

Gesevent. Gestió d'events.

Desenvolupament d'aplicacions amb tecnologies J2EE.

Floren Llanos Ruiz

ETIG

Albert Grau Perisé

20 de juny de 2011

Dedicatòria

Als meus pares pel suport que sempre m'han donat.

A Susanna.

1. Resum del treball de final de carrera

Java és en aquests moments un referent en el món del desenvolupament d'aplicacions, en particular la seva plataforma dedicada a aplicacions empresarials, més coneguda com a *Java Enterprise Edition (J2EE)*¹. La qual per la seva modularitat i robustesa ha anat guanyant importància en el món de la informàtica.

El projecte que aquí presentem pretén ser una aproximació real a aquest complex món del desenvolupament d'aplicacions amb J2EE, amb la utilització de diferents patrons i tecnologies; algunes de les quals eren completament desconegudes per la nostra part.

Per tal de realitzar aquest procés d'auto aprenentatge en aquesta plataforma ens hem fet servir d'un problema real com és la gestió d'events de qualsevol empresa, de manera que per arribar a la resolució d'aquesta necessitat hem fet servir de una tecnologia orientada a objectes (Java), basada en components (Enterprise Java Beans, EJB²) i amb un client lleuger i flexible com són les pàgines JSP³ i Servlets⁴ (en el nostre cas utilitzant un bastiment molt popular anomenat Struts⁵).

En aquest treball exposarem el problema amb detall, l'aproximació funcional utilitzada per la seva resolució així com la gran varietat de tecnologies emprades per resoldre teològicament la necessitat exposada, explicant per cadascuna d'aquestes els patrons en la que es basen i el seu funcionament.

¹ <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

² <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

³ <http://download.oracle.com/javaee/5/tutorial/doc/bnagx.html>

⁴ <http://www.oracle.com/technetwork/java/javaee/servlet/index.html>

⁵ <http://struts.apache.org/download.cgi#struts2211>

Índex de continguts

2. Introducció	7
2.1. Justificació del TFC: punt de partida i aportació.	7
2.2. Objectius del TFC	8
2.3. Enfocament i mètode seguit	9
2.4. Planificació del projecte	10
2.5. Productes obtinguts	13
2.6. Contingut de la memòria.....	14
3. Especificació funcionalitats i anàlisi de requeriments	15
3.1. Introducció	15
3.2. Descripció del projecte	15
3.3. Rols de l'aplicació i casos d'ús per rol.....	16
3.4. Subsistemes i funcionalitats per subsistema.....	18
4. Diagrames dinàmics de disseny	21
4.1. Introducció.	21
4.2. Casos d'ús per subsistema.....	21
4.3. Diagrames de seqüència. Especificació dels casos d'ús.	25
4.4. Diagrames d'estat.....	32
5. Diagrames estàtics de disseny.....	34
5.1. Distribució de l'aplicació i diagrama de paquets.....	34
5.2. Diagrama de classes.	35
5.3. Model entitat-relació.	37
6. Arquitectura de l'aplicació	41
6.1. Introducció.	41
6.2. Servidor d'aplicacions.....	41
6.3. Model de capes MVC.	43
6.4. Patró objectes façana (Session Facade).....	45
6.5. Struts2.....	46
6.6. Tiles.	47
6.7. EJB (versió 3).	48
7. Valoració econòmica.	50
8. Conclusions.	51
9. Glossari.	52
10. Bibliografia.....	53
11. Annexos.....	55

Índex de figures

Fig 1 Llistat de tasques	10
Fig 2 Diagrama de Gantt	12
Fig 3 Cas d'ús subsistema de manteniment (events i usuaris).....	22
Fig 4 Cas d'ús subsistema de manteniment (tipologies).....	22
Fig 5 Cas d'ús subsistema d'autenticació	23
Fig 6 Cas d'ús subsistema de registre	24
Fig 7 Cas d'ús de comunicació. Taulell de missatges	25
Fig 8 Cas d'ús de informació. Extracció de llistat d'usuaris inscrits	25
Fig 9 Diag. seq. alta d'usuari	26
Fig 10 Diag. seq. Modificació d'usuari.....	27
Fig 11 Dia.seq. Baixa d'usuari.....	27
Fig 12 Diag.seq. Gestió d'autenticació	28
Fig 13 Diag.seq. Inscripció a event	28
Fig 14 Diag.seq. Baixa de inscripció a event	29
Fig 15 Diag.seq. Creació informació a event	30
Fig 16 Diag.seq. Consulta de missatges de un taulell	30
Fig 17 Diag.seq. Enviament de missatges	31
Fig 18 Diag.seq. Extracció de llistats de usuaris inscrits	31
Fig 19 Estat. Estat registre usuari extern	32
Fig 20 Estat. Estat creació event.....	33
Fig 21 Diagrama de paquets	35
Fig 22 Diagrama de classes	36
Fig 23 Model entitat-relació	37
Fig 24 Esquema model-vista-controlador	43
Fig 25 MVC, distribuït per tecnologies i contenidors	45
Fig 26 Patró façana (Session facade)	46
Fig 27 Struts2 amb la crida a EJB Façana.....	47
Fig 28 Crida a un objecte remot EJB	49
Fig 29 Pantalla de autenticació	55
Fig 30 Pantalla de registre d'usuari.....	55
Fig 31 Menú del rol Administrador i pantalla de manteniment de persones	56
Fig 32 Menú del rol Gestor i pantalla de gestió d'events	56
Fig 33 Pantalla de extracció de llistats d'assistents per el rol Gestor	57
Fig 34 Menú del rol Usuari i pantalla de inscripció a events	57
Fig 35 Pantalla de seguiment d'events pel rol de Usuari	58

2. Introducció

2.1. Justificació del TFC: punt de partida i aportació.

Aquest treball bé justificat per l'interès personal en aprendre noves tecnologies relacionades amb la plataforma J2EE, en particular pretén ser un punt de entrada a la tecnologia de component de negoci Enterprise Java Beans (EJB)⁶ els quals eren completament desconeguts abans de la realització d'aquest projecte. A més és vol profunditzar en noves versions del popular bastiment de desenvolupament Struts el qual s'ha utilitzat en la seva versió 2.2.

Junt amb l'interès de profunditzar en aquestes tecnologies rau la intenció de estudiar els patrons de disseny en les que estan basades i faciliten als programadors la del manteniment i desenvolupament. Els patrons estudiats per treure'n màxim profit de la plataforma són: model-vista-controlador (MVC⁷) i el Session Facade⁸ utilitzat pels components de negoci EJB.

Un altre decisió que s'ha pres a l'hora de la realització del treball és fer el desenvolupament de l'aplicació utilitzant aplicacions gratuïtes de les que podem esmentar breument: el sistema operatiu Linux⁹, l'entorn integrat de desenvolupament eclipse¹⁰ i moltes aplicacions més que esmentarem en un capítol a part ja que considerem que han tingut molta rellevància a l'hora de portar a bon terme el treball realitzat.

Per un altre banda esmentar que la nostra intenció inicial era també realitzar tota la documentació amb tecnologies sense llicència de pagament. Per això és va escollir en una primera opció Openoffice el qual al donar-nos alguns problemes durant la redacció d'aquesta memòria i davant del poc temps disponible varem optar per substituir-lo per l'entorn d'ofimàtica de Microsoft el qual vist els resultats i la facilitat d'ús el considerem més madur en aquest sentit.

Per tal de poder portar a terme en pràctica les tecnologies descrites hem escollit el problema que pot esdevindre per a qualsevol organització que tingui la necessitat de gestionar diferents events que pot oferir tant als seus empleats com a clients. Es va considerar aquesta necessitat com la ideal ja que ens permetia explotar total el potencial de una tecnologia orientada a objectes com és Java que permet un alt grau de re utilització així com la plataforma J2EE en particular que permet uns alts graus de escalabilitat.

El punt de partida del projecte han sigut certs coneixements del llenguatge Java a més dels coneixements teòrics i pràctics adquirits durant els anys d'estudi a la Universitat Oberta de Catalunya (UOC). De tots els coneixements adquirits volem destacar els especialment útils

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

7 <http://java.sun.com/blueprints/patterns/MVC-detailed.html>

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

9 <http://es.wikipedia.org/wiki/GNU/Linux>

10 <http://www.eclipse.org/>

per aquest projecte com han sigut els d'enginyeria del programari, els adquirits en les assignatures de base de dades i els de desenvolupament; sense oblidar-nos de les capacitats adquirides per la cerca de informació i aplicabilitat de la mateixa a un problema real.

L'aportació que es pretén a l'hora de la realització d'aquest treball és que a partir de la tecnologia descrita, del problema i dels nostres coneixements previs obtenir un producte que no només serveixi com un demostrador de la tecnologia emprada si no com una aplicació que sigui cent per cent funcional.

2.2. Objectius del TFC

Els objectius d'aquest treball de final de carrera els podem dividir en dos grups: els generals que són els referits a l'aplicabilitat de la solució al problema proposat i els específics que es componen principalment pels reptes afrontats pel fet de fer ús de tecnologies desconegudes fins a les hores gràcies als coneixements adquirits durant els anys d'estudi a la Universitat Oberta de Catalunya.

Objectius generals

Els objectius generals del projecte han sigut:

- Obtenció de una eina 100% funcional per satisfer les necessitats d'una possible empresa o grup d'empreses referents a la gestió dels seus events.
- Millorar la interacció entre els possibles usuari de l'eina i l'empresa responsable de la seva explotació.
- Implantar un sistema de gestió d'events de forma eficaç, eficient i escalable.

Objectius específics

Els objectius específics del projecte són:

- L'ús dels coneixements adquirits durant l'estada a la UOC per desenvolupar de forma autònoma una aplicació de gestió que sigui capaç de resoldre un problema real.
- Aprenentatge i ús de tecnologia distribuïda en J2EE (Java Enterprise Edition), especialment EJB (Enterprise Java Beans) per augmentar la seguretat i escalabilitat de l'aplicació a desenvolupar.
- Aplicabilitat del conjunt de coneixements esmentats, tecnologies i recursos tècnics per tal de obtenir un producte amb valor de per si, tant funcional com econòmic.

un problema concret i que a la vegada gràcies a l'ús dels recursos i tècniques comentades pugui créixer segons les noves necessitats empresarials que puguin sorgir.

2.3. Enfocament i mètode seguit

La idea de partida ha sigut la resolució d'una necessitat empresarial a la hora de gestionar els events que disposa pels seus empleats, pels seus clients i el públic en general. A partir d'aquí s'ha estudiat com resoldre aquest problema de manera que l'aplicació obtinguda durant el desenvolupament del treball sigui una primera aproximació de resolució del problema establert a més que pugui ser estesa posteriorment.

L'enfocament al projecte s'ha descompost en quatre fases, tres de les quals han sigut independents amb els temps i la quarta s'ha anat produint al mateix que les tres primeres fases.

Les fases esmentades són:

- Anàlisi de requisits funcionals de l'aplicació. En aquest punt s'ha definit un marc genèric d'actuació de l'aplicació que es volia desenvolupar. En aquest marc s'ha definit el problema a resoldre i la millor aproximació possible contant ja el temps disponible per a la realització del projecte.
- Disseny de l'aplicació. Durant la fase de disseny s'ha tingut en compte la tecnologia distribuïda que es volia aprofitar per la realització del producte així com els patrons necessaris per portar amb èxit aquesta tasca.
- Desenvolupament de l'aplicació. L'etapa de desenvolupament s'ha centrat en l'aplicació de la tecnologia escollida per resoldre el problema proposat i obtenir un producte que satisfaci els requisits funcionals proposats en la primera fase.

Com s'ha dit aquestes tres primeres fases han tingut un desenvolupament individual entre elles, tot i així el procés seguit ha sigut iteratiu, és a dir, s'han anat repassant tant els requisits demanats com el disseny de l'aplicació, durant el desenvolupament, per adequar-los els errors i/o omissions trobats.

Junt amb aquestes tres fases s'ha treballat una quarta per tal d'anar fent proves tècniques amb les aplicacions i eines escollides per produir el producte desitjat. Fer constar que ha sigut una tasca dura i costosa a nivell temporal degut al desconeixement de la majoria de tecnologies emprades.

2.4. Planificació del projecte

La planificació inicial pensada va anar variant a mesura que avançava el projecte degut a canvis imprevistos que van fer desviar-se la planificació. A més de errors i millores del disseny detectats durant la implementació van fer que aquesta part patís una desviació i necessités un esforç extra per acomplir els objectius temporals proposats.

Les tasques en les que es va descomposar el projecte les podem veure en la imatge següent:

WBS	Nombre	Inicio	Fin	Trabajo	Duración
1	▼ TFC	mar 12	jun 19	129d	100d
1.1	Estudi i proves unitaries J2EE	mar 12	abr 8	28d	28d
1.2	Pla de treball	mar 12	mar 19	8d	8d
1.3	▼ Anàlisi de requisits i UML	mar 20	abr 7	19d	19d
1.3.1	Especificació de funcionalitats	mar 20	mar 22	3d	3d
1.3.2	Actors i responsabilitats	mar 23	mar 23	1d	1d
1.3.3	Casos d'ús	mar 24	mar 28	5d	5d
1.3.4	Diagrames de classes	mar 29	abr 2	5d	5d
1.3.5	Diagrames de seqüències	abr 3	abr 5	3d	3d
1.3.6	Diagrama E-R	abr 6	abr 7	2d	2d
1.4	▼ Implementació	abr 8	may 16	40d	39d
1.4.1	Creació de BBDD	abr 8	abr 8	1d	1d
1.4.2	▼ Subsistema de manteniment	abr 8	abr 25	18d	18d
1.4.2.1	Alta d'events	abr 8	abr 11	4d	4d
1.4.2.2	Baixa d'events	abr 12	abr 12	1d	1d
1.4.2.3	Modificació d'events	abr 13	abr 15	3d	3d
1.4.2.4	Alta usuaris (administradors, gestors, assistents)	abr 16	abr 20	5d	5d
1.4.2.5	Baixa usuaris (administradors, gestors, assistents)	abr 21	abr 23	3d	3d
1.4.2.6	Modificació usuaris (administradors, gestors, assistents)	abr 24	abr 25	2d	2d
1.4.3	▼ Subsistema de autenticació	abr 26	may 3	8d	8d
1.4.3.1	Gestor de autenticació	abr 26	abr 29	4d	4d
1.4.3.2	Gestor de rols i menús	abr 30	may 3	4d	4d
1.4.4	▼ Subsistema de registre	may 4	may 10	7d	7d
1.4.4.1	Consulta d'events	may 4	may 7	4d	4d
1.4.4.2	Inscripció als events	may 8	may 9	2d	2d
1.4.4.3	Baixa dels events (baixa de l'inscripció)	may 10	may 10	1d	1d
1.4.5	▼ Subsistema de comunicació	may 11	may 13	3d	3d
1.4.5.1	Implementació de forum de l'event	may 11	may 13	3d	3d
1.4.6	▼ Subsistema de informació	may 14	may 16	3d	3d
1.4.6.1	Extracció dels llistats dels assistents	may 14	may 16	3d	3d
1.5	Testing i qualitat	may 17	may 22	6d	6d
1.6	Memòria i presentació	may 23	jun 19	28d	28d

Fig 1 Llistat de tasques

Les principals diferències entre el llistat de tasques i el real són les proves unitàries que es van allargar fins ben bé la finalització de l'etapa de desenvolupament degut a que a mesura que avançava el projecte anava millorant les habilitats amb les eines escollides cosa que feia que es busquessin de noves per millorar certes parts de la lògica de negoci i/o de la gràfica; a més va haver un grau de desviament en la fase de implementació degut a uns errors detectats en el disseny del diagrama de classes i en el diagrama entitat-relació.

Els canvis en la planificació van exigir allargar uns dies les fases de disseny i implementació així com reduir els dies disponibles per a la realització de proves i confecció de la memòria. Aquest fet es pot veure en el diagrama de Gantt¹¹ que es va utilitzar pel seguiment de les tasques abans esmentades.

¹¹ http://es.wikipedia.org/wiki/Diagrama_de_Gantt. Eina gràfica que te per propòsit mostrar el temps de dedicació previst per diferents tasques o activitats.

Gesevent. Gestió d'events.

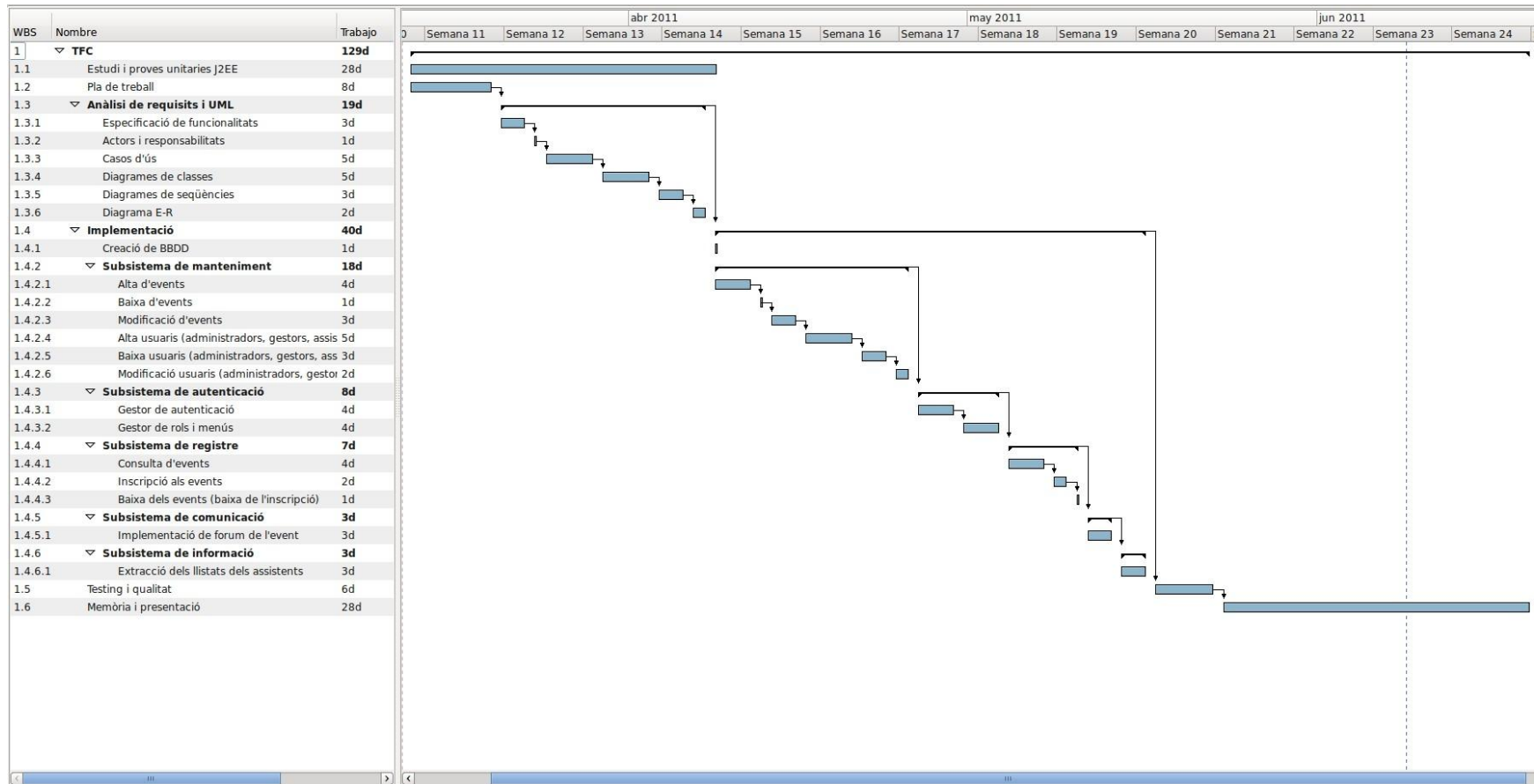


Fig 2 Diagrama de Gantt

En el diagrama de Gantt és pot veure el fet que la desviació a la hora de realitzar el disseny va fer desplaçar els dies necessaris per la realització d'aquesta tasca que va passar de dinou dies a vint-i-set dies. Això va fer endarrerir l'inici de la fase de implementació que a la vegada per si mateixa va suposar un cost temporal més alt del previst i va passar de trenta-nou dies a quaranta-cinc dies.

Aquest fet també va fer desplaçar-se les tasques de proves i redacció de la memòria que per suplir aquest desviament ocasionat es van reduir de sis dies a quatre dies per la fase de proves, i de vint-i-vuit dies a disset dies per la de la redacció de la documentació.

2.5. Productes obtinguts

El producte obtingut és una aplicació empresarial anomenada Gesvent. L'aplicació és realitzada en tecnologia Java, plataforma J2EE, amb components de negoci EJB de manera que per la seva execució és necessària la utilització d'un servidor d'aplicacions que tingui un contenidor per executar aquest tipus de components.

El producte obtingut es pot dividir en diferents parts:

- Aplicació empresarial amb el client web i els components de negoci empaquetats en un fitxer de tipus "ear". Aquest serà el fitxer de desplegament de l'aplicació en el servidor d'aplicacions.
- Fitxers de configuració del servidor d'aplicacions jBoss (versió 5.0) que és el recomanat degut a que l'aplicació s'ha desenvolupat sobre ell.
- Una carpeta amb la llibreria necessària per la connexió amb la base de dades. En aquest cas és recomana PostgreSQL 8.4, ja que ha sigut la base de dades sobre la qual s'ha desenvolupat la aplicació.
- Dos fitxers de configuració de la pròpia base de dades. El primer d'aquests fitxers amb conté els scripts necessaris per la creació de la base de dades de l'aplicació. El segon fitxer disposa d'una sèrie de sentències SQL que donen d'alta les dades mínimes per poder treballar amb l'aplicació.
- És disposa de la documentació del producte en format Javadoc per si hi ha necessitat de fer una extensió del producte. Es recorda que l'aplicació és realitzada de manera que sobre els components de negoci puguin interactuar més d'un client diferent i no només el que es presenta amb aquest treball.
- Un manual de instal·lació pas a pas del producte.
- Un manual d'ús del producte.

2.6. Contingut de la memòria

En els següents capítols de la memòria presentada es podran trobar comentats els diferents aspectes els quals corresponen típicament al procés de desenvolupament de un producte informàtic d'aquest tipus: requisits funcionals del sistema, disseny dinàmic del sistema, disseny estàtic del sistema i arquitectura emprada en la implementació.

A més també es trobaran aspectes purament informatius sobre el programari utilitzat en el desenvolupament del projecte de manera que pugui servir com a referència a posteriors desenvolupadors que vulguin de realitzar una extensió del producte o un desenvolupament similar i estiguin cercant referències.

3. Especificació funcionalitats i anàlisi de requeriments

3.1. Introducció

En aquest capítol es plantejarà el problema a resoldre, és a dir, la necessitat que sorgeix a la hora de gestionar els events que vol impulsar una empresa determinada i quina és la primera aproximació necessària per tenir un producte cent per cent funcional.

A partir de la especificació del problema a resoldre és farà un anàlisi dels diferents usuaris que interactuaran amb ella i una primera descomposició de funcionalitats per usuari per després poder agrupar aquestes funcionalitats en diferents subsistemes que podran ser desenvolupats com a paquets independents.

3.2. Descripció del projecte

Moltes organitzacions inverteixen una quantitat important de recursos per tal de donar a conèixer les seves activitats tant als seus clients, com als possibles clients futurs així com als seus empleats.

El que es pretén millorar és la gestió dels events, de qualsevol tipus, que aquestes empreses volen presentar. Per aquest motiu el programa haurà de ser capaç de definir i publicar aquests events, fet que implicarà la necessitat d'una quantitat mínima de personal que els gestioni, ja que principalment l'usuari serà l'encarregat de registrar-se en els diferents events en els que estigui interessat. Això comportarà una reducció del cost a nivell organitzatiu.

Els diferents actes promocionats poden ser de qualsevol tipus i és poden celebrar a qualsevol lloc (podem pensar en grans empreses amb diferents sucursals) de manera que la informació que se li ha de subministrar a l'usuari és el tipus de event: si és xerrada, taula rodona; la temàtica i l'ubicació del mateix.

En una primera instància l'organització haurà de disposar d'un usuari administrador que serà l'encarregat de donar d'alta els diferents actes que l'empresa vol oferir als usuaris del portal de gestió d'events. Un cop donat d'alta l'acte amb les dades necessàries, l'administrador assignarà a una persona que s'encarregarà de gestionar l'event i aquesta interactuarà amb les persones usuàries inscrites a l'event mitjançant una sèrie de taulells de missatges que el propi gestor podrà crear amb els diferents temes que necessiti tractar amb els usuaris inscrits.

A més, aquest gestor de l'event serà l'encarregat de donar més informació sobre el propi event, indicar les places disponibles, treure llistats amb els inscrits, coordinar els taulells de preguntes sobre els events així com analitzar la satisfacció de les persones usuàries inscrites

als events.

Per facilitar als usuaris el ventall dels últims actes oferts hi haurà en la pantalla principal del portal els últims events publicats de forma que els usuaris visitants, encara que no estiguin registrats, podran veure els últims events donats d'alta al sistema i així decidir si volen inscriure's en algun d'ells.

En cas d'interès per part de l'usuari d'assistir a un dels events promocionats, haurà de realitzar el registre a l'aplicació emplenant un formulari amb les seves dades personals. Un cop emplenat el formulari amb les dades mínimes obligatòries, el sistema li retornarà un identificador d'usuari i una contrasenya, la qual podrà ser modificada per ell sempre que ho desitgi.

Un cop dintre del sistema, l'usuari podrà cercar tots els events que l'organització hagi publicat i podrà inscriure's en aquells que estigui interessat. La cerca d'events disponibles estarà limitada en forma de filtratge a la consulta de manera que cap usuari no pugui enfonsar el sistema al demanar tots els events disponibles si n'hi han molts events donats d'alta.

Un cop inscrit en un event podrà demanar més informació sobre el mateix al fòrum del propi event. Així mateix, podrà donar-se de baixa d'events on s'hagi inscrit; a més que en cas que l'usuari ho desitgi es podrà donar de baixa del portal de gestió d'events. En cas que a posteriori un usuari donat de baixa es vulgui tornar a registrar al portal se li donarà un nou codi d'usuari de manera que resti un històric de codis de usuari inscrits a l'aplicació.

3.3. Rols de l'aplicació i casos d'ús per rol

Es pot diferenciar clarament tres tipus de rols o usuaris diferents: l'administrador, el gestor i l'usuari pròpiament dit, que és el destinatari final dels actes promocionats. Es pot fer una primera descomposició de les responsabilitats que hauran d'assumir cada rol dintre de l'aplicació.

La descripció dels rols a l'aplicació és la següent:

- Administrador: aquest s'encarregarà de donar d'alta, baixa i modificació dels diferents events que l'eina gestionarà, així com podrà donar d'alta els usuaris gestors de events i qualsevol altre tipus de usuari si fos necessari (altres administradors o usuaris).
- Gestor d'events: aquest usuari serà l'encarregat de un cop informat l'event a l'eina, per part de l'administrador, d'indicar l'aforament màxim de l'event, podrà treure llistats dels assistents als events, penjar informació sobre el mateix (abans i després de la seva realització), fer de moderador en fòrums de informació sobre l'event així

com en fòrums de satisfacció del mateix.

- Assistents als events (usuaris): aquest usuari s'haurà de enregistrar a l'eina per tal de poder inscriure's als events, veure els events on està inscrit i veure els que va assistir i estan finalitzats de forma que pugui accedir a la diferents documentació que pugui ser penjada sobre aquests a més de poder participar en fòrums de opinió sobre els mateixos.

A continuació i a partir dels requisits establerts i de l'anàlisi dels rols implicats és poden veure les responsabilitats que te cada rol. Per al cas de l'administrador és poden descompre en:

- Alta d'events: l'usuari administrador podrà donar d'alta qualsevol event en el sistema. Aquests els haurà de categoritzar per tipus (conferències, cursos, ...), temàtiques (sanitat, tecnologia, ...), zona geogràfica on s'imparteixen (país, carrer, ubicació, sala, ...), gestor encarregat, etc. Totes aquestes dades necessàries per donar d'alta l'event estaran tabulades, és a dir, el primer cop que s'introdueixin es guardaran en el sistema de forma que si són un altre cop utilitzades per informar cap altre event aquestes ja estaran disponibles.
- Baixa d'events: en cas que l'event no s'arribi a publicar per qualsevol tipus de motiu: anulació, cap usuari inscrit, etc. L'administrador l'eliminarà del sistema.
- Modificació d'events: en cas d'un error en les dades o un canvi en la ubicació de l'event l'administrador del sistema podrà realitzar les modificacions oportunes sobre el mateix.
- Alta de usuaris gestors d'events: cada event tindrà un usuari encarregat de penjar informació del mateix així com de respondre al assistents sobre els dubtes que aquests puguin tenir sobre el mateix. La gestió d'aquests usuaris la farà el mateix usuari administrador, així com l'assignació d'aquests usuaris als diferents events que en aquell moment s'estiguin donant d'alta en el sistema.
- Baixa de usuaris gestors d'events: podrà donar de baixa lògica als usuaris gestors del events incloent la seva data de fi de contracte.
- Modificació d'usuaris de gestor d'events: podrà modificar les dades de qualsevol persona registrada en l'aplicació en cas que les dades siguin incorrectes.

L'usuari de gestor d'events tindrà les següents funcionalitats al seu abast.

- Modificació de l'event: l'usuari amb rol de gestió d'events pot modificar certa informació dels events que s'han donat d'alta prèviament. Part de les seves tasques són definir el nombre de places a l'event, ja que ,encara que en una ubicació pot haver un aforament màxim, l'event pot requerir que el nombre de places i l'aforament no coincideixin. A més aquest usuari també s'encarregarà d'assignar els ponents a l'event. Recordem que aquests ponents els donarà d'alta l'administrador, junt amb l'alta de l'event.
- Informar dades relatives a l'event: un cop donat d'alta l'event i aquest hagi estat assignat a

un gestor, el gestor s'haurà d'encarregar d'afegir dades significatives sobre el mateix, és a dir, afegir informació relativa a l'acte tal com: ubicació, activitats de l'event, etc.

- Manteniment dels taulells de missatges de l'event: els assistents a l'event es podran posar en contacte amb el gestor de l'event mitjançant un taulell (o taulells) de missatges en el qual s'intercanviaran comunicacions per escrit, els quals poden ser de tipus preguntes respostes o simplement, un cop finalitzat l'event, del grau de satisfacció dels usuaris amb l'acte en qüestió o millores per properes edicions. Aquests taulells de missatges seran definits pel gestor en tipus i en número adequant-se a les necessitats de cada acte i moment.
- Extracció de llistat dels assistents: el gestor podrà extreure un llistat dels assistents de l'event.

Les funcionalitats que tindran a l'abats els als usuaris assistents seran les següents:

- Registre d'usuari: els usuaris podran realitzar el registrar a l'aplicació per tal de veure els events disponibles i poder-se inscriure's.
- Modificació de les dades: l'usuari en tot moment podrà modificar les seves dades així com canviar-se la contrasenya.
- Anul·lació del registre: l'usuari podrà anul·lar el seu registre al portal de gestió d'events. En cas que en un futur es tornés a registrar el sistema li assignaria un identificador nou. Per motius de manteniment de històric els identificadors de usuaris no seran re aprofitables.
- Participació en els taulells de missatges de l'event inscrit: els usuaris que estiguin inscrits a un event podran participar en el mateix donant la seva opinió sobre l'acte o realitzant preguntes al gestor de l'event per tal resoldre els seus dubtes.
- Consulta d'events: els usuaris podran consultar els events passats, en curs o futurs. En quant a la consulta d'events futurs els usuaris podran cercar els que els interessin per poder realitzar la inscripció.
- Inscripció als events: l'usuari podrà inscriure's a un event, si aquest l'interessa.
- Baixa de inscripció a un event: si per qualsevol tipus de motiu l'usuari no pot assistir a l'acte podrà donar-se de baixa de l'event.

3.4. Subsistemes i funcionalitats per subsistema

A partir de les funcionalitats vistes en l'apartat anterior podem agrupar-les en cinc subsistemes diferents:

Subsistema de manteniment: conté tota la funcionalitat encarregada de nodrir de la base dades amb la informació necessària pel bon funcionament de l'aplicació.

Subsistema d'autenticació: les funcionalitats que inclou són la gestió de l'autenticació dels usuaris i la gestió per rols de l'aplicació, així com carregar els menús depenent del rol autenticat.

Subsistema de registre: la seva principal funció és permetre que una persona usuària interessada consulti, es doni d'alta i de baixa dels diferents events publicats en l'aplicació.

Subsistema de comunicació: és l'encarregat de la gestió de missatges entre els usuaris de tipus assistents i els gestors d'events. El subsistema contindrà la lògica del taulell de missatges de l'aplicació

Subsistema d'informació: compleix la funcionalitat estadística de l'aplicació permeten als usuaris gestor de l'aplicació treure els llistats dels assistents als events que s'han inscrit.

Un cop descompost els requisits inicials en possibles subsistemes, s'agrupen les funcionalitats vistes en aquests subsistemes que s'acaben de establir:

1 . Subsistema de manteniment: Les funcionalitats assignades a aquest subsistema seran,

- 1.1. Alta, modificació, baixa i consulta de tipologies d'events.
- 1.2. Alta, modificació, baixa i consulta de temàtiques.
- 1.3. Alta, modificació, baixa i consulta de països.
- 1.4. Alta, modificació, baixa i consulta de carrers.
- 1.5. Alta, modificació, baixa i consulta d'ubicacions.
- 1.6. Alta, modificació, baixa i consulta d'events.
- 1.7. Alta, modificació, baixa i consulta d'usuaris.

2. Subsistema d'autenticació: Les funcionalitats assignades al subsistema d'autenticació són les següents:

- 2.1. Gestor d'autenticació: és la part del programari que s'encarrega del control d'accés a l'aplicació dels usuaris registrats en la mateixa.
- 2.2. Gestor de rols i menús: és la part del sistema que publicarà a cada usuari unes opcions de menú o unes altres depenent del seu rol.

3. Subsistema de registre: el subsistema de registre conté les funcionalitats:

- 3.1. Inscripció als events: permet que un usuari interessat en un event es pugui inscriure.

3.2. Baixa els events (baixa de l'inscripció): aquesta funcionalitat permet que un usuari es doni de baixa d'un event al que s'ha inscrit prèviament i que encara no s'ha iniciat.

3.3. Subsistema de comunicació: el subsistema conté la funcionalitat que permet als usuaris comunicar-se entre si mitjançant els taulers de missatges.

4. Subsistema de comunicació: el subsistema de comunicació conté la següent funcionalitat:

4.1. Tauler de missatges dels events: aquesta funcionalitat permet gestionar els diferents missatges que han de circular entre els usuaris per cada event publicat en el sistema.

4.2. Tauler d'informació: aquesta funcionalitat permet al gestor dels events publicar qualsevol informació d'interès sobre un event, de manera que els usuaris puguin consultar les últimes notícies o informacions.

5. Subsistema d'informació: el subsistema d'informació o estadístic és aquell que permetrà que un usuari amb el rol adequat (usuari gestor d'events) pugui obtenir un llistat dels usuaris inscrits en un event determinat.

5.1. Extracció dels llistats dels assistents: s'encarregarà de que un usuari amb rol de gestor d'events pugui realitzar una extracció en forma de llistat dels assistents que estan registrats a un event determinat.

4. Diagrames dinàmics de disseny

4.1. Introducció.

En els següents apartats es formalitza els requeriments funcionals de l'aplicació mitjançant el llenguatge formal de modelatge UML¹². Aquests llenguatge de modelatge permet una formalització en la documentació i facilita una millora en els processos de desenvolupament gràcies a la seva estandardització en el món del desenvolupament amb la teoria d'objectes.

4.2. Casos d'ús per subsistema

En aquest apartat es descriuen els casos d'ús per subsistema així com els actors involucrats en aquests casos d'ús. En el cas que hi hagi casos d'ús relacionats entre subsistemes, aquests són explicitats en els diagrames corresponents.

Subsistema de manteniment

En aquest subsistema intervenen els actors: administrador i gestor. En el cas d'ús estan representats els casos d'alta, baixa i modificació tant per els events com pels usuaris. L'única diferència entre ells és quin o quins actors interactuen amb ells, sent la gestió d'usuaris només permesa a l'actor amb rol d'administrador.

També es representa una cas de manteniment de topologies d'event. La resta de manteniments bàsics (temàtiques, països, carrers i ubicacions) s'han obviat degut a la seva similitud. En aquest cas d'ús l'únic actor responsable és l'usuari administrador.

En tots els casos el funcionament del cas d'ús és similar, abans de realitzar qualsevol acció de baixa o modificació, el sistema comprova, mitjançant una consulta, si l'element a tractar existeix. En cas de no existir, el cas d'ús no finalitzaria de forma correcte donant una excepció. Per tant, és precondition que abans de modificar o donar de baixa un element aquest hagi d'existir.

En les accions de modificació i alta d'events s'han obviat per motius de claredat la dependència que hi ha amb els casos d'ús de manteniment de dades bàsiques dels events: tipologies, temàtiques, països, carrers, ubicacions i usuaris, ja que cada cop que s'ha de donar d'alta o modificar un event aquest ha de realitzar consultes sobre aquests casos per omplir les seves dades.

Ara bé, en cas de modificació si l'element (event, usuari, ...) no existeix automàticament es donaria d'alta en el sistema.

¹² http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

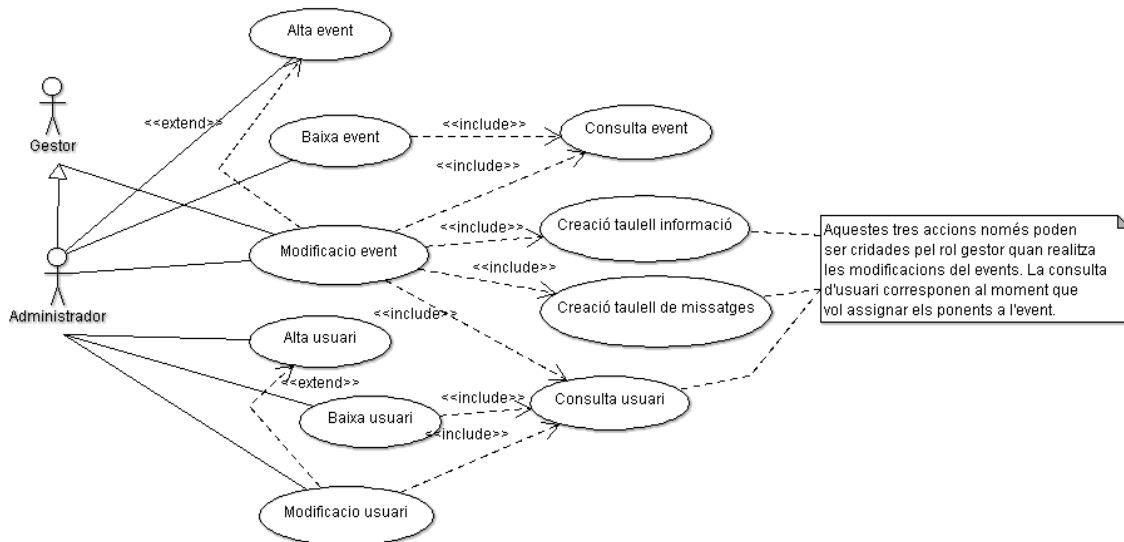


Fig 3 Cas d'ús subsistema de manteniment (events i usuaris)

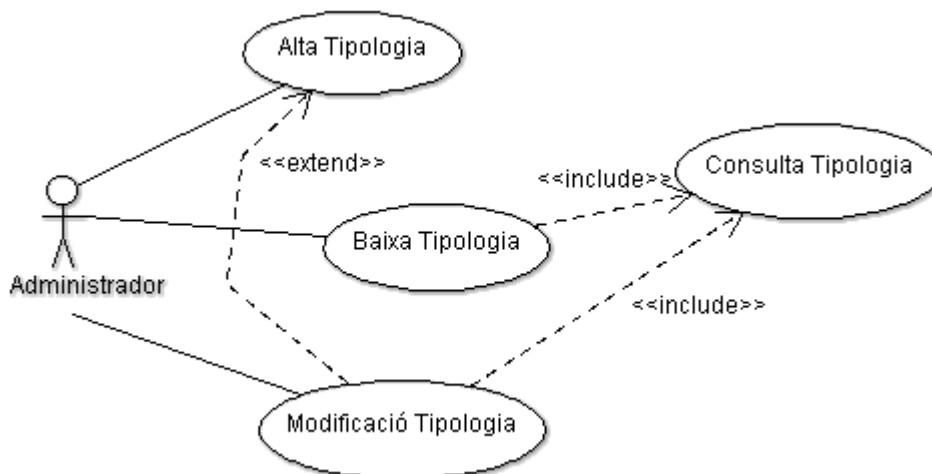


Fig 4 Cas d'ús subsistema de manteniment (tipologies)

Subsistema d'autenticació

En aquest subsistema intervenen com a actors tots els rols descrits per l'aplicació: administrador, gestor d'events i usuari assistent. Això és degut a que aquest subsistema s'encarrega de l'autenticació de l'usuari i de mostrar les diferents opcions o funcionalitats disponibles a l'aplicació depenent del seu rol.

En el cas d'ús es representa com a l'autenticació de l'usuari intervenen de forma indirecta dos casos d'ús: el de consulta que retorna al cas d'autenticació si aquest usuari existeix i quin rol té i el de rols i menús que indica al cas d'autenticació sobre quins serveis pot demanar aquest usuari un cop autenticat.

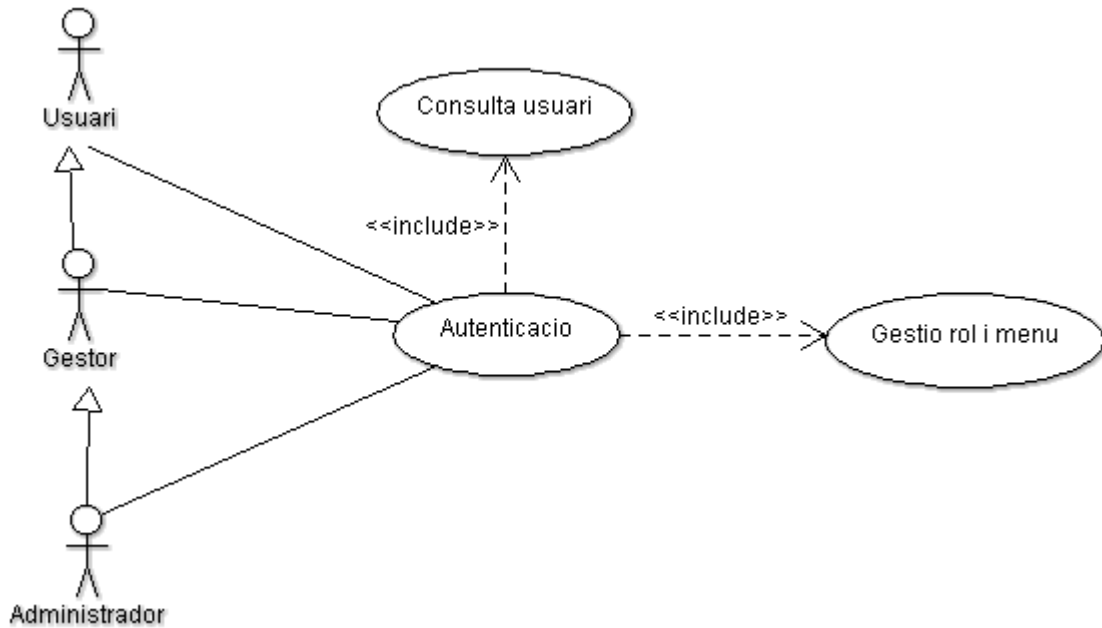


Fig 5 Cas d'ús subsistema d'autenticació

Subsistema de registre

En el subsistema de registre intervé l'usuari que serà l'assistent als events que s'inscriu, a més es pot veure que el cas d'ús consulta d'events és cridat des d'aquest subsistema per realitzar la consulta d'existència d'un event per part de l'usuari, prèvia a la seva inscripció.

També hi ha la utilització del cas de consulta d'event per tal de que l'usuari es pugui donar de baixa d'una inscripció realitzada ja que abans de fer-ho haurà de cercar l'event de on estigui inscrit.

El cas d'ús consulta d'events a més pot ésser cridat de forma independent ja que és la forma que permet a l'usuari consultar si hi ha cap event nou que li pugui interessar per, a posteriori, si escau, fer la inscripció.

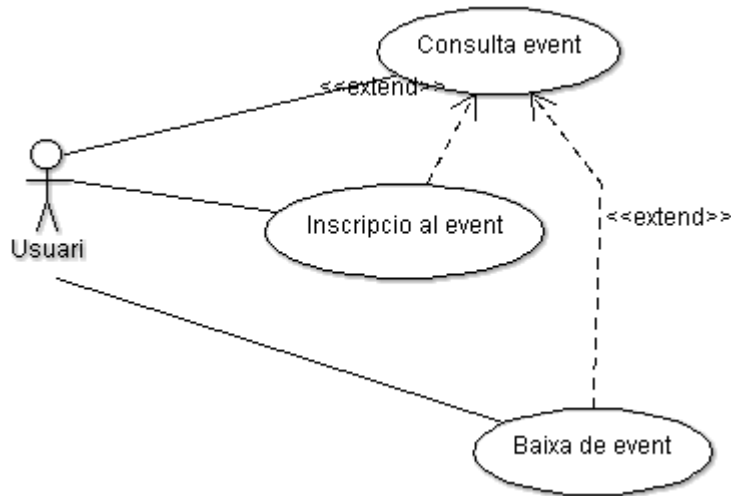


Fig 6 Cas d'ús subsistema de registre

Subsistema de comunicació

El subsistema conté els casos d'ús que permeten als usuaris assistents i als gestors d'events comunicar-se entre si mitjançant la utilització d'un taulell o taulells de missatges associats als events i els d'informació. Una precondició d'aquests casos d'ús és que l'usuari assistent estigui inscrit a l'event i que el gestor estigui associat a aquell mateix event.

Tant l'usuari com el gestor poden enviar missatges al taulell de missatges desitjat per fer-ho primer hauran de seleccionar, en cas que hi hagi més d'un, el taulell de missatges que correspongui amb la temàtica del missatge a escriure. Un cop seleccionat podran escriure el seu missatge per que aquest es publiqui.

A més l'actor gestor podrà publicar informació referent a l'event que l'usuari podrà consultar en qualsevol moment; aquesta informació, però, no permet una interacció entre ells ja que només és de consulta per part de l'usuari. Per la realització d'aquesta interactivitat s'han de remetre als taulells de missatges.

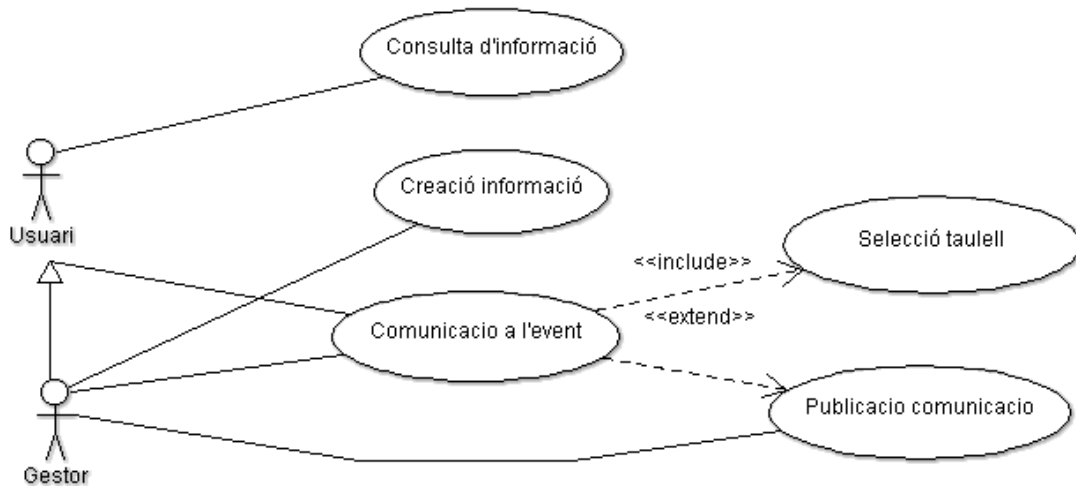


Fig 7 Cas d'ús de comunicació. Taulell de missatges

Subsistema d'informació

És el subsistema a partir del qual un gestor d'events extreu informació dels usuaris inscrits a un event. Aquest cas d'ús és molt simple i només hi ha una petició contra el sistema d'extracció de la informació que conformarà el llistat. Així doncs, el gestor d'events realitzarà un consulta sobre les dades de l'event per recopilar els usuaris inscrits.

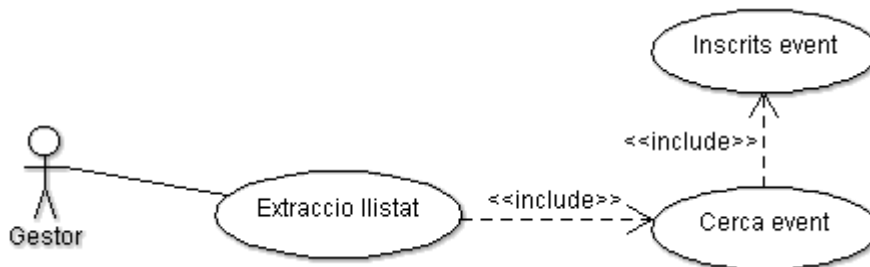


Fig 8 Cas d'ús de informació. Extracció de llistat d'usuaris inscrits.

4.3. Diagrames de seqüència. Especificació dels casos d'ús.

Ens els següents diagrames de seqüència es veuran representats els casos d'ús especificats en l'apartat anterior mitjançant la interacció de l'usuari amb l'aplicació i la interacció de les diferents capes que corresponen a la mateixa. Aquests representen explícitament l'ordre temporal dels missatges i de les operacions que es duen a terme, tant entre objectes de l'aplicació com entre diferents capes. El patró representat per aquestes capes en que es basa l'aplicació és el model-vista-controlador (MVC¹³) que serà explicat més endavant, en

¹³ Model-Vista-Controlador, és un patró de disseny de programari que separa les dades de l'aplicació (model), de la capa de presentació (vista) i aquestes a la vegada de la lògica de control del negoci (controlador).

l'apartat de arquitectura de l'aplicació.

En aquests mateixos diagrames es pot apreciar la separació entre capes i la responsabilitat de cadascuna d'elles. Tots els diagrames tenen els mateixos objectes generals: un actor que llença una acció contra una pantalla la qual li passa les dades introduïdes a un objecte controlador. Aquest s'encarrega de cridar a un objecte remot de la capa de model i, a l'hora, accedeix a la base de dades mitjançant un objecte entitat.

Subsistema de manteniment

Només s'exposen els diagrames de seqüència del manteniment d'usuaris. Els diagrames de manteniments d'events i manteniment de dades bàsiques (tipologies, temàtiques, ...) no es representaran degut a la seva similitud amb els d'usuari, l'única diferència entre ells és l'objecte amb el que treballen (usuari, event, etc) i el rol que posa en marxa la seqüència.

· Alta d'usuari

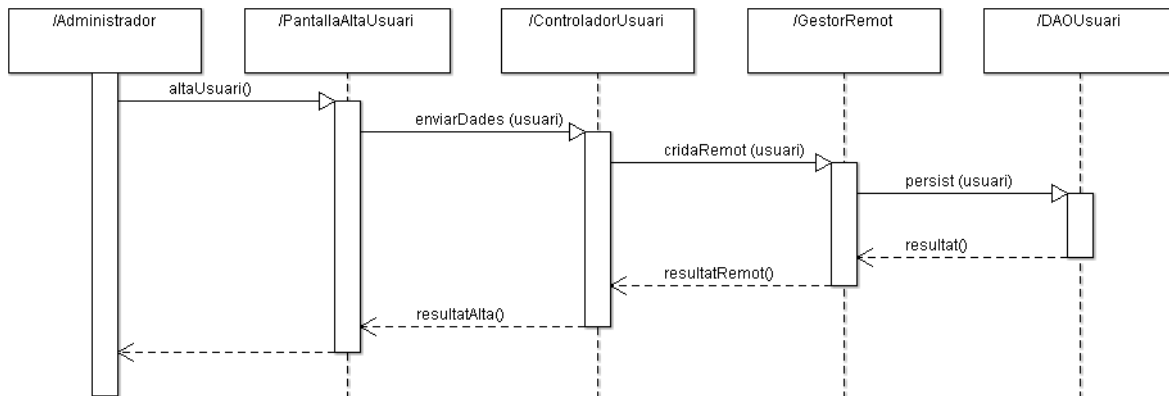


Fig 9 Diag. seq. alta d'usuari

· Modificació usuari

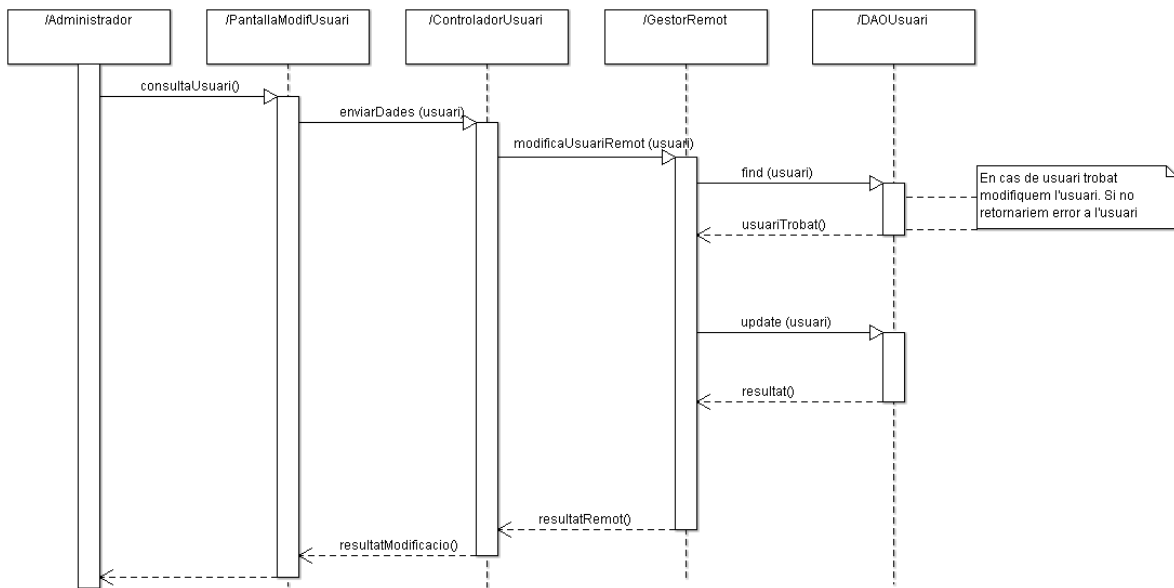


Fig 10 Diag. seq. Modificació d'usuari

· Baixa d'usuari

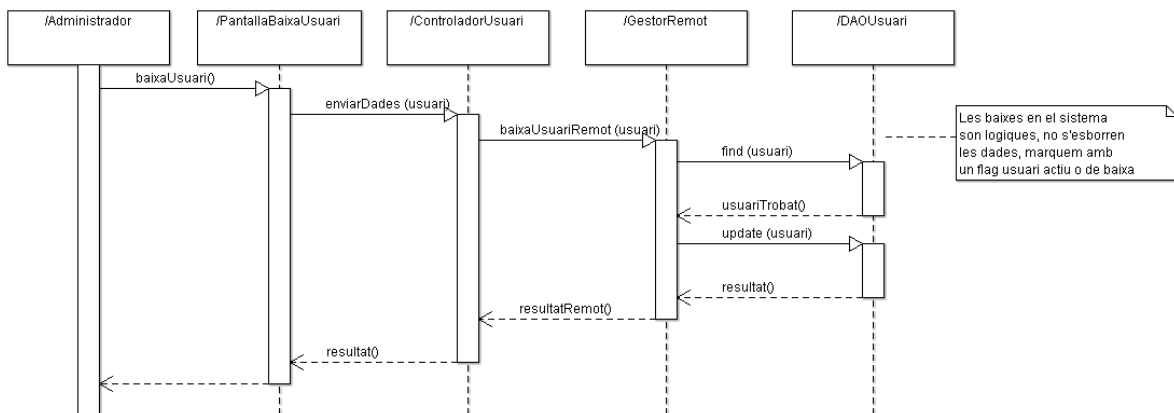


Fig 11 Dia.seq. Baixa d'usuari

Subsistema d'autenticació

Els diagrames de seqüència següents mostren el comportament dinàmic dels casos d'ús del subsistema d'autenticació.

Gesevent. Gestió d'events.

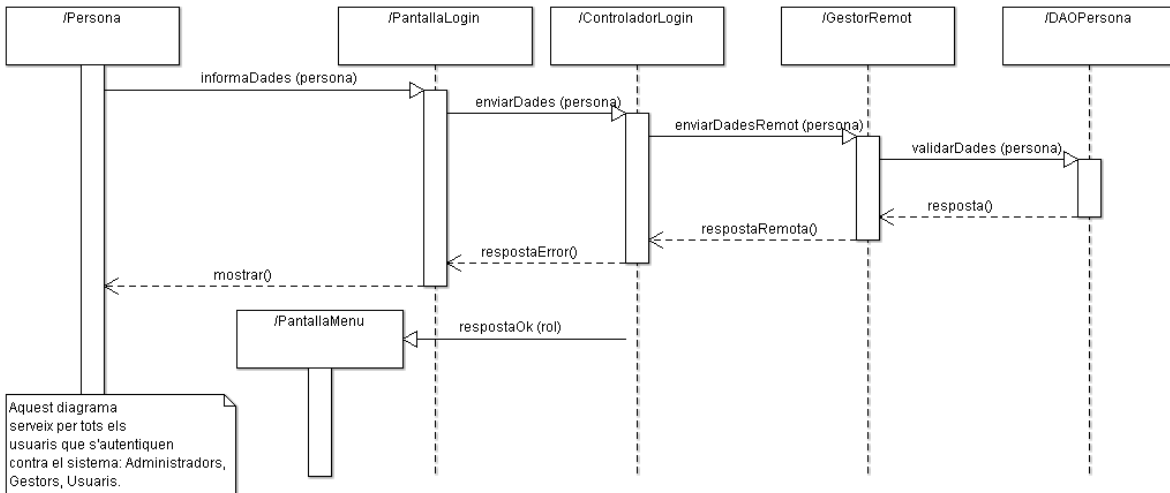


Fig 12 Diag.seq. Gestió d'autenticació

Subsistema de registre

Els diagrames de seqüència següents mostren el comportament dinàmic a la hora de que un usuari es registri a un event determinat, així com la seva baixa. Es pot apreciar com les dos seqüències tenen un comportament dinàmic molt similar fent primerament una cerca de l'objecte que interessa (un event per inscriure's o una inscripció per donar-se de baixa) per després aplicar l'acció corresponent.

· Inscripció a un event.

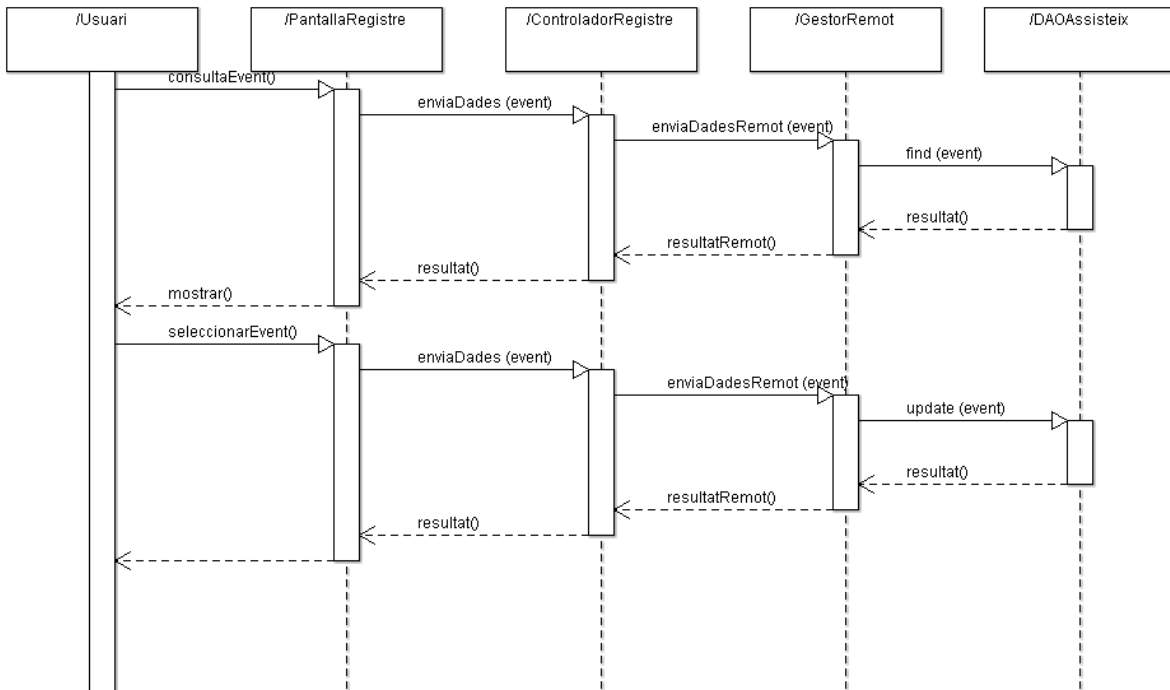


Fig 13 Diag.seq. Inscripció a event

· Baixa de inscripció a event.

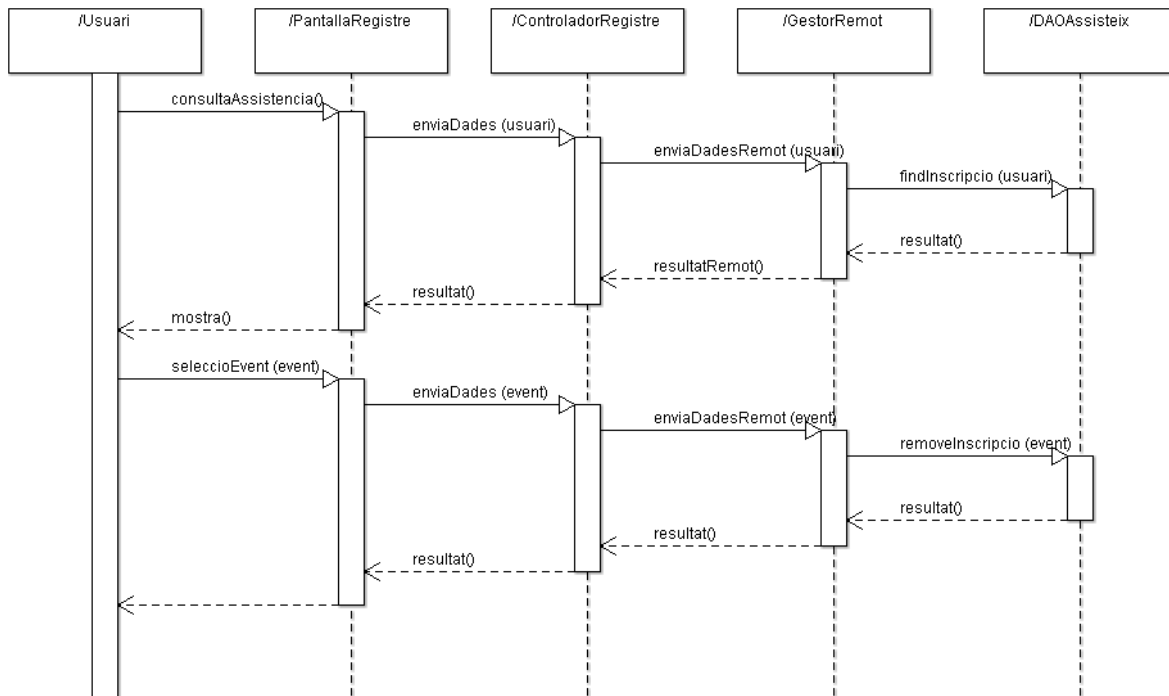


Fig 14 Diag.seq. Baixa de inscripció a event

Subsistema de comunicació

En aquest conjunt de diagrames es representa la interacció dinàmica de l'usuari gestor amb la creació d'informació per a un event del qual es responsable, la consulta d'aquesta informació per part de un usuari assistent a aquests event i el posterior creació i publicació de missatges entre usuaris. Per simplificar la representació en aquesta última seqüència de comunicació només està indicat la creació de un usuari qualsevol, tot i així aquesta creació és por extrapolar a qualsevol dels usuaris que tenen permís per realitzar aquesta acció (gestor i usuari assistent).

· Creació informació.

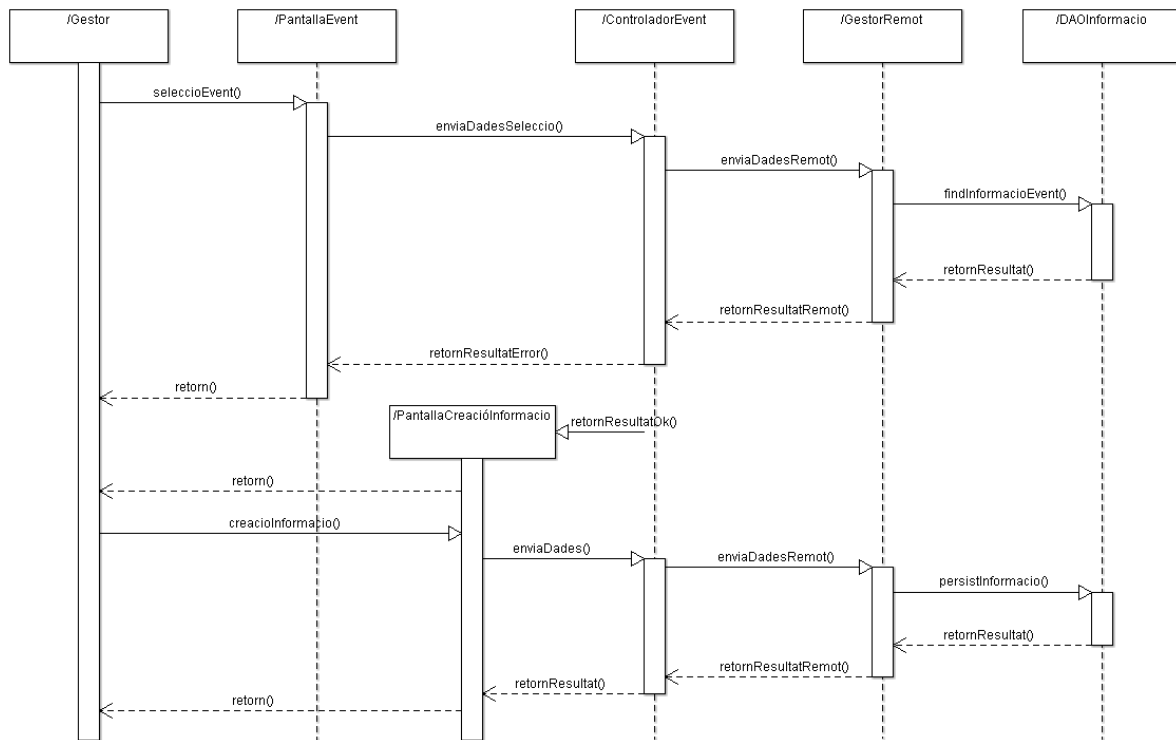


Fig 15 Diag.seq. Creació informació a event

· Consulta d'informació.

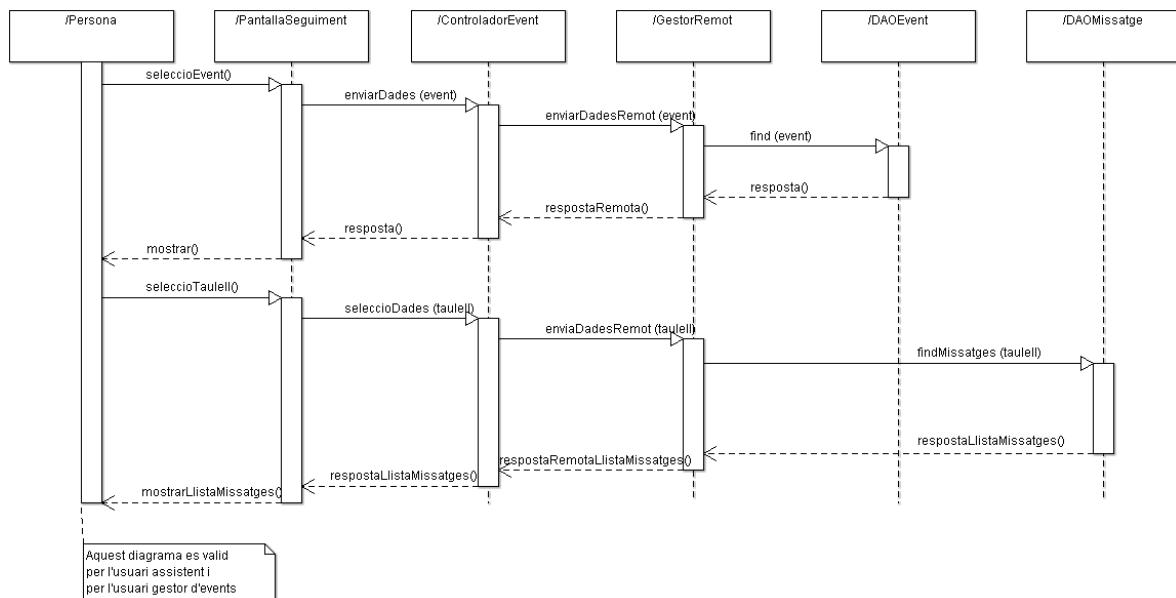


Fig 16 Diag.seq. Consulta de missatges de un taulell

· Enviament de missatges al taulell.

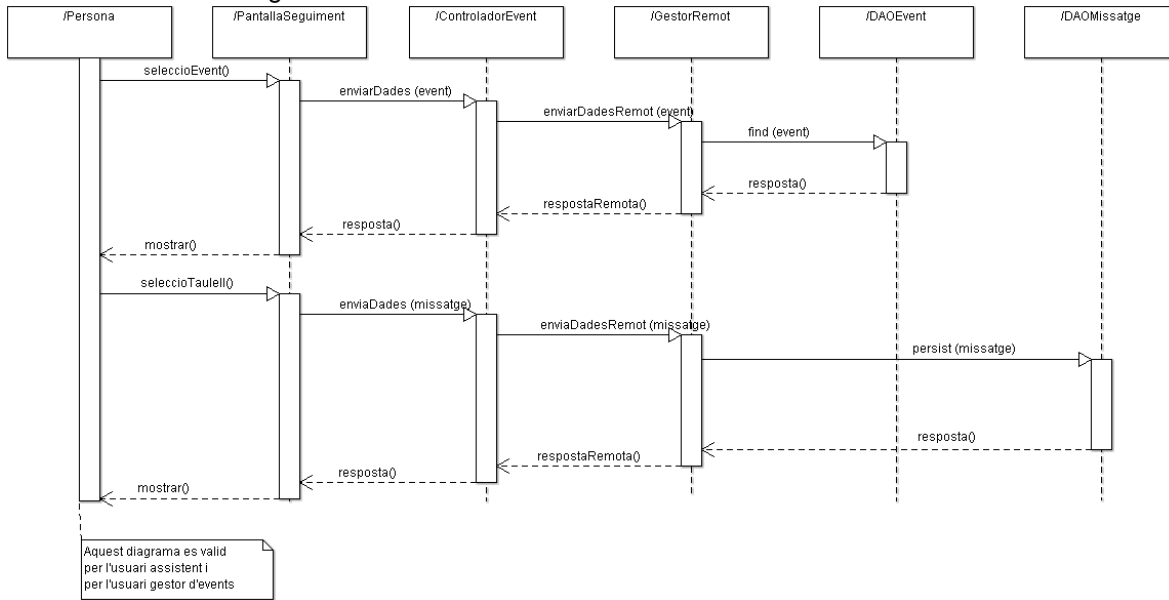


Fig 17 Diag.seq. Enviament de missatges

Subsistema d'informació

El subsistema de informació o estadístic és aquell que permet extreure la informació dels assistents a un event. En aquest cas l'actor que interactua amb el sistema és el gestor d'events que un cop cerca l'event desitjat pot extreure un llistat dels usuaris que hi estiguin inscrits.

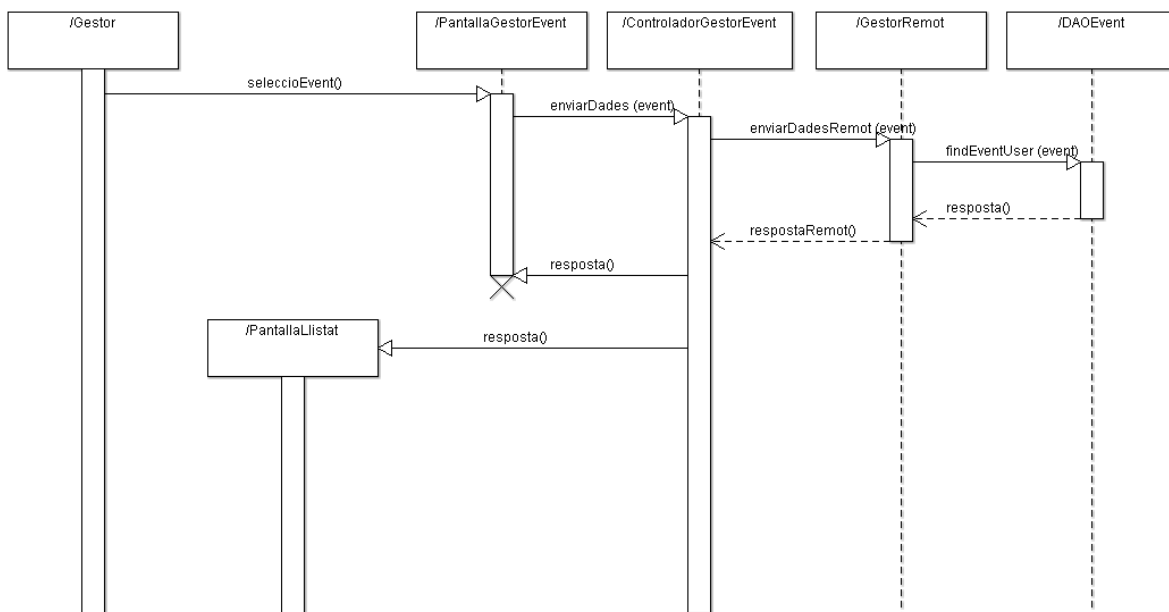


Fig 18 Diag.seq. Extracció de llistats de usuaris inscrits

4.4. Diagrames d'estat.

En aquest apartat es mostren els diagrames d'estats mes importants del sistema, els quals representen els diferents moments pels que passa un element de l'aplicació durant la seva vida. En concret es presenten els dos més importants de l'aplicació, l'accés a l'aplicació d'usuaris externs (usuaris que es registren de forma autònoma en el sistema, sense l'acció de l'administrador) i la gestió dels events.

· Gestió de l'accés a l'aplicació d'usuaris.

El primer diagrama correspon a la fase d'autenticació pels que pot passar un usuari extern que estigui interessat en algun dels events oferts pel sistema. Si l'usuari estava prèviament registrat al sistema només ha de autenticar-se contra ell de manera que un cop autenticat té accés a l'aplicació.

En cas que no tingui un usuari en l'aplicació haurà de registrar-se omplint un formulari per aquest fi. En cas que falti alguna dada obligatòria per donar-lo d'alta el sistema li avisarà amb el missatge corresponent fins que les dades estiguin complertes, en aquest cas l'aplicació li retornarà el seu codi de usuari amb el qual la persona ja podrà autenticar-se dintre del sistema i operar amb ell.

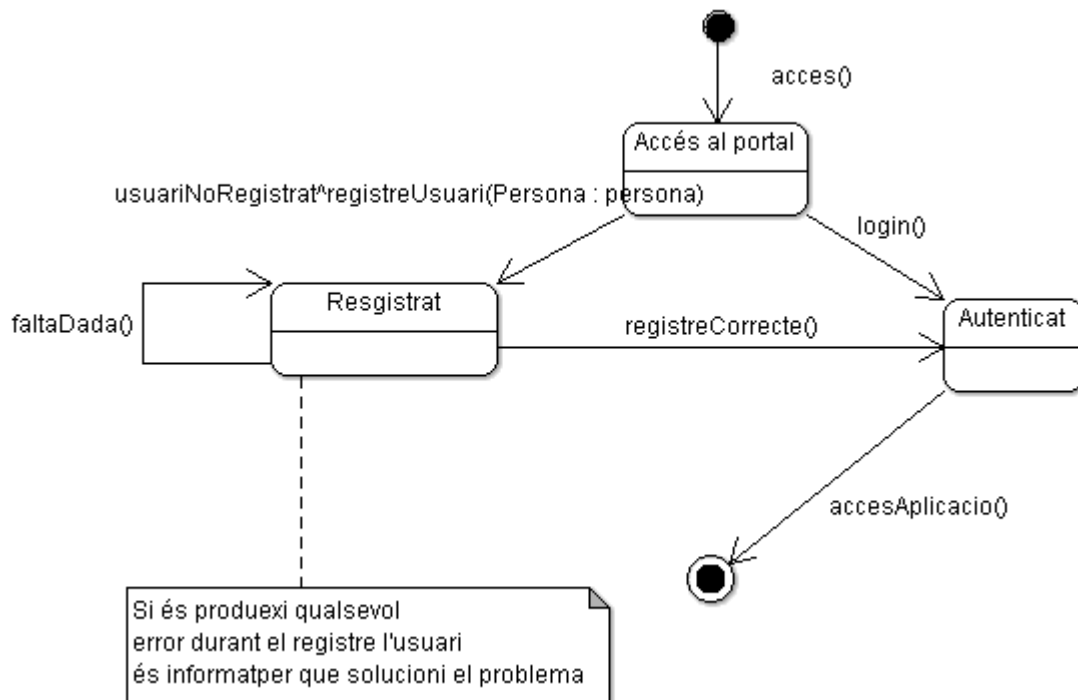


Fig 19 Estat. Estat registre usuari extern

Un altre element de l'aplicació que passa per diferents estats és l'objecte Event. Un cop el crea l'usuari amb el rol administrador passa a estat *creació* que permetrà, posteriorment, passar a l'estat *informat pel l'administrador*, el qual introduirà les dades mínimes de l'event. Un cop ho hagi fet assignarà el gestor que se'n carregarà d'acabar de informar les dades de l'event (ponents, informació, taulells de missatges) i el publicarà per que així sigui visible pels usuaris assistents i, en cas que hi estiguin interessats, puguin realitzar la inscripció pertinent.

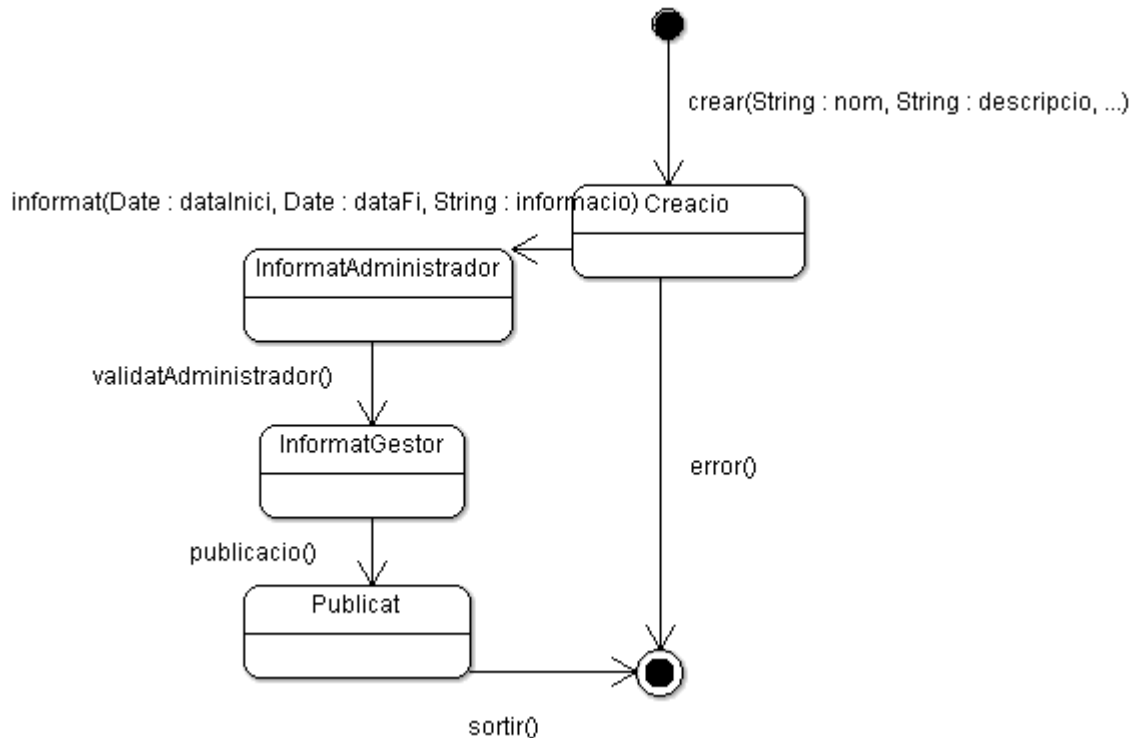


Fig 20 Estat. Estat creació event

5. Diagrames estàtics de disseny

5.1. Distribució de l'aplicació i diagrama de paquets.

Com es veurà més endavant l'arquitectura de l'aplicació utilitza les funcionalitats dels components EJB per separar la lògica de negoci de la capa de control i de la capa de vista. A més, l'accés a les dades s'ha dissenyat utilitzant la *Java Persistence API*¹⁴ la qual és un bastiment que permet tenir una relació entre el món dels objectes i el relacional, és a dir, la base de dades, de manera que es pot treballar sempre pensant en la orientació a objectes i accedir d'una manera transparent al model de dades relacionals.

Per aquest motiu l'aplicació està desenvolupada en tres projectes diferents que faciliten la seva distribució i portabilitat. Els tres projectes en que està dividida l'aplicació són:

- geseventWeb → aquesta és la part client de l'aplicació, la qual conté la capa de control i part de la lògica de validació de dades, la que s'encarrega de la validació de dades lògiques com: introducció correcte de enters, format de dates, etc.
- geseventJPA → conté els Entity Beans¹⁵ de l'aplicació on es troben els objectes que tenen equivalència amb les taules de la base de dades.
- geseventEJB → aquest conté els objectes de la lògica de negoci. Aquest projecte conté els objectes façana¹⁶ que són presentats al client per que pugui crida a la lògica de negoci desitjada. Aquests objectes façana criden als diferents components de negoci que estan ubicats als seus paquets corresponents.

Aquests dos últims projectes conformen la part servidora del sistema, la qual pot donar serveis tant al client que s'ha implementat en aquest treball com a qualsevol altre client que es desenvolupi, poden ser aquests de tipus serveis web (*Web service*)¹⁷, clients pesats tals com Swing¹⁸, etc. Això permet la reutilització de la part servidor mitjançant altres clients.

En el diagrama de paquets que es mostra a continuació es pot veure la distribució del programari en els diferents subsistemes exposats i la interacció entre ells. En el cas que s'exposa es pot veure com tots els paquets interaccionen amb el paquet d'autenticació, ja que és aquest el que dona l'entrada al sistema pels usuaris, a més que el paquet de registre interacciona, mitjançant la cerca d'events, amb el paquet de manteniment.

¹⁴ <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>

¹⁵ Objecte que representa un objecte de negoci en el mecanisme de persistència. http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/EJBConcepts4.html

¹⁶ Patro de disseny que utilitza un objecte, façana, per publicar els serveis als clients dels objectes de negoci.

¹⁷ http://es.wikipedia.org/wiki/Servicio_web

¹⁸ http://es.wikipedia.org/wiki/Swing_%28biblioteca_gr%C3%A1fica%29

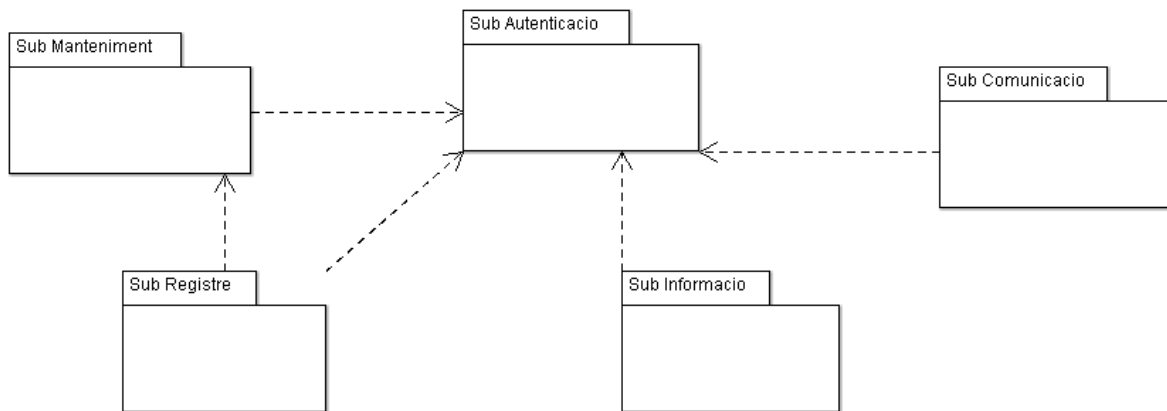


Fig 21 Diagrama de paquets

5.2. Diagrama de classes.

En el següent diagrama estàtic de disseny només estan representades les classes d'entitats i s'obvien la resta de classes per tal de proporcionar una imatge el més aclaridora possible del sistema. Igualment, per motius de claredat del diagrama només s'han representat els atributs de les classes, sense cap tipus de mètode.

Les classes representades han sigut extretes de l'estudi dels requisits presentats al capítol tres d'aquesta memòria. Mitjançant el seu anàlisi s'han extret les classes representades així com els seus atributs.

Com a puntualització esmentar que en el diagrama de classes són representades les classes *Gestor*, *Ponent*, *Usuari* i *Administrador*. Tot i que apareixen en realitat són els diferents rols (veure classe entitat *Rol* del diagrama) que gestiona l'aplicació. La seva representació es deu a que depenent del rol la interacció amb la classe *Event* es produeix d'una forma o un altre i el que es pretén és reflectir de manera més entenedora aquestes interaccions.

Gesevent. Gestió d'events.

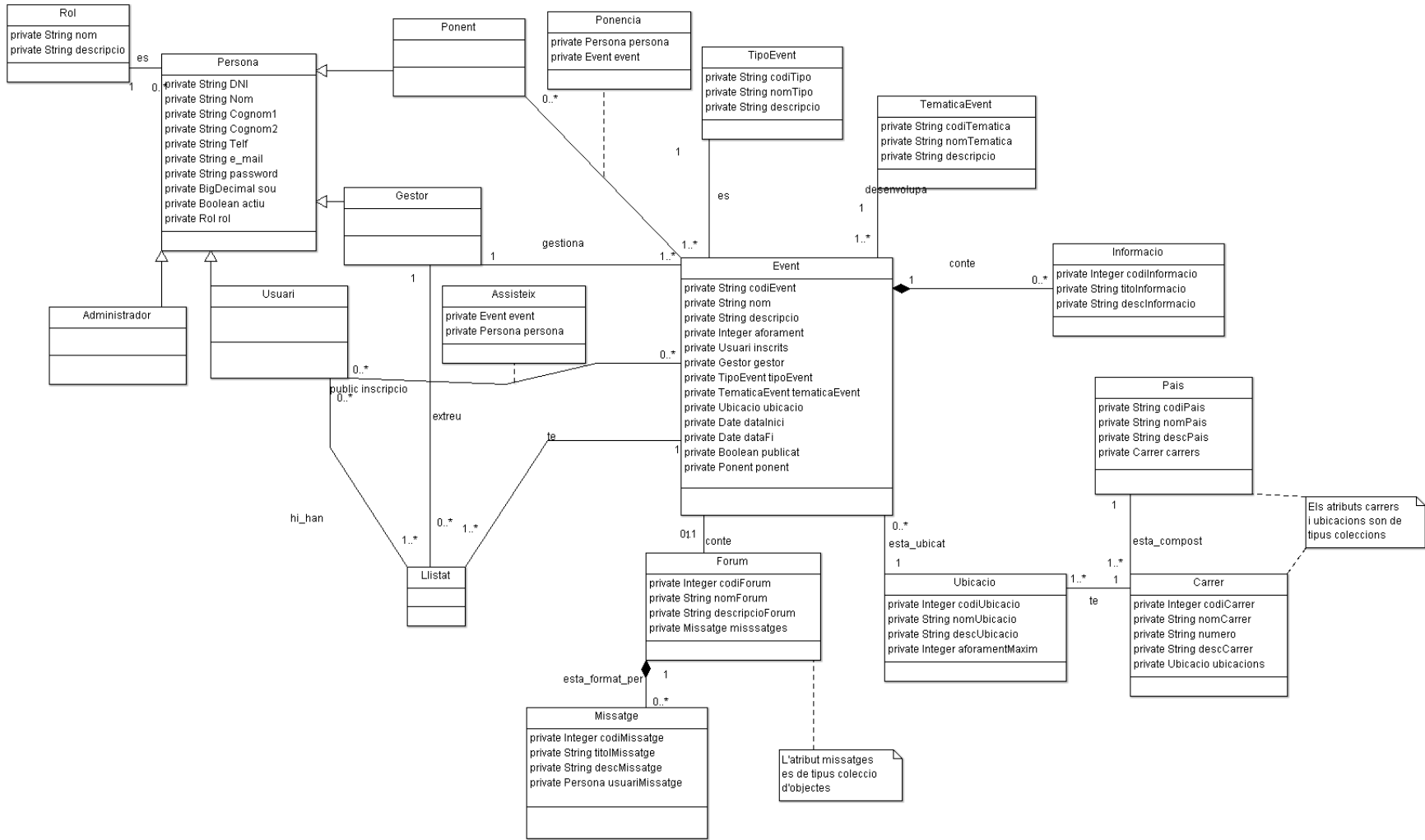


Fig 22 Diagrama de classes

5.3. Model entitat-relació.

Pel disseny de la base de dades s'utilitza el model entitat-relació que per la seva senzillesa i independència del gestor de base de dades relacional permet mostrar la relació entre taules i les diferents cardinalitats.

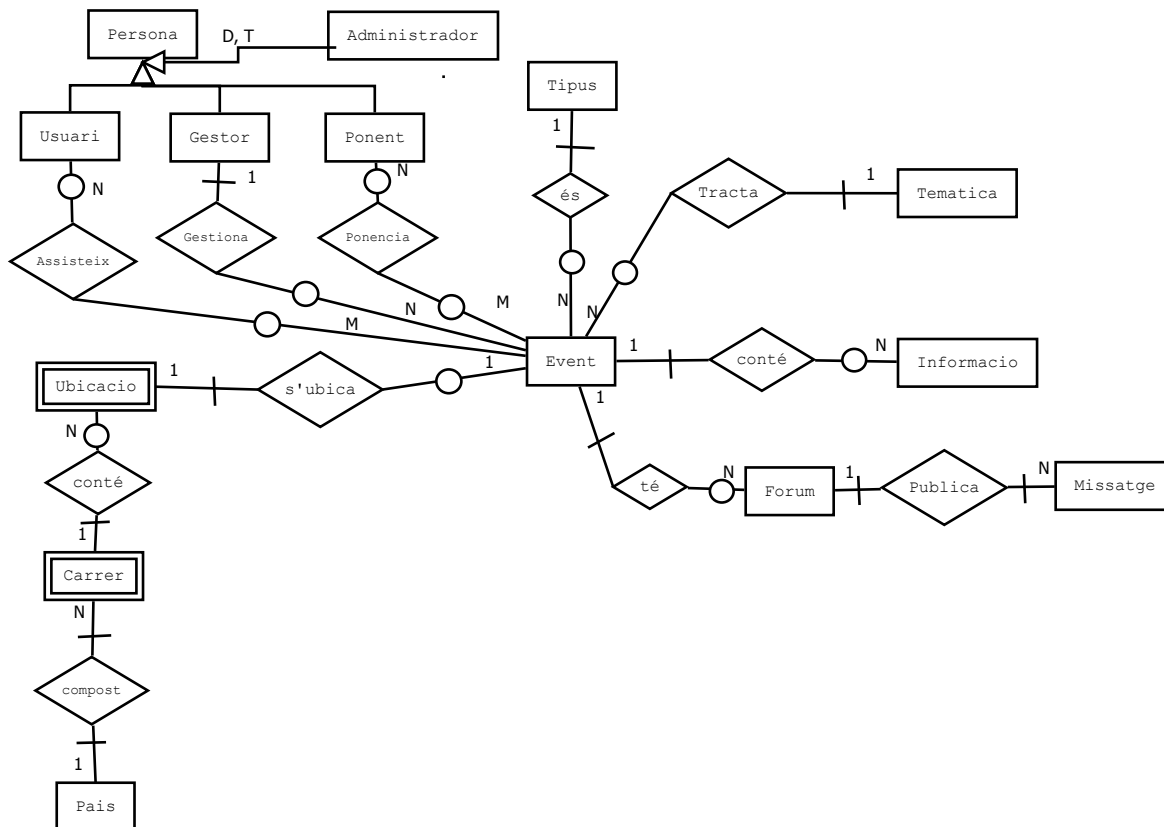


Fig 23 Model entitat-relació

En el diagrama es pot veure que hi ha dos interrelacions amb cardinalitats M-N (Usuari-Event i Ponent-Event). Aquestes seran transformades en noves relacions en el model relacional.

Els atributs de les entitats que figuren en el diagrama entitat-relació anterior són:

EVENT

codi_event, nom, descripcio, aforament, codi_gestor, codi_tipus, codi_tematica, codi_pais, codi_carrer, codi_ubicacio, data_inici, data_fi, publicat

on {codi_gestor} referència GESTOR (codi_gestor)

on {codi_tipus} referència TIPUS (codi_tipus)

on {codi_tematica} referència TEMATICA (codi_tematica)

Gesevent. Gestió d'events.

on {codi_pais} referència PAIS (codi_pais)
on {codi_carrer} referència CARRER (codi_carrer)
on {codi_ubicacio} referència UBICACIO (codi_ubicacio)

TIPUS

codi_tipus, desc_tipus

TEMATICA

codi_tematica, desc_tematica

INFORMACIO

codi_informacio, codi_event, titol_informacio, desc_informacio
on {codi_event} referència EVENT (codi_event)

FORUM

codi_forum, codi_event, nom_forum, descripcio_forum
on {codi_event} referència EVENT (codi_event)

MISSATGE

codi_missatge, codi_forum, titol_missatge, desc_missatge, codi_usuari,
codi_missatge_pare, codi_missatge_fill
on {codi_forum} referència FORUM (codi_forum)
on {codi_usuari} referència PERSONA (codi_persona)
on {codi_missatge_pare} referència MISSATGE (codi_missatge)
on {codi_missatge_fill} referència MISSATGE (codi_missatge)

UBICACIO (entitat dèbil de CARRER)

codi_ubicacio, codi_carrer, nom_ubicacio, desc_ubicacio, aforament_max
on {codi_carrer} referència CARRER (codi_carrer)

CARRER (entitat dèbil de PAIS)

codi_carrer, codi_pais, nom_carrer, num, desc_carrer
on {codi_pais} referència PAIS (codi_pais)

Gesevent. Gestió d'events.

PAIS

codi_pais, nom_pais, desc_pais

PERSONA

codi_persona, doc_identificatiu, nom, cognom1, cognom2, telf, email

ADMINISTRADOR (subclasse de PERSONA)

codi_persona, password, rol

USUARI (subclasse de PERSONA)

codi_persona, password, rol, actiu

ASSISTEIX

codi_persona, codi_event

on {codi_persona} referència a PERSONA (codi_persona)

on {codi_event} referència a EVENT (codi_event)

GESTOR (subclasse de PERSONA)

codi_persona, password, rol, data_inici_contracte, data_fi_contracte

PONENT (subclasse de PERSONA)

codi_persona, password, rol, salari_dia

PONENCIA

codi_persona, codi_event

on {codi_persona} referència PERSONA (codi_persona)

on {codi_event} referència EVENT (codi_event)

Es pot observar que s'han creat dos relacions més ASSISTEIX i PONENCIA que són les que transformen les interrelacions M:N a relacions amb els seus propis atributs.

En aquest model es té en compte les herències i relacions de les subclasses que hereten de la classe persona: *Gestor*, *Usuari*, *Ponent* i *Administrador* que també estan reflectides en el diagrama de classes de l'aplicació. Un cop s'ha de passar d'aquest model entitat-relació al model relacional pur de implementació en base de dades es desfà el concepte de herència dels usuaris implantant tots els atributs dels mateixos en la taula de *Persona*. Aquest fet queda reflectit en les sentències SQL¹⁹ de creació de la base de dades les quals es poden trobar a l'annex d'aquest memòria.

¹⁹ SQL o llenguatge de consulta estructurat és un llenguatge declaratiu per l'accés a base de dades relacionals.

6. Arquitectura de l'aplicació

6.1. Introducció.

En aquest capítol es pretén donar una visió general de la arquitectura utilitzada en les diferents parts de l'aplicació. Aquesta es descomposarà en les diferents capes que hi intervenen així com es realitza una breu explicació sobre els patrons que s'aporten junt amb les arquitectures descrites. No només es dona una visió de la arquitectura del sistema implantant en aquest treball si no també per poder fer-se servir com una referència futura per treballs similars o per una posterior extensió d'aquest mateix.

L'arquitectura general de l'aplicació es basa en la plataforma empresarial J2EE de la que s'aprofita tant per fer la realització de la part client com la part servidora. Les diferents arquitectures, patrons i tecnologies d'aquestes dues capes generals es descriuen en els següents punts d'aquest capítol.

J2EE és una plataforma que permet el desenvolupament i execució de aplicacions en N capes distribuïdes²⁰, en aquest treball el sistema consta de tres capes, model-vista-controlador els qual estan dividits a la vegada en tres tecnologies diferents; per la part de control s'utilitza el bastiment Struts en la seva versió 2.2, per la capa vista s'utilitza tecnologia basada en JSP aprofitant les funcionalitats que també presenta el bastiment Struts2 i el bastiment Tiles²¹, i per la part de model s'utilitza la tecnologia de components de negoci EJB emprant JPA (amb beans de entitat, *Entity beans*) per la abstracció entre el model de objectes i la capa relacional.

6.2. Servidor d'aplicacions.

Per realitzar l'execució d'aquestes tecnologies s'ha de disposar de un servidor d'aplicacions. En aquest cas un servidor d'aplicacions és una capa de programari que permet l'execució de les diferents tecnologies de la plataforma J2EE.

El servidor d'aplicacions, a més, és compost d'una sèrie de capes o contenidors que permeten l'execució de cadascuna de les capes de forma independent i amb seguretat. A part de conformar un ecosistema que permet l'execució, en cas que fos necessari, de més d'una aplicació diferent amb tota seguretat ja que és ell mateix que proveeix d'independència entre aplicacions i capes.

En el cas del sistema Gesvent el servidor d'aplicacions escollit és JBoss²² el qual disposa de dos contenidors:

- Contenedor web: el contenedor web és l'encarregat de gestionar les peticions via

²⁰ Definició extreta de: <http://es.wikipedia.org/wiki/J2EE> [data de consulta: 12/06/2011]

²¹ <http://tiles.apache.org/>

²² <http://www.jboss.org/jbossas>

protocol [http](#)²³ (una explicació a fons d'aquest protocol queda fora de l'abast d'aquesta memòria, en cas de més interès preguem recorrim a la referència del peu de pàgina) generades per la pàgina web (part client) que és servida a l'usuari; capa de vista.

Aquesta part client presentada a l'usuari és realitzada amb pàgines web dinàmiques generades pel propi contenidor web gràcies a la tecnologia de JSP. Aquesta tecnologia es compon de una sèrie de etiquetes que són interpretades dinàmicament pel servidor, el qual les tradueix en etiquetes estàtiques de HTML²⁴. Per aquest projecte s'han utilitzat les etiquetes estàndard del fabricant principal de tecnologia de J2EE, les JSTL²⁵ a més de les proporcionades pel bastiment Struts2 (Struts2 tags²⁶).

A més aquest contenidor web és l'encarregat de realització l'execució de la capa de control. La capa de control en tecnologia J2EE és realitzada amb l'especificació de Servlets²⁷ de J2EE la qual és implementada en l'aplicació de gestió d'events amb el bastiment Struts2 que permet amb molta facilitat, estenent de molt poques classes, un ús del patró MVC (això es veurà en el següent punt de la memòria).

- Contenedor EJB: és l'encarregat de l'execució dels components de negoci EJB, a més en l'aplicació Gesevent s'encarrega de la gestió de transaccions, persistència gestionada pel contenidor (CMP).

L'aplicació de gestió d'events disposa d'una capa realitzada amb components *Entity Beans* que són els encarregats de realitzar l'abstracció entre el model relacionals de la base de dades i el model de orientació a objectes que presenta la plataforma Java. Aquesta abstracció permet al desenvolupador desentendre's del model relacional que hi ha per sota i sempre treballar amb orientació a objectes.

Aquesta forma de treballar permet una reutilització del codi ja que sempre es treballa amb el mateix àmbit, objectes, sense haver de dependre de quina tecnologia de emmagatzematges persistent hi ha per sota. A la vegada el fet d'utilitzar les capacitats dels contenidor per realitzar la persistència permet que l'optimització de les transaccions recaigui en el servidor d'aplicacions i allibera al programador de la responsabilitats de fer-ho.

²³ És el protocol de transferència d'hipertext, és el protocol usat principalment en el de les pàgines web. <http://www.w3.org/>

²⁴ Llenguatge de marcat de hipertext. És un llenguatge d'etiquetes que interpreten els diferents navegadors web del mercat i permeten formatejar text i dades per la seva visualització (i interacció) amb l'usuari. Per més informació: <http://www.w3.org/html/>

²⁵ <http://www.oracle.com/technetwork/java/index-jsp-135995.html>

²⁶ <http://struts.apache.org/2.0.14/docs/tag-reference.html>

²⁷ Un Servlet és un objecte que s'executa en un servidor d'aplicacions web que cobreixi les especificacions J2EE i que permet implementar una lògica dinàmica en el servidor.

6.3. Model de capes MVC.

L'aplicació segueix el patró model-vista-controlador (Model-View-Controller, MVC)²⁸. Aquest és un patró general d'arquitectura de programari que separa l'aplicació en tres parts o capes: la capa de vista que s'encarrega de la interfície amb el client, la capa controladora (o de control) que s'encarrega de redirigir les crides del client cap a la lògica de negoci (i viceversa) i la capa de model que és la que conté la lògica de negoci en si, a més de les dades relatives a aquesta lògica.

Es pot veure un exemple gràfic de la interacció entre capes²⁹:

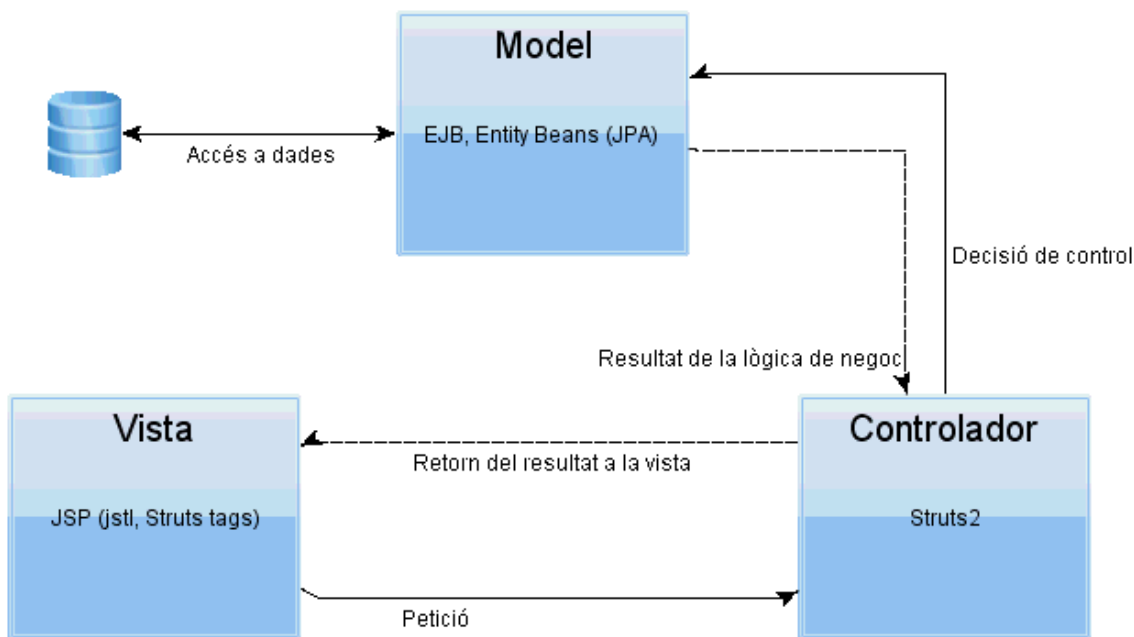


Fig 24 Esquema model-vista-controlador

Aquest patró aconsegueix independència entre les capes de lògica, control i vista de manera que poden ser aprofitades de forma autònoma facilitant així el seu manteniment i la seva reutilització. Per exemple la capa de model podria ser aprofitada per més d'un client amb tecnologies diferents. Aquest hauria de fer només les crides a aquesta capa amb els paràmetres correctes i, d'altra banda, la lògica aplicada serà la que ja es disposaria en la

²⁸<http://java.sun.com/blueprints/patterns/MVC-detailed.html>

²⁹ Imatge creada a partir de la referència obtinguda de la web: <http://www.oracle.com/technetwork/articles/javase/mvc-136693.html>,
imatge: **Figure1. A Common MVC Implementation** [data de consulta: 10/03/2011]

capa model reutilitzada.

Com s'ha comentat, en l'aplicació de gestió d'events s'utilitza la plataforma J2EE la qual ens proporciona diferents tecnologies des de on es pot aplicar fàcilment el patró MVC. Per la vista s'utilitza Java Server Pages (JSP)³⁰, per la capa controladora es fa servir el bastiment Struts³¹ i per la capa de model els Enterprise Java Beans (EJB)³². Aquesta última és la que accedeix a les dades utilitzant *Entity Beans* amb un model de gestió de la transacció basada en el contenidor (CMP), del servidor d'aplicacions.

Aquestes tecnologies esmentades es comuniquen entre elles de la forma que es pot veure en el gràfic següent. A més en aquest gràfic es pot apreciar exactament com estan distribuïdes aquestes tecnologies dintre dels diferents contenidors de que disposa el servidor d'aplicacions.

³⁰<http://download.oracle.com/javaee/5/tutorial/doc/bnagx.html>

³¹<http://struts.apache.org/download.cgi#struts2211>

³²<http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

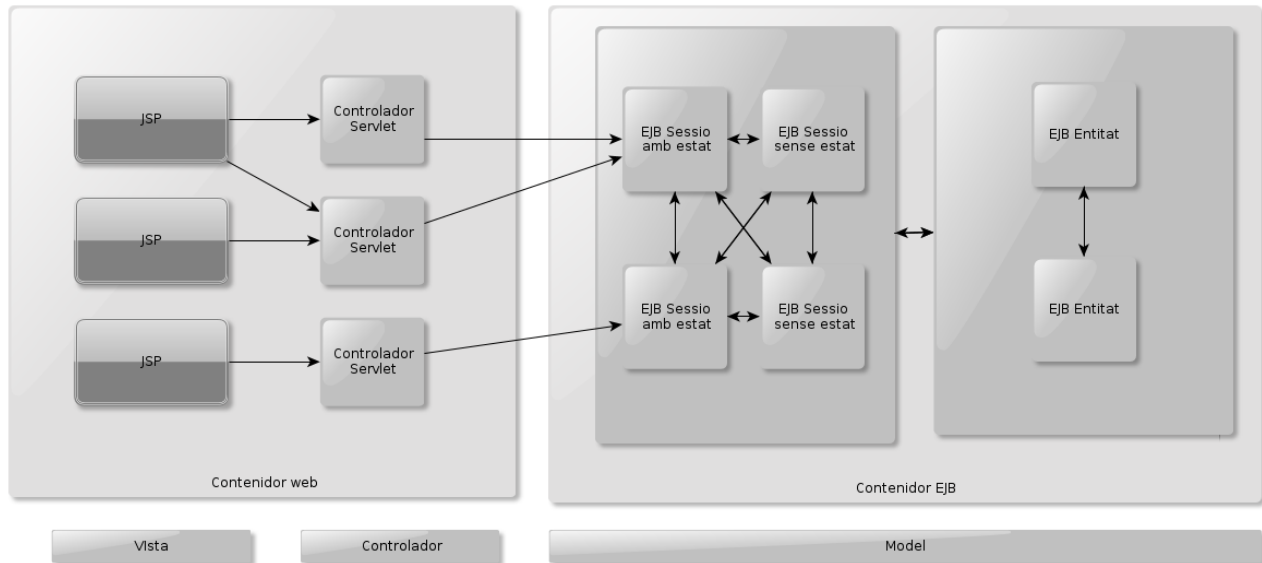


Fig 25 MVC, distribuït per tecnologies i contenidors

La capa de vista envia les dades i la petició de l'usuari a la capa de control (Servlets que en aquest cas és implementat gràcies al bastiment Struts). La capa de control decideix, amb les dades que tingui, cap a quin objecte de la capa de model ha de redirigir la petició. Un cop les dades es trobin a la capa de model aquestes són processades per la lògica que aquí hi resideix (en el cas de l'aplicació Gesevent són EJB sense estat); aquests EJB sense estat no mantenen un diàleg amb el client si no que són invocats per ell i un cop retornen la resposta es deixen a disposició de ser utilitzats per qualsevol crida de qualsevol altre client. Finalment i en cas necessari s'accedeix a la base de dades per recuperar o per fer persistents les dades treballades (EJB de entitat).

6.4. Patró objectes façana (Session Facade).

A part del patró MVC utilitzat, aquest es combina amb un altre molt important anomenat, patró façana. Aquest patró s'utilitza amb els components EJB fent que es faci ús d'un únic objecte EJB (o pocs objectes EJB) de sessió amb la lògica que es presentarà al client de forma que faci de façana entre les crides d'aquest i la resta de components de negoci de l'aplicació.

Això aconsegueix reduir l'acoblament entre classes del client i les classes del model. El fet de reduir l'acoblament millora que les classes de client depenguin, el menys possible, de les del model i s'afavoreixi, per tant, la reutilització i el manteniment del codi.

Així mateix, es redueixen el nombre de crides entre client i EJB de model de manera que es millora el rendiment de l'aplicació. S'ha de pensar que les crides a EJB es fan mitjançant estructures remotes el que fa que cada cop que hi ha una crida, el servidor d'aplicacions ha de generar l'Skeleton i l'Stub corresponent per la comunicació entre les dos parts.

Per últim, es redueixen les interfícies a les que el client pot cridar de forma que l'accés d'aquests a la lògica de negoci és molt uniforme. El diagrama de classes d'aquest patró es representa amb el diagrama següent³³:

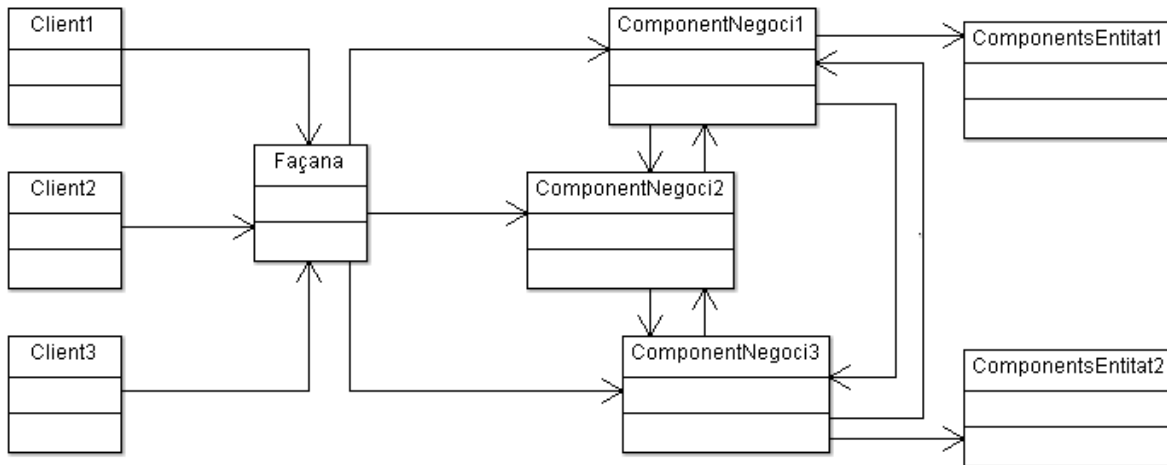


Fig 26 Patró façana (Session facade)

6.5. Struts2

Struts2 és un bastiment de desenvolupament d'aplicacions que es basa en el patró MVC vist en l'apartat 6.3. El bastiment es divideix en si en les tres capes comentades abans:

- Model: en el cas de Struts2 la capa de model correspon a les classes Action les quals contenen la lògica de negoci de l'aplicació. En el cas que ens ocupa aquests accions deriven en peticions als components de negoci EJB façana que són els que per el cas de Gesevent s'encarreguen d'executar la lògica de negoci i accedir a les dades de l'aplicació.

³³ Imatge creada a partir de la referència extreta de: <http://java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html> , imatge Figure 8.15 Session Facade class diagram. [data de consulta: 13/06/2011]

- Vista: esta composta de JSP amb la utilització de etiquetes que proporciona el propi bastiment de Struts2 (entre d'altres), amb això aconseguix mostrar a l'usuari els diferents formularis així com els resultats que se'n deriven de l'ús de l'aplicació.
- Controlador: fa d'intermediari entre la vista i el model en el bastiment Struts aquest comportament és definit per el filtre *FilterDispatcher*, el qual redirigeix les peticions de la vista (JSP) a la capa de model de Struts2 corresponent (Action).

A més Struts2 defineix unes classes intermèdies anomenades *Interceptors* per les quals passen totes les peticions que redirigeix el *FilterDispatcher* cap als Action de manera que es poden controlar quines dades i de quina manera arriben a les classes Action definides. Aquests es són cridats per un ordre definit al fitxer de configuració de manera que ho fan en forma de pila, passant a executar-se un darrera de l'altre.

El funcionament general de les peticions dintre del bastiment Struts2 és pot veure més amb el gràfic següent, en el qual també s'ha afegit com la classe Action a la que arriba la petició de la vista fa la crida al EJB de façana per que accedeixi a les dades:

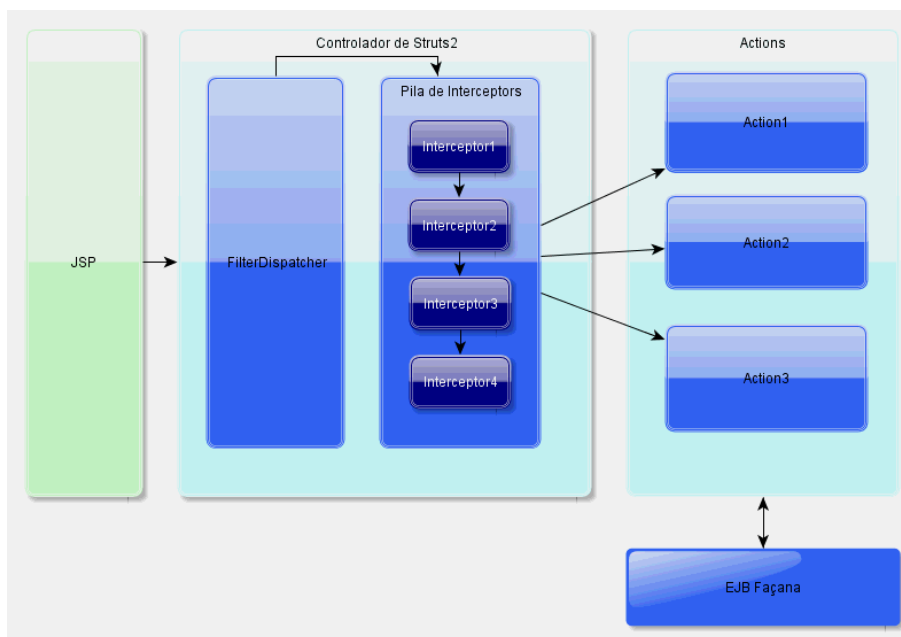


Fig 27Struts2 amb la crida a EJB Façana

6.6. Tiles.

Tiles³⁴ és un bastiment que permet una fàcil i ràpida implementació d'interfícies visuals utilitzant JSP. Els concepte en que es basa és en l'ús de plantilles genèriques en les quals

³⁴ Bastiment Tiles, el qual permet una fàcil implementació i manteniment de la capa de vista de l'aplicació.
<http://tiles.apache.org/>

s'insereixen fragments de pàgines compostes de JSP o HTML pur per tal de forma una única pàgina única.

La configuració del bastiment és realitza mitjançant un fitxer XML de propietats on s'indiquen les plantilles que es volen fer servir i com el bastiment ha de compondre la pàgina a partir de les plantilles i les JSP que es volen incorporar a les mateixes.

El bastiment Tiles es pot definir dintre de Struts2 com una extensió del propi bastiment de manera que aquest no crida directament a les JSP si no que ho fa a les definicions fetes al fitxer de configuració de la capa de treball Tiles, el qual serà l'encarregat de muntar i servir la capa de vista a l'usuari.

Aquest bastiment permet un alt grau de reusabilitat de la part de vista de l'aplicació així com una major facilitat en el manteniment ja que les modificacions es realitzen sobre les plantilles aconseguint i no s'han de modificar les pàgines JSP una a una.

6.7. EJB (versió 3).

Els components de negoci EJB formen part de la especificació de la plataforma J2EE. La versió utilitzada en aquest treball és la última definida pel *Java Community Process* el qual defineix els estàndards que ha de complir la plataforma Java en general. En el cas dels EJB és un API que ha de implementar tots servidors d'aplicacions que vulgui estandarditzar el seu ús amb la plataforma J2EE.

Els EJB són objectes que són cridats mitjançant interfícies remotes de manera que permeten molt fàcilment la seva distribució. A més entre els propis components EJB també es criden entre si mitjançant aquestes interfícies el qual per una banda, com s'ha dit, permet que siguin fàcilment distribuïbles però per un altre limiten el rendiment de l'aplicació al haver de realitzar tot el procés de serialització dels objectes i el seu pas per socket.

Es pot veure el procés de crida d'un objecte remot en el següent gràfic on un client qualsevol (podria ser qualsevol tipus de classe fins i tot, com s'ha mencionat, un EJB) crida a l'interfície de un component de negoci (EJB) per tal de que executi algun dels seus mètodes.

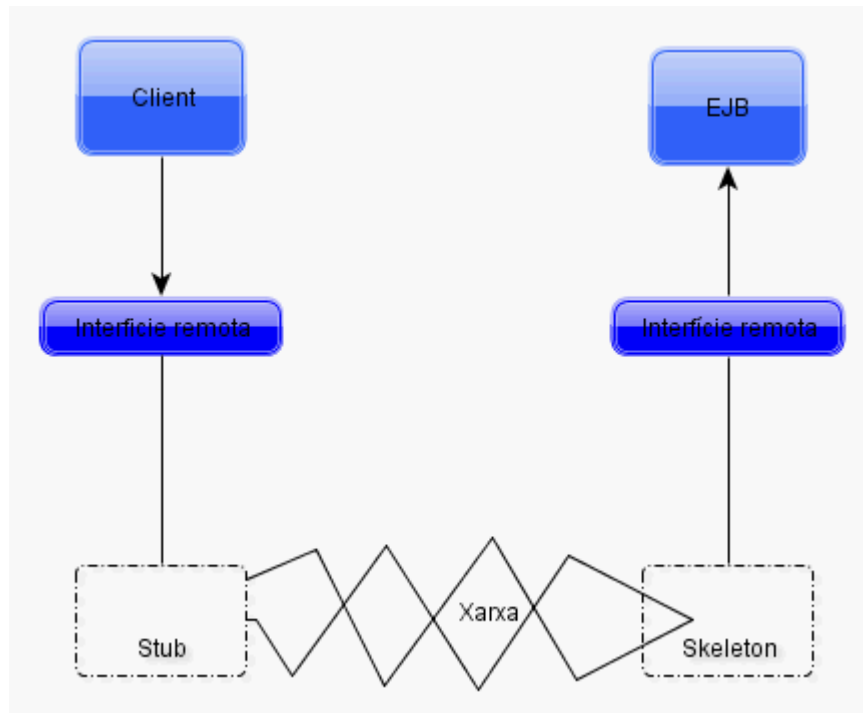


Fig 28 Crida a un objecte remot EJB³⁵

De EJB n'hi ha de tres tipus essencialment:

- **Beans de sessió amb estat (Stateful beans):** són components de negoci que permeten un diàleg amb el client de manera que aquests emmagatzemant l'estat entre crida i crida d'un mateix client.
- **Beans de sessió sense estat (Stateless beans):** aquests components no mantenen cap diàleg amb el client. El client els crida, tornen el resultat i esperen una altre crida de un client que potser el mateix o no. No mantenen la sessió entre crides.
- **Beans d'entitat (Entity beans):** s'encarreguen de modelitzar el model de dades de l'aplicació. Permeten tenir una abstracció entre el model de emmagatzematge de dades i el model d'objectes de l'aplicació.

En l'aplicació Gesevent s'utilitzen els dos últims tipus descrits.

³⁵ Gràfic creat a partir de la referència extreta de: <http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Teor%C3%ADa+workshop+de+EJB+2.0>, [data de consulta: 17/06/2011]

7. Valoració econòmica.

En aquest treball de final de carrera no aplica parlar d'un cost de desenvolupament de l'aplicació, ja que ha sigut realitzat com una fase més dels estudis universitaris i el temps invertit en l'aprenentatge de les tecnologies utilitzades no es pot valorar econòmicament.

L'única valoració econòmica possible seria la del cost del programari utilitzat durant tot el treball de final de carrera. Aquests en la seva majoria ha constatat de programari de llicències gratuïtes les quals no tenen cap cost per al desenvolupador. No obstant com ja s'ha explicat als capítols introductoris i es veurà més endavant en l'annex del programari utilitzat s'ha acabat optant per realitzar la documentació final amb programari el qual té un cost per llicència.

Com s'ha explicat no es pot donar un cost econòmic del treball realitzat ja que no consta com a part de l'objectiu del mateix però s'ha de valorar que part del projecte ha necessitat de programari que té un cost en el mercat.

8. Conclusions.

Com a conclusió trobem molt positiva l'elecció de l'àrea del treball de final així com les tecnologies utilitzades. Ens ha permès aplicar el coneixement adquirit durant la nostra trajectòria universitària a la realització, des de zero, d'un projecte on la majoria de components a utilitzar eren desconeguts.

Ens hem adonat de la utilitat de l'ús i seguiment de la documentació que s'ha anat generant durant les primeres fases del projecte així com la realització de proves unitàries de les diferents tecnologies desconegudes de manera que fer l'aplicabilitat de les mateixes a l'etapa de desenvolupament ha sigut menys dramàtica.

La tecnologia de components EJB que tant interès ens despertava i que va ser el seu aprenentatge i ús el que ens va fer decantar-nos per aquest area de treball, ens ha sorprès gratament amb la seva facilitat d'utilització gràcies a les anotacions, així com a la seva potència.

Tot i així pensem que la majoria d'aplicacions actuals no necessiten aquesta tecnologia per poder portar a terme un producte de qualitat però si que pensem que per aplicacions de gran volum de transaccions entre diferents clients i servidors potser un punt clau i estratègic la seva elecció, ja que un cop vist les possibilitats creiem que te una gran capacitat i fàcil escalabilitat.

Per últim comentar que ens hem adonat d'una carència personal en el desenvolupament d'aquest tipus d'aplicacions: la interfície d'usuari. Tot i la simplicitat i potència del HTML i de les etiquetes que serveixen els diferents bastiments utilitzats, pensem que les interfícies creades no són prou satisfactòries tant a nivell de disseny, com a nivell d'usabilitat. És una crítica constructiva personal i de millora a la feina realitzada sense tampoc descuidar que l'esforç fet en la resta de parts del treball (documentació, planificació, desenvolupament, aprenentatge) ha sigut molt positiu i enriquidor, del qual estem molt satisfets.

9. Glossari.

Bastiment: estructura conceptual i de programari definida pensada per la utilització en desenvolupament de programari de forma que faciliti la tasca de creació i manteniment del mateix.

EJB: *Enterprise Java Beans*, és un API que prove de components al costat del servidor per la plataforma J2EE de Java.

Event: aconteixement, circumstància o succés que es realitza entre dos dates determinades i en el qual hi ha pot haver uns conferenciants o ponents, i assistents o participants.

Framework: entorn de treball o capa de treball és un conjunt de llibreries que faciliten el desenvolupament de certes parts de una aplicació proveint de estàndards i patrons a la mateixa.

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

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

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

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

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

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

Usuari: qualsevol persona que s'autentiqui contra el sistema i interaccioni amb ell.

10. Bibliografia.

Basham, B; Sierra, K; Bates, B. (2004) *Head First. Servlets & JSP*. Sebastopol (Estats Units): O'Really

Eckstein, Robert (2007). *JavaSE Application Design with MVC*. [en línia]. <http://www.oracle.com/technetwork/articles/javase/mvc-136693.html> [dia de consulta: 12/04/2011]

Epididata Consulting (2006). *Teoria Workshop EJB 2.0*. [en línia]. <http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Teor%C3%ADa+workshop+de+EJB+2.0> [dia de consulta: 17/06/2011]

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

Oracle Corporation (2010). *Enterprise JavaBeans Technology*. [en línia]. <http://www.oracle.com/technetwork/java/javaee/ejb/index.html> [dia de consulta: 04/04/2011]

Oracle Corporation (2010). *The Java EE 5 Tutorial*. [en línia]. <http://download.oracle.com/javaee/5/tutorial/doc/bnagx.html> [dia de consulta: 03/04/2011]

Oracle Corporation (2010). *Java EE Tutorials*. [en línia]. <http://www.oracle.com/technetwork/java/javaee/documentation/tutorials-137605.html>. [dia de consulta: 03/04/2011]

Rozanski, Uwe (2009). *Enterprise JavaBeans 3.0 con Eclipse y Jboss*. Barcelona (Espanya): Marcombo, Ediciones Técnicas.

Sun Microsystems, Inc. (2001-2002). *Core J2EE Patterns Session Facade*. [en línia]. <http://java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html> . [dia de consulta: 06/04/2011]

Sun Microsystems, Inc. (2000-2002). *Java BluePrints. Model-View-Controller*. [en línia]. <http://java.sun.com/blueprints/patterns/MVC-detailed.html>. [dia de consulta: 01/04/2011]

The Apache Software Foundation (2000-2011). *Struts*. [en línia].
Floren Llanos Ruiz

Gesevent. Gestió d'events.

<http://struts.apache.org/download.cgi#struts2211>. [dia de consulta: 03/04/2011]

The Apache Software Foundation (2000-2011). *Tiles*. [en línia]. <http://tiles.apache.org/>. [dia de consulta: 03/06/2011]

11. Annexos.

11.1 Interfícies d'usuari.

Es presenten unes captures de pantalla de l'aplicació per que serveixin a mode de exemple de l'estil utilitzat per capa de vista de la mateixa.

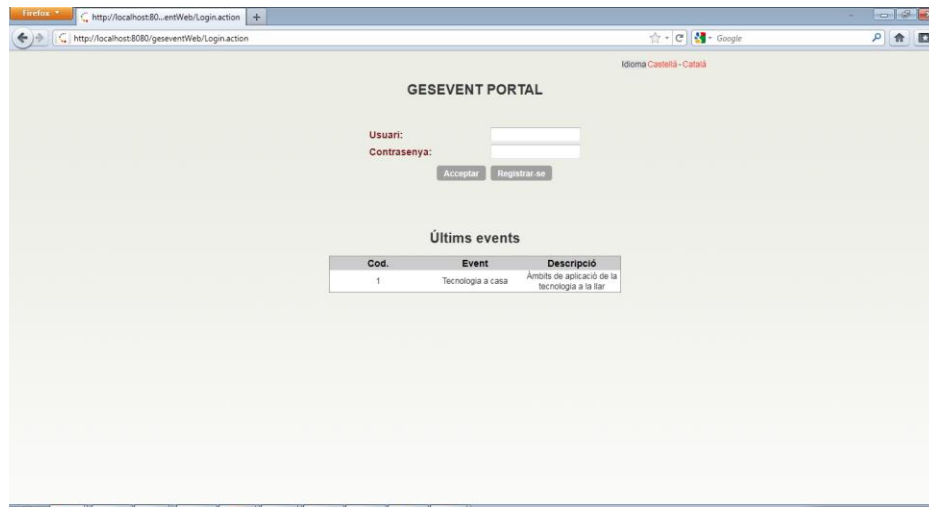


Fig 29 Pantalla de autenticació

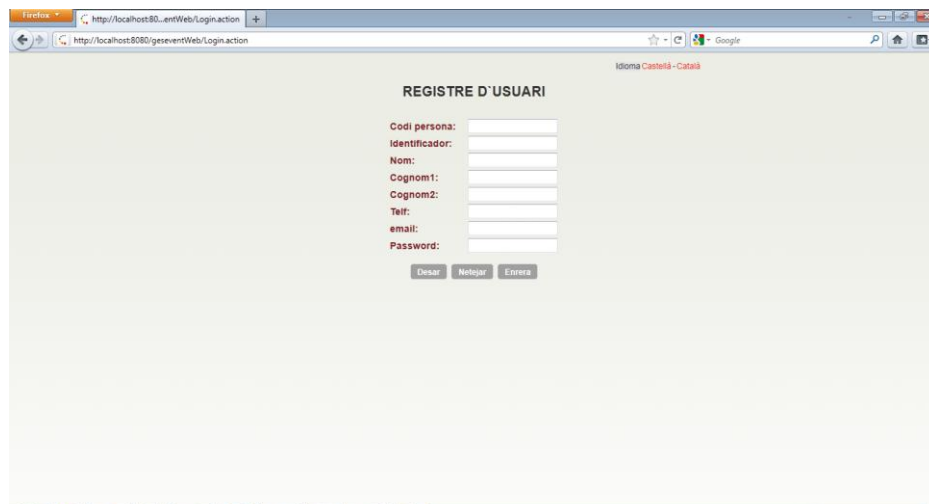


Fig 30 Pantalla de registre d'usuari

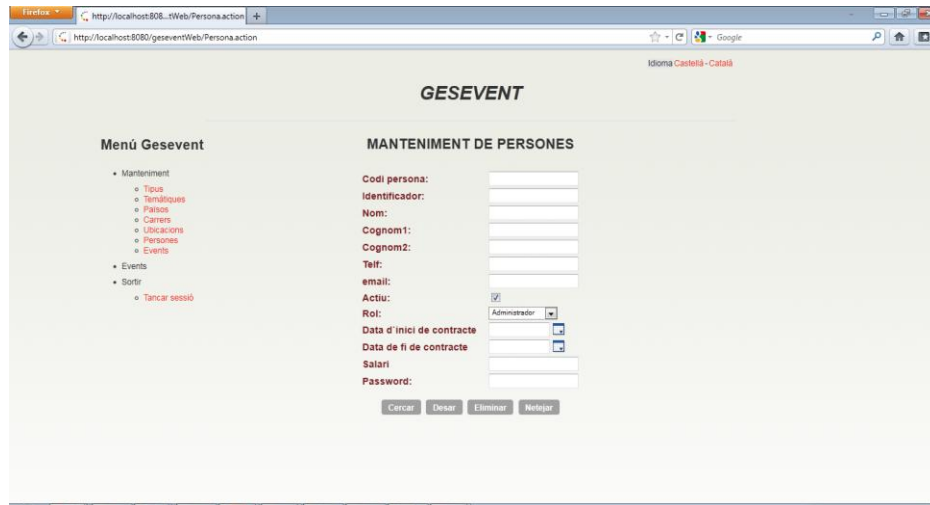


Fig 31 Menú del rol Administrador i pantalla de manteniment de persones

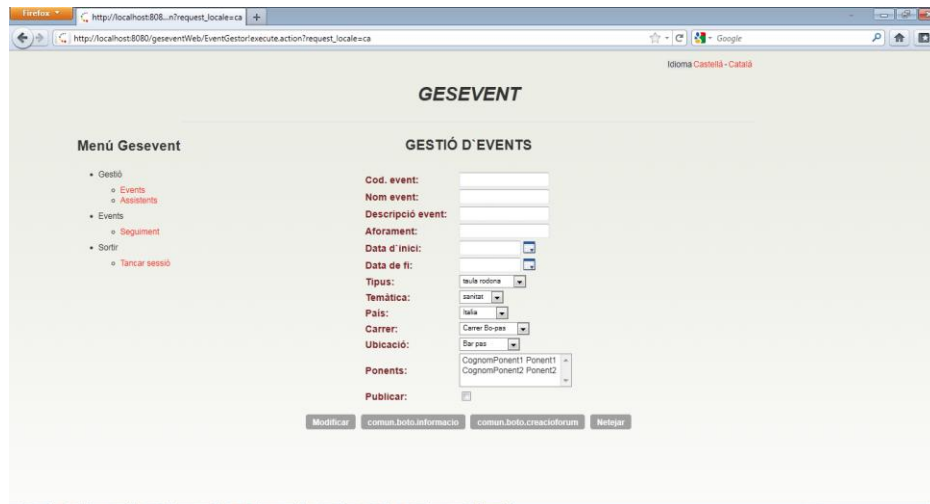


Fig 32 Menú del rol Gestor i pantalla de gestió d'events

Gesevent. Gestió d'events.



Fig 33 Pantalla de extracció de llistats d'assistents per el rol Gestor



Fig 34 Menú del rol Usuari i pantalla de inscripció a events

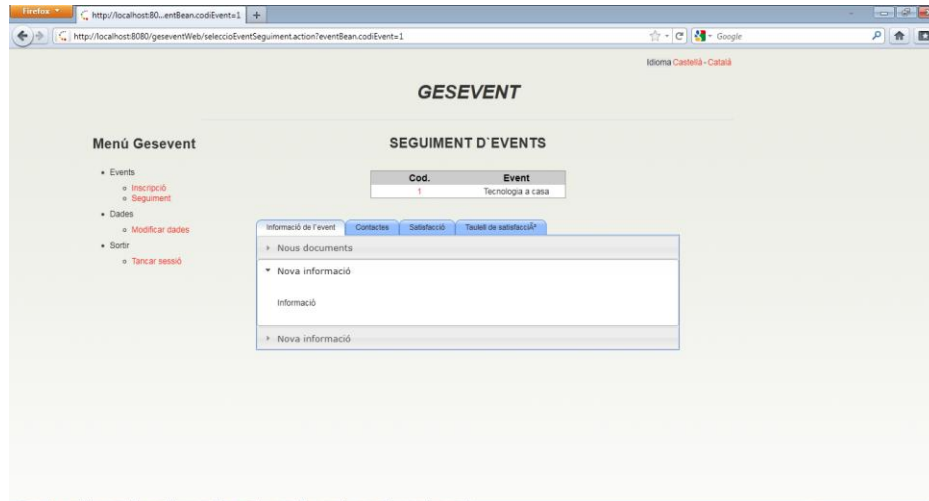


Fig 35 Pantalla de seguiment d'events pel rol de Usuari

11.2. Programari.

En aquest annex s'enumera el programari utilitzat per la realització d'aquest treball de manera que pugui ser utilitzat per valorar la utilitat del mateix així com fer-se servir com a referència per futurs projectes de característiques similars.

Com a sistemes operatius s'han utilitzat dos:

- Ubuntu 10.04 (Linux): sistema operatiu d'escriptori utilitzat per realitzar el desenvolupament de l'aplicació de gestió d'events. <http://www.ubuntu.com/>.
- Microsoft Windows 7: ha sigut el sistema operatiu d'escriptori utilitzat per elaborar la documentació i aquesta memòria, ja que s'ha realitzat amb l'entorn d'ofimàtica de Microsoft, només executable en sistemes operatius de la pròpia empresa. <http://www.microsoft.com/es/es/default.aspx>

Programari d'ofimàtica:

- OpenOffice→suite ofimàtica utilitzada per l'entrega dels documents de la avaluació continuada. <http://es.openoffice.org/>
- Microsoft Word→ editor de text utilitzat per la redacció d'aquesta memòria. <http://www.microsoft.com/es/es/default.aspx>

Microsoft PowerPoint→editor de diapositives utilitzat per la creació de la presentació d'aquest treball de final de carrera. <http://www.microsoft.com/es/es/default.aspx>

Entorns de desenvolupament i eines de suport:

- ArgoUML: eina de modelatge en UML. <http://argouml.tigris.org/>.

Gesevent. Gestió d'events.

- yEd Graph Editor: eina per la creació de gràfics. http://www.yworks.com/en/products_yed_about.html.
- Màquina virtual de Java: entorn de execució de llenguatge Java. <http://openjdk.java.net/>.
- Eclipse Helios: entorn de desenvolupament integrat on una de les plataformes que suporta és Java. <http://www.eclipse.org/>.
- KompoZer: entorn de desenvolupament per pàgines web HTML. <http://kompozer.net/>

Base de dades:

- PostgreSQL: base de dades relacional. <http://www.postgresql.org/>

Navegadors web:

- Firefox: navegador web de Mozilla. <http://www.mozilla-europe.org/es/>