestes iteracions pot perjudicar en part la planificació inicial del projecte, però també ajuden a determinar i diferenciar les funcionalitats més importants i que no poden faltar en el producte final, d'aquelles que poden quedar per a "segones fases" pel seu baix impacte funcional.

L'enfocament bàsic del projecte es des del punt de vista del desenvolupament de eines de gestió internes d'una companyia, de manera que tant els requisits funcionals com tècnics venen donats per les necessitats de gestió del negoci, i no de tercers com seria el cas d'un desenvolupament d'un producte per a un client.

Planificació

La feina ha estat concentrada els caps de setmana. Durant la setmana el meu horari laboral no em permet fer gaires coses, tot i que ha estat necessari un esforç en aquest sentit per portar a terme el projecte. La necessitat de concentrar la feina en els caps de setmana ha estat un punt a tenir en compte en la planificació, on la majoria de les tasques finalitzen un diumenge o un dilluns.

Per mantenir un control sobre la marxa del projecte es va desenvolupar un diagrama de Gantt inicial, que s'ha anat actualitzant a mida que el procés avançava. La planificació final, i per tant, l'estat final del projecte son els següents:

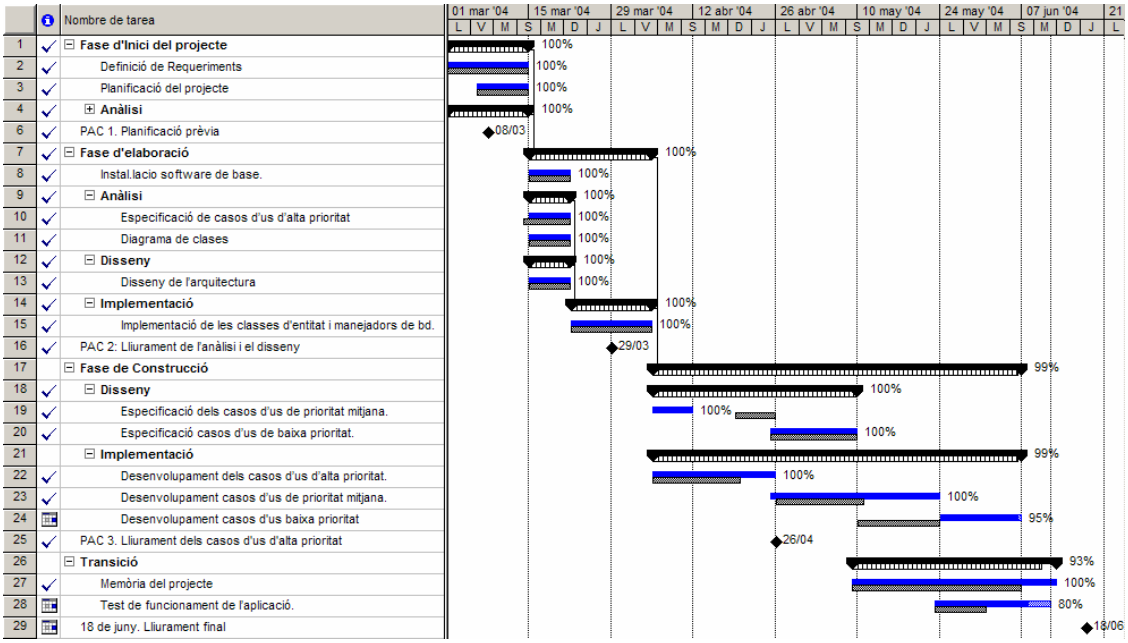
Gantt de seguimiento

	Nombre de tarea	Duración	Comienzo	Fin
1	**Fase d'Inici del projecte**	**14 días**	**lu 01/03/04**	**do 14/03/04**
2	Definició de Requeriments	14 días	lu 01/03/04	do 14/03/04
3	Planificació del projecte	9 días	sá 06/03/04	do 14/03/04
4	**Anàlisi**	**14 días**	**lu 01/03/04**	**do 14/03/04**
6	PAC 1. Planificació prèvia	1 día	lu 08/03/04	lu 08/03/04
7	**Fase d'elaboració**	**21 días**	**lu 15/03/04**	**do 04/04/04**
8	Instal.lacio software de base.	7 días	lu 15/03/04	do 21/03/04
9	**Anàlisi**	**7 días**	**lu 15/03/04**	**do 21/03/04**
10	Especificació de casos d'us d'alta prioritat	7 días	lu 15/03/04	do 21/03/04
11	Diagrama de clases	7 días	lu 15/03/04	do 21/03/04
12	**Disseny**	**7 días**	**lu 15/03/04**	**do 21/03/04**
13	Disseny de l'arquitectura	7 días	lu 15/03/04	do 21/03/04
14	**Implementació**	**14 días**	**lu 22/03/04**	**do 04/04/04**
15	Implementació de les classes d'entitat i manejadors de bd.	14 días	lu 22/03/04	do 04/04/04
16	PAC 2: Lliurament de l'anàlisi i el disseny	1 día	lu 29/03/04	lu 29/03/04
17	**Fase de Construcció**	**63 días**	**lu 05/04/04**	**do 06/06/04**
18	**Disseny**	**35 días**	**lu 05/04/04**	**do 09/05/04**
19	Especificació dels casos d'us de prioritat mitjana.	7 días	lu 05/04/04	do 11/04/04
20	Especificació casos d'us de baixa prioritat.	15 días	do 25/04/04	do 09/05/04
21	**Implementació**	**63 días**	**lu 05/04/04**	**do 06/06/04**
22	Desenvolupament dels casos d'us d'alta prioritat.	21 días	lu 05/04/04	do 25/04/04
23	Desenvolupament casos d'us de prioritat mitjana.	29 días	do 25/04/04	do 23/05/04
24	Desenvolupament casos d'us baixa prioritat	14 días	lu 24/05/04	do 06/06/04
25	PAC 3. Lliurament dels casos d'us d'alta prioritat	1 día	lu 26/04/04	lu 26/04/04
26	**Transició**	**35 días**	**do 09/05/04**	**sá 12/06/04**
27	Memòria del projecte	35 días	do 09/05/04	sá 12/06/04
28	Test de funcionament de l'aplicació.	20 días	do 23/05/04	vi 11/06/04
29	18 de juny. Lliurament final	1 día	vi 18/06/04	vi 18/06/04

Il·lustració 1. Planificació final del projecte

Queden resaltades les dues tasques que no s'han complert al 100%. La tasca de desenvolupament de casos d'us de baixa prioritat queda incomplerta per que no s'ha desenvolupat la utilitat de modificació de la fitxa personal dels usuaris. La tasca de test de funcionament de l'aplicació queda pendent de finalitzar el test de les pàgines HTML.

Si veiem aquestes dades en forma de diagrama de Gantt, tindrem una visió més general del projecte:



Il·lustració 2. Gantt de Planificació del projecte

Com s'indicava abans, es mostra que les tasques 24 i 28 queden incomplertes. La tasca 24 – Desenvolupament de casos d'us de baixa prioritat queda completada en un 95%, mentre la tasca 28 – Test de funcionament de l'aplicació queda completada al 80%.

Producte obtingut

El producte obtingut l'he anomenat *Xhelp OutsourcingService*. Aquest nom ve de l'aplicació que la eina que he desenvolupat te per a empreses d'outsourcing que fan servir un servei de HelpDesk per gestionar la comunicació amb els clients.

L'aplicació que he desenvolupat es una aplicació orientada a gestionar serveis d'aquest tipus, enregistrant les peticions que fan els clients al servei que tenen contractat, i també aquelles actuacions que els tècnics realitzen per tal de resoldre-les.

El producte es una aplicació desenvolupada amb tecnologies modernes: J2EE, Apache Struts i JBOSS-Tomcat.

J2EE es l'especificació Java per a aplicacions de servidor, que proporciona diverses eines per tal de desenvolupar aplicacions orientades cap a client-servidor.

Apache Struts es una plataforma desenvolupada per la institució Apache (que es dedica a establir estàndards en el mon Java). Aquesta plataforma pretén establir un mètode de treball per desenvolupar aplicacions web.

Jboss es un servidor d'aplicacions que te un contenidor d'EJB. Els EJB son una part fonamental de J2EE que dona suport a la lògica de negoci de les aplicacions i permet executar-la en un entorn transaccional i gestionat pel contenidor d'EJB. La versió de Jboss que he fet servir, porta integrat un servidor Tomcat. Tomcat, es un servidor d'aplicacions Java, que dona suport per a servlets i pàgines de servidor de Java (JSP).

Els detalls de l'arquitectura de l'aplicació i de l'entorn en que s'executa els veurem més endavant en la documentació pròpia del programari.

Contingut específic de la memòria

En aquesta memòria hi trobarem els documents que freqüentment es desenvolupen per tal de documentar un projecte de construcció de programari. Aquests documents es van creant a mida que el projecte avança, i normalment son els següents:

1. Document de Requisits del projecte
2. Document de definició d'actors i casos d'us.
3. Glossari de termes, per situar al lector en l'àmbit de l'aplicació.
4. Anàlisi dels Requisits de l'aplicació
5. Disseny dels requisits
6. Arquitectura
7. Manual d'usuari

L'aplicació Xhelp OutsourcingService.

Requisits funcionals

L'eina permetrà fer el seguiment de les peticions dels usuaris, de les accions que emprenen els tècnics per tal de portar-les a terme i d'uns nivells de servei prèviament acordats entre la empresa client i el servei de manteniment.

Els usuaris de la companyia client faran peticions, mitjançant una interfície web, un missatge de e-mail, una trucada telefònica o un fax al CAU (Centre d'assistència a l'usuari) de l'empresa de serveis. Si la petició entra al sistema per una via diferent de la web, el tècnic del CAU la introdueix, aportant tota la informació de que disposa. Un empleat d'aquesta, comprova les peticions (que poden ser dels següents tipus: evolutius, incidències, consultes i tasques periòdiques), i les assignarà a un grup de tècnics.

Una vegada una petició ha estat assignada, la petició passa a estar en estat "en procés", que indica a l'usuari demandant que la seva petició està sent atesa.

Mentre una petició està sent atesa, l'usuari demandant, i el cap del seu departament, poden accedir en qualsevol moment a consultar les accions que el/s tècnic/s porten a terme.

Cada cop que un tècnic realitza una acció sobre una petició es genera un registre. Aquest registre inclou dades com la data i hora de l'acció, l'objectiu que es pretén aconseguir i el temps invertit en realitzar-la. També es poden afegir "attachments" com ara dgcuments word, fulls de càlcul, etc.

Un tècnic pot canviar l'estat d'una petició, deixant-la com a "En espera", si necessita fer alguna consulta a l'usuari, i no pot continuar treballant en aquesta petició fins que no tingui resposta, o be com a "Oberta", el qual informa que està realitzant alguna acció sobre la petició.

Una vegada la petició es considera finalitzada, el tècnic pot realitzar el seu tancament. La petició romandrà pendent de confirmació del tancament per l'usuari.

Els nivells de servei es mesuren en temps de resposta i resolució de les peticions. El client contractarà amb el cap del servei de manteniment els nivells de servei que vol obtenir. El cap del servei de manteniment obtindrà un informe mensual sobre el compliment d'aquests nivells de servei on s'hi reflexa el temps invertit en total per resoldre totes les peticions d'aquell mes.

D'aquesta manera el cap del servei pot saber el cost en unitats de temps que suposa el servei. Un exemple de contracte podria ser:

- Temps de resolució per a peticions de tipus "Consulta": 4 hores.
- Temps de resolució per a peticions de tipus "Incidència:" 16 hores per a peticions de prioritat "Baixa", 8 per a prioritat "Alta" i 4 per a prioritat "Molt Alta".
- Les peticions de tipus Evolutiu i Tasca Periòdica, degut a que han de ser planificades no tenen SLA (Service Level Agreement).

Considerarem sempre, que si una petició, una vegada arribada al tècnic, necessita més temps del que hi consta al contracte de nivells de servei, el seu tipus haurà de canviar a "Evolutiu". El tècnic hi farà un tancament administratiu de la petició. Una vegada oberta com a "Evolutiu" el tècnic proposarà un temps de resolució, el qual podrà ser aprovat o denegat pel client. En cas de denegar-lo es tancarà la petició i es considerarà "desestimada".

Les peticions rebudes estaran monitoritzades pel sistema en forma de semàfors. En funció del temps restant per a la resolució, el semàfor de cada petició podrà estar en color verd, si encara resta més del d'una hora per arribar al temps màxim contractat, taronja si queda menys d'una hora i vermell si el temps contractat ha estat superat. En aquests càlculs no es tenen en compte els períodes en que la petició està en estat "En espera" o en qualsevol altre estat que no imputi temps per a l'SLA.

Glossari de Termes.

Dins el llenguatge utilitzat a l'aplicació, es considera útil definir els següents conceptes:

Projecte d'outsourcing: Contracte entre la empresa client i la empresa de serveis d'outsourcing pels quals, la empresa client delega tota la responsabilitat tècnica dels elements dels seus sistemes informàtics objecte del contracte sobre la empresa d'outsourcing.

Nivell de Servei ó S.L.A.: (Service Level Agreement) A l'inici del projecte s'acorden uns nivells de resposta per al servei. Per exemple, s'indica que una consulta haurà de ser resposta abans de 4 hores a partir de la seva recepció.

Petició: Comunicació del client per tal de realitzar alguna acció sobre el projecte d'outsourcing.

Incidència: Error de programari, o de maquinari que impedeix la realització de determinades tasques habituals al negoci del client. Es dona per fet, que la incidència recau sobre algun dels elements objecte del servei d'outsourcing.

Evolutiu: Sol·licitud de canvi a algun dels elements de programari o maquinari per tal d'adaptar-los a noves situacions dins el negoci del client. També inclou qualsevol modificació necessària per corregir el funcionament d'una aplicació que, per la seva complexitat resulta impossible solucionar dins dels paràmetres de SLA contractats per incidències.

Tasca Periòdica: Tasques que el client desitja contractar al servei d'outsourcing que impliquen planificació temporal i una repetició periòdica de les tasques. Per exemple, planificació i monitorització de còpies de seguretat semanals.

Consulta: Petició d'informació sobre algun dels elements objecte del servei, tant des del punt de vista funcional com tècnic.

Definició d'actors i casos d'us

A continuació s'especifiquen els diferents actors que hi intervenen en el projecte, així com les seves responsabilitats dins el sistema.

ACTORS		
Nom	*Descripció*	*Responsabilitats*
Responsable Outsourcing del Client	Es la persona al client que es responsabilitza de coordinar el servei d'outsourcing amb el responsable intern de la companyia de serveis. Generalment es tracta del Controller, o be, del cap del departament d'informàtica.	Acorda, junt amb el responsable del servei, els nivells de servei del projecte d'outsourcing. Aprova, mensualment, l'informe SLR (Service Level Report) que li proporciona el sistema. Designa els responsables de departament de cara al projecte d'outsourcing.
Responsable de departament del client	Es la persona encarregada, a cada departament del client, de prioritzar els evolutius i les tasques periòdiques.	Sol·licita evolutius i tasques periòdiques al servei d'outsourcing Aprova les valoracions, en temps, proporcionades pels tècnics d'outsourcing.
Usuari Client	Qualsevol empleat del client	Pot comunicar, directament, incidències relacionades amb el servei d'outsourcing. Aprovarà les resolucions proporcionades pels tècnics d'outsourcing a les incidències que ell planteja.
Cap del projecte d'outsourcing	Responsable intern de la empresa de serveis de l'outsourcing d'un o més projectes.	Acorda, junt amb el Responsable d'outsourcing del client els nivells de servei del projecte. Controla i gestiona la marxa del servei i el compliment dels nivells acordats. Crea i gestiona els equips de tècnics i de CAU.
Tècnic de CAU	Empleat de la empresa de serveis que proporciona assistència a l'arribada de peticions al sistema, rep trucades telefòniques, fax, mails, etc.	Rep sol·licituds per vies externes a l'aplicació (fax, mail, telèfon) i les introdueix al sistema. Fa una primera assignació de les peticions a un grup de tècnics i/o a un tècnic en concret.
Tècnic d'Outsourcing	Empleat de la empresa de serveis que resol les peticions arribades al CAU	Avalua les peticions arribades al cau Resol les peticions Pot comunicar-se amb l'usuari final per tal d'obtenir detalls sobre una petició en concret.

Il·lustració 3. Actors de XHelp OutsourcingService

Els casos d'us identificats directament dels requisits son els següents. A la taula, s'han prioritzat per importància, de manera que els marcats com *imprescindibles* i *d'alta importància*, no poden faltar del producte final. Els marcats com opcionals, es desenvoluparan en funció de les possibilitats que doni la planificació final.

Prioritat		Cas d'Us	Descripció
	1	Comunicació Peticions	Introducció al sistema de peticions per part del client, o del tècnic de HelpDesk.
	2	Recepció Peticions	Assignació de tècnics a una petició, i enviament de la primera resposta al client.
	3	Resolució de Peticions	Enregistrament de les accions que es porten a terme per resoldre una petició. Adjunció de documentació associada a la petició. Control del compliment de l'SLR mitjançant semàfors indicatius.
	4	Tancament de Peticions	Comunicació al client de l'estat de la petició, una vegada rebuda l'autorització per tancar la petició, aquesta queda com resolta.
	5	Configuració Projecte d'outsourcing	Designació dels nivells de servei en funció dels tipus de peticions (Incidències, evolutius, tasques periòdiques i consultes). Creació de grups funcionals per la gestió i obtenció d'informes. (per ex. àrea tècnica, àrea funcional, etc.)
	6	Administració Equip d'Outsourcing	Gestió d'equips de treball d'outsourcing, creació d'usuaris tècnics i de helpdesk i la seva assignació a equips concrets.
	7	Administració equip client	El responsable del client delimitarà responsabilitats per fer sol·licituds d'evolutius i incidències al servei d'outsourcing. Aquesta tasca pot fer-la conjuntament amb el cap del servei d'outsourcing, però la responsabilitat del resulta caurà sobre ell.
	8	Obtenció Informes Servei (SLR)	Informes per el control i gestió de la marxa del servei.

Il·lustració 4. Classificació dels casos d'us per prioritats

Imprescindible	Alta importància	Opcional

Classificació dels Actors del sistema.

Els usuaris tenen uns rols assignats que els hi donaran accés a les diferents funcionalitats de l'aplicació.

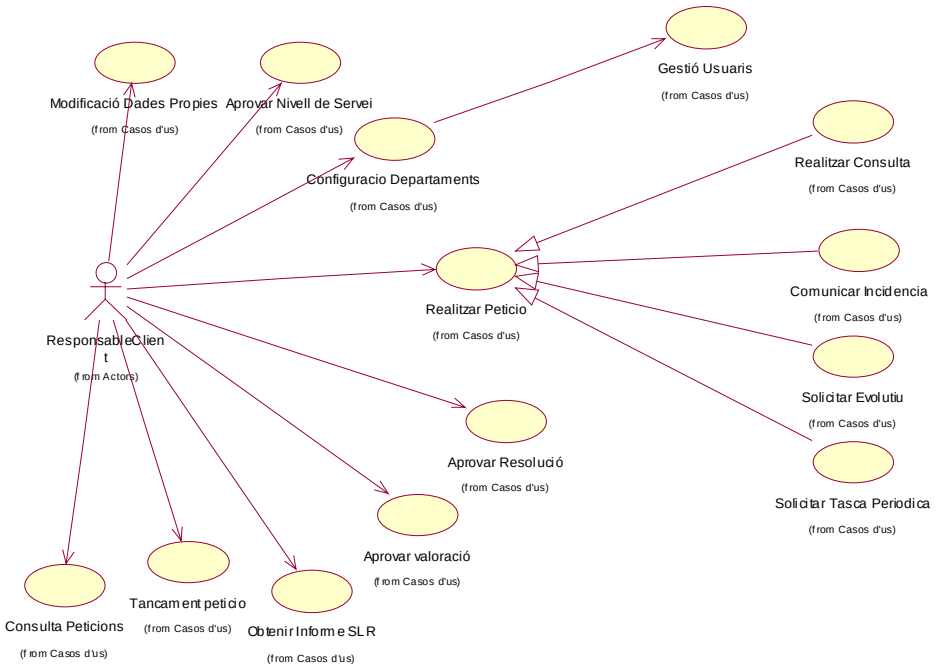
Es contemplen 6 tipus d'usuari diferents:

- Responsable del client. Es tracta de la persona que contracta els serveis de X-help. Pot realitzar qualsevol tipus de petició.
- Cap de departament. Es tracta d'un responsable de departament de l'empresa client. Aquest cap de departament pot demanar qualsevol tipus de petició.
- Usuari bàsic. Es tracta de l'usuari més habitual de l'aplicació que només podrà demanar consultes i comunicar incidències. D'aquesta manera es canalitza el manteniment funcional de les aplicacions en els caps de departament.
- Cap Outsourcing. Es tracta de la persona responsable de X-help de acordar els detalls del servei amb el responsable del client. Configurarà els equips de tècnics i obtindrà els informes SLR (Service Level Report) mensuals. Aquest usuari no pot fer peticions al sistema. La idea es que canalitzi possibles necessitats en el client, o en algun dels tècnics del projecte.
- Tècnic CAU. La finalitat d'aquest usuari es rebre peticions per mitjans externs a l'aplicació i introduir-les al sistema. Pot realitzar accions sobre peticions. Pot introduir qualsevol tipus de petició al sistema, però sempre en nom d'un altre usuari.
- Tècnic. Aquest usuari no pot realitzar peticions al sistema. La seva intervenció es limita a la realització d'accions resolutòries.

En una primera aproximació al sistema, es presenten els diagrames de casos d'us detallats de cada actor. A continuació, es realitza una descripció de cada cas d'us.

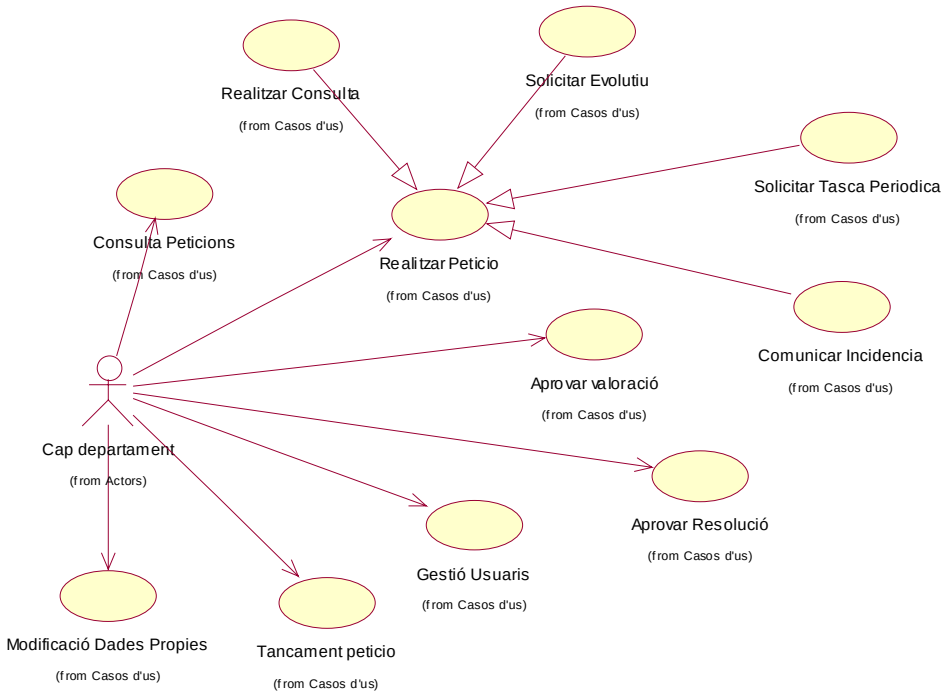
Casos d’us dels actors.

Casos d’us del Responsable del Client



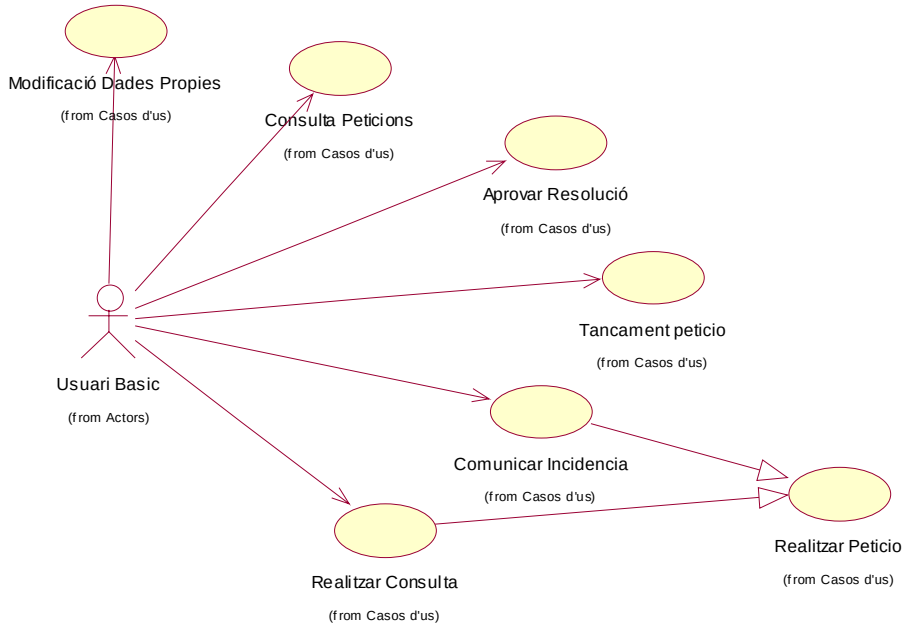
Il·lustració 5. Casos d'us del Responsable del client

Casos d'us del cap de departament



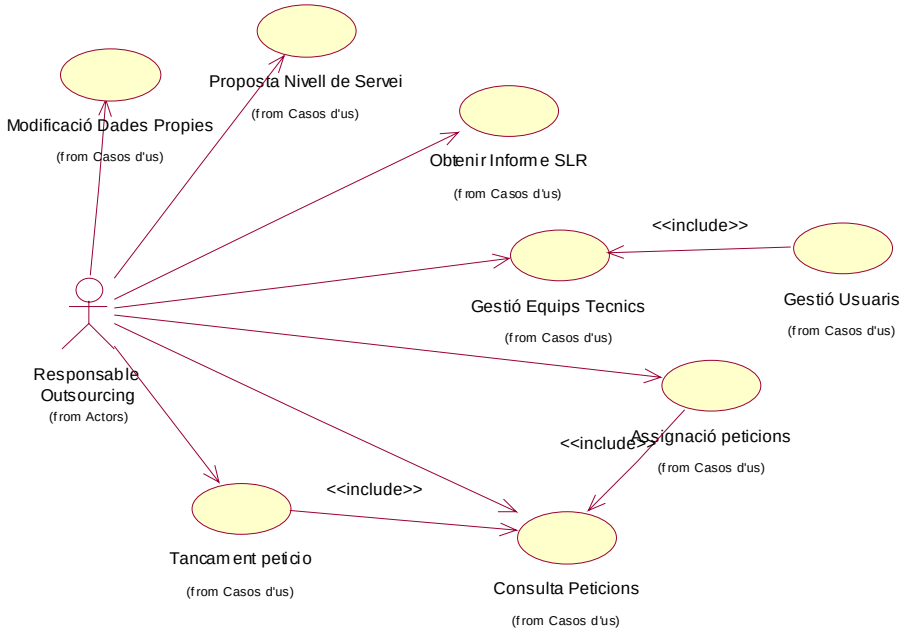
Il·lustració 6. Casos d'us d'un cap de departament

Casos d'us de l'usuari bàsic



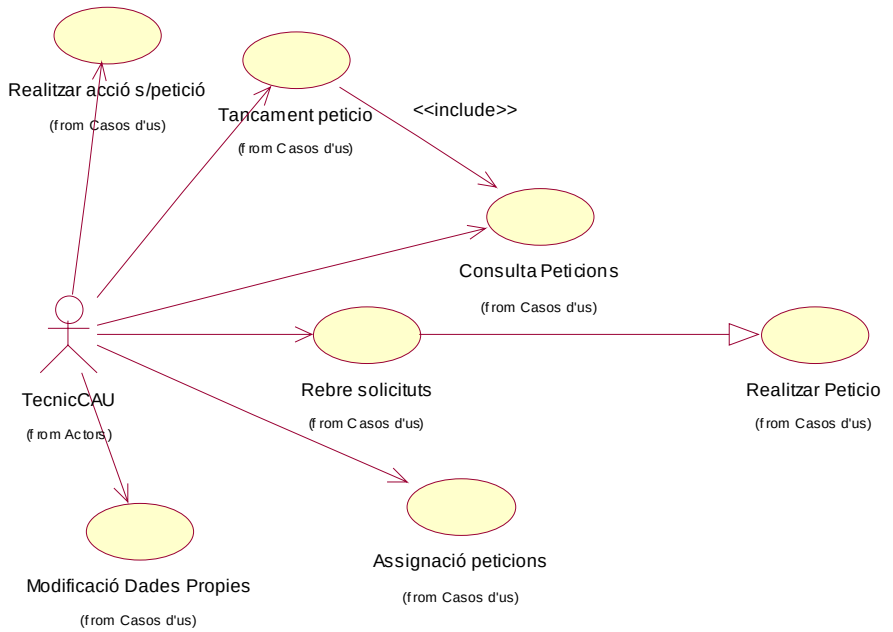
Il·lustració 7.Casos d'us d'un usuari bàsic

Casos d'us del responsable d'outsourcing



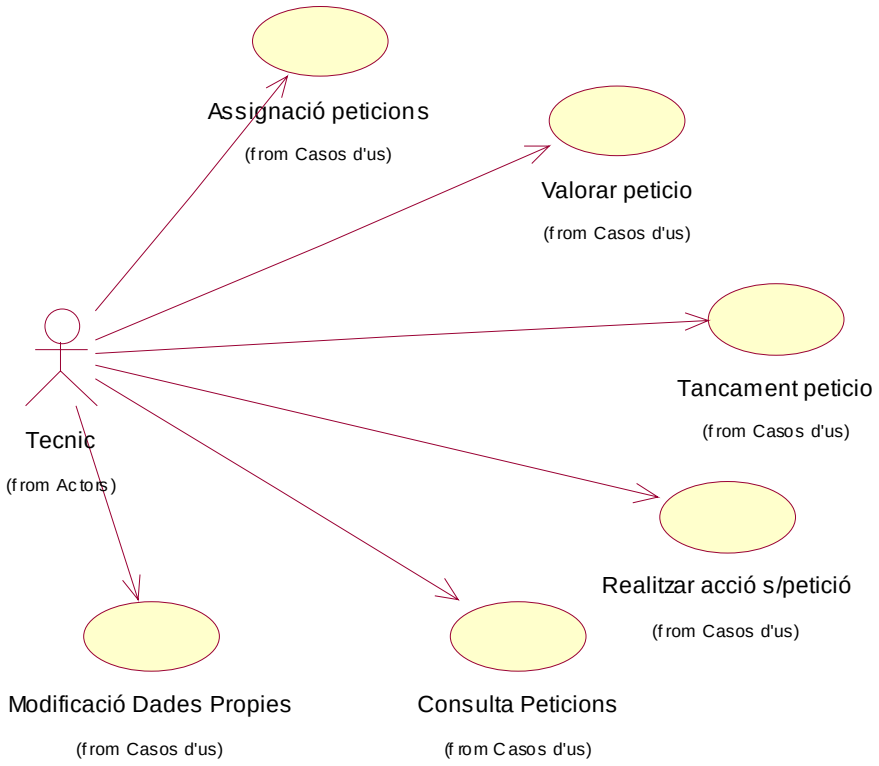
Il·lustració 8. Casos d'us del Responsable d'outsourcing

Casos d'us del tècnic del CAU



Il·lustració 9. Casos d'us del Tècnic del C.A.U.

Casos d'us del tècnic



Il·lustració 10. Casos d'us del Tècnic Resolutor

Descripció dels casos d'Us.

Comunicació de peticions al servei de HelpDesk.

Casos d'us associats:

- Petició de consultes
- Petició d'evolutius
- Petició de tasques periòdiques
- Petició d'Incidències
- Rebre sol·licitud

Aquests casos d'us es poden considerar especificacions del cas d'us genèric "Realitzar Peticions".

Els actors implicats en aquest cas d'us son:

- Responsable client
- Cap departament
- Usuari bàsic
- Tècnic CAU

Com es pot observar als diagrames, es poden fer quatre tipus diferents de peticions al HelpDesk:

- Consultes
- Incidències
- Evolutius
- Tasques periòdiques

Les *consultes* son peticions de les quals l'única actuació esperada per part dels tècnics, o del CAU es una petita investigació i una resposta en format de fitxer de text (word, excel, etc., però mai desenvolupaments). Aquestes peticions tenen un caràcter urgent, i s'han de resoldre en poc temps (determinat segons el contracte de nivells de servei).

Les *incidències*, son errors en el programari o maquinari que requereixen un esforç de desenvolupament limitat (també contemplat al contracte de nivells de servei, des d'ara SLA) o de gestions per part de l'equip tècnic per tal de posar-se en contacte amb l'empresa que porti el manteniment tècnic i el seguiment de la resolució de la incidència. Els tècnics de X-help no realitzaran accions de manteniment de maquinari personalment.

Els *evolutius*, son millores al programari o maquinari, amb l'objectiu, be de potenciar el seu us o de resoldre problemes derivats d'errors greus, que donada la seva complexitat no poden considerar-se *incidències*.

Les tasques periòdiques son tasques que s'han d'executar en determinats moments i de les quals no s'esperen problemes que esdevinguin incidències o responsabilitats per part dels tècnics de X-help, per exemple, realització de backup, execució de processos en caps de setmana, etc.

En termes bàsics, la realització d'una petició segueix les següents passes:

L'usuari es connecta al sistema i accedeix a una pantalla d'introducció de peticions. En funció dels rols assignats a l'usuari la pantalla li permetrà introduir els diferents tipus de peticions.

Totes les peticions comparteixen un conjunt de dades bàsic:

- Data i hora de la petició
- Tipus de petició
- Element de programari o maquinari per a la qual es fa la petició
- Descripció breu
- Descripció llarga
- Fitxers adjunts (es fixa un límit de 5 fitxers)
- Usuari demandant.

Una vegada introduïdes les dades, el sistema informarà a l'usuari. L'usuari podrà consultar el llistat de les peticions en qualsevol moment.

Les peticions no es poden modificar mai. En tot cas es poden resoldre administrativament per acord amb l'usuari i tornar a obrir una de nova.

El cas d'us "Rebre sol·licitut" queda desestimat, i es cobrirà mitjançant el cas d'us "Realitzar petició". El motiu es per que el tècnic del CAU no sempre pot conèixer la identitat del demandant, doncs de vegades es fan servir comptes de e-mail corporatius.

Consulta de peticions.

Casos d'us relacionats: Cap
Actors que hi intervenen: Tots

Els usuaris poden accedir al sistema en qualsevol moment per consultar l'estat de les peticions que tenen obertes.

A partir d'aquesta llista es pot consultar la petició, descarregar els fitxers associats, veure les actuacions realitzades, etc.

Assignació de peticions.

Casos d'us relacionats: Rebre sol·licituds

Actors que hi intervenen: Cap outsourcing, Tècnic CAU, Tècnic.

Una vegada s'ha introduït una petició els tècnics CAU faran una primera assignació de l'equip de tècnics (cas d'us Rebre sol·licitud) que s'encarregarà de resoldre la petició. Aquesta assignació la faran en base a unes premisses acordades verbalment a l'engegament del projecte.

Les assignacions no es faran personalment, es a dir, no s'assignaran peticions a un únic tècnic, si no a un grup de tècnics. D'aquesta manera s'eviten problemes de disponibilitat d'una persona en concret.

Gestió d'usuaris.

Casos d'us relacionats: Rebre sol·licituds, Modificació dades pròpies.

Actors que hi intervenen: Cap outsourcing, Cap Departament, Responsable Client.

La gestió d'usuaris permet crear, modificar dades, i esborrar usuaris del sistema. Aquestes accions estan controlades pel sistema en funció dels rols que te l'usuari que intenta fer les accions. El Cap d'outsourcing pot gestionar usuaris de tipus tècnic i tècnic CAU, mentre que els usuaris amb rols "Cap departament" i "Responsable Client" poden crear usuaris bàsics, i usuaris bàsics i caps de departament respectivament.

Configuració de Departaments.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Responsable Client.

Una de les primeres tasques del responsable per part del client haurà de fer es configurar els departaments que integren la seva empresa, i assignar a un usuari (que haurà de crear ell mateix) com a cap de departament (aquest usuari estarà autoritzat a demanar evolutius).

Tancament de Peticions.

Casos d'us relacionats: Aprovar Resolució, Tancament de Peticions (de cada actor).

Actors que hi intervenen: Tots.

El tancament real d'una petició requereix que tots els usuaris implicats (l'usuari demandant i el(s) tècnic(s) resolutor(s)) la donin per tancada, en primer lloc el tècnic resolutor la donarà per tancada, l'usuari demandant haurà de confirmar la resolució de la petició. Si l'usuari demandant considera que aquesta no ha estat satisfactòriament resolta, pot anular el tancament del tècnic i posar la petició en estat "En curs" de nou.

Si un tècnic resolutor determina que no es necessiten accions per a una petició per que aquesta ha estat oberta per error, o per algun altre motiu per l'usuari demandant, ambdós, tancaran la petició seguint el mateix protocol, però el tècnic resolutor afegirà una acció en la qual farà el comentari de que la petició ha estat tancada per petició de l'usuari demandant. A aquesta acció se li anomena "Tancament administratiu".

Les tasques periòdiques no es consideren tancades fins que l'usuari ho demana expressament. A cada execució finalitzada, la petició roman en estat "Planificada", fins que l'usuari final indica, en el moment del tancament d'una execució que la tasca no s'haurà d'executar més.

Valoració de Peticions.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Tècnic.

Quan un tècnic rep una petició assignada, la primera acció que haurà de fer serà fer una valoració del temps necessari per tal de resoldre-la. Aquesta valoració haurà de ser aprovada per un cap de departament per tal d'engegar la planificació de l'execució de les tasques necessàries.

En un principi es suposava que hauria de ser el cap del departament demandant de la petició qui hauria d'aprovar les valoracions, però en moltes ocasions les peticions es realitzen conjuntament per diversos departaments. Es permetrà l'aprovació per qualsevol cap de departament, però aquesta acció generarà una "actuació" al sistema que identificarà la persona que ha fet l'aprovació.

Si parlem de peticions de tipus incidències, i com que aquestes peticions tenen caràcter urgent, l'aprovació de l'usuari demandant no serà necessària a menys que aquesta valoració sigui superior al temps màxim establert per considerar que una petició esdevé una incidència o un evolutiu (recordem que si la petició supera aquest temps màxim, haurà de ser planificada com a evolutiu i no com a incidència).

Aprovació de Valoracions de Peticions.

Casos d'us relacionats: Cap.

Actors que hi intervenen: usuari final.

Veure "Valoració de Peticions".

Realització d'accions sobre peticions.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Tècnic, Tècnic CAU.

Amb la finalitat de mantenir el sistema ben documentat durant tota la vida del servei d'outsourcing, i per tal de controlar les modificacions que es facin en el programari i el maquinari, els tècnics i els tècnics del CAU documenten la seva activitat en *Actuacions*. Aquestes actuacions podran anar acompanyades de documents adjunts que serveixin per documentar les solucions aportades. Les accions es faran servir també per calcular el temps total invertit en la resolució de cada petició.

Proposta de Nivells de Servei.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Cap Outsourcing, Responsable Client.

Aquest cas d'us serveix per establir la parametrització bàsica de l'aplicació.

El Cap d'outsourcing acordarà en una reunió presencial amb el Responsable del Client els nivells de servei contractuals. Una vegada arribat l'acord s'introdueix al sistema per el Cap d'Outsourcing. Aquesta parametrització inclou les dades:

- Temps màxim de resolució per a consultes
- Temps màxim de resolució per a incidències (un temps per a cada tipus de urgència: Molt urgent, urgent, normal, poc urgent, molt poc urgent)
- Temps màxim d'umbral entre incidències i evolutius. Aquest temps marcarà que les incidències obertes per els usuaris que tinguin una valoració superior a aquest temps esdevindran evolutius pendents d'aprovació pel cap de departament.

Aprovació de Nivells de Servei.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Cap Outsourcing, Responsable Client.

Els nivells de servei acordats per el responsable del client amb el cap d'outsourcing i introduïts per aquest darrer al sistema, rebran el vist i plau per el responsable del client abans d'engegar el servei. L'acció d'aprovar els nivells de servei es reflexa al contracte de manteniment, pel que no es considera necessari desenvolupar cap utilitat per reflexar això en el sistema.

Informe SLR Mensual.

Casos d'us relacionats: Cap.

Actors que hi intervenen: Cap Outsourcing.

El cap del servei d'outsourcing podrà obtenir un informe mensual que resumirà els nivells de servei que s'han ofert, on s'inclouran les dades de les peticions que hagin complert els nivells acordats, i es donaran dades estadístiques sobre temps de resposta i compliment dels objectius dels usuaris finals.

Especificació dels casos d'Us.

Diagrames de seqüències.

En aquest document es recullen els principals diagrames de seqüències, on es podrà comprovar que el model escollit per al sistema es basa en MVC en diverses capes.

Concretament definirem una sèrie de classes de frontera que s'encarregaran de gestionar les peticions dels usuaris. Aquesta gestió pot incloure en algun cas alguna decisió de nivell de lògica de negoci de primer nivell (com la decisió de mostrar o no determinades dades per pantalla en funció de l'usuari, o auto-informar determinats camps, etc.), però no serà mai contractual amb el sistema, es a dir, els següents nivells del model faran també les validacions necessàries per tal que el model de negoci no depengui mai d'una classe de frontera.

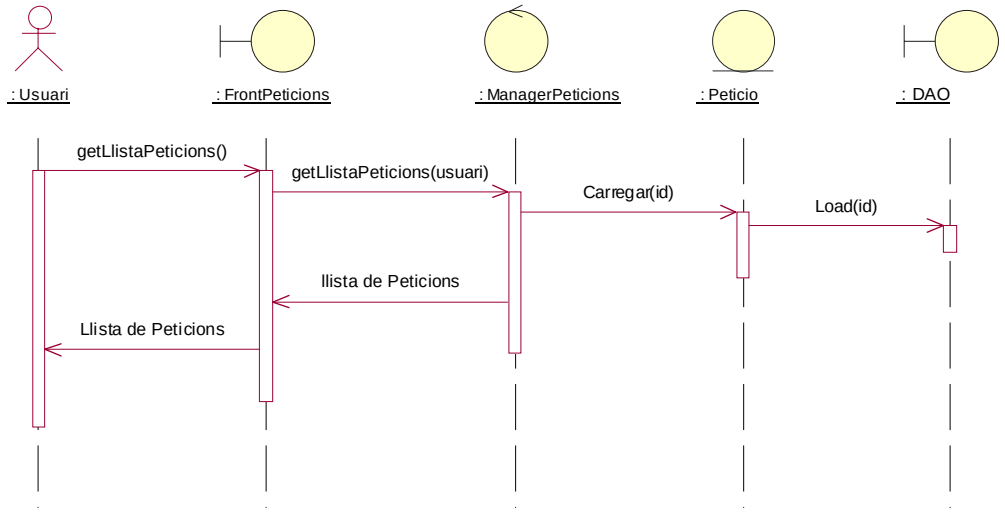
Un segon nivell es concentrarà en controlar els diversos casos d'us i en l'aplicació de la lògica de negoci. Aquest nivell rebrà les peticions de les classes de frontera i farà crides a la capa de negoci.

El següent nivell considera les entitats del sistema i la lògica de negoci.

Els casos d'us que s'ha considerat interessant especificar en aquest apartat, be per que serveixen d'exemple per altres molt semblants, be per la seva complexitat son els següents:

- Consulta de peticions
- Gestió d'usuaris
- Tancament de petició
- Realització d'acció sobre una petició
- Avaluació de peticions
- Realització de peticions
- Gestió de departaments

Diagrama de seqüències de: Consulta de Peticions.

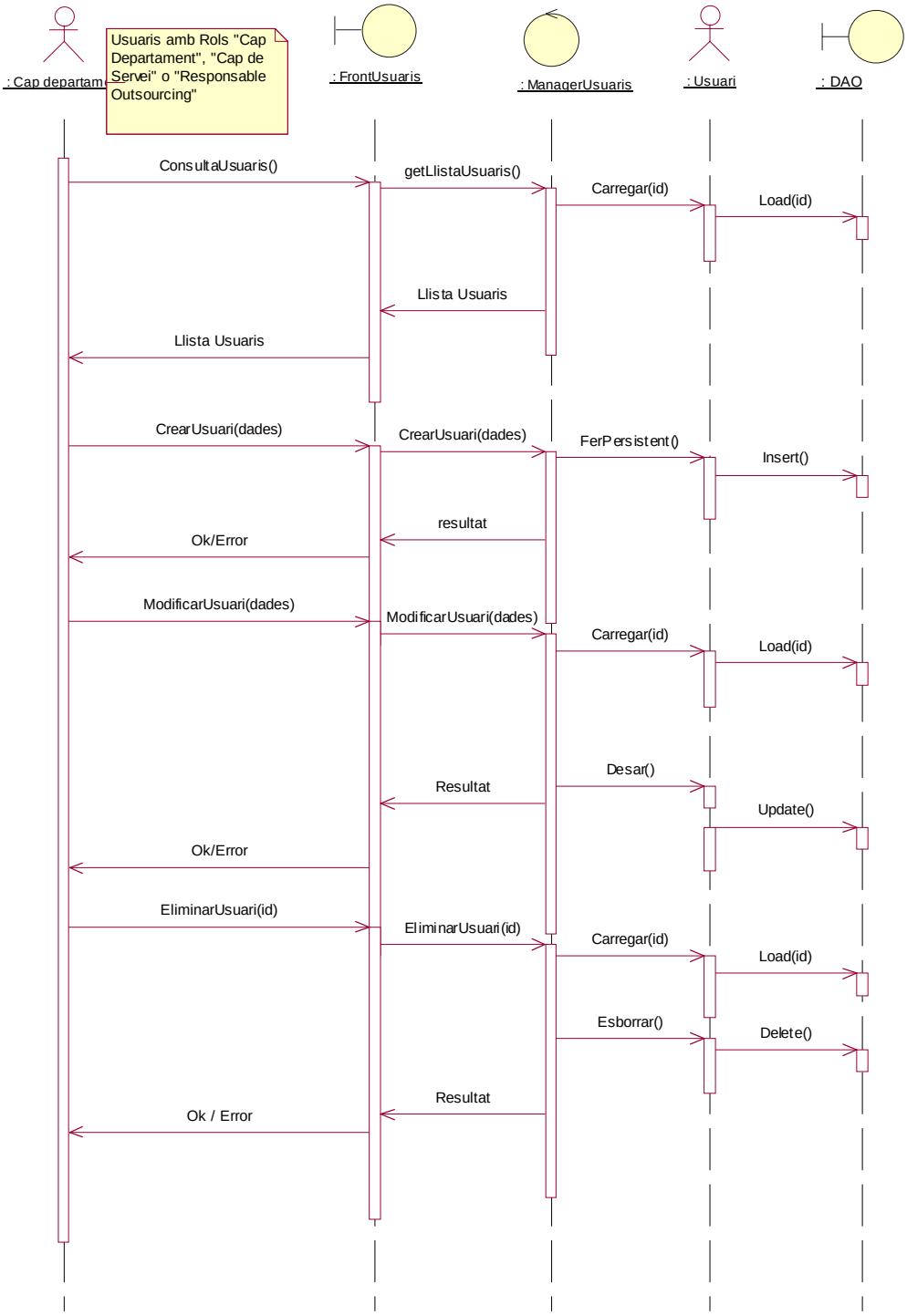


Il·lustració 11. Diagrama de seqüències **Consulta de Peticions**

Com es pot veure, el model que es farà servir es MVC. En aquest exemple, podem veure com la Vista be donada per una classe de frontera (probablement una pàgina JSP, veure el document de disseny per obtenir detalls). Com a model, les classes d'entitat faran aquest rol, delegant en un nivell addicional (en el model s'ha anomenat DAO) la persistència.

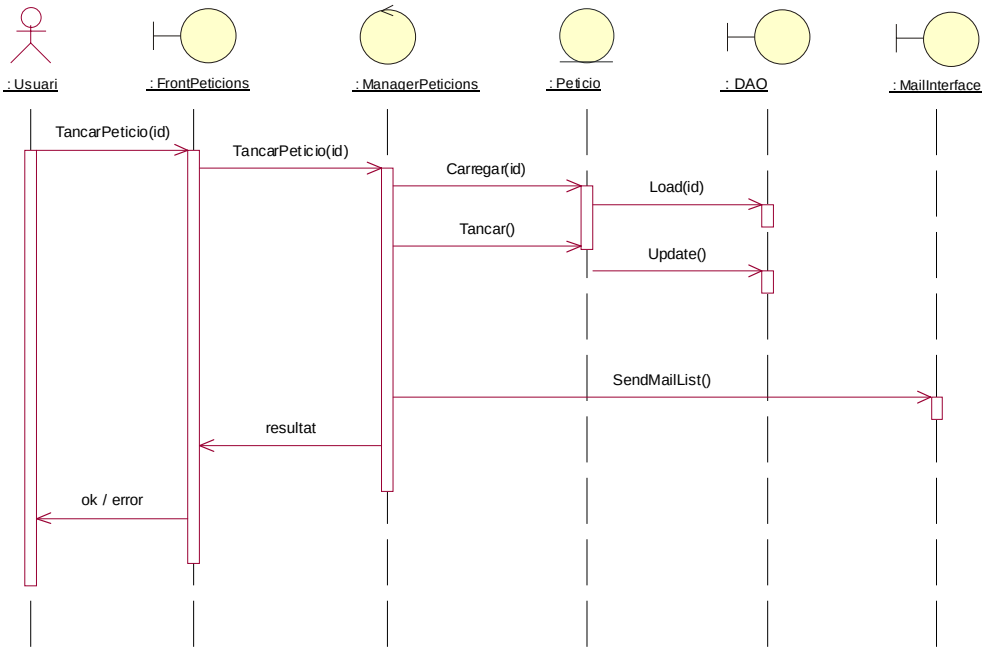
Com a controladors, es fan servir classes on es desenvoluparà la lògica de negoci.

Diagrama de seqüències de: Gestió d'usuaris.



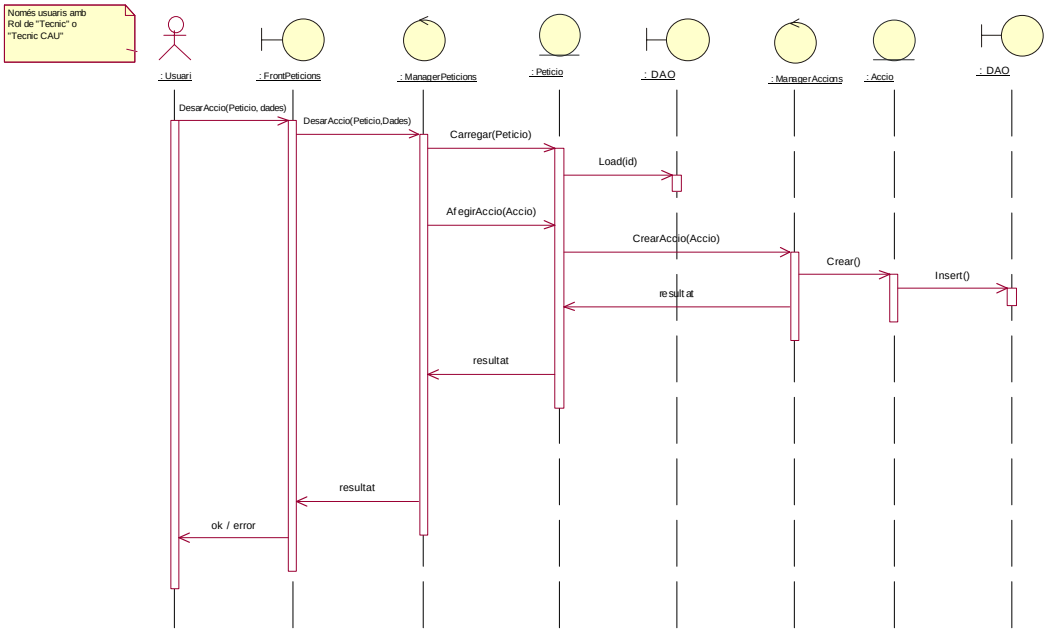
Il·lustració 12. Diagrama de seqüències **Gestió d'usuaris**.

Diagrama de seqüències de: Tancament de peticions.



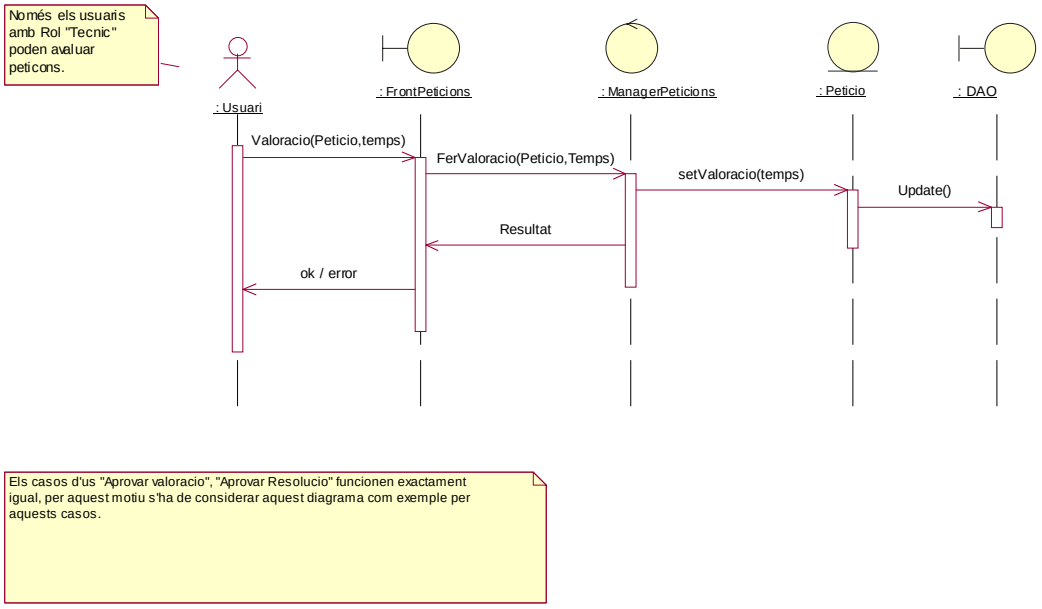
Il·lustració 13. Diagrama de seqüències **Tancament de peticions.**

Diagrama de seqüències de: Accions sobre peticions.



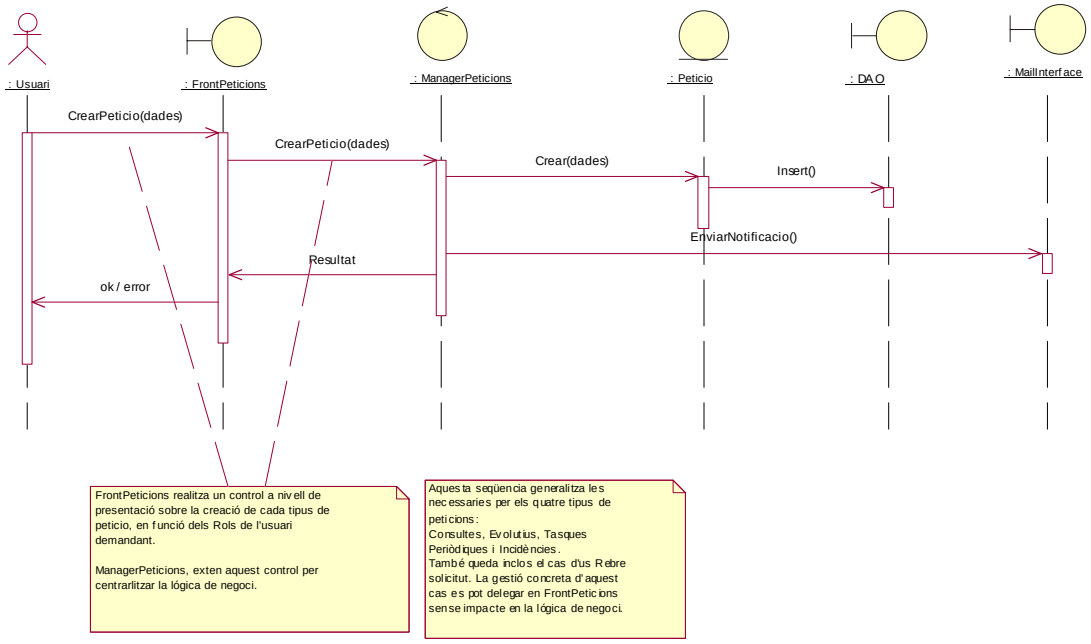
Il·lustració 14. Diagrama de seqüències **Actuacions sobre peticions.**

Diagrama de seqüències de: Avaluació de peticions.



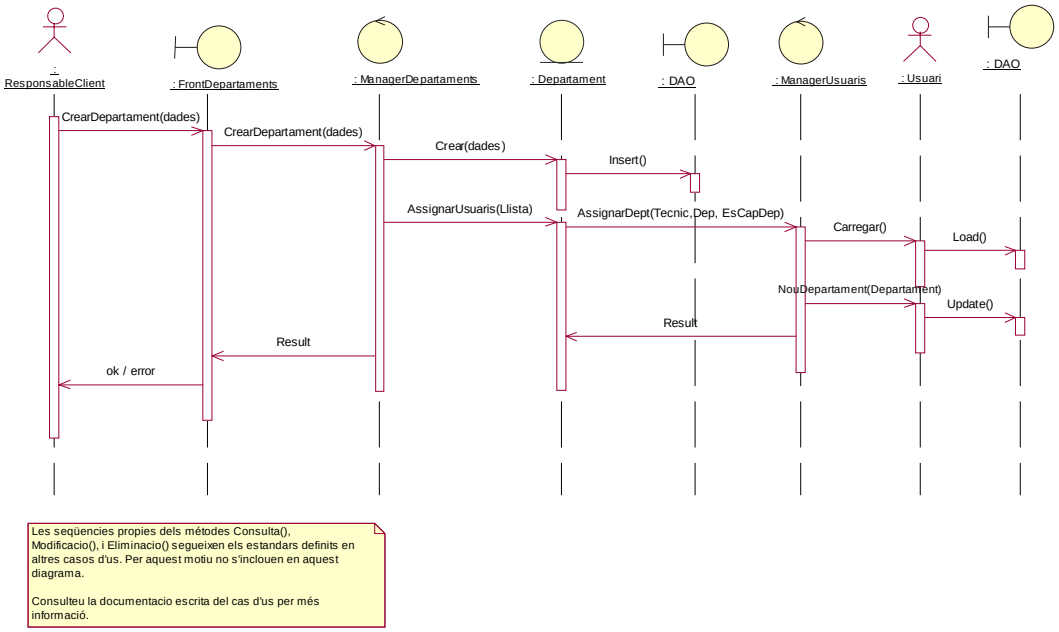
Il·lustració 15. Diagrama de seqüències **Avaluació de peticions.**

Diagrama de seqüències de: Realització de peticions.



Il·lustració 16. Diagrama de seqüències **Realització de peticions.**

Diagrama de seqüències de: Gestió de departaments.



Il·lustració 17. Diagrama de seqüències **Gestió de departaments.**

Disseny tècnic del sistema

Entitats derivades de l'anàlisi.

A continuació es dona la llista d'entitats detectades en l'anàlisi de requisits acompanyades d'una breu descripció.

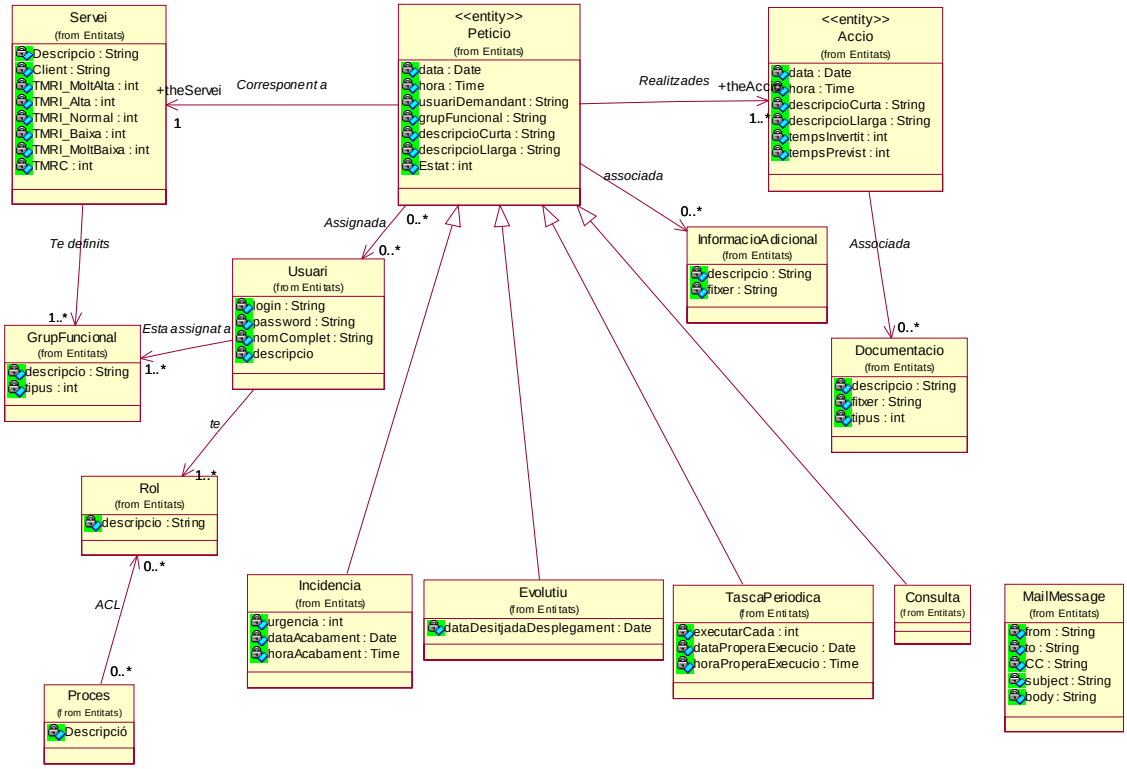
- Usuari: Modela un usuari del sistema.
- Rol: Els usuaris poden tenir diversos rols, els rols poden ser tants com actors tenim al sistema.
- Petició: Cada sol·licitud que un usuari client fa al sistema es una Petició, aquesta petició pot ser de diferents tipus:
    - Incidència: (veure document d'anàlisi)
    - Evolutiu: (veure document d'anàlisi)
    - Consulta: (veure document d'anàlisi)
    - Tasca Periòdica: (veure document d'anàlisi)
- Documentació: Modela els arxius adjunts que pot tenir una Acció.
- Informació Addicional: Arxius adjunts que pot tenir una Petició.
- Actuació: Modela les activitats realitzades pels tècnics sobre una petició.
- Servei: Tant els usuaris, com les peticions que es realitzen estan vinculats a un servei, doncs l'aplicació pot gestionar més d'un servei. Inclou la parametrització de nivells de servei.
- Grup Funcional: Agrupa usuaris en departaments, o grups (per als tècnics).

Com s'ha pogut deduir de la presa de requeriments i del posterior anàlisi dels requisits, la entitat Petició es la més complexa i el centre de tota la funcionalitat del sistema.

S'ha cregut important especificar els diferents estats en que es pot trobar una petició, així com el camí que aquesta segueix des del moment en que es sol·licitada fins que es considera resolta.

Com hem vist, s'han deduït quatre subclasses de la superclasse Petició. Cadascuna d'aquestes subclasses te associat un diagrama d'estats que exemplifica el seu comportament.

Diagrama de Classes.



Il·lustració 18. Diagrama de classes

Arquitectura de la Persistència i ús d'EJB d'entitat.

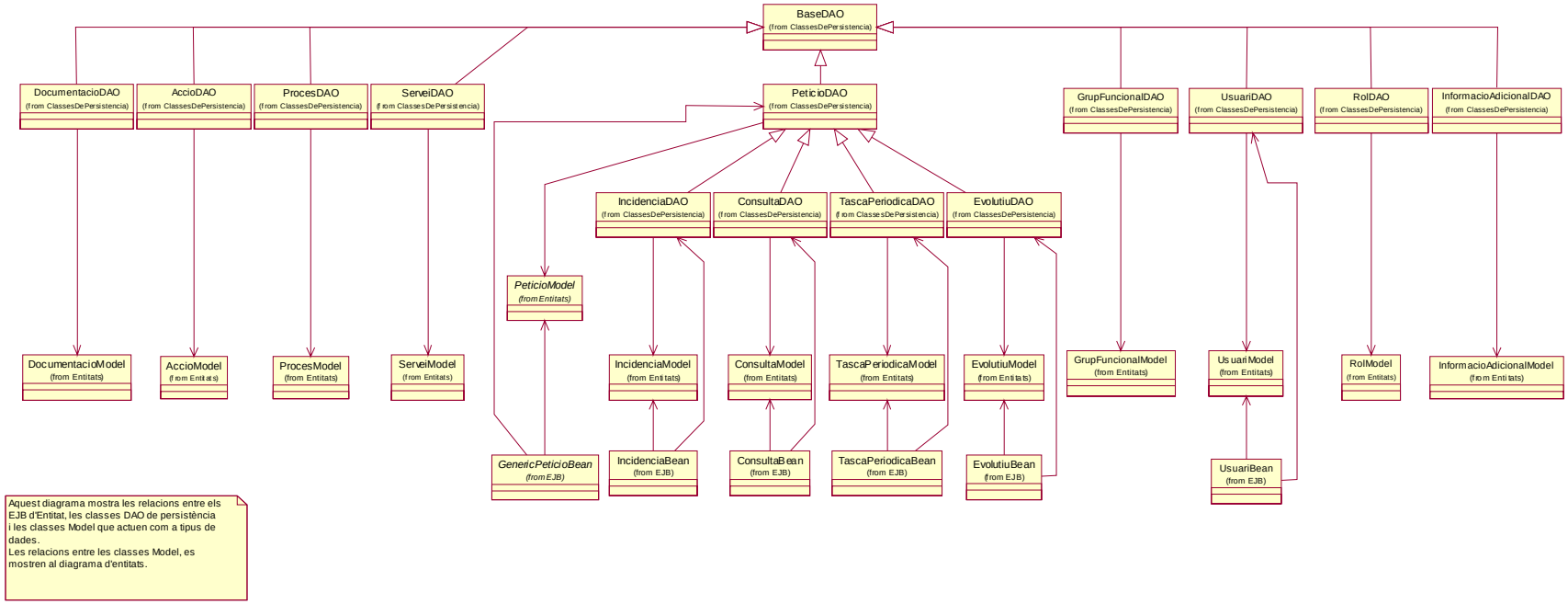
Els EJB d'entitat permeten mantenir objectes remots en entorns diferents i accedir-hi a ells mitjançant interfícies remotes. Aquest tipus d'accessos fa un us relativament intensiu dels recursos de xarxa del servidor on s'executen, tant el client d'EJB com el servidor.

Per aquest motiu, no totes les entitats detectades en l'anàlisi dels requisits s'implementaran com a EJB d'entitat, si no que s'ha fet una selecció d'aquelles entitats que, per possibles motius funcionals futurs, podrien haver-se d'executar en un entorn distribuït.

La implementació de la persistència dels Enterprise Java Beans s'ha determinat desenvolupar-la mitjançant el model BMP (Bean Managed Persistence) i fent servir un patró de disseny molt popular, DAO (Data Access Objects). Aquest patró desenvolupa una nova capa de persistència que permet la total separació del codi dels EJB dels detalls de la base de dades i la interfície JDBC.

Bàsicament es composa d'una classe que encapsula tot l'accés a base de dades per una altra que implementa el model de dades. D'aquesta manera, l'EJB d'entitat només ha de gestionar el seu model de dades i enviar missatges a la classe DAO apropiada per obtenir-ne els serveis de persistència.

D'aquestes conclusions es deriva el següent diagrama:

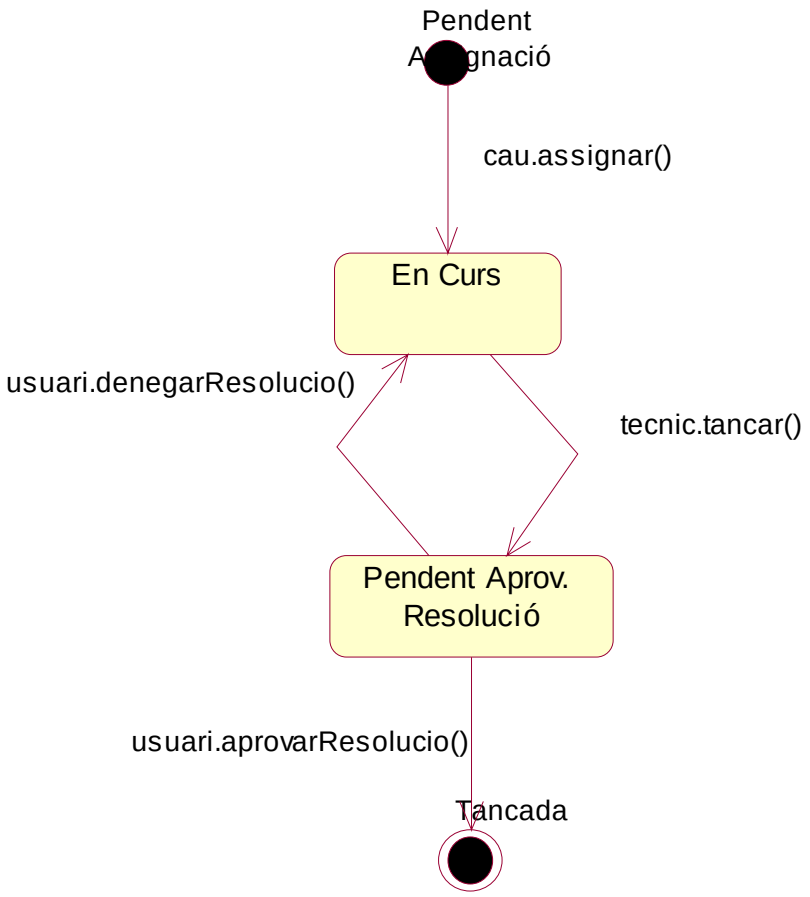


Il·lustració 19. Diagrama de classes EJB i disseny de la persistència.

Per motius de complexitat no es mostren totes les entitats del sistemes, només les més importants funcionalment, i els Enterprise Java Beans. Totes les entitats de Xhelp fan servir exactament la mateixa arquitectura de persistència que les que es mostren en aquest diagrama.

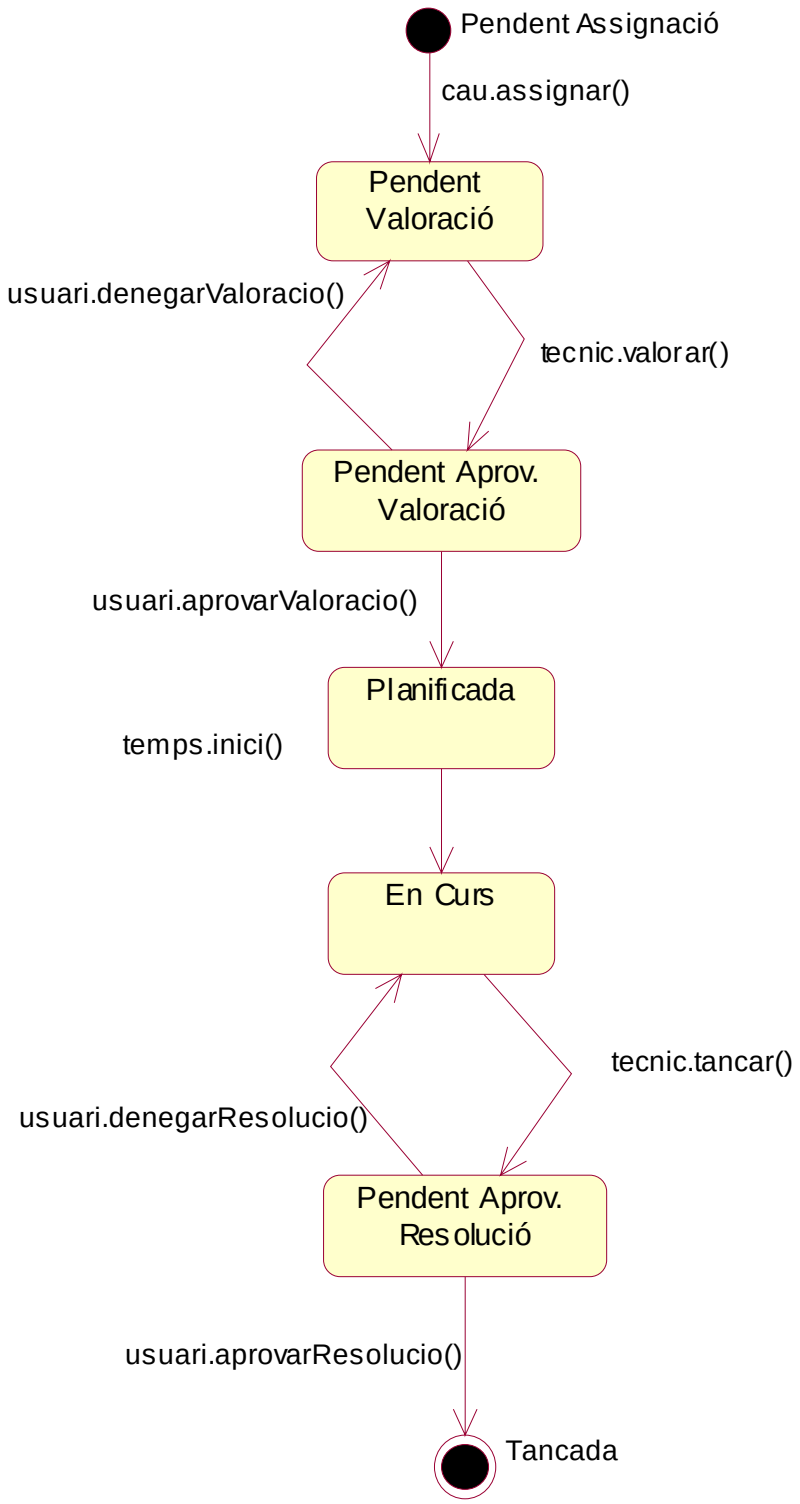
Diagrames d'estat

Diagrama d'Estats d'Incidència



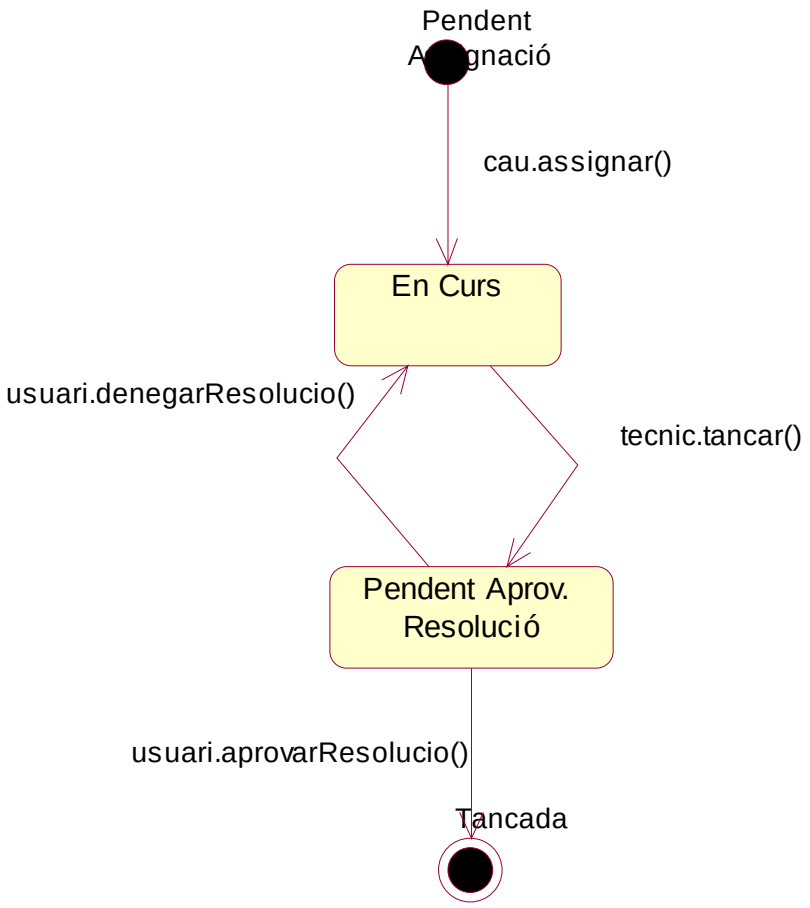
Il·lustració 20. Diagrama d'estats de **Consulta**.

Diagrama d'Estats d'Evolutiu



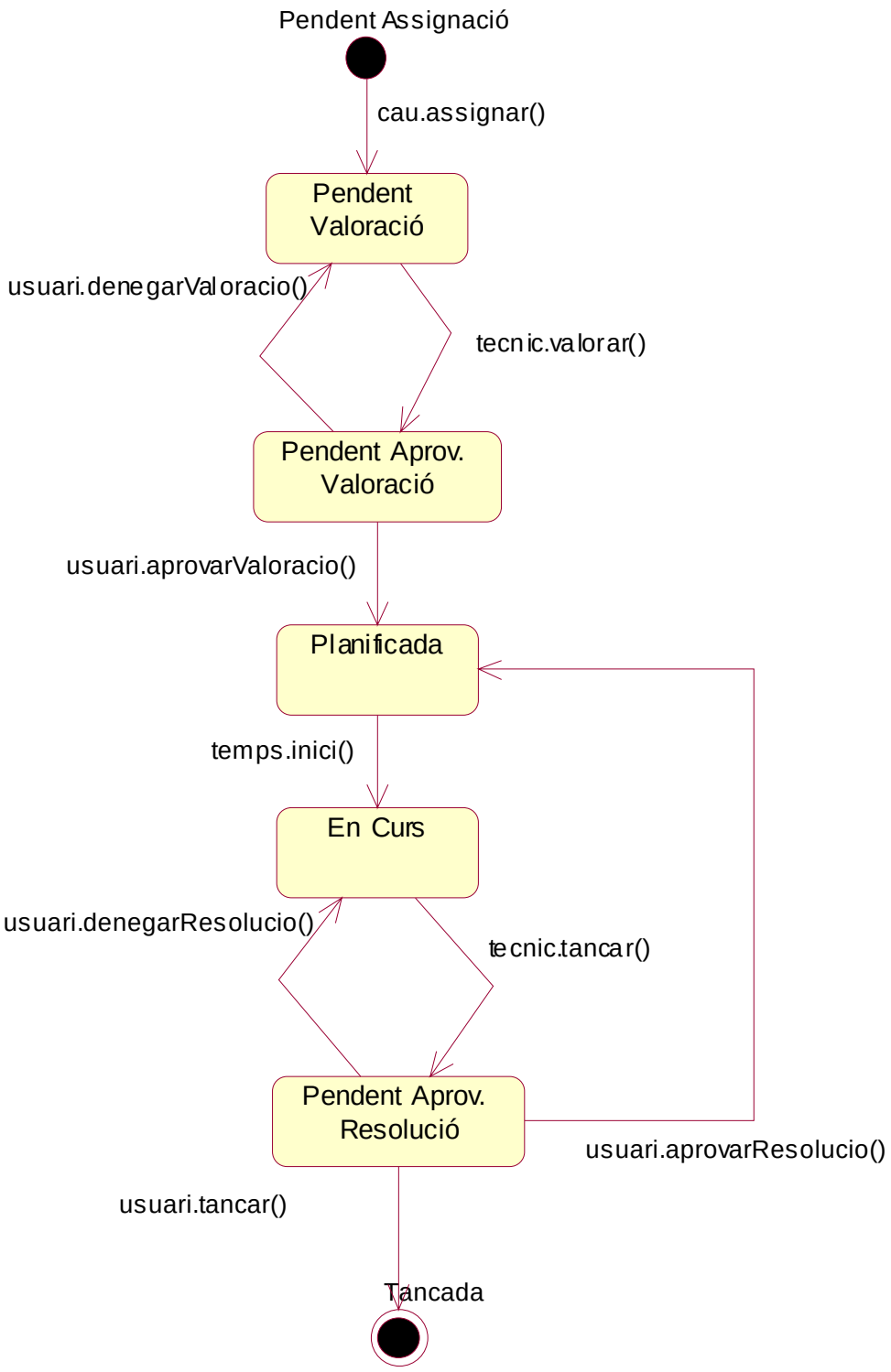
Il·lustració 21. Diagrama d'estats d'**Evolutiu**.

Diagrama d'estats de Consulta



Il·lustració 22. Diagrama d'estats d'**Incidència**.

Diagrama d'estats de Tasca Periòdica



Il·lustració 23. Diagrama d'estats de **Tasca periòdica**.

Com es pot veure, el comportament de cada tipus de petició es diferent. Per donar la màxima flexibilitat a aquesta circumstància, s'ha desenvolupat una petita màquina d'estats que gestiona el flux de cada tipus de petició mitjançant dades parametritzades a la base de dades. Aquestes dades inclouen permisos d'execució per cada rol de l'aplicació.

Disseny de l'arquitectura

El patró de disseny MVC (Model view controller). Apache Struts.

Un dels patrons més utilitzats en el desenvolupament de aplicacions web es el patró MVC.

Aquest patró va ser originat en el mon SmallTalk i defineix la disposició de les classes, de les seves responsabilitats i les seves relacions com a solució reutilizable.

Com el seu nom indica està dividit en tres tipus de components en funció de la seva responsabilitat en el sistema:

- El Model: Es el component de dades i lògica de negoci. Aquest model de dades pot tenir diverses vistes (interfícies).

- La Vista: Es el component de presentació de dades a l'usuari. Una mateixa vista pot donar diferents presentacions de dades als usuaris. Rep dades del model que mostra a l'usuari.

- El Controlador: Es el component que respon a les entrades de l'usuari, tradueix events d'interfície d'usuari en crides al Model i defineix com reacciona la interfície d'usuari amb aquest canvis.

Fent servir aquest patró s'obtenen grans avantatges:

- Les aplicacions son més flexibles davant futurs canvis. De fet, permet modificar completament el disseny d'una aplicació web, sense que això afecti al Model. O fins i tot, realitzar canvis d'implementació a la lògica i al Model, sense que sigui necessari modificar el disseny de la interfície d'usuari.

- Els programadors i els dissenyadors gràfics, o els maquetadors poden treballar independentment, tan sols respectant un mínim contracte.

- La separació de la lògica d'empresa i la de presentació facilita en gran part la documentació del codi.

- La lògica de negoci no haurà de dependre mai de components de comunicació a l'usuari o a tercers actors. Per exemple, no tindrà mai necessitats de coneixement de protocol HTTP, o de qualsevol altre, permetent la seva execució des de qualsevol entorn que implementi el model de Vista i Controlador.

A l'aplicació X-Help es faran servir pàgines JSP (incloses dins el model J2EE, Java 2 Enterprise Edition) per implementar el component de Vista.

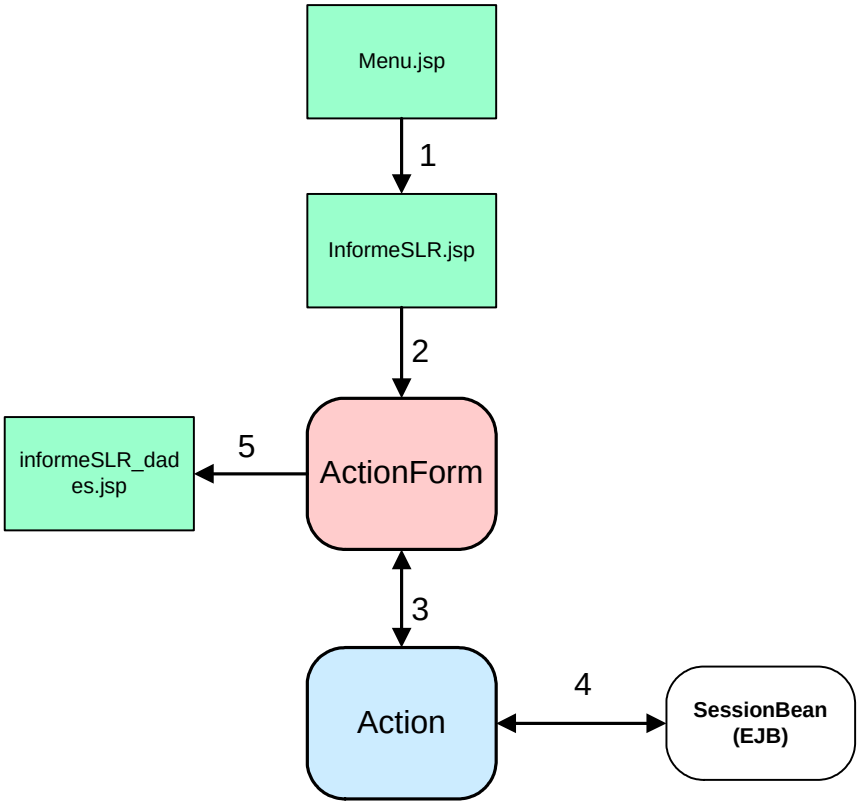
El component controlador serà implementat fent servir una combinació de servlet i JSP, i probablement algun EJB (Enterprise Java Bean) més alguna classe de recolzament.

El component del model serà implementat fent servir EJB. D'aquesta manera tota la lògica de negoci s'executarà dins el contenidor d'aplicacions. Aquest model es recolzarà en una capa de clases de tipus TAD (Tipus Abstracte de Dades).

Apache Struts.

Quan a les capes de vista i controlador, s'ha escollit una solució facilitada per la institució Apache (encarregada de estabilir estàndards en el mon Java) anomenada Apache Struts.

Aquesta implementació permet establir un mètode uniforme per desenvolupar la comunicació entre la capa de vista i la controladora. Veiem un petit gràfic sobre el funcionament d'Struts amb un exemple de l'aplicació XHelp:



Il·lustració 24. Exemple d'arquitectura basat en Apache Struts.

J2EE i el patró MVC

A continuació present una llista amb els productes escollits per desenvolupar el sistema.

La primera elecció ha estat J2EE com a llenguatge i plataforma de programació. Java es un llenguatge obert, amb multitud de plataformes que el suporten en múltiples entorns (Unix, Linux, Windows, etc.) i que assegura portabilitat i robustesa de l'aplicació.

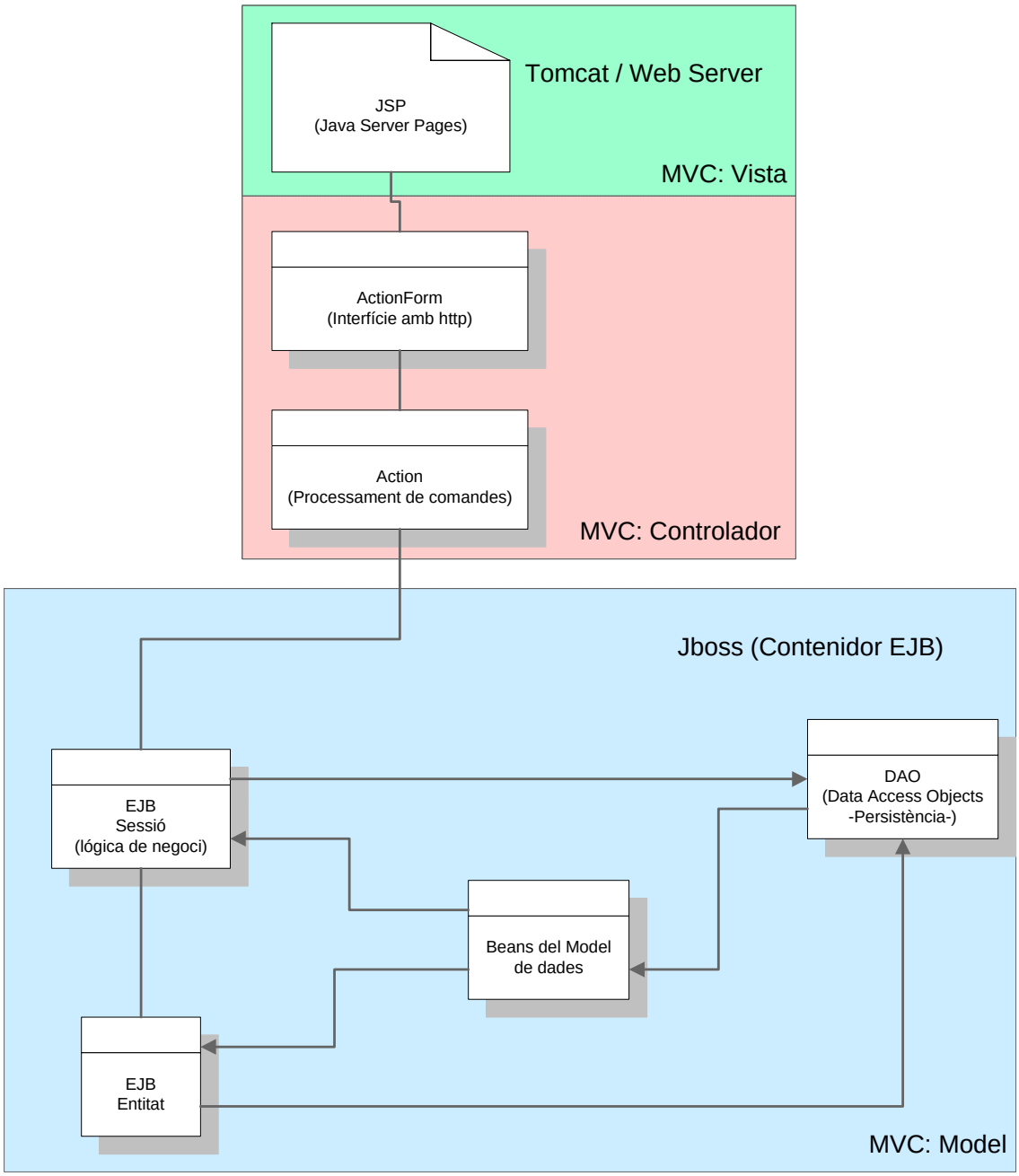
Per utilitzar les característiques més importants de J2EE necessitem un bon contenidor, o servidor d'aplicacions J2EE. En el mercat es troben molts i entre els servidors d'aplicacions de lliure distribució trobem que un dels més populars es JBOSS.

Per a la interfície d'usuari la elecció es un entorn web, doncs permet deslligar completament les màquines clients de les necessitats de l'aplicació i només necessitarà d'un browser per poder-la fer servir. Aquesta interfície estarà desenvolupada amb HTML i Javascript del costat client, i JSP del costat servidor.

El punt d'entrada dels navegadors clients al sistema serà un servidor web amb capacitat de gestionar servlets de Java. Trobem que Tomcat es un bon servidor, i en farem servir el servidor inclòs amb JBOSS.

Les dades l'aplicació s'emmagatzemaran en una base de dades mySQL. MySQL es una base de dades Opensource que te prou reconeixement com per tractar-la de manera seriosa en entorns professionals.

A continuació es presenta un diagrama aclaridor dels productes utilitzats, així com dels fluxes d'informació entre els diferents entorns. Aquest diagrama exemplifica també el patró Model-Vista-Controlador seleccionat per al projecte.



Il·lustració 25. Arquitectura general de Xhelp OutsourcingService

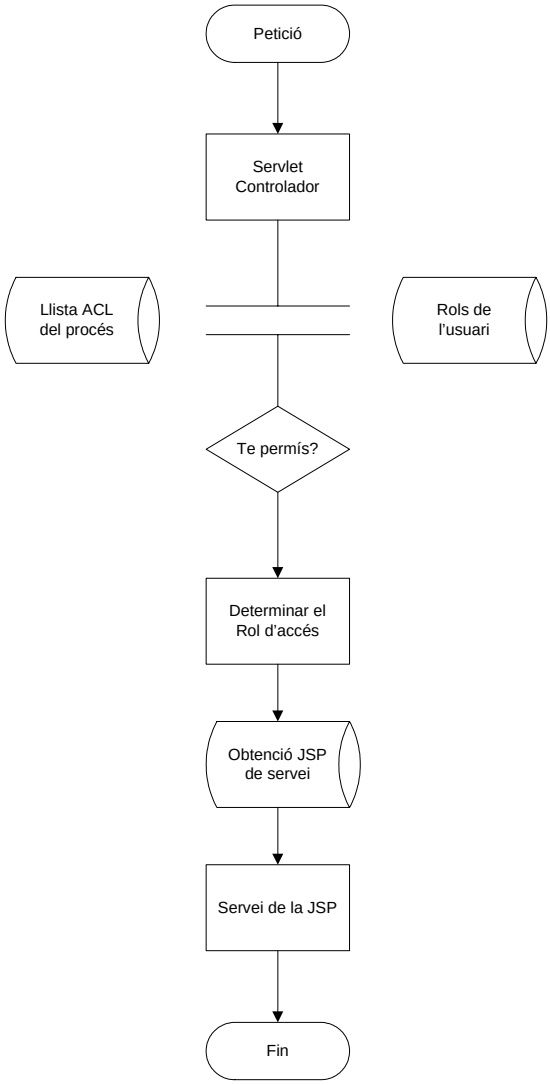
Disseny del control d'accés i d'usuaris

El sistema de control d'accessos i usuaris a les diferents aplicacions es farà mitjançant un patró basat en una llista de control d'accés, o ACL.

Cada procés del sistema que requereixi un cert nivell de seguretat tindrà una llista de rols que hi poden accedir. Si algun procés es obert a qualsevol usuari no tindrà llista de rols. Les llistes ACL es proporcionen ja configurades i sense utilitats de manteniment.

Cada usuari te assignats una llista de rols. Cada cop que un usuari vol accedir a un procés determinat, el sistema de seguretat comprovarà que te algun dels rols restringits i aplicarà aquell de major d'accés.

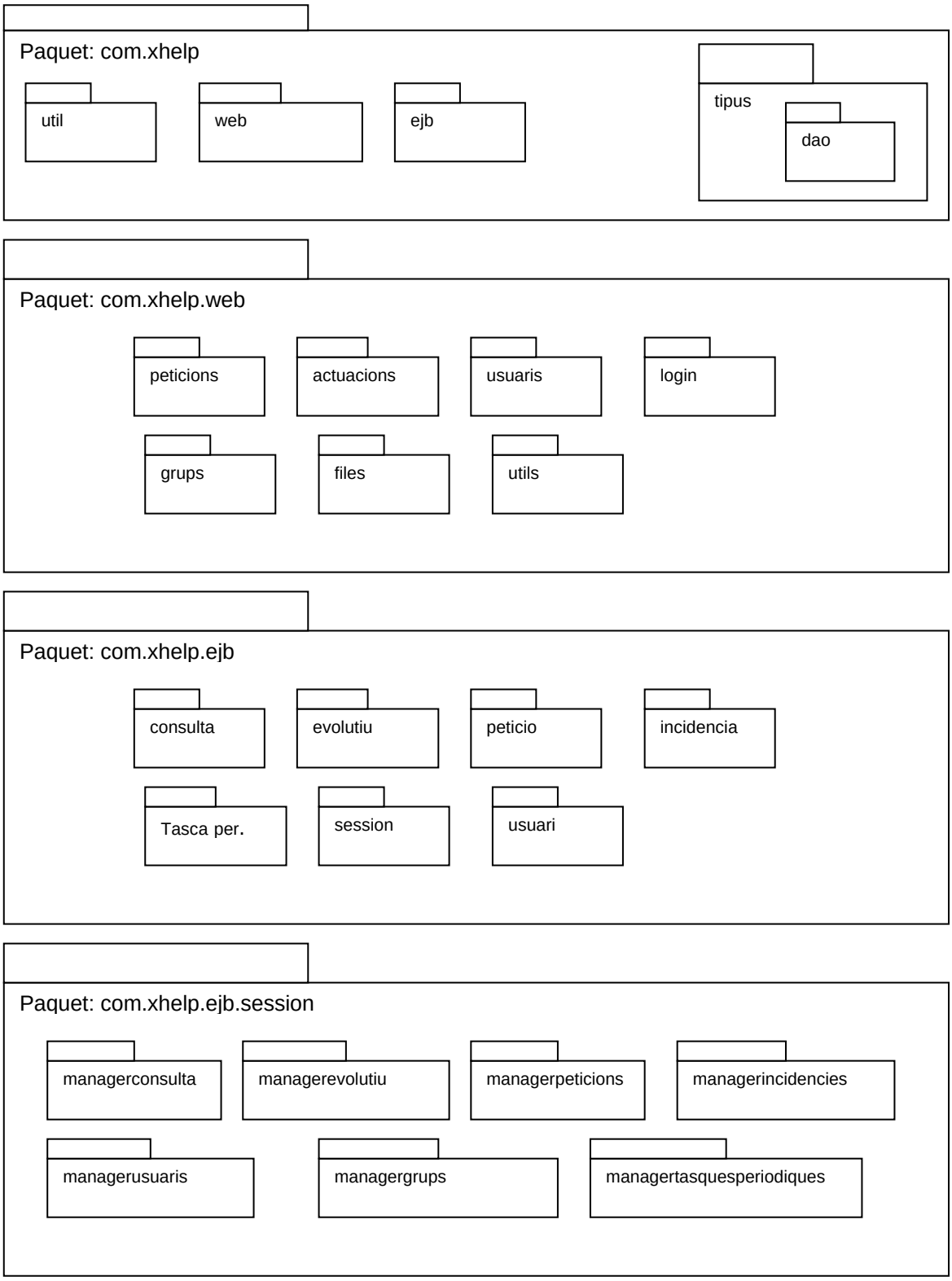
A la configuració de cada procés s'especifica la interfície .JSP que es farà servir per al cas d'us en funció del rol d'usuari aplicable.



Il·lustració 26. Disseny del control d'accés.

Distribució del programari en paquets.

Per fer el programari més portable i fàcil de mantenir es distribuirà el programari en paquets. Aquests paquets es faran en funció del paper, dins del sistema, de les classes que contindran.

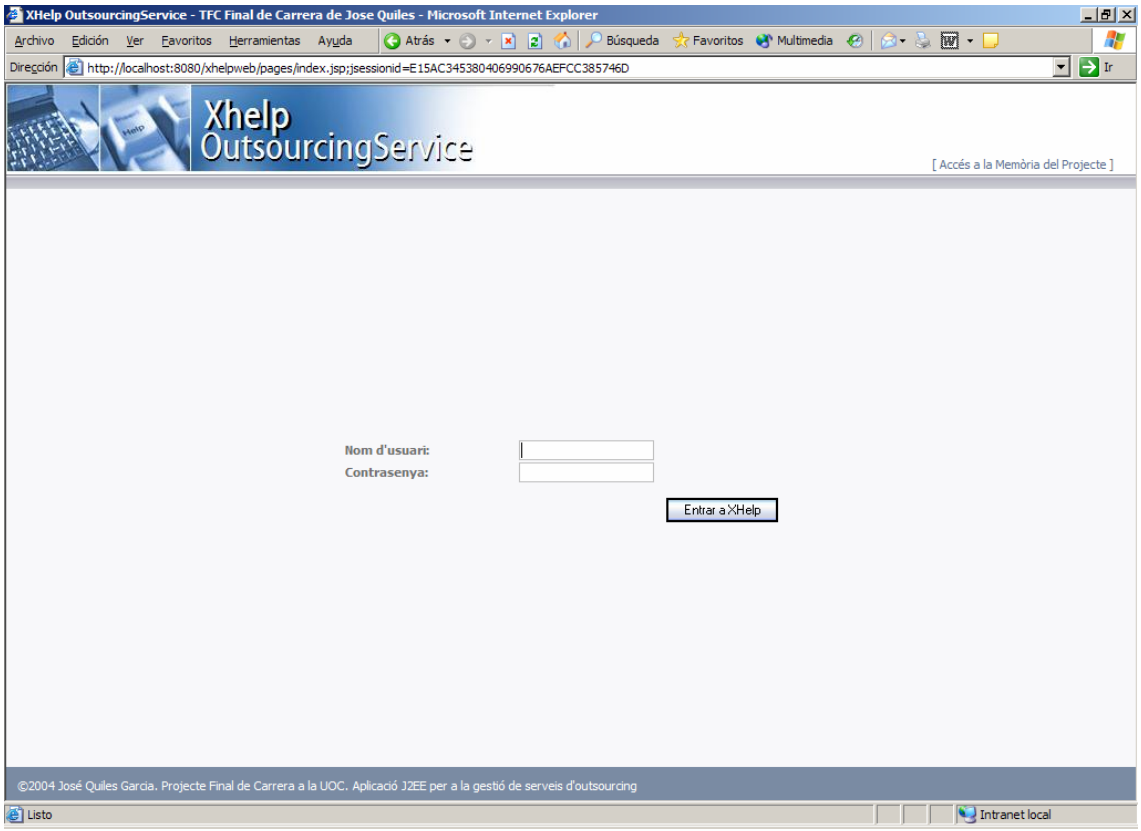


Com es pot veure tenim tres paquets principals: ***com.xhelp.web***, que contindrà altres subpaquets amb classes per la comunicació amb l'usuari, ***com.xhelp.ejb***, que contindrà la

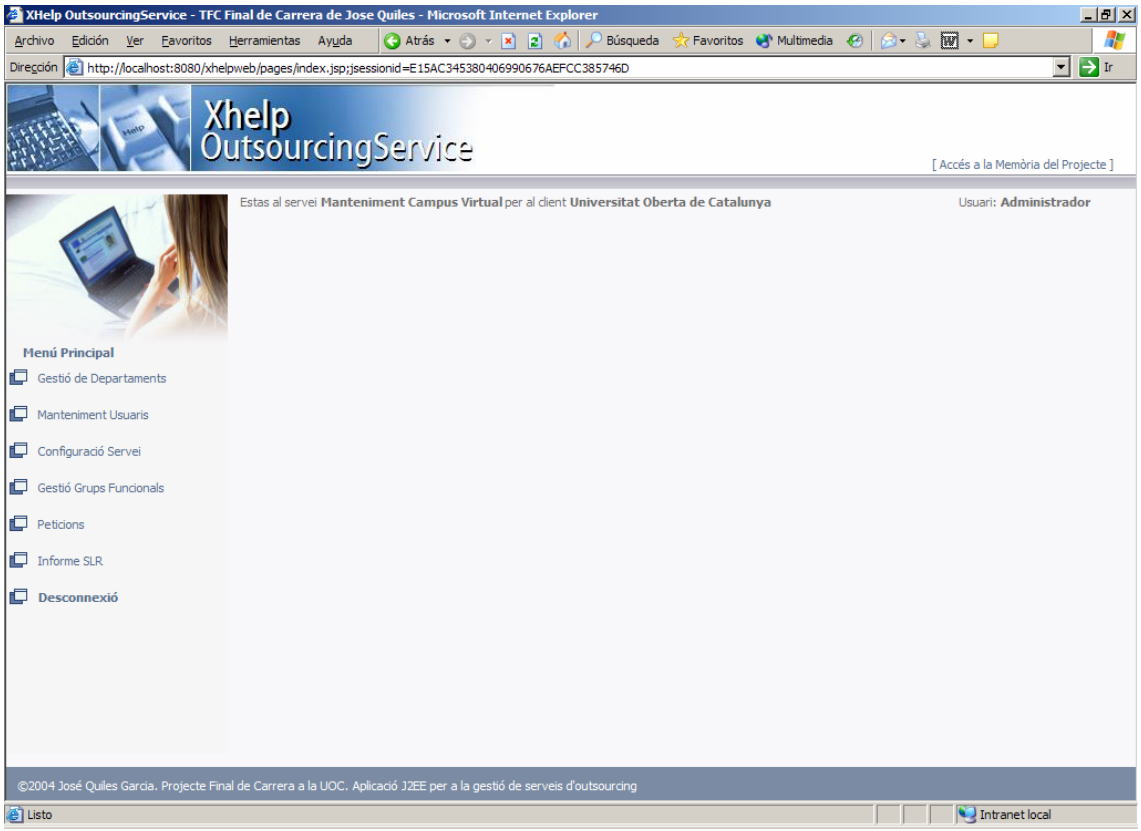
lògica de negoci, dins del paquet ***com.xhelp.ejb.session*** i els ejb d'entitat, i ***com.xhelp.tipus***, que contindrà els models de l'aplicació i la seva implementació d'accés a base de dades.

Interfície d'usuari.

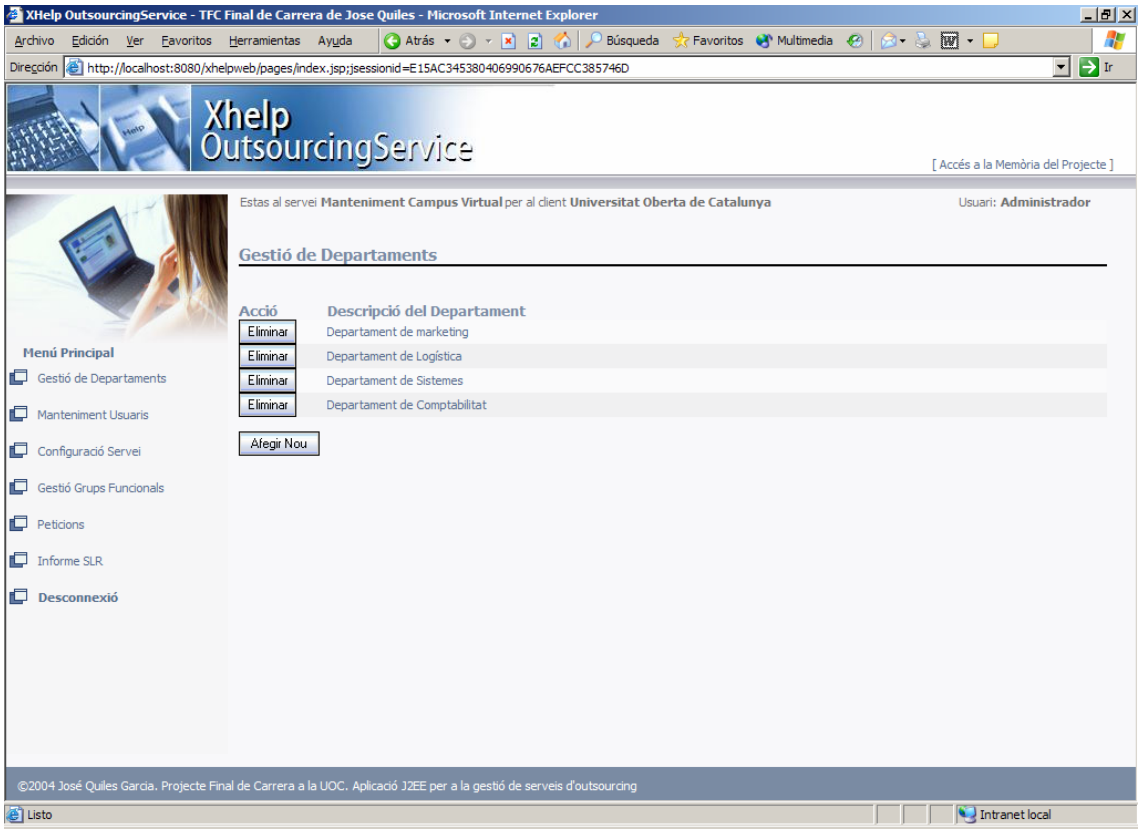
La interfície d'usuari s'ha dissenyat amb planes HTML. Veiem un petit circuit de navegació per l'aplicació:



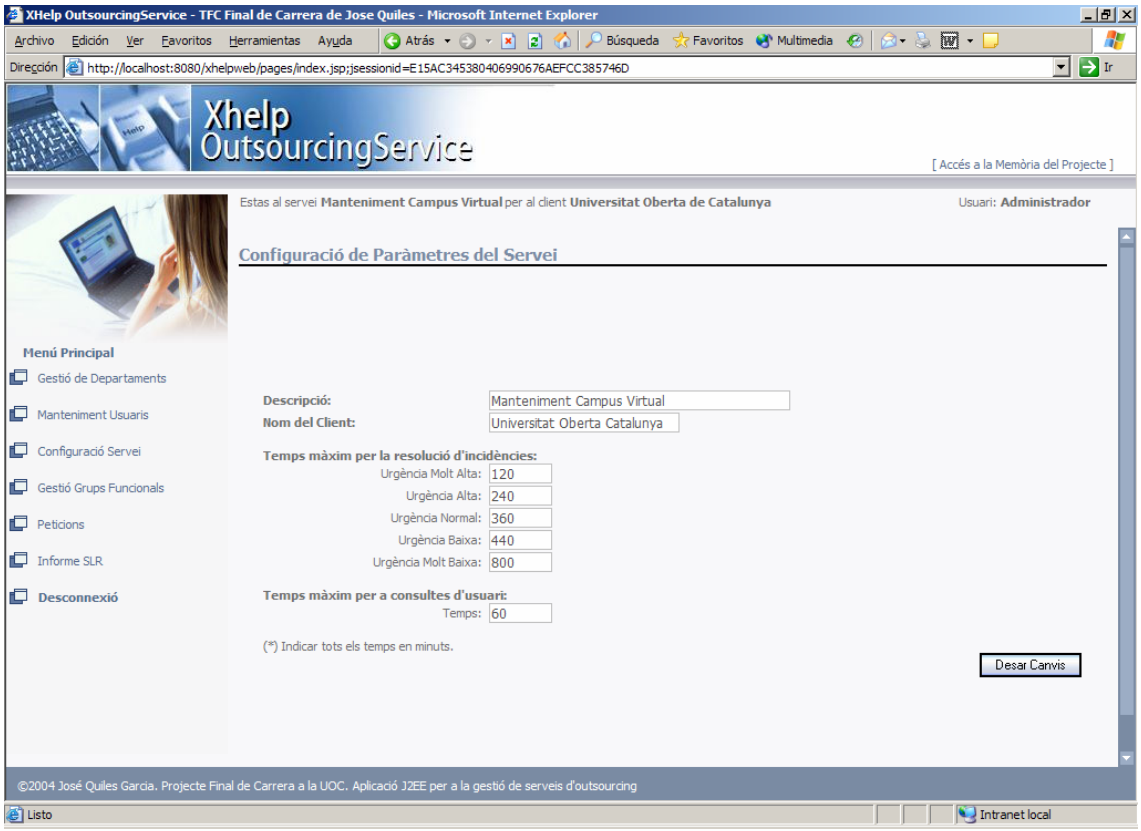
Il·lustració 27. Plana d'accés a Xhelp OutsourcingService



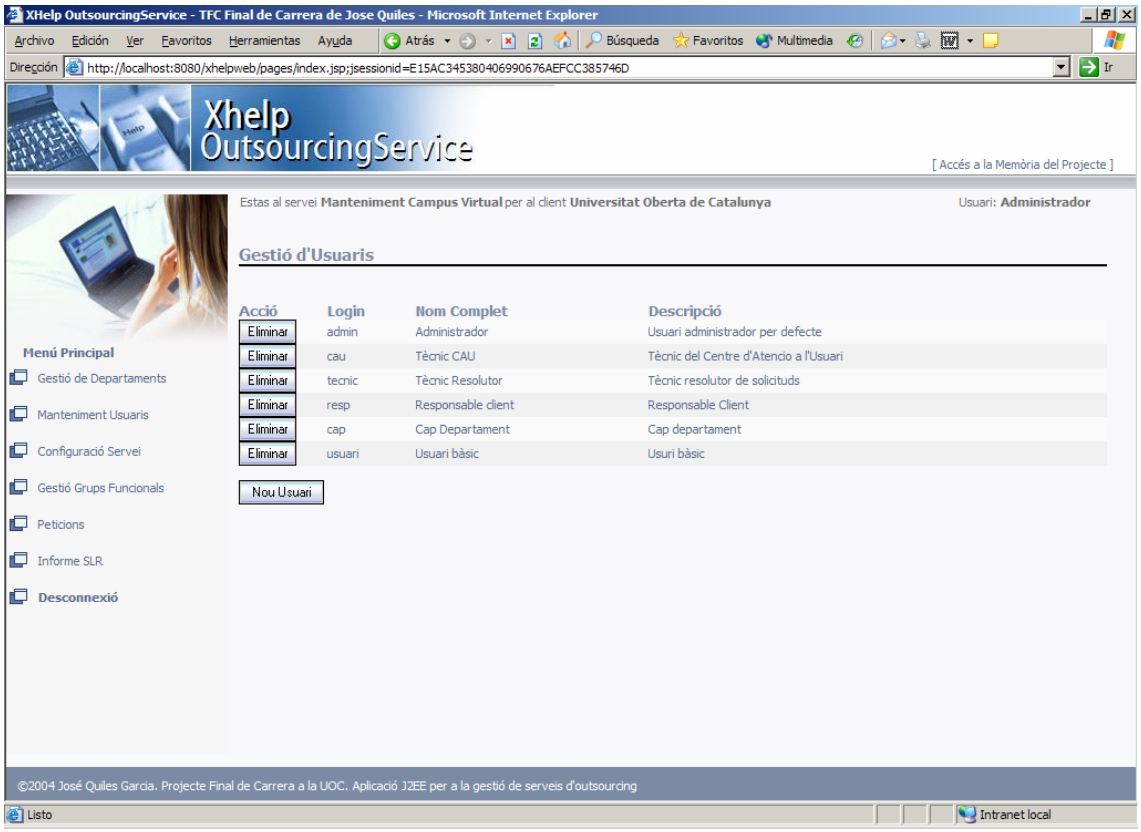
Il·lustració 28. Menú principal de l'Administrador



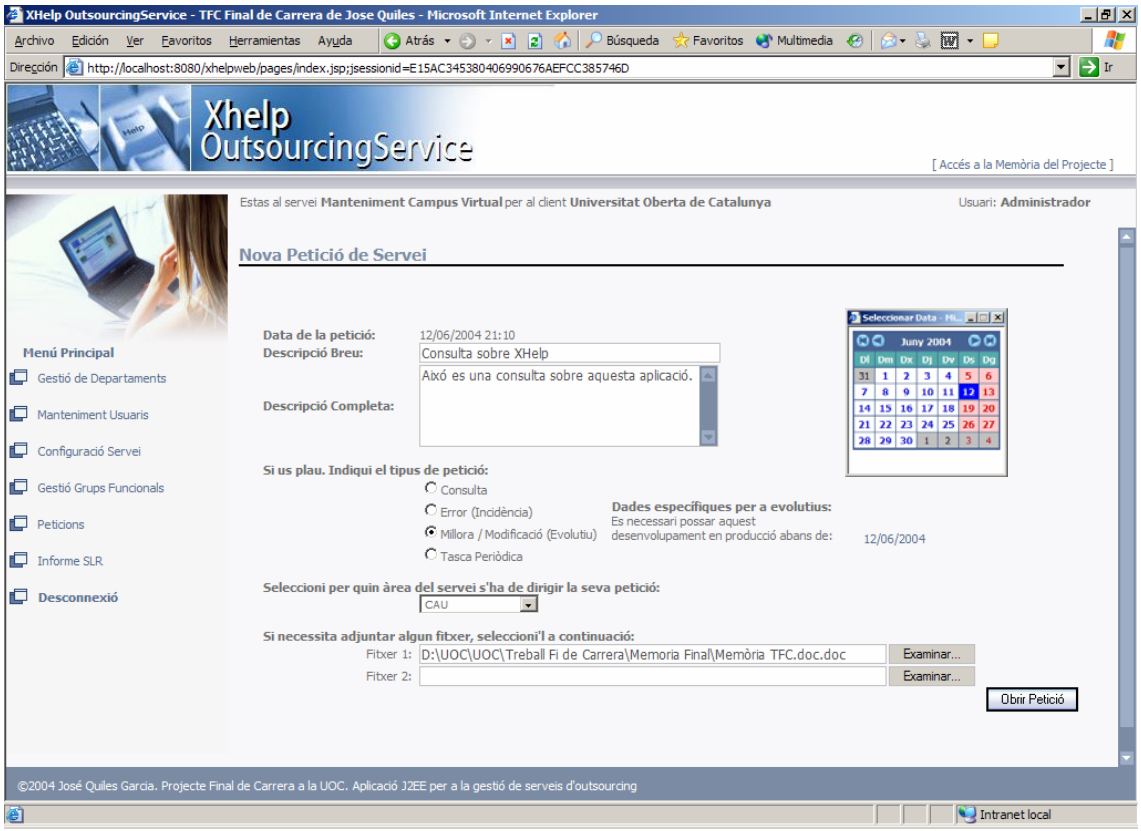
Il·lustració 29. Manteniment de departaments del client / grups funcionals



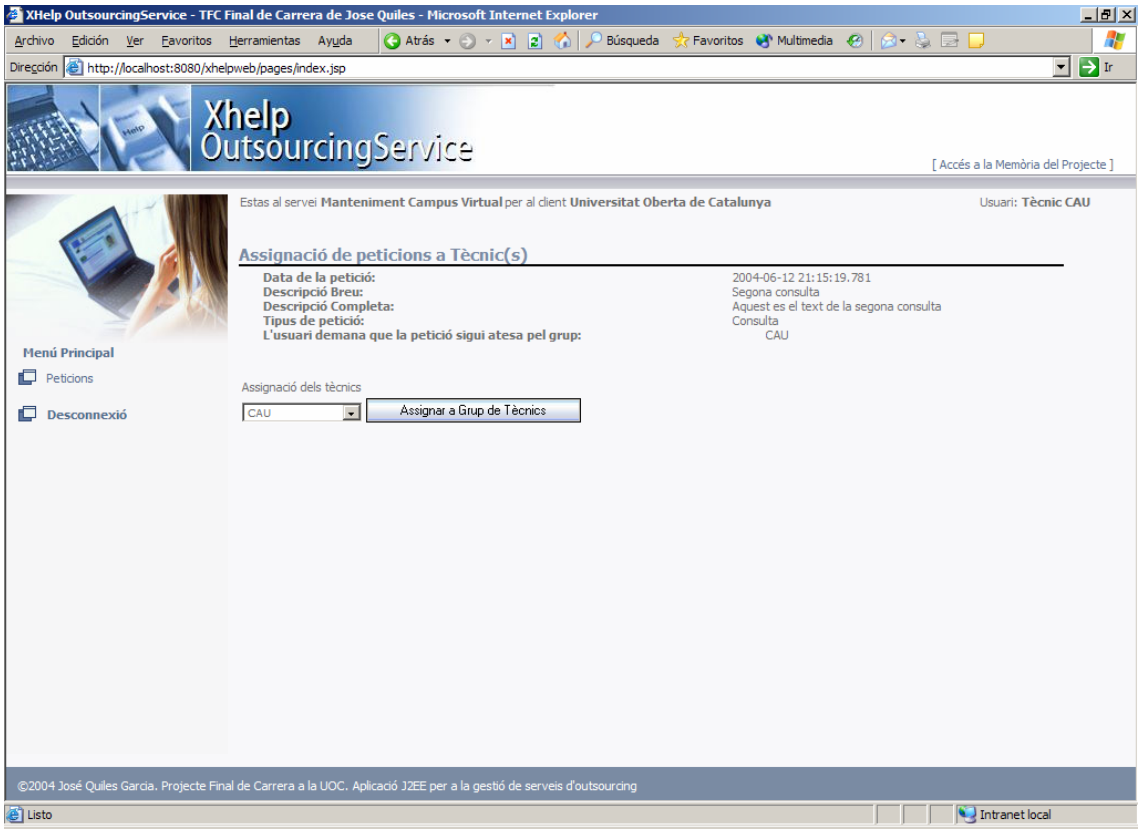
Il·lustració 30. Configuració dels nivells de servei



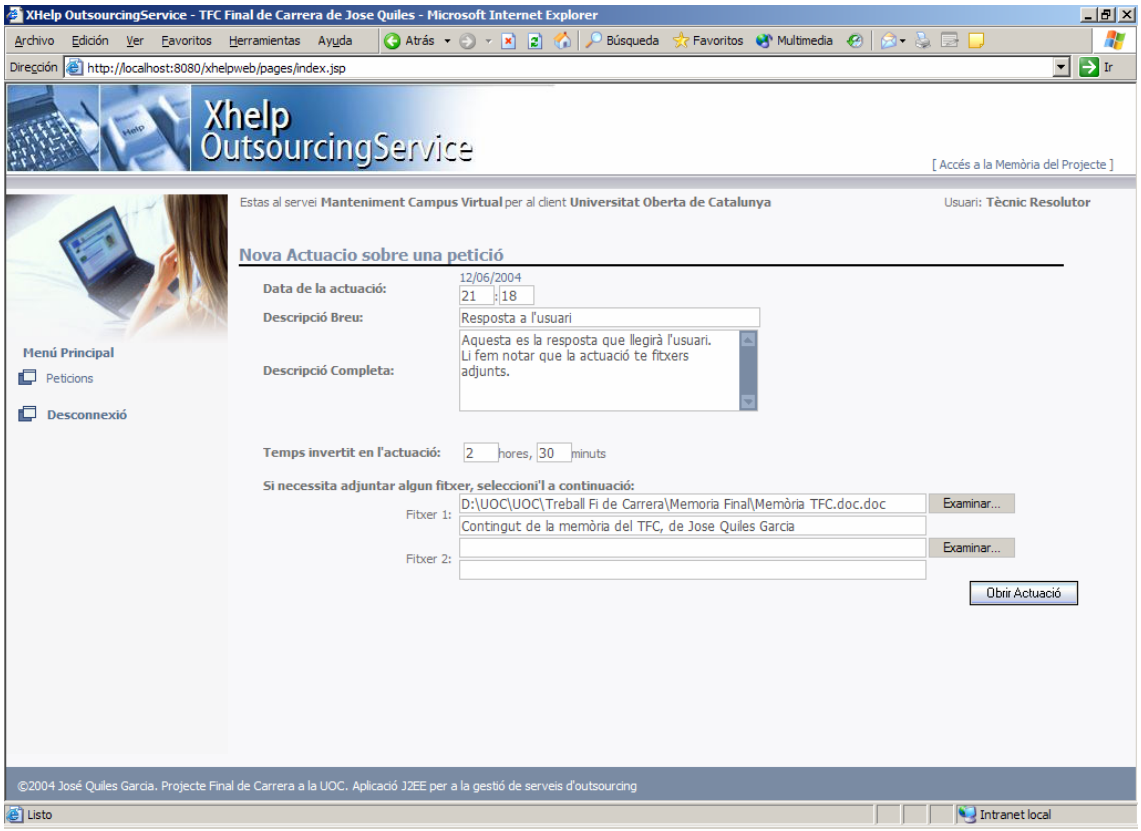
Il·lustració 31. Manteniment d'usuaris



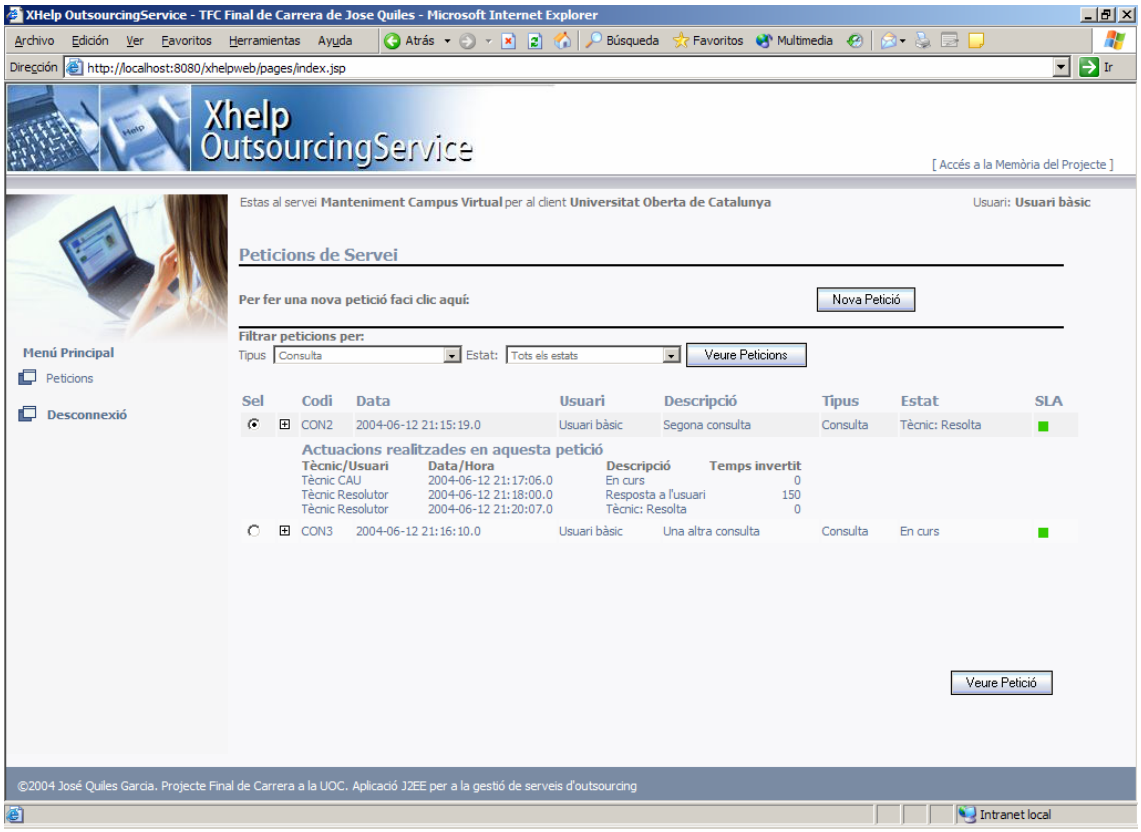
Il·lustració 32. Creació d'una petició al sistema



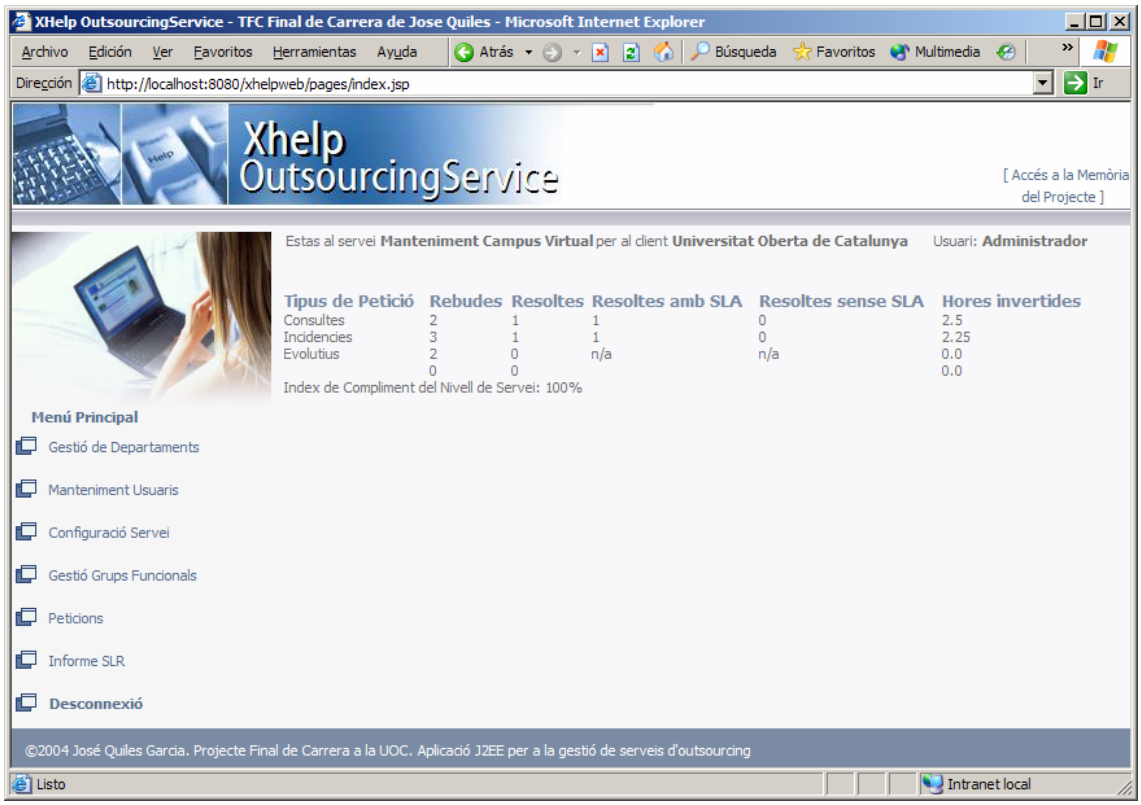
Il·lustració 33. Assignació a un grup de tècnics



Il·lustració 34. Actuació sobre una petició



Il·lustració 35. Consulta de peticions, amb l'historial d'actuacions.



Il·lustració 36. Informe SLR

Conclusions

La meva conclusió sobre aquest projecte es molt positiva. He après el funcionament de la tecnologia J2EE, concretament de les pàgines JSP, dels EJB de sessió (sense estat) i dels EJB d'entitat. Penso que probablement es una tecnologia que ha de madurar més, però que es situa en el camí més adient per fer-lo. En el mercat de les eines de desenvolupament, trobo que Java ha aconseguit un lloc de preferència gràcies la professionalitat i solidesa de la seva arquitectura.

Des del punt de vista del projecte, com a projecte de construcció de programari, he après que portar la planificació dia a dia ajuda a controlar i gestionar millor el projecte. Es una tasca difícil, però necessària, per que ajuda a detectar les desviacions a temps, i a corregir la marxa del projecte si es necessari. Fins i tot, ajuda a predir amb antelació els elements del projecte que no es podran dur a terme.

Ha estat una tasca enriquidora, per veure totes les passes necessàries per construir una aplicació de programari: des de les "suposades" entrevistes amb un client fins la presentació del programari, la documentació de la feina feta (com aquesta memòria), i la posta en marxa de l'aplicació.