Projecte fi de carrera

Disseny i implementació d'un sistema web de geolocalització i promoció d'events

Jordi Fernández Camps

Director: Dídac Barragán Pascual

Girona, setembre de 2008

Índex

I Descripció, origen i objectius del projecte **1**

1 Antecedents **5**
1.1 World Wide Web 5
1.1.1 Web 1.0 5
1.1.2 Xarxes socials a Internet 5
1.1.3 Web 2.0 6

2 Objectius **9**

II Cicle de vida **11**

3 Introducció **13**

4 Planificació de Sistemes d'Informació (PSI) **15**
4.1 Anàlisi de la necessitat del PSI 15
4.2 Identificació de l'abast del PSI 15

5 Estudi de Viabilitat del Sistema (EVS) **17**
5.1 Establir l'abast del sistema 17
5.1.1 Estudi de la sol·licitud 17
5.1.2 Identificació de l'abast del sistema 17
5.2 Definició de requisits del sistema 18
5.2.1 Identificació de directrius tècniques i de gestió 18
5.2.2 Identificació de requisits del sistema 18
5.3 Estudi d'alternatives de solució 19
5.3.1 Preselecció d'alternatives de solució 19
5.3.2 Descripció de les alternatives de solució 19
5.3.3 Estudi dels riscos 20
5.4 Selecció de la solució 20

6 Anàlisi del Sistema d'Informació (ASI) **23**
6.1 Definició del sistema 23
6.1.1 Determinar l'abast del sistema 23
6.1.2 Identificació de l'entorn tecnològic 23
6.1.3 Especificació d'estàndards i normes 24
6.1.4 Identificació dels usuaris participants i finals 24
6.2 Establiment de requisits 24
6.2.1 Obtenció de requisits 24
6.2.2 Especificació de casos d'ús 29

6.3 Identificació de subsistemes d'anàlisi 42
6.4 Identificació de classes associades a un cas d'ús 43
6.5 Descripció de la interacció d'objectes 44
6.6 Anàlisi de classes 58
6.6.1 Identificació de responsabilitats i atributs 58
6.7 Elaboració del model de dades 65
6.7.1 Elaboració del model conceptual de dades 65
6.8 Interfícies d'usuari 67
6.8.1 Especificació dels principis generals de la interfície 67
6.8.2 Especificació de formats individuals d'interfície 67

7 Disseny del Sistema d'Informació (DSI) 79
7.1 Definició de l'arquitectura del sistema 79
7.1.1 Definició de nivells d'arquitectura 79
7.1.2 Especificació d'estàndards i normes de disseny i construcció 79
7.1.3 Especificació de l'entorn tecnològic 81
7.1.4 Especificació de requisits d'operació i seguretat 82
7.2 Disseny de classes 82
7.2.1 Identificació de classes addicionals 82
7.3 Disseny físic de dades 84
7.3.1 Disseny del model físic de les dades 84
7.4 Especificació de camins d'accés a les dades 84
7.5 Optimització del model físic de dades 90
7.6 Especificacions de construcció 90
7.6.1 Especificacions de l'entorn de construcció 90
7.6.2 Definició de components i subsistemes de construcció 96
7.7 Especificació del pla de proves 96
7.7.1 Especificació tècnica de nivells de prova 96

8 Construcció del Sistema d'Informació (CSI) 97
8.1 Preparació de l'entorn de generació i construcció 97
8.1.1 Implantació de la base de dades física o fitxers 97
8.1.2 Preparació de l'entorn de construcció 97
8.2 Generació del codi dels components i procediments 103
8.2.1 Generació del codi dels components 103
8.3 Execució de les proves unitàries 112
8.4 Elaboració dels manuals d'usuari 113

9 Implantació i Acceptació del Sistema (IAS) 115
9.1 Incorporació del sistema a l'entorn final 115
9.1.1 Realització de la instal·lació 115
9.2 Realització de proves d'implantació 117
9.3 Realització de proves d'acceptació del sistema 117

III Conclusions i futur 119

10 Conclusions 121

11 Futur 123

11.1 Millores i ampliacions 123

Bibliografia (Llibres i artícles) **125**

Bibliografia (Altres fonts) **127**

Índex de figures

1.1 Web 1.0 . . . 6
1.2 Web 2.0 . . . 7

5.1 Diagrama de Gantt . . . 21

6.1 Cas d'ús general . . . 25
6.2 Casos d'ús de l'actor Usuari . . . 26
6.3 Casos d'ús de l'actor Usuari anònim . . . 27
6.4 Casos d'ús de l'actor Usuari registrat . . . 28
6.5 Casos d'ús de l'actor Administrador . . . 29
6.6 Casos d'ús de l'actor Sistema . . . 30
6.7 Subsistemes d'anàlisi . . . 43
6.8 Diagrama de seqüència: Canviar idioma . . . 44
6.9 Diagrama de seqüència: Cercar per mapa . . . 45
6.10 Diagrama de seqüència: Cercar per text . . . 46
6.11 Diagrama de seqüència: Cercar per RSS . . . 46
6.12 Diagrama de seqüència: Votar event . . . 47
6.13 Diagrama de seqüència: Crear compte . . . 48
6.14 Diagrama de seqüència: Identificar-se . . . 48
6.15 Diagrama de seqüència: Restablir contrasenya . . . 49
6.16 Diagrama de seqüència: Desconnectar-se . . . 49
6.17 Diagrama de seqüència: Enviar event . . . 50
6.18 Diagrama de seqüència: Modificar event . . . 51
6.19 Diagrama de seqüència: Modificar preferències . . . 51
6.20 Diagrama de seqüència: Afegir comentari . . . 52
6.21 Diagrama de seqüència: Afegir amic . . . 52
6.22 Diagrama de seqüència: Eliminar amic . . . 53
6.23 Diagrama de seqüència: Afegir event a preferits . . . 53
6.24 Diagrama de seqüència: Eliminar event de preferits . . . 54
6.25 Diagrama de seqüència: Xatejar . . . 54
6.26 Diagrama de seqüència: Afegir categoria . . . 55
6.27 Diagrama de seqüència: Eliminar categoria . . . 55
6.28 Diagrama de seqüència: Modificar categoria . . . 56
6.29 Diagrama de seqüència: Banejar usuari . . . 56
6.30 Diagrama de seqüència: Treure baneig a usuari . . . 57
6.31 Diagrama de seqüència: Enviar notificacions . . . 57
6.32 Diagrama de seqüència: Recalcular karma . . . 58
6.33 Diagrama de seqüència: Generar núvol d'etiquetes . . . 59
6.34 Diagrama de classes inicial . . . 65
6.35 Diagrama entitat relació . . . 66
6.36 Estructura de les pantalles . . . 68
6.37 Pantalla principal . . . 69

6.38 Pantalla d'exploració 70
6.39 Pantalla enviament d'events 72
6.40 Pantalla de detall d'event 73
6.41 Pantalla generació de ruta 74
6.42 Pantalla de perfil d'usuari 75
6.43 Pantalla de preferències d'usuari 76
6.44 Pantalla cerca d'events 77

7.1 Arquitectura del sistema 80
7.2 Diagrama de classes 85
7.3 Esquema del framework Pylons 92

8.1 Trac: exploració del repositori 102
8.2 Trac: canvis entre revisions 103
8.3 Núvol d'etiquetes 110
8.4 Wiki: manual d'usuari 114

Índex de taules

7.1 Taula avatars 84
7.2 Taula bans 85
7.3 Taula categories 86
7.4 Taula chats 86
7.5 Taula comments 86
7.6 Taula event_markers 86
7.7 Taula event_tags 87
7.8 Taula event_votes 87
7.9 Taula events 87
7.10 Taula favorites 88
7.11 Taula friends 88
7.12 Taula languages 88
7.13 Taula notifications 88
7.14 Taula tags 89
7.15 Taula user_markers 89
7.16 Taula users 89

Índex de llistats

8.1 Instal·lació de Pylons 97
8.2 Configuració de Nginx 98
8.3 Tasques d'actualització dels índexs de Sphinx 99
8.4 Configuració de Sphinx 99
8.5 Instal·lació i configuració d'Apache + Subverison 101
8.6 Càrrega de l'API de Google Maps 104
8.7 Cadena de text dins la funció gettext 105
8.8 Entrada d'un fitxer .pot 105
8.9 Transformació de direccions IP 105
8.10 Implementació de la fórmula de Haversine 106
8.11 Algorisme de la distància euclidiana 108
8.12 Algorisme del coeficient de correlació de Pearson 109
8.13 Execució de les proves unitàries 112
8.14 Resultat proves unitàries 112
8.15 Resultat proves unitàries 112
9.1 Instal·lació de l'entorn virtual 115
9.2 Instal·lació del programari 115
9.3 Configuració de bash 116
9.4 Creació d'un egg del projecte 116
9.5 Instal·lació de l'aplicació 117

Part I

Descripció, origen i objectius del projecte

En aquesta primera part s'introduirà el context del web actual i es definiran els conceptes fonamentals per situar el lector en el marc principal del projecte. Un cop introduït tot això es passarà a descriure els objectius del projecte.

Els capítols en que consta aquesta part són:

- **Antecedents**. Es parlarà del *web* en general, comentant la seva evolució des dels seus inicis fins al que actualment es coneix com a *web 2.0*. També es farà una introducció a les xarxes socials i el paper que juguen en l'actualitat.
- **Objectius**. Es descriuran els objectius del projecte, així com les motivacions i el mètode que s'utilitzarà per portar-lo a terme.

1. Antecedents

La finalitat d'aquest capítol és exposar el rerefons i context del web i l'estat actual en que es troba per entendre millor les motivacions i els objectius que es pretenen en aquest projecte.

1.1 World Wide Web

Ha passat molt de temps des que, al 1989, Tim Berners-Lee va crear el **WWW** (acrònim en anglès de **World Wide Web**, *Teranyina d'abast mundial*) o simplement **web**. Des de llavors moltes persones comparteixen coneixement a través d'una extensa xarxa d'ordinadors interconnectats des de diversos llocs del món.

1.1.1 Web 1.0

Els primers anys poques persones tenien accés al contingut de la xarxa; documents generalment confidencials per les seves característiques d'investigació o seguretat nacional. Amb el pas del temps es va observar el gran potencial que podia tenir aquest invent, va començar un gran intercanvi d'informació i amb ell la necessitat de compartir aquesta informació d'una manera senzilla.

L'evolució que ha tingut el web ha sigut contínua. Primer van arribar les imatges per acompanyar els textos, van aparèixer les primeres animacions, les primeres eines interactives i l'evolució va continuar.

Durant aquests anys el web ha sigut sempre un lloc informatiu, un lloc de lectura on poques persones escrivien i la resta ens dedicàvem a llegir. Segons molts autors això és el que podria anomenar-se el **Web 1.0**.

Amb l'arribada de les companyies .COM al 2001 es va evolucionar de les pàgines HTML estàtiques al següent nivell on la major part del contingut era generat dinàmicament a través de diversos sistemes i bases de dades. A més es va tenir en compte factors importants com l'estètica i la funcionalitat.

1.1.2 Xarxes socials a Internet

Una xarxa social és una estructura social que es pot representar en forma d'un o varis grafs en el qual els nodes representen individus i les arestes relacions entre ells. Les relacions poden ser de diferents tipus, com intercanvis financers, amistats, relacions sexuals o rutes aèries.

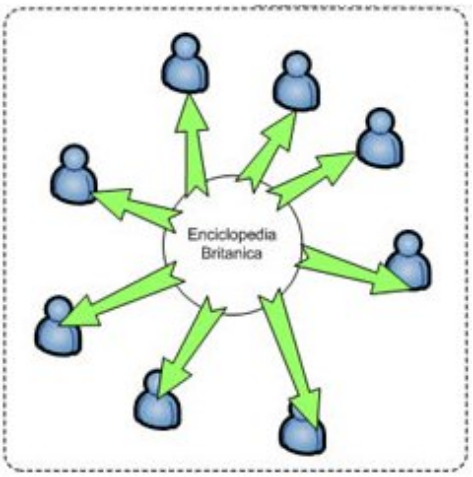

Figura 1.1: Web 1.0

Les xarxes socials a Internet s'han guanyat un lloc de manera vertiginosa convertint-se en prometedors negocis per empreses i sobretot, en llocs per establir relacions socials permetent interactuar amb altres persones encara que no les coneguem.

Aquestes xarxes socials es basen en la teoria dels sis graus. Sis graus de separació és la teoria que qualsevol pot estar connectat a qualsevol altre persona del planeta a través d'una cadena de coneguts que no té més de quatre intermediaris. La teoria va ser inicialment proposada el 1929 per l'escriptor hongarès Frigyes Karinthy en una curta història anomenada Chains. El concepte està basat en la idea que el nombre de coneguts creix exponencialment amb el nombre d'enllaços de la cadena, i només un petit nombre d'enllaços és necessari perquè el conjunt de coneguts es converteixi en la població humana sencera.

Segons aquesta teoria, cada persona coneix de mitja, entre amics, familiars i companys de feina o escola, a unes 100 persones. Si cadascú d'aquests amics o coneguts propers es relaciona amb unes altres 100 persones, qualsevol individu pot passar un encàrrec a 10.000 persones més només demanant a un amic que passi el missatge als seus amics.

Si aquests 10.000 coneixen a uns altres 100, la xarxa ja s'ampliaria a 1.000.000 de persones connectades en un tercer nivell, a 100.000.000 en un quart nivell, a 10.000.000.000 en un cinquè nivell i a 1.000.000.000.000 en un sisè nivell. En sis passos, i amb les tecnologies disponibles, es podria enviar un missatge a qualsevol persona del planeta.

1.1.3 **Web 2.0**

El terme **Web 2.0** va ser esmentat per Tim O'Reilly l'any 2004 per referir-se a una segona generació de Web basada en comunitats d'usuaris i una gama especial de serveis, com les xarxes socials, els wikis... que fomenten la col·laboració i l'intercanvi àgil d'informació entre els usuaris.

En general, quan mencionem el terme Web 2.0 ens referim a un sèrie d'aplicacions i pàgines d'Internet que utilitzen la intel·ligència col·lectiva per proporcionar serveis interactius a

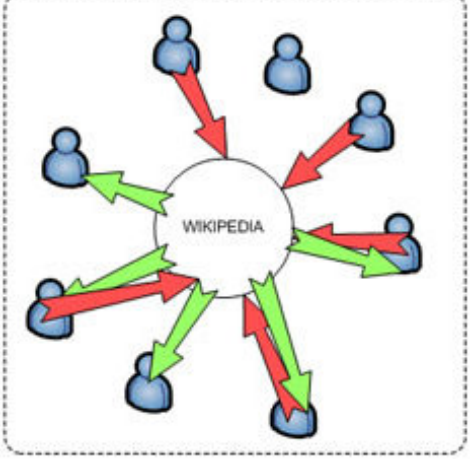

Figura 1.2: Web 2.0

una xarxa donant a l'usuari el control de les seves dades.

Així doncs, podem entendre com 2.0 totes aquelles utilitats i serveis d'Internet que s'alimenten d'una base de dades, la qual pot ser modificada pels usuaris del servei, ja sigui en el seu contingut (afegint, modificant o eliminant informació o associant dades a la informació existent), com en la forma de presentar-los, o en el contingut i forma simultàniament.

2. Objectius

Degut a la falta d'informació, de temps, no saber a on buscar... moltes vegades no ens assabentem, o ho fem massa tard, d'events als que ens hauria agradat assistir, com podrien ser concerts, conferències, activitats esportives, etc.

L'objectiu d'aquest projecte serà aprofitar les capacitats de les xarxes socials per crear un lloc web que permeti enviar i geolocalitzar[1] events que podran ser revisats i promoguts pels usuaris de forma que es pugui suplir aquesta mancança.

La solució implementada haurà de proporcionar les següents funcionalitats:

- Enviament d'events: Permetrà afegir les dades principals d'un event i geolocalitzar-lo en el mapa.
- Organització de la informació: Es disposarà de categories i metacategories[2] per agrupar els events, a més d'un sistema d'etiquetes que facilitarà les cerques en el contingut del web.
- Exploració dels events existents: Mitjançant el mapa es podrà veure les dades de qualsevol event.
- Sistema de votació: Atorgarà la capacitat per poder decidir quina informació és més rellevant.
- Agenda personal: Servirà per registrar events i d'aquesta manera poder rebre notificacions que informin de canvis o simplement que serveixin com a recordatori.
- Comunicació entre usuaris: Es realitzarà a través de comentaris al peu dels events i/o d'un xat intern.
- Sindicació web: Distribuirà el contingut utilitzant l'estàndard RSS.
- Disponibilitat d'una API simple: Permetrà l'accés a certa informació des d'aplicacions externes.

[1] Identificar la localització geogràfica real d'un ordinador connectat a Internet, un dispositiu mòbil, un usuari d'una pàgina web o qualsevol altre element amb una localització física.

[2] Categoria que agrupa un conjunt de subcategories

Part II

Cicle de vida

3. Introducció

En l'elaboració d'aquest projecte s'utilitza la metodologia Mètrica en la versió 3. És l'última i més recent metodologia de planificació, desenvolupament i manteniment dels sistemes d'informació proporcionada pel Ministeri d'Administracions Públiques (MAP). És una metodologia àmpliament estesa, reconeguda i comprensible per la majoria d'analistes.

Aquest instrument proporciona una sistematització de les activitats que donen suport al cicle de vida del programari dins del marc que permet obtenir els següents objectius:

- Proporcionar i definir Sistemes d'Informació que ajuden a aconseguir les finalitats de l'Organització mitjançant la definició d'un marc estratègic pel desenvolupament dels mateixos.
- Dotar a l'Organització de productes de programari que satisfacin les necessitats dels usuaris donant una major importància a l'anàlisi de requisits.
- Millorar la productivitat dels departaments de Sistemes i Tecnologies de la Informació i les Comunicacions, permeten millorar la capacitat d'acceptació als canvis i tenint en compte la reutilització.
- Facilitar la comunicació i enteniment entre els diferents participants en la producció de programari al llarg del cicle de vida del projecte, tenint en compte el seu parer i responsabilitat, així com les necessitats de tots i cadascun d'ells.
- Facilitar l'operació, manteniment i ús dels productes de programari obtinguts.

A continuació s'exposa l'estructura de la Mètrica v.3:

- Planificació de Sistemes d'Informació (PSI).
- Desenvolupament de Sistemes d'Informació (DSI) que, degut a la seva complexitat, està dividit en cinc processos:
  - Estudi de Viabilitat del Sistema (EVS).
  - Anàlisi del Sistema d'Informació (ASI).
  - Disseny del Sistema d'Informació (DSI).
  - Construcció del Sistema d'Informació (CSI).
  - Implantació i Acceptació del Sistema (IAS).
- Manteniment de Sistemes d'Informació (MSI).

Dins aquest capítol, es desglossarà cadascun dels processos en les activitats i tasques corresponents, aplicat al projecte que ens ocupa.

4. Planificació de Sistemes d'Informació (PSI)

4.1 Anàlisi de la necessitat del PSI

El projecte documentat en aquesta memòria neix com a necessitat de disposar d'una eina útil per trobar informació d'events de qualsevol temàtica i situats arreu del món aprofitant les enormes capacitats de les xarxes socials a Internet.

Gràcies a aquest fenomen, que promou la comunicació entre usuaris, la creació de comunitats i la cooperació entre ells, s'obre un gran ventall de possibilitats que hauria de permetre la creació d'un sistema capaç de suplir aquesta mancança.

4.2 Identificació de l'abast del PSI

L'objectiu del projecte és aprofitar les capacitats de les xarxes socials per crear un lloc web que permeti enviar i geolocalitzar events existents a Internet, que podran ser revisats i promoguts pels usuaris.

Per altra banda, la solució escollida ha de tenir la capacitat de ser entesa, apresa, utilitzada i atractiva per l'usuari. També ha de permetre la comunicació entre usuaris i la creació de comunitats per poder compartir informació entre usuaris amb els mateixos interessos. Cal posar especial atenció en l'aspecte de la cooperació, ja que l'èxit del sistema depèn plenament en que els usuaris tinguin tant les facilitats d'obtenir informació com d'introduir-ne de nova.

A tot això se li ha de sumar una estratègia per aconseguir que el sistema sigui rentable a llarg termini.

5. Estudi de Viabilitat del Sistema (EVS)

5.1 Establir l'abast del sistema

5.1.1 Estudi de la sol·licitud

Com ja s'ha comentat anteriorment, la necessitat exposada és la de crear un sistema que permeti l'enviament, la geolocalització i la promoció d'events per part dels usuaris.

Així doncs la solució obtinguda haurà de permetre:

- L'enviament d'events existents a Internet.
- L'organització de la informació existent.
- L'exploració dels events existents..
- Un sistema de votació.
- Una agenda personal.
- La comunicació entre usuaris.
- Sindicació web.
- La disponibilitat d'una API simple.

5.1.2 Identificació de l'abast del sistema

L'objectiu d'aquesta aplicació és oferir una eina que permeti als usuaris la geolocalització i promoció d'events situats en qualsevol lloc del món a través de la navegació per mapes o mitjançant cerques.

Aquesta eina ha de ser capaç de funcionar sense la intervenció de cap editor (excepte en casos puntuals on el sistema es pugui veure afectat per temes legals), de manera que els mateixos usuaris siguin els encarregats d'alimentar el sistema amb nova informació, decidir quina informació és rellevant i quina no i realitzar modificacions de les dades.

A més ha de permetre als usuaris l'accés a la informació per diferents vies (navegació web, sindicació web, API...), rebre notificacions personalitzades i la comunicació amb altres usuaris.

5.2 Definició de requisits del sistema

5.2.1 Identificació de directrius tècniques i de gestió

1. **Qüestions tècniques**
   - Pel desenvolupament del sistema s'utilitza l'última versió de la metodologia que ofereix el Ministeri d'Administracions Públiques (MAP), anomenada Mètrica v.3. Aquesta metodologia contempla tant un estudi basat en processos com orientat a objectes i per tant, permet realitzar un estudi adaptat als temps actuals.
   - Caldrà disposar d'un **control de versions** que permeti centralitzar tant el programari com les imatges, documentació... en un repositori[1] que sigui accessible a través d'Internet.
   - Pel desenvolupament del programari s'utilitzarà l'arquitectura **Model Vista Controlador (MVC)**, que separa les dades d'una aplicació, la interfície d'usuari i la lògica de control en tres components diferents. Aquest patró de disseny es veu freqüentment en aplicacions web, on la vista és la pàgina HTML i el codi que proveeix de dades dinàmiques a la pàgina, el model és el Sistema de Gestió de Base de Dades i la Lògica de negoci i el controlador és el responsable de rebre els events d'entrada des de la vista.
   - El programari de l'aplicació haurà de seguir un conjunt d'**estàndards** per facilitar tant la seva nova creació com el seu manteniment.
   - La base de dades tindrà una estructura relacional que permeti complir regles d'integritat referencial.
   - El sistema haurà de tenir una **arquitectura distribuïda** i contemplar la possibilitat d'un ràpid creixement per a poder garantir la seva disponibilitat.
2. **Qüestions de seguretat**
   - L'accés a l'aplicació serà obert a qualsevol usuari tot i que disposarà de més o menys funcionalitats en funció de si està registrat o no i de si es tracta d'un usuari més o menys actiu.
   - Per evitar abusos en el sistema de votació es controlarà el vot per IP.
   - S'haurà de disposar d'un sistema de còpies de seguretat que garanteixi la integritat de les dades en cas de pèrdua.

5.2.2 Identificació de requisits del sistema

Un cop estudiades les necessitats, s'ha identificat els requisits que ha de contemplar el sistema a desenvolupar:

- Enviament d'events: Permetre afegir les dades principals d'un event existent a Internet i geolocalitzar-lo en el mapa.
- Organització de la informació: Es disposarà de categories i metacategories per agrupar els events, a més d'un sistema d'etiquetes que faciliti les cerques del contingut.

[1] Lloc en el que s'emmagatzemen les dades actualitzades i històriques.

- Exploració dels events existents: Mitjançant un mapa s'ha de poder veure les dades de qualsevol event.
- Sistema de votació: Atorgarà la capacitat de poder decidir quina informació és més rellevant.
- Agenda personal: Servirà per registrar events i d'aquesta manera poder rebre notificacions que informin de canvis o simplement serveixin com a recordatori.
- Comunicació entre usuaris: Es realitzarà a través de comentaris al peu dels events i/o del xat intern.
- Sindicació web: Distribuirà el contingut de l'aplicació utilitzant l'estàndard RSS.
- Disponibilitat d'una API simple: Permetrà l'accés a certa informació des d'aplicacions externes.

5.3 Estudi d'alternatives de solució

5.3.1 Preselecció d'alternatives de solució

Donada la situació actual, es poden plantejar les següents opcions:

- Realització d'un producte a mida.
- Escollir una sèrie de paquets del mercat, comprar-los i adaptar-los als requeriments del projecte.
- Escollir una sèrie de paquets lliures i gratuïts i adaptar-los als requeriments del projecte.

5.3.2 Descripció de les alternatives de solució

■ Realització d'un producte a mida

Aquesta alternativa és la més costosa en temps de totes. Un cop establerts els requisits del sistema es procedeix a realitzar el desenvolupament del projecte des de zero.

■ Escollir una sèrie de paquets del mercat, comprar-los i adaptar-los als requeriments del projecte

Aquesta alternativa es té en compte per la seva ràpida forma de resoldre el problema. Un cop establerts els requisits es busca les eines existents que s'hi adaptin millor i es procedeix a fer-hi els canvis necessaris per obtenir el sistema.

■ Escollir una sèrie de paquets lliures i gratuïts i adaptar-los als requeriments del projecte

Aquesta alternativa és molt similar a l'anterior, excepte que no es disposa del suport de l'empresa propietària del programari adquirit, fet que en alguns casos pot significar que

l'adaptació sigui més costosa.

5.3.3 Estudi dels riscos

■ Realització d'un producte a mida

En aquest cas els principals riscos que s'ha torbat són els següents:

1. Desenvolupar tot el sistema des de zero pot ser molt costós degut a la gran complexitat d'alguna de les seves parts i, per tant, no s'amortitzarà fins passat molt de temps.
2. El projecte es pot desviar molt en temps i/o preu de les prediccions fetes a priori.

■ Escollir una sèrie de paquets del mercat, comprar-los i adaptar-los als requeriments del projecte

Aquesta alternativa té com a riscos:

- El preu d'algun dels components pot ser molt elevat i no amortitzar-se fins passat molt de temps.
- El fet de no disposar del codi font d'alguns paquets pot dificultar molt l'adaptació d'aquests al sistema implementat.
- En cas que es compri paquets propietaris, el seu cost i no disposar del codi, font pot augmentar molt la dependència amb l'empresa propietària del paquet.

■ Escollir una sèrie de paquets lliures i gratuïts i adaptar-los als requeriments del projecte

Per aquesta opció tenim els riscos següents:

- El fet de no disposar, en alguns casos, de suport dels paquets per part de l'empresa creadora pot dificultar l'adaptació al sistema desenvolupat. Sobretot si no hi ha al darrere una gran comunitat d'usuaris.

5.4 Selecció de la solució

En aquest cas la millor alternativa és la tercera, ja que requereix una despesa inferior i l'obtenció de beneficis és quasi immediata. Per altra banda, es dóna molta importància a la utilització de programari lliure, degut a la gran flexibilitat que aporta i al seu baix cost. Per tant, l'alternativa escollida és: *Escollir una sèrie de paquets lliures i gratuïts i adaptar-los als requeriments del projecte.*

Així doncs, el pla de treball queda representat, en el diagrama de Gantt de la figura 5.1.

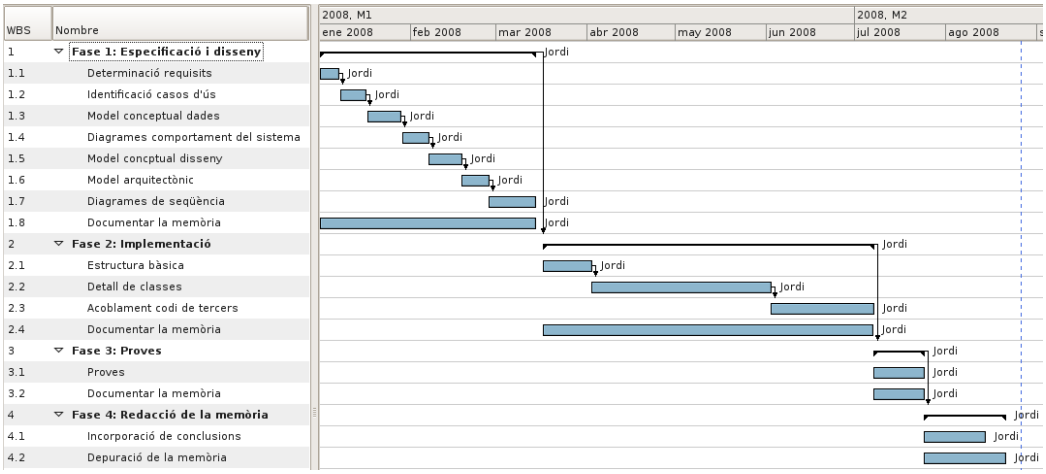

Figura 5.1: Diagrama de Gantt

6. Anàlisi del Sistema d'Informació (ASI)

6.1 Definició del sistema

6.1.1 Determinar l'abast del sistema

El sistema haurà de ser capaç de proporcionar als usuaris la possibilitat de gestionar events existents a Internet. Per això els usuaris podran crear referències amb les principals dades dels events i d'aquesta manera, disposar d'un lloc on trobar fàcilment la informació que busquen.

Cada usuari podrà di posar d'un compte personal per administrar els events que siguin del seu interès i poder rebre notificacions sobre aquests.

Es podrà accedir a la informació de diverses maneres, ja sigui a través de l'exploració per mapa, utilitzant cerques per text, mitjançant sindicació web o a partir d'una API.

En tot moment els usuaris tindran el control de quina informació es visualitza en el sistema, ja que ells mateixos seran els que l'alimentaran. També podran decidir quina informació és més rellevant gràcies a un sistema de votació que facilitarà l'accés dels events més populars entre els usuaris.

A més el sistema farà ús de l'anomenada intel·ligència col·lectiva per proporcionar als usuaris recomanacions a partir del comportament d'altres usuaris amb gustos similars.

6.1.2 Identificació de l'entorn tecnològic

L'aplicació s'executarà sobre el sistema operatiu GNU/Linux i en un entorn web accessible des d'Internet. Per tant, els usuaris finals només necessitaran disposar d'un navegador web per poder utilitzar el sistema.

A l'hora de triar les eines per la creació del sistema es valorarà molt favorablement l'ús de solucions lliures degut a la gran flexibilitat que normalment proporcionen.

Per la creació del sistema es disposarà de tres entorns de treball:

- Entorn de desenvolupament: Permetrà desenvolupar les noves funcionalitats i solucionar problemes en les ja existents.
- Entorn de preproducció: Haurà de permetre provar les funcionalitats acabades a l'entorn de desenvolupament amb la certesa que funcionaran de la mateixa manera a l'entorn de producció.
- Entorn de producció: Executarà la versió estable del sistema i serà la que interactuarà amb els usuaris finals.

6.1.3 Especificació d'estàndards i normes

Respecte al compliment d'estàndards o normes, cal dir que en les fases d'anàlisi i disseny cal tenir en compte la nomenclatura definida pel Llenguatge Unificat de Modelatge (UML) en la il·lustració de diagrames i figures.

Pel que fa al programari, caldrà seguir uns estàndards ja establerts i reconeguts com seria, en el cas del Web, els proporcionats per la **World Wide Web Consortium**[1] (**W3C**).

6.1.4 Identificació dels usuaris participants i finals

Els usuaris participants i finals seran internautes d'arreu del món amb perfils molt diversos, encara que la fase de proves serà realitzada per professionals de reconeguda vàlua professional.

6.2 Establiment de requisits

6.2.1 Obtenció de requisits

Tot seguit es defineixen els requisits que ha de complir el sistema, segons els guions establerts, mitjançant el cas d'ús general de la figura 6.1.

Com es pot apreciar, hi ha quatre tipus d'usuaris: **Usuari**, **Usuari anònim**, **Usuari registrat** i **Administrador**. L'actor Usuari no seria un participant real del sistema ja que la seva funció és englobar les accions comunes entre els altres tres usuaris. Per tant, l'Usuari anònim i l'Usuari registrat són una especialització d'Usuari i només es mostrarà les accions que els diferència. Al mateix temps, l'actor Administrador és una especialització d'Usuari registrat que afegeix dues accions més. Per altra banda, també es representa l'actor **Sistema**, que és l'encarregat de realitzar les tasques automàtiques.

A continuació es mostrarà amb més detall els casos d'ús de cada actor:

■ Usuari

Com ja hem comentat anteriorment aquest actor engloba les accions comunes dels tres usuaris reals del sistema. Els casos d'ús en els que interactua són:

- Canviar idioma
- Cercar events
- Cercar per mapa
- Cercar per text
- Cercar per RSS
- Votar event

[1] Consorci internacional que treballa per desenvolupar i promocionar estàndards per al Web.

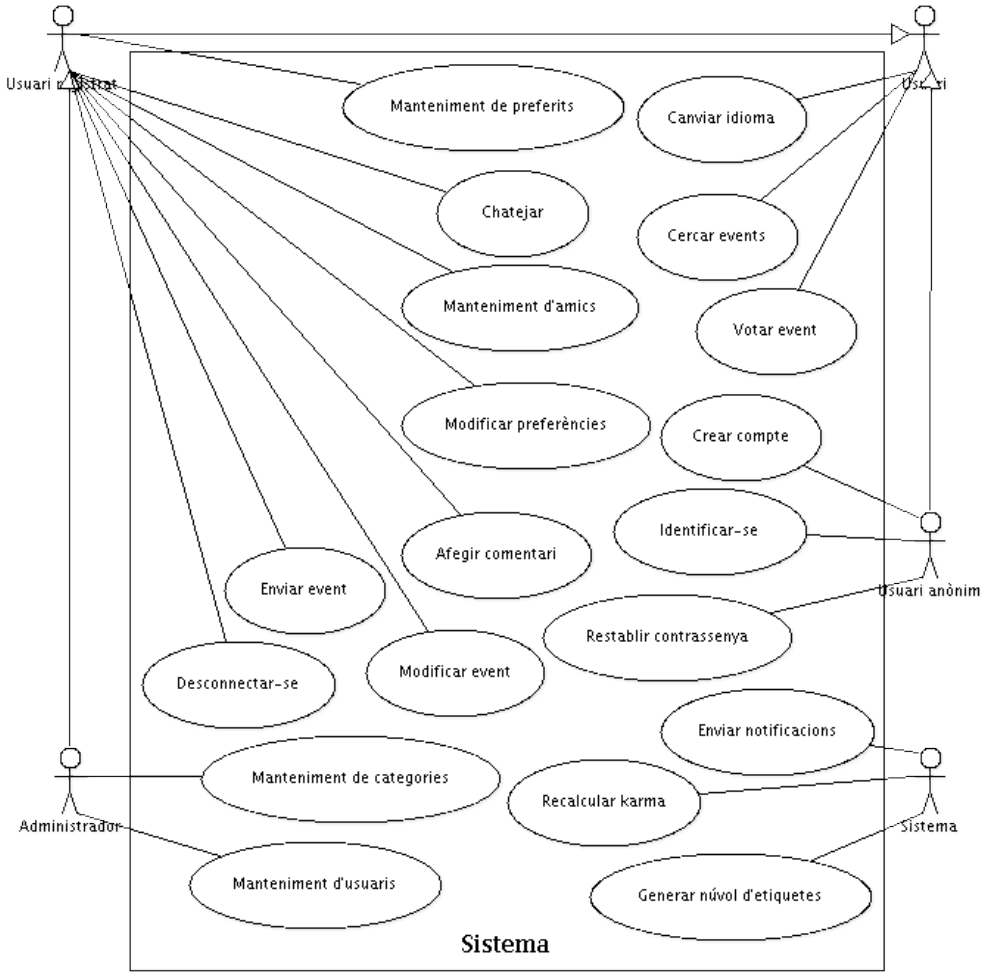

Figura 6.1: Cas d'ús general

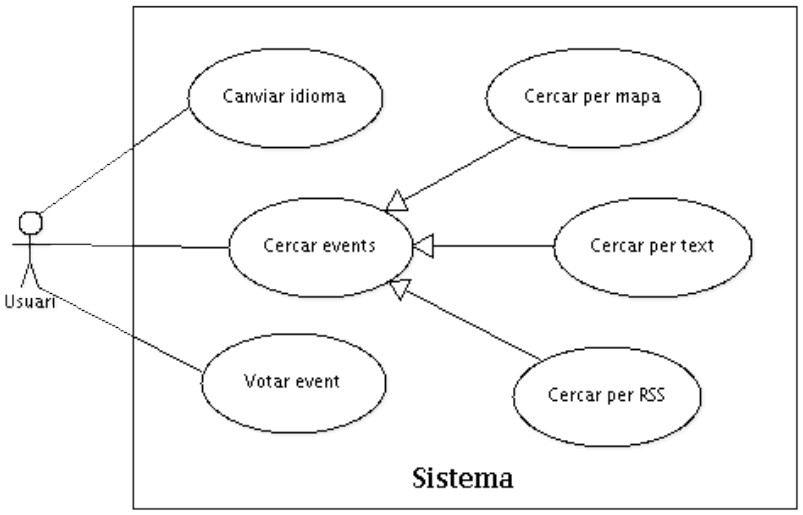

Figura 6.2: Casos d'ús de l'actor Usuari

Com es pot apreciar en el cas d'ús (veure figura 6.2), hi apareixen tres relacions de generalització sobre *Cercar events*. Això és així ja que *Cercar per mapa*, *Cercar per text* i *Cercar per RSS* són tres tipus de cerca diferents i el que fan és sobreescriure el comportament del cas d'us inicial.

Cal comentar que en el cas que s'afegís funcionalitat al comportament del cas d'ús inicial, les relacions haurien de ser d'extensió (p.ex. *Cercar events i notificar*) i no de generalització.

■ Usuari anònim

Aquest actor representa l'usuari que no disposa de compte personal dins l'aplicació i que, com a conseqüència, en pot fer un ús més limitat. Els casos d'ús en els que interactua són:

- Crear compte
- Modificar preferències
- Identificar-se

En aquest cas (veure figura 6.3) ens trobem amb una relació d'inclusió. Aquesta relació ens indica que el cas d'ús *Modificar preferències* és requerit per poder realitzar el cas d'ús *Crear compte*.

■ Usuari registrat

Aquest actor representa l'usuari que disposa d'un compte dins l'aplicació i per tant, té la capacitat d'utilitzar el sistema amb totes les seves funcionalitats. Els casos d'ús en els que interactua són:

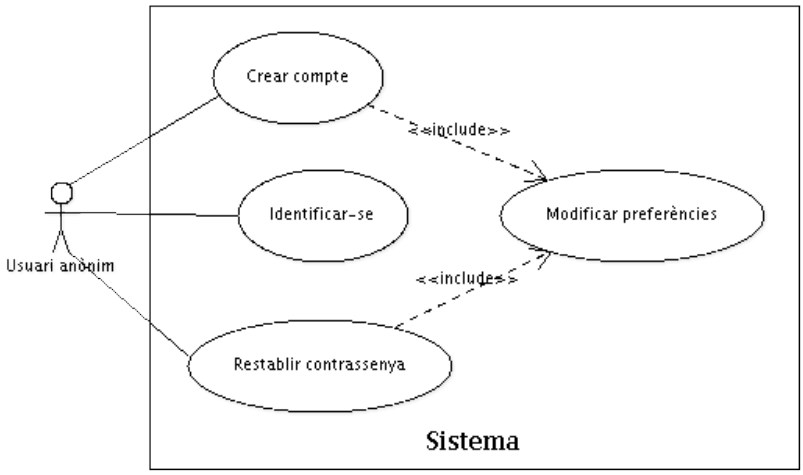

Figura 6.3: Casos d'ús de l'actor Usuari anònim

- Enviar event
- Votar event
- Modificar event
- Modificar preferències
- Afegir comentari
- Afegir amic
- Eliminar amic
- Afegir event a preferits
- Eliminar event de preferits

Com en el cas anterior, aquí (veure figura 6.4) també ens trobem amb una relació d'inclusió. Per tant, s'entén que en el moment d'enviar un event aquest serà votat per l'usuari que l'envia.

■ Administrador

Aquest actor representa l'administrador del sistema, que és una especialització de l'actor Usuari registrat. Els casos d'ús en els que interactua són:

- Afegir categoria
- Eliminar categoria
- Modificar categoria
- Banejar usuari
- Treure baneig a usuari

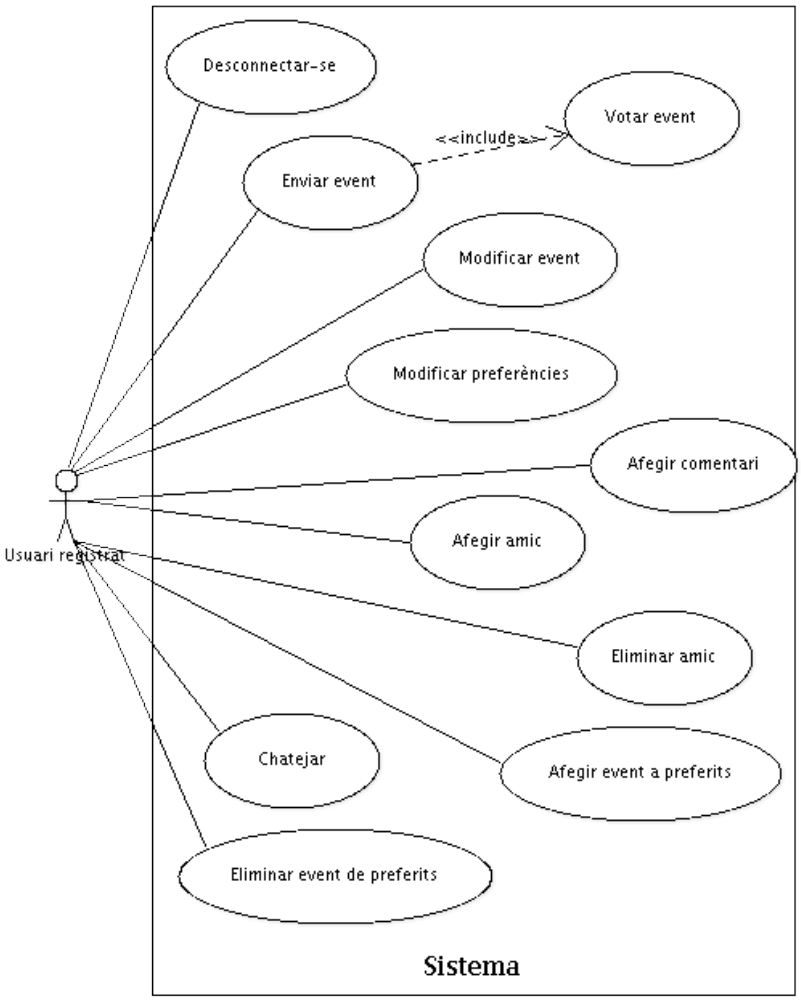

Figura 6.4: Casos d'ús de l'actor Usuari registrat

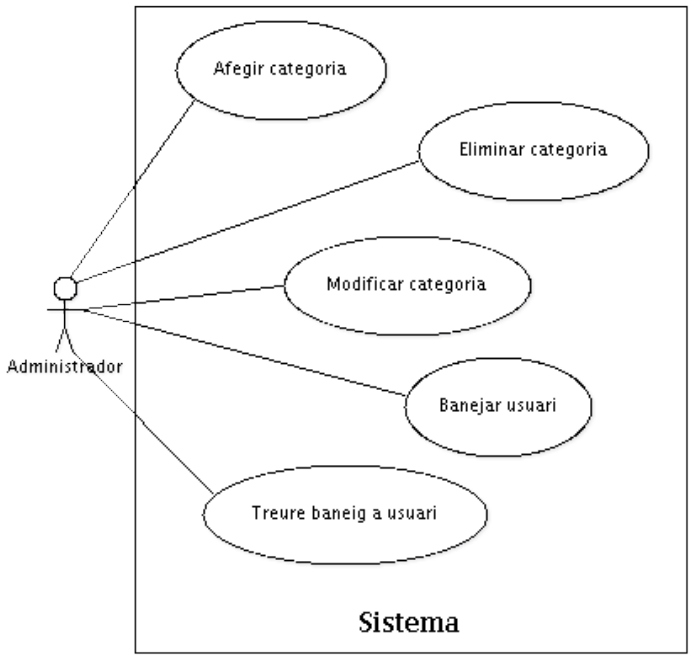

Figura 6.5: Casos d'ús de l'actor Administrador

■ **Sistema**

En aquest cas (veure figura 6.6) l'actor no representa cap usuari del sistema, sinó que es tracta del mateix sistema realitzant tasques automàtiques. Els casos d'ús en els que interactua són:

- Enviar notificacions
- Recalcular karma
- Generar núvol d'etiquetes

6.2.2 Especificació de casos d'ús

En aquest punt es detallarà cadascun dels casos d'ús mostrats anteriorment segons un esquema comú.

Cas d'ús: Canviar idioma

Descripció: Permet canviar l'idioma de l'aplicació.

Actors: Usuari

Precondició: - - -

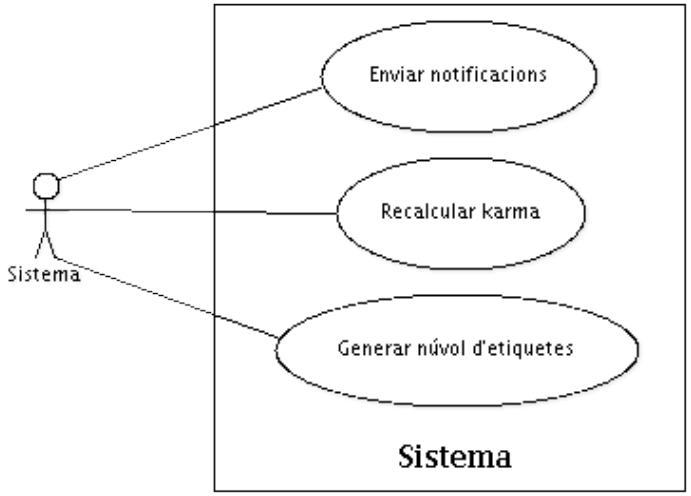

Figura 6.6: Casos d'ús de l'actor Sistema

Flux Principal:

1. L'usuari selecciona l'idioma desitjat.
2. Es recarrega la pàgina actual amb el nou idioma.

Fluxos Alternatius:

2. Si no es troba l'idioma seleccionat es carrega la pàgina amb l'idioma per defecte.

Postcondició: L'idioma de l'aplicació ha canviat pel seleccionat.

Cas d'ús: Cercar per mapa

Descripció: Permet cercar events mitjançant un mapa.

Actors: Usuari

Precondició: - - -

Flux Principal:

1. L'usuari selecciona els criteris de cerca.
2. El sistema aplica els criteris seleccionats.
3. L'usuari modifica la situació del mapa per la desitjada.
4. El sistema obté els events de l'àrea seleccionada segons els criteris establerts. Els events són representats en el mapa mitjançant marcadors. Es mostra una breu descripció dels events seleccionats a la llista lateral.

Fluxos Alternatius:

4. Si no hi ha events que s'ajustin a la cerca, la llista d'events es mostra buida i el mapa sense cap marcador.

Postcondició: Es mostren els events situats a l'àrea seleccionada que compleixen els criteris seleccionats

Cas d'ús: Cercar per text

Descripció: Permet cercar events mitjançant paraules clau.

Actors: Usuari

Precondició: - - -

Flux Principal:

1. L'usuari entra la/es paraula/es clau que vol cercar i prem el botó de cerca.
2. El sistema obté els events que s'ajustin a la cerca.
3. Es mostra la pantalla de cerques amb el llistat dels events seleccionats.

Fluxos Alternatius:

2. Si la cerca no ha obtingut resultats es mostra la pantalla de cerques amb un missatge que informi a l'usuari que no s'ha trobat events.

Postcondició: Es mostren els events que s'ajusten a la cerca entrada per l'usuari.

Cas d'ús: Cercar per RSS

Descripció: Permet obtenir una llista d'events per RSS a partir d'uns criteris seleccionats.

Actors: Usuari

Precondició: L'usuari disposa d'una aplicació amb suport de RSS.

Flux Principal:

1. L'usuari selecciona l'enllaç a la subscripció RSS que s'ajusta a la cerca desitjada.
2. El sistema obté els events resultants de la cerca seleccionada i genera el fitxer RSS.
3. L'aplicació de l'usuari obté el fitxer RSS i es subscriu a la cerca.

Fluxos Alternatius:

Postcondició: L'usuari es subscriu a la cerca seleccionada i obté periòdicament els nous

events que s'ajusten als criteris seleccionats.

Cas d'ús: Votar event

Descripció: Permet votar un event.

Actors: Usuari

Precondició: L'usuari no pot haver votat l'event anteriorment. L'event no pot haver estat votat per cap usuari des de la mateixa direcció IP.

Flux Principal:

1. L'usuari prem el botó de votar l'event.
2. El sistema comprova que l'event és votat per l'usuari que utilitza l'aplicació.
3. El comptador de vots de l'event és incrementat o decrementat en funció de si el vot és positiu o negatiu.
4. El karma de l'event és incrementat o decrementat per el karma de l'usuari o per la puntuació base si l'usuari és anònim.
5. S'actualitza la pantalla amb la informació de l'event modificada pel vot.

Fluxos Alternatius:

2. L'usuari que està votant no és el mateix que ha carregat la pantalla o no ho fa des de la mateixa direcció IP (normalment es tractaria d'un intent d'abús). Es mostra un missatge informant de que s'ha produït un error.

Postcondició: L'usuari es subscriu a la cerca seleccionada i obté periòdicament els nous events que s'ajusten als criteris seleccionats.

Cas d'ús: Crear compte

Descripció: Permet crear un compte personal a un usuari anònim.

Actors: Usuari anònim

Precondició: El compte creat no pot existir.

Flux Principal:

1. L'usuari entra les dades del seu nou compte personal.
2. El sistema comprova que el nom d'usuari entrat és vàlid.
3. El sistema valida que el nom d'usuari entrat no ha estat utilitzat.
4. El sistema comprova que el correu electrònic entrat és vàlid.
5. El sistema valida que el correu electrònic no ha estat utilitzat.

6. El sistema valida que la contrasenya entrada compleix amb els requisits mínims de seguretat.
7. El sistema comprova que la contrasenya de confirmació entrada coincideix amb la contrasenya inicial.
8. El sistema guarda les dades del compte i el deixa en estat inactiu.
9. El sistema envia un correu de confirmació a l'adreça proporcionada.
10. L'usuari valida el compte utilitzant l'enllaç proporcionat en el correu.
11. El sistema activa el compte.

Fluxos Alternatius:

2. El nom d'usuari no és vàlid. El sistema mostra un missatge per informar a l'usuari.
3. El nom d'usuari ja està utilitzat per un altre usuari. El sistema mostra un missatge per informar a l'usuari.
4. El correu electrònic entrat no és vàlid. El sistema mostra un missatge per informar a l'usuari.
5. El correu electrònic ja ha estat utilitzat per un altre usuari. El sistema mostra un missatge per informar a l'usuari.
6. La contrasenya entrada no compleix els requisits mínims de seguretat. El sistema mostra un missatge per informar a l'usuari.
7. Les dues contrasenyes entrades no coincideixen. El sistema mostra un missatge per informar a l'usuari.

Postcondició: S'ha creat un compte d'usuari.

Cas d'ús: Identificar-se

Descripció: Permet a un usuari anònim accedir al seu compte personal.

Actors: Usuari anònim

Precondició: L'usuari ha de disposar d'un compte personal.

Flux Principal:

1. L'usuari entra el seu nom d'usuari i la seva contrasenya.
2. El sistema comprova que el nom d'usuari existeix.
3. El sistema comprova que l'usuari no està banejat.
4. El sistema comprova que el compte del nom d'usuari entrat s'identifica amb la contrasenya proporcionada.
5. El sistema identifica a l'usuari.

Fluxos Alternatius:

2. El nom d'usuari entrat no existeix. El sistema mostra un missatge informant que el nom d'usuari o la contrasenya és incorrecte.
3. L'usuari està banejat. El sistema mostra un missatge informant que el compte de l'usuari està banejat i els motius.
4. La contrasenya proporcionada no identifica el nom d'usuari entrat. El sistema mostra un missatge informant que el nom d'usuari o la contrasenya és incorrecte.

Postcondició: L'usuari ha estat identificat i passa de ser un usuari anònim a ser un usuari registrat.

Cas d'ús: Restablir contrasenya

Descripció: Permet a un usuari anònim restablir la contrasenya del seu compte.

Actors: Usuari anònim

Precondició: L'usuari ha de disposar d'un compte personal.

Flux Principal:

1. L'usuari entra el seu nom d'usuari.
2. El sistema comprova que el nom d'usuari existeix.
3. El sistema envia un correu de confirmació a l'adreça electrònica relacionada amb el nom d'usuari entrat.
4. L'usuari restableix la contrasenya utilitzant l'enllaç proporcionat en el correu.

Fluxos Alternatius:

2. El nom d'usuari entrat no existeix. El sistema mostra un missatge informant que el nom d'usuari és incorrecte.

Postcondició: L'usuari ha estat identificat i té l'opció de modificar la seva contrasenya.

Cas d'ús: Desconnectar-se

Descripció: Permet a un usuari registrat desconnectar-se del seu compte.

Actors: Usuari registrat

Precondició: - - -

Flux Principal:

1. L'usuari prem el botó de desconnexió.
2. El sistema desconnecta a l'usuari del seu compte.

Fluxos Alternatius:

Postcondició: L'usuari ha estat desconnectat i passa a ser un usuari anònim.

Cas d'ús: Enviar event

Descripció: Permet a un usuari enviar un event al sistema.

Actors: Usuari registrat

Precondició: - - -

Flux Principal:

1. L'usuari entra l'URL de l'event.
2. El sistema comprova que l'URL no ha estat entrada anteriorment.
3. El sistema comprova que l'URL existeix a Internet.
4. El sistema obté el títol de la pàgina que enllaça l'URL entrada.
5. El sistema mostra la pantalla per introduir les dades de l'event. El títol de l'event està omplert amb el títol obtingut anteriorment.
6. L'usuari emplena les dades de l'event i prem el botó enviar.
7. El sistema valida que tots els camps obligatoris estan entrats.
8. El sistema valida que la data entrada és correcte.
9. El sistema guarda les dades de l'event.

Fluxos Alternatius:

2. L'URL de l'event ja ha estat entrada anteriorment. El sistema mostra la previsualització de l'event i un missatge informant que l'event ja existeix.
3. L'URL no existeix a Internet. El sistema mostra un missatge informant de que l'URL és incorrecte.
4. El sistema no pot obtenir el títol de la pàgina. El títol és inicialitzat amb espais.
7. No s'ha entrat totes les dades de l'event. El sistema mostra un missatge informant les dades que falta entrar.
8. La data entrada és invàlida. El sistema mostra un missatge informant que la data és incorrecte.

Postcondició: L'event entrat és desat a la base de dades i accessible des de l'aplicació.

Cas d'ús: Modificar event

Descripció: Permet a un usuari registrat modificar un event.

Actors: Usuari registrat

Precondició: L'usuari és el mateix que ha enviat l'event i fa menys de N segons que l'ha enviat o té més de M punts de karma.

Flux Principal:

1. El sistema comprova que l'usuari pot modificar l'event.
2. L'usuari modifica les dades de l'event.

Fluxos Alternatius:

1. El sistema comprova que l'usuari ha enviat l'event i que fa menys del temps especificat per modificacions o que l'usuari té més karma del especificat per modificacions de qualsevol event.

Postcondició: Les dades de l'event han estat modificades.

Cas d'ús: Modificar preferències

Descripció: Permet a un usuari registrat modificar les preferències del seu compte personal.

Actors: Usuari registrat

Precondició: L'usuari és el propietari del compte a modificar.

Flux Principal:

1. L'usuari modifica les dades del compte i prem el botó guardar.
2. El sistema comprova que les dades entrades són correctes.

Fluxos Alternatius:

2. Les dades entrades són incorrectes. El sistema mostra un missatge informant dels errors.

Postcondició: Les dades del compte han estat modificades.

Cas d'ús: Afegir comentari

Descripció: Permet a un usuari afegir un comentari en el detall d'un event.

Actors: Usuari registrat

Precondició: - - -

Flux Principal:

1. L'usuari entra el comentari i prem el botó enviar.
2. El sistema valida que el comentari no conté instruccions no permeses (HTML, Javascript...).

Fluxos Alternatius:

2. El comentari conté instruccions no permeses. El sistema elimina les instruccions o, en cas de no poder eliminar-les, mostra un missatge informant de l'error.

Postcondició: S'ha afegit un comentari en el detall de l'event.

Cas d'ús: Afegir amic

Descripció: Permet afegir com amic a un usuari.

Actors: Usuari registrat

Precondició: L'usuari no és amic

Flux Principal:

1. L'usuari prem el botó d'afegir amic.

Fluxos Alternatius:

Postcondició: S'ha afegit un usuari com amic.

Cas d'ús: Eliminar amic

Descripció: Permet eliminar un usuari de la llista d'amics.

Actors: Usuari registrat

Precondició: L'usuari és amic

Flux Principal:

1. L'usuari prem el botó d'eliminar amic.

Fluxos Alternatius:

Postcondició: S'ha eliminat un usuari de la llista d'amics.

Cas d'ús: Afegir event a preferits

Descripció: Permet afegir un event a la llista de preferits.

Actors: Usuari registrat

Precondició: L'event no està a la llista de preferits.

Flux Principal:

1. L'usuari prem el botó d'afegir event a preferits.

Fluxos Alternatius:

Postcondició: S'ha afegit un event a la llista de preferits.

Cas d'ús: Eliminar event de preferits

Descripció: Permet eliminar un event de la llista de preferits.

Actors: Usuari registrat

Precondició: L'event està a la llista de preferits.

Flux Principal:

1. L'usuari prem el botó d'eliminar event de preferits.

Fluxos Alternatius:

Postcondició: S'ha eliminat un event de la llista de preferits.

Cas d'ús: Xatejar

Descripció: Permet la comunicació entre usuaris.

Actors: Usuari registrat

Precondició:

Flux Principal:

1. L'usuari entra el missatge i prem el botó enviar.
2. El sistema mostra el missatge a la llista de missatges enviats.

Fluxos Alternatius:

Postcondició: S'ha afegit un missatge a la sala del xat.

Cas d'ús: Afegir categoria

Descripció: Permet afegir una nova categoria al sistema.

Actors: Administrador

Precondició: La categoria no existeix.

Flux Principal:

1. L'administrador entra el nom de la categoria i selecciona de quina metacategoria és filla. Tot seguit prem el botó afegir.
2. El sistema valida que s'ha entrat el nom de la categoria.
3. El sistema valida que s'ha seleccionat la metacategoria.

Fluxos Alternatius:

2. No s'ha entrat el nom de la categoria. El sistema mostra un missatge informant que cal entrar el nom.
3. No s'ha seleccionat la metacategoria pare. El sistema marca la nova categoria com a metacategoria.

Postcondició: S'ha afegit una nova categoria al sistema.

Cas d'ús: Eliminar categoria

Descripció: Permet eliminar una categoria del sistema.

Actors: Administrador

Precondició: La categoria existeix.

Flux Principal:

1. L'administrador selecciona la categoria a eliminar i prem el botó eliminar.
2. El sistema valida que la categoria seleccionada no és una metacategoria amb fills.

Fluxos Alternatius:

2. S'ha seleccionat una metacategoria amb fills. El sistema mostra un missatge informant que la metacategoria no es pot eliminar.

Postcondició: S'ha eliminat una categoria del sistema.

Cas d'ús: Modificar categoria

Descripció: Permet modificar una categoria del sistema.

Actors: Administrador

Precondició: La categoria existeix.

Flux Principal:

1. L'administrador selecciona la categoria a modificar
2. L'administrador modifica les dades de la metacategoria i prem el botó modificar.
3. El sistema valida que el nou nom de la categoria no existeix.

Fluxos Alternatius:

3. El nou nom de categoria ja existeix. El sistema mostra un missatge informant-ho.

Postcondició: S'ha modificat una categoria del sistema.

Cas d'ús: Banejar usuari

Descripció: Permet banejar un usuari del sistema.

Actors: Administrador

Precondició: L'usuari no està banejat.

Flux Principal:

1. L'administrador entra el nom d'usuari, el motiu del baneig i la durada.
2. El sistema deshabilita el compte de l'usuari.
3. El sistema envia un correu informatiu a l'adreça electrònica associada a l'usuari informant dels motius i durada del baneig.

Fluxos Alternatius:

Postcondició: S'ha banejat un usuari.

Cas d'ús: Treure baneig a usuari

Descripció: Permet treure el baneig a un usuari del sistema.

Actors: Administrador

Precondició: L'usuari està banejat.

Flux Principal:

1. L'administrador entra el nom de l'usuari a treure el baneig.
2. El sistema habilita el compte de l'usuari.
3. El sistema envia un correu informatiu a l'adreça electrònica associada a l'usuari informant que el baneig ha sigut retirat.

Fluxos Alternatius:

Postcondició: S'ha tret el baneig a l'usuari.

Cas d'ús: Enviar notificacions

Descripció: Permet enviar per correu electrònic les notificacions pendents.

Actors: Sistema

Precondició:

Flux Principal:

1. El sistema obté les notificacions pendents.
2. El sistema envia un correu electrònic per cada notificació.

Fluxos Alternatius:

Postcondició: S'ha enviat les notificacions pendents.

Cas d'ús: Recalcular karma

Descripció: Permet recalcular el karma dels usuaris i els events.

Actors: Sistema

Precondició:

Flux Principal:

1. El sistema obté tots els events pendents dels sistema.
2. El sistema recalcula el karma dels events en funció dels paràmetres establerts.
3. El sistema obté tots els usuaris del sistema.
4. El sistema recalcula el karma dels usuaris en funció dels paràmetres establerts.

Fluxos Alternatius:

Postcondició: S'ha recalculat el karma dels events i els usuaris.

Cas d'ús: Generar núvol d'etiquetes

Descripció: Permet generar un núvol amb les etiquetes més utilitzades.

Actors: Sistema

Precondició: Hi ha etiquetes a la base de dades.

Flux Principal:

1. El sistema obté les N etiquetes més utilitzades juntament amb la seva freqüència d'aparició.
2. El sistema calcula el coeficient d'ajustament del tamany de les etiquetes.
3. El sistema calcula el tamany de cadascuna de les etiquetes.
4. El sistema ordena les etiquetes per ordre alfabètic.
5. El sistema genera el núvol d'etiquetes.

Fluxos Alternatius:

Postcondició: S'ha generat el núvol d'etiquetes.

6.3 Identificació de subsistemes d'anàlisi

Un dels avantatges de l'orientació a objectes és que els casos d'ús aporten molta informació i molt valuosa. A l'haver estructurat de forma general els casos d'ús s'ha aconseguit que aquest pas de la metodologia ja estigui quasi donat. Els subsistemes d'anàlisi que es troben són bàsicament els serveis que s'ofereixen. Això simplifica enormement la tasca de reagrupar casos d'ús ja que a l'enfocar el projecte des de la funcionalitat i no des d'un altre punt de vista ja tenim aïllats i reagrupats els casos d'ús que fan referència a un mateix tipus de funcionalitat.

Així, només cal adjuntar un diagrama (veure figura 6.7) que comprengui les relacions entre els diferents subsistemes d'anàlisi, de forma que quedi completament especificat el conjunt de relacions entre els subsistemes.

- Gestió d'events: Paquet que hauria de gestionar tot el relacionat amb els events.
- Gestió d'usuaris: Paquet per gestionar el que fa referència als usuaris.
- Administració: Paquet encarregat de controlar i gestionar les tasques d'administració del sistema.

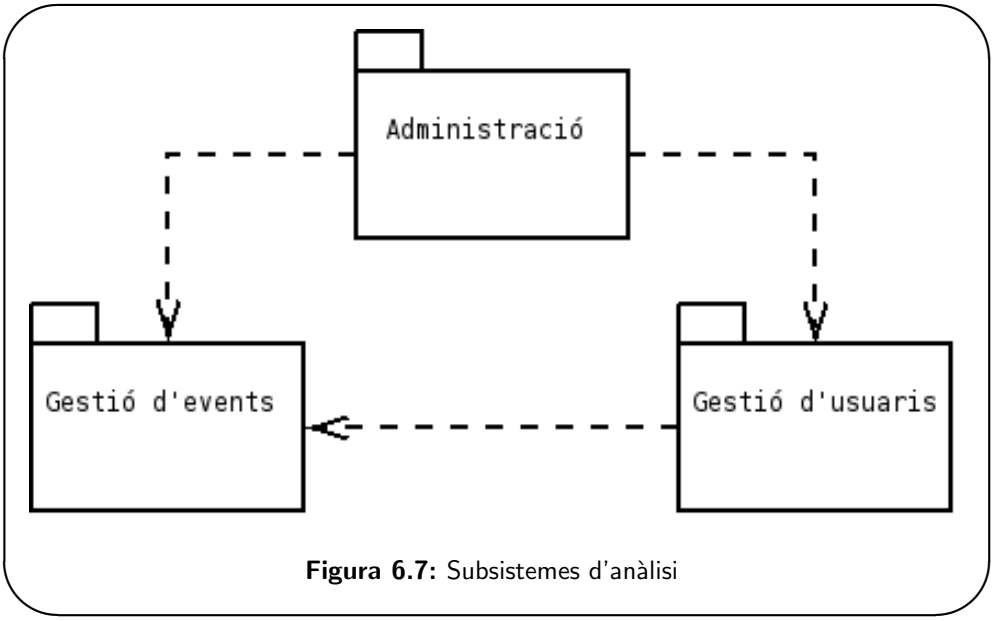

Figura 6.7: Subsistemes d'anàlisi

6.4 Identificació de classes associades a un cas d'ús

En aquesta tasca s'identificarà els objectes necessaris per realitzar els casos d'ús basant-nos en l'especificació que tenim de l'apartat anterior.

Les classes que s'identifiquen en aquesta tasca poden ser:

- Classes d'Entitat (representen la informació manipulada en el cas d'ús).
- Classes d'Interfície d'Usuari (s'utilitzen per descriure la interacció entre el sistema i els seus actors. Solen representar abstraccions de finestres, interfícies de comunicació, formularis...).
- Classes de Control (són responsables de la coordinació, seqüència de transaccions i control dels objectes relacionats amb un cas d'ús).

A continuació s'indica les classes que s'identifiquen a cada cas d'ús:

- **Canviar idioma:** Pantalla d'entrada, LanguageController.
- **Cercar per mapa:** Pantalla d'exploració, MapController, Event.
- **Cercar per text:** Pantalla d'entrada, SearchController, Event.
- **Cercar per RSS:** Pantalla d'entrada, RSS2Controller, Event.
- **Votar event:** Pantalla d'entrada, EventController, Event. User.
- **Crear compte:** Formulari de registre, UserController, User, Mailer.
- **Identificar-se:** Formulari d'identificació, AuthController, User.
- **Restablir contrasenya:** Formulari restablir contrasenya, UserController, User, Mailer.
- **Desconnectar-se:** Pantalla d'entrada, AuthController, User.
- **Enviar event:** Formulari entrada URL, Formulari d'events, EventController, Event.

- **Modificar event:** Formulari d'events, EventController, User, Event.
- **Modificar preferències:** Formulari de preferències, UserController, User.
- **Afegir comentari:** Formulari de comentari, EventController, Comment, Event.
- **Afegir amic:** Pantalla perfil d'usuari, UserController, User.
- **Eliminar amic:** Pantalla perfil d'usuari, UserController, User.
- **Afegir event a preferits:** Pantalla d'event, EventController, Event, User.
- **Eliminar event de preferits:** Pantalla de preferits, EventController, Event, User.
- **Xatejar:** Pantalla de xat, ChatController, User.
- **Afegir categoria:** Formulari de categories, CategoryController, Category.
- **Eliminar categoria:** Formulari de categories, CategoryController, Category.
- **Modificar categoria:** Formulari de categories, CategoryController, Category.
- **Banejar usuari:** Formulari de baneig, UserController, User, Ban, Mailer.
- **Treure baneig a usuari:** Formulari de baneig, UserController, User, Ban, Mailer.
- **Enviar notificacions:** Notification, Event, User, Mailer.
- **Recalcular karma:** Event, User.
- **Generar núvol d'etiquetes:** Tag, Pantalla d'entrada.

6.5 Descripció de la interacció d'objectes

En aquesta secció es mostra la interacció d'objectes mitjançant diagrames de seqüència. Aquests diagrames fan referència als casos d'ús comentats anteriorment i són els següents:

Canviar idioma (veure figura 6.8)

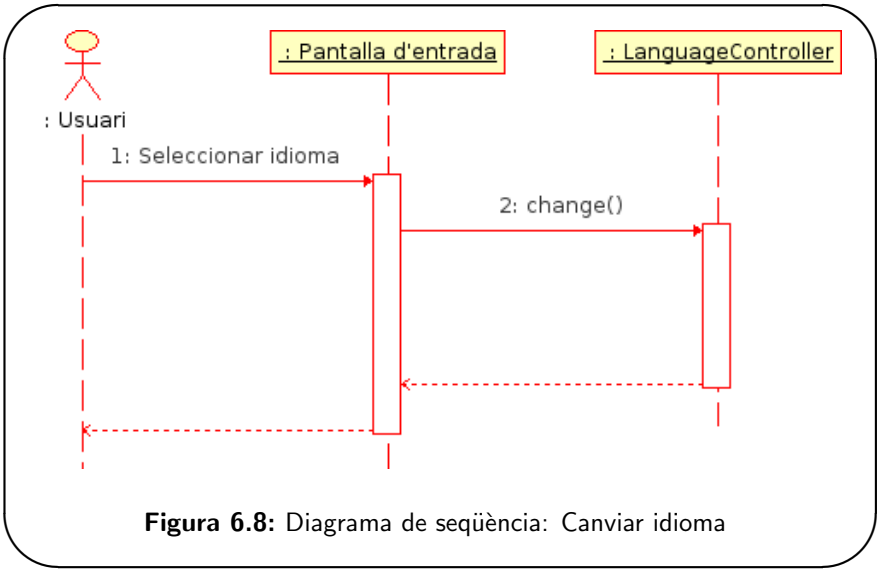

Figura 6.8: Diagrama de seqüència: Canviar idioma

Aquesta acció pot ser executada des de qualsevol pantalla de l'aplicació. L'usuari disposarà d'un desplegable per seleccionar l'idioma desitjat i tot segut es recarregarà la pàgina amb el nou idioma.

Cercar per mapa (veure figura 6.9)

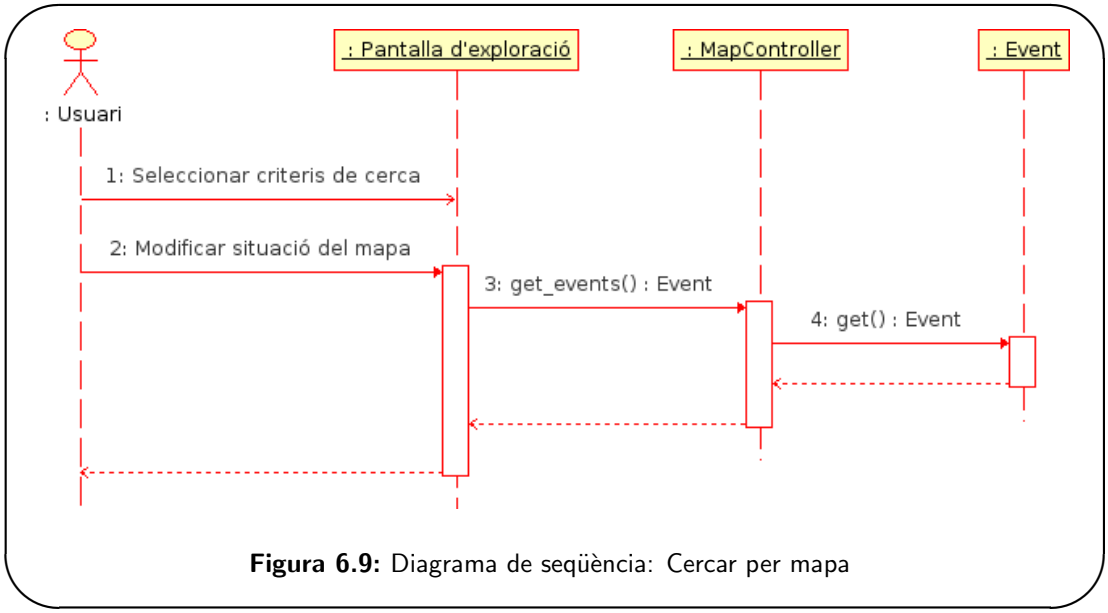

Figura 6.9: Diagrama de seqüència: Cercar per mapa

L'usuari disposarà d'un conjunt de criteris a seleccionar per poder adaptar la cerca a les seves necessitats. Un cop seleccionats els criteris es podrà navegar pel mapa desplaçant-lo a la zona desitjada. Les dades de la nova posició seran enviades, juntament amb els criteris, al sistema i aquest obtindrà els events corresponents. Tot seguit, el mapa mostrarà els nous events obtinguts.

Cercar per text (veure figura 6.10)

Aquesta acció pot ser executada des de qualsevol pantalla de l'aplicació. L'usuari disposarà d'una caixa de text on entrar les paraules clau amb que vol cercar els events. Un cop entrades les paraules clau el sistema cercarà els events que hi coincideixin i els retornarà en un llistat.

Cercar per RSS (veure figura 6.11)

Per realitzar aquesta acció l'usuari necessita un navegador amb suport de RSS o un lector de RSS. Un cop seleccionada la subscripció desitjada el sistema obtindrà els events que compleixin els criteris de la subscripció i els retornarà a l'usuari utilitzant RSS.

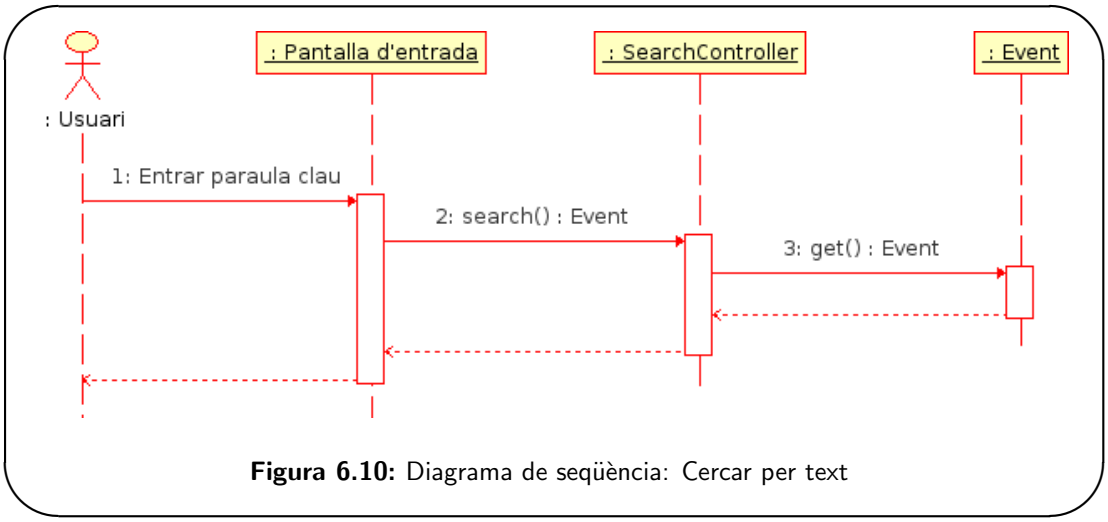

Figura 6.10: Diagrama de seqüència: Cercar per text

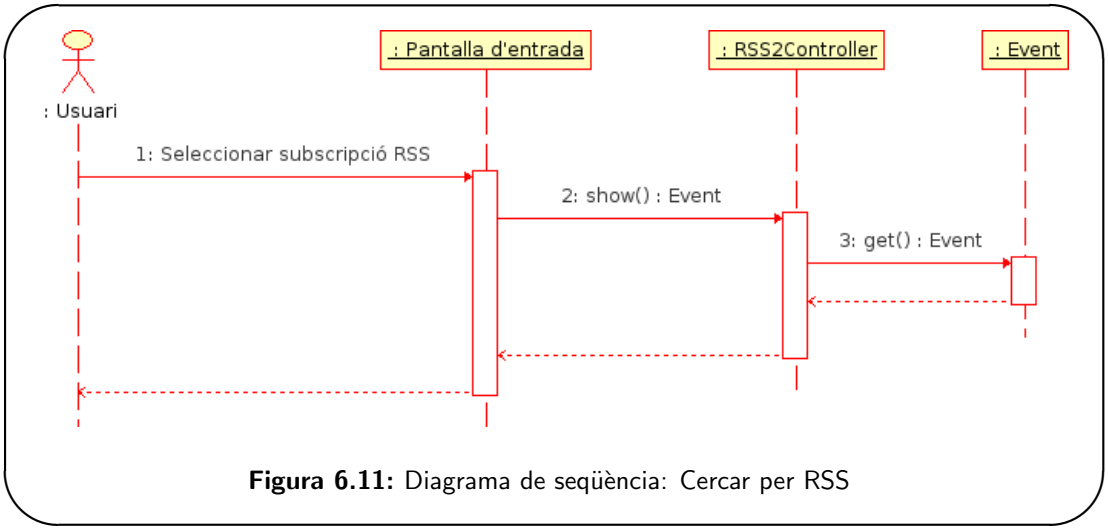

Figura 6.11: Diagrama de seqüència: Cercar per RSS

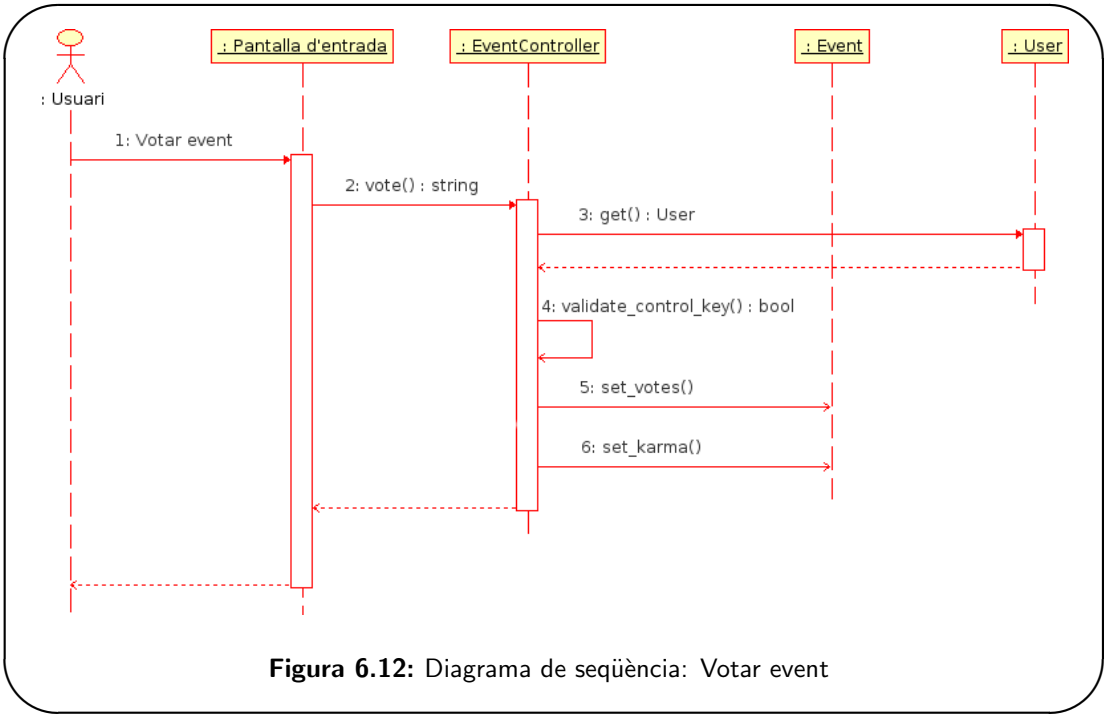

Figura 6.12: Diagrama de seqüència: Votar event

Votar event (veure figura 6.12)

En totes les pantalles on es mostri la informació referent a un event es disposarà dels botons per votar. L'adreça que enllaçarà cada botó haurà estat generada segons unes dades de control (usuari, direcció IP, valor aleatori...) i d'aquesta manera serà única. Quan l'usuari voti un event, el sistema obtindrà les dades d'aquest usuari i tornarà a generar la clau de control. Si aquesta coincideix amb l'obtinguda es passarà a realitzar la votació i a incrementar el comptador de vots i el karma.

Crear compte (veure figura 6.13)

Aquesta acció es podrà realitzar des de qualsevol pantalla sempre i quan l'usuari no estigui identificat. En primer lloc es demanarà les dades del compte, que l'usuari entrarà i enviarà al sistema. Tot seguit el sistema validarà que les dades són correctes i, en cas afirmatiu, crearà el compte desactivat. Per poder activar-lo, l'usuari rebrà un correu electrònic a la direcció especificada amb un enllaç que, al seguir-lo, permetrà al sistema activar el compte.

Identificar-se (veure figura 6.14)

Aquesta acció es podrà realitzar des de qualsevol pantalla sempre i quan l'usuari no estigui ja identificat i disposi d'un compte. Per això entrarà les seves dades identificatives que el sistema rebrà. Un cop comprovat que les dades són correctes l'usuari quedarà identificat.

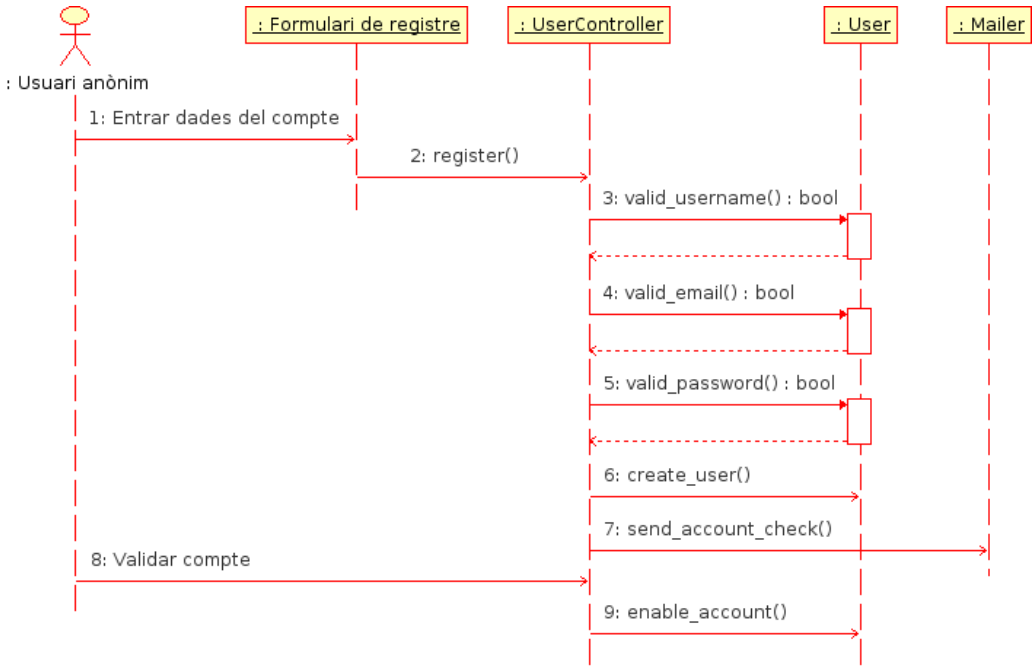

Figura 6.13: Diagrama de seqüència: Crear compte

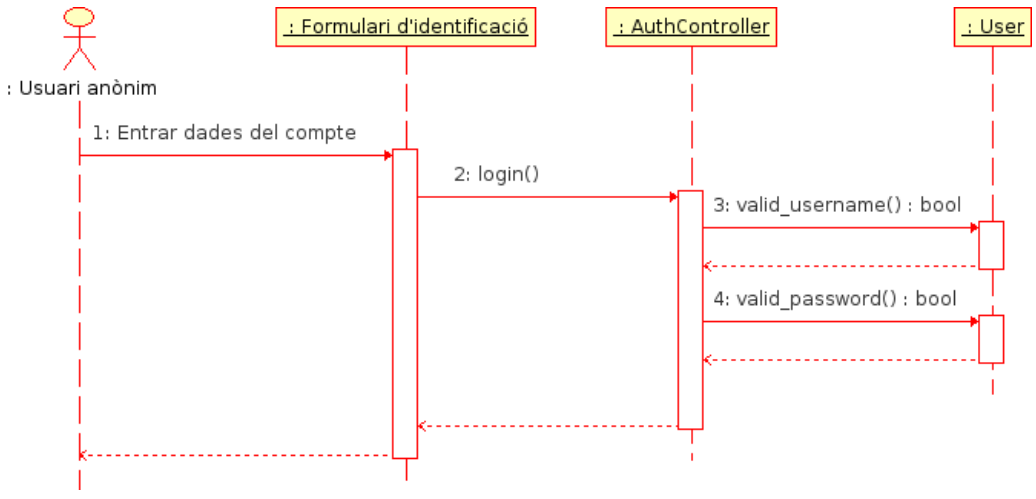

Figura 6.14: Diagrama de seqüència: Identificar-se

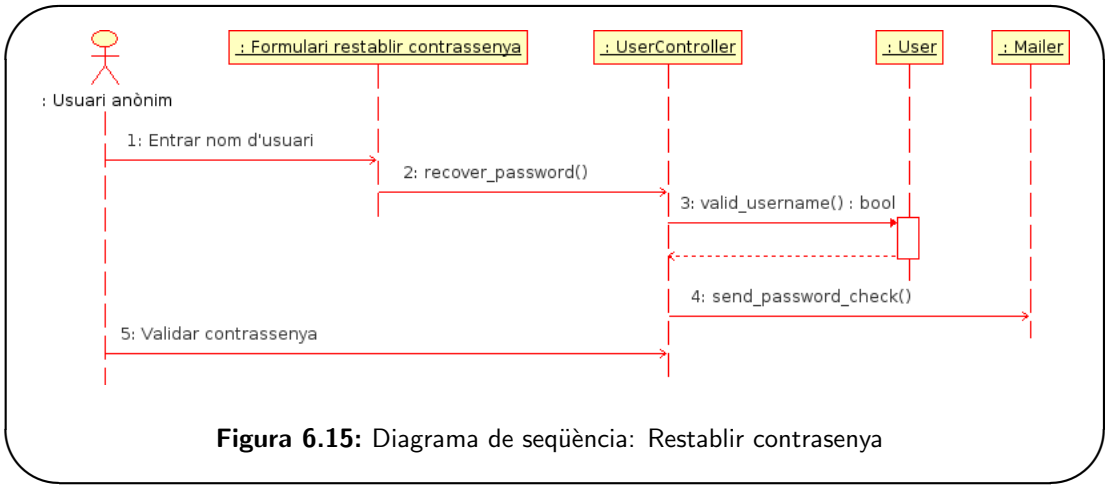

Figura 6.15: Diagrama de seqüència: Restablir contrasenya

Restablir contrasenya (veure figura 6.15)

En primer lloc, l'usuari entrarà el seu nom d'usuari. El sistema validarà que existeixi i obtindrà el correu electrònic associat. Tot seguit enviarà un correu amb la nova contrasenya i un enllaç de confirmació a la direcció de correu obtinguda. L'usuari haurà de seguir l'enllaç proporcionat i d'aquesta manera el compte disposarà d'una nova contrasenya.

Desconnectar-se (veure figura 6.16)

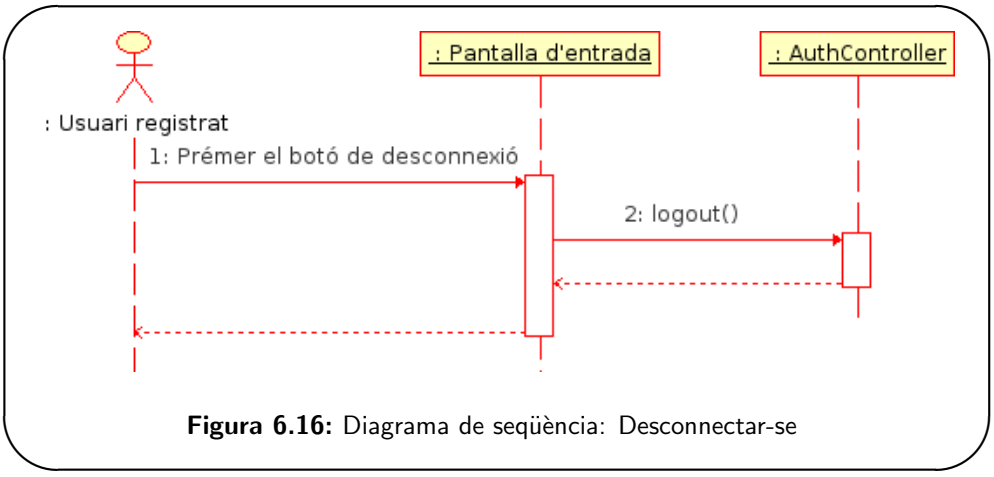

Figura 6.16: Diagrama de seqüència: Desconnectar-se

Aquesta acció es podrà realitzar des de qualsevol pantalla sempre i quan l'usuari estigui identificat. S'haurà de prémer el botó o enllaç de desconnexió i automàticament es passarà a ser un usuari anònim.

Enviar event (veure figura 6.17)

Aquesta acció està formada per dos passos. En primer lloc l'usuari haurà d'entrar l'URL de l'event (normalment la direcció oficial) i el sistema validarà que existeixi a Internet i, si

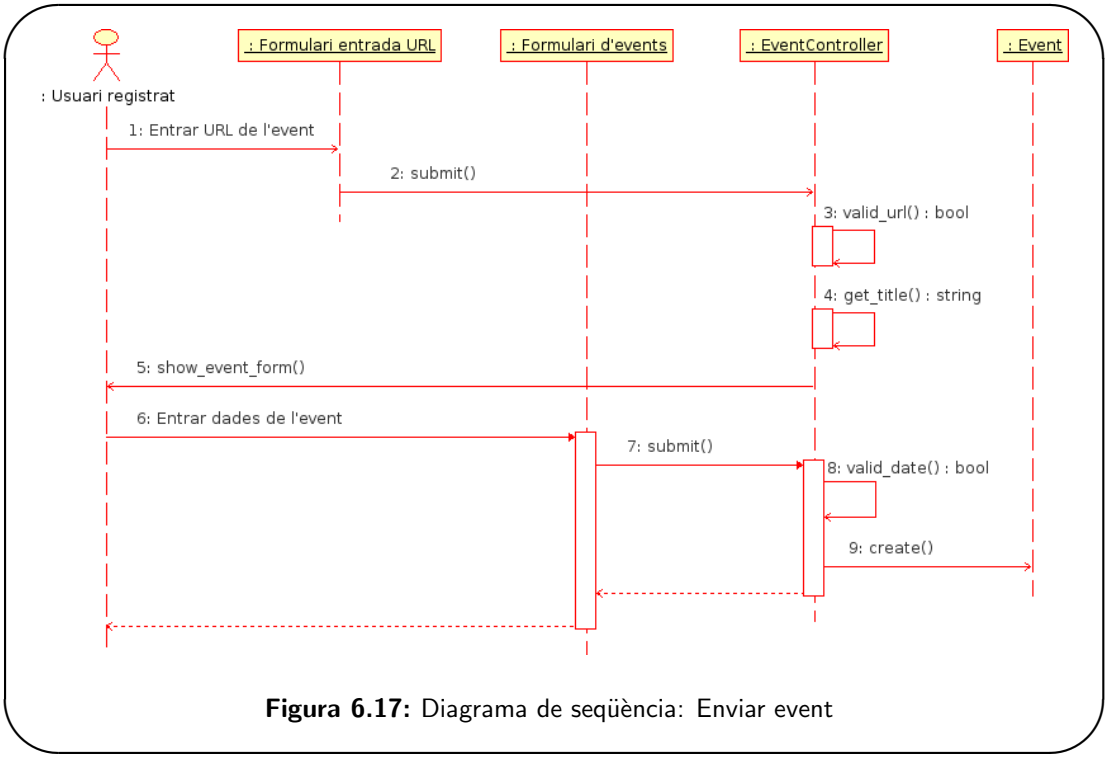

Figura 6.17: Diagrama de seqüència: Enviar event

pot, obtindrà el seu títol. Tot seguit es mostrarà el formulari per donar d'alta l'event, on s'haurà d'entrar totes les dades excepte el títol si s'ha pogut obtenir i no es vol modificar. Un cop entrades, el sistema validarà que són correctes i donarà d'alta l'event.

Modificar event (veure figura 6.18)

Quan un usuari visualitzi el detall d'un event, el sistema comprovarà que té privilegis suficients per modificar-lo. En cas afirmatiu és mostrarà els valors en camps editables. L'usuari entrarà les dades a modificar i les enviarà perquè el sistema les validi i, si són correctes, les modifiqui.

Modificar preferències (veure figura 6.19)

L'usuari entra les noves preferències i les envia al sistema. Aquest valida que siguin correctes i les modifica a la base de dades.

Afegir comentari (veure figura 6.20)

L'usuari entra el comentari a la caixa de text de la pantalla de detall d'un event i l'envia. Tot seguit, el sistema valida que el comentari no contingui etiquetes HTML ni codi Javascript i crea el comentari per desar-lo a la base de dades.

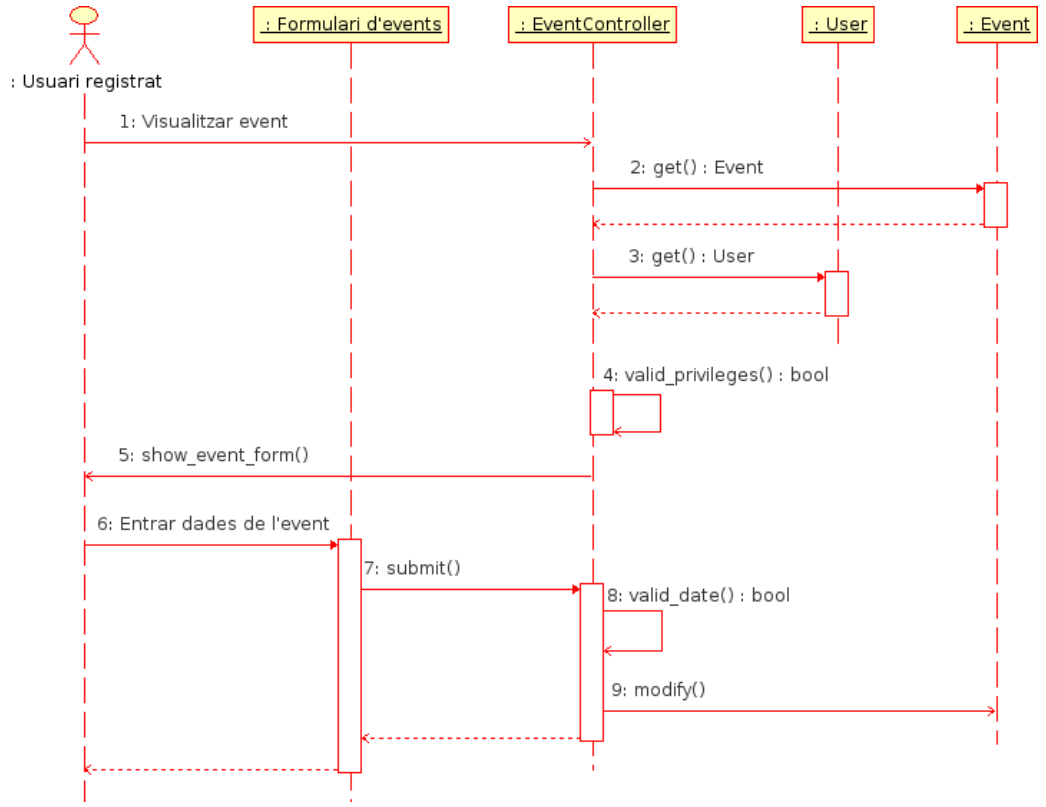

Figura 6.18: Diagrama de seqüència: Modificar event

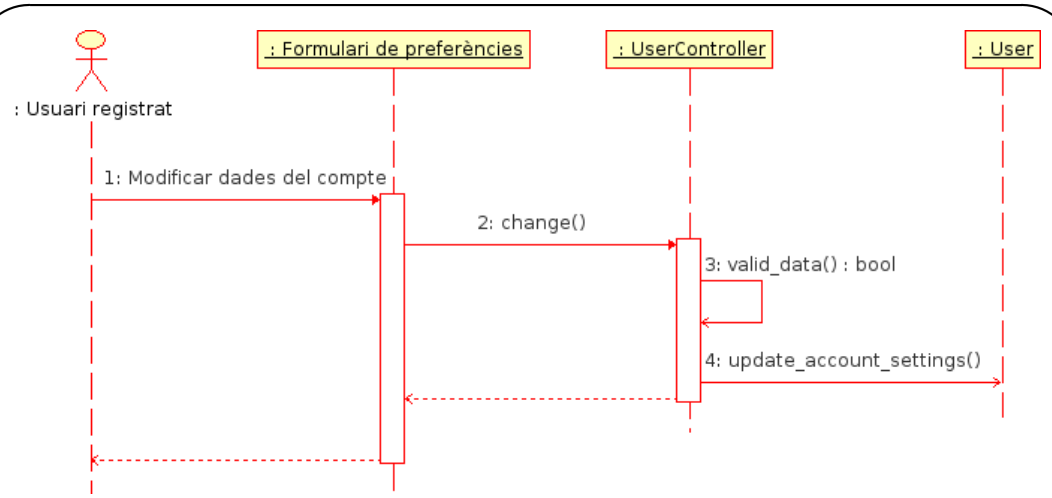

Figura 6.19: Diagrama de seqüència: Modificar preferències

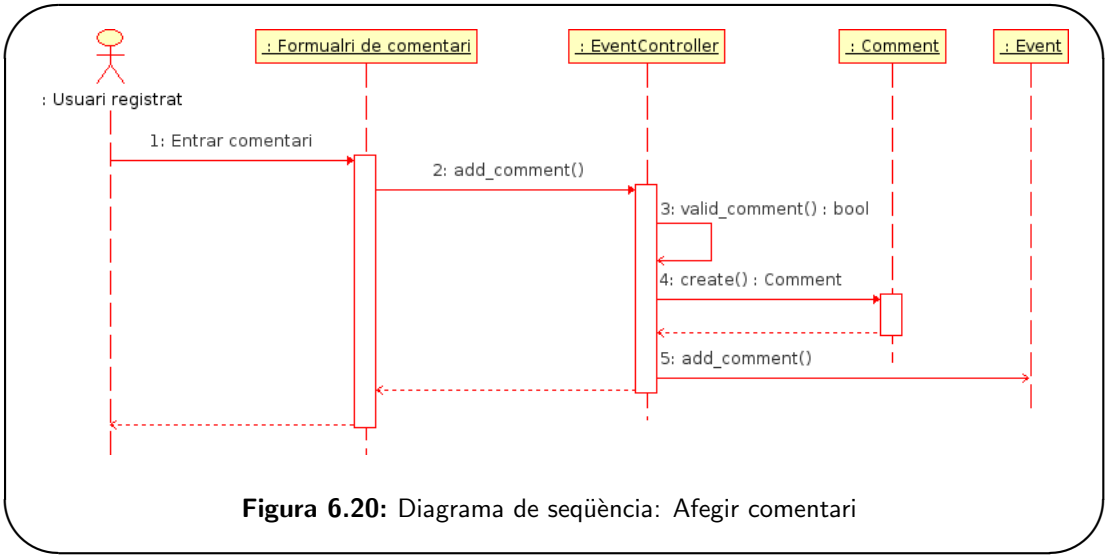

Figura 6.20: Diagrama de seqüència: Afegir comentari

Afegir amic (veure figura 6.21)

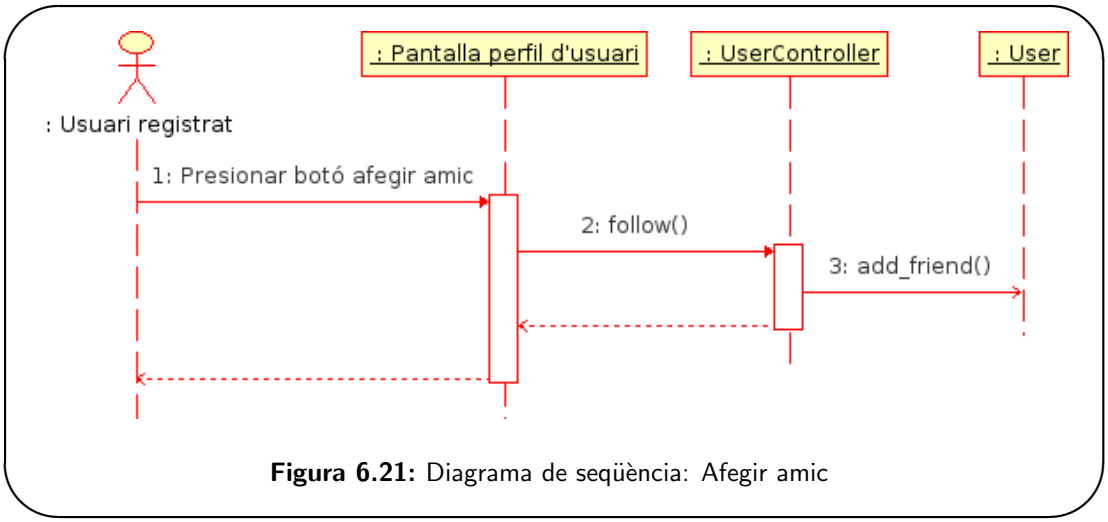

Figura 6.21: Diagrama de seqüència: Afegir amic

L'usuari pressiona l'enllaç per afegir un usuari com amic i el sistema l'afegeix a la seva llista d'amics.

Eliminar amic (veure figura 6.22)

L'usuari pressiona l'enllaç per eliminar un amic i el sistema el treu de la seva llista d'amics.

Afegir event a preferits (veure figura 6.23)

Aquesta acció es podrà realitzar des de qualsevol pantalla on es visualitzi la informació d'un event. L'usuari pressionarà el botó d'afegir event a preferits i d'aquesta manera, el sistema l'afegirà a la seva llista de preferits.

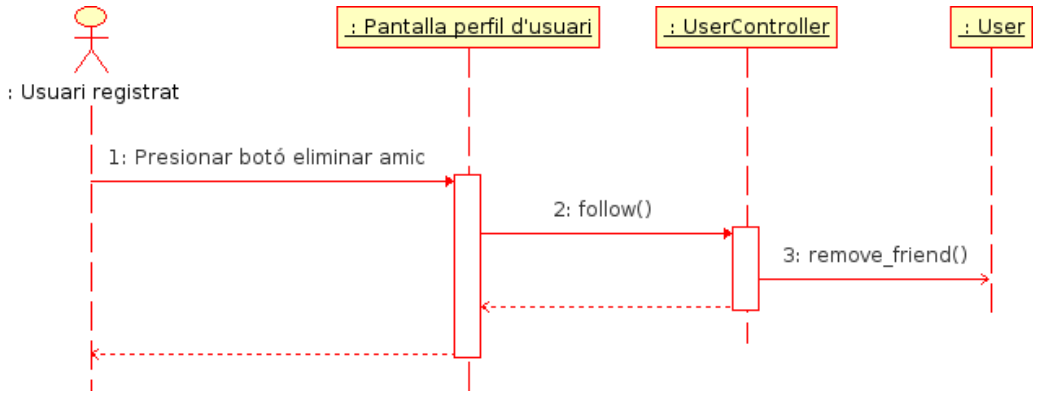

Figura 6.22: Diagrama de seqüència: Eliminar amic

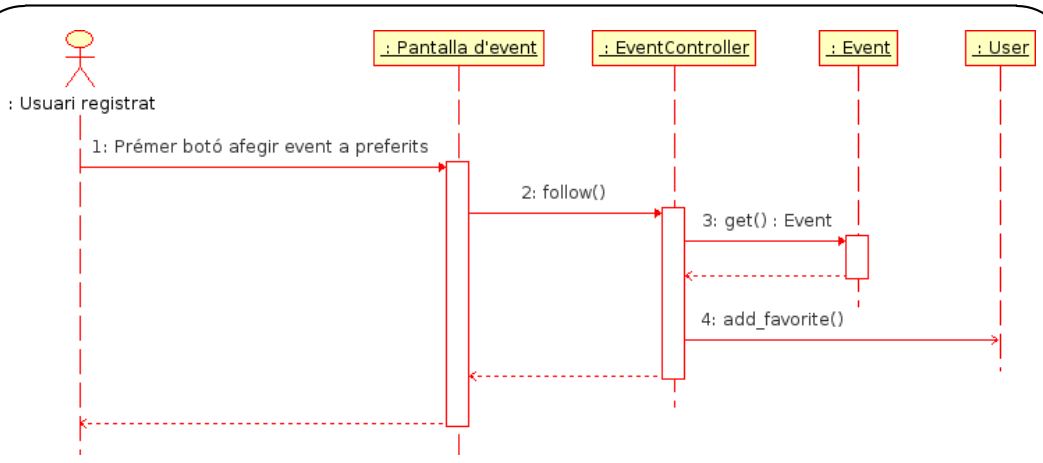

Figura 6.23: Diagrama de seqüència: Afegir event a preferits

Eliminar event de preferits (veure figura 6.24)

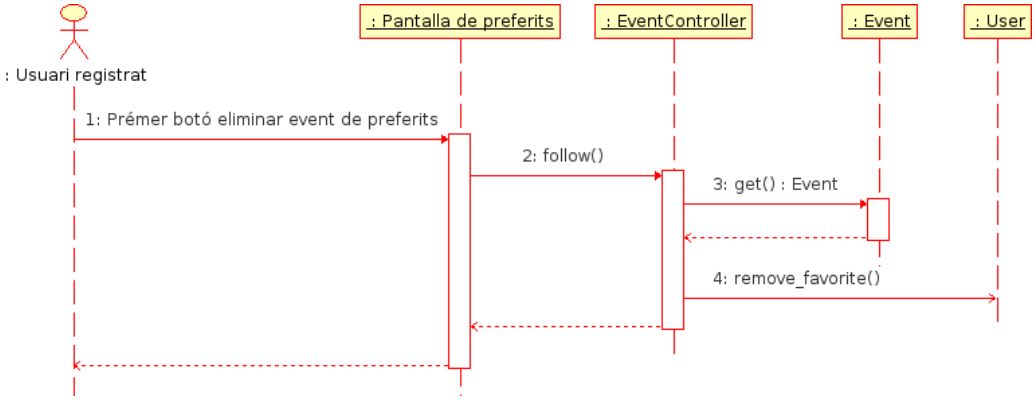

Figura 6.24: Diagrama de seqüència: Eliminar event de preferits

Aquesta acció es podrà realitzar des de qualsevol pantalla on es visualitzi la informació d'un event. L'usuari pressionarà el botó d'eliminar l'event de preferits i, tot seguit, el sistema l'eliminarà de la seva llista de preferits.

Xatejar (veure figura 6.25)

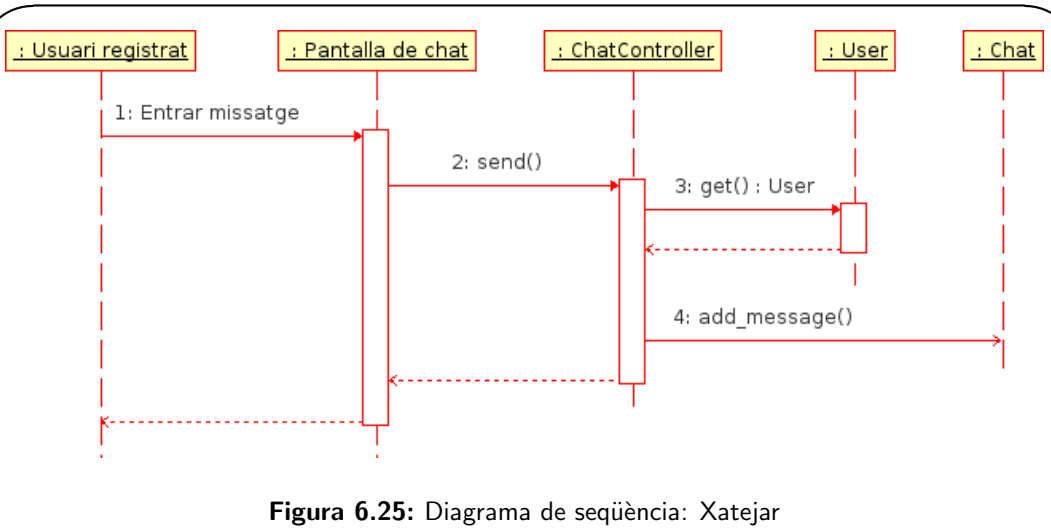

Figura 6.25: Diagrama de seqüència: Xatejar

L'usuari entrarà un missatge a la caixa de text i pressionarà el botó d'enviar. En aquest moment el sistema obtindrà les dades necessàries de l'usuari i guardarà el missatge perquè tots els altres usuaris el puguin veure.

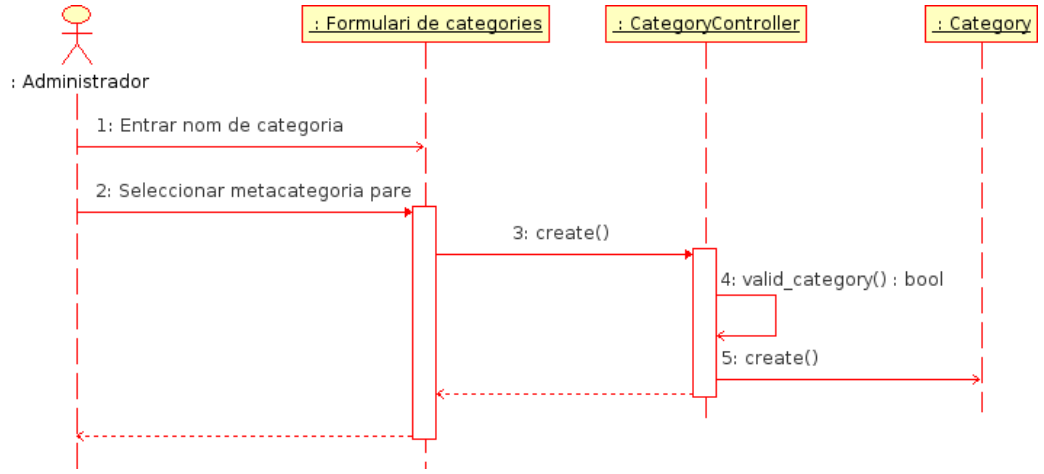

Figura 6.26: Diagrama de seqüència: Afegir categoria

Afegir categoria (veure figura 6.26)

L'administrador entrarà el nom de la categoria que vol crear i, si cal, seleccionarà la metacategoria de la que penjarà. Un cop fet això, el sistema validarà que el nom de la categoria no existeixi i la guardarà a la base de dades.

Eliminar categoria (veure figura 6.27)

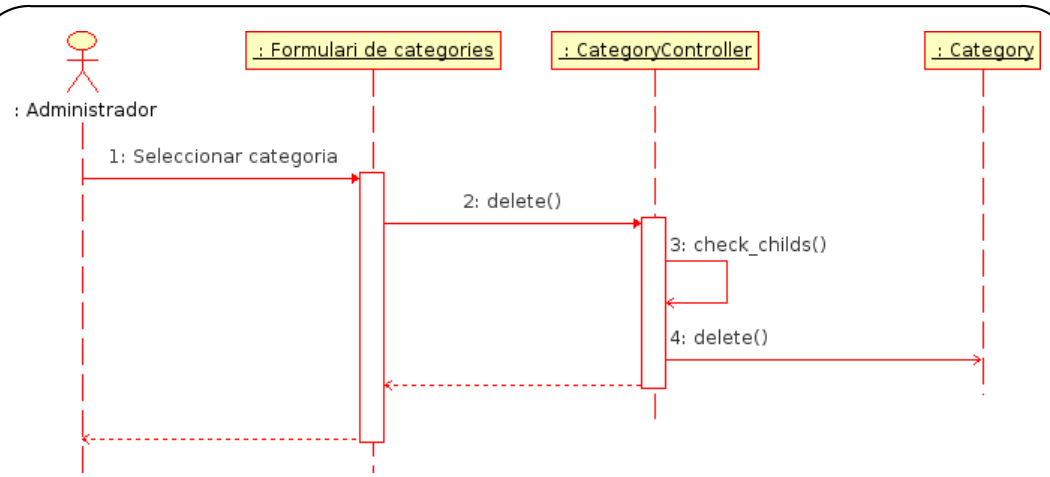

Figura 6.27: Diagrama de seqüència: Eliminar categoria

L'administrador seleccionarà la categoria que desitja eliminar i s'enviarà al sistema. Aquest comprovarà si és una metacategoria i si està buida. En cas afirmatiu la categoria s'eliminarà del sistema.

Modificar categoria (veure figura 6.28)

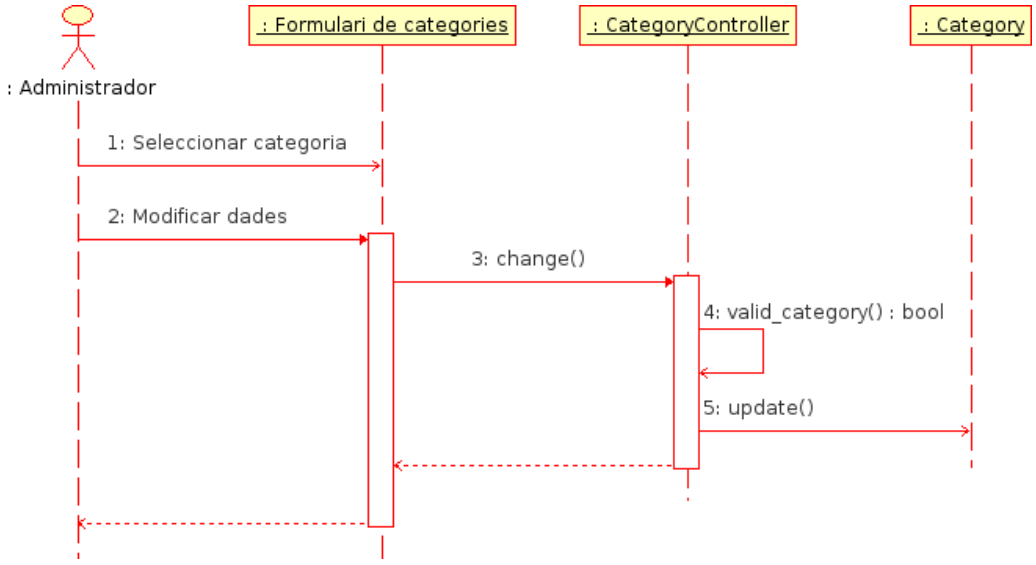

Figura 6.28: Diagrama de seqüència: Modificar categoria

En aquest cas l'administrador seleccionarà la categoria a modificar i entrarà les dades que vulgui canviar. El sistema les rebrà i validarà que el nom de categoria no existeixi. Un cop fet això s'aplicarà els canvis.

Banejar usuari (veure figura 6.29)

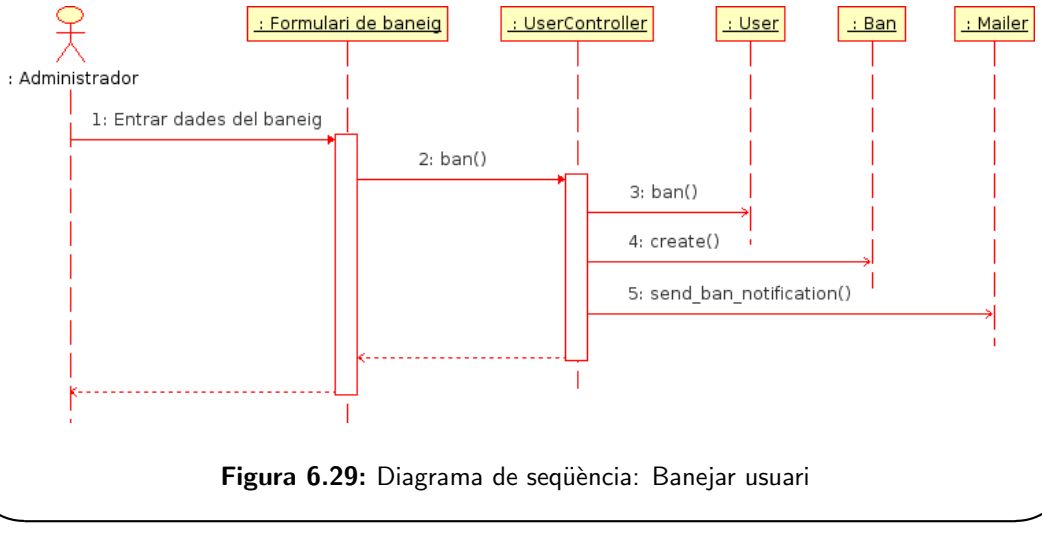

Figura 6.29: Diagrama de seqüència: Banejar usuari

L'administrador entra les dades del baneig i les envia al sistema que, en primer lloc, desactiva el compte de l'usuari i tot seguit li envia un correu electrònic per informar-lo del baneig.

Treure baneig a usuari (veure figura 6.30)

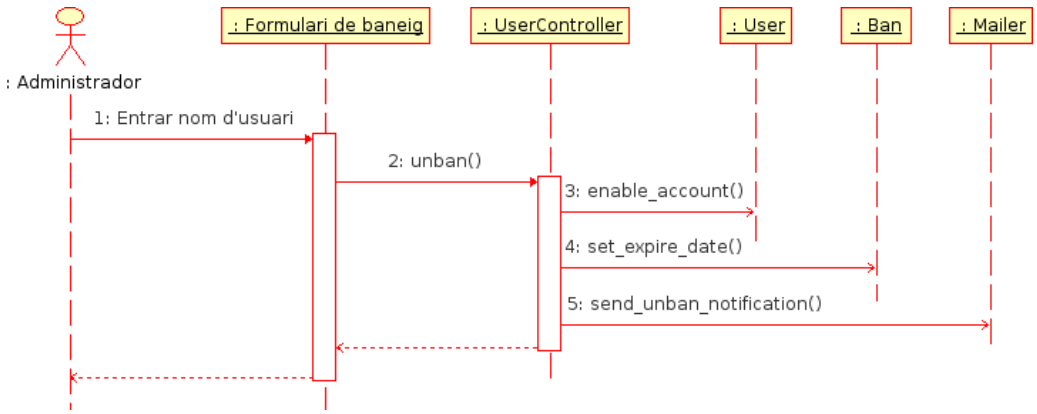

Figura 6.30: Diagrama de seqüència: Treure baneig a usuari

L'administrador entra el nom de l'usuari i l'envia al sistema. Aquest activa el compte de l'usuari i modifica la data d'expiració del baneig perquè ja no tingui efecte.

Enviar notificacions (veure figura 6.31)

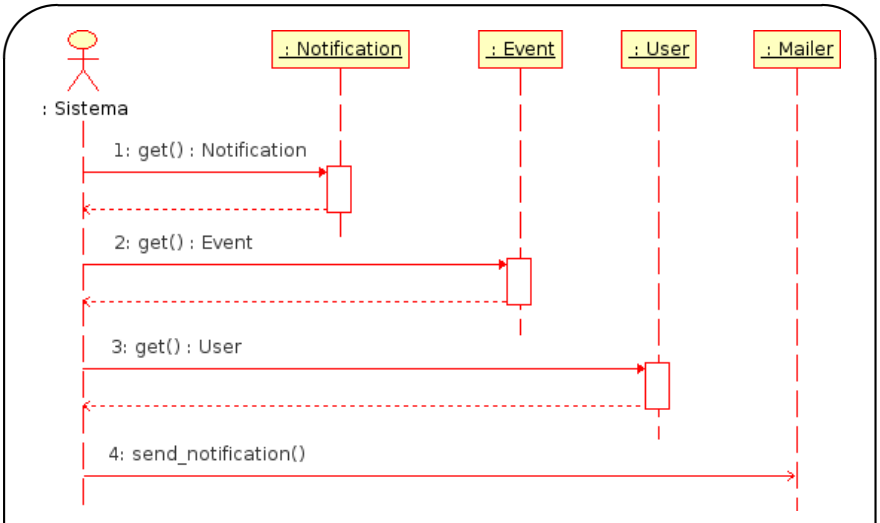

Figura 6.31: Diagrama de seqüència: Enviar notificacions

El sistema obté totes les notificacions pendents per enviar i, per cadascuna, obté les dades de l'usuari i l'event que afecta. Tot seguit envia la notificació a l'adreça de correu electrònic associada al compte de l'usuari.

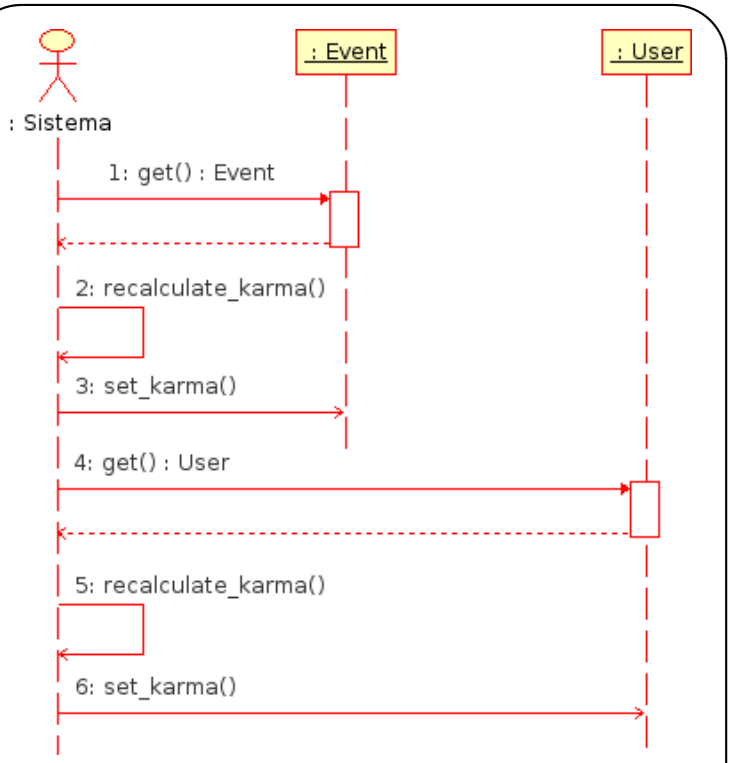

Figura 6.32: Diagrama de seqüència: Recalcular karma

Recalcular karma (veure figura 6.32)

El sistema obté tots els events que encara no s'han acabat i recalcula el seu karma en funció de l'activitat des de l'últim recàlcul. El mateix passa amb els usuaris, en funció del seu comportament el seu karma podrà augmentar, disminuir o mantenir-se.

Generar núvol d'etiquetes (veure figura 6.33)

El sistema obté les etiquetes que s'ha utilitzat més vegades i la seva freqüència. En funció del tamany mínim i màxim de lletra desitjada i el nombre d'aparicions de l'etiqueta més freqüent es calcula el coeficient d'ajustament del tamany. Per cada etiqueta, el sistema calcula el seu tamany utilitzant el coeficient obtingut. Finalment s'ordena alfabèticament les etiquetes i es pinta el núvol a la pantalla.

6.6 Anàlisi de classes

6.6.1 Identificació de responsabilitats i atributs

A continuació es definirà les responsabilitats i atributs de les classes que s'ha identificat en les tasques anteriors.

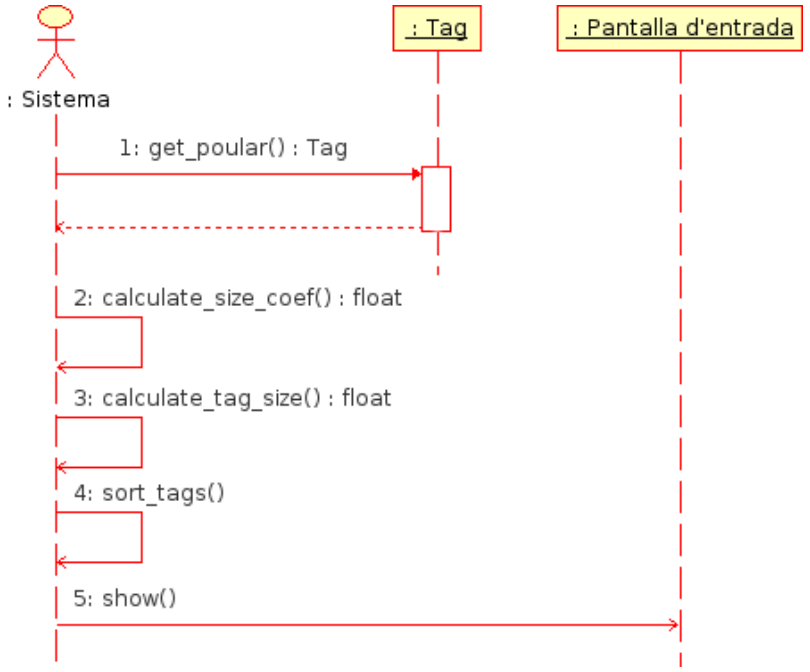

Figura 6.33: Diagrama de seqüència: Generar núvol d'etiquetes

Classe AuthController

Mètodes

- login(): Permet identificar a un usuari.
- logout(): Permet desconnectar a un usuari.

Classe Ban

Atributs propis

- Integer id: Codi únic que identifica el baneig en el sistema.
- Unicode type: Tipus de baneig
- Unicode text: Text que identifica el baneig segons el tipus.
- Datetime date: Data de creació del baneig.
- Datetime expire: Data d'expiració del baneig.
- Unicode comment: Comentari sobre el baneig.

Mètodes

- create(): Permet gravar una baneig a la base de dades.
- set_expire_date(): Permet modificar la data d'expiració del baneig.

Classe Category

Atributs propis

- Integer id: Codi únic que identifica la categoria en el sistema.
- Unicode name: Nom de la categoria.

Atributs agregats

- Category parent: Metacategoria de la qual és filla.

Mètodes

- create(): Permet gravar una categoria a la base de dades. Estableix la relació entre la categoria creada i la metacategoria pare.
- delete(): Permet eliminar una categoria de la base de dades.
- update(): Permet modificar les dades d'una categoria de la base de dades.

Classe CategoryController

Mètodes

- create(): Permet crear una nova categoria amb les dades passades.
- valid_category(): Permet validar les dades d'una categoria.
- delete(): Permet eliminar una categoria.
- check_childs(): Permet comprovar si una categoria és metacategoria, és a dir, si te fills.
- change(): Permet modificar les dades d'una categoria.

Classe ChatController

Mètodes

- send(): Permet enviar un missatge al xat.

Classe Comment

Atributs propis

- Integer id: Codi únic que identifica el comentari en el sistema.
- Datetime date: Data en la que s'ha creat el comentari.
- Integer ip: Direcció IP des de la que s'ha creat el comentari.
- Integer order: Ordre del comentari respecte tots els comentaris de l'event.
- Unicode content: Contingut del comentari.

Atributs agregats

- Event event: Event al que pertany el comentari.
- User user: Usuari que ha creat el comentari.

Mètodes

- create(): Permet gravar un comentari a la base de dades. Estableix la relació entre el comentari creat, l'event i l'usuari.

Classe Event

Atributs propis

- Integer id: Codi únic que identifica l'event en el sistema.
- Datetime start_date: Data d'inici de l'event.
- Datetime end_date: Data de finalització de l'event.
- Datetime date: Data en la que s'ha creat l'event.
- Unicode status: Estat de l'event.
- Integer randkey: Nombre aleatori generat al crear l'event. S'utilitza per validar l'autenticitat del vots.
- Integer votes: Nombre de vots que ha rebut l'event.
- Float karma: Puntuació de l'event.
- Datetime modified: Data de l'última modificació.
- Integer ip: Direcció IP des de la que s'ha creat l'event.
- Unicode url: URL de la pàgina oficial de l'event.
- Unicode title: Títol de l'event.
- Unicode content: Descripció/Resum de l'event.
- Unicode tags_list: Llista d'etiquetes.

Atributs agregats

- User user: Usuari que ha enviat l'event.
- Category category: Categoria a la que pertany l'event.

Mètodes

- get(): Permet obtenir un event.
- set_votes(): Permet modificar el nombre de vots d'un event.
- set_karma(): Permet modificar el karma d'un event.
- create(): Permet gravar un event a la base de dades. Estableix les relacions entre l'event, l'usuari que l'envia i la categoria a la que pertany.
- modify(): Permet modificar les dades d'un event.
- add_comment(): Permet afegir un comentari a un event.

Classe EventController

Mètodes

- vote(): Permet votar un event.
- validate_control_key(): Permet validar que l'usuari que vota l'event és el mateix que ha carregat la pàgina.
- valid_url(): Permet validar que l'URL entrada és correcta i existeix.
- submit(): Permet afegir un event amb les dades entrades.
- get_title(): Permet obtenir el títol de l'event a partir de l'URL entrada.
- valid_date(): Permet validar la data entrada.
- valid_privileges(): Permet validar que l'usuari que modifica l'event té els privilegis per fer-ho.
- add_comment(): Permet afegir un comentari a un event.
- valid_comment(): Permet validar que el comentari entrat és correcte.
- follow(): Permet afegir un event a la llista de preferits.

Classe LanguageController

Mètodes

- change(): Permet modificar l'idioma de la pàgina.

Classe Mailer

Mètodes

- send_account_check(): Permet enviar un correu de confirmació per la creació d'un compte d'usuari.
- send_password_check(): Permet enviar un correu de confirmació per restablir el password del compte d'un usuari.
- send_ban_notification(): Permet enviar un correu informatiu sobre el baneig d'un compte.
- send_unban_notification(): Permet enviar un correu informatiu sobre la retirada del baneig d'un compte.
- send_notification(): Permet enviar una notificació sobre un event.

Classe MapController

Mètodes

- get_events(): Permet obtenir els events segons una zona geogràfica concreta.

Classe Notification

Atributs propis

- Integer id: Codi únic que identifica la notificació en el sistema.
- Unicode type: Tipus de notificació.
- Datetime date: Data en la que s'ha d'enviar la notificació.
- Unicode status: Estat de la notificació.

Atributs agregats

- User user: Usuari que ha de rebre la notificació.
- Event event: Event que produeix la notificació.

Mètodes

- get(): Permet obtenir una notificació.

Classe RSS2Controller

Mètodes

- show(): Permet mostrar events mitjançant RSS2.

Classe SearchController

Mètodes

- search(): Permet cercar i mostrar events en funció dels paràmetres passats.

Classe Tag

Atributs propis

- Integer id: Codi únic que identifica l'etiqueta.
- Unicode word: Paraula que conté l'etiqueta.

Classe User

Atributs propis

- Integer uid: Codi únic que identifica l'usuari en el sistema.
- Unicode username: Nom d'usuari.
- Unicode password: Contrasenya de l'usuari.

- Datetime date: Data de creació del compte de l'usuari.
- Datetime modified: Data de l'última modificació en el compte de l'usuari.
- Datetime validated_date: Data de la validació del compte.
- Unicode name: Nom real de l'usuari.
- Unicode email: Correu electrònic de l'usuari.
- Unicode username_register: Nom d'usuari inicial del compte.
- Unicode email_register: Correu electrònic amb el que s'ha creat el compte.
- Integer ip: Direcció IP des la que s'ha creat el compte.
- Float karma: Puntuació de l'usuari.
- Unicode public_info: Informació personal de l'usuari.
- Unicode website: Pàgina personal de l'usuari.
- Unicode profile_views: Nombre de visites al perfil de l'usuari.

Mètodes

- get(): Permet obtenir un usuari.
- valid_username(): Permet validar que el nom d'usuari és correcte.
- valid_email(): Permet validar que el correu electrònic és correcte.
- create_user(): Permet crear un compte d'usuari.
- enable_account(): Permet activar un compte d'usuari.
- valid_password(): Permet validar que una contrasenya és vàlida i compleix els requisits mínims de seguretat.
- update_account_settings(): Permet modificar les dades d'un compte d'usuari.
- add_firend(): Permet afegir un usuari com amic.
- remove_friend(): Permet eliminar un usuari de la llista d'amics.
- add_favorite(): Permet afegir un event a la llista de preferits.
- remove_favorite(): Permet eliminar un event de la llista de preferits.
- ban(): Permet desactivar el compte d'un usuari.
- set_karma(): Permet modificar el karma d'un usuari.

Classe UserController

Mètodes

- register(): Permet crear un compte d'usuari a partir de les dades entrades.
- recover_password(): Permet recuperar la contrasenya d'un compte d'usuari.
- change(): Permet modificar un compte d'usuari amb les dades passades.
- valid_data(): Permet validar les dades d'un compte d'usuari.
- follow(): Permet afegir com amic a l'usuari passat.
- ban(): Permet banejar un compte d'usuari.

- unban(): Permet treure el baneig d'un compte d'usuari.

Un cop s'ha identificat totes les entitats participants es podria crear el diagrama de classes inicial (veure figura 6.34).

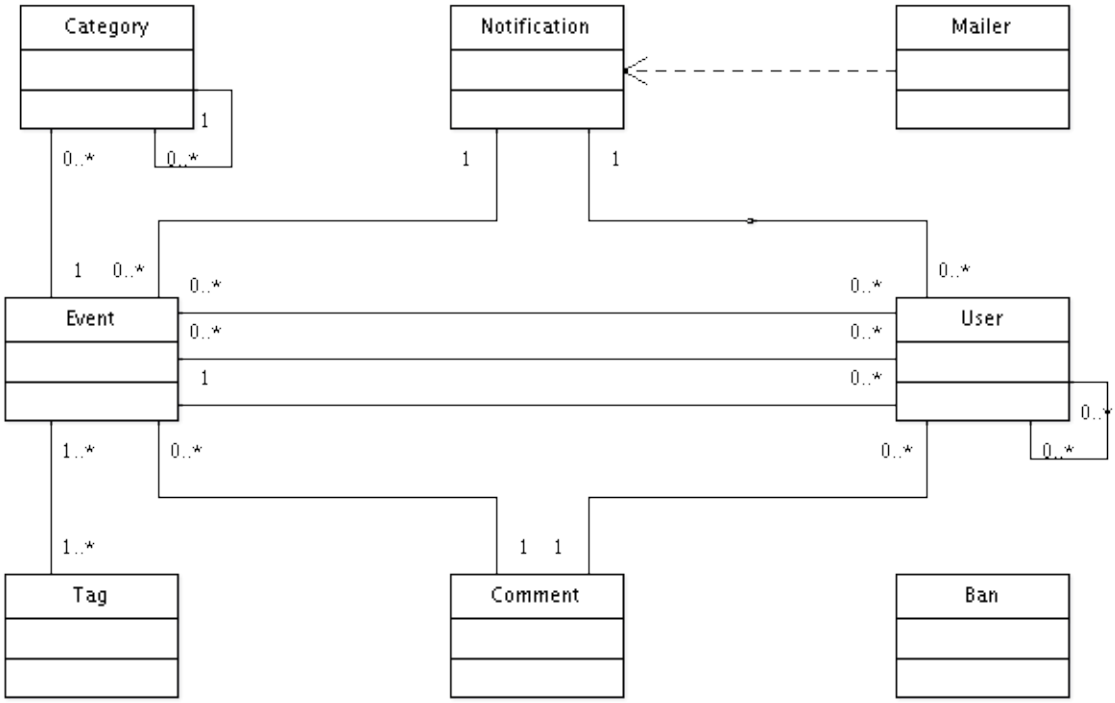

Figura 6.34: Diagrama de classes inicial

6.7 Elaboració del model de dades

6.7.1 Elaboració del model conceptual de dades

En aquest punt s'ha identificat les entitats que queden dins l'àmbit del sistema d'informació i les relacions existents entre elles. També s'identifiquen aquelles entitats de dades que no formen part del model, però que estan relacionades amb alguna entitat del mateix. El resultat queda reflexat en el model entitat relació de la figura 6.35.

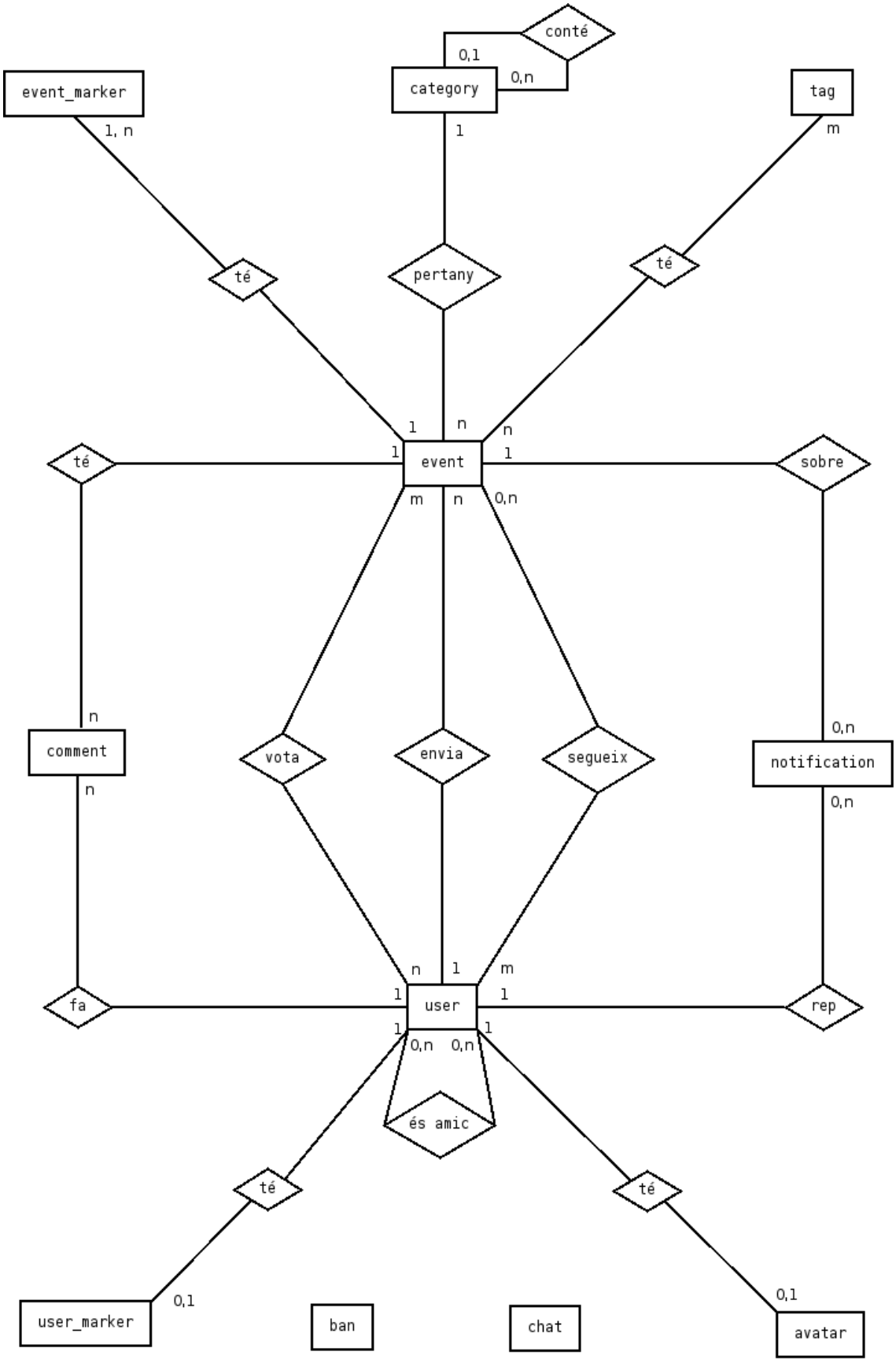

Figura 6.35: Diagrama entitat relació

6.8 Interfícies d'usuari

6.8.1 Especificació dels principis generals de la interfície

■ Principis generals

Per la creació de la interfície s'utilitzarà DHTML[2] amb la combinació de XHTML, Javascript i CSS.

L'aplicació mantindrà el criteri en tots els elements de la interfície, és a dir, formularis, botons, marcs, colors, fonts....

■ Instal·lació

No serà necessària cap instal·lació especial en el sistema client, únicament disposar d'un navegador web compatible amb els estàndards; es recomana la utilització del navegador Mozilla Firefox en la versió 2 o superior. Cal comentar que el navegador Internet Explorer no compleix amb els estàndards establerts i, per aquest motiu, la interfície no està dissenyada pel seu ús. De totes maneres s'està treballant per adaptar-la a aquest navegador.

La resolució de pantalla no haurà de ser d'un tamany fix, però si serà recomanable una resolució de 1024x768 píxels o superior.

■ Composició de pantalles i ubicació d'elements

La interfície tindrà una estructura similar en cadascuna de les pantalles. Aquesta estructura (veure figura 6.36) estarà composta de les següents parts:

- Capçalera.
- Menú de navegació.
- Contingut de la pàgina.
- Peu.

6.8.2 Especificació de formats individuals d'interfície

A continuació es mostra les principals pantalles de l'aplicació.

■ Pantalla principal (veure figura 6.37)

La portada ha de generar una bona primera impressió als usuaris, que en molts de casos pot ser el factor determinant perquè un usuari es quedi o marxi de l'aplicació. Cal que sigui senzilla però agradable i que mostri algunes pinzellades del funcionament del sistema.

[2] Col·lecció de tecnologies utilitzades juntes per crear pàgines web interactives i dinàmiques utilitzant una combinació d'un llenguatge de marques estàtic, un llenguatge interpretat en el costat client, un llenguatge de presentació i la jerarquia d'objectes del DOM.

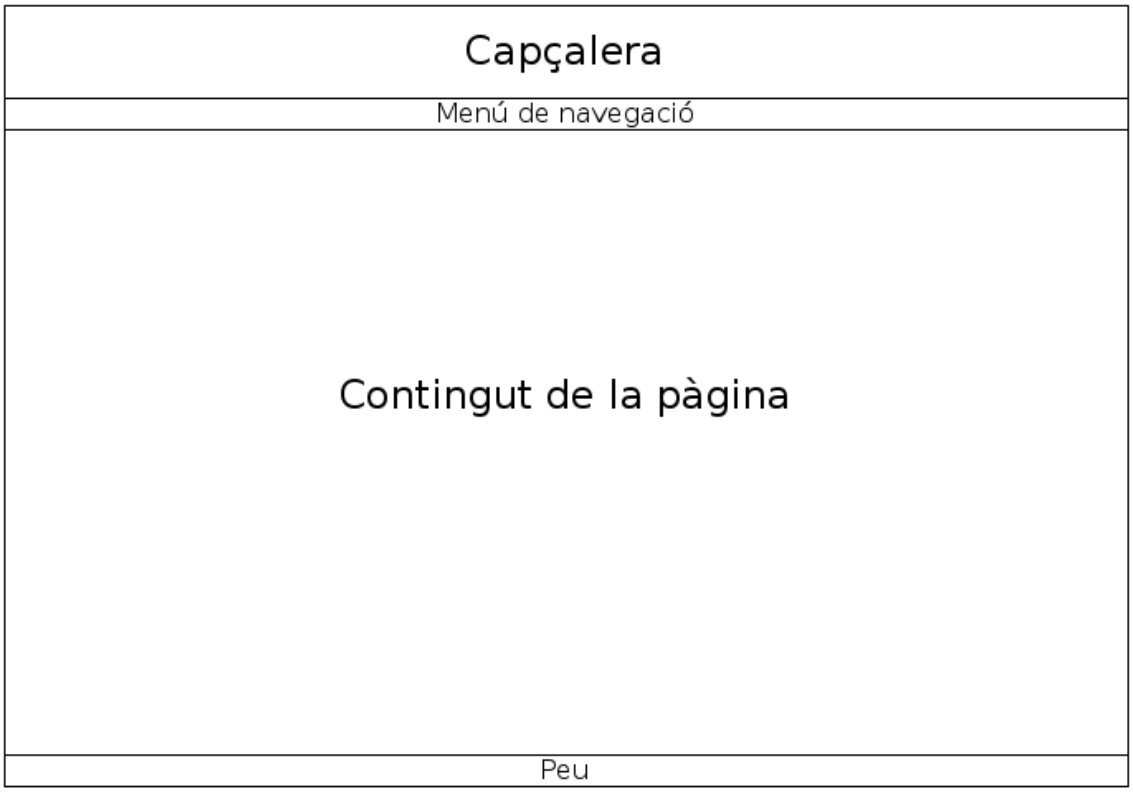

Figura 6.36: Estructura de les pantalles

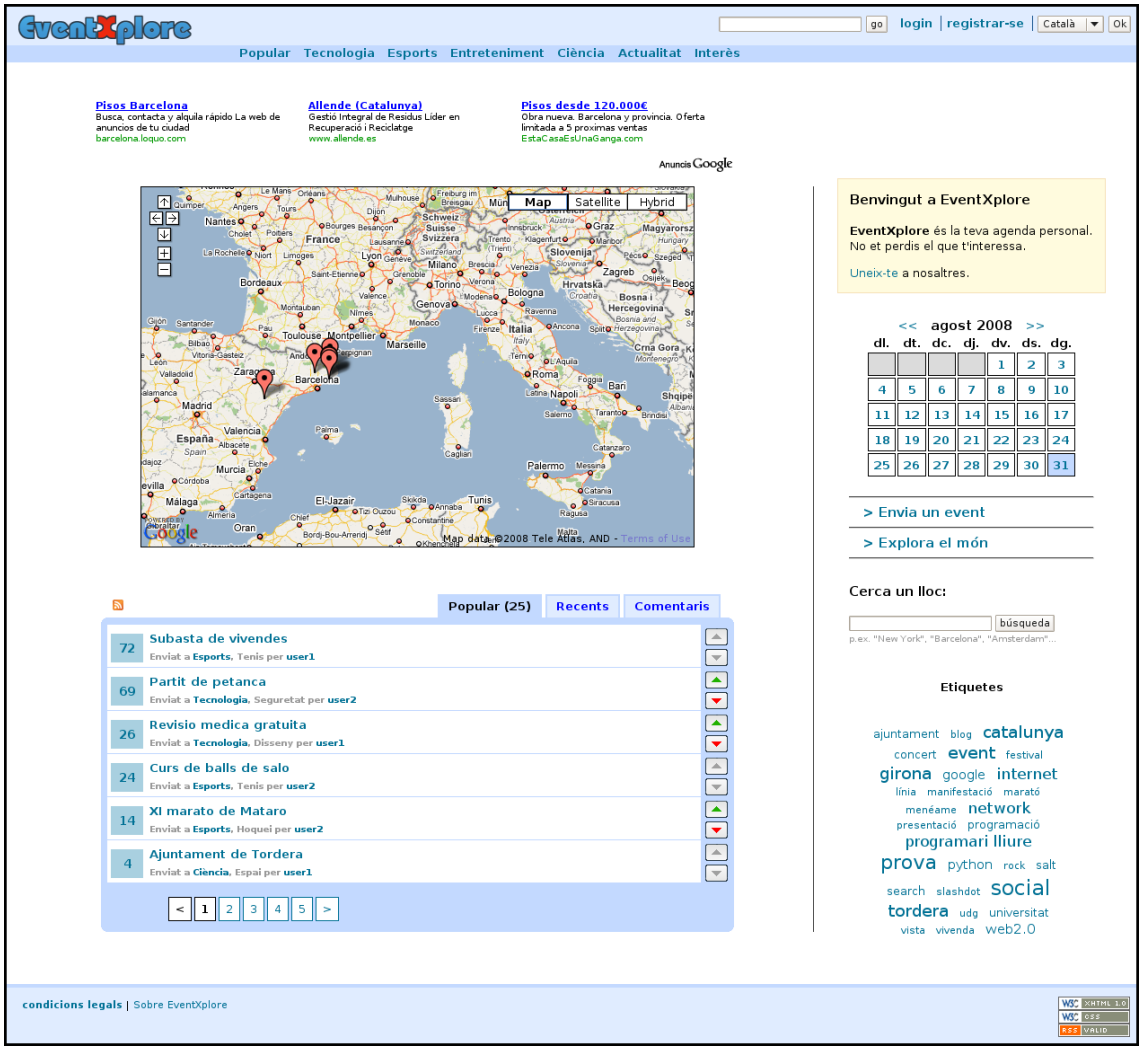

Figura 6.37: Pantalla principal

Les principals funcionalitats són:

- **Exposició d'un punt fort del sistema:** En aquest cas es tracta de la navegació mitjançant el mapa. Per tant, disposarà d'un petit mapa amb els events més populars que permetrà iniciar a l'usuari en el funcionament de l'aplicació.
- **Facilitat per ingressar en el sistema:** Caldrà que els enllaços d'indentificació i de registre estiguin situats en una zona de fàcil accés.
- **Ràpid accés a altres funcionalitats:** Haurà de permetre accedir a les diferents funcionalitats de l'aplicació de manera intuïtiva. Per això es disposarà de l'enviament d'events, del menú de navegació per metacategories, el calendari, el cercador i el núvol d'etiquetes.

■ Pantalla d'exploració (veure figura 6.38)

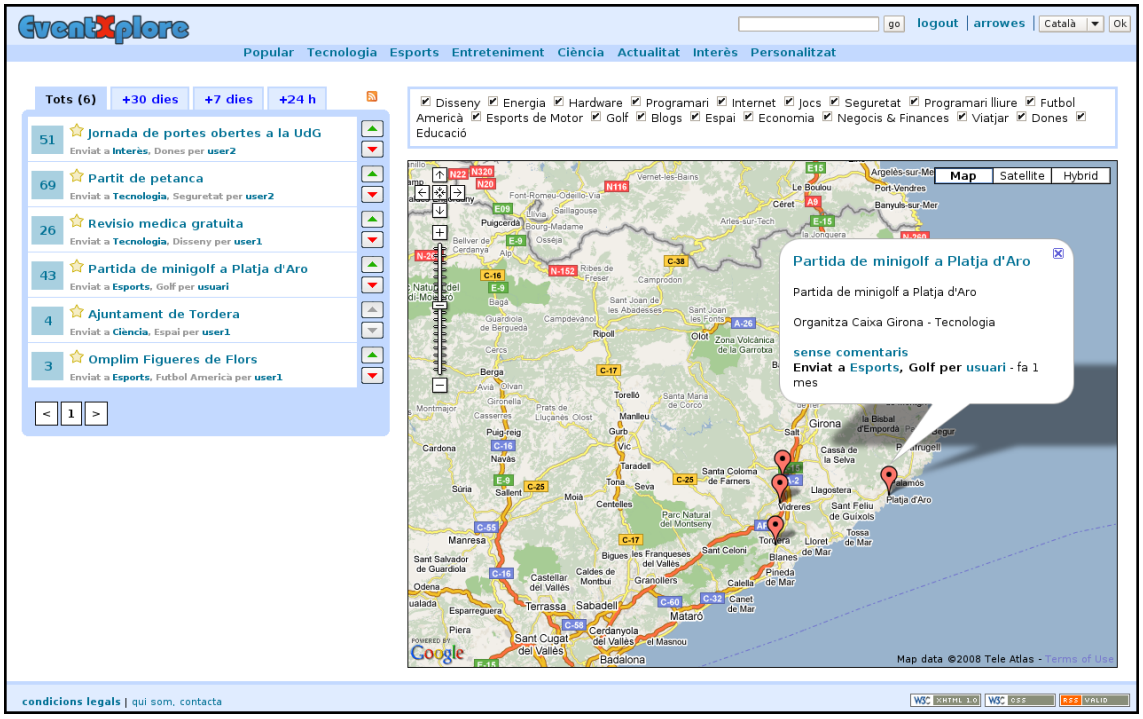

Figura 6.38: Pantalla d'exploració

Aquesta pantalla és el plat fort de l'aplicació, per tant ha de permetre accedir a qualsevol event del sistema de forma fàcil.

Les principals funcionalitats són:

- **Visualització dels events més populars:** El sistema premia els events més populars entre els usuaris mostrant-los en primer lloc. Per poder veure un event menys popular de la mateixa zona, caldrà utilitzar el paginador o les funcions de navegació del mapa.
- **Visualització de qualsevol event del sistema:** Gràcies a la navegació a través del mapa l'usuari ha de poder arribar a qualsevol event, ja sigui desplaçant el mapa

cap als costats, ampliant-lo o disminuint-lo.

- **Possibilitat de filtrar informació:** L'usuari en tot moment ha de poder escollir quina informació vol visualitzar. Així doncs es proporcionarà els següents filtres:
  - Àrea del mapa visualitzada.
  - Interval de temps.
  - Categories i metacategories.
- **Visualització de la principal informació:** Es proporcionarà una previsualització dels events mostrats a la llista lateral. En cas de voler veure més informació, l'usuari tindrà l'opció d'obrir una finestra informativa o d'accedir directament al detall de l'event desitjat.

■ Pantalla enviament d'events (veure figura 6.39)

Aquesta pantalla ha de permetre a l'usuari donar d'alta un event al sistema. Les dades que cal entrar són:

- Títol
- Etiquetes
- Descripció
- Categoria
- Data d'inici
- Data de finalització
- Localització

■ Pantalla detall d'event (veure figura 6.40)

Aquesta pantalla ha de permetre a l'usuari obtenir tota la informació disponible sobre l'event detallat.

Les principals funcionalitats són:

- **Visualització de tota la informació disponible sobre l'event:** La informació de l'event haurà d'estar situada correctament per permetre a l'usuari trobar-la fàcilment. Per tant, s'haurà de mostrar les dades més importants, com el títol, data, descripció, situació... en llocs molt visibles.
- **Proporcionar informació addicional:** Cal tenir en compte que un usuari content té més possibilitats de tornar que un que no ho estigui. Fins i tot és possible que recomani l'aplicació a altres persones. Per tant, tots els serveis addicionals que s'ofereixin augmentaran les possibilitats de que això passi. Alguns d'aquest serveis són:
  - Comunicació amb altres usuaris mitjançant comentaris.
  - Visualització dels usuaris que han votat l'event.
  - Visualització dels usuaris que han afegit l'event a preferits.
  - Informació del temps que falta per començar l'event.

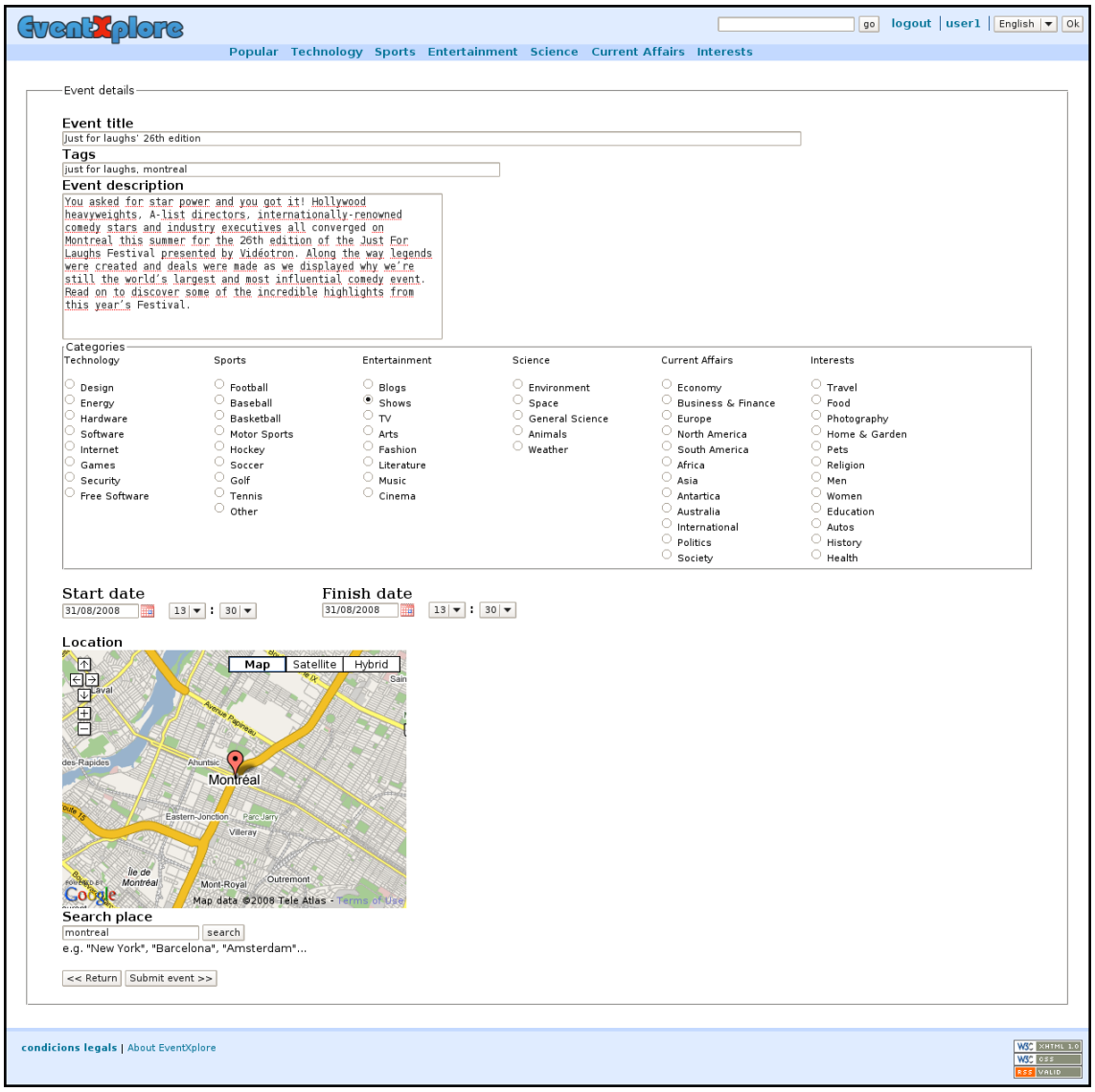

Figura 6.39: Pantalla enviament d'events

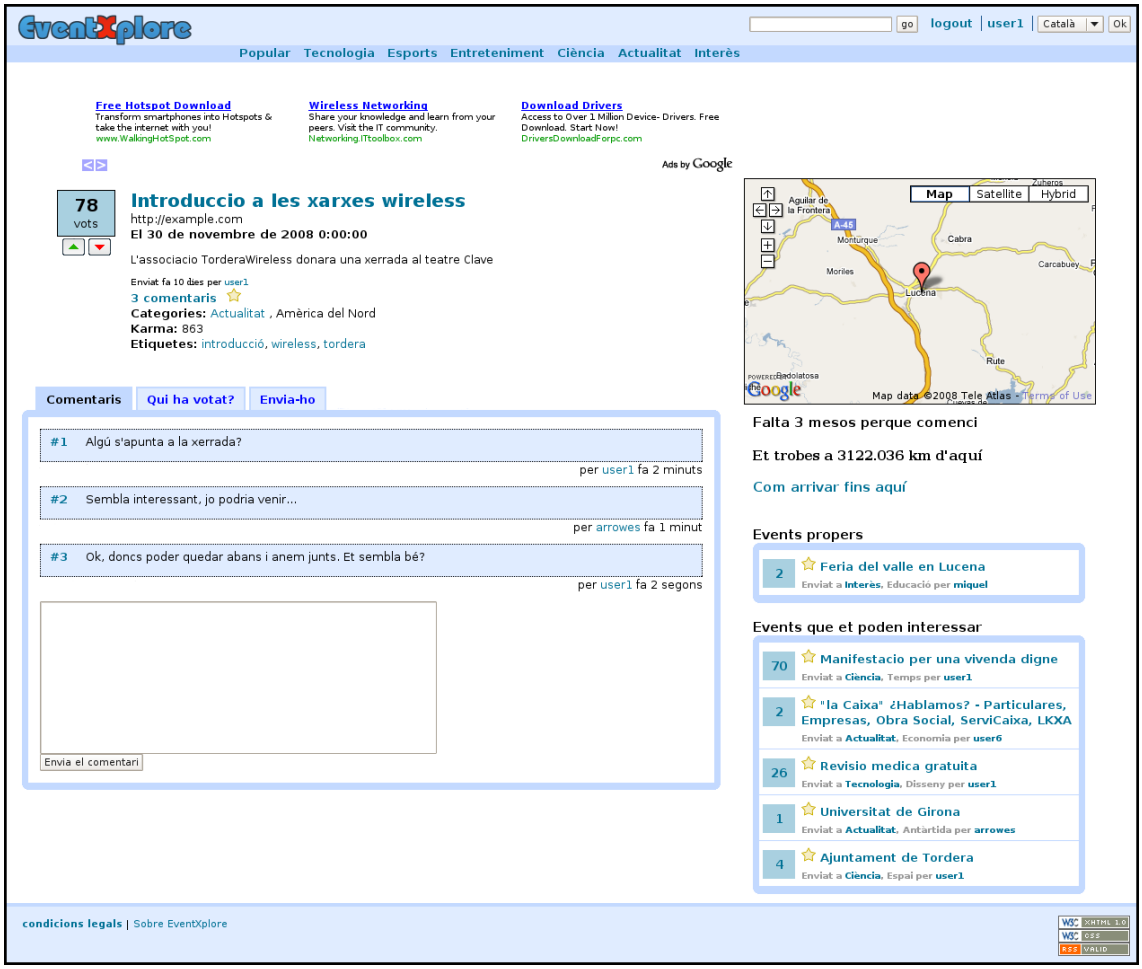

Figura 6.40: Pantalla de detall d'event

– Informació de la distància a que es troba l'event de l'usuari.
– Informació de la ruta a seguir per arribar fins a l'event.
– Visualització d'events propers.

■ Pantalla generació de ruta (veure figura 6.41)

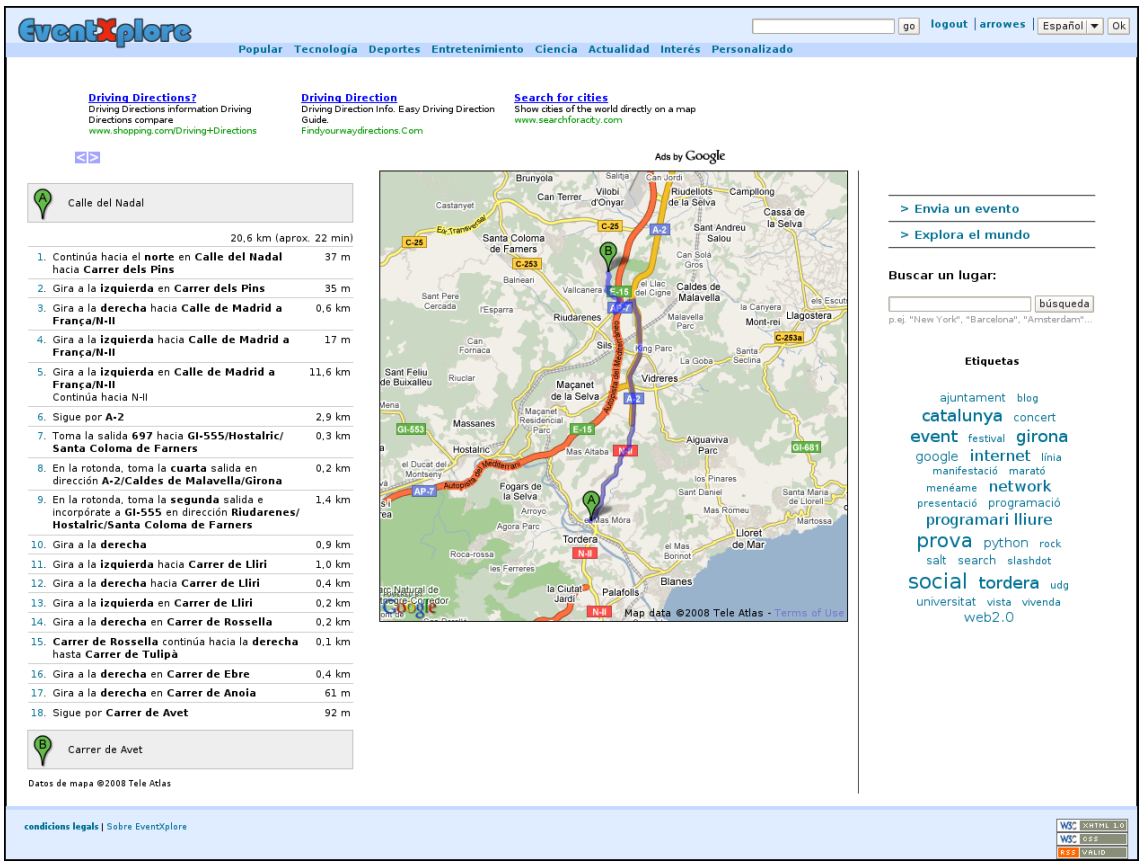

Figura 6.41: Pantalla generació de ruta

Aquesta pantalla mostra la ruta que ha de seguir l'usuari per arribar fins on tindrà lloc l'event. Com és lògic, només es podrà veure la pantalla si l'usuari ha entrat en el seu perfil la seva localització.

■ Pantalla perfil d'usuari (veure figura 6.42)

Aquesta pantalla ha de permetre a l'usuari veure la principal informació d'un altre usuari o d'ell mateix.

Les principals funcionalitats són:

- **Visualització de les principals dades:** Es mostrarà les dades principals de l'usuari, com podria ser el nombre d'events enviats, el seu karma, una breu descripció sobre ell, la seva localització...

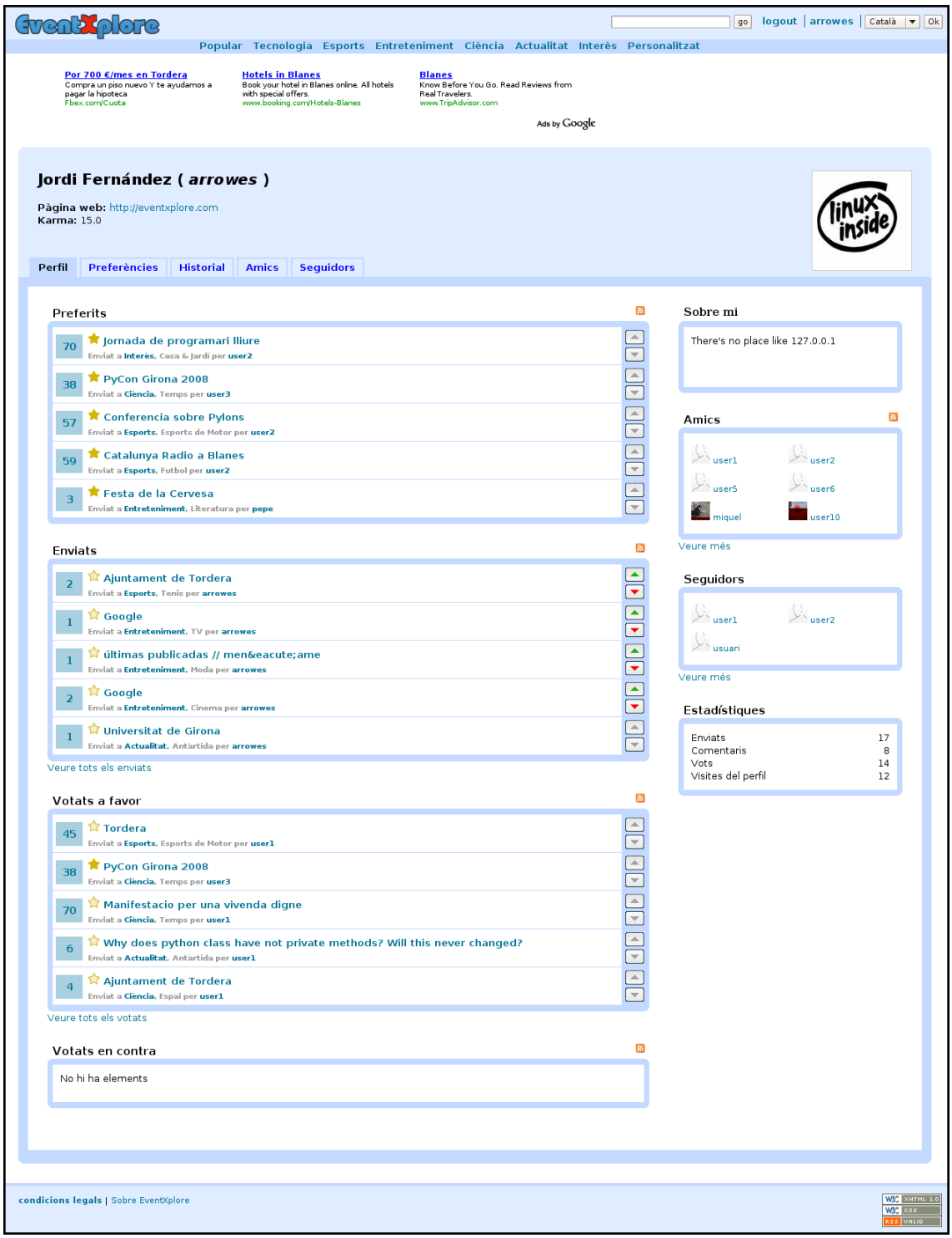

Figura 6.42: Pantalla de perfil d'usuari

- **Visualització del seu comportament:** L'usuari podrà consultar quins events ha enviat, a quins ha votat, quins té com a preferits, els seus comentaris, els seus amics...

■ Pantalla preferències d'usuari (veure figura 6.43)

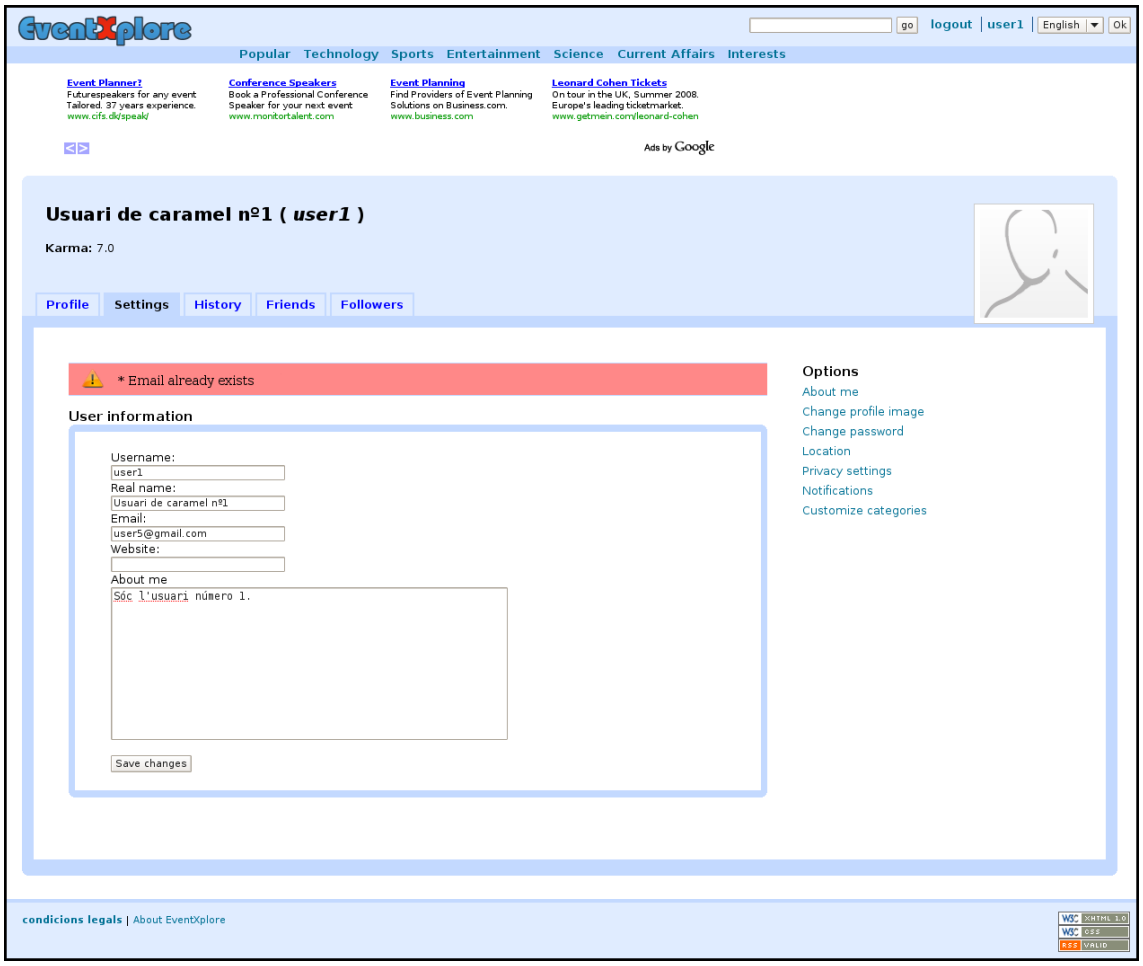

Figura 6.43: Pantalla de preferències d'usuari

Aquesta pantalla ha de permetre a l'usuari poder modificar qualsevol informació referent al seu compte personal.

Les principals funcionalitats són:

- **Modificació de les dades del compte:** Permet modificar qualsevol dada referent al compte de l'usuari. Aquestes dades estan organitzades en el menú de la dreta per facilitar el seu accés.
- **Personalització de les notificacions:** L'usuari podrà configurar les notificacions que desitja rebre sobre els events. Alguns exemples són: Rebre una notificació quan un event preferit és modificat, rebre una notificació quan falti N minuts pel començament d'un event preferit...

- **Personalització de categories:** L'usuari podrà crear una nova metacategoria que contindrà totes les categories que desitgi. D'aquesta manera es proporciona un filtre adaptat als interessos de l'usuari.
- **Errors centralitzats:** Tots el missatges d'error en els formularis estan agrupats al principi de la pàgina i ressaltats en color vermell. Aquest format d'error és comú per tota l'aplicació.

■ Pantalla cerca d'events (veure figura 6.44)

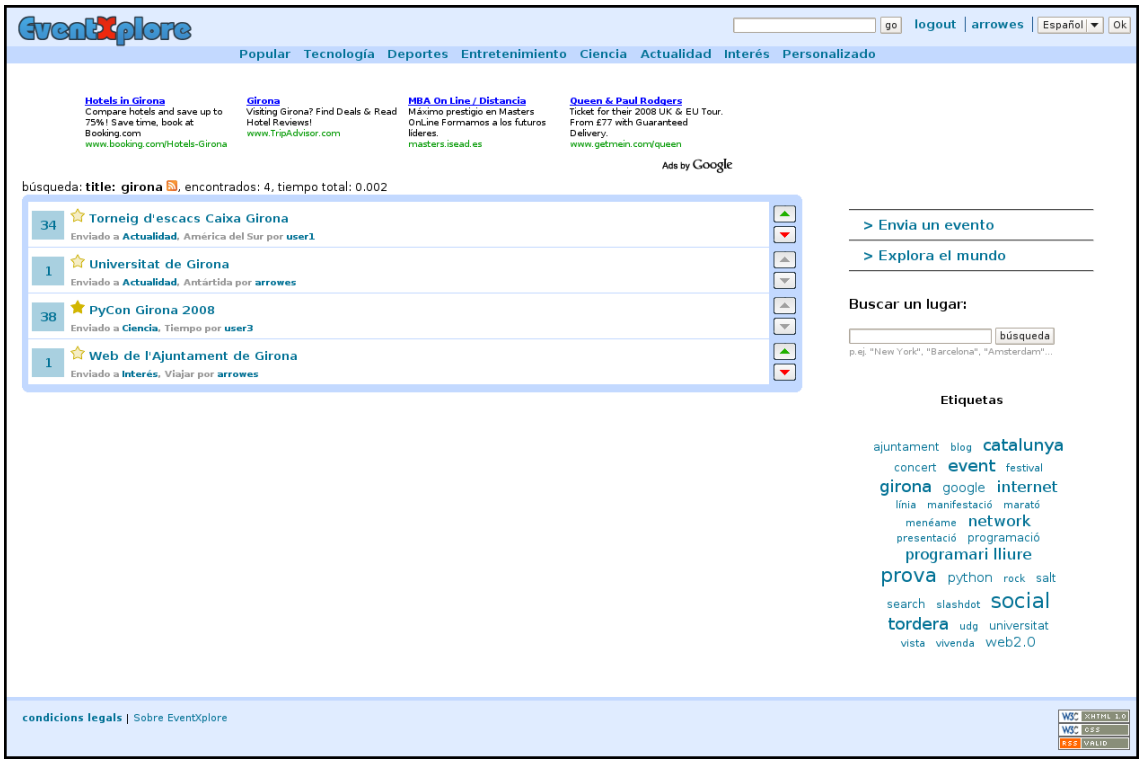

Figura 6.44: Pantalla cerca d'events

En aquesta pantalla es mostra els resultats de la cerca realitzada. Es pot veure quin text de cerca s'ha entrat, el nombre de resultats que s'ha obtingut i el temps que s'ha tardat en trobar els resultats.

Per cada event trobat es mostra la seva previsualització, amb opció d'accedir al seu detall mitjançant els enllaços del títol o del nombre de vots.

7. Disseny del Sistema d'Informació (DSI)

7.1 Definició de l'arquitectura del sistema

7.1.1 Definició de nivells d'arquitectura

Per realitzar el disseny de l'arquitectura del sistema s'ha tingut molt en compte la possibilitat d'escalar[1] el sistema en cas que aquest es quedi petit.

Quan es parla d'escalabilitat cal comentar dos mètodes:

- **Escalabilitat vertical:** afegir més recursos a un únic node en particular dins d'un sistema, com pot ser afegir més memòria o CPUs.
- **Escalabilitat horitzontal:** agregar més nodes a un sistema, com pot ser afegir un servidor nou a una aplicació distribuïda.

Degut al baix cost dels ordinadors i el gran increment en el seu rendiment s'ha optat per formar el sistema de diferents components independents que, en cas de necessitat, permetin escalar horitzontalment.

- **Firewall:** És la primera defensa cap el món exterior. Només ha de permetre les connexions necessàries pel correcte funcionament del sistema.
- **Proxy invers:** És l'encarregat d'administrar les peticions que arriben des d'Internet. En el cas que es tractin de peticions de recursos estàtics les servirà directament; si es tracta d'un recurs dinàmic passarà la petició a l'aplicació.
- **Aplicació:** S'encarrega de generar els recursos dinàmics a partir de les peticions dels usuaris i les dades de la base de dades.
- **Base de dades:** Conté tota la informació disponible al sistema.

A la figura 7.1 podem veure com quedaria el sistema en el cas que s'hagués de separar en diferents servidors.

7.1.2 Especificació d'estàndards i normes de disseny i construcció

El conjunt d'estàndards i normes utilitzats és el següent:

- Per la creació de codi es seguirà les convencions establertes a la *PEP 8 – Style Guide for Python Code*.

[1] Habilitat per administrar el creixement continu de treball de manera fluïda o per estar preparat per fer-se més gran sense perdre qualitat en els serveis oferts.

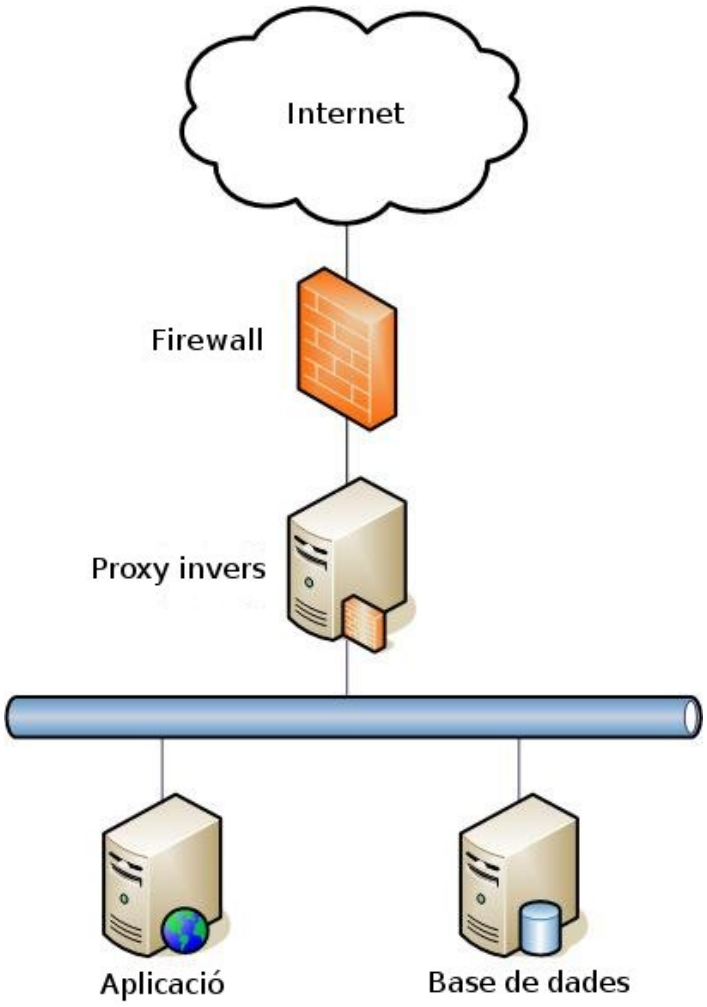

Figura 7.1: Arquitectura del sistema

- El llenguatge de marcatge seguirà les especificacions oficials de XHTML del World Wide Web Consortium (W3C).
- El llenguatge de presentació seguirà les especificacions oficials de CSS del World Wide Web Consortium (W3C).
- El format de dades utilitzat per la difusió dels continguts seguirà les especificacions oficials del RSS 2.0.

7.1.3 Especificació de l'entorn tecnològic

A continuació es defineixen en detall els diferents elements de la infraestructura tècnica que donaran suport al sistema d'informació:

■ Hardware

Per la creació del sistema s'utilitzarà tres entorns de treball amb diferents funcionalitats. Així doncs es disposarà d'un entorn de desenvolupament, un de preproducció i un de producció. A continuació es descriurà la funció de cadascun dels entorns:

- L'entorn de desenvolupament haurà de ser una recreació de l'entorn de producció per poder afegir-hi noves funcionalitats i poder solucionar problemes existents amb certa compatibilitat. En aquest entorn s'hi permetrà la utilització de noves versions del programari utilitzat o de noves aplicacions per poder incorpora-les en un futur a l'entorn de producció.
- L'entorn de preproducció, en canvi, serà una còpia exacte de l'entorn de producció. La seva funció serà la de provar les noves funcionalitats i les correccions dels problemes existents amb la certesa que seran aptes per funcionar a producció.
- L'entorn de producció serà on hi haurà la versió estable del sistema i l'encarregat d'interactuar amb els usuaris finals. No serà permesa la incorporació de cap element a producció sense abans haver passat pels dos entorns anteriors.

■ Programari

El programari utilitzat per albergar el sistema d'informació en cadascun dels entorns mostrats anteriorment serà el següent:

Sistema operatiu: Debian GNU/Linux

Kernel: 2.6.X

Gestor Base de Dades: MySQL 5.0

Sistema de fitxers: ext3

■ Comunicacions

El sistema tindrà una arquitectura client/servidor i inicialment estarà centralitzat en un únic servidor encara que el seu disseny permetrà escalar horitzontalment en cas de neces-

sitat.

7.1.4 Especificació de requisits d'operació i seguretat

Per tal de garantir la protecció del sistema i minimitzar el risc de pèrdua d'informació, a continuació es defineixen els requisits de seguretat i control del sistema:

- Cal que la modificació de les dades es realitzi complint estrictament les regles d'integritat referencial en bases de dades relacionals. Amb això hi ha la seguretat que no queden dades inconsistents i/o incoherents.
- L'accés a registres i la posterior actualització es realitzarà mitjançant sistema transaccional per evitar que possibles accions errònies no puguin tenir un retrocés al seu estat inicial.
- L'accés a la base de dades només estarà obert per peticions locals dins del mateix servidor.
- L'accés al servidor per realitzar el manteniment només estarà permès des d'unes direccions IP concretes i únicament es realitzarà per SSH amb clau pública/privada RSA.
- El sistema disposarà d'un procés automàtic de copies de seguretat que consistirà en crear copies diferencials diàriament i completes setmanalment.

7.2 Disseny de classes

7.2.1 Identificació de classes addicionals

A continuació es descriuen les classes addicionals que formen part del sistema i no s'ha identificat en els casos d'ús descrits en la part d'anàlisi.

Classe ApiController

Mètodes

- url(): Permet consultar informació sobre l'estat d'una URL.
- submit(): Permet enviar un event.

Classe BaseController

Mètodes

- __before__(): Aquest mètode s'executa abans de crear l'objecte. Permet controlar l'idioma de l'aplicació.

Classe CalendarController

Mètodes

- show(): Permet mostrar un calendari HTML.

Classe EventMarker

Atributs propis

- Integer id: Codi únic que identifica el marcador de l'event.
- Float lat: Latitud del marcador.
- Float lng: Longitud del marcador.

Atributs agregats

- Event event: Event al que pertany el marcador.

Classe ErrorController

Mètodes

- document(): Mostra la pantalla d'error.

Classe FrontPageController

Mètodes

- show(): Mostra la pàgina principal de l'aplicació.

Classe HTMLCalendar

Mètodes

- create(): Permet generar un calendari en HTML.

Classe PageController

Mètodes

- __before__(): Aquest mètode s'executa abans de crear l'objecte. Permet inicialitzar totes les variables comunes entre les pàgines de l'aplicació.

Classe SearchController

Mètodes

- search(): Permet mostrar la llista d'events que coincideixin amb els criteris entrats.

Classe SphinxClient

Mètodes

- search_events(): Permet obtenir la llista d'events que coincideixin amb els criteris entrats.

Classe TagController

Mètodes

- show(): Permet mostrar el núvol d'etiquetes populars del sistema.

Així doncs, el diagrama de classes un cop revisat el disseny es pot veure a la figura 7.2.

7.3 Disseny físic de dades

7.3.1 Disseny del model físic de les dades

En aquest apartat es realitza el disseny físic de dades del sistema, a partir dels models de classes de disseny del sistema. Per tal d'obtenir aquest disseny s'apliquen les regles de transformació que proporciona la mètrica 3.

El resultat es pot veure des de la taula 7.1 fins la 7.16.

Nom columna	**Tipus**	**Mida**	**Descripció**
id	int	11	Identificador únic d'avatar.
user_uid	int	11	Identificador de l'usuari propietari de l'avatar.
image	blob		Imatge.
Clau primària:	id		
Claus foranes:	user_uid		

Taula 7.1: Taula avatars

7.4 Especificació de camins d'accés a les dades

El sistema accedirà a les dades, del gestor de base de dades MySQL 5.0, a través del programari *SQLAlchemy*, que proporciona el mapeig objecte-relacional per convertir els

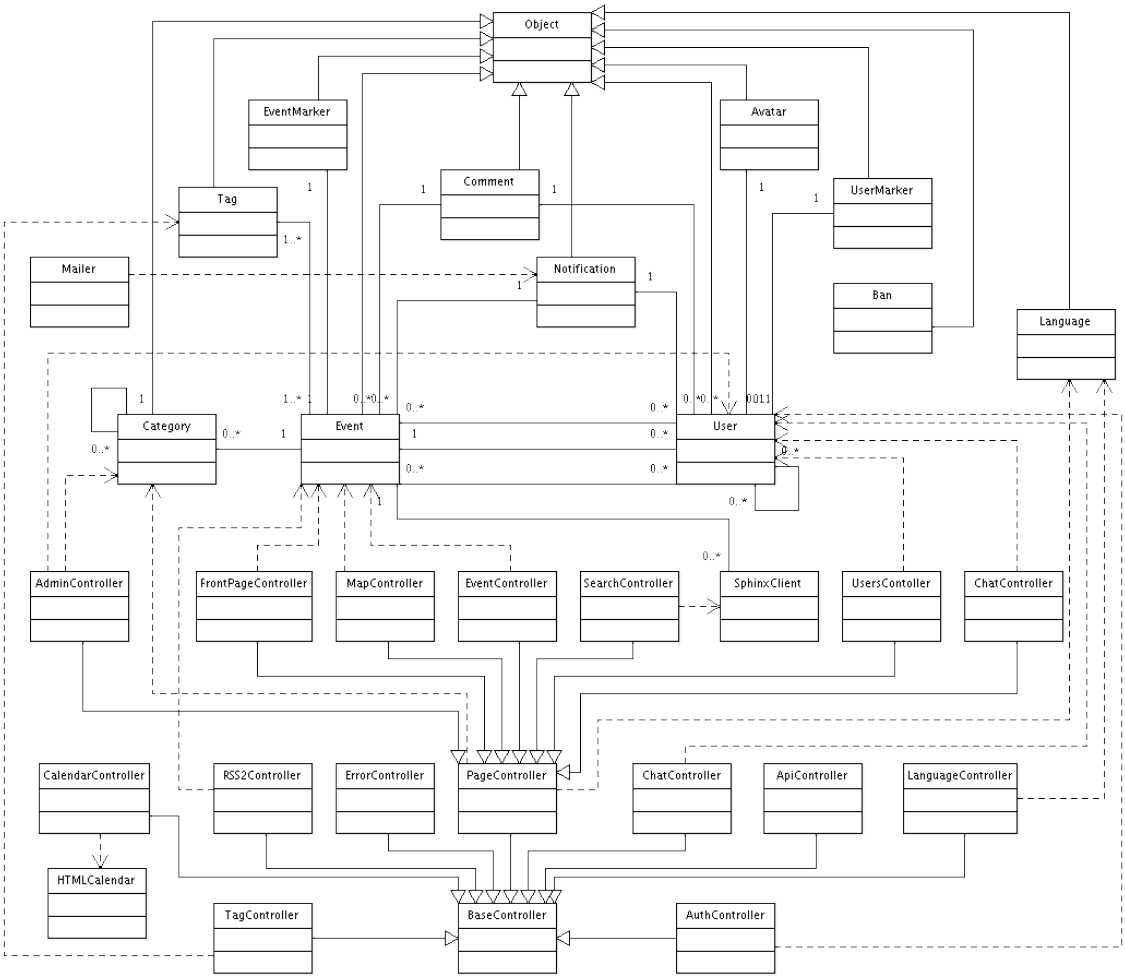

Figura 7.2: Diagrama de classes

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de ban.
type	varchar	15	Tipus de ban.
text	varchar	64	Text del tipus de ban.
date	datetime		Data del ban
expire	datetime		Data en que caduca el ban.
comment	varchar	100	Comentari sobre els motius del ban.
Clau primària:	id		
Claus foranes:			

Taula 7.2: Taula bans

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de la categoria.
parent_id	int	11	Identificador de la metacategoria pare.
name	varchar	32	Nom de la categoria.
Clau primària:	id		
Claus foranes:	parent_id		

Taula 7.3: Taula categories

Nom columna	Tipus	Mida	Descripció
time	unsigned int	10	Data del missatge.
uid	int	11	Identificador del xat.
room	varchar	15	Nom de la sala del xat.
user	varchar	32	Nom de l'usuari que ha enviat el missatge.
text	varchar	255	Text del missatge enviat.
Clau primària:			
Claus foranes:			

Taula 7.4: Taula chats

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic del comentari.
event_id	int	11	Ident. de l'event al que pertany el comentari.
user_uid	int	11	Ident. de l'usuari que ha escrit el comentari.
date	datetime		Data en que s'ha enviat el comentari.
ip	unsigned int	11	Direcció IP des de la qual s'ha enviat.
order	int	6	Ordre del comentari en la llista.
content	text		Contingut del comentari.
Clau primària:	id		
Claus foranes:	parent_id event_id user_uid		

Taula 7.5: Taula comments

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic del marcador.
event_id	int	11	Identificador de l'event al que pertany el marcador.
lat	float	10, 6	Latitud del marcador.
lng	float	10, 6	Longitud del marcador.
Clau primària:	id		
Claus foranes:	event_id		

Taula 7.6: Taula event_markers

Nom columna	Tipus	Mida	Descripció
event_id	int	11	Identificador de l'event al que pertany l'etiqueta.
tag_id	int	11	Identificador de l'etiqueta que es fa referència.
date	datetime		Data de creació de l'etiqueta.
Clau primària:	event_id, tag_id		
Claus foranes:	event_id tag_id		

Taula 7.7: Taula event_tags

Nom columna	Tipus	Mida	Descripció
event_id	int	11	Identificador de l'event al que pertany el vot.
user_uid	int	11	Identificador de l'usuari que ha fet el vot.
date	datetime		Data en que s'ha fet el vot.
value	int	4	Valor del vot.
Clau primària:	event_id, user_uid		
Claus foranes:	event_id user_uid		

Taula 7.8: Taula event_votes

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de l'event.
user_uid	int	11	Identificador de l'usuari que l'ha enviat.
category_id	int	11	Identificador de la categoria.
start_date	datetime		Data en la que comença l'event.
end_date	datetime		Data en la que acaba l'event.
date	datetime		Data en que s'ha enviat.
votes	int	6	Vots que té l'event.
karma	float	10, 2	Karma de l'event.
modified	datetime		Data de l'última modificació.
ip	unsigned int	11	IP des de la que s'ha enviat.
uri	varchar	100	Direcció interna de l'event.
url	varchar	250	Direcció "oficial" de l'event.
title	varchar	100	Títol de l'event.
content	text		Contingut de l'event.
status	varchar	15	Estat en que es troba.
randkey	int	20	Clau de control de l'event.
tags_list	varchar	200	Llista d'etiquetes.
Clau primària:	id		
Claus foranes:	user_uid category_id		

Taula 7.9: Taula events

Nom columna	Tipus	Mida	Descripció
user_uid	int	11	Identificador de l'usuari.
event_id	int	11	Identificador de l'event.
date	datetime		Data en que s'ha afegit a preferits.
Clau primària:	user_uid, event_id		
Claus foranes:	user_uid event_id		

Taula 7.10: Taula favorites

Nom columna	Tipus	Mida	Descripció
friend_from	int	11	Identificador de l'usuari que ha afegit l'amic.
friend_to	int	11	Identificador de l'usuari afegit.
date	datetime		Data en que s'ha creat la relació.
Clau primària:	friend_from, friend_to		
Claus foranes:	friend_from friend_to		

Taula 7.11: Taula friends

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de l'idioma.
locale	varchar	5	Abreviatura de l'idioma.
name	varchar	64	Nom de l'idioma.
Clau primària:	id		
Claus foranes:			

Taula 7.12: Taula languages

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de la notificació.
user_uid	int	11	Identificador de l'usuari que la rebrà.
event_id	int	11	Identificador de l'event sobre el que tracta.
type	varchar	15	Tipus de notificació.
date	datetime		Data en que s'ha d'enviar la notificació.
stauts	varchar	15	Estat de la notificació.
Clau primària:	id		
Claus foranes:	user_uid event_id		

Taula 7.13: Taula notifications

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic de l'etiqueta.
word	varchar	40	Paraula de l'etiqueta.
Clau primària:	id		
Claus foranes:			

Taula 7.14: Taula tags

Nom columna	Tipus	Mida	Descripció
id	int	11	Identificador únic del marcador.
user_uid	varchar	40	Identificador de l'usuari al que pertany.
lat	float	10, 6	Latitud del marcador.
lng	float	10, 6	Longitud del marcador.
Clau primària:	id		
Claus foranes:	user_uid		

Taula 7.15: Taula user_markers

Nom columna	Tipus	Mida	Descripció
uid	int	11	Identificador únic de l'usuari.
username	varchar	64	Nom d'usuari en el sistema.
password	varchar	64	Contrasenya de l'usuari.
date	datetime		Data de creació.
modified	datetime		Data de l'última modificació.
validated_date	datetime		Data de validació.
name	varchar	64	Nom real de l'usuari.
email	varchar	64	Direcció de correu electrònic.
username_register	varchar	64	Nom d'usuari inicial.
email_register	varchar	64	Correu inicial.
ip	unsigned int	11	IP des d'on es va crear el compte.
karma	float	10, 2	Karma de l'usuari.
public_info	text		Informació sobre l'usuari.
website	varchar	128	Adreça web personal.
profile_views	int	11	Número de visites al seu perfil.
Clau primària:	id		
Claus foranes:			

Taula 7.16: Taula users

objectes utilitzats en el sistema a formes que puguin ser emmagatzemades a la base de dades i recuperades fàcilment més tard, mentre es preserven les propietats dels objectes i les seves relacions, és a dir, objectes persistents.

7.5 Optimització del model físic de dades

Per qüestions de rendiment s'ha hagut de desnormalitzar[2] la taula *events*. Els camps afegits són:

- votes: Permet obtenir el nombre de vots d'un event sense haver de contar-los d'un en un de la taula event_votes.
- tags_list: Permet obtenir la llista d'etiquetes d'un event sense haver d'accedir a les taules event_tags i tags.

Per altre banda també cal comentar que totes les taules del model, excepte una, utilitzen el motor **InnoDB** ja que proporciona transaccions que compleixen amb ACID. D'aquesta manera es garanteix que les dades sempre són coherents tot i que es perd rendiment en les consultes. En cas que es detecti que amb el creixement de la base de dades i/o el nombre d'usuaris el rendiment baixi notablement es passarà a utilitzar el motor **MyISAM**.

Com a excepció tenim la taula *chats* que utilitza el motor **Memory**. Aquest motor guarda les dades a memòria, augmentant la velocitat de lectura i escriptura enormement. La utilització d'aquest motor és necessària ja que l'accés a aquesta taula ha de ser molt ràpid per poder oferir als usuaris una comunicació en temps real.

7.6 Especificacions de construcció

7.6.1 Especificacions de l'entorn de construcció

Per la construcció del sistema d'informació cal definir quin és l'entorn, segons els següents conceptes:

- Entorn tecnològic: hardware, programari i comunicacions.
- Eines de construcció: llenguatge de programació, compiladors...

■ Hardware

Com ja s'ha comentat anteriorment es disposarà de tres entorns de treball. A continuació es detalla el hardware utilitzat per les estacions de treball de cada entorn:

- Estació de desenvolupament 1:

  Processador: Intel Pentium III 800MHz
  Memòria RAM: 256Mb DIMM

[2]Procés d'optimitzar una base de dades agregant dades redundants.

1r Disc dur: 60Gb
2n Disc dur: 30Gb
Tipus de disc: IDE

- Estació de desenvolupament 2:

  Processador: Intel Core 2 Duo 2 x 1'8GHz
  Memòria RAM: 1Gb DDR
  Disc dur: 100Gb
  Tipus de disc: SATA2

- Servidor de preproducció:

  Processador: AMD Athlon 64 3'2GHz
  Memòria RAM: 2Gb DDR
  Disc dur: 120Gb
  Tipus de disc: SATA

- Servidor de producció:

  Processador: Intel Core 2 Duo E6550 2 x 2.33GHz
  Memòria RAM: 4 GB DDR2
  Disc dur: 750 GB
  Tipus de disc: SATA2

■ Programari

Per la construcció dels sistema s'ha intentat trobar una solució que permeti la reusabilitat i la separació de funcionalitat en diferents mòduls. Per això s'ha optat per la utilització del framework web **Pylons**, que combina les millors idees dels mons de Ruby, Python i Perl, proporcionant un framework estructurat però extremadament flexible, a més de ser un dels primers projectes d'utilitzar l'estàndard WSGI.

A continuació es detalla els components que formen part d'aquest framework:

- **Paste:** conjunt de llibreries que fan ús de l'estàndard WSGI per oferir funcionalitats diferents a la creació de contingut. A més proporciona un senzill servidor web que pot produir peticions WSGI.
- **SQLAlchemy:** eina de mapeig objecte-relacional pel llenguatge Python que facilita el mapeig d'atributs entre una base de dades relacional tradicional i el model d'objectes d'una aplicació.
- **Mako:** llibreria de plantilles escrita en Python. Proporciona una sintaxi familiar que es compila en mòduls Python per un màxim rendiment.
- **WebHelpers:** paquet creat amb la intenció de proporcionar funcions d'ajuda pel seu ús dins d'aplicacions web.
- **Routes:** reimplementació en Python del sistema de rutes de Rails pel mapeig d'URLs a controladors/accions i la generació d'URLs.

- **Beaker:** capa WSGI que proporciona sessions en una aplicació web.

A la figura 7.3 podem veure el procés pel que passa una petició HTTP a través del framework Pylons.

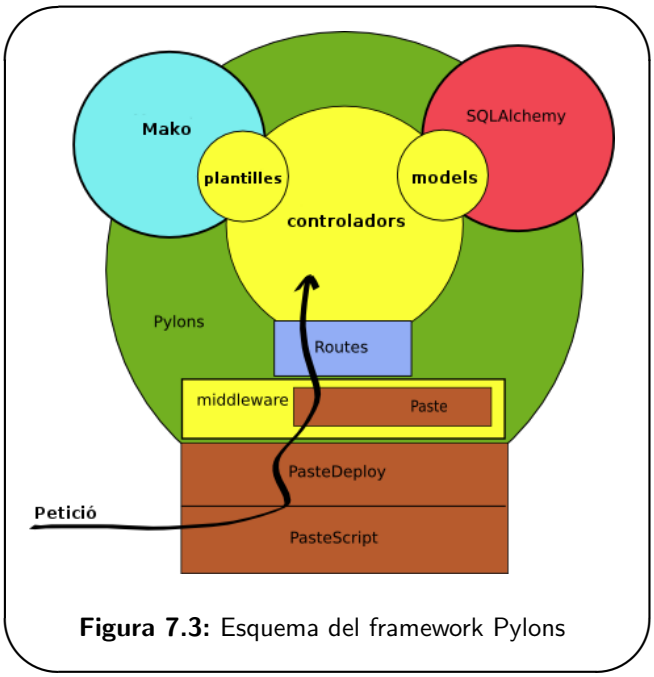

Figura 7.3: Esquema del framework Pylons

Com a sistema de gestió de base de dades s'ha optat per la utilització de **MySQL** degut a la seva gran flexibilitat per utilitzar diferents motors d'emmagatzematge (MyISAM, InnoDB, Memory...), la gran comunitat d'usuaris que té al darrere i la seva adaptació, sobradament demostrada, en aplicacions web.

Pel control de versions s'utilitzarà **subversion**, que permetrà administrar els fitxers i directoris de l'aplicació juntament amb els canvis que es vagin fent. A més es podrà recuperar en qualsevol moment versions velles o consultar un històric de com han canviat els fitxers.
Per fer més fàcil l'ús del sistema de control de versions s'utilitzarà el programari **Trac**, que proporciona una interfície web tant per la consulta de l'històric de canvis com pel registre i assignació d'incidències del sistema.

Com ja s'ha comentat anteriorment, es disposarà de tres entorns de treball que funcionaran amb el següent programari:

- Estació de desenvolupament 1:

  Sistema operatiu: Debian GNU/Linux lenny/sid
  Kernel: 2.6.25
  Gestor Base de Dades: MySQL 5.0
  Servidor web: Paste 1.7
  Framework web: Pylons 0.9.6.2

- Estació de desenvolupament 2:

Sistema operatiu: Debian GNU/Linux lenny/sid
Kernel: 2.6.25
Gestor Base de Dades: MySQL 5.0
Servidor web: Paste 1.7
Framework web: Pylons 0.9.6.2

- Servidor de preproducció:

  Sistema operatiu: Debian GNU/Linux lenny
  Kernel: 2.6.19
  Gestor Base de Dades: MySQL 5.0
  Servidor web: Nginx 0.5.35-2 - Paste 1.7
  Framework web: Pylons 0.9.6.2

- Servidor de producció:

  Sistema operatiu: Debian GNU/Linux lenny
  Kernel: 2.6.19
  Gestor Base de Dades: MySQL 5.0
  Servidor web: Nginx 0.5.35-2 - Paste 1.7
  Framework web: Pylons 0.9.6.2

■ Llenguatge de programació

L'elecció del framework Pylons com a base del sistema ja ens marca el llenguatge de programació a utilitzar, que serà **Python**.

Python és un llenguatge de programació interpretat i multiparadigme. Això significa que estalvia un temps considerable en el desenvolupament ja que no cal compilar ni enllaçar i que permet varis estils de programació: programació orientada a objectes, programació estructurada i programació funcional.

L'intèrpret de Python també disposa d'un mode interactiu en el que s'escriuen les instruccions en una espècie de shell: les expressions poden ser introduïdes una a una, podent veure el resultat de la seva avaluació immediatament. Aquest fet facilita experimentar amb característiques del llenguatge i provar funcions durant el desenvolupament del programa.

■ Comunicacions

Pel que fa a la comunicació amb el sistema implementat, s'utilitzarà el servidor web **Nginx** com a proxy invers cap al servidor web **Paster**, incorporat en el framework. D'aquesta manera, en cas que el sistema tingui la necessitat de créixer ràpidament, serà força senzill afegir nous servidors amb una instància de l'aplicació i utilitzar el Nginx com a balancejador de càrrega.

Per altra banda, cal tenir en consideració diferents aspectes sobre el procés de comunicació entre l'usuari i el servidor per poder dissenyar un sistema que garantitzi una comunicació fiable i ràpida.

La velocitat de càrrega de les pàgines web ve determinada per molts factors, però els més importants són la combinació de la latència[3] i el nombre de recursos a descarregar.

En la pràctica, la latència i l'ample de banda de l'usuari s'hauran de prendre com elements incontrolables. Com que l'ample de banda ha anat creixent exponencialment des de fa uns anys, aquest element ja no té la importància que tenia anteriorment, i el temps d'espera degut a la latència i el nombre de recursos a descarregar ha agafat més importància.

Quan un navegador ha de mostrar una pàgina web nova, per exemple http://eventxplore.com/, primer ha de convertir el nom de domini a una direcció IP amb la que connectar, així que contacta amb els DNS del ISP per aconseguir aquesta direcció IP. Un cop obtinguda la direcció, intenta establir una connexió TCP amb el servidor, seguint el procés conegut com "**three-way-handshake**". El navegador envia un paquet SYN i espera la resposta del servidor, que serà un paquet SYN + ACK. Quan el navegador rebi el paquet de resposta del servidor ja pot enviar un paquet ACK de resposta i immediatament llançar una petició HTTP GET del recurs que li interessa. D'aquesta manera el servidor respondrà enviant el contingut del recurs sol·licitat.

Si el temps que tarda un paquet en arribar del navegador al servidor és de 100 ms, es necessitarà 400 ms degut a la latència abans que el navegador obtingui el primer byte de contingut real. A partir d'aquí i fins que el navegador acabi de rebre el recurs passarà un temps en funció del seu tamany i l'ample de banda que es disposi. Suposant que el recurs pesa 100 Kb i es disposa d'una connexió de 4 Mbits/s el temps aproximat que tardarà el navegador a descarregar el recurs serà:

$$t = 100\ Kb\ /\ ((4 * 1024)\ /\ 8)\ Kb/s\ *\ 1.2 = 235\ ms$$

El factor "màgic" 1.2 té en consideració la sobrecàrrega típica de dades a enviar degut a l'encapsulació d'aquestes dades en paquets.

Un cop amb el contingut, el navegador procedeix a interpretar-lo i, normalment, trobarà dins l'HTML referències a altres recursos, com fulles d'estil CSS, Javascript, imatges, etc. Per cadascun d'aquests recursos extra es repeteix tot el procés anterior, així doncs si es fa referència a 10 recursos, un nombre molt conservador avui en dia, es perdria el següent temps extra degut a la latència.

$$t = 10 * 400\ ms = 4\ s$$

Afortunadament el navegador paral·lelitza aquestes peticions, però per no sobrecarregar el servidor no fa més de dos peticions al mateix domini. Els fitxers Javascript són una excepció i no es descarreguen en paral·lel. Si el recurs obtingut no utilitza cap fitxer Javascript, els 10 recursos afegiran:

$$t = 10 * 400\ ms\ /\ 2 = 2\ s$$

Un cop vist el problema cal plantejar solucions:

[3] Temps que triga un paquet de dades (o una petició d'usuari) d'es d'un punt fins a un altre, altrament anomenat retard.

En primer lloc es pot "enganyar" al navegador perquè descarregui més fitxers en paral·lel. Existeix un truc molt senzill, és crear diferents subdominis que apuntin al mateix servidor i dividir els enllaços de l'HTML als recursos entre aquests subdominis. El navegador comença a descarregar en paral·lel un màxim de 2 recursos per subdomini, sobre un màxim de 4 dominis simultàniament. Fent això el temps perdut establint les connexions TCP i començant a descarregar els recursos serà de 400 ms per establir les primeres 8 connexions més 400 ms per les dues últimes.

$$t = 400 + 400 = 0.8\ s$$

També hi ha una forma més senzilla de reduir aquest temps d'espera. Si el servidor no tanca la connexió TCP amb el client en cada petició HTTP, utilitzant **KeepAlives**, es divideix per dos l'efecte de la latència en cada recurs que es descarrega en una connexió ja establerta. L'efecte de KeepAlives depèn del nombre de subdominis que es disposi, si es serveix tot des del mateix domini, en lloc dels 4 s anteriors es tardarà:

$$t = 400 + 200 * (8\ /\ 2) = 1.2\ s$$

400 ms per establir les dues primeres connexions en paral·lel més 200 ms per cadascun dels altres 8 recursos a descarregar dividit entre 2, que és el nombre de recursos que es descarreguen en paral·lel.

Utilitzant 4 subdominis el temps es reduiria encara més ja que es podrien establir les primeres 8 connexions en paral·lel i les dues últimes ja estarien obertes i també es produirien en paral·lel:

$$t = 400 + 200 = 0.6\ s$$

Com a última solució, es comenta una que sobrepassa a qualsevol altre tècnica per reduir el temps d'espera quan es vol començar a utilitzar un recurs, no enviar res al navegador, sempre que aquest tingui a caché una còpia del recurs necessitat.

El navegador divideix els recursos que té a caché en dos tipus, els "frescs" i els "dubtosos". Els recursos frescs poden utilitzar-se sense preguntar res al servidor. Els recursos dubtosos es poden utilitzar, però abans s'ha de preguntar al servidor que el va enviar si segueix sent vàlid o si s'ha de tornar a descarregar.

Per controlar si un recurs entra a la caché i, si ho fa, per quin temps, s'utilitza les capçaleres **Cache-Control** i **Expires**. En el sistema dissenyat s'utilitzarà la capçalera Expires que conté una data a partir de la qual el recurs ha de ser considerat com a dubtós. Si la data està en el passat, o si no és una data (p.ex "0"), es considerarà immediatament el recurs com a dubtós.

Així doncs, la solució dissenyada per al sistema és mantenir sempre els recursos frescs afegint una capçalera Expires, per exemple, 10 anys endavant de la data actual. L'únic inconvenient és com aconseguir que els canvis aplicats a un recurs siguin detectats pel navegador i els descarregui, ja que d'aquesta manera no ho faria. Aquesta solució serà mostrada i comentada més endavant.

7.6.2 Definició de components i subsistemes de construcció

A més de la utilització del framework s'ha necessitat la incorporació de components de tercers per poder realitzar diferents tasques que d'altre forma no s'aconseguirien o seria molt costós implementar-les. Aquesta és la descripció del components utilitzats:

- **Google Maps:** servei de cartografia en línia gratuït de Google. Ofereix mapes i imatges reals provinents de satèl·lits i avions, que es poden consultar des del web de Google Maps o com a incrustacions a terceres webs a través de l'API de Google Maps.
- **AdSense:** sistema de publicitat ideat per Google. L'Adsense ha esdevingut molt popular ja que els anuncis són més respectuosos que la majoria de banners de publicitat i el contingut d'ells s'adequa al de la web.
- **Authkit:** framework complet per autenticació i autorització en aplicacions WSGI.
- **Babel:** col·lecció d'eines per internacionalitzar aplicacions Python.
- **Formencode:** paquet per la validació i creació de formularis.
- **Python Imaging Library (PIL):** llibreria pel tractament d'imatges amb Python.
- **JQuery:** biblioteca o framework Javascript que permet simplificar la manera d'interactuar amb documents HTML, permetent manipular events, afegir animacions i agregar interacció amb la tecnologia AJAX a pàgines web.
- **JQuery datePicker:** plugin per a JQuery que permet la creació de calendaris Javascript.
- **PyRSS2Gen:** generador de documents RSS 2.0 per Python.
- **Simplejson:** llibreria per codificar i descodificar dades en JSON.
- **Sphinx:** motor de cerca de text pensat per proporcionar funcions de cerca a altres aplicacions.
- **Jsjam:** programa escrit en Perl per comprimir fitxers Javascript i d'aquesta manera proporcionar una descàrrega més ràpida als navegadors.

7.7 Especificació del pla de proves

7.7.1 Especificació tècnica de nivells de prova

- **Proves unitàries:** Es realitzarà les proves de caixa blanca i caixa negra a tots els mètodes implementats en cadascuna de les classes desenvolupades. Aquestes proves seran realitzades pels programadors en el moment de desenvolupar les classes.
- **Proves d'integració:** Es realitzarà proves d'integració fins arribar a nivells acceptables de funcionalitat. Aquestes proves seran realitzades per experts de reconeguda vàlua.
- **Proves del sistema:** Es realitzarà proves d'estrès en els diferents entorns. Aquestes proves permetran saber si el sistema compleix el requeriments de rendiment establertes.

8. Construcció del Sistema d'Informació (CSI)

8.1 Preparació de l'entorn de generació i construcció

8.1.1 Implantació de la base de dades física o fitxers

Com ja s'ha exposat anteriorment el sistema gestor de base de dades emprat és MySQL en la versió 5.0. Serà doncs amb aquesta aplicació que es generaran les taules, índexs, claus primàries i relacions.

Per facilitar la feina a l'hora de treballar amb el gestor s'utilitzarà les següents aplicacions obtingudes del repositori oficial de Debian:

- mysql-admin
- mysql-query-browser

8.1.2 Preparació de l'entorn de construcció

En aquesta tasca es prepara l'entorn en el que es construiran els components del sistema d'informació.

■ Pylons

En primer lloc cal instal·lar el framework Pylons, que serà la base del sistema. Per fer-ho primer es necessita disposar del mòdul **easy_install** que permet descarregar, construir, instal·lar i administrar paquets Python. S'obté de la següent manera:

```
1 wget http://peak.telecommunity.com/dist/ez_setup.py
2
3 python ez_setup.py Pylons==0.9.6.2
```

Llistat 8.1: Instal·lació de Pylons

Un cop instal·lat, cal crear un script d'inici perquè s'executi automàticament el servidor web **paster** que porta incorporat. Aquest servidor haurà d'escoltar al port 5000 les peticions que li enviarà el proxy Nginx.

■ Nginx

El següent pas és instal·lar i configurar Nginx com a proxy invers. Aquest servidor web es troba als repositoris oficials de Debian, per tant s'instal·la amb el gestor de paquets del

sistema operatiu.

A continuació és mostra la part de configuració més interessant del servidor per fer-lo treballar com a proxy invers. D'aquesta manera s'enviarà les peticions de recursos dinàmics al servidor web paster i es servirà directament les peticions de recursos estàtics.

```
http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    access_log /var/log/nginx/access.log;

    gzip         on;
    gzip_proxied any;
    gzip_comp_level 2;
    gzip_types   text/plain text/html text/css application/x-javascript text/xml
        application/xml application/xml+rss text/javascript;
    sendfile     on;
    tcp_nopush   on;
    tcp_nodelay  on;
    keepalive_timeout 75 20;

    include /etc/nginx/sites-enabled/*;

}

...

server {
    listen  80;
    server_name eventxplore.com;

    access_log /var/log/nginx/eventxplore.access.log;

    location ~ "^/static/" {
        root   /var/www/eventxplore/eventxplore/public;
        expires 10y;
        proxy_hide_header Etag;
    }

    location / {
        include /etc/nginx/conf/proxy.conf;
        proxy_pass http://127.0.0.1:5000;
        proxy_redirect default;
    }
}
```

Llistat 8.2: Configuració de Nginx

Com aspectes a destacar tenim:

- El contingut s'envia comprimit per disminuir el temps de resposta i reduir el tràfic utilitzat.
- L'opció keepalives està activada per mantenir les connexions amb el client obertes

durant un temps de 75 segons. D'aquesta manera s'agilitza la descàrrega de recursos, ja que el cost d'establir connexions noves és molt elevat.

- El contingut estàtic és servit directament, amb una data d'expiració de 10 anys endavant, per qüestions d'optimització que es comentaran més endavant.
- Les peticions de recursos dinàmics són enviades al servidor web paster que escolta pel port 5000.

■ Sphinx

Sphinx permet dotar a una aplicació d'un cercador de contingut a la base de dades molt eficient. Per aconseguir-ho indexa els camps de les taules que se li configuri i així permet obtenir uns resultats de forma molt ràpida. El problema és que per defecte només ve configurat amb un índex i, quan aquest és molt gran, les cerques queden desactualitzades, ja que no es pot estar reindexant cada moment degut al seu gran cost.

La solució al problema és crear un segon index més petit que s'actualitzarà molt sovint. Així s'aconsegueix que els resultats que no estiguin en l'índex principal estaran en el secundari.

Aquests índexs es reindexen mitjançant les dues tasques programades següents:

```
1 */2 * * * * /usr/local/sphinx/bin/indexer delta --rotate
2 0 4 * * * /usr/local/sphinx/bin/indexer --merge main delta --rotate
```

Llistat 8.3: Tasques d'actualització dels índexs de Sphinx

Com es pot apreciar l'índex secundari s'actualitza cada 2 minuts i el principal cada dia a les 4 de la matinada.

La configuració per aconseguir aquest comportament és:

```
1  source main
2  {
3      type                = mysql
4      sql_host            = localhost
5      sql_user            = eventxplore
6      sql_pass            = eventxplore
7      sql_db              = eventxplore
8      sql_port            = 3306 # optional, default is 3306
9
10     sql_query_pre = REPLACE INTO sph_counter SELECT 1, MAX(id) FROM events
11
12     sql_query           = \
13         SELECT id, url, title, content, tags_list, UNIX_TIMESTAMP(start_date) AS
               start_date \
14         FROM events \
15         WHERE id<=( SELECT max_doc_id FROM sph_counter WHERE counter_id=1 )o
16
17     sql_attr_timestamp    = start_date
18     sql_ranged_throttle = 0
19     sql_query_info    = SELECT * FROM events WHERE id=$id
```

```
}

source delta : main
{
    sql_query_pre =

    sql_query             = \
        SELECT id, url, title, content, tags_list, UNIX_TIMESTAMP(start_date) AS
            start_date \
        FROM events \
        WHERE id>( SELECT max_doc_id FROM sph_counter WHERE counter_id=1 )
}

index main
{

    source        = main
    path          = /usr/local/sphinx/var/data/main
    docinfo       = extern
    mlock         = 0
    morphology    = none
    min_word_len  = 1
    charset_type  = sbcs
    html_strip    = 0
}

index delta : main
{
    source = delta
    path = /usr/local/sphinx/var/data/delta
}

indexer
{
    mem_limit         = 32M
}

searchd
{
    port              = 3312
    log               = /usr/local/sphinx/var/log/searchd.log
    query_log         = /usr/local/sphinx/var/log/query.log
    read_timeout      = 5
    max_children      = 30
    pid_file          = /usr/local/sphinx/var/log/searchd.pid
    max_matches       = 1000
    seamless_rotate   = 1
    preopen_indexes   = 0
    unlink_old        = 1
}
```

Llistat 8.4: Configuració de Sphinx

Com aspectes a destacar tenim:

- Per fer coordinar els dos índexs s'ha creat una taula sph_counter on es guarda quin

ha sigut l'últim event indexat per l'índex principal. Així l'índex secundari només ha d'indexar la resta d'events.

- Els camps indexats són el títol, el contingut, l'URL, i la llista d'etiquetes.

■ Eclipse

Com a IDE de desenvolupament s'utilitza Eclipse ja que proporciona una gran flexibilitat gràcies a l'enorme quantitat de plugins que disposa. En aquest cas s'utilitza els següents:

- **Pydev:** Permet treballar amb el llenguatge de programació Python.
- **TeXlipse:** Permet la creació i manipulació de documents amb LaTeX.
- **Subclipse:** Permet l'administració de repositoris subversion.

■ Subversion

Subversion és un gestor de versions que permet administrar els fitxers i directoris de l'aplicació juntament amb els canvis que es vagi fent. A més es pot recuperar en qualsevol moment versions velles o consultar un històric de com han canviat els fitxers.

El repositori que es crearà serà accessible des de Internet, així es podrà treballar amb l'aplicació des de qualsevol lloc.

Per accedir-hi s'utilitzarà el servidor web Apache. El procés de instal·lació i configuració és el següent:

```
1  apt-get install apache2
2  apt-get install subversion
3  apt-get install libapache2-svn
4
5  # Es crea el directori que albergarà els fitxers
6  mkdir /var/svn
7
8  # Es crea el repositori
9  svnadmin create /var/svn/eventxplore
10
11 # Es dona permisos a l'usuari d'apache
12 chown -R www-data:www-data /var/svn/*
13
14 # Creació de l'estructura del repositori
15 mkdir /tmp/eventxplore
16 mkdir /tmp/eventxplore/branches
17 mkdir /tmp/eventxplore/tags
18 mkdir /tmp/eventxplore/trunk
19 svn import -m "Import inicial del repositori eventxplore" \
20 /tmp/eventxplore file:///var/svn/eventxplore
21
22 # Ja està configurat Subversion, ara toca fer la configuració de WebDav
23 # de Subversion que permetrà accedir al repositori mitjançant apache.
24 # Es carreguen els següents mòduls.
25 a2enmod dav
```

```
a2enmod dav_svn

# Es crea el fitxer d'accés
htpasswd -c /etc/apache/svn.passwd eventxplore

# Finalment es configura el mòdul dav_svn deixant el fitxer
# /etc/apache2/mods-available/dav_svn.conf així:
<location>
    DAV svn
    SVNPath /var/svn/web1
    AuthName "Subversion Repositori eventxplore"
    AuthUserFile /etc/apache2/svn.passwd
    Require valid-user
</location>

# Ja només queda reiniciar el servei
/etc/init.d/apache2 force-reload
```

Llistat 8.5: Instal·lació i configuració d'Apache + Subverison

■ Trac

Per fer més fàcil l'ús del sistema de control de versions s'utilitzarà el programari **Trac** que proporciona una interfície web tant per la consulta de l'històric de canvis com pel registre i assignació d'incidències del sistema.

Com exemple, a la figura 8.1 podem veure una mostra de l'exploració del repositori del projecte i, a la figura 8.2, una captura de la pantalla de canvis entre revisions.

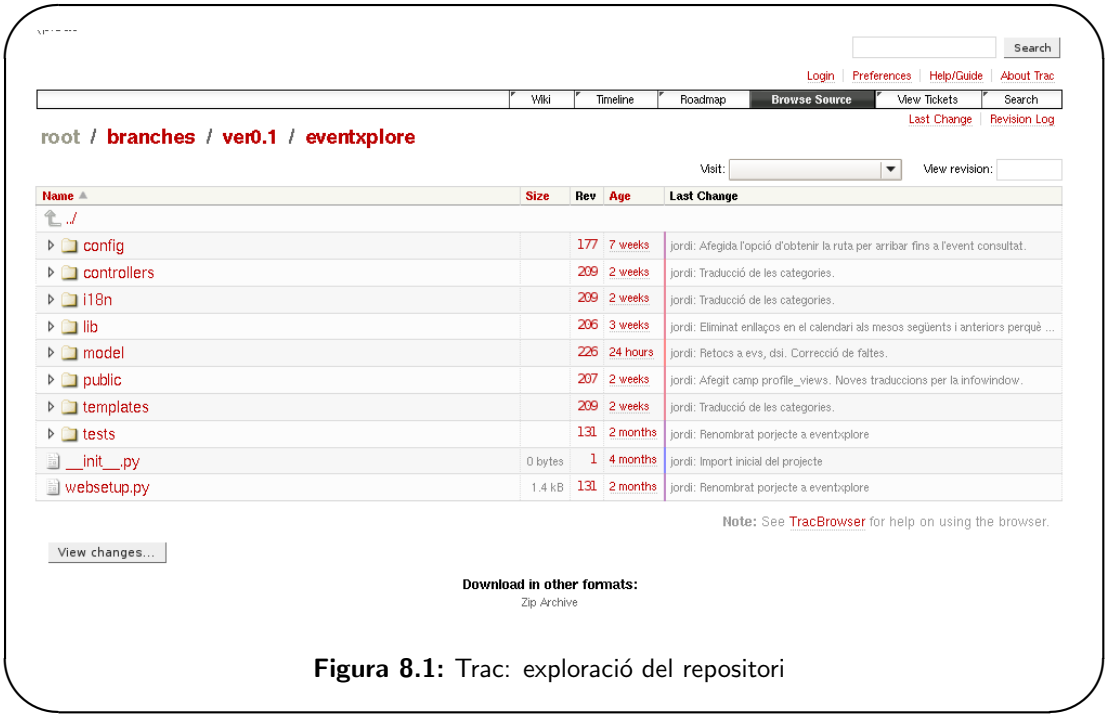

Figura 8.1: Trac: exploració del repositori

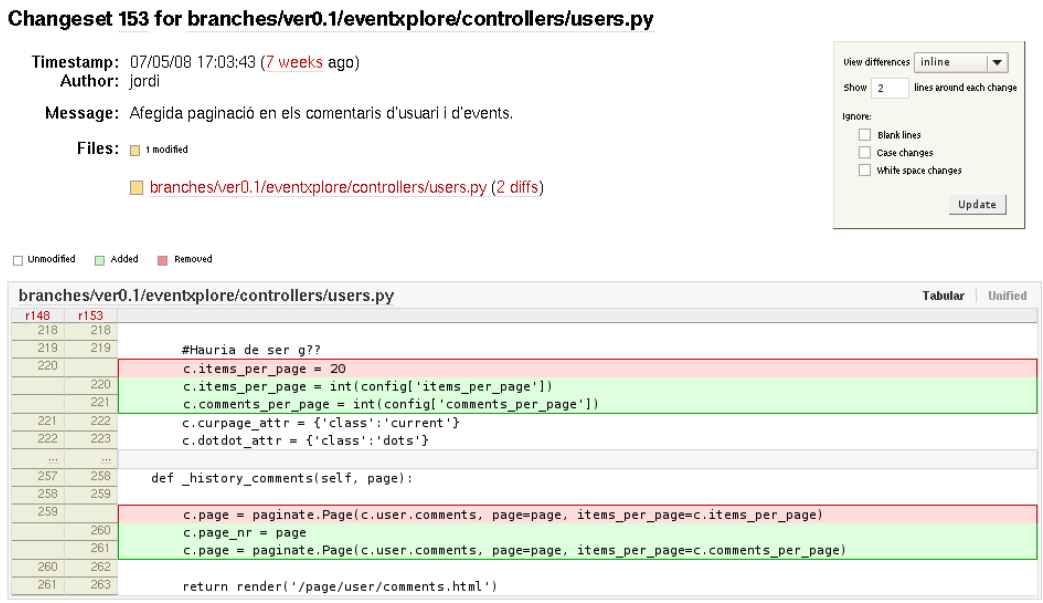

Figura 8.2: Trac: canvis entre revisions

8.2 Generació del codi dels components i procediments

8.2.1 Generació del codi dels components

■ Sistema de caché i optimització de recursos

Com ja s'ha comentat anteriorment, l'aplicació disposarà d'un sistema de caché que permetrà que els navegadors dels usuaris es guardin els recursos estàtics fins que aquests canviïn. El problema està en com saber que un recurs ha canviat sense preguntar al servidor.

La solució implementada consisteix en que en lloc de posar a l'HTML enllaços a "imatge.png", es posarà un enllaç a "imatge.png?v=85cb69dd5b91a6713300e51fc5db9d1b", on aquest codi és la codificació MD5 del fitxer. El servidor evidentment ignorarà el paràmetre amb la codificació, però el navegador això no ho sap. Quan s'actualitzi "imatge.png" aquesta codificació canviarà i per tant l'enllaç també. Així doncs, en el mateix instant que tinguem una versió nova de "imatge.png" el navegador haurà de demanar-la al servidor (ell no té res a la seva caché sobre el nou enllaç).

Per a dur a terme la solució s'ha establert diferents fases. En primer lloc, quan es modifica un recurs estàtic (full d'estils, fitxer Javascript o imatge) cal executar un script que fa les següents tasques:

1. Obté tots els fitxers Javascript i els minimitza utilitzant el compressor **jsjam**. Un cop fet, els desa en el directori públic.
2. Copia totes les fulles d'estil al directori públic.
3. Copia totes les imatges al directori públic.

4. Per cada fitxer del directori públic en crea un altre amb el mateix nom, però acabat amb l'extensió .md5, que conté la seva codificació MD5.

El segon pas succeeix quan s'inicia l'aplicació. Aquesta obté tots els fitxers .md5 i crea un diccionari amb el nom del recurs com a clau i la codificació md5 com a valor.

El tercer i últim pas té lloc a l'hora de generar les pàgines web. Totes les URLs de recursos de l'aplicació estaran creades a través d'una funció que els hi afegirà la codificació MD5 corresponent i els hi assignarà el subdomini corresponent per poder realitzar descàrregues paral·leles.

■ Google Maps

Per poder fer ús de totes les funcionalitats que ofereix Google Maps primer cal obtenir una clau de la seva API. Un cop obtinguda només cal afegir el següent codi dins les pàgines que necessitaran el mapa.

```
1 <script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg"
2       type="text/javascript">
3 </script>
```

Llistat 8.6: Càrrega de l'API de Google Maps

Aquesta URL apunta a la localització dels fitxers Javascript que inclouen tot el necessari per utilitzar l'API de Google Maps. En aquest cas la clau utilitzada és "abcdefg" i identifica el lloc web que utilitza les funcions.

Un cop fet això, ja s'esta en disposició d'utilitzar el servei. Algunes funcionalitats que s'ha implementat són:

- **Cerca de zones segons el text entrat:** Alguns exemples de cerca serien noms de ciutat, direccions de carrers, noms d'edificis...
- **Obtenció de la localització d'una IP:** Aquesta funcionalitat permet situar el mapa a la zona des d'on està connectat l'usuari, ja que segurament serà on voldrà cercar events.
- **Generació de rutes:** Si un usuari ha entrat la seva localització, el sistema podrà generar la ruta que ha de seguir per arribar fins un event concret.

■ Internacionalització

Per arribar a un gran nombre d'usuaris cal oferir l'aplicació en diferents llenguatges. Per aconseguir-ho s'utilitza el programari **babel**, que fa ús de la llibreria **gettext** per traduir aplicacions fetes en Python.

Aquest sistema consisteix en envoltar les cadenes de text amb la funció gettext que, per estalviar temps d'escriptura, està normalment assignada a _.

Així doncs, si es vol traduir una cadena cal escriure-la de la següent manera:

```
1 print _("My name is %s.") % name
```

Llistat 8.7: Cadena de text dins la funció gettext

Un cop es té totes les cadenes dins la funció gettext s'extreu els missatges i es desen en una plantilla **.pot**. El següent pas és generar un fitxer **.po**, a partir de la plantilla, i per cada llenguatge desitjat crear un fitxer del tipus es_CA.po. Una entrada senzilla per traduir el missatge anterior al català es veuria així:

```
1 #: src/name.py:36
2 msgid "My name is %s."
3 msgstr "El meu nom és %s."
```

Llistat 8.8: Entrada d'un fitxer .pot

Finalment, els arxius .po són compilats en un arxiu binari **.mo** amb msgfmt[1]. D'aquesta manera estan preparats per ser distribuïts amb l'aplicació.

■ Optimització en el tractament de direccions IP

Degut a que moltes taules de la base de dades contenen un camp per guardar les direccions IP, ja que és el principal sistema per evitar abusos dels usuaris, s'ha buscat un sistema per accelerar tant la lectura, com l'escriptura de les mateixes.

Si es guarda les direccions IP amb el seu format original, s'han de desar en camps de text i això comporta un augment considerable en el tamany tant dels camps, com dels índexs. Fet que amb un gran volum de dades redueix molt el rendiment general. Per això s'ha buscat un sistema per desar-les com a enters utilitzant la següent fórmula, com es mostra en els exemples:

$$192.168.0.1 = 192 * 256^3 + 168 * 256^2 + 0 * 256^1 + 1 * 256^0 = 3232235521$$

$$192.16.80.1 = 192 * 256^3 + 16 * 256^2 + 80 * 256^1 + 1 * 256^0 = 3222294529$$

Degut a que el llenguatge de programació Python no incorpora cap funció per fer la transformació, s'ha hagut de crear unes funcions que ho facin utilitzant llibreries ja existents.

```
1 def ip2long(ip):
2     from struct import unpack
3     from socket import inet_aton
4 
5     return unpack('!L', inet_aton(ip))[0]
6 
7 def long2ip(long):
8     from struct import pack
```

[1]Utilitat de Unix per compilar plantilles de missatges .po a un format binari sense canviar les plantilles.

```
9     from socket import inet_ntoa
10
11    return inet_ntoa(pack('!L', long))
```

Llistat 8.9: Transformació de direccions IP

■ Localització d'events propers

El sistema disposa de la funcionalitat per trobar els events propers a un event concret. Per trobar-los s'utilitza la formula de **Haversine**, que permet calcular distàncies entre un parell de coordenades en una esfera. La implementació utilitzada en el sistema és la següent:

```
1  def get_closest(self, magnitude='km', distance=25, limit=5):
2
3      if magnitude == 'km':
4          ratio = 6371
5      else:
6          ratio = 3959
7
8      sql = "SELECT event_id, ( %d * acos( cos( radians(%3.6f) ) * cos( radians(
           lat ) ) * cos( radians( lng ) - radians(%3.6f) ) + sin( radians(%3.6f) )
           * sin( radians( lat ) ) ) ) AS distance FROM event_markers HAVING
           distance < %d ORDER BY distance LIMIT 0 , %d" % (ratio, self.markers[0].
           lat, self.markers[0].lng, self.markers[0].lat, distance, limit)
9
10     events = meta.Session.execute(sql).fetchall()
11
12     return events
```

Llistat 8.10: Implementació de la fórmula de Haversine

Aquesta funció permet obtenir els events més propers en un radi de distància. Permet escollir les unitats de mesura (kilòmetres o milles), el nombre d'events obtinguts i el radi de la circumferència que contindrà els events.

■ Karma i usuaris

En primer lloc cal comentar que la implementació del karma utilitzada en el sistema és una adaptació de la creada per al lloc web **menéame.net**.

Funcionament del càlcul de karma

El karma és un valor que va de 1 a 20, sent 6 el que obté un usuari al registrar-se en el sistema.

El càlcul del karma es realitza amb l'activitat realitzada en el sistema en els 15 dies anteriors a l'execució del procés que el calcula, qualsevol vot anterior no hi influeix. Si no s'ha votat cap event en els últims 15 dies és molt probable que es perdi karma, però en

cap cas pujarà, ja que fa falta haver realitzat un mínim de 4 vots per pujar-lo. El karma s'actualitza tots els dies a les 6:00 AM. El nou karma serà la combinació de 3 elements:

Vots rebuts sobre els events de l'usuari

- Els vots positius sobre un event augmenten el karma de l'usuari que l'ha enviat i els negatius el disminueixen.
- Si un event d'un usuari és descartat fa disminuir el karma.

Vots positius realitzats

- Els vots positius a events que no han estat descartats sempre augmenten el karma, encara que en un valor molt petit.
- Votar després de l'inici d'un event no afecta en res.
- Els vots positius sobre events que han sigut descartats disminueix el karma només si han estat realitzats abans de 24h.

Vots negatius realitzats

- Els vots negatius sobre events que han sigut descartats augmenten aquest valor sempre i quan hagin estat realitzats durant les 24 hores següents a l'enviament de l'event. Votar negatiu després d'aquest període a un event que vagi a ser descartat o votar a un event ja descartat fa temps no afecta al karma.
- Els vots negatius sobre events que no han estat descartats disminueix aquest valor. En aquest cas es penalitza més que en el cas de vots positius.

La suma d'aquests tres elements dóna el valor per calcular el nou karma.

Nou karma

- Si aquest nou karma és major que el que ja tenia l'usuari, es conserva el 80% del seu karma actual i se li suma el 20% del nou karma.
- Si aquest nou karma és menor que el que ja tenia l'usuari, es conserva el 95% del seu karma actual i se li suma el 5% del nou karma.

No s'especifica les fórmules de càlcul de cadascun dels tres elements del nou karma ja que el sistema no està prou testejat com per garantir que els coeficients de transformació són correctes.

■ Karma i events

El càlcul del karma dels events és més senzill que el dels usuaris. Cada vot positiu augmentarà el karma de l'event en el valor del karma de l'usuari. El mateix passarà amb un vot negatiu però enlloc de sumar, restarà.

A més hi haurà un procés diari que recalcularà el karma dels events en funció del temps que falti pel seu inici. D'aquesta manera s'aconseguirà augmentar les possibilitats perquè els usuaris trobin events a falta de poc temps pel seu inici.

■ Recomanacions utilitzant intel·ligència col·lectiva

Es sap que la millor manera d'obtenir recomanacions de productes, pel·lícules o events és preguntant als amics. Normalment alguns dels amics tenen millors "gustos" que altres, i això s'aprèn a mesura d'observar que a ells els hi agrada les mateixes coses que a un mateix. Com més opcions hi ha disponibles, més difícil és decidir basant-se en les respostes d'un petit grup de gent. Aquest és el motiu pel qual s'ha desenvolupat un conjunt de tècniques anomenades **filtratge col·laboratiu**.

Un algorisme de filtratge col·laboratiu normalment funciona buscant en un gran grup de gent i trobant un petit conjunt amb gustos similars a un individu. Mira les altres coses que els hi agrada al grup i crea un rànking de suggeriments. Hi ha diferents maneres de decidir quines persones són similars i combinar els seus gustos per crear una llista. A continuació es mostra algunes tècniques:

Distància euclidiana

Una forma molt fàcil de calcular la similaritat és utilitzant la distància euclidiana, que agafa els events que la gent ha votat en comú i els utilitza com a eixos d'un gràfic. Es situa la gent en el gràfic i es veu la proximitat a la que estan uns dels altres.

$$ed = \sum_{i=1}^{n} \sqrt{(p_i - q_i)^2}$$

Aquesta fórmula calcula la distància, que és inferior per la gent més similar. No obstant, es necessita que doni valors més alts a la gent més similar. Això s'aconsegueix afegint 1 al coeficient de la funció (per no obtenir un error de divisió entre zero) i invertint-lo.

A continuació es pot veure la implementació utilitzada en el sistema.

```
# Returns a distance-based similarity score for person1 and person2
def sim_distance(prefs, person1, person2):
    # Get the list of shared_items
    si = {}
    for item in prefs[person1]:
        if item in prefs[person2]:
            si[item] = 1

    # if they have no ratings in common, return 0
    if len(si) == 0: return 0

    # Add up the squares of all the differences
    sum_of_squares = sum([pow(prefs[person1][item] - prefs[person2][item], 2)
                          for item in prefs[person1] if item in prefs[person2]])

    return 1 / (1 + sum_of_squares)
```

Llistat 8.11: Algorisme de la distància euclidiana

Coeficient de correlació de Pearson

Una forma una mica més sofisticada de determinar la similaritat entre els interessos de persones és utilitzant el coeficient de correlació de Pearson. El coeficient de correlació és una mesura de com encaixen dos conjunts de dades en una línia recta. La seva fórmula és més complicada que la de la distància euclidiana, però tendeix a donar millors resultats.

$$pcs = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}$$

Aquesta funció torna un valor entre -1 i 1. Un valor de 1 significa que les dues persones tenen els mateixos vots sobre els mateixos events. A diferència amb el resultat de la distància euclidiana, no es necessita modificar aquest valor per obtenir el resultat en l'escala correcta.

A continuació es pot veure la implementació utilitzada en el sistema.

```
# Returns the Pearson correlation coefficient for p1 and p2
def sim_pearson(prefs, p1, p2):
    # Get the list of mutually rated items
    si = {}
    for item in prefs[p1]:
        if item in prefs[p2]: si[item] = 1

    # Find the number of elements
    n = len(si)

    # if they are no ratings in common, return 0
    if n == 0: return 0

    # Add up all the preferences
    sum1 = sum([prefs[p1][it] for it in si])
    sum2 = sum([prefs[p2][it] for it in si])

    # Sum up the squares
    sum1Sq = sum([pow(prefs[p1][it], 2) for it in si])
    sum2Sq = sum([pow(prefs[p2][it], 2) for it in si])

    # Sum up the products
    pSum = sum([prefs[p1][it] * prefs[p2][it] for it in si])

    # Calculate Pearson score
    num = pSum - (sum1 * sum2 / n)
    den = sqrt((sum1Sq - pow(sum1, 2) / n) * (sum2Sq - pow(sum2, 2) / n))
    if den == 0: return 0

    r = num / den

    return r
```

Llistat 8.12: Algorisme del coeficient de correlació de Pearson

Així doncs, el sistema implementat disposa d'un procés de recomanacions, que utilitza els algorismes descrits, per mostrar als usuaris events que els podrien resultar interessants.

■ Núvol d'etiquetes

Per oferir un ràpid accés a les cerques d'events, l'aplicació disposa d'un "núvol" amb les etiquetes més utilitzades en els events entrats pels usuaris (veure figura 8.3). Aquesta funcionalitat permet a l'usuari fer-se ràpidament una idea de la temàtica dels events més populars, ja que hi ha una gran influència del factor visual, i en cas d'interessar-li, realitzar la cerca només seguint l'enllaç de l'etiqueta desitjada.

La solució implementada en el sistema consisteix en obtenir les etiquetes més utilitzades ordenades per la seva freqüència d'aparició. Tot seguit s'estableix un rang en el tamany de les lletres (p.ex. entre 8pt i 20pt) i s'obté el coeficient del tamany de les etiquetes. Suposant que la freqüència de l'etiqueta més repetida és 50, s'obtindrà:

$$coef = (max_pts - min_pts) \ / \ max_freq = (20 - 8) \ / \ 50 = 0.24$$

Un cop es té el coeficient, es passa a obtenir el tamany de cadascuna de les etiquetes. Suposant que es té una etiqueta que apareix 28 vegades s'obtindrà:

$$size = min_pts + count * coef = 8 + 28 * 0.24 = 14.72 \ pt$$

Figura 8.3: Núvol d'etiquetes

■ Cerques

El sistema disposa del motor de cerca **sphinx** per obtenir els events que millor s'adeqüin al text entrat. Aquest programari permet indicar quins camps d'una base de dades es vol

utilitzar per realitzar les cerques dins el seu contingut.

En el sistema dissenyat s'ha cregut convenient realitzar les cerques en els següents camps:

- Títol
- Descripció
- URL
- Etiquetes

Per defecte el cercador busca el text entrat en tots els camps, però, per agilitzar l'ús als usuaris, s'ha dotat al sistema d'un conjunt de filtres mitjançant paraules claus al principi de la cerca:

- title: Permet realitzar la cerca d'events que continguin el text entrat en el títol.
- url: Permet realitzar la cerca només mirant en el contingut de l'URL.
- tag o tags: Permet realitzar la cerca d'events que continguin etiquetes amb el text entrat.

Alguns exemples de cerques poden ser:

1. "title: concert rock"
2. "url: google"
3. "tag: girona udg"

■ API

El sistema disposa d'una API per proporcionar un conjunt de funcions per ser utilitzades per altres aplicacions.

Consulta de l'estat d'una URL

És un script al que se li pot consultar sobre l'estat d'una URL i retorna una línia de text amb informació rellevant. L'URL de l'script és http://eventxplore.com/api/url i com argument ha de rebre l'URL d'algun event. Per exemple:

http://eventxplore.com/api/url?url=http://www.barcelonamarato.es

Retornarà:

OK http://eventxplore.com/event/marato-de-barcelona 35 pending

Són quatre camps. El primer (OK o KO) indica si l'URL ja està en el sistema. El segon camp és l'URL d'aquest event. El tercer camp és el nombre de vots que té. El quart camp és l'estat de l'event, els valors possibles són *pending* si l'event encara no ha començat, *started* si l'event s'està realitzant en aquest mateix instant, *finished* si ja s'ha acabat i *discard* si ha estat descartat.

Si l'event no existeix sortirà una línia com:

KO http://eventxplore.com/submit?url=http://www.barcelonamarato.es

El primer camp indica que no està en el sistema i el segon l'URL per enviar aquest event.

Enviament d'events

Aquesta funcionalitat permet afegir a la pàgina "oficial" d'un event un enllaç cap al procés d'enviament d'events de l'aplicació. D'aquesta manera es permet que qualsevol visitant de l'event pugui enviar-lo al sistema directament. L'URL de l'enllaç és http://eventxplore.com/submit i rep com argument l'URL de l'event. Per exemple:

http://eventxplore.com/submit?url=http://us.pycon.org

8.3 Execució de les proves unitàries

Per l'execució de les proves unitàries amb el framework Pylons s'ha utilitzat el programari **nose**. Executant *nosetests* en el directori del projecte permet realitzar totes les proves unitàries que s'hagi creat en el directori tests.

```
1 nosetests --with-pylons=test.ini
```

Llistat 8.13: Execució de les proves unitàries

Si l'execució de totes les proves és correcta obtindrem el següent resultat:

```
1 ..........................
2 ----------------------------------------------------------------------
3 Ran 26 tests in 58.999s
4
5 OK
```

Llistat 8.14: Resultat proves unitàries

En canvi, si l'execució d'alguna de les proves falla es mostrarà la sortida de l'error produït:

```
1  ..........F..............
2  ======================================================================
3  FAIL: test_show_votes (eventxplore.tests.functional.test_event.
       TestEventController)
4  ----------------------------------------------------------------------
5  Traceback (most recent call last):
6    File "/home/jordi/workspace/eventxplore/eventxplore/tests/functional/test_event
         .py", line 27, in test_show_votes
7      self._test_show(response, event.uri)
8    File "/home/jordi/workspace/eventxplore/eventxplore/tests/functional/test_event
         .py", line 33, in _test_show
9      assert response.c.map_vars
10 AssertionError
11
```

```
12 ----------------------------------------------------------------------
13 Ran 26 tests in 57.620s
14
15 FAILED (failures=1)
```

Llistat 8.15: Resultat proves unitàries

8.4 Elaboració dels manuals d'usuari

Per l'elaboració dels manuals d'usuari s'ha optat per la utilització d'un wiki[2]. El principal avantatge del wiki és que permet crear i millorar les pàgines de forma instantània, donant una gran llibertat a l'usuari, i a través d'una interfície molt simple.

Per la seva creació s'ha aprofitat que ja es disposa del programari **Trac** per al control de versions i que també permet la creació de pàgines wiki. Els resultats es poden veure a la figura 8.4.

[2]Lloc web on les seves pàgines poden ser editades per múltiples voluntaris a través del seu navegador web.

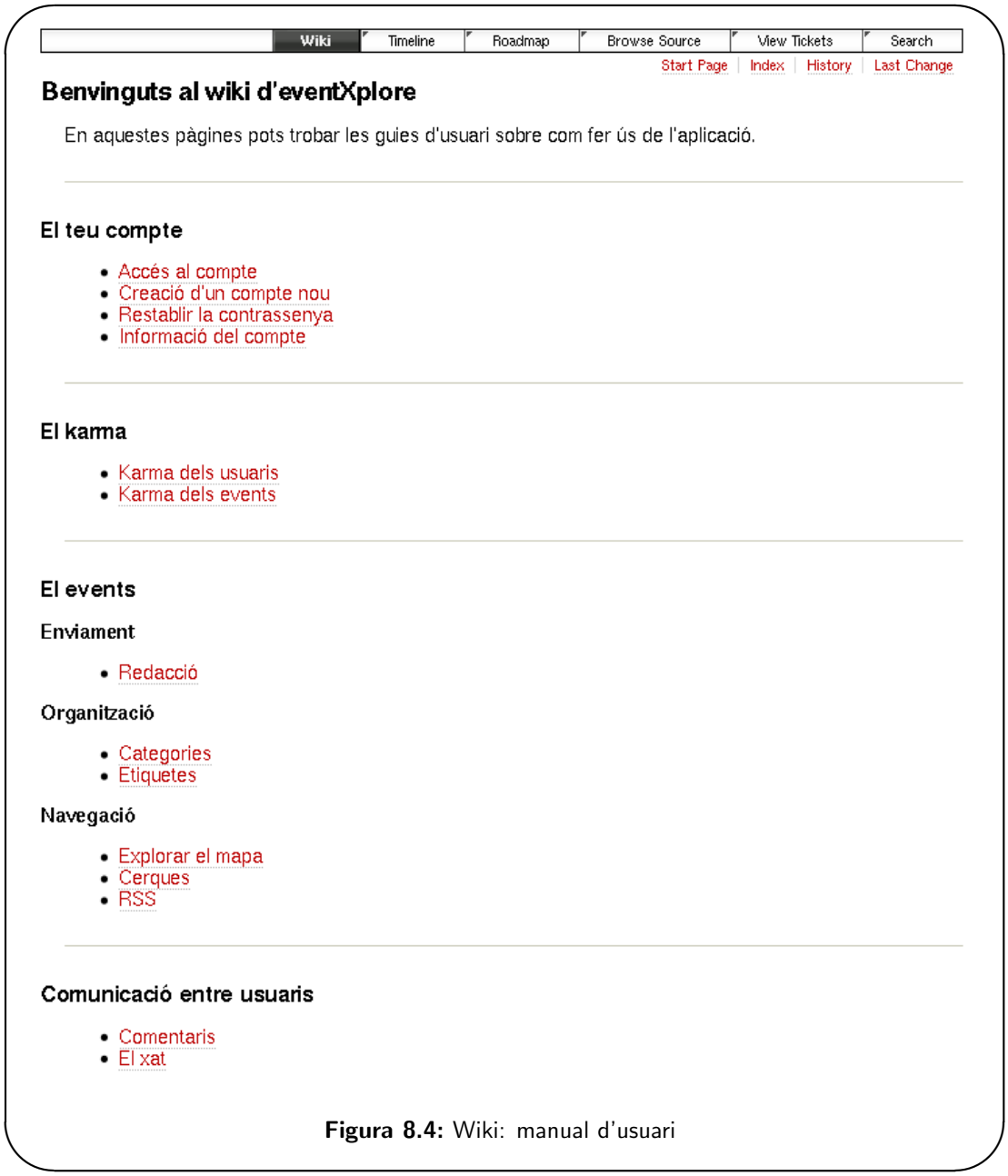

Figura 8.4: Wiki: manual d'usuari

9. Implantació i Acceptació del Sistema (IAS)

9.1 Incorporació del sistema a l'entorn final

9.1.1 Realització de la instal·lació

Si s'està utilitzant un servidor de producció amb múltiples aplicacions que utilitzen Pylons, és útil ser capaç de mantenir-les completament separades unes de les altres. D'aquesta manera es permet utilitzar diferents versions de les mateixes llibreries Python per cada instància que s'executa. Això es pot aconseguir en sistemes Unix creant diferents usuaris per cada aplicació Pylons i utilitzant una instal·lació virtual de Python per cadascun.

En el sistema implementat s'ha seguit el següent procés per a crear un entorn aïllat que permeti executar una instància de l'aplicació:

```
1  # Primer es crea l'usuari eventxplore.
2  adduser eventxplore
3
4  # Tot seguit s'instal·la el programari necessari.
5  apt-get install python python-dev subversion
6
7  # Un cop es té tot el necessari instal·lat cal assegurar-se que s'està utilitzant
8  # el sistema amb l'usuari creat i des del seu directori personal.
9  su eventxplore
10 cd ~
11
12 # Es descarrega el fitxer necessari per crear la instal·lació virtual.
13 wget http://peak.telecommunity.com/dist/virtual-python.py
14
15 # Després s'inicia la instal·lació virtual de python utilitzant la instal·lació
16 # de Python del sistema anfitrió.
17 python virtual-python.py
```

Llistat 9.1: Instal·lació de l'entorn virtual

Això copia el binari de Python al directori /home/eventxplore/bin de l'usuari eventxplore i crea els enllaços simbòlics de les llibreries del sistema. Això significa que l'executable /bin/python tindrà accés a les mateixes llibreries que la instal·lació Python del sistema, però qualsevol programa afegit no afectarà al sistema amfitrió.

En aquest punt ja es disposa de l'entorn virtual, però falta instal·lar i configurar tot el programari necessari.

```
1  # Cal obtenir l'instal·lador easy_install:
2  wget http://peak.telecommunity.com/dist/ez_setup.py
```

```
~/bin/python ez_setup.py

# Aquestes comandes afegeixen la comanda easy_install al directory bin
# de l'usuari eventxplore, per tant ja es pot instalar Pylons:
~/bin/easy_install Pylons

# Si es vol instal·lar més paquets es pot fer així:
~/bin/easy_install package_name

# Els paquests seran instal·lats a l'entorn virtual sense afectar
# al sistema anfitrió.
```

Llistat 9.2: Instal·lació del programari

■ Configurant Bash

Per facilitar les coses es pot configurar el compte de l'usuari perquè, per defecte, siguin executats els scripts del seu directori bin. Això significa que es pot executar les comandes python, paster i easy_install i s'utilitzarà les de l'entorn virtual i no les del sistema.

Només cal editar el fitxer .bash_profile del directori personal de l'usuari eventxplore i afegir:

```
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
    PATH=~/bin:"${PATH}"
fi
```

Llistat 9.3: Configuració de bash

Un cop fet això, quan es torni a accedir al compte de l'usuari, els executables del directori personal bin tindran preferència sobre els del sistema, fet que pot estalviar molts mals de cap.

■ Instal·lació de l'aplicació

Per evitar problemes, la instal·lació de l'aplicació ha de ser el més senzilla possible. El sistema implementat disposa un conjunt d'eines per empaquetar l'aplicació en arxius **.egg** de Python. Aquest sistema permet comprovar dependències i instal·lar-les si es necessita.

Per crear un egg de l'aplicació cal situar-se en el seu directori arrel i executar:

```
python setup.py bdist_egg
```

Llistat 9.4: Creació d'un egg del projecte

El resultat d'aquesta comanda serà la creació d'un fitxer .egg, amb el nom de l'aplicació i el número de versió, situat en el directori dist. Suposant que l'aplicació es troba en la versió 0.1 s'obtindrà el fitxer eventxplore-0.1.0dev-py2.4.egg.

Per instal·lar l'aplicació només caldrà fer ús de la comanda easy_install:

```
easy_install -f /directori/on/es/troba/el/fitxer/egg/ eventxplore==0.1.0
```

Llistat 9.5: Instal·lació de l'aplicació

9.2 Realització de proves d'implantació

9.3 Realització de proves d'acceptació del sistema

En una aplicació web, és fonamental que els usuaris hi puguin accedir amb certa fluïdesa. Per això és necessari que el sistema respongui en un temps raonable sota qualsevol circumstància.

Tot seguit es mostra el resultat de les proves d'estrès que s'ha realitzat, tenint en compte que n és el nombre de peticions a realitzar i c el nombre de peticions concurrents. Les proves han estat realitzades amb el programa **Apache Benchmark** sobre el servidor de preproducció.

Rendiment servint un fitxer estàtic de 50k (n=1000, c=10)

- Requests per second: 17.31 [#/sec] (mean)
- Time per request: 577.651 [ms] (mean)
- Time per request: 57.765 [ms] (mean, across all concurrent requests)
- Transfer rate: 999.15 [Kbytes/sec] received

Accedint a la pàgina principal, aproximadament 10k (n=1000, c=1)

- Requests per second: 1.56 [#/sec] (mean)
- Time per request: 642.997 [ms] (mean)
- Time per request: 642.997 [ms] (mean, across all concurrent requests)
- Transfer rate: 26.26 [Kbytes/sec] received

Accedint massivament a la pàgina principal (n=250, c=250)

- Requests per second: 6.81 [#/sec] (mean)
- Time per request: 36735.477 [ms] (mean)
- Time per request: 146.942 [ms] (mean, across all concurrent requests)
- Transfer rate: 116.14 [Kbytes/sec] received

Percentage of the requests served within a certain time (ms)
50% 19730
66% 24681

75% 27923
80% 30273
90% 33526
95% 34698
98% 35140
99% 35193
100% 35867 (longest request)

Accedint a la pàgina principal durant una llarga estona (n=5000, c=50)

- Requests per second: 7.20 [#/sec] (mean)
- Time per request: 6940.087 [ms] (mean)
- Time per request: 138.802 [ms] (mean, across all concurrent requests)
- Transfer rate: 121.65 [Kbytes/sec] received

Percentage of the requests served within a certain time (ms)
50% 6971
66% 7348
75% 7502
80% 7589
90% 7785
95% 7940
98% 8150
99% 8460
100% 12002 (longest request)

Com es pot comprovar el sistema aguanta bé en condicions de poc tràfic, però si augmenta el nombre de peticions els temps de resposta es disparen.

En l'hipotètic cas d'hora punta, amb 250 usuaris accedint al mateix temps, és capaç de respondre el 50% de les peticions en menys de 20 segons. Aquest temps és inacceptable per una aplicació web.

En el cas que hi hagi 50 usuaris al mateix temps durant una llarga estona, és capaç de respondre el 95% de les peticions en menys de 8 segons. Aquests temps no són gaire bons i ens indiquen que el sistema necessita una ampliació.

Cal tenir en compte que, per problemes de disponibilitat, les proves han estat realitzades en el servidor de preproducció, que es tracta d'un ordinador personal, i no en el de producció. L'estimació és que les mateixes proves en el servidor de producció, no haurien de suposar cap problema i els temps de resposta serien acceptables.

Part III

Conclusions i futur

10. Conclusions

Finalitzat el projecte, cal valorar l'assoliment de tots els objectius que es van marcar a l'inici. Si s'analitza objectivament, s'ha de dir que l'aplicació resultant compleix amb tots els requisits i suggeriments especificats.

S'ha aconseguit implementar una aplicació que supleix la mancança que va motivar la seva realització. D'aquesta manera es disposa d'un lloc on els usuaris poden compartir tota la informació referent als events que creuen interessants. El fet d'haver dissenyat el sistema des de l'inici amb la intenció d'aprofitar les capacitats de les xarxes socials a Internet, fa que s'obri un gran ventall de possibilitats cap al futur, un futur on tot sembla apuntar que aquest tipus d'aplicacions hi tenen molt a dir.

La realització d'aquest projecte ha suposat per a l'autor l'assoliment de nous coneixements, alguns dels quals eren familiars i altres completament desconeguts. Si s'ha de destacar alguns dels aspectes més interessants del projecte, sens dubte són:

- La utilització del framework per desenvolupament web, molt flexible i àgil.
- La construcció d'un sistema escalable tenint en compte petits, però importants, detalls d'optimització.
- El descobriment del fascinant món de la intel·ligència col·lectiva.

A l'inici del projecte es definien, en un diagrama de Gantt, unes fases i unes dates límit pel seu desenvolupament. Ara, un cop finalitzat, es pot concloure que s'ha seguit força bé i sense contratemps importants. No obstant, destacar que la redacció de la memòria ha durat més del que es pensava degut a, entre altres problemes, la falta de familiaritat amb LaTeX.

Finalment, un punt important a tenir en compte, a més d'adquirir nous coneixements, ha estat la presa de decisions; sobretot en tot allò que fa referència a decisions de com implementar certs mòduls i com definir i especificar els requisits del sistema.

En termes generals, s'està molt satisfet del projecte i es té la convicció que a més d'ampliar-lo i publicar-lo a Internet, serà atractiu i de gran ajuda per a molts internautes d'arreu del món.

11. Futur

11.1 Millores i ampliacions

Com a possibles millores i ampliacions del projecte s'ha plantejat:

- Fer un ús més ampli de les funcionalitats que proporciona l'API de Google Maps. Un exemple seria l'ús de marcadors i finestres informatives personalitzades.
- La creació d'un arxiu KML que permeti l'exportació de tota la informació del sistema com una capa de l'aplicació Google Earth.
- La utilització de l'API de Google Calendar per permetre als usuaris registrar events del sistema en els seus calendaris personals.
- Entrar més a fons en el món de la intel·ligència col·lectiva ja que permet proporcionar als usuaris funcionalitats molt útils, que faciliten l'ús diari de l'aplicació.
- L'extensió de l'API del sistema per fomentar-ne l'ús des d'aplicacions externes.
- La creació d'un sistema de monitorització que permeti veure en temps real tota l'activitat que es produeix en el sistema: l'enviament d'events nous, els vots realitzats, el comentaris...

Bibliografia (Llibres i artícles)

[1] Brian W. Fitzpatrick i C. Michael Pilato Ben Collins-Sussman, editor. *Version Control with Subversion*. TBA, 2008.

[2] Magnus Lie Hetland, editor. *Beginning Python: From Novice to Professional*. Apress, 2005.

[3] Mark Lutz i David Ascher, editor. *Learning Python*. O'Reilly, 1999.

[4] Jonathan Chaffer i Karl Swedberg, editor. *Learning jQuery: Better Interaction Design and Web Development with Simple JavaScript Techniques*. Packt Publishing, 2007.

[5] Bear Bibeault i Yehuda Katz, editor. *jQuery in Action*. Manning Publications Co., 2007.

[6] Toby Segaran, editor. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly, 2007.

[7] Steve Souders, editor. *High Performance Web Sites*. O'Reilly, 2007.

Bibliografia (Altres fonts)

[8] Authkit - WSGI Authentication and Authorization Tools [online]. URL: `http://authkit.org/`.

[9] Babel - A collection of tools for internationalizing Python applications [online]. URL: `http://babel.edgewall.org/`.

[10] Best Practices for Speeding Up Your Web Site [online]. URL: `http://developer.yahoo.com/performance/rules.html`.

[11] Escalabilidad en el servidor [online]. Març 2008. URL: `http://programa-con-google.blogspot.com/2008/03/escalabilidad-en-el-servidor.html`.

[12] FormEncode [online]. URL: `http://formencode.org/`.

[13] Google Maps API [online]. URL: `http://code.google.com/apis/maps/index.html`.

[14] Google Maps API Tutorial [online]. URL: `http://econym.googlepages.com/index.htm`.

[15] jQuery [online]. URL: `http://jquery.com/`.

[16] jQuery UI [online]. URL: `http://ui.jquery.com/`.

[17] jsjam - Compress JavaScript code [online]. URL: `http://www.anvilon.com/software/jsjam/`.

[18] Mako Templates for Python [online]. URL: `http://www.makotemplates.org/`.

[19] Métrica Versión 3 [online]. URL: `http://www.csi.map.es/csi/metrica3/index.html`.

[20] Nginx [online]. URL: `http://wiki.codemongers.com/Main`.

[21] Python Imaging Library (PIL) [online]. URL: `http://www.pythonware.com/products/pil/`.

[22] Pylons Python Web Framework [online]. URL: `http://pylonshq.com/`.

[23] PyRSS2Gen [online]. URL: `http://www.dalkescientific.com/Python/PyRSS2Gen.html`.

[24] Python Programming Language – Official Website [online]. URL: `http://www.python.org/`.

[25] PEP 8 – Style Guide for Python Code [online]. URL: `http://www.python.org/dev/peps/pep-0008/`.

[26] simplejson - A simple, fast, extensible JSON encoder and decoder [online]. URL: `http://simplejson.googlecode.com/svn/tags/simplejson-1.9.1/docs/index.html`.

[27] Sphinx - Free open-source SQL full-text search engine [online]. URL: `http://www.sphx.org/`.

[28] SQLAlchemy - The Database Toolkit for Python [online]. URL: `http://www.sqlalchemy.org/`.

[29] Creating a Store Locator with PHP, MySQL and Google Maps [online]. Gener 2008. URL: `http://code.google.com/support/bin/answer.py?answer=87134&topic=11364&fromblog=true`.

[30] The Trac Project [online]. URL: `http://trac.edgewall.org/`.

[31] WebHelpers [online]. URL: `http://pylonshq.com/WebHelpers/index.html`.