



TFC
Treball Final de Carrera

Projecte J2EE
Agenda d'Actes i Gestió fotogràfica
Web
- Memòria del Projecte -

Moisès Fernández Blanco
ETIG

Josep Joan Rodríguez

18 de Juny del 2007

Resum

L'objectiu principal d'aquest projecte ha estat aprofundir en la construcció de programari, abordant totes les etapes d'un projecte de construcció de programari des de la perspectiva de l'enginyeria del software (anàlisi, disseny, implementació i proves) i utilitzant el paradigma de programació Orientada a l'Objecte mitjançant l'ús de la tecnologia J2EE, conjuntament amb bastions de programari de gran importància en el món real i per tant, en l'àmbit de desenvolupament de programari i tecnològic actuals.

El projecte escollit ha estat el desenvolupament d'un prototip d'aplicació per la gestió d'actes i esdeveniments conjuntament amb la gestió de fotografies associades d'una entitat, departament, o empresa, exemplificat concretament en l'àmbit hipotètic de la UOC.

El present document, constitueix una síntesi i recull de l'evolució del projecte, s'inicia amb una introducció en la que es citen aspectes com la justificació, els objectius, i la planificació, prosseguint amb l'anàlisi i disseny, passant per l'arquitectura de l'aplicació, una síntesi de les proves realitzades i finalitzant amb la conclusió.

Cal destacar, l'ús de tecnologies punteres i innovadores, i més concretament l'ús de Struts per la implementació del patró MVC, l'ús de Hibernate per la capa de persistència de l'aplicació, Spring per la injecció de dependències i serveis, seguretat basada en filtres mitjançant Acegi, AOP-AspectJ per la gestió de transaccions i el patró valuelist per la creació de llistats.

D'aquesta forma s'ha obtingut un producte final robust, estable i altament escalable.

Índex de continguts

RESUM	2
ÍNDEX DE CONTINGUTS	3
ÍNDEX DE FIGURES	4
1 INTRODUCCIÓ	6
1.1 JUSTIFICACIÓ DEL PROJECTE. PER QUÈ J2EE ?	6
1.2 OBJECTIUS DEL PROJECTE	6
1.2.1 BREU DESCRIPCIÓ FUNCIONAL DEL PROJECTE	6
1.2.2 TECNOLOGIES UTILITZADES	8
1.2.3 OBJECTIUS GENERALS	8
1.2.4 OBJECTIUS ESPECÍFICS	8
1.3 ENFOCAMENT I MÈTODE SEGUIT	9
1.4 PLANIFICACIÓ I TEMPORITZACIÓ	10
1.5 PRODUCTES OBTINGUTS	11
1.6 CONTINGUT DELS SEGÜENTS CAPÍTOLS DE LA MEMÒRIA	12
2 ESPECIFICACIÓ I DISSENY	12
2.1 ACTORS	12
2.2 CASOS D'ÚS	13
2.2.1 DIAGRAMA DE CASOS D'ÚS PER A LA GESTIÓ D'ACTES	13
2.2.2 DOCUMENTACIÓ TEXTUAL	14
2.2.3 DIAGRAMA DE CASOS D'ÚS PER A LA GESTIÓ DE FOTOGRAFIES	19
2.2.4 DOCUMENTACIÓ TEXTUAL	20
2.2.5 DIAGRAMA DE CASOS D'ÚS PER A LA GESTIÓ DE DADES BÀSIQUES	22
2.2.6 GUIÓ DEL MODEL DE PERFILS I SEGURETAT	23
2.2.7 CAS D'ÚS AUTENTICAR-SE	23
2.3 DIAGRAMA DE CLASSES	24
2.3.1 DIAGRAMA DEL MODEL – I	24
2.3.2 DIAGRAMA DEL MODEL – II	25
2.4 ESPECIFICACIÓ DE PANTALLES	26
MENÚ PRINCIPAL	26
CERCA D'ACTES	27
LLISTAT D'ACTES	28
AFEGIR NOU ACTE / CONSULTA DE L'ACTE	29
AFEGIR/ELIMINAR INTERVINENTS	31
MODIFICAR TASCA	32
AFEGIR / ELIMINAR ASSISTENTS	32
MODIFICAR CÀRREC	33
GESTIÓ DE FOTOGRAFIES	34
MENÚ PER AL MANTENIMENT DE TAULES AUXILIARS	35

MANTENIMENT DE COMARQUES, MUNICIPIS I LLOCS	36
AFEGIR/MODIFICAR COMARCA, MUNICIPI O LLOC	37
LLISTAT DE COMARQUES, MUNICIPIS O LLOCS	38
2.5 DISSENY DE LA BASE DE DADES	39
3 ARQUITECTURA	41
3.1 MVC (MODEL VISTA CONTROLADOR)	42
3.2 CAPA DE PERSISTÈNCIA. HIBERNATE	43
3.3 INVERSIÓ DEL CONTROL. SPRING.	44
3.4 EL PATRÓ VALUE LIST HANDLER	44
3.5 SEGURETAT BASSADA EN FILTRES MITJANÇANT ACEGI	45
3.6 ALTRES ASPECTES DE L'ARQUITECTURA	47
4 IMPLEMENTACIÓ	48
4.1 PROGRAMARI I MAQUINARI	48
4.2 DISTRIBUCIÓ I ESTRUCTURA DE PAQUETS DE L'APLICACIÓ	49
4.3 INTEGRACIÓ DE FRAMEWORKS	51
4.3.1 STRUTS	52
4.3.2 SPRING	53
4.3.3 PERSISTÈNCIA. HIBERNATE.	54
5 PROVES FUNCIONALS	56
6 CONCLUSIONS	57
6 GLOSSARI	58
7 BIBLIOGRAFIA	59

Índex de figures

Planificació, calendari	10
Planificació, diagrama de Gantt	11
Diagrama de casos d'ús per a la gestió d'actes	13
Diagrama de casos d'ús per a la gestió de fotografies	19
Diagrama de casos d'ús per a la gestió de dades bàsiques	22
Diagrama de classes I	24
Diagrama de classes II	25
Pantalla del menú principal	26
Pantalla per la cerca d'actes	27
Pantalla per al llistat d'actes	28
Pantalla per afegir nou acte/consulta de l'acte, captura I	29
Pantalla per afegir nou acte/consulta de l'acte, captura II	30
Pantalla per afegir/eliminar intervinents	31
Pantalla per modificar la tasca	32
Pantalla per afegir/eliminar assistents	32

Pantalla per modificar el càrrec	33
Pantalla per la gestió de fotografies	34
Pantalla de menú per al manteniment de taules auxiliars	35
Pantalla per al manteniment de comarques, municipis i llocs	36
Pantalla per afegir/modificar comarques, municipis o llocs	37
Pantalla del llistat de comarques, municipis o llocs	38
Disseny de la base de dades I	39
Disseny de la base de dades II	40
Diagrama de l'arquitectura	41
Diagrama del model MVC	42
Diagrama del model de persistència	43
Diagrama de classes del patró Value List Handler	45
Acegi, cicle de vida en l'encadenament de filtres	46
Distribució i estructura de paquets de l'aplicació	49
Taula resum de proves funcionals	56

1 Introducció

1.1 Justificació del projecte. Per què J2EE ?

J2EE (Java to Enterprise Edition, avui en la versió 6.0), és una plataforma de programació, per al desenvolupament i execució d'aplicacions distribuïdes amb arquitectura multicapa, bassades en components de software altament modulars executant-se sobre un servidor d'aplicacions, aquesta plataforma està definida per una especificació, i està considerada informalment gairebé com un estàndard.

Java EE inclou varies especificacions API, tals com JDBC, RMI, JMail, JMS, Web Services ... i defineix com utilitzar-les. Java EE inclou també algunes especificacions úniques per a components, aquestes inclouen Enterprise JavaBeans, servlets, portlets, JavaServer Pages i varies tecnologies de serveis web.

Avui en dia és una plataforma altament reconeguda en el món professional i per tant, altament demandada. La seva qualitat, solidesa i antiguitat en el sector empresarial la fan una molt bona elecció de cara al desenvolupament de programari distribuït.

Per altra banda, cal dir que això ha fomentat l'aparició de tota una comunitat de programari obert, així com, la creació al seu voltant de tota una sèrie de patrons o bastions de programari (Struts, Hibernate, Tapestry, JSF, etc) que en alguns casos han adquirit tant de protagonisme que s'han consolidat quasi com veritables estàndards.

1.2 Objectius del projecte

1.2.1 Breu descripció funcional del Projecte

La UOC i concretament els seus organismes encarregats del protocol i realització d'actes, així com, la seva Unitat fotogràfica en el seu interès per millorar la gestió dels actes protocol·laris o no protocol·laris, que són realitzats setmanal o mensualment, i als quals assisteix la Rectora, o els representants oficials de la UOC, assessorats per la Unitat de Informàtica han decidit adquirir una solució a mida que els hi permeti dur a terme la gestió dels actes abans mencionats, així com, la gestió de la documentació associada (expedients), i la gestió fotogràfica d'una forma eficient i eficaç, amb aquest motiu han sol·licitat els nostres serveis.

El projecte *eAgenda*, persegueix els objectius abans mencionats, així doncs amb aquest programa es podran donar d'alta actes, entenent-se per un acte, un event (seminari, certamen, presentació, etc) ja sigui de caràcter protocol·lari (oficial) o no, en el que participen o al qual assisteixen els membres o màxims representants de la Universitat, aquests actes posteriorment mitjançant l'agenda d'actes que incorporarà el nostre projecte podran ésser consultats,

tanmateix s'ha de permetre la baixa lògica d'aquests en el cas de que així es desitgi.

La Unitat encarregada de la celebració dels actes, amb el recull de tota la informació d'un acte genera un expedient, així doncs, l'aplicació ha de permetre gestionar la documentació associada. Aquesta documentació podrà ésser emmagatzemada en carpetes, així el nostre programa ens permetrà crear carpetes per posteriorment annexar-ne la documentació pertinent, en qualsevol moment es pot voler eliminar tant la documentació associada com les seves carpetes, respecte aquestes últimes serà requisit per dur a terme l'acció esmentada que no tinguin cap document associat.

Per altra banda, aquests actes acostumen a ésser coberts per el personal de la unitat fotogràfica, els quals s'encarreguen de treure fotografies dels actes, així doncs, el programa permetrà gestionar les fotografies vinculades als actes, les quals podran ésser donades d'alta, consultades, modificades i eliminades, a aquestes gestions s'haurà d'associar també la gestió dels personatges fotografiats. Posteriorment els assistents als actes, així com, els organismes de la UOC encarregats del protocol i de la realització dels actes podran sol·licitar o cursar peticions de les fotos triades, les quals seran enviades a un laboratori fotogràfic, per la posterior recollida per la persona o entitat demandant, amb aquest motiu és mantindran les dades dels destinataris, podent ésser gestionades des del programa, i entenent-se per gestió la consulta, alta, modificació o eliminació de les dades del destinatari.

L'aplicació contemplarà també la gestió de les dades de taules auxiliars, com poden ésser la taula de països, la de comarques, la de localitats, la de càrrecs, etc, aquestes seran esmentades amb més detall en la fase d'anàlisi i disseny del projecte.

Evidentment, no tots els usuaris tindran els mateixos drets, és per aquest motiu que existiran diversos perfils d'usuari amb diversos nivells d'autorització, l'aplicació, per tant, disposarà de perfils d'accés que permetran filtrar l'accés a fotografies d'actes de la Rectora, o als actes d'altres membres de la UOC, o a altres divisions que es puguin afegir..

Per altra banda, aquests perfils permetran als usuaris de la unitat de fotografia, així com, a aquells d'altres departaments que estiguin autoritzats, la modificació de les fotografies i la seva consulta. En les fases d'anàlisi i disseny s'especificaran aquests amb més detall.

1.2.2 Tecnologies utilitzades

Com a tecnologia per al desenvolupament de l'aplicació s'ha utilitzat J2EE conjuntament amb bastions de programari (*frameworks*), així doncs, per la capa de persistència s'ha utilitzat Hibernate 3.2.2, per al disseny MVC (Model Vista Controlador) s'ha utilitzat Struts 1.3 (controladors) conjuntament amb JSP i tecnologia JSTL (JavaServer Pages Standard Tag Library) per la capa de presentació. Els diversos serveis involucrats en l'aplicació han estat gestionats per Spring 2.0.2 utilitzant el patró de Inversió de Control (IoC), també anomenat de dependència i injecció.

Per altra banda, donada la seva qualitat, així com la seva economicitat s'ha utilitzat programari de codi obert (Open Source) , així doncs com a servidor d'aplicacions s'ha utilitzat Tomcat 5.5.23, conjuntament amb Oracle 10g com a base de dades relacional (aquesta és lliure en les versions de desenvolupament però cal tenir llicència si es pensa utilitzar en producció)

Totes les connexions a la BBDD seran gestionades mitjançant un pool de connexions.

Com a entorn de desenvolupament s'ha utilitzat Eclipse 3.1 amb l'ús de diversos plugins, com pot ésser el WTP (Web Tools Platform). Per altra banda, per al disseny de classes UML s'ha utilitzat Magic Draw versió 11.0.

La gestió de les transaccions realitzades contra la base de dades ha estat implementada mitjançant AOP (Programació Orientada a l'aspecte).

1.2.3 Objectius generals

Amb aquest treball és persegueix com a primer objectiu, i objectiu general, el desenvolupament d'una aplicació de gestió, així com, experimentar la problemàtica real que una aplicació d'aquest tipus comporta. Les aplicacions empresarials comporten, un esforç d'adaptació dels desenvolupadors, analistes i cap de projectes per conèixer la lògica de negoci, així com, l'esforç d'adaptació per adquirir el mateix punt de vista que l'usuari final o client que serà el que finalment farà servir l'aplicació. Tot això comporta una feina d'anàlisi i reflexió, que duta acuradament permetrà aconseguir l'objectiu primordial, on la usabilitat, la senzillesa i eficàcia han de predominar com a trets característics de l'aplicació.

Per altra banda, dir que les aplicacions web caracteritzades per un baix ús de requeriments per part de l'usuari final, són aplicacions molt atractives en l'àmbit de sistemes distribuïts, permetent a l'usuari final la seva utilització en qualsevol lloc i en qualsevol moment, tan sols amb una connexió de xarxa i un navegador.

1.2.4 Objectius específics

L'aplicació com ja s'ha explicat anteriorment, ha d'ésser una eina encaminada a gestionar actes, o sigui, una mena d'agenda actes amb la possibilitat de gestionar les fotografies d'aquests. Des del punt de vista de la seva utilització

es contemplen varis grups o perfils d'accés, bàsicament de consulta i de creació i modificació.

A banda, de l'objectiu de proporcionar una eina de treball per l'usuari final, amb el desenvolupament d'aquesta aplicació es persegueixen d'altres objectius específics, els quals s'anomenen a continuació:

- ✓ Adquirir experiència en el terreny de l'anàlisi i disseny de les aplicacions de gestió, conjuntament amb l'ús d'eines de disseny CASE.
- ✓ Adquirir experiència en l'àmbit de la usabilitat i de la interacció humana respecte el programari.
- ✓ Aprofundir en el desenvolupament de programari distribuït i les problemàtiques que aquest comporta, així com, en el model client – servidor, el que ens porta cap a l'aprofundiment en l'ús de servidors, ja siguin de contingut estàtic o servidors d'aplicacions.
- ✓ Aprofundir en l'ús de tecnologies J2EE, i concretament en l'ús de tecnologies punteres, algunes d'elles basades en frameworks com poden ésser Spring, Struts, o Hibernate.

1.3 Enfocament i mètode seguit

En quant a la metodologia seguida ha estat el cicle de vida clàssic o en cascada, on no s'ha passat a la següent etapa fins a haver acabat l'etapa que la precedia. Les diferents etapes que han intervingut en el projecte han estat: l'anàlisi, el disseny, la implementació i la de proves. Respecte a la fase de requisits previs, cal comentar que aquesta "no ha estat duta a terme" en aquest projecte, ja que es partia d'un cas real de l'àmbit laboral, i estava prou definida abans del inici d'aquest projecte, dit d'una altra forma es partia amb aquesta ja feta. Per últim, cal comentar també, que cadascuna de les etapes ha estat supervisada per el consultor de forma que no s'ha passat a la següent fins a haver modificat o corregit aquesta, ateses les modificacions que el consultor ha indicat que calia fer.

1.4 Planificació i temporització

A continuació en l'arxiu PDF incrustat, es presenta una planificació de les diverses tasques, bàsicament la planificació es pot dividir en tres grans grups: disseny i anàlisi, implementació de l'aplicació, fase de testing i elaboració de la memòria.

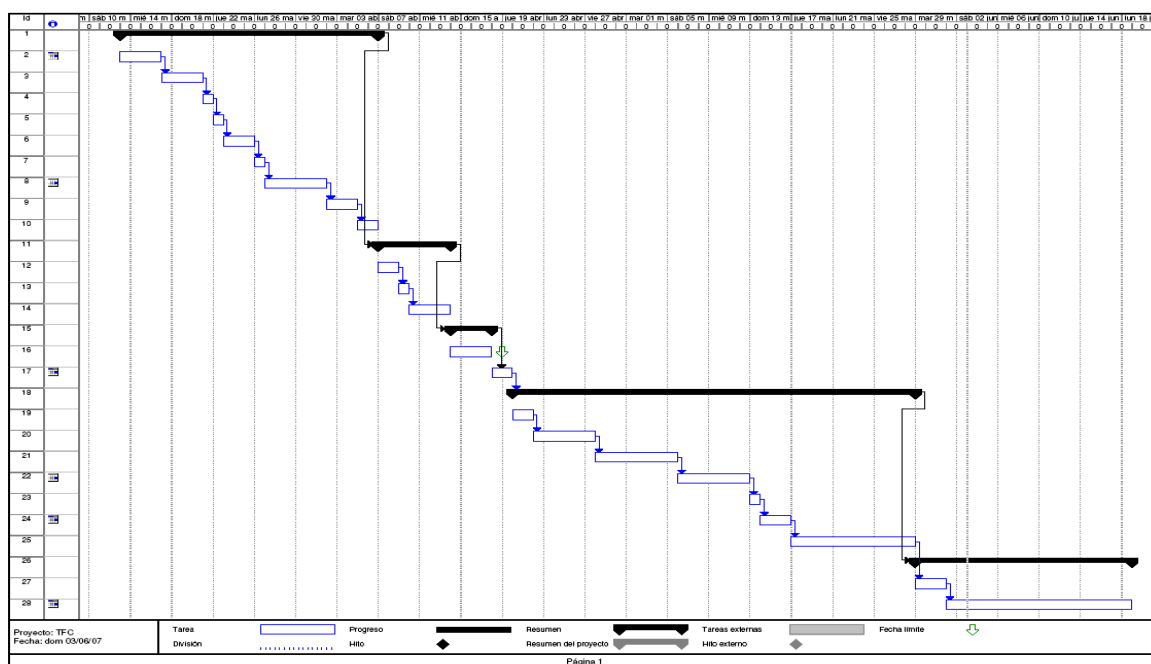
Nota: cal fer doble clic en el quadre annex en la pàgina següent per visualitzar el contingut del document PDF.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	Anàlisi i Disseny	25 dias	mar 13/03/07	vie 06/04/07		
2	Especificació dels casos d'ús per a la gestió d'actes	4 dias	mar 13/03/07	vie 16/03/07		
3	Especificació dels casos d'ús per a la gestió de fotografies	4 dias	sáb 17/03/07	mar 20/03/07	2	
4	Especificació dels casos d'ús per a la gestió de dades bàsiques	1 dia	mié 21/03/07	mié 21/03/07	3	
5	Guió del model de perfils i seguretat	1 dia	jue 22/03/07	jue 22/03/07	4	
6	Diagrama de classes del model I	3 dias	vie 23/03/07	dom 25/03/07	5	
7	Diagrama de classes del model II	1 dia	lun 26/03/07	lun 26/03/07	6	
8	Especificació de pantalles	6 dias	mar 27/03/07	dom 01/04/07	7	
9	Disseny de la base de dades	3 dias	lun 02/04/07	mié 04/04/07	8	
10	Especificació de l'arquitectura	2 dias	jue 05/04/07	vie 06/04/07	9	
11	Instal·lació i configuració	7 dias	sáb 07/04/07	vie 13/04/07	1	
12	Instal·lació i configuració d'Oracle 10g	2 dias	sáb 07/04/07	dom 08/04/07		
13	Instal·lació i configuració d'Apache-Tomcat	1 dia	lun 09/04/07	lun 09/04/07	12	
14	Creació de projecte i configuració frameworks	4 dias	mar 10/04/07	vie 13/04/07	13	
15	Formació	4 dias	sáb 14/04/07	mar 17/04/07	11	
16	Formació en Spring Framework	4 dias	sáb 14/04/07	mar 17/04/07		
17	Creació EBDD i inserció de dades per proves	2 dias	mié 19/04/07	jue 19/04/07	15	
18	Implementació	39 dias	vie 20/04/07	lun 28/05/07	17	
19	Implementació de classes del model	2 dias	vie 20/04/07	sáb 21/04/07		
20	Implementació Data Access Objects	6 dias	dom 22/04/07	vie 27/04/07	19	
21	Implementació dels Business Objects	8 dias	sáb 28/04/07	sáb 05/05/07	20	
22	Implementació dels controladors de l'aplicació	7 dias	dom 06/05/07	sáb 12/05/07	21	
23	Implementació del servei de traces	1 dia	dom 13/05/07	dom 13/05/07	22	
24	Implementació del servei de seguretat	3 dias	lun 14/05/07	mié 16/05/07	23	
25	Implementació de la interfície gràfica	12 dias	jue 17/05/07	lun 28/05/07	24	
26	Fase final	21 dias	mar 29/05/07	lun 18/06/07	18	
27	Proves funcionals	3 dias	mar 29/05/07	jue 31/05/07	25	
28	Memòria Final	18 dias	vie 01/06/07	lun 18/06/07	27	

Página 1

A continuació en la pàgina següent es presenta el diagrama de Gantt associat a les tasques definides:

Nota: cal fer doble clic en el quadre annex en la pàgina següent per visualitzar el contingut del document PDF.



1.5 Productes obtinguts

El producte final obtingut ha estat un prototipus el qual bàsicament permet realitzar la gestió d'actes o events d'una entitat o empresa, entenent-se per gestió, l'alta, baixa, modificació i consulta d'actes, també s'ha implementat un cercador per tal de realitzar la cerca d'actes, així com les funcionalitats per gestionar les fotografies associades a l'acte, podent-se annexar o afegir fotografies, esborrar-les o descarregar-les en forma de fitxer comprimit.

S'han implementat també les funcionalitats relatives a tres manteniments bàsics (comarques, municipis i llocs), la finalitat d'aquestes darreres implementacions ha estat practicar i continuar aprofundint en l'àmbit tecnològic adquirit en el desenvolupament de les funcionalitats esmentades en el paràgraf anterior.

Conjuntament s'ha lliurat una mini aplicació que atesa la mida considerable de la base de dades, té la finalitat de creació d'aquesta de forma pràcticament automàtica.

Cal considerar també, els diferents documents que han estat lliurats al llarg de les diferents etapes del projecte i que han estat: document amb els objectius i planificació, així com una breu especificació funcional del projecte, document relatiu a l'anàlisi i disseny del projecte, i el document de proves funcionals que serà lliurat conjuntament amb aquesta memòria.

1.6 Contingut dels següents capítols de la memòria

En els següent capítol de la memòria s'explica en detall l'etapa d'especificació i disseny, per continuar amb un capítol sencer en el qual es fa referència a l'arquitectura J2EE utilitzada en la realització del projecte, fent referència al model MVC i la tecnologia Struts, a la capa de persistència implementada mitjançant Hibernate, passant per la inversió de control mitjançant Spring, el patró value-list, i finalitzant el capítol d'arquitectura amb un apartat dedicat al model de seguretat basada en filtres i implementat amb Acegi.

A continuació en el següent capítol es fa una síntesi d'aquells aspectes més rellevants de l'etapa d'implementació. En el capítol cinquè es fa una síntesi del resultat de les proves funcionals dutes a terme sobre el prototip obtingut.

Per finalitzar amb un capítol dedicat a les conclusions finals i amb la bibliografia utilitzada.

2 Especificació i disseny

2.1 Actors

Com a principal divisió podem distingir en primera instància entre dos tipus d'actors, els usuaris pertanyents a la unitat de protocol i realització d'actes de la Universitat Oberta de Catalunya, i els pertanyents a la unitat de fotografia. Els primers gestionen tota la documentació associada amb els actes a que assisteix la rectora, així doncs, tota la documentació d'un acte és recollida en un expedient.

En quant als segons, la unitat de fotografia, són els encarregats de cobrir actes diversos, i són els encarregats de gestionar l'arxiu fotogràfic.

Un tercer tipus d'actors, esporàdicament altres departaments o unitats podran accedir a mode de consulta a determinats tipus d'actes i determinats tipus de fotografies.

Aquests tipus de classificació d'usuaris condicionarà l'existència de varis tipus de perfils d'accés a l'aplicació amb diferents privilegis d'accés.

Així doncs, els usuaris de la unitat de fotografia podran tenir accés a actes de caràcter protocol·lari i a esdeveniments de la unitat de fotografia, els usuaris de la unitat de protocol podran tenir accés a actes protocol·laris i a esdeveniments de la unitat de protocol, i la resta d'usuaris només podran accedir a actes de caràcter protocol·lari.

Posteriorment s'especificarà tota una secció dedicada a descriure tots els diferents nivells d'accés.

Per últim, es contempla l'existència d'un quart tipus d'usuari (superusuari) amb accés total a tota l'aplicació, és a dir, tant a la gestió d'actes de tota mena, així com, amb accés a la gestió d'usuaris i de dades o taules auxiliars.

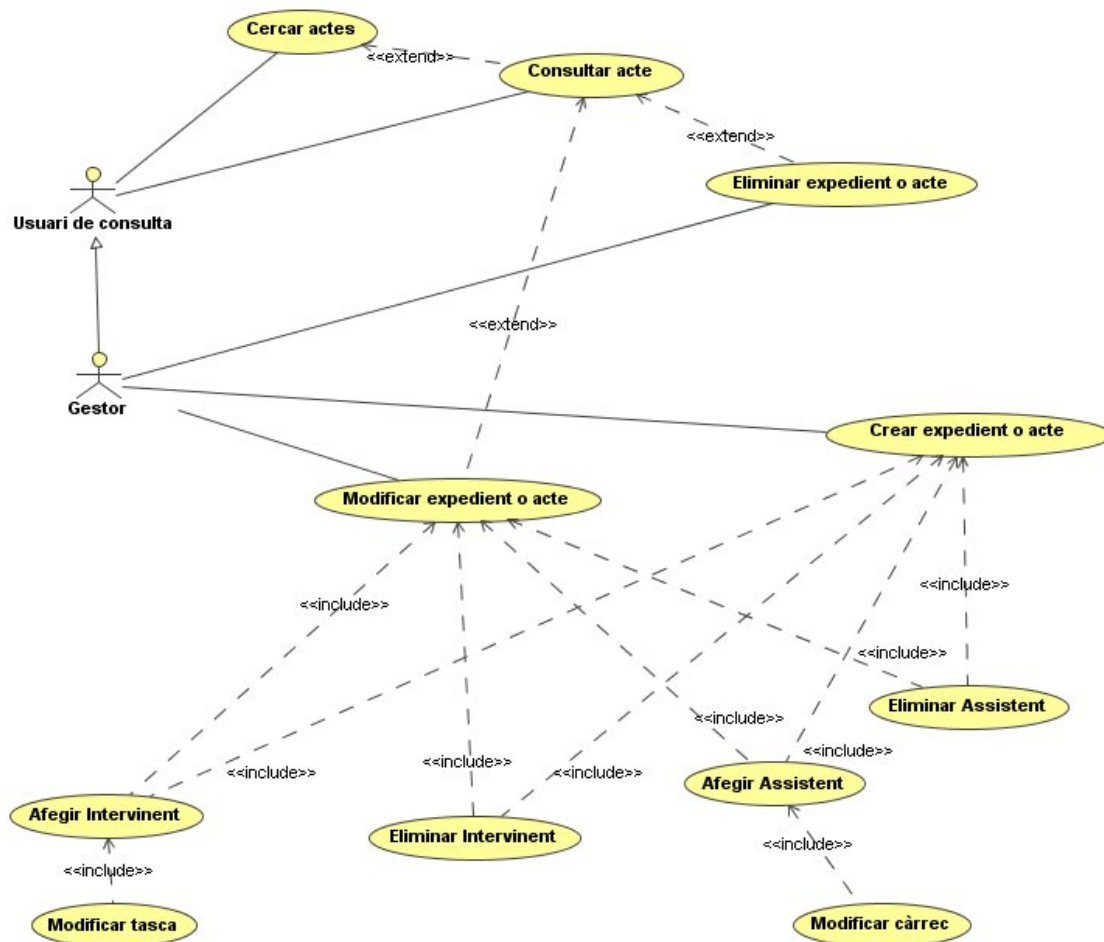
2.2 Casos d'ús

Per motius de simplificació s'ha realitzat una divisió en tres diagrames de casos d'ús, aquesta divisió s'ha fet atenent a la funcionalitat, així doncs, s'ha fet una separació en tres mòduls funcionals, atenent a la gestió d'actes, a la gestió de fotografies i a la gestió de dades bàsiques.

Per altra banda, per motius d'espai no s'han especificat en els diagrames de casos d'ús ni en la documentació textual tots els casos d'ús que s'especificaren en les etapes d'anàlisi i disseny, només s'han especificat aquells que s'han dut a terme en l'etapa de implementació, deixant de banda la resta per el motiu esmentat anteriorment i per qüestions de temps.

2.2.1 Diagrama de casos d'ús per a la gestió d'actes

S'ha realitzat una simplificació i no es realitzen distincions de si els usuaris pertanyen a la unitat de protocol o a la unitat fotogràfica, així doncs, en aquest diagrama es contempen només dos tipus d'actors, que seran els usuaris de consulta i els usuaris amb permís de creació.



Anotació: s'haurà de tenir en compte que tots els casos d'ús aquí exposats depenen d'un cas d'ús no exposat en el diagrama anterior per motius de senzillesa, aquest cas d'ús el podríem anomenar Autenticar-se, amb el qual l'usuari introdueix les dades necessàries per accedir a l'aplicació amb el perfil assignat.

2.2.2 Documentació textual

Cas d'ús número 1: "Crear expedient o acte"

Actors: usuaris amb permís de creació (Gestors) de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari s'ha validat i vol donar d'alta un acte o esdeveniment , que no existeix.

Postcondició: l'acte o esdeveniment ha quedat registrat.

Casos d'ús relacionats: "Autenticar-se"

Resum de la funcionalitat: El procés permetrà donar d'alta actes protocol·laris i esdeveniments o expedients que no es corresponen amb actes protocol·laris.

Flux d' events principal:

Es sol·licitarà la informació obligatòria per donar d'alta l'acte o esdeveniment, en cas de que es tracti d'un event d'aquest tipus.

- La descripció de l'acte
- La data de realització
- El tipus d'acte
- El lloc
- El municipi
- La comarca
- El país

En cas de que no es tracti ni de un esdeveniment d'un acte, sinó de que es tracti d'un expedient, només es sol·licitarà:

- Descripció
- Data de realització
- Tipus

Una vegada omplerta la informació anterior l'usuari si ho desitja podrà informar els següents camps:

- Hora Inicial
- Data Inici
- Hora final
- Data de Fi

- Intervinents a l'acte
- Ciutats visitades (si cal)
- Personatges fotografiats
- Temàtica

En el cas de que es tracti d'un acte de protocol es sol·licitaran a més a més les següents dades:

- Departament responsable
- Nombre d'assistents

Una vegada informades totes les dades i després de demanar confirmació per part de l'usuari, les dades quedaran emmagatzemades, generant-ne un codi identificatiu de l'acte consistent en *aammnn*, on *aa* són dos dígitos que identifiquen l'any, *mm* són dos dígitos que identifiquen el mes i *nn* un número seqüencial que s'inicialitzarà a 01 al inici de cada mes.

Observacions: cal distingir, entre actes protocol·laris, esdeveniments de la unitat de fotografia, i expedients de la unitat de protocol. Aquests seran referenciats seguint els següents valors:

- 1: Acte protocol·lari.
- 2: Esdeveniment Unitat de Fotografia.
- 3: Expedient Unitat de Protocol.

Cas d'ús número 2: "Cercar actes"

Actors: usuaris amb permís de consulta sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari prèvia validació accedeix a la cerca d'actes.

Postcondició: l'usuari obté un llistat amb els actes resultat de la cerca.

Casos d'ús relacionats: "Autenticar-se" i "Consultar acte"

Resum de la funcionalitat: L'usuari podrà realitzar la cerca sobre els actes protocol·laris, sobre els esdeveniments o bé, sobre expedients. Per realitzar la cerca disposarà de possibilitats de cercar els actes o events mitjançant un calendari, i a més a més, podrà acotar aquesta cerca mitjançant la introducció de: la descripció de l'acte, la data de inici, la data final, la hora de inici, el tipus d'acte, el departament responsable, el lloc de realització, el municipi, la comarca, el país, la temàtica i els personatges fotografiats.

Flux d'events principal: l'usuari accedeix a la zona de cerca i omple les dades o criteris desitjats, per posteriorment efectuar l'acció de cerca pròpiament dita. Com a resultat obté una llista amb els actes o events que compleixen els criteris de cerca.

Observacions: l'accés als actes, esdeveniments, o bé expedients, siguin de la Unitat de protocol o bé de la Unitat de Fotografia vindran condicionats per el tipus d'usuari.

Cas d'ús número 3: "Consultar acte"

Actors: usuaris amb permís de consulta sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari mitjançant la cerca d'actes ha obtingut el llistat d'actes desitjat, i es decideix a triar un.

Postcondició: l'usuari obté les dades de l'acte desitjat.

Casos d'ús relacionats: "Autenticar-se", "Cercar actes", "Modificar expedient o acte", "Eliminar expedient o acte" i "Accedir a Fotografies" (veure el diagrama de casos d'ús número 2).

Resum de la funcionalitat: es visualitzen totes les dades d'un acte, esdeveniment o expedient, concretament del que ha triat l'usuari. Les dades que es mostren seran la descripció de l'acte, la data i hora d'inici, la data i hora de fi, el tipus d'acte, el lloc, el municipi, la comarca, el país, el departament responsable, els personatges fotografiats i les carpetes amb la relació de documents continguts.

Flux d'events principal: l'usuari prèvia cercar dels actes que segueixen un determinat criteri, tria un acte i aquest és visualitzat.

Observacions: A banda de les comentades per al cas d'ús número dos, cal dir, que aquest cas d'ús té sentit, sempre i quan existeixin perfils de consulta i modificació diferents, de no ésser així, aquest cas d'ús i el següent (Modificar expedient o acte) es podrien unificar, ja que la modificació per defecte comporta la consulta de les dades.

Cas d'ús número 4: "Modificar expedient o acte"

Actors: usuaris amb permís de creació o modificació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari accedeix a les dades d'un acte en particular.

Postcondició: les dades de l'acte són modificades.

Casos d'ús relacionats: "Autenticar-se" i "Consultar Acte"

Resum de la funcionalitat: l'usuari accedeix o consulta un acte determinat, i modifica les dades d'aquest. Les possibles dades a modificar són totes les comentades per al cas d'ús o guió número 3 ("Consultar Acte").

Flux d'events principal: l'usuari accedeix o consulta un acte determinat, modifica les dades d'aquest, i realitza l'acció pròpiament dita de modificació de l'acte, una vegada fet això se li informa a l'usuari de la correcta modificació de l'acte, o en cas contrari li adverteix d'una situació errònia.

Observacions: l'accés als actes, esdeveniments, o bé expedients, siguin de la Unitat de protocol o bé de la Unitat de Fotografia vindran condicionats per el tipus d'usuari.

Cas d'ús número 5: "Eliminar expedient o acte"

Actors: usuaris amb permís de creació o modificació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari accedeix a l'acte que desitja eliminar.

Postcondició: l'acte triat queda anul·lat.

Casos d'ús relacionats: "Autenticar-se" i "Consultar Acte"

Resum de la funcionalitat: l'usuari accedeix a l'acte que desitja eliminar, posteriorment l'usuari dur a terme l'eliminació de l'acte pròpiament dita, i es realitza per tant, l'eliminació d'aquest.

Flux d'events principal: l'usuari accedeix o consulta un acte determinat, posteriorment l'usuari executa l'acció d'eliminar o donar de baixa l'acte, fet que comporta confirmació per part de l'usuari, duta a terme la confirmació es realitza l'eliminació d'aquest i se li informa a l'usuari de la realització satisfactòria de l'acció.

Cas d'ús número 6: "Afegir intervinent"

Actors: usuaris amb permís de creació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari crea o accedeix/consulta a un determinat acte.

Postcondició: l'usuari ha afegit un intervinent a un acte determinat.

Casos d'ús relacionats: "Crear expedient i acte", "Modificar expedient o acte", "Modificar tasca"

Resum de la funcionalitat: l'usuari afegeix un intervinent a un acte.

Flux d'events principal: l'usuari que està consultant o creant un acte, decideix afegir un intervinent, a continuació se li mostra una llista d'intervinents, aquest tria un de la llista, informa de la seva tasca i a continuació realitza pròpiament l'acció d'afegir-lo a l'acte.

Cas d'ús número 7: "Afegir assistent"

Actors: usuaris amb permís de creació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari crea o accedeix/consulta a un determinat acte.

Postcondició: l'usuari ha afegit un assistent a un acte determinat.

Casos d'ús relacionats: "Crear expedient i acte", "Modificar expedient o acte", "Modificar càrrec"

Resum de la funcionalitat: l'usuari afegeix un assistent a un acte.

Flux d'events principal: l'usuari que està consultant o creant un acte, decideix afegir un assistent, a continuació se li mostra la llista d'assistents/persones, aquest tria un de la llista, informa el seu càrrec i tasca i a continuació realitza pròpiament l'acció d'afegir-lo a l'acte.

Cas d'ús número 8: "Modificar/afegir tasca"

Precondició: l'usuari es troba afegint un intervinent.

Postcondició: l'usuari ha especificat la tasca de l'intervinent.

Casos d'ús relacionats: "Afegir intervinent"

Resum de la funcionalitat: l'usuari especifica la tasca de l'intervinent.

Flux d'events principal: l'usuari que es troba afegint un intervinent decideix especificar la tasca d'aquest, acte seguit introdueix la tasca que realitzarà i realitza l'acte de introduir la tasca pròpiament.

Cas d'ús número 9: "Modificar/afegir càrrec"

Precondició: l'usuari es troba afegint un assistent.

Postcondició: l'usuari ha especificat el càrrec de l'assistent.

Casos d'ús relacionats: "Afegir assistent"

Resum de la funcionalitat: l'usuari especifica el càrrec d'un assistent.

Flux d'events principal: l'usuari que es troba afegint un assistent decideix especificar el càrrec d'aquest, acte seguit introdueix el càrrec que realitzarà i realitza l'acte de introduir el càrrec pròpiament.

Cas d'ús número 10: "Eliminar intervinent"

Actors: usuaris amb permís de creació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari crea o accedeix/consulta a un determinat acte.

Postcondició: l'usuari ha eliminat un intervinent d'un acte determinat.

Casos d'ús relacionats: "Crear expedient i acte", "Modificar expedient o acte"

Resum de la funcionalitat: l'usuari elimina un intervinent d'un acte.

Flux d'events principal: l'usuari que està consultant o creant un acte, decideix eliminar un intervinent, selecciona el intervinent desitjat de la llista de intervinents i l'esborra.

Cas d'ús número 11: "Eliminar assistent"

Actors: usuaris amb permís de creació (Gestors) sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari crea o accedeix/consulta a un determinat acte.

Postcondició: l'usuari ha eliminat un assistent a un acte determinat.

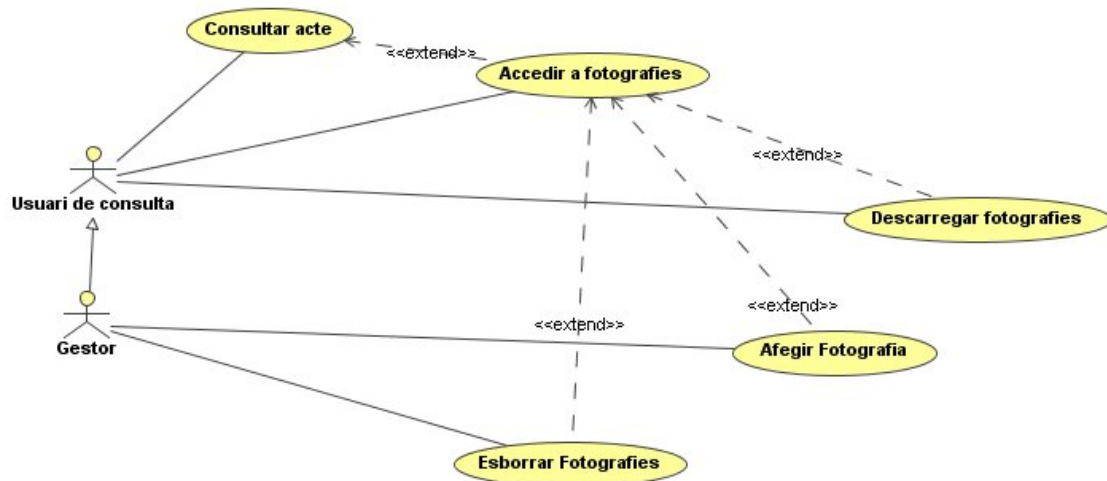
Casos d'ús relacionats: "Crear expedient i acte", "Modificar expedient o acte"

Resum de la funcionalitat: l'usuari elimina un assistent a un acte.

Flux d'events principal: l'usuari que està consultant o creant un acte, decideix eliminar un assistent, selecciona l'assistent desitjat de la llista d'assistents i l'esborra.

2.2.3 Diagrama de casos d'ús per a la gestió de fotografies

Els casos d'ús que es presenten en el següent diagrama fan referència a la gestió de fotografies associades a un acte, i per tant, en la majoria de casos per accedir a la part funcional de gestió de fotografies s'haurà d'haver accedit a l'acte vinculat prèviament.



2.2.4 Documentació textual

Cas d'ús número 12: "Accedir a fotografies"

Actors: usuaris amb permís de consulta sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari accedeix/consulta un determinat acte.

Postcondició: l'usuari visualitza les fotos associades i pot accedir a la resta de casos d'ús vinculats.

Casos d'ús relacionats: "Consultar acte", "Afegir fotografies", "Eborrar fotografies", i "Descarregar fotografies"

Resum de la funcionalitat: Permet accedir a les fotografies d'un acte i fa de interfície per a la resta d'accions relatives a la gestió de fotografies vinculades a l'acte.

Flux d'events principal: l'usuari consulta un acte i des d'aquí accedeix a la funcionalitat citada.

Cas d'ús número 13: "Afegir fotografia"

Actors: usuaris gestors de la Unitat de Fotografia.

Precondició: l'usuari de la unitat fotogràfica accedeix/consulta les fotografies d'un determinat acte.

Postcondició: l'usuari afegeix una nova fotografia a l'acte.

Casos d'ús relacionats: "Accedir a fotografies"

Resum de la funcionalitat: permet afegir una nova fotografia a l'acte.

Flux d'events principal: l'usuari accedeix a les fotografies d'un acte des d'aquí decideix afegir una nova fotografia, tria una fotografia emmagatzemada al seu ordinador i realitza la pujada al servidor, fet això se li demana confirmació i posteriorment la fotografia es definitivament afegida a l'acte.

Cas d'ús número 14: “Esborrar fotografies”

Actors: usuaris gestors de la Unitat de Fotografia.

Precondició: l'usuari de la unitat fotogràfica accedeix/consulta les fotografies d'un determinat acte.

Postcondició: l'usuari ha esborrat algunes de les fotografies vinculades a l'acte.

Casos d'ús relacionats: “Accedir a fotografies”

Resum de la funcionalitat: permet esborrar fotografies vinculades a l'acte.

Flux d'events principal: l'usuari accedeix a les fotografies d'un acte, a continuació selecciona les fotografies que desitja esborrar, se li demana confirmació i les fotografies són eliminades.

Cas d'ús número 15: “Descarregar fotografies”

Actors: usuaris amb permís de consulta sobre els actes de la Unitat de Protocol, o bé, de la Unitat de Fotografia.

Precondició: l'usuari accedeix a les fotografies de l'acte.

Postcondició: l'usuari aconsegueix descarregar les fotografies desitjades.

Casos d'ús relacionats: “Accedir a fotografies”

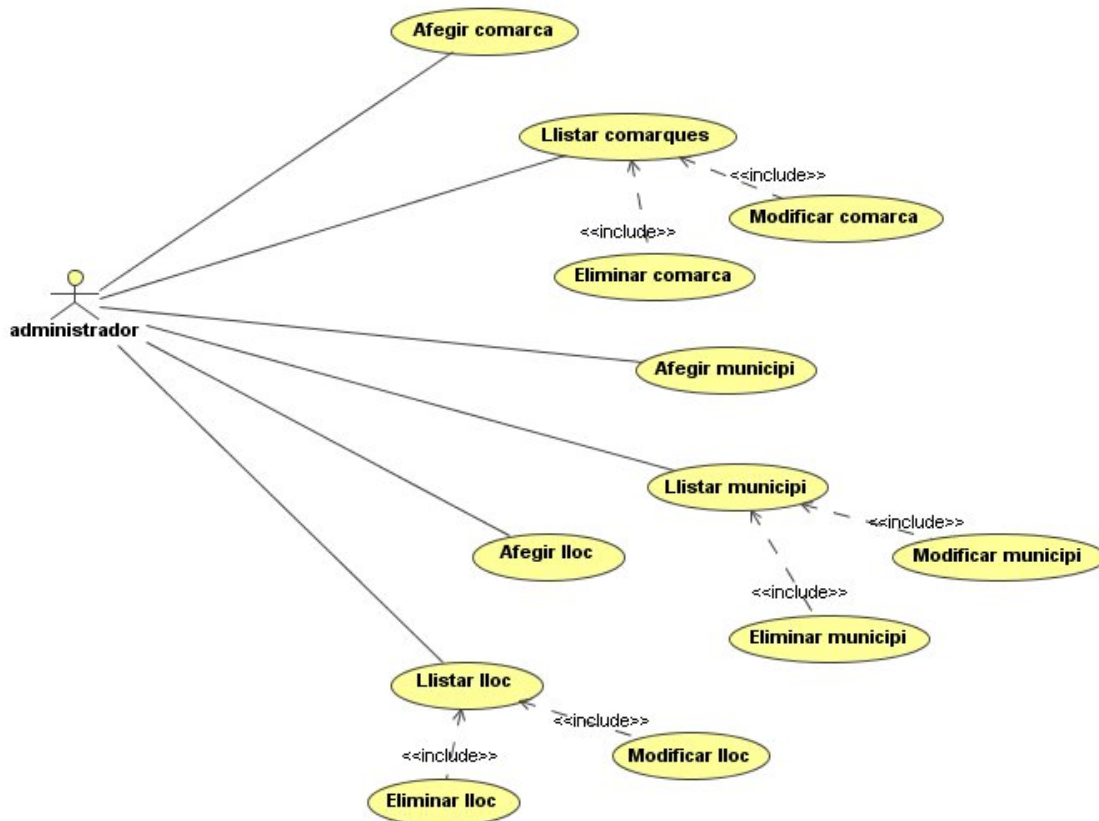
Resum de la funcionalitat: permet descarregar fotografies associades a l'acte.

Flux d'events principal: l'usuari accedeix a les fotografies d'un acte, a continuació selecciona o marca les fotografies que desitja descarregar, selecciona un directori o carpeta de destí i descarrega les fotografies.

2.2.5 Diagrama de casos d'ús per a la gestió de dades bàsiques

A continuació es mostra el diagrama de casos d'ús per al manteniment de dades bàsiques o auxiliars. Inicialment es varen agrupar varis casos d'ús "senzills" en un de sol, no estava previst implementar les funcionalitats relatives a alguns d'aquests casos d'ús, i com finalment han estat implementades per aquest motiu s'inclouen aquests casos d'ús.

Cal dir, no obstant que s'han eliminat els casos d'ús genèrics no implementats, com són: gestionar responsables, gestionar personatges fotografiats, gestionar assistents, gestionar intervinents, gestionar tasques, gestionar càrrecs, gestionar destinataris, gestionar temes, gestionar països i gestionar usuaris.



Donada la senzillesa dels casos d'ús, donat que la finalitat d'aquesta memòria és fer una síntesi del treball dut a terme i donat l'espai limitat d'aquesta no s'ha cregut necessari incloure la documentació textual dels casos d'ús abans mencionats.

2.2.6 Guió del model de perfils i seguretat

L'aplicació disposarà de diferents perfils d'accés.

Un perfil d'accés vindrà determinat pels següents atributs:

- Nom del perfil.
- Tipus d'accés:
 - F: Unitat de Fotografia, accedirà als tipus d'actes 1 i 2.
 - P: Unitat de Protocol, accedirà als tipus d'actes 1 i 3.
 - A: Altres, accediran als tipus d'actes 1.

 - 1: Acte protocol·lari.
 - 2: Esdeveniment Unitat de Fotografia.
 - 3: Expedient Unitat de Protocol.
- Consulta. SI/NO es permet la consulta de les dades bàsiques dels actes.
- Actualització. SI/NO es permet l'actualització de les dades bàsiques dels actes.
- Manteniment de fotografies. SI/NO es permet actualitzar les fotografies dels actes.
- Accés fotografies. SI/NO es permet accedir a les fotografies dels actes.
- Descàrrega fotografies. SI/NO es permet descarregar fotografies.
- Realització de Peticions. SI/NO es permet realitzar peticions de còpies en paper.

A banda dels perfils determinats per els atributs citats anteriorment que defineixen o diferencien el que serien usuaris de consulta i gestors, està contemplat el perfil de l'administrador amb accés total a totes les funcionalitats de l'aplicació i a qualsevol tipus d'acte.

NOTA: Aquesta secció no ha estat duta a terme donada la complexitat d'aquesta, i la manca de temps disponible per la seva implementació, no obstant es manté en la memòria per tal de reflectir i distingir autenticació (sí implementada) respecte autenticació. A nivell conceptual crec oportú reflectir-ho.

2.2.7 Cas d'ús Autenticar-se

En molts casos d'ús dels abans especificats apareix el cas d'ús Autenticar-se que es relaciona o del que depenen altres casos d'ús, aquest es presenta a continuació:

Actors: usuaris (sense determinar tipus concret)

Precondició: l'usuari accedeix a l'aplicació.

Postcondició: es permet l'accés a l'aplicació i el tipus d'usuari està determinat.

Resum de la funcionalitat: permet a l'usuari validar-se i dona pas a la resta de funcionalitats de l'aplicació.

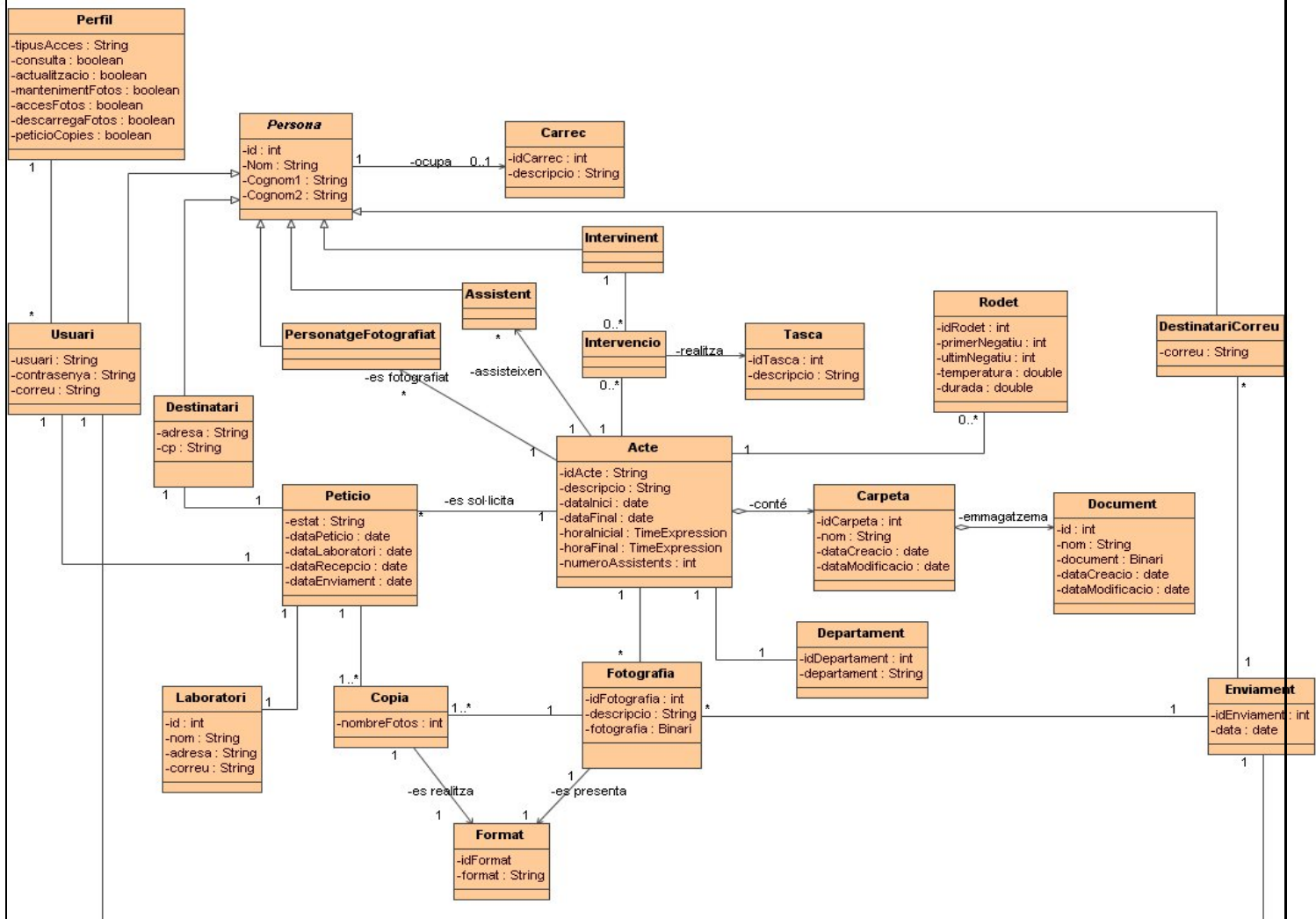
Flux d'events principal: l'usuari introdueix les seves dades, nom d'usuari i contrasenya, una vegada fet això, realitza l'acció d'autenticació pròpiament dita, si l'usuari és vàlid, aquest accedeix a l'aplicació amb un perfil donat..

2.3 Diagrama de classes

A continuació es presenten els diagrames de classes del model, s'ha fet una divisió en dos diagrames per motius de claredat i espai.

2.3.1 Diagrama del Model – I

Aquest diagrama representa les relacions entre les diverses classes del model, no obstant per tenir una idea concisa d'aquestes relacions cal estudiar també el segon diagrama, doncs només es representen algunes de les relacions i classes del model però no totes.



S'han utilitzat tipus genèrics i no adreçats a un llenguatge de programació en particular, a continuació es presenten els significats:

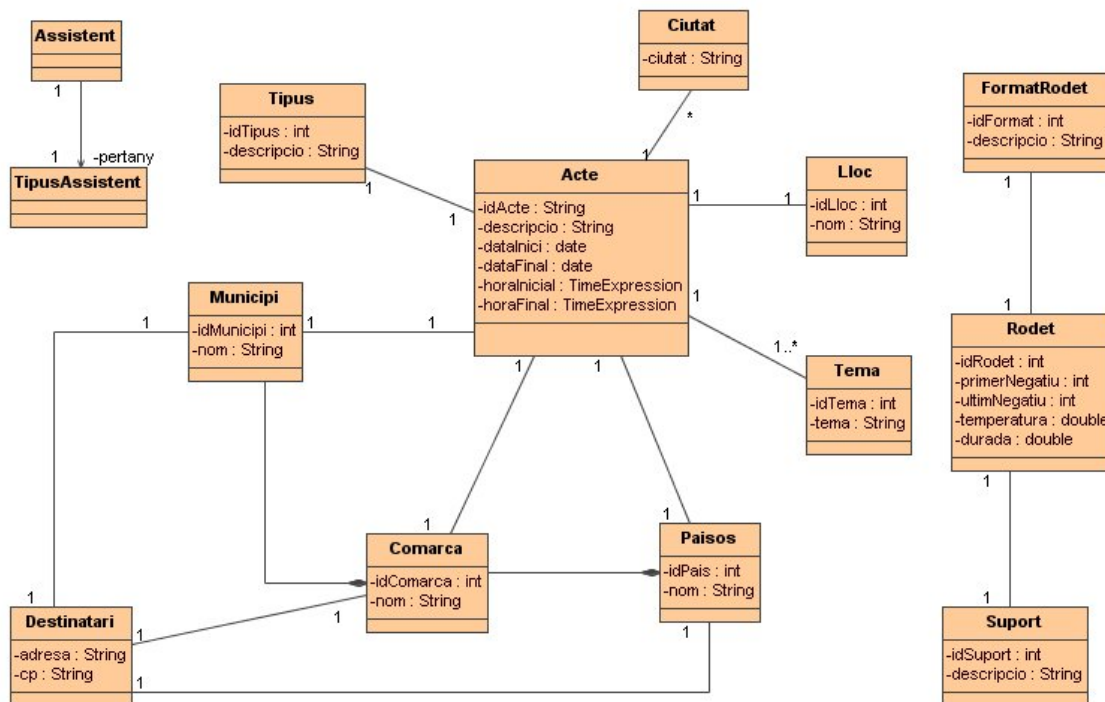
- String: cadena de text
- int: tipus numèrics (primitiu), enter.
- double: tipus decimal (primitiu), real.
- date: tipus data (inclou l'hora si cal).
- boolean: tipus booleà (primitiu).
- TimeExpression: expressió de temps, utilitzada per emmagatzemar l'hora.
- Binari: classe per emmagatzemar dades binàries, utilitzada per emmagatzemar documents i fotografies.

Per altra banda cal comentar, que els objectes pertanyents a la classe Usuari es relacionen amb les classes Acte, Petició, Enviament, Carpeta i Document doncs es vol tenir constància dels usuaris que han creat aquests objectes, aquestes relacions no són reflectides en el diagrama UML anterior.

Cal esmentar també, que una Petició serà una sol·licitud al laboratori fotogràfic de còpies d'unes determinades fotografies i un enviament un enviament de correu electrònic d'unes determinades fotografies, per aquest motiu es diferencien els Destinataris (classe Destinatari) del primer cas dels destinataris del segon cas (classe DestinatariCorreu).

2.3.2 Diagrama del Model – II

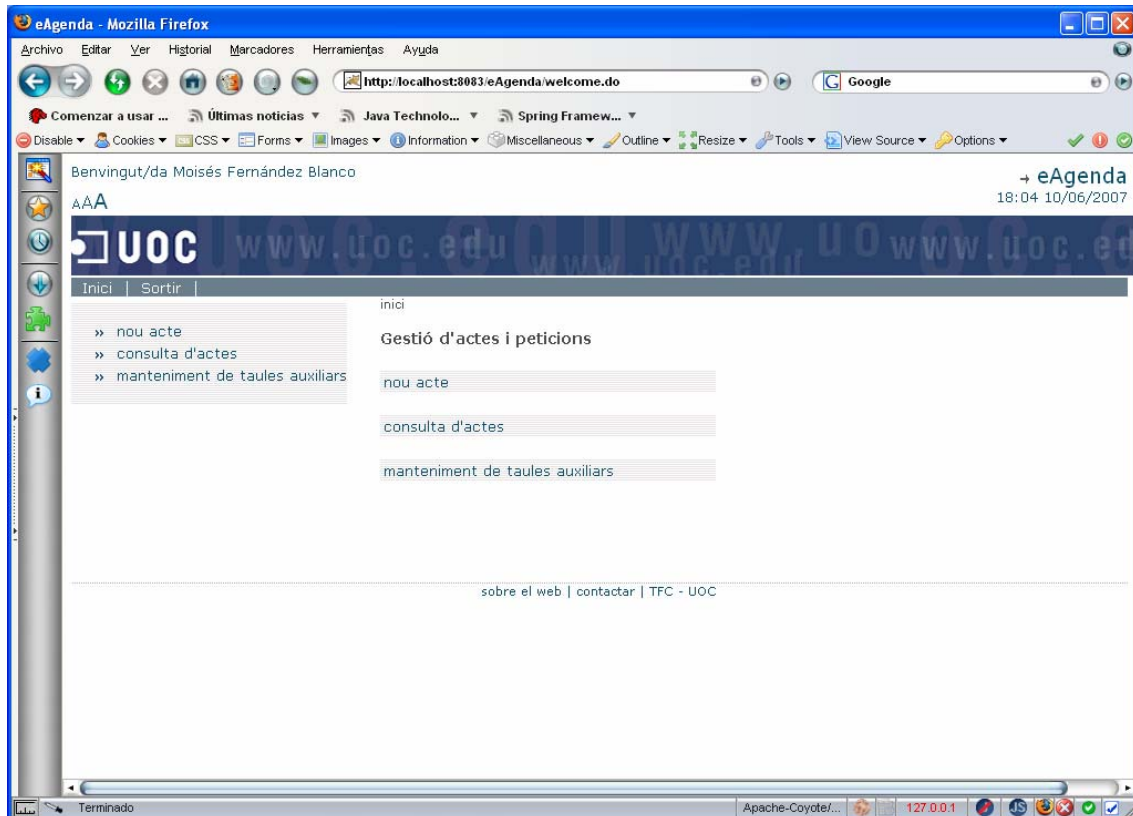
En el següent diagrama és presenten la resta de classes i relacions.



2.4 Especificació de pantalles

Menú principal

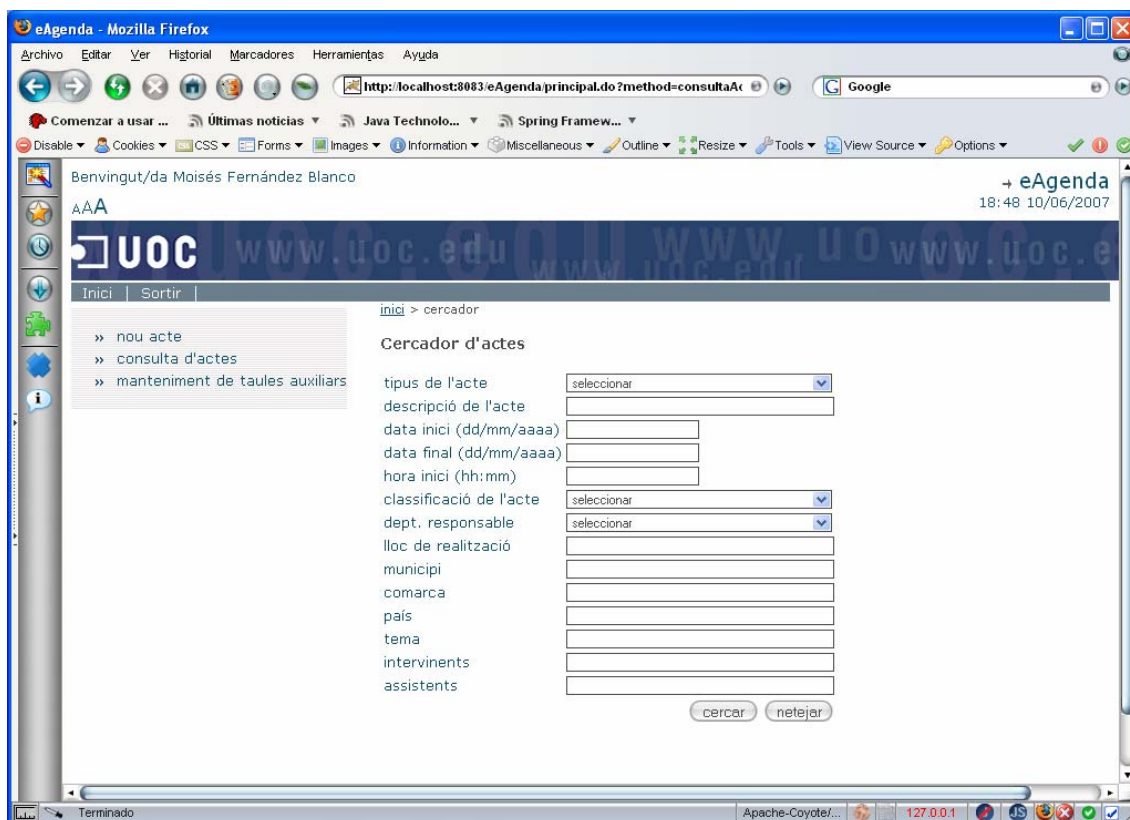
Aquesta és la pantalla a que accedeix l'usuari quan entra a l'aplicació. Presenta les principals opcions de l'aplicació *eAgenda*. Les opcions disponibles a aquest menú dependran del perfil d'accés de l'usuari.



Per tal d'accedir a l'opció corresponent només caldrà fer clic sobre l'opció desitjada.

Cerca d'actes

Aquesta pantalla s'utilitza per fer cerques a la base de dades d'actes, esdeveniments o expedients i s'arriba mitjançant l'enllaç "consulta d'actes", el cas d'ús vinculat és "Cercar actes".

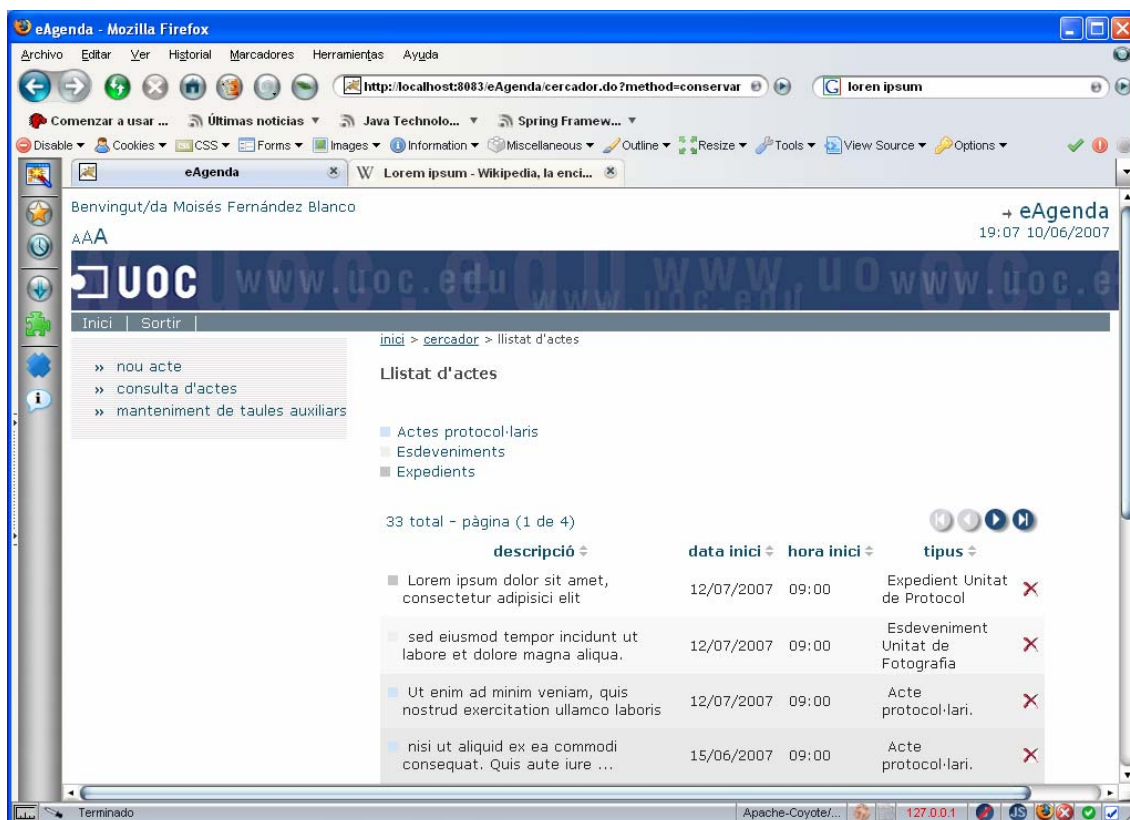


Nota: La capçalera i el peu de pàgina seran els mateixos per a totes les pantalles, i seran els que s'aprecien a la primera pantalla, encara que en aquesta darrera captura per les dimensions de la pantalla no es poden apreciar correctament.

El botó "netejar" permetrà esborrar els criteris de la pantalla. Per altra banda podrem utilitzar el fil d'Ariadna per tornar al menú principal, o bé, fer clic en el botó cercar per tal de realitzar la cerca.

Llistat d'actes

Aquesta pantalla mostrarà el resultat d'una cerca a la base de dades. Els casos d'ús relacionats són "Cercar actes", "Consultar actes" i "Eliminar expedient o acte".



Les icones amb els diferents colors permeten fer la distinció entre Actes protocol·laris, Esdeveniments i Expedients.

Per altra banda, l'aspa vermella permetrà anul·lar un acte, demanant abans confirmació per dur a terme l'esborrat.

Per accedir a les dades d'un acte només caldrà posicionar-nos a sobre de la filera relativa a un acte, fent clic a sobre d'aquest podrem accedir a les dades esmentades.

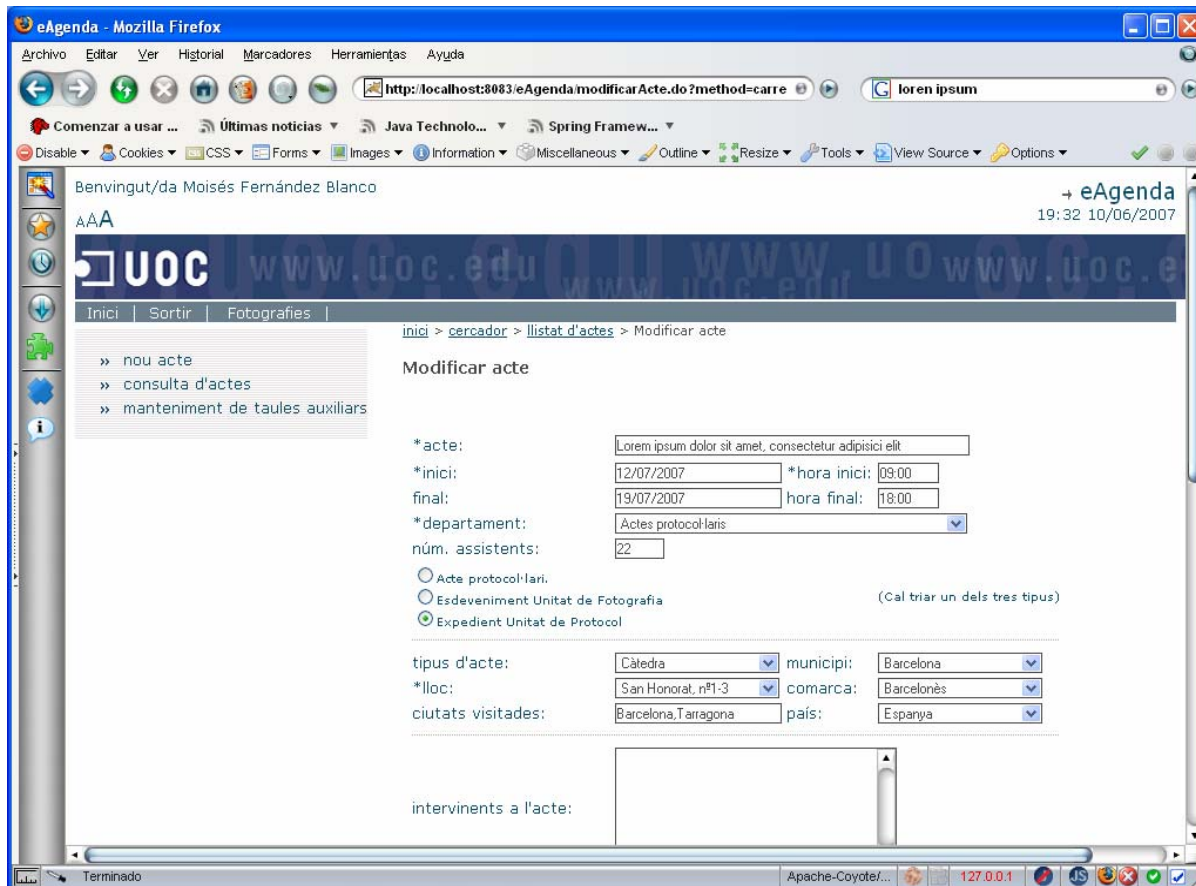
Cal dir, que els resultats de la cerca es mostraran en un llistat paginat d'un màxim de 10 resultats per pàgina, per altra banda, els botons amb les fletxes situats al cantó superior dret ens permetran moure'ns a través de les diferents pàgines del llistat, es mostraran habilitats (color blau fosc) o deshabilitats (color gris clar) en funció de les accions de que disposem.

El fil d'Ariadna ens permetrà tornar al inici o bé a la pantalla de cerca, per altra banda, el menú principal de l'esquerra ens permetrà accedir a les mateixes opcions que a les pantalles anteriors.

Afegir nou acte / consulta de l'acte

Havent seleccionat o fet clic en l'acte es mostren les dades d'aquest, aquesta pantalla permetrà l'edició o modificació de les dades de l'acte. Els casos d'ús relacionats són "cercar actes", "crear expedient o acte", "consultar acte", "modificar expedient o acte" (Vegis les observacions relatives al cas d'ús número tres, "Consultar acte", pàgines 14-15)

Captura 1:



Captura 2:

tipus d'acte: Càtedra municipi: Barcelona

*lloc: San Honorat, nº1-3 comarca: Barcelonès

ciutats visitades: Barcelona, Tarragona país: Espanya

intervinents a l'acte:

afegir eliminar

assistents a l'acte:

afegir eliminar

personatges fotografiats: Imma Tubella

afegir eliminar

temes: Informàtica i multimèdia

afegir eliminar

* Camps obligatoris

desar netejar

Passant a les funcionalitats de la pantalla, la barra de menú mostrarà la nova opció "Fotografies" que ens permetrà accedir a la gestió de fotografies fent clic a sobre d'aquesta.

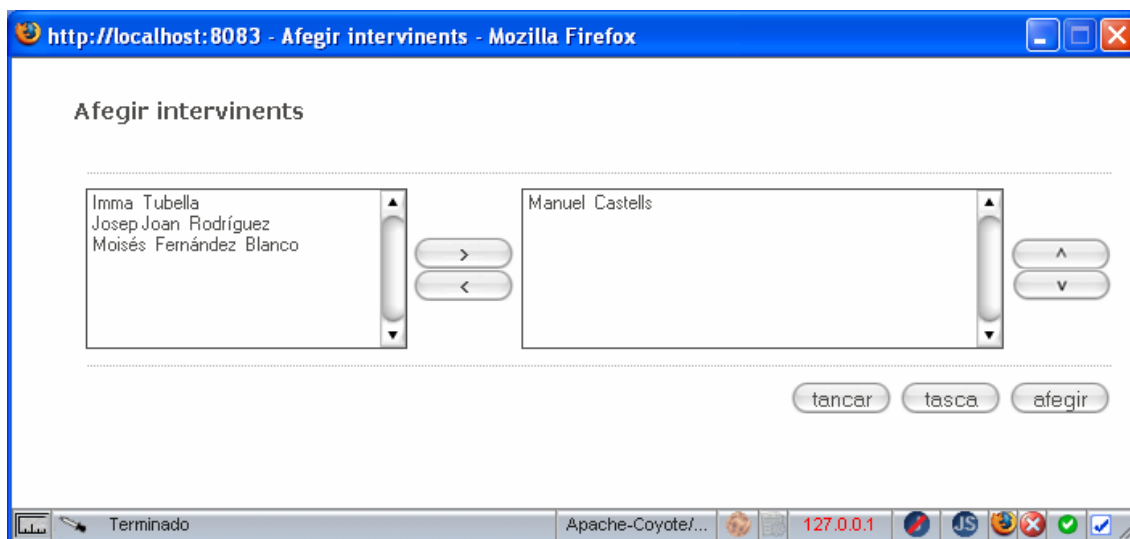
Es mostraran llistes de selecció múltiple conjuntament amb botons, per tal d'afegir o eliminar intervinents i assistents de l'acte, per a més detall consultar la funcionalitat d'afegir un nou acte (anàloga a la modificació).

El fil d'Ariadna ens permetrà tornar al llistat d'actes, a la cerca d'actes, o bé, a la pantalla d'inici. Per altra banda, cal dir que es mostren les mateixes opcions que per a pantalles anteriors en el menú de l'esquerra.

Cal dir, que aquesta pantalla podem accedir des de "Nou acte" en cas de voler donar d'alta un acte nou, o des de l'opció modificació en la consulta de l'acte.

Afegir/Eliminar intervinents

Aquesta pantalla ens permet afegir i eliminar les persones que intervien a la preparació de l'acte o a la realització d'aquest. Els casos d'ús relacionats són "afegir intervinent" i "eliminar intervinent".



Podrem seleccionar una persona de la llista de l'esquerra i utilitzar el boto ">" per afegir-la al llistat de persones que intervien a l'acte (llista de la dreta). També podem seleccionar una persona de la llista de intervinents i utilitzar el botó "<" per eliminar-la del llistat. El botó "modificar tasca" ens permetrà canviar la tasca predeterminada que realitza la persona a l'acte. A la llista de intervinents podem seleccionar-ne un i canviar la seva posició al llistat mitjançant els botons "^" i "v".

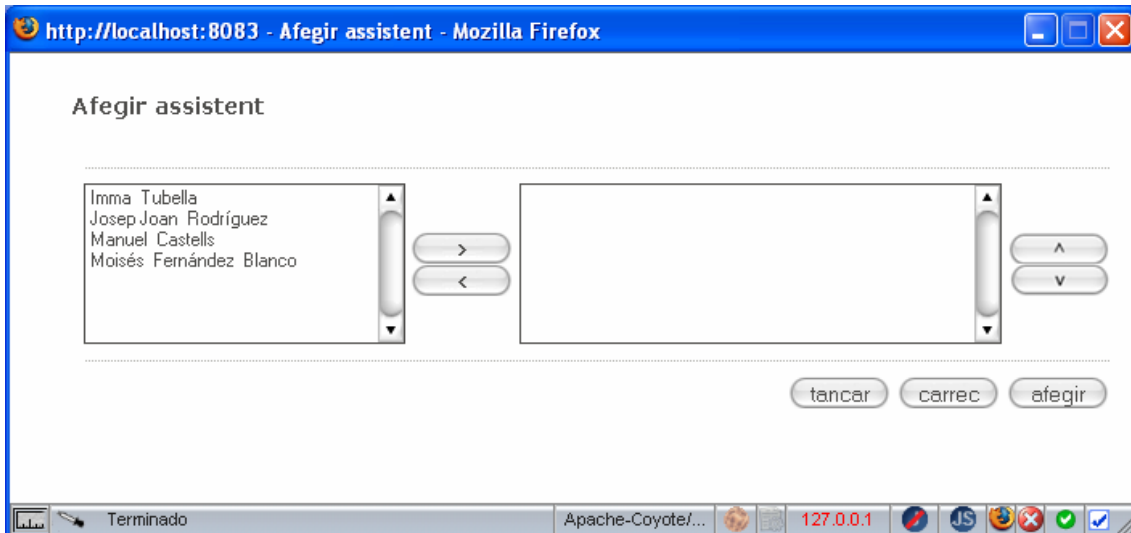
Modificar tasca

Aquesta pantalla ens permetrà canviar la tasca predeterminada que realitza la persona que intervé a la realització o a la preparació de l'acte. El cas d'ús relacionat és "Modificar tasca".



Afegir / Eliminar assistents

Aquesta pantalla ens permetrà afegir i eliminar les persones que assistiran a l'acte. Els casos d'ús relacionats són "Afegir assistent" i "Eliminar assistent".



Aquesta pantalla tindrà un funcionament similar al de la pantalla d'afegir/eliminar intervinents, així doncs, podrem seleccionar una persona de la llista de l'esquerra i utilitzar el botó ">" per tal d'afegir-lo al llistat d'assistents a l'acte (llista de la dreta). També podrem seleccionar una persona del llistat d'assistents i utilitzar el botó "<" per eliminar-lo d'aquest.

El botó "Modificar càrrec" ens permetrà canviar el càrrec predeterminat de la persona que assisteix a l'acte.

Podrem canviar la posició d'un assistent de la llista d'assistents a l'acte mitjançant els botons "∧" i "∨".

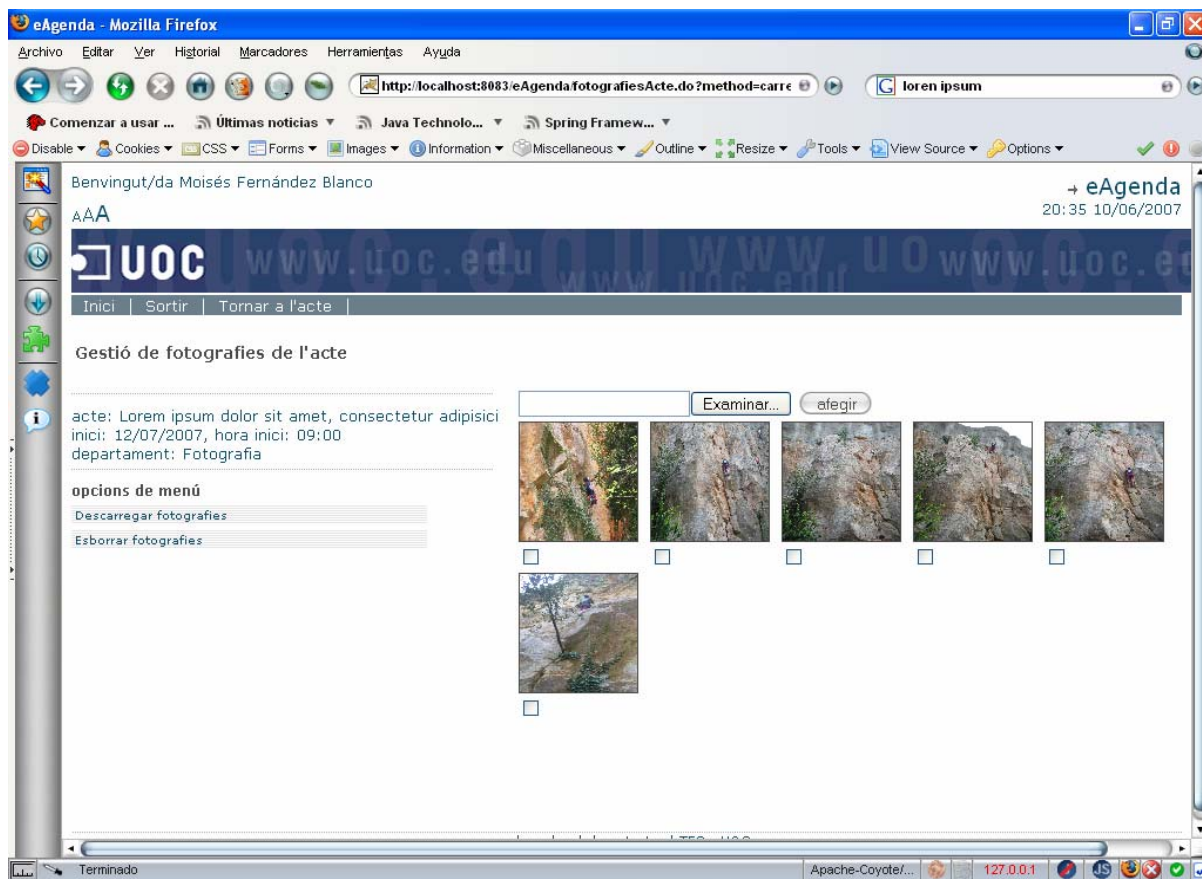
Modificar càrrec

Aquesta pantalla ens permet canviar el càrrec que té l'assistent a l'acte. El cas d'ús relacionat és "Modificar càrrec".



Gestió de fotografies

Els casos d'ús relacionats són: "Accedir a fotografies", "Afegir fotografia", "Esborrar fotografies" i "Descarregar fotografies". Aquesta pantalla ens mostra les fotografies en format digital de l'acte o esdeveniment i ens permet realitzar tot tipus de tasques amb aquestes.



La barra de menú ens permetrà tornar al acte al que estan vinculades les fotografies, així com, anar al inici o sortir del programa.

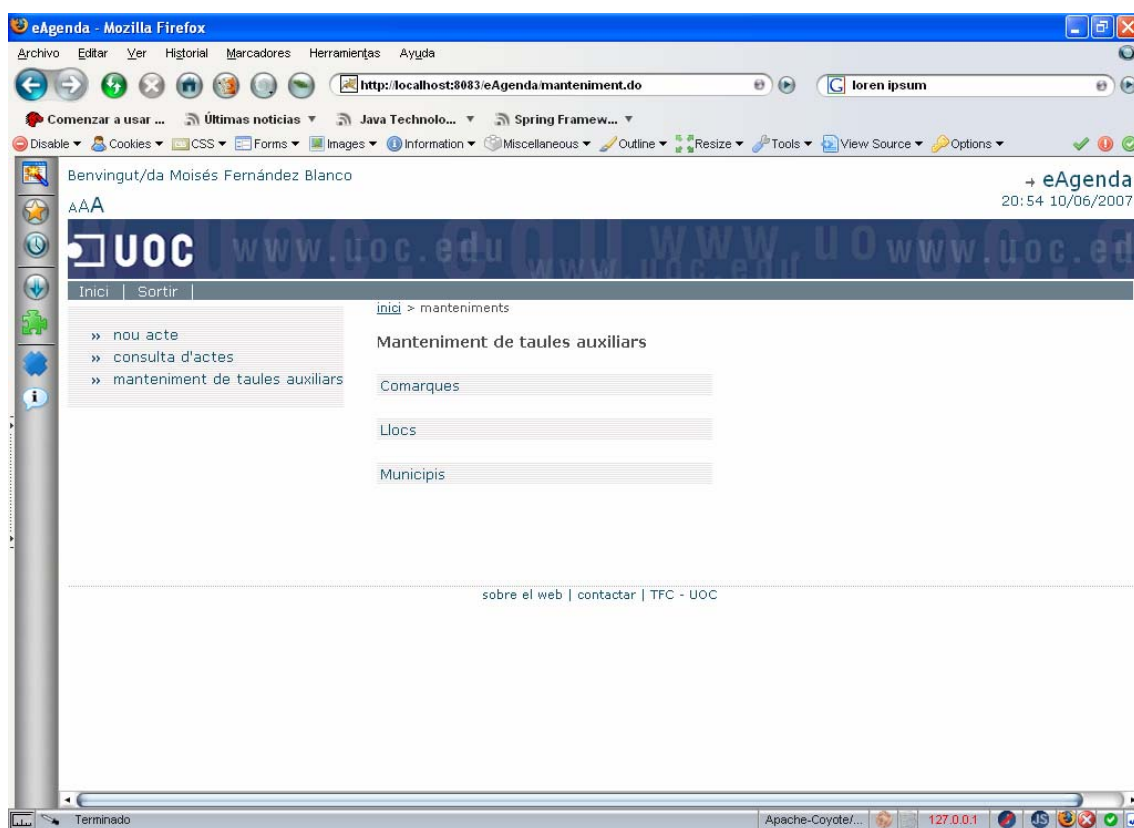
Prèvia selecció de la fotografia a afegir mitjançant el botó "examinar" es podran afegir noves fotografies (botó "afegir") o bé, esborrar les fotografies seleccionades (opció de menú "esborrar fotografies"). Al fer clic en el botó "examinar", l'usuari accedirà a una pantalla on haurà d'especificar l'arxiu de la fotografia.

Per altra banda, per tal de dur a terme l'esborrat de fotografies caldrà seleccionar-les prèviament mitjançant les caselles d'opció ubicades sota de cada fotografia.

L'usuari podrà descarregar les fotografies seleccionades (mitjançant les caselles d'opció) en format comprimit (.zip) a una unitat local fent ús de l'opció de menú "descarregar fotografies", a continuació es demanarà a quina unitat i carpeta voldrem emmagatzemar les fotografies.

Menú per al manteniment de taules auxiliars

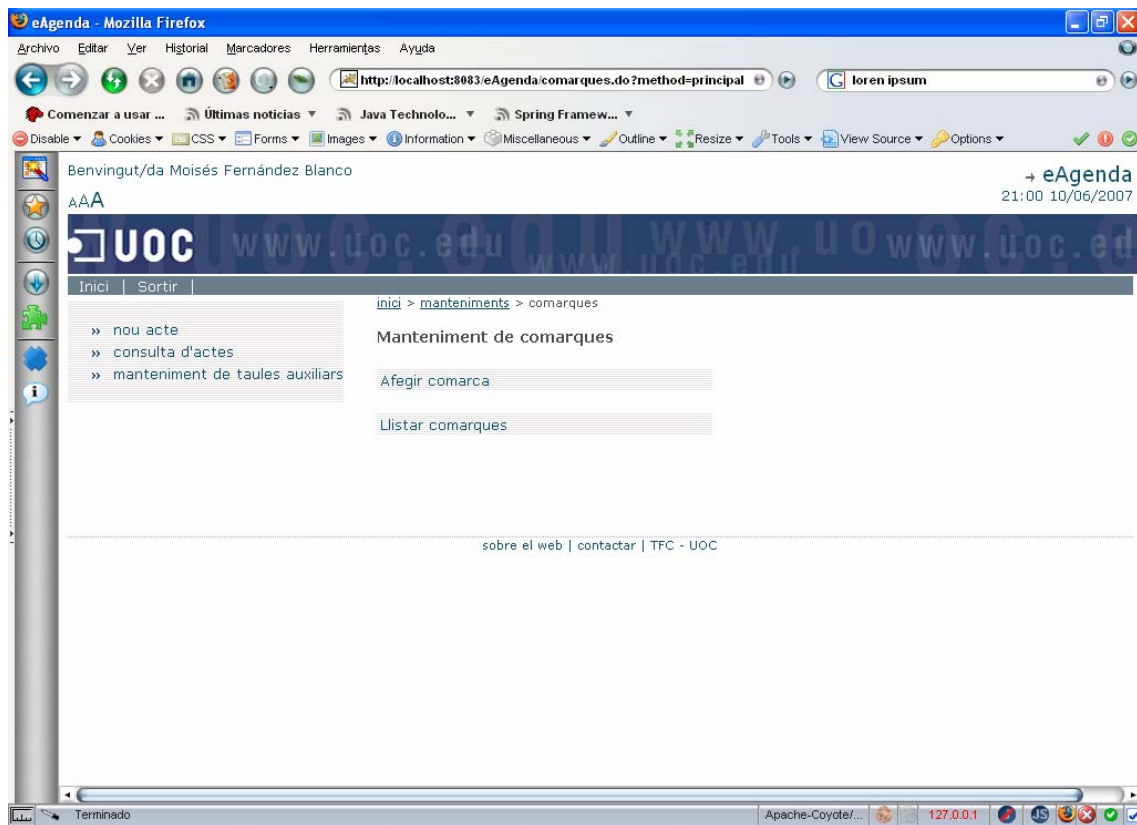
Aquesta pantalla dona accés a les diferents opcions de manteniment de taules auxiliars, així doncs, des d'aquesta es podrà accedir al manteniment de dades de les comarques, dels llocs i dels municipis.



Aquesta pantalla mantindrà totes les opcions d'accessibilitat implementades per la resta de pantalles anteriors.

Manteniment de comarques, municipis i llocs

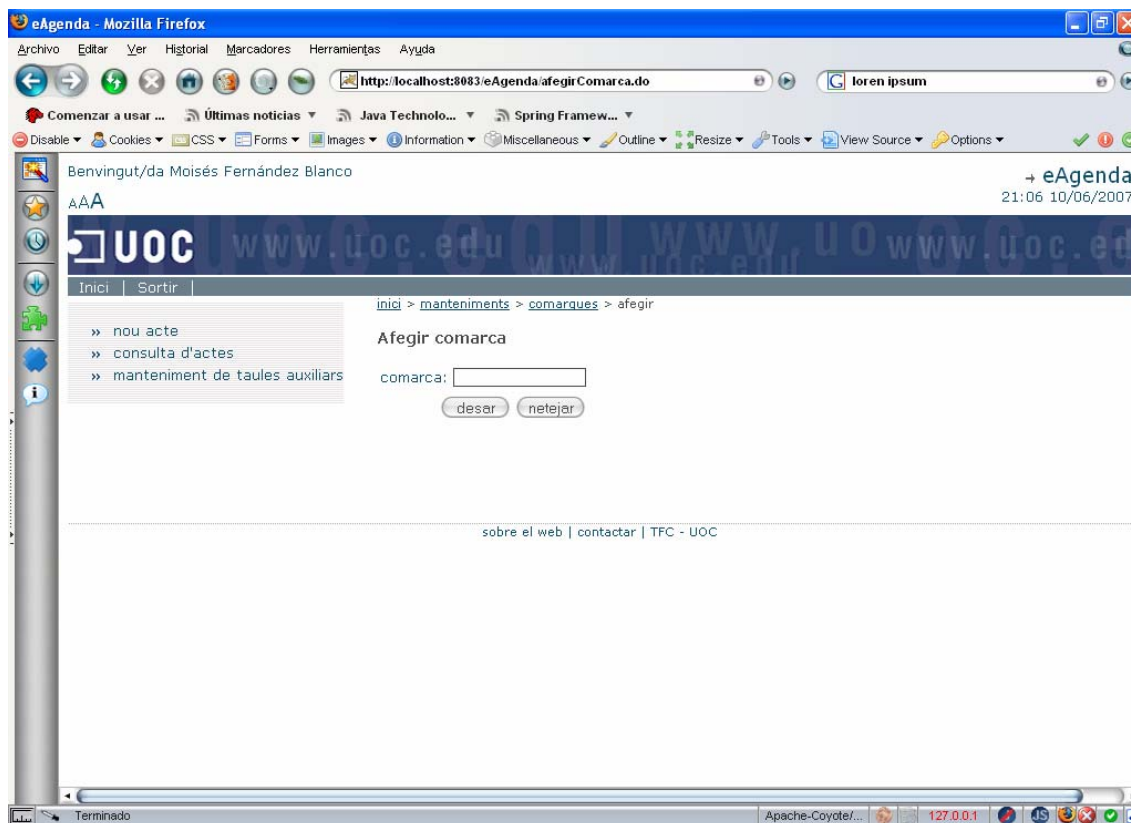
Aquesta pantalla permetrà accedir al manteniment de comarques o municipis o llocs. La pantalla donarà les opcions d'afegir i llistar una comarca, municipi o lloc. A continuació es presenta una captura de pantalla del manteniment de comarques, els manteniments de municipis i llocs, tindran pantalles anàlogues a aquesta.



Es mantindran les opcions d'accessibilitat implementades per a pantalles anteriors.

Afegir/Modificar comarca,municipi o lloc

Aquesta pantalla permetrà afegir una comarca, municipi o lloc. Els casos d'ús relacionats amb aquesta pantalla són "Afegir comarca", "Modificar comarca", "Afegir municipi", "Modificar municipi", "Afegir lloc" i "Modificar Lloc". A continuació es presenta una captura de pantalla per l'afegit de comarques, les pantalles d'afegir municipis i llocs són anàlogues aquesta.



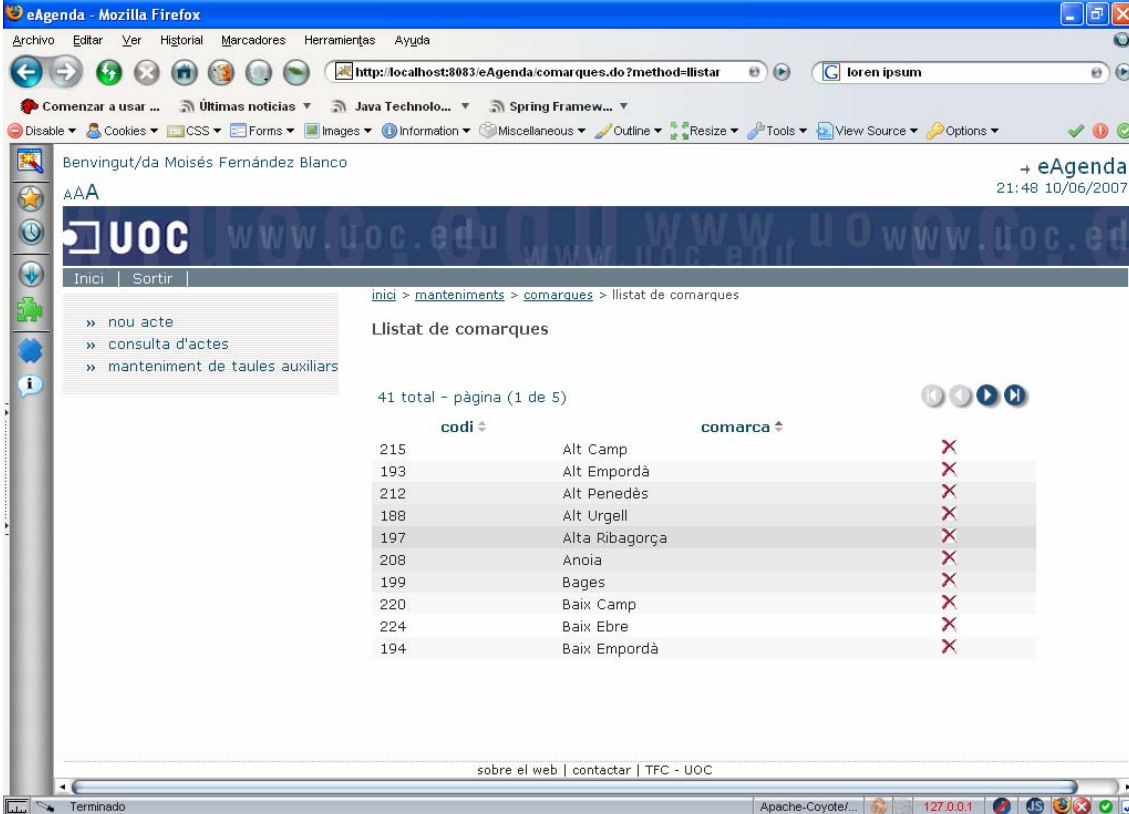
La pantalla tindrà un control per tal d'introduir el nom de la comarca o municipi o lloc, i presentarà botons per tal de desar i per tant, afegir l'entitat (comarca, municipi o lloc) corresponent, i per netejar la descripció de l'entitat.

Com en les pantalles anteriors es mantenen les opcions d'accessibilitat.

Per altra banda, cal comentar que les pantalles per tal de dur a terme la modificació de comarques, municipis i llocs, seran idèntiques a aquesta, en el camp de text apareixerà la descripció de l'entitat a modificar.

Llistat de comarques, municipis o llocs

Aquesta pantalla permetrà visualitzar el llistat de comarques, municipis o llocs. Els casos d'ús relacionats amb aquesta pantalla són "Llistar comarques", "Eliminar comarca", "Llistar municipis", "Eliminar municipi", "Llistar llocs" i "Eliminar lloc". A continuació es mostra una captura de pantalla del llistat de comarques, les pantalles del llistat de municipis i llocs són anàlogues a aquesta.



The screenshot shows a Mozilla Firefox browser window displaying the eAgenda application. The address bar shows the URL `http://localhost:8083/eAgenda/comarques.do?method=llistar`. The page content includes a navigation menu with options like "nou acte", "consulta d'actes", and "manteniment de taules auxiliars". The main content area displays a table titled "Llistat de comarques" with 41 total entries on page 1 of 5. The table has columns for "codi" and "comarca".

codi	comarca	
215	Alt Camp	X
193	Alt Empordà	X
212	Alt Penedès	X
188	Alt Urgell	X
197	Alta Ribagorça	X
208	Anoia	X
199	Bages	X
220	Baix Camp	X
224	Baix Ebre	X
194	Baix Empordà	X

En el llistat es mostrarà el codi i la descripció o nom de l'entitat en qüestió, a més a més es mostrarà de forma paginada de deu en deu amb controls per avançar de pàgina, retrocedir de pàgina, anar a la inicial i a la última pàgina. Per altra banda, es permetrà l'endreqat per codi o descripció de comarca, municipi o lloc.

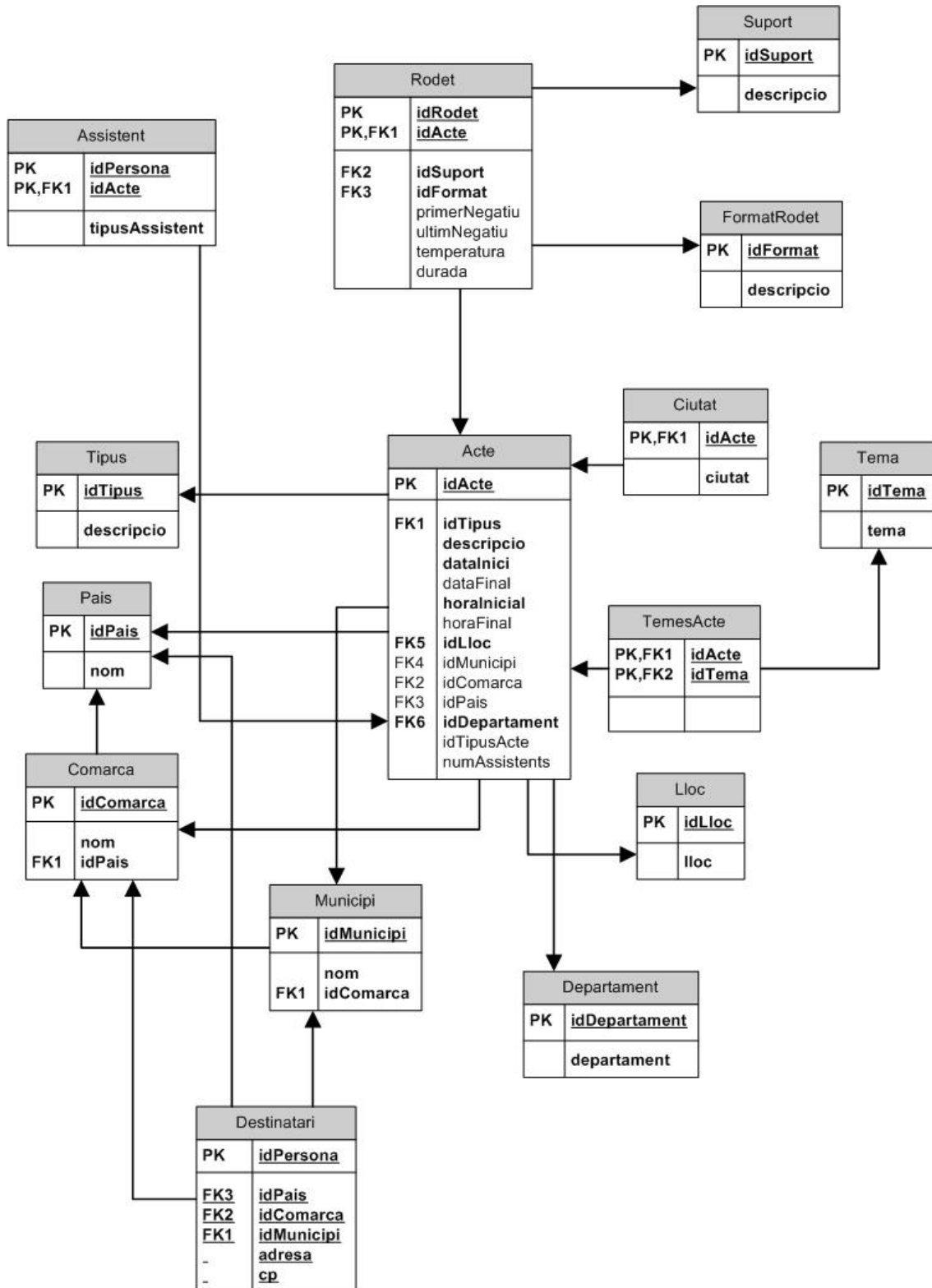
Des del mateix llistat es podrà dur a terme l'esborrat d'una entitat mitjançant l'aspa vermella que apareix a la dreta de cada filera.

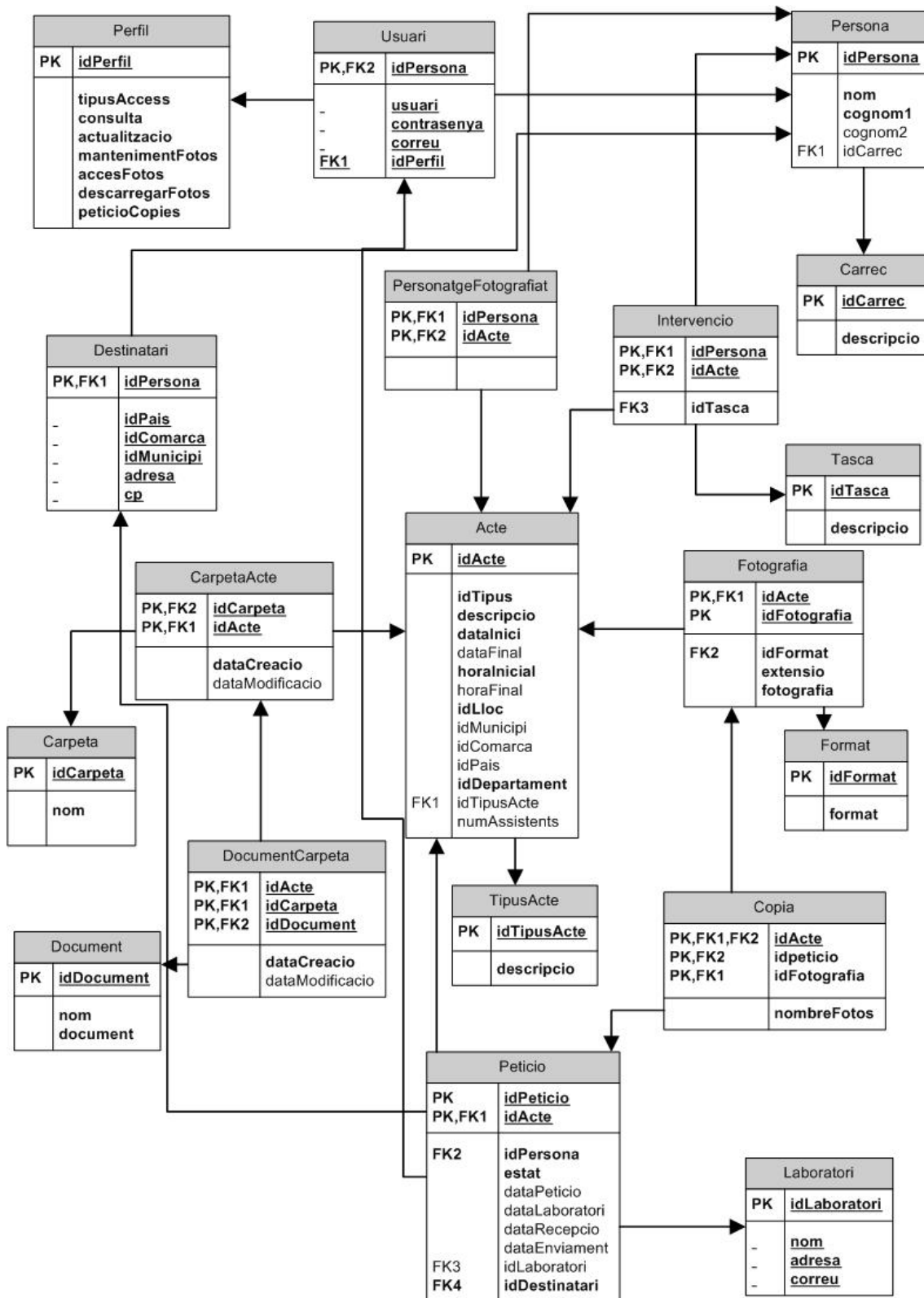
Com en les pantalles anteriors es mantenen les opcions d'accessibilitat.

2.5 Disseny de la base de dades

Les sigles PK identifiquen la clau primària, a més a més, els camps que la component apareixen subratllats.

Les sigles FK identifiquen les claus foranes i els camps en negreta són camps obligatoris.

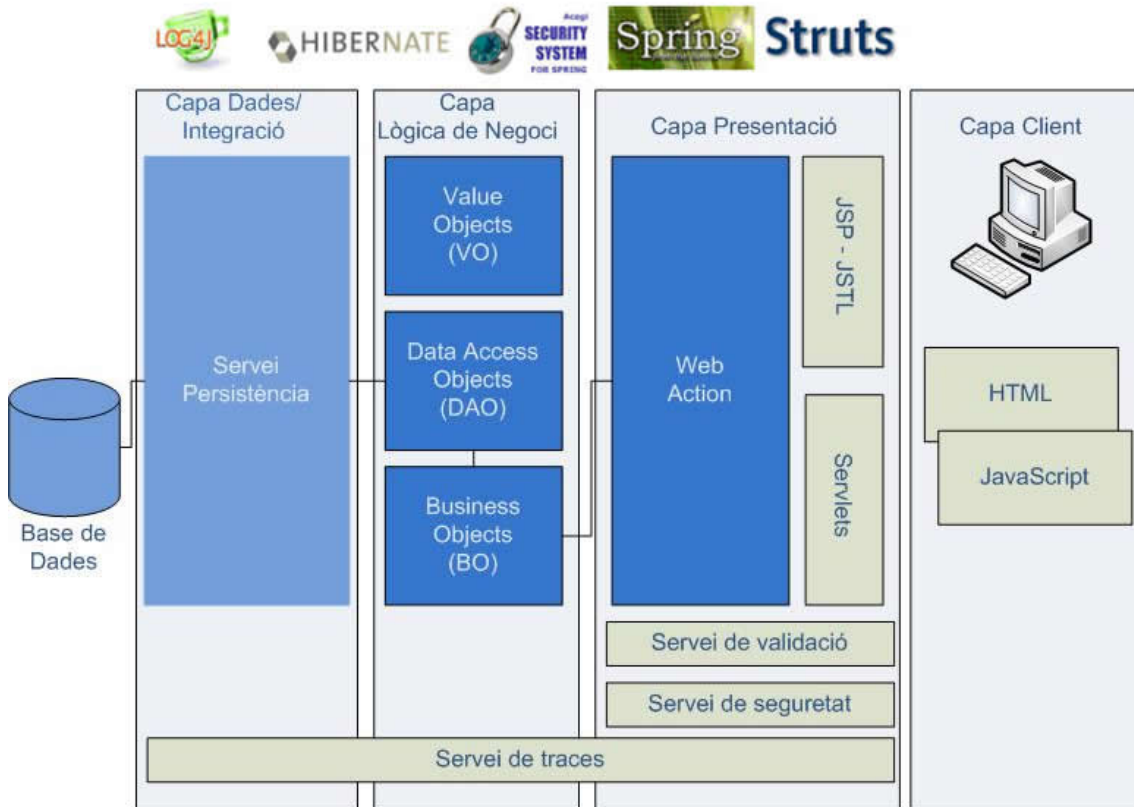




No s'han reflectit les relacions resultants dels enviaments de fotografies via correu electrònic, i per tant, aquestes dades no seran persistents.

3 Arquitectura

A continuació es presenta una figura amb el disseny de l'arquitectura utilitzada en el projecte, la figura es bastant descriptiva.



Del costat del client podem trobar el codi HTML generat per les JSP, conjuntament amb el llenguatge interpretat javascript, aquesta capa podria ésser perfectament englobada en la capa de presentació, s'ha separat d'aquesta forma per motius conceptuals. En la capa de presentació intervenen per una banda, les JSP, implementades fent ús de llibreries taglibs de JSTL i de Struts, i per altra banda, podríem englobar el servlet de Struts conjuntament amb els web action. Transversalment intervenen també en aquesta capa el servei de validació de Struts, i el servei de seguretat basada en filtres d'Acegi. Respecte la capa de la lògica de negoci intervenen els Business Objects que faran ús d'objectes plans Java (VO, també coneguts com POJO) on s'encapsularan les dades del model i cridaran (els BO) a mètodes dels Data Access Objects per tal de recuperar dades de la capa de persistència, lògicament en aquesta capa s'encapsula tota la lògica de negoci.

Per últim, podem veure la capa de persistència implementada mitjançant Hibernate, la qual estableix connexió i accedeix a la base de dades mitjançant la factoria de sessió, en aquesta capa es mapejen els descriptors XML contra la base de dades, per posteriorment transformar les dades en objectes del model. Podem apreciar a nivell transversal, el servei de traces, que s'encarrega de registrar tots els events que es donen lloc al llarg de les diverses capes.

Un altre apreciació és que al llarg de totes les capes intervé Spring, gestionant des de les transaccions contra la BBDD, passant per la injecció dels DAO en

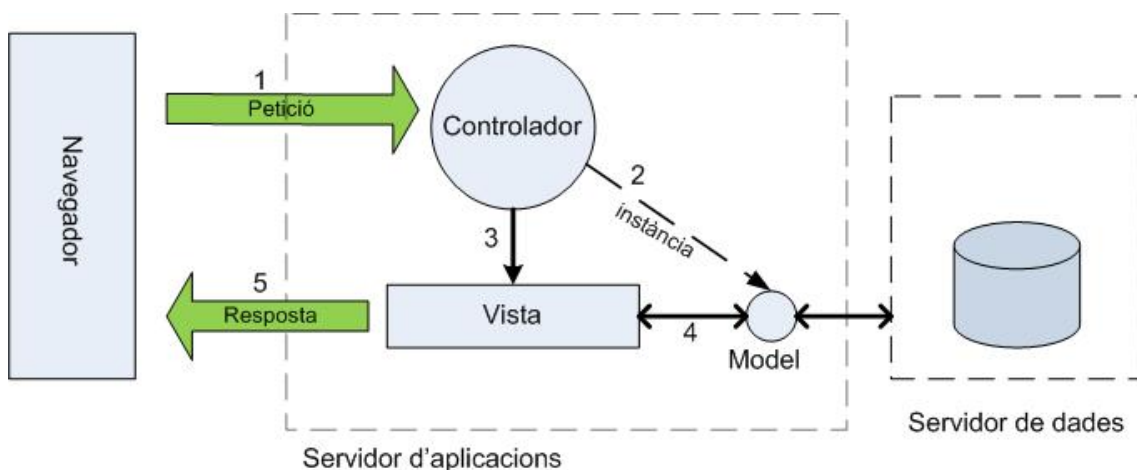
els BO, fins a gestionar l'obertura automàtica de connexions del pool a la BBDD en els action de Struts.

En la visió donada en aquest diagrama s'ha enfocat l'arquitectura des d'una altra perspectiva funcional que la perspectiva que ens dona el model MVC, i per això s'ubiquen els Web Action conjuntament amb les JSP, encara que són capes ben diferenciades en el model MVC.

3.1 MVC (Model Vista Controlador)

El model vista controlador és un bastió de disseny que separa les aplicacions en tres capes: la capa de presentació (Vistes), les classes del domini i accés a dades (Model), i les classes que regulen o controlen el flux de l'aplicació.

En aquest projecte el model MVC serà implementat mitjançant el bastió de Struts. Struts proporciona una implementació del model MVC basada en tecnologies estàndard com són Servlets Java, JavaBeans, fitxers de recursos i XML.



L'arquitectura MVC divideix els objectes implicats en una aplicació en tres tipus:

- Model, representa les dades de l'aplicació i les regles de negoci, està format per les classes del model o domini (Java Beans), per la capa d'accés a dades que serà implementada mitjançant DAOs (Data Access Objects) i per els objectes de negoci BOs (Business Objects).
- Vista, aquesta capa accedirà a les dades del model i s'encarregarà de mostrar per pantalla les dades de l'aplicació, així com, de interactuar amb l'usuari, redirigint les accions o events d'aquest cap el controlador.

Serà duta a terme mitjançant fulles JSP, i utilitzant l'especificació JSTL (JavaServer Pages Standard Tag Library), que proporciona unes

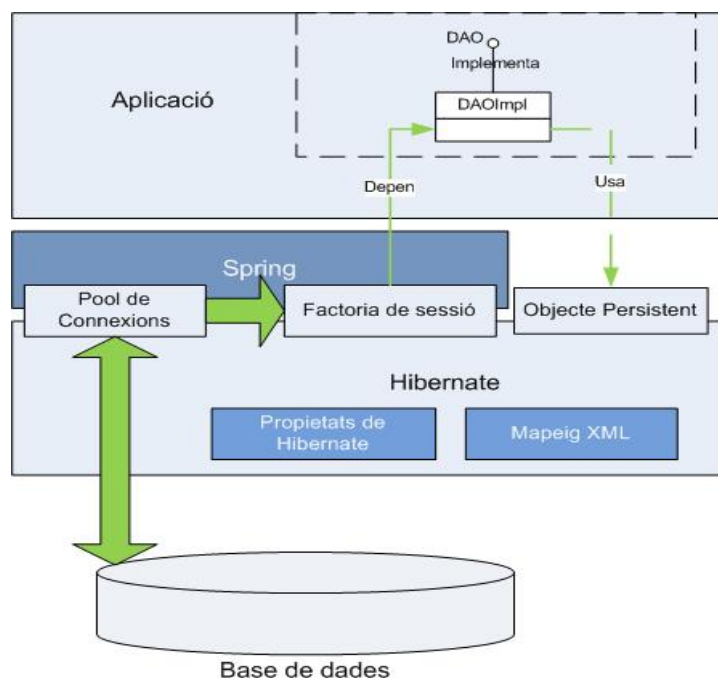
llibreries d'etiquetes que facilitaran la presentació de dades en pantalla, així com, permetran l'ús de lògica sense utilitzar codi java incrustat en aquestes.

- Controlador, aquesta capa definirà el comportament de l'aplicació, serà la que rebí les peticions o events de l'usuari, traduint-los en accions a realitzar sobre el model, una vegada processades les peticions, seran redirigides cap a les vistes.
El controlador serà implementat mitjançant objectes de la classe Action de Struts i els seus derivats. Per al control del flux s'utilitzarà el fitxer de configuració struts-config.xml, que definirà el mapa de rutes de l'aplicació.

3.2 Capa de persistència. Hibernate

La capa de persistència en el nivell més alt d'abstracció ha estat implementada per interfícies del tipus Data Access Object, interfícies que són a la vegada implementades per les seves classes de implementació que deriven de classes de Hibernate i Spring (es donaran més detalls en la secció relativa a la implementació), i que utilitzaran les classes del model per encapsular les dades. Aquestes classes (d'implementació de DAO) gestionaran les seves connexions, per mitjà de la factoria de sessió que a la vegada utilitzarà el pool de connexions de l'aplicació, tant la sessió com el pool seran creats i injectats en les classes abans mencionades mitjançant Spring.

Cal mencionar que la gestió de transaccions contra la base de dades serà gestionada per aquest últim, mitjançant l'ús de AspectJ (AOP), per a tal fi es definirà un aspecte que serà executat com a resposta a un event desencadenat per la sol·licitud d'una operació d'accés a la base de dades.



Per altra banda, al nivell més baix tindrem Hibernate, framework que mitjançant l'ús de descriptors XML s'encarregarà de transformar els registres de la base de dades en classes del model, així cada classe tindrà una correspondència directa amb una taula de la base de dades (OR mapping – Object Relational Mapping)

Per finalitzar aquesta secció, cal dir que la base de dades utilitzada com ja s'ha dit anteriorment ha estat Oracle 10g, i en la majoria de casos, quan s'ha requerit s'ha utilitzat el llenguatge HQL per fer consultes sobre la BBDD.

3.3 Inversió del Control. Spring.

Spring és un framework de programari lliure de tipus lleuger, lleuger en quant a grandària, ja que el framework sencer pot ésser distribuït en un únic jar de poc més de dos megas. Spring permet utilitzar objectes plans Java per aconseguir funcionalitats que abans només podien ésser aconseguides amb l'ús de EJBs, i dota a les aplicacions Java de simplicitat, testeabilitat i baix acoblament.

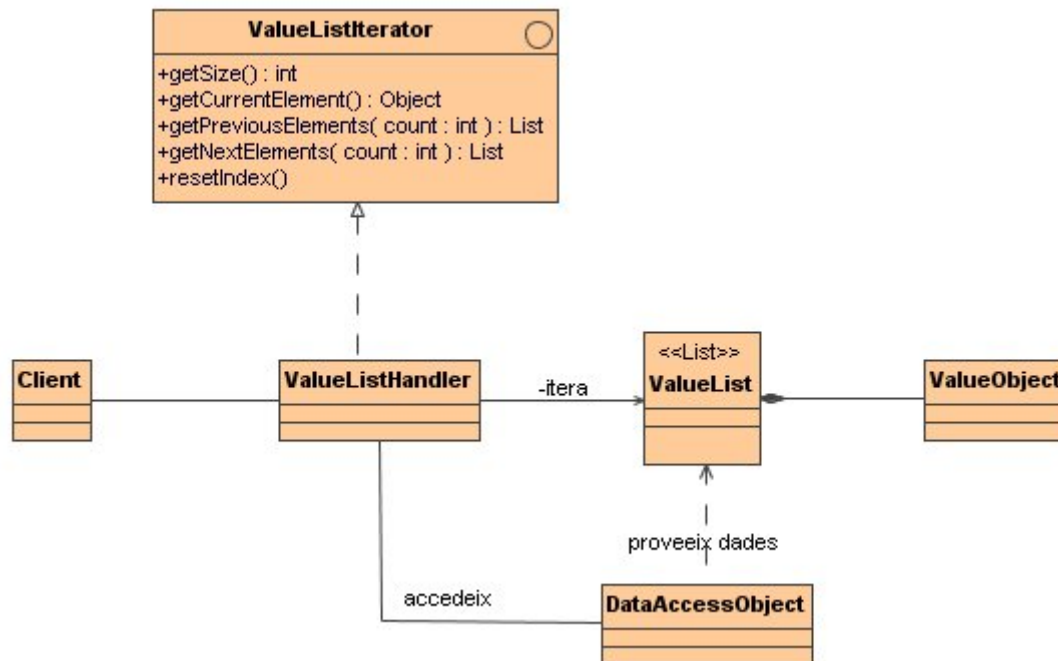
Cal comentar que a banda les funcionalitats ja esmentades en altres apartats de la memòria, s'ha fet ús de Spring amb la finalitat d'utilitzar la tècnica de IoC (Inversió de control), reduint d'aquesta forma l'acoblament entre classes considerablement. La tècnica de IoC es basa en la injecció de dependències en els objectes que és aplicada, és a dir, als objectes se'ls hi passen les seves dependències en lloc de crear-les i cercar-les. Podriem pensar en IoC com una mena JNDI però en el sentit invers, en comptes de que un objecte faci la cerca de les seves dependències en un contenidor, les seves dependències li són proporcionades a la instància de l'objecte per aquest contenidor, sense que l'objecte tingui que demanar-les.

3.4 El patró Value List Handler

Per tal de dur a terme la implementació de llistats s'ha utilitzat una implementació del patró de disseny J2EE anomenat Value List Handler definit per Sun Microsystems i implementat per MlavWILSON, una implementació open source del patró esmentat, amb llicència Apache License V2.0.

Aquest patró dona una solució al problema que tenen moltes aplicacions J2EE de realitzar cerques i llistar dades. En alguns casos aquestes cerques esdevenen quantitats enormes de dades, cosa que fa impossible retornar tot el conjunt de resultats de la consulta. Aquest patró permet recórrer els resultats de la consulta, mitjançant l'emmagatzematge d'aquests en memòria cau, proporcionant a més a més, funcionalitats de paginació i ordenació de dades.

A continuació es mostra el diagrama de classes del patró esmentat:



La interfície ValueListIterator proporciona mètodes per tal de recórrer els resultats d'una consulta, aquesta interfície es veu implementada per la classe ValueListHandler, la qual executarà la consulta sol·licitada per el Client i obtindrà els resultats d'aquesta, gestionant-los mitjançant una col·lecció d'objectes ValueObject, la col·lecció d'objectes esmentada és implementada mitjançant la classe ValueList. Per altra banda, la classe ValueListHandler farà ús de la classe DataAccessObject per tal de separar la implementació de l'accés a la base de dades.

Com ja s'ha dit anteriorment MlavWILSON implementa aquest patró de disseny, concretament utilitzant Spring, i a més a més, proporciona una llibreria d'etiquetes que fan més fàcil la seva integració en l'entorn web.

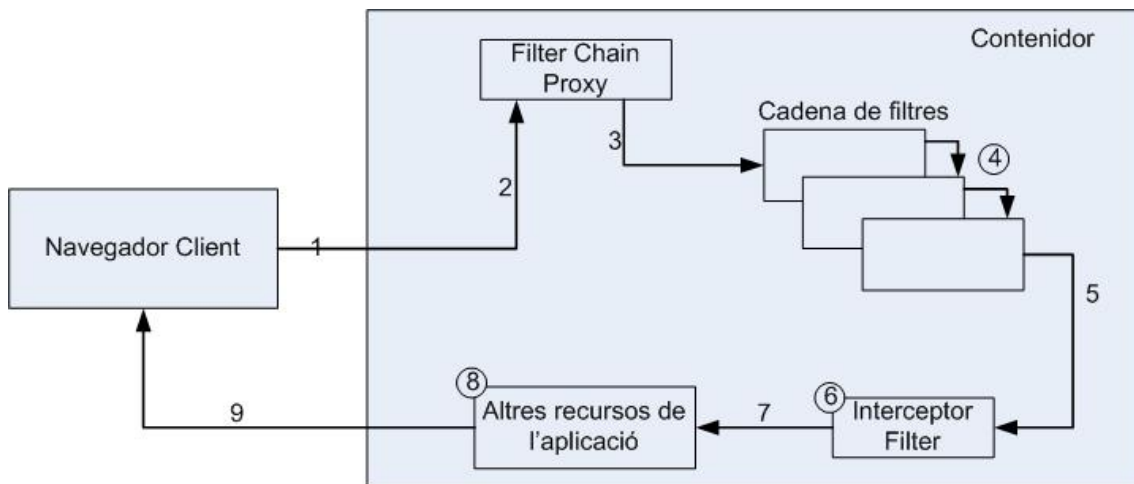
Per últim, cal dir, que com a llenguatge per les consultes s'ha utilitzat tant SQL com HQL, depenent dels requeriments d'aquestes.

3.5 Seguretat basada en filtres mitjançant Acegi

Per la implementació del servei de seguretat s'ha utilitzat Acegi Security, una solució open source, amb llicència Apache, orientada a aplicacions J2EE que fan ús de Spring, la qual proveeix a les aplicacions d'autenticació, autorització, control d'accés de instàncies, etc.

Malgrat totes les funcionalitats esmentades només s'ha utilitzat per implementar la funcionalitat d'autenticació de l'aplicació, concretament utilitzant diferents filtres que aquesta proporciona.

En la figura següent es pot veure com treballen els filtres davant d'una petició realitzada al contenidor que hostatja la cadena de filtres d'Acegi:



A continuació es descriuen els diferents passos en el cicle de vida en l'encadenament de filtres:

1. El client realitza una petició HTTP a través del navegador sol·licitant un recurs de l'aplicació.
2. El contenidor rep la petició HTTP, i s'encarrega de crear un objecte tipus *request* i un altre de tipus *response*, el primer contindrà dades de la petició realitzada i el segon servirà per preparar la resposta al client. Posteriorment es realitza una crida al *Filter Chain Proxy*, aquest *proxy* coneix els filtres els quals deuen ésser aplicats, ja que quan el contenidor invoca el *proxy*, aquest li passa la *request*, la *response* i la cadena de filtres.
3. El filtre de tipus *proxy* invoca el primer filtre en la cadena, passant-li la *request*, la *response* i la cadena de filtres.
4. Els filtres de la cadena són processats d'un amb un, quan un filtre finalitza el seu procés invoca al següent en la cadena. Pot passar que en un moment donat no calgui processar el següent filtre (per exemple quan un dels filtres detecta que per servir el recurs sol·licitat no es requereix autenticació).
5. Quan s'ha finalitzat el procés de tots els filtres, es passa el control al *Interceptor Filter* configurat en l'aplicació, conjuntament se li passen la *request* i la *response*.
6. És el *Interceptor* el que decideix si el client està autoritzat a accedir al recurs sol·licitat o no.
7. El *Interceptor* posteriorment transfereix el control a l'aplicació (per exemple, a la pàgina sol·licitada per el client en cas de correcta autenticació i autorització)
8. L'aplicació escriu la resposta sobre l'objecte *response*.
9. El client rep la resposta HTTP del contenidor.

Fins aquí arriba aquest apartat en l'arquitectura de l'aplicació, només comentar que es podria haver implementat l'autorització mitjançant els *ACL* que *Acegi* proporciona conjuntament amb *AOP*.

3.6 Altres aspectes de l'arquitectura

Tiles

Cal comentar l'ús d'un sistema de plantilles per tal de realitzar la capa de presentació, concretament fent ús de *Tiles (de Struts)*. Es bastant freqüent en l'entorn web que varies pàgines tinguin elements en comú, per aquest motiu i per tal de no repetir codi, s'ha utilitzat el sistema de plantilles esmentat. Una plantilla de Tiles no deixa d'ésser una *JSP*, en la qual mitjançant l'ús de llibreries d'etiquetes s'insereixen els diferents elements que la componen (per exemple, capçalera, menú, cos de la pàgina, peu de pàgina...), aquests elements són substituïts en temps d'execució per codi *JSP*. La correspondència entre els diferents elements que componen una pàgina i el codi *JSP*, s'especifica en un arxiu anomenat *tiles-defs.xml*.

Validacions

La majoria de les validacions de formularis s'han realitzat tant del costat client com del costat servidor, això proporciona un major control sobre les verificacions a realitzar per l'aplicació.

Per tal de realitzar les validacions esmentades s'ha fet ús del Validator de Struts (validacions de la banda client), incloent les validacions a realitzar en el fitxer *validation.xml* i utilitzant el fitxer de regles de validació *validator-rules.xml*, per posteriorment inserir-les en la pàgina *JSP* mitjançant una etiqueta especial. Per altra banda per fer les validacions de la banda servidor, s'han estès els formularis en que ha calgut realitzar validació, de la classe *ValidatorForm*, implementant el mètode *validate*, destinat per a tal fi.

Transaccions

Per finalitzar aquest apartat relatiu a l'arquitectura m'agradaria destacar l'ús de *AOP (AspectJ)* conjuntament amb *Spring* per tal de definir els serveis transaccionals en les operacions d'accés a la base de dades. Així doncs, s'ha definit un aspecte que és executat (mitjançant una classe *Proxy*) cada vegada que s'invoca un mètode pertanyent a un *BusinessObject*, els nivells transaccionals definits han estat per operacions de lectura, *read_only* i per a operacions més complexes amb escriptura de dades, *propagation_required*, el primer nivell permet a les operacions de lectura gaudir d'optimitzacions a nivell de rendiment, en quant al segon nivell defineix que el mètode en execució t'he que executar-se com si fos una transacció, de forma que si una transacció s'està executant en aquell moment s'haurà d'executar conjuntament amb aquesta. Aquests nivells transaccionals s'han definit de forma declarativa en un dels fitxers *XML* de definició de *beans* utilitzats per *Spring*, anomenat *datasource.xml*.

4 Implementació

4.1 Programari i maquinari

En quant al programari utilitzat per la realització del projecte, cal comentar en primer lloc que per la planificació s'ha utilitzat Microsoft Project, en segon lloc com a eina per al disseny Magic Draw UML, i com a entorn de desenvolupament s'ha utilitzat Eclipse 3.2, concretament la distribució Callisto conjuntament amb els plugins de Exadel i de Spring IDE.

Per altra banda, per lo que respecta al servidor d'aplicacions s'ha utilitzat la versió 5.5 del servidor Tomcat, corrent amb la màquina virtual de Java 5.0, cal comentar que aquest es de tipus Open Source, i es pot utilitzar sota llicència Apache License v2.0. Les decisions principals d'utilitzar aquest servidor han estat per una banda, que es un producte força estable i sobradament testejat, i per altra banda, que és un servidor d'aplicacions molt utilitzat en el mon professional.

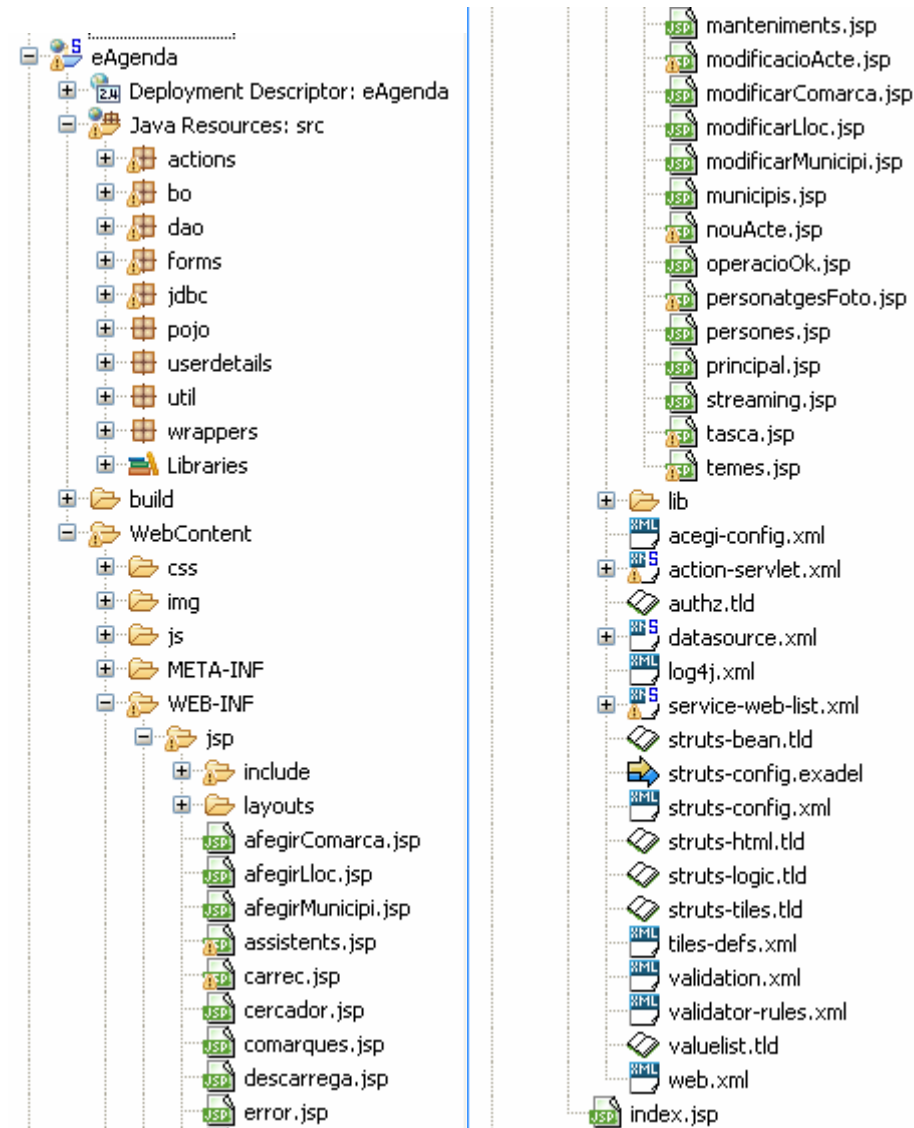
En lo que respecta a base de dades, l'aplicació fa ús d'una de les últimes versions de Oracle, aquesta és Oracle 10g, s'ha pres la decisió d'adoptar aquesta base de dades, donat que el projecte parteix d'un cas real, en el que la base de dades utilitzada per el client era la ja esmentada. Cal dir, que la llicència d'aquesta és lliure per al desenvolupament, però cal adquirir o tenir una llicència específica (de caràcter propietari) en el cas d'utilitzar-la en un entorn de producció.

L'aplicació ha estat optimitzada i provada en els navegadors, Internet Explorer 5.0 i 6.0, així com, en Mozilla Firefox 2.0 amb una resolució de 1024x768 píxels.

Per lo que respecta a maquinari, per posar l'aplicació en funcionament s'ha utilitzat una màquina tipus Pentium a 3.5 GHz, i amb 2 Gb de RAM, no obstant es podria utilitzar a mode de proves un equip de inferiors prestacions (1.5 – 2 GHz) amb 512 Mb de RAM com a mínim, encara que no es garanteix un rendiment òptim.

4.2 Distribució i estructura de paquets de l'aplicació

A continuació es presenta l'estructura de paquets i directoris de l'aplicació:



En quant a l'estructura de paquets en que s'han agrupat les classes i la seva finalitat, tenim:

- ✓ **actions**: agrupa tots els controladors que intervenen en l'aplicació. Totes les classes pertanyents a aquest paquet hereten de la classe `org.springframework.web.struts.DispatchActionSupport`, això és així, per una banda per poder dotar a la classe del context de Spring, i per un altra per poder gaudir de la funcionalitat que aporten els `DispatchAction` de Struts.
- ✓ **bo**: agrupa les interfícies dels Business Objects (BO) i les classes que les implementen, la finalitat de les quals com el nom indica (BO) és

realitzar la implementació dels mètodes de negoci de l'aplicació. Tenen múltiples dependències amb els DAO, ja que els utilitzen.

- ✓ dao: agrupa les interfícies dels Data Access Objects (DAO) i les classes que les implementen, la seva finalitat es encarregar-se de l'accés a la base de dades. Aquestes hereten de la classe `org.springframework.orm.hibernate3.support.HibernateDaoSupport`, "plantilla" (template) de Hibernate que els hi permetrà heretar serveis, així com, permetrà la injecció de la factoria de sessió en aquestes.
- ✓ forms: agrupa tots els formularis de Struts, la finalitat dels quals és l'intercanvi de dades entre la capa de presentació i els controladors. En trobem dos tipus d'herència entre aquestes classes, de `org.apache.struts.action.ActionForm` i de `org.apache.struts.validator.ValidatorForm`, el primer cas representa un formulari de Struts "senzill" i el segon inclou validacions de la banda servidor.
- ✓ jdbc: alberga la classe `JdbcDaoImpl`, classe utilitzada com a `UserDetailsService` per el `DaoAuthenticatorProvider` i a la vegada per el `AuthenticationManager` que és utilitzat per el `FilterInvocationInterceptor`, hereta de `org.acegisecurity.userdetails.jdbc.JdbcDaoImpl` i sobreescriu els seus mètodes amb la finalitat de personalitzar la rutina d'autenticació d'Acegi (recupera també les credencials de la consulta d'autenticació).
- ✓ pojo: (Plain Old Java Object) agrupa les classes del domini, com el seu nom indica són objectes plans, la finalitat dels quals és l'encapsulament de dades del model. Per altra banda, en el mateix directori del paquet es troben els descriptors o mappings XML de Hibernate utilitzats per l'ORM (Object-Relational Mapping), la majoria d'aquests fitxers s'han generat automàticament, encara que en molts casos s'han hagut de modificar per assolir el comportament desitjat.
- ✓ userdetails: alberga la classe `User`, objecte pla que encapsula les dades de l'usuari i hereta de `org.acegisecurity.userdetails.User`, és utilitzat per el procés d'autenticació d'Acegi.
- ✓ util: agrupa dos classes, `ActeProxy` i `Option`, utilitzades amb el propòsit de servir d'utilitat per emmagatzemar dades d'un acte i de les opcions d'un control selecció respectivament.
- ✓ wrappers: alberga la classe `ActeWrapper` que implementa la interfície `net.mlw.vlh.adapter.util.ObjectWrapper`, utilitzada per recuperar les dades d'un acte d'una consulta SQL realitzada per un objecte `ValueList` (recordatori: el patró `ValueListHandler`)

Fins a aquest punt s'ha tractat l'estructura relativa als paquets que componen l'aplicació, per continuar, es descriurà la estructura de directoris de recursos tals com fitxers de configuració, JSPs, etc.

A partir de l'arrel de l'aplicació podem trobar els següents directoris i fitxers:

- ✓ index.jsp: és la pàgina JSP d'entrada a l'aplicació
- ✓ css: la carpeta css conté com el seu nom indica totes les fulles d'estils css que utilitza l'aplicació.
- ✓ img: conté totes les imatges utilitzades per l'aplicació.
- ✓ js: conté els fitxers javascript utilitzats per l'aplicació.
- ✓ WEB-INF: conté diverses carpetes, així com, fitxers xml de configuració i fitxers de definició de llibreries d'etiquetes.

Dintre del directori WEB-INF es poden trobar els següents directoris:

- ✓ jsp: conté els directoris /include i /layouts, així com, pràcticament totes les jsp utilitzades en l'aplicació.
- ✓ jsp/include: conté els diversos components d'una pàgina jsp, com poden ésser la capçalera, els menús, el peu de pàgina, etc.
- ✓ jsp/layouts: conté dos plantilles que seran utilitzades conjuntament amb Tiles i les jsp esmentades anteriorment, per formar les diverses pantalles de l'aplicació.
- ✓ lib: conté les llibreries (jar) utilitzades per les classes de l'aplicació.
- ✓ classes: a banda dels paquets utilitzats en l'aplicació, conté els fitxers de propietats utilitzats per treure tant els literals de l'aplicació, com els missatges, títols, etc.

4.3 Integració de frameworks

En aquests tres o quatre apartats he intentat definir allò que he considerat prioritari, no obstant, hi ha nombrosos temes per parlar, molt interessants i que no s'han tractat, des de la integració d'Acegi, passant per la implementació i us de Tiles, la injecció de propietats o atributs en les classes mitjançant Spring, la implementació del servei de traces utilitzant log4j, etc. Potser alguns d'aquests temes es podrien haver inclòs en la memòria, no obstant, no he fet per motius d'espai, ja que de haver-ho fet hagués resultat un document massa estens, és una llàstima!

Per altra banda, dir que aquesta secció podria haver estat incorporada tant en la implementació, com potser en l'arquitectura, finalment s'ha decidit fer-ho en la implementació doncs es tracten aspectes potser més propers a aquesta.

4.3.1 Struts

Per tal de integrar Struts en l'aplicació, a banda de incloure les llibreries d'aquest framework dintre de l'aplicació (en el directori WEB-INF/lib), ha estat necessari modificar el fitxer web.xml ubicat dintre de la carpeta WEB-INF. Amb les línies següents s'especifica la classe o servlet encarregat de rebre les peticions, el fitxer de configuració de Struts, i l'ordre de carrega d'aquest en carregar l'aplicació:

```
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>
    org.apache.struts.action.ActionServlet
  </servlet-class>

  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Com a segon pas, cal especificar el patró d'adreça url mitjançant la qual es realitzaran les peticions als actions de Struts, les quals seran ateses per el servlet definit amb anterioritat:

```
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

Una vegada fet això només caldrà incloure en el fitxer struts-config.xml: els formularis utilitzats en l'aplicació, els controladors Action, la definició del *controller* (s'explica en el següent apartat), i la definició dels plugins utilitzats, en aquest cas el plugin de Tiles i el del Validator (Validador) de Struts:

```
<plug-in className="org.apache.struts.tiles.TilesPlugin">
  <set-property
    property="definitions-config"
    value="/WEB-INF/tiles-defs.xml" />
  <set-property property="moduleAware" value="true" />
</plug-in>

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property
    property="pathnames"
    value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
</plug-in>
```

4.3.2 Spring

Al igual que amb el framework anterior per integrar Spring en l'aplicació ha fet falta per una banda incloure les llibreries de Spring en el directori WEB-INF/lib, per posteriorment modificar el fitxer web.xml:

1. Es defineix la ubicació dels fitxers utilitzats per Spring.

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/action-servlet.xml,/WEB-INF/acegi-config.xml
  </param-value>
</context-param>
```

2. Posteriorment s'ha definit el filtre interceptor de peticions.

```
<listener>
  <listener-class>
    org.springframework.web.context.ContextLoaderListener
  </listener-class>
</listener>
```

Una vegada fet això, caldrà incloure en el fitxer struts-config.xml la definició del controlador de Tiles:

```
<controller
processorClass="org.springframework.web.struts.DelegatingTilesRequestProcessor" />
```

Per posteriorment incloure el plugin especificant la ubicació del fitxer de context:

```
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
  <set-property property="contextConfigLocation"
value="/WEB-INF/action-servlet.xml" />
</plug-in>
```

Amb això acaba la integració de Spring, només cal dir que, s'han utilitzat els fitxers action-servlet.xml, datasource.xml i acegi-config.xml, per incloure els beans utilitzats per Spring amb diverses finalitats.

4.3.3 Persistència. Hibernate.

Per tal de integrar Hibernate amb Struts cal incloure les llibreries de Hibernate en el directori WEB-INF/lib, una vegada fet això el següent pas ha estat incloure en el fitxer datasource.xml utilitzat per Spring, la definició del datasource, la de la factoria de sessió i especificar el dialecte HQL a utilitzar:

```
<!-- DataSource de l'aplicació -->
<bean class="org.apache.commons.dbcp.BasicDataSource"
  destroy-method="close" id="myDataSource">
  <property name="driverClassName"
value="oracle.jdbc.driver.OracleDriver"/>
  <property name="url" value="jdbc:oracle:thin:@netmaster:1521/orcl"/>
  <property name="username" value="AGENDA_ADM"/>
  <property name="password" value="*****"/>
</bean>
<!-- Factoria per la sessió de Hibernate -->
<bean
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean"
destroy-method="destroy" id="sessionFactory">
  <property name="dataSource" ref="myDataSource"/>
  <property name="mappingResources">
  <list>
    .....
    <!-- Fitxers ORM -->
    .....
  </list>
</property>
  <property name="hibernateProperties">
  <value>
    hibernate.dialect=org.hibernate.dialect.Oracle9Dialect
  </value>
</property>
</bean>
```

Fet això, ja tenim Hibernate integrat amb Spring.

Cal destacar també la gestió de les connexions, s'ha definit un filtre en el fitxer web.xml, per tal de que cada vegada que es fa una petició a un controlador s'obri la connexió de Hibernate i es tanqui automàticament, això s'ha aconseguit inclouent les següents línies:

```
<filter>
  <filter-name>openSessionInViewFilter</filter-name>
  <filter-class>
    org.springframework.orm.hibernate3.support.OpenSessionInViewFilter
  </filter-class>
</filter>
```

Per últim i per finalitzar aquesta secció cal destacar la gestió de transaccions, ja esmentada en l'apartat d'arquitectura, s'ha implementat seguint els següents passos:

1. Definir les notificacions o events que es produiran cada vegada que s'invoqui un mètode que encaixi amb els patrons definits:

```
<tx:advice id="txAdvice" transaction-manager="txManager">
  <tx:attributes>
    <tx:method name="get*" read-only="true"/>
    <tx:method name="load*" read-only="true"/>
    <tx:method name="donam*" read-only="true"/>
    <tx:method name="carregar*" read-only="true"/>
    <tx:method name="preparar*" read-only="true"/>
    <tx:method name="*" />
  </tx:attributes>
</tx:advice>
```

2. Definir un aspecte que s'executi cada vegada que s'executi un mètode pertanyent a un Business Object (BO), d'aquesta forma es produirà una transacció cada vegada que es realitzi una crida a un mètode del BO:

```
<aop:config>
  <aop:pointcut id="boService" expression="execution(* bo.*(..))"/>
  <aop:advisor advice-ref="txAdvice" pointcut-ref="boService"/>
</aop:config>
```

3. Per finalitzar, definir el gestor de transaccions:

```
<bean id="txManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="myDataSource"/>
</bean>
```

5 Proves funcionals

Per tal de verificar el correcte funcionament de l'aplicació s'han realitzat tota una sèrie de proves de caràcter funcional, els resultats d'aquestes han estat força satisfactoris, lo que assegura que s'ha obtingut un producte final el qual compleix les especificacions i requisits marcats en les etapes d'anàlisi i disseny. Els resultats de les proves es presenten a continuació.

Funcionalitat provada	Nº de proves realitzades	Puntuació
Autenticació de l'aplicació	3	10
Cerca d'actes	19	9
Llistat d'actes	9	10
Afegir/eliminar intervinents	11	10
Afegir/eliminar assistents	11	10
Eliminació d'actes	2	10
Alta/modificació d'actes	8	10
Consulta de fotografies vinculades	1	10
Afegir fotografia	3	6
Eliminar fotografies	3	10
Descarrega de fotografies	2	10
Alta de comarques	2	10
Llistat de comarques	9	10
Eliminació de comarques	1	10
Modificació d'una comarca	2	10
Alta de municipis	2	10
Llistat de municipis	9	10
Eliminació de municipis	1	10
Modificació d'un municipi	2	10
Alta de llocs	2	10
Llistat de llocs	9	10
Eliminació de llocs	1	10
Modificació d'un lloc	2	10
Resultat final	114	9,9

Cal comentar que el número de proves realitzades no ha estat aleatori sinó que depèn totalment de les diverses casuístiques de cada unitat funcional, per altra banda, s'han realitzat més proves que les reflectides en aquesta taula i els resultats han estat al igual que anteriorment força satisfactoris.

Resumint, es pot afirmar que s'ha obtingut un producte de qualitat.

6 Conclusions

Aquest projecte ha estat el resultat de mesos de treball i esforç intens, en aquest s'han posat en pràctica coneixements adquirits en altres assignatures al llarg de la carrera, tals com Enginyeria del Programari, Bases de Dades I i II, o Tècniques de desenvolupament de Programari.

Malgrat haver aconseguit desenvolupar un prototip el qual no contempla totes les especificacions inicials, si puc ben dir que s'han aconseguit acomplir tots els objectius proposats, per una banda s'ha aconseguit realitzar una aplicació J2EE robusta, escalable i de qualitat, i per un altre s'ha aconseguit tal i com m'havia proposat, acomplir tots els objectius específics marcats al inici del projecte, adquirir major experiència en l'ús de eines desenvolupament de programari, aprofundir en l'àmbit de la usabilitat, així com, en el desenvolupament de programari i concretament en l'àmbit o ús de tecnologies J2EE.

En quant a l'ús de tecnologies J2EE, puc dir que malgrat tenir ja experiència en l'àmbit acadèmic i professional, aquest projecte plantejava un repte que era tant la utilització de tecnologies més o menys novetoses per una banda, com la integració i utilització d'aquestes en el seu conjunt en un únic projecte, al final he aconseguit superar aquest repte, treballant , aprofundint i adquirint nous coneixements durant l'ús d'aquestes.

Per lo que respecta a una línia de treball futura, caldria esmentar que és podria completar l'aplicació implementant totes les funcionalitats inicialment proposades per una banda, així com, implementant tot el sistema de permisos i control d'autenticació, i afegint capacitats de gestió documental a la gestió d'actes.

6 Glossari

Acte: fet o esdeveniment de caràcter públic, per exemple, potser un congrés o un seminari, etc.

Acte protocol·lari: event o esdeveniment de caràcter diplomàtic i oficial.

Assistent: persona que assisteix a un acte.

Càrrec: funció jeràrquica d'un assistent a un acte.

Esdeveniment: fet extraordinari, pot qualificar-se com un acte.

Expedient: documentació associada o que genera un acte.

Intervinent: persona que desenvolupa una tasca durant un acte.

Tasca: treball a realitzar durant la celebració d'un acte.

Unitat: àrea d'un departament destinada a realitzar una funció, i per tant, especialitzada en un àmbit.

Unitat de fotografia: àrea d'un departament destinada a la realització i gestió de fotografies

Unitat de protocol: àrea d'un departament dedicada a l'organització i gestió d'esdeveniments de caràcter protocol·lari

7 Bibliografia

Llibres

Professional Jakarta Struts

James Goodwill, Richard Hightower, Editorial: WROX (www.wrox.com)

Professional Hibernate

Eric Pugh, Josep D. Gradecki, Editorial: WROX

Spring in action

Craig Walls, Ryan Breindenbach,

Editorial: Manning (<http://www.manning.com/walls2/>)

Recursos a Internet

<http://java.sun.com>

Pàgina oficial de SUN Microsystems relativa al llenguatge Java.

<http://struts.apache.org/1.3.8/index.html>

Pàgina web del framework de Struts.

<http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>

Article de la revista Java World sobre el model MVC.

http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html

Pàgina de SUN Microsystems relativa al model MVC

<http://hibernate.org/5.html>

Pàgina web del framework de Hibernate, accés a la documentació.

<http://static.springframework.org/spring/docs/2.0.x/reference/index.html>

Pàgina web de la documentació relativa al framework de Spring.

<http://valuelist.sourceforge.net/>

Pàgina web del patró Value List Handler de MlavWILSON.

<http://java.sun.com/blueprints/corej2eepatterns/Patterns/ValueListHandler.html>

Pàgina web de SUN Microsystems sobre el patró Value List Handler.

<http://apache.org/>

Pàgina web de la Apache Software Foundation.

<http://acegisecurity.org/>

Pàgina web relativa al framework de seguretat d'Acegi.