

# **Desenvolupament d'un gestor de trobades, reunions i convocatòries usant la plataforma WAC**

**Jorge Torrente Tomas**  
Enginyeria Informàtica

**Carlos Sánchez Rosa**

09/01/2012

Al meu fill Pau

## Agraïments

Primer de tot he de donar les gràcies d'una forma molt i molt especial a la Mercè per la paciència que ha demostrat en molts dels innumerables dies (i nits) que he hagut de passar dedicat exclusivament a les PACs i pràctiques de la UOC.

Als meus pares, germana, sogres i cunyat per tot l'ajut i suport prestat durant la realització dels estudis.

I finalment a tota la comunitat de docents, estudiants i altre personal de la UOC per fer possible una educació a distància de tanta qualitat.

## Resum

El projecte final de carrera objectiu del present document, inclou l'anàlisi, disseny, implementació i documentació d'una aplicació per a dispositius mòbils utilitzant la plataforma WAC.

Funcionalment parlant, es tracta d'implementar una proposta de solució a una problemàtica molt freqüent en la vida real: la utilització del mòbil per tal d'organitzar les trobades d'un grup de persones o amics. D'una banda, quasi sempre existeix un intercanvi previ de missatges curts (SMS) amb la informació del lloc i hora, d'altra els assistents, moltes vegades, recorren a l'ús de mapes digitals per trobar el lloc acordat. Inclús quan quasi tothom ha arribat a la destinació però falta algú, és molt freqüent intercanviar un seguit de missatges curts per esbrinar on està la persona en qüestió i quan li manca per acudir. Aquests són els principals aspectes que l'aplicació construïda intenta facilitar.

Cal tenir present que el treball fet és la culminació dels estudis d'una enginyeria i en conseqüència ha de il·lustrar el fet que s'han comprès els conceptes i coneixements adquirits durant els estudis, a més a més de saber aplicar-los a la resolució d'un problema real. En aquest sentit, a nivell de metodologia s'ha utilitzat un cicle de vida en cascada, recurrent a la notació UML en l'elaboració dels documents d'anàlisi i disseny.

# Índex

1	Introducció.....	8
1.1	Justificació i context del PFC.....	8
1.2	Objectius.....	8
1.3	Enfocament i mètode seguit.....	9
1.4	Planificació.....	9
1.5	Productes obtinguts.....	9
1.6	Descripció dels capítols.....	12
2	Documentació de Requisits.....	12
2.1	Introducció.....	13
2.2	Domini de l'aplicació.....	13
2.3	Guions.....	13
2.3.1	Guió per a la instal·lació de l'aplicació.....	14
2.3.2	Guió per a la gestió de contactes.....	14
2.3.3	Guió per a l'organització de trobades.....	14
2.3.4	Guió per a l'assistència a una trobada.....	14
2.4	Casos d'ús.....	15
2.4.1	Actors.....	15
2.4.2	Diagrama de casos d'ús.....	15
2.4.3	Documentació textual.....	18
2.5	Requisits no funcionals.....	34
2.6	Requisits de la UI.....	34
3	Anàlisi de Requisits.....	35
3.1	Revisió de casos d'ús.....	35
3.2	Paquets d'anàlisi i serveis.....	35
3.3	Classes d'entitat.....	36
3.4	Diagrama d'estats.....	37
3.5	Classes de frontera, de control i de les operacions.....	37
4	Disseny.....	41
4.1	Arquitectura.....	41
4.2	Frameworks i programari base.....	43
4.3	Patrons.....	43
4.4	Persistència. Entitats i atributs.....	44
4.5	Revisió model estàtic.....	45
4.6	Classes de disseny.....	45
4.7	Disseny de la UI.....	47
5	Implementació.....	49
5.1	Introducció.....	49
5.2	Aspectes a destacar de la solució.....	49
5.2.1	Utilització de Sencha Touch segons el patró MVC.....	49
5.2.2	Reutilització de components.....	50
5.2.3	Invocació de l'aplicació client als serveis remots.....	51
5.2.4	Minimització de les dades enviades.....	51
5.2.5	Utilització de Google® Maps API.....	52
5.2.6	Georeferenciació.....	52
5.2.7	Operacions amb un elevat temps de resposta.....	53
5.2.8	Emmagatzematge local de les credencials.....	53
5.2.9	Servidor: Sessions.....	53
5.2.10	Servidor: Transaccionalitat.....	54
5.2.11	Servidor: Desconnexió dels objectes de negoci.....	54
5.2.12	Servidor: connexions JDBC.....	54
5.2.13	Servidor: Log.....	55
6	Evolució.....	56

.....	56
<a href="#">7 Conclusions.....</a>	<a href="#">57</a>
<a href="#">8 Glossari.....</a>	<a href="#">58</a>
<a href="#">Bibliografia.....</a>	<a href="#">59</a>
<a href="#">Annex 1: Manual d'Usuari.....</a>	<a href="#">60</a>

# Índex de figures

Figura 1: Distribució de tasques.....	10
Figura 2: Planificació del PFC.....	11
Figura 3: Diagrama del model del domini.....	13
Figura 4: Diagrama d'Actors.....	15
Figura 5: Diagrama de casos d'ús (I).....	15
Figura 6: Diagrama de casos d'ús (II).....	16
Figura 7: Diagrama de casos d'ús (III).....	17
Figura 8: Diagrama de casos d'ús (IV).....	18
Figura 9: Diagrama de paquets d'anàlisi.....	35
Figura 10: Diagrama de classes d'entitat.....	36
Figura 11: Diagrama d'estats d'un contacte/assistent.....	37
Figura 12: Diagrama de Seqüència "Organitzar trobada".....	38
Figura 13: Diagrama de Seqüència de "Detall trobada".....	38
Figura 14: Diagrama de Seqüència de "Modificar trobada".....	39
Figura 15: Diagrama de Seqüència de "Cancel·lar trobada".....	39
Figura 16: Diagrama de Seqüència corresponent a "Afegir assistent".....	40
Figura 17: Diagrama de desplegament.....	41
Figura 18: Jerarquia de classes de les ubicacions.....	45
Figura 19: Estructura de classes del component servidor.....	46
Figura 20: Diagrama de classes de servidor associades a la gestió de contactes.....	46
Figura 21: Diagrama de classes de servidor associades a la gestió de trobades.....	46
Figura 22: Diagrama de classes de servidor associades als serveis d'usuaris.....	46
Figura 23: Exemple de disseny gràfic de la interfície d'usuari.....	48
Figura 24: Aplicació del patró MVC en la UI.....	49
Figura 25: Definició de la gestió d'esdeveniments.....	50
Figura 26: Exemple de vista.....	50
Figura 27: Exemple de petició de refresc.....	51
Figura 28: Exemple d'invocació al component remot.....	51
Figura 29: Càlcul de la ruta per anar a una ubicació.....	52
Figura 30: Bloqueig de la UI en les invocacions remotes.....	53
Figura 31: Utilització de SQLite.....	53

# 1 Introducció

## 1.1 Justificació i context del PFC

Actualment és molt comú el que un grup de persones o amics quedin en un lloc per tal parlar, prendre alguna cosa, sopar, etc. Cada cop més s'utilitza el telèfon mòbil per a organitzar aquestes trobades: d'una banda mitjançant l'intercanvi de missatges curts (SMS) amb la informació del lloc i hora, d'altra mitjançant l'ús de mapes digitals que faciliten arribar al lloc acordat. Inclús quan quasi tothom ha arribat a la destinació indicada però falta algú, és molt freqüent intercanviar un seguit de missatges curts per esbrinar on està la persona en qüestió i quan li manca per acudir.

Consegüentment l'objectiu serà el d'integrar tots aquests aspectes de les trobades o reunions en una única aplicació de forma que es faciliti la seva organització i la forma d'arribar-hi.

Cal tenir present que l'orientació que vol donar-se a l'aplicació és d'una eina social però utilitzada dins d'àmbits reduïts, és a dir, entre grups de persones no massa amplis. Tot i que no existirà cap impediment per utilitzar-la per concentracions multitudinàries no se'n recomana el seu ús ja que, en aquest cas, molt possiblement existiran components de la interfície gràfica que poden resultar difícils d'utilitzar.

## 1.2 Objectius

Òbviament el primer objectiu del treball realitzat és l'obtenció de l'aplicació que satisfaci els requeriments funcionals anteriorment esmentats: facilitar l'organització i la forma d'arribar a una trobada.

Ara bé, al mateix temps podem identificar altres objectius com:

- Aplicar de forma pràctica els coneixements de programació orientada a objectes per tal de resoldre un problema real, i de forma anàloga en l'ús de la notació UML per especificar i dissenyar la proposta d'una solució
- Aprofundir en el coneixement de la tecnologia de serveis web i en l'ús de la tecnologia JPA
- Aprendre l'ús de *frameworks* de presentació basats en el llenguatge Javascript: Sencha Touch, Google® Maps API
- Aprendre les possibilitats que ofereix la plataforma WAC per al desenvolupament d'aplicacions mòbils
- Aplicar de forma pràctica molts dels patrons de disseny més populars

Finalment, tot i considerar-lo més aviat secundari, es troba el fet d'exemplificar com l'ús de les noves tecnologies pot augmentar la interrelació física de les persones, donat que cada cop més es tendeix a la posició contrària, és a dir, a traslladar al món virtual la majoria de les relacions socials.



## 1.3 Enfocament i mètode seguit

El cicle de vida del programari està constituït per la programació i tot el conjunt d'etapes que la precedeixen i la succeeixen. Per l'elaboració del PFC s'ha seguit el cicle de vida clàssic o en cascada, el qual es caracteritza perquè en cada etapa s'obtenen uns documents (*deliverables*) que són les bases de partida de l'etapa següent (que per tant, no pot començar abans que hagi acabat l'anterior) i mai no es torna a etapes passades.

Les fases per les que s'ha passat són:

Anàlisi prèvia. Es defineixen els grans trets del sistema de programari a desenvolupar. Es correspon amb la PAC1.

Anàlisi (de requisits). Definició detallada de les necessitats d'informació que haurà de resoldre el programari, sense tenir en compte els mitjans tècnics amb que s'haurà de dur a terme el desenvolupament del programari. Es correspon a la meitat de la PAC2.

Disseny. S'especifica "com el programari ha de fer la seva funció": arquitectura general, estructures de dades (base de dades), interfícies d'usuari. Es correspon a l'altra part de la PAC2.

Programació o codificació. Es tradueix el disseny a codi executable per l'ordinador. Es correspon amb la PAC3.

Prova. Consisteix a provar el programari des de diversos punts de vista d'una manera planificada i naturalment, localitzar i corregir els errors que es detectin.

Pel que fa a l'aprenentatge de noves tecnologies, *frameworks* o components a utilitzar dins el PFC, sempre s'ha seguit el mateix procés:

- Llegir la documentació associada.
- Cercar i seguir un conjunt de *tutorials* sobre el component, tecnologia o *framework*.
- Integració dins l'aplicació.

## 1.4 Planificació

Les principals activitats d'alt nivell a dur a terme són:

- Formació. Auto-aprenentatge de la plataforma WAC.
- Configuració de l'entorn de desenvolupament i de test.
- Anàlisi. Definició dels casos d'ús.
- Disseny. Definició de l'arquitectura i model de dades.
- Implementació. Codificació de la solució.
- Proves.
- Documentació.

En la planificació següent la durada correspon als dies durant els quals es realitza la tasca, no al seu esforç en hores.














		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		 <b>Formació plataforma WAC</b>	<b>12 días</b>	<b>lun 03/10/11</b>	<b>vie 14/10/11</b>	
2		Lectura de documentació	5 días	lun 03/10/11	vie 07/10/11	
3		Seguiment tutorials	7 días	sáb 08/10/11	vie 14/10/11	2
4		 <b>Configuració entorn desenvolupament</b>	<b>2 días</b>	<b>sáb 08/10/11</b>	<b>dom 09/10/11</b>	
5		IDE	1 día	sáb 08/10/11	sáb 08/10/11	2
6		Emulador / Disp. Físic	1 día	dom 09/10/11	dom 09/10/11	5
7		 <b>Configuració de l'entorn de test</b>	<b>5 días</b>	<b>lun 10/10/11</b>	<b>vie 14/10/11</b>	
8		Estudi de mercat	3 días	lun 10/10/11	mié 12/10/11	
9		Selecció proveïdor	0 días	mié 12/10/11	mié 12/10/11	8
10		Configuració entorn	2 días	jue 13/10/11	vie 14/10/11	9
11						
12		<b>Acceptació Pla de Treball</b>	<b>0 días</b>	<b>vie 07/10/11</b>	<b>vie 07/10/11</b>	
13		 <b>Anàlisi</b>	<b>15 días</b>	<b>sáb 15/10/11</b>	<b>sáb 29/10/11</b>	
14		Gestió de contactes	5 días	sáb 15/10/11	mié 19/10/11	1;4;12
15		Organització de trobades	5 días	jue 20/10/11	lun 24/10/11	14
16		Guia / Mapa	5 días	mar 25/10/11	sáb 29/10/11	15
17		 <b>Disseny</b>	<b>8 días</b>	<b>dom 30/10/11</b>	<b>dom 06/11/11</b>	
18		Arquitectura	5 días	dom 30/10/11	jue 03/11/11	13
19		Model de dades	3 días	vie 04/11/11	dom 06/11/11	18
20		<b>PAC2</b>	<b>0 días</b>	<b>vie 04/11/11</b>	<b>vie 04/11/11</b>	
21		 <b>Implementació</b>	<b>52 días</b>	<b>lun 07/11/11</b>	<b>mié 28/12/11</b>	
22		Component servidor	8 días	lun 07/11/11	lun 14/11/11	17
23		Comp. Mòbil - Gestió de contactes	12 días	mar 15/11/11	sáb 26/11/11	22
24		Comp. Mòbil - Organització de trobades	12 días	dom 27/11/11	jue 08/12/11	23
25		Comp. Mòbil - Guia / Mapa	20 días	vie 09/12/11	mié 28/12/11	24
26		<b>PAC3</b>	<b>0 días</b>	<b>vie 09/12/11</b>	<b>vie 09/12/11</b>	
27		<b>Proves</b>	<b>2 días</b>	<b>jue 29/12/11</b>	<b>vie 30/12/11</b>	<b>21</b>
28		<b>Documentació</b>	<b>9 días</b>	<b>sáb 31/12/11</b>	<b>dom 08/01/12</b>	<b>27</b>
29		<b>PAC4</b>	<b>0 días</b>	<b>lun 09/01/12</b>	<b>lun 09/01/12</b>	

Figura 1: Distribució de tasques

És important esmentar el fet que les dates clau del treball estan prefixades per la direcció docent de la UOC, de forma que la planificació de la resta de tasques cal adaptar-la a elles.

I el diagrama de Gantt associat:

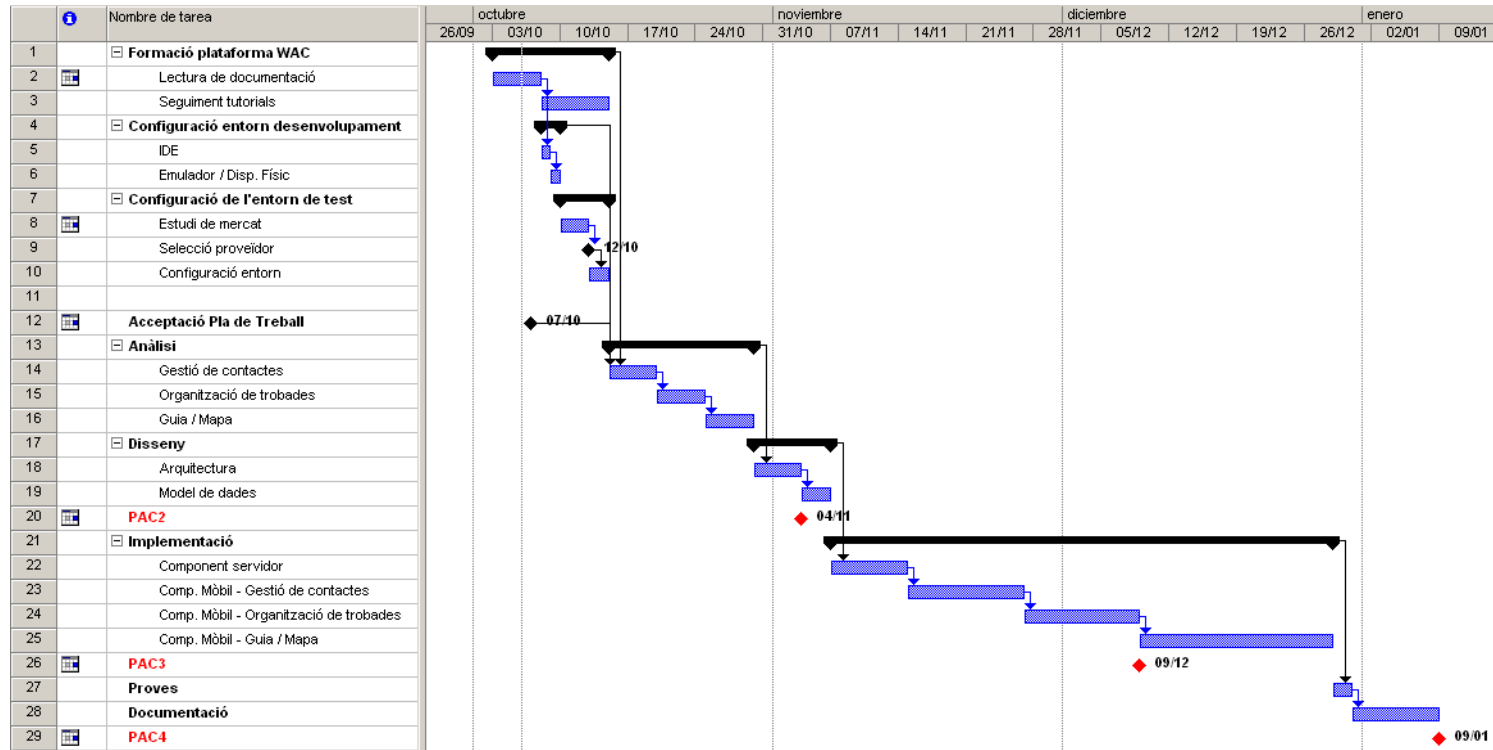


Figura 2: Planificació del PFC

Respecte aquesta planificació inicial cal esmentar que la finalització de la fase d'implementació ha sofert un retard d'una setmana que ha afectat al temps destinat a redactar la documentació. El motiu d'aquest retard el trobem en l'augment del temps necessari per aprendre a utilitzar correctament el component *Sencha Touch* i en la valoració optimista de l'esforç a dedicar al desenvolupament de la interfície gràfica. Tot i això, sempre s'ha intentat respectar al màxim les dates clau de les diferents entregues.

## 1.5 Productes obtinguts

Els productes obtinguts com a fruit del treball són:

- El fitxer instal·lable de l'aplicació mòbil (*widget*)
- Fitxer de distribució .war amb el component servidor (desplegat en el *núvol*)
- El codi font de totes les classes utilitzades (servidor/JAVA) i de tota la interfície d'usuari (client/Javascript)
- El fitxer de creació de les taules de la base de dades de l'aplicació
- El fitxer amb les sentències d'inicialització del model de dades creat
- El manual d'usuari de l'aplicació mòbil

## 1.6 Descripció dels capítols

En la resta de capítols de la memòria es comentaran les fases d'anàlisi, disseny i implementació per les quals ha passat el desenvolupament d'aquest PFC:

- Capítol 2. Documentació de requisits
- Capítol 3. Anàlisi
- Capítol 4. Disseny
- Capítol 5. Implementació
- Capítol 6. Evolució

## 2 Documentació de Requisits

### 2.1 Introducció

Cal recordar que l'objectiu principal del nou sistema és l'obtenció de l'aplicació que faciliti l'organització i la forma d'arribar a una trobada. Tot seguit es descriu un possible escenari de funcionament:

Una persona es descarrega l'aplicació al seu telèfon i la primera vegada que l'executa es registra en el sistema. Un cop registrat les opcions disponibles per l'usuari es dividiran en tres grups independents:

1. **Gestió de contactes.** On es permetrà:
  - Importar totes aquelles entrades de l'agenda del telèfon que es corresponguin a altres usuaris registrats.
  - Convidar, mitjançant un missatge SMS, a un usuari no registrat a que es descarregui l'aplicació i es registri en el sistema
2. **Organització de trobades.** Un usuari registrat podrà crear una reunió introduint una data i un conjunt de detalls complementaris. Un cop definida la trobada, l'usuari podrà convocar a qui vulgui i automàticament cada convidat rebrà una notificació en el seu terminal, que podrà ser rebutjada o acceptada.
3. **Guia / Mapa.** Utilitzant la càmera del dispositiu l'aplicació mostrarà juntament a la imatge capturada la ubicació de la trobada, la distància fins a ella i unes mínimes indicacions de com arribar-hi. Per als assistents que no ho hagin fet, existirà la possibilitat d'obtenir-ne la seva ubicació GPS mostrant-la en un mapa 2D (Google® Maps).

### 2.2 Domini de l'aplicació

A primer cop d'ull s'identifiquen els objectes o classes 'contacte', 'grup', 'assistent', 'organitzador', 'trobada', 'ubicació'. S'han posat alguns atributs a les classes de manera orientativa. Aquest diagrama no es farà servir per a etapes posteriors.

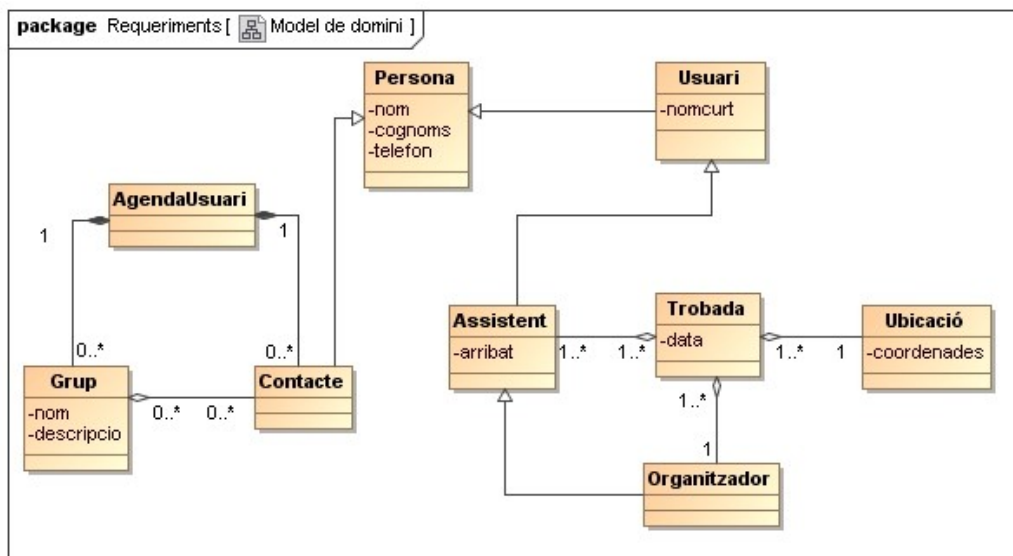


Figura 3: Diagrama del model del domini

## 2.3 Guions

### 2.3.1 Guió per a la instal·lació de l'aplicació

Una persona es descarrega l'aplicació al seu telèfon mitjançant un navegador i la primera vegada que l'executa es registra en el sistema. Aquest procés de registre sol·licitarà informació personal que, de forma segura, s'incorporarà al component servidor.

### 2.3.2 Guió per a la gestió de contactes

Amb l'objectiu de facilitar l'organització de les trobades, l'aplicació incorporarà una agenda interna de contactes on es podrà afegir, esborrar, editar o consultar entrades.

Una de les primeres accions que realitzarà l'usuari en començar a utilitzar l'aplicació serà la importació de les entrades existents en l'agenda del seu telèfon que es corresponguin a altres usuaris registrats del sistema. Si més no, també podrà enviar un SMS als seus contactes no registrats, informant-los de l'existència de l'aplicació i convidant-los a utilitzar-la.

Finalment, podrà organitzar en grups totes les entrades, de forma que en el moment de seleccionar els assistents a una trobada no calgui afegir tots els destinataris un a un. És a dir, un grup es comportarà com qualsevol altre contacte de l'agenda però amb un tractament especial en ser afegit com assistent: el grup com tal no apareixerà com a assistent de la reunió sinó que seran tots els seus membres els que passaran a formar-ne part.

### 2.3.3 Guió per a l'organització de trobades

Un usuari registrat podrà crear una reunió introduint una data i un conjunt de detalls complementaris, convertint-se d'aquesta manera en el seu únic organitzador (no es planteja la necessitat de compartir aquest rol entre diferents usuaris). Pel que fa al lloc de la trobada, existiran dos modes de definició:

- Mode "ubicació exacta", l'organitzador introdueix l'adreça exacta del lloc on cal acudir.
- Mode "segueix-me", la posició que s'agafarà com a referència per a "guiar" a tots els convidats serà l'última posició GPS publicada per l'organitzador.

Un cop definida la trobada, l'usuari podrà convocar a qualsevol dels seus contactes que tinguin instal·lada l'aplicació i automàticament cada convidat rebrà una notificació en el seu terminal que podrà ser acceptada o rebutjada.

Una conseqüència d'aquest funcionament és que l'usuari no podrà eliminar de la seva llista de contactes aquelles entrades que apareguin com assistents d'una futura trobada que ell ha organitzat.

### 2.3.4 Guió per a l'assistència a una trobada

Utilitzant la càmera del dispositiu l'aplicació mostrarà juntament a la imatge capturada la ubicació de la trobada, la distància fins a ella i unes mínimes indicacions de com arribar-hi. Per als assistents que no ho hagin fet, existirà la possibilitat d'obtenir-ne la seva ubicació GPS mostrant-la en un mapa 2D (Google® Maps).

És important tenir present que els assistents tindran l'opció de comunicar "manualment" al sistema la seva arribada a la destinació.

## 2.4 Casos d'ús

### 2.4.1 Actors

S'han identificat 4 actors: usuari, organitzador, assistent i el rellotge del propi sistema ja que aquest és l'encarregat d'enviar les notificacions i de sincronitzar les dades.

L'organitzador de la trobada també n'és un assistent i per tant es modelarà com una especialització d'aquest darrer actor, de forma anàloga a la que aquest especialitza a l'actor usuari (cal recordar que un assistent obligatòriament ha de tenir instal·lada l'aplicació al seu telèfon).

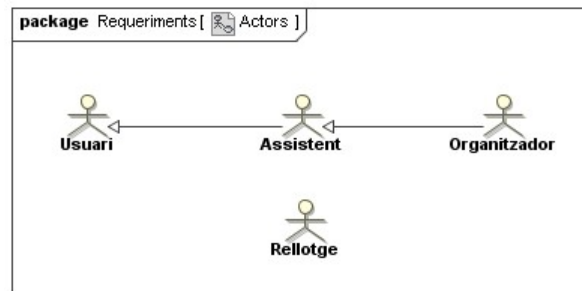


Figura 4: Diagrama d'Actors

### 2.4.2 Diagrama de casos d'ús

Amb la finalitat de facilitar-ne la representació i la comprensió s'han dividit tots els casos d'ús en diversos diagrames, mantenint les relacions existents entre ells.

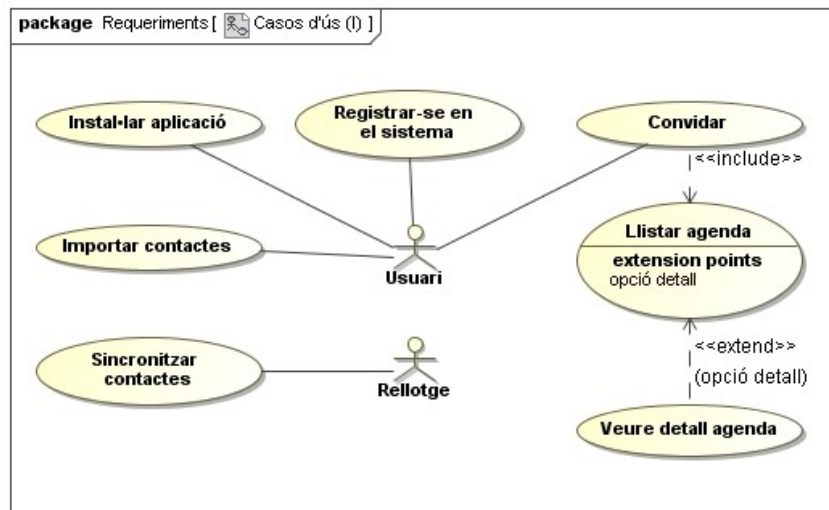


Figura 5: Diagrama de casos d'ús (I)

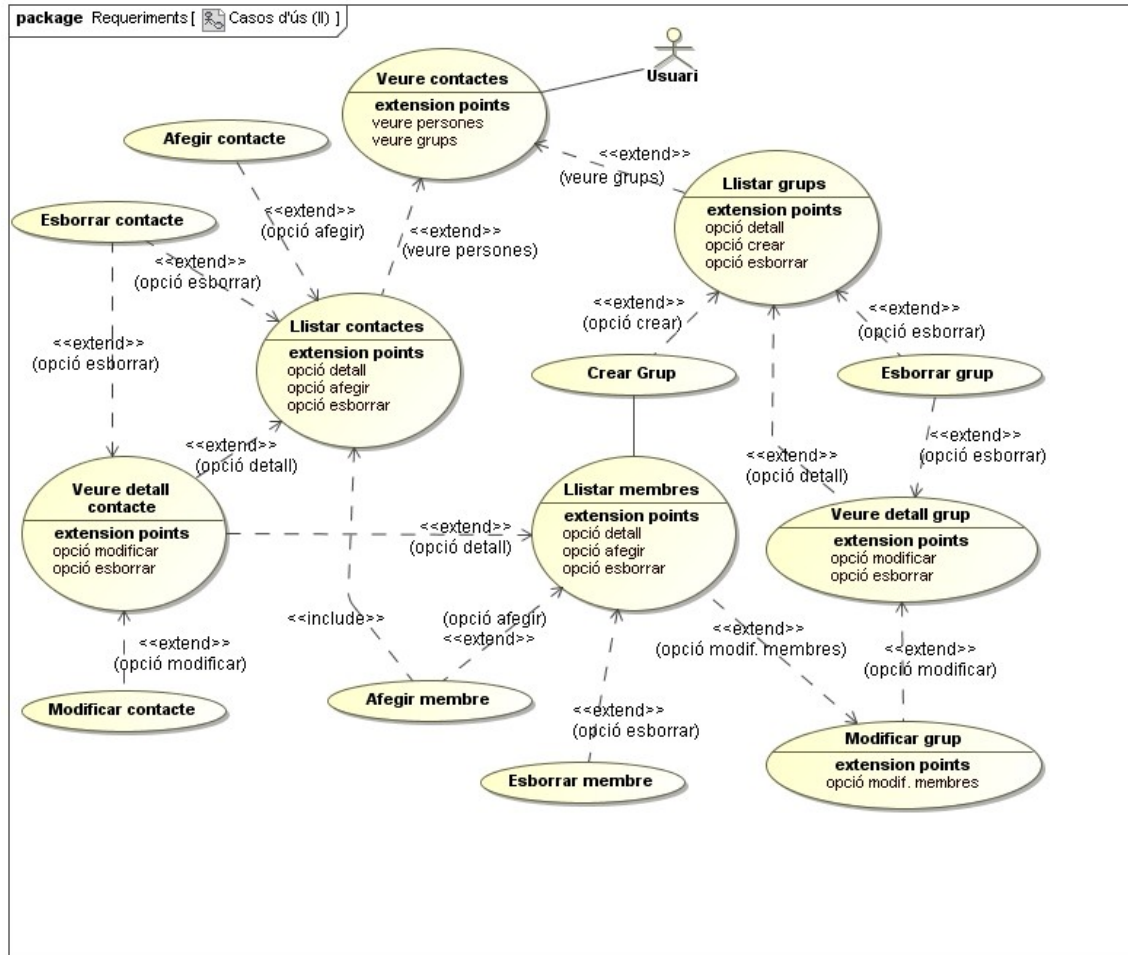


Figura 6: Diagrama de casos d'ús (II)



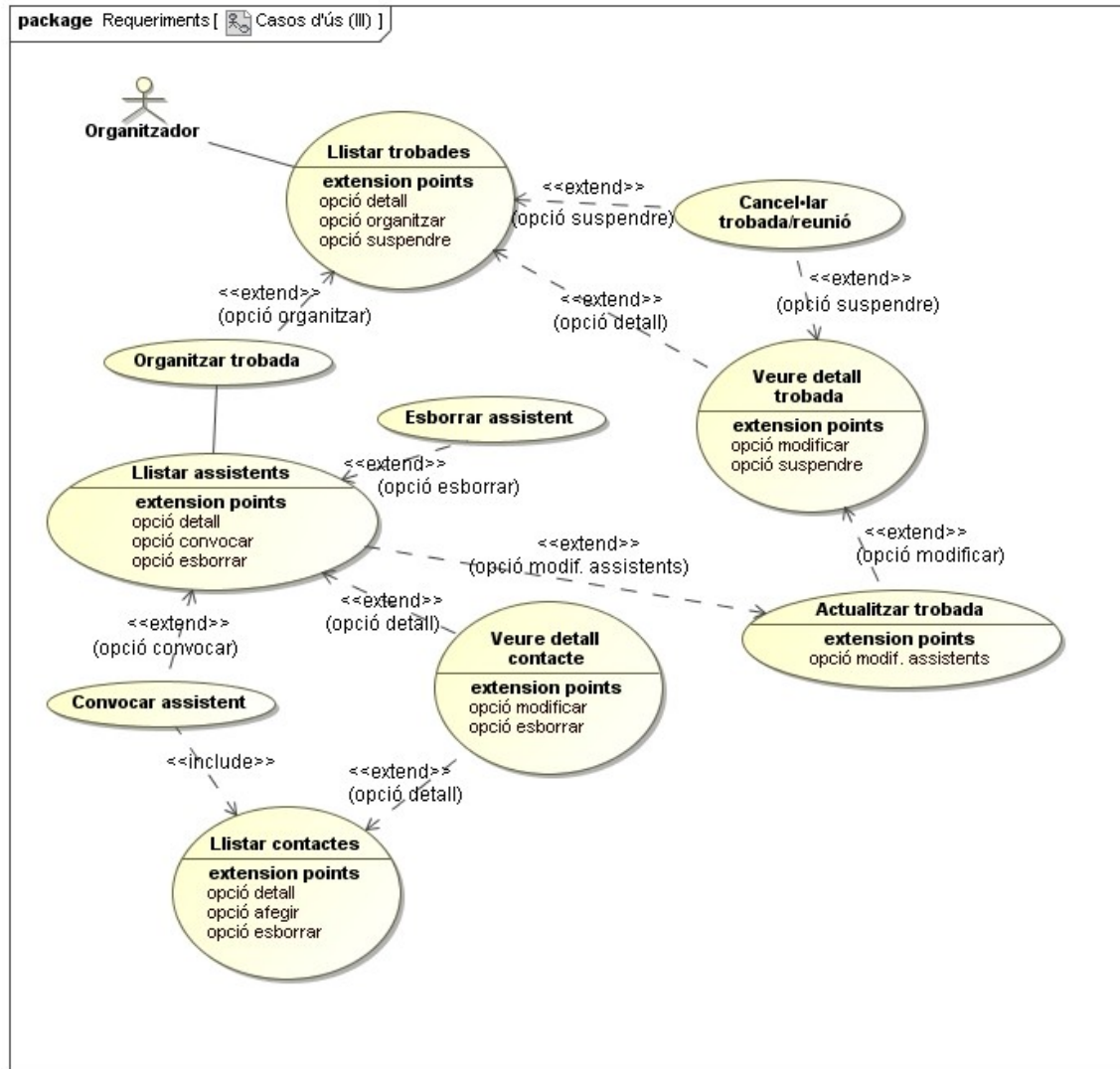


Figura 7: Diagrama de casos d'ús (III)

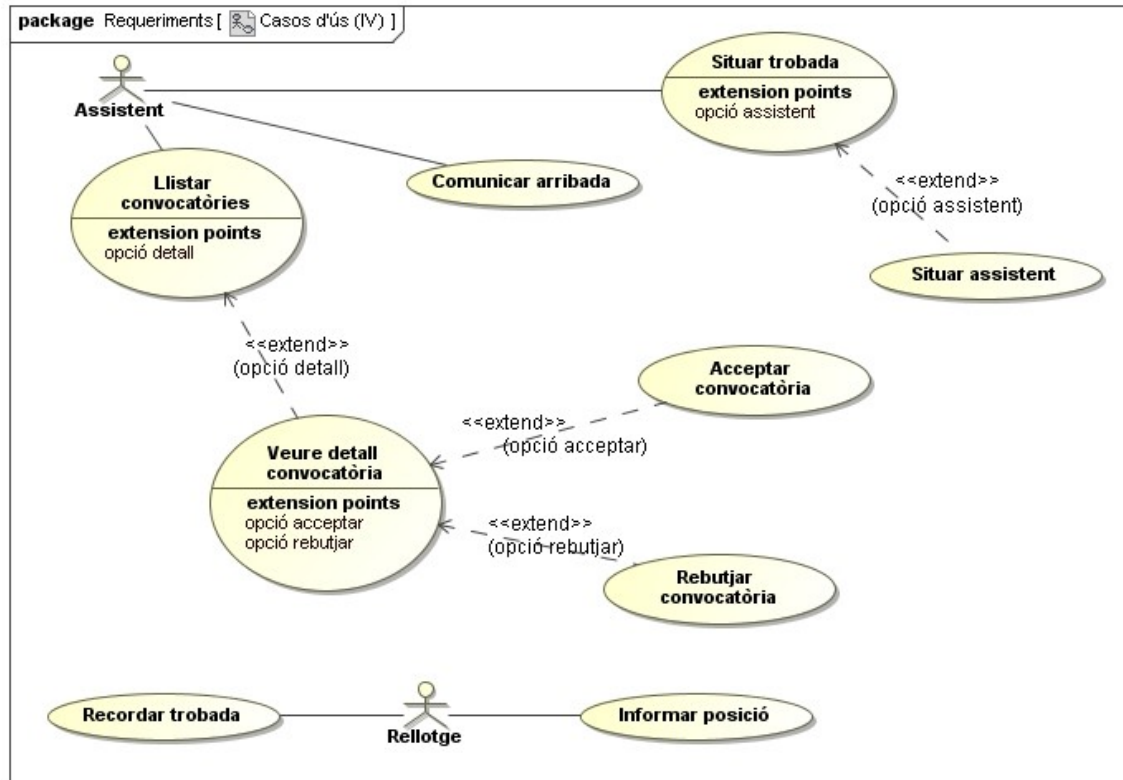


Figura 8: Diagrama de casos d'ús (IV)

### 2.4.3 Documentació textual

#### Cas d'ús número 1. "Instal·lar aplicació".

Resum de la funcionalitat: L'usuari instal·la l'aplicació al seu telèfon

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: L'usuari coneix l'adreça web d'on obtenir l'instal·lable de l'aplicació i disposa de connexió a Internet en el seu dispositiu

Postcondició: L'aplicació queda llesta per ser utilitzada

Procés normal principal:

1. L'usuari obre un navegador i es connecta a l'adreça web on hi ha accessible l'instal·lable de l'aplicació
2. El navegador descarrega el fitxer i executa de forma automàtica el procés d'instal·lació
3. L'aplicació demana els permisos necessaris per funcionar correctament
4. L'usuari atorga els permisos requerits en el pas 3

### **Cas d'ús número 2. "Registrar-se en el sistema".**

Resum de la funcionalitat: L'usuari instal·la l'aplicació al seu telèfon

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: L'aplicació està correctament instal·lada i es disposa de connexió a Internet en el telèfon

Postcondició: L'usuari queda registrat en el sistema

Procés normal principal:

1. L'aplicació demana el nom complet de l'usuari i un nom curt
2. L'usuari introdueix les dades indicades al pas 2
3. El sistema emmagatzema les dades

### **Cas d'ús número 3. "Importar contactes".**

Resum de la funcionalitat: S'incorporen com a contactes de l'usuari dins l'aplicació les entrades existents en l'agenda del telèfon que corresponen a altres usuaris registrats

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: L'agenda del telèfon conté alguna entrada

Postcondició: Les entrades de l'agenda del telèfon formen part de la llista de contactes de l'usuari

Procés normal principal:

1. L'usuari confirma l'operació

### **Cas d'ús número 4. "Convidar".**

Resum de la funcionalitat: L'usuari convida un conegut seu per a que s'instal·li l'aplicació mitjançant l'enviament d'un missatge curt (SMS). El text del missatge enviat serà fixat per l'aplicació i inclourà l'adreça web d'on obtenir l'instal·lable de l'aplicació

Actors: Usuari

Casos d'ús relacionats: Llistar agenda

Precondició: L'agenda del telèfon conté alguna entrada

Postcondició: L'aplicació envia un SMS al/s telèfon/s seleccionat/s

Procés normal principal:

1. include:: Llistar agenda
2. A partir del llistat obtingut en el punt 1, l'usuari selecciona el/s destinatari/s
3. L'aplicació envia un SMS als telèfons del/s contacte/s seleccionat/s en el pas 2

#### **Cas d'ús número 5. "Sincronitzar contactes".**

Resum de la funcionalitat: s'envia cap a la BBDD remota (component servidor) tota la informació sobre contactes que hi ha a la BBDD local del telèfon de l'usuari i que ha estat creada/modificada des de la data de l'última sincronització

Actors: Rellotge

Casos d'ús relacionats: Cap

Precondició: Es disposa de connexió a Internet en el dispositiu

Postcondició: La BBDD remota conté totes les entrades de contactes de l'usuari

Procés normal principal:

1. El rellotge dispara el procés de sincronització
2. L'aplicació envia totes les noves dades de contactes cap al component servidor

#### **Cas d'ús número 6. "Llistar agenda".**

Resum de la funcionalitat: L'usuari obté una llista de totes les entrades que hi ha a l'agenda del telèfon de l'usuari

Actors: Usuari

Casos d'ús relacionats: Veure detall agenda

Precondició: Cap

Postcondició: L'usuari obté la llista ordenada per cognom/nom de totes les entrades que hi ha a l'agenda o una llista buida si no n'hi ha cap

Procés normal principal:

1. L'aplicació recupera i mostra la llista de totes les entrades que hi ha a l'agenda del telèfon

Alternatives de procés i excepcions:

- 1a. L'usuari escull l'opció de veure el detall d'una entrada concreta
- 1a1 L'aplicació passa a executar el cas d'ús "Veure detall agenda"

#### **Cas d'ús número 7. "Veure detall agenda".**

Resum de la funcionalitat: recuperació de totes les dades d'una entrada de l'agenda del telèfon

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: L'entrada existeix

Postcondició: L'usuari obté totes les dades de l'entrada

Procés normal principal:

1. L'aplicació recupera i mostra totes les dades de l'entrada

#### **Cas d'ús número 8. "Veure contactes".**

Resum de la funcionalitat: L'usuari obté una llista de totes les persones o grups que pot convocar a una reunió/trobada

Actors: Usuari

Casos d'ús relacionats: Llistar contactes, Llistar grups

Precondició: Cap

Postcondició: L'usuari obté la llista de contactes del tipus seleccionat

Procés normal principal:

1. L'usuari escull quin tipus d'informació vol veure (persones o grups)

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de veure les persones (opció per defecte)

1a1 L'aplicació passa a executar el cas d'ús "Llistar contactes"

1b. L'usuari escull l'opció de veure els grups

1b1 L'aplicació passa a executar el cas d'ús "Llistar grups"

### **Cas d'ús número 9. "Llistar contactes".**

Resum de la funcionalitat: L'usuari obté una llista de totes les persones que pot convocar a una reunió/trobada

Actors: Usuari

Casos d'ús relacionats: Afegir contacte, Veure detall contacte, Esborrar contacte

Precondició: Cap

Postcondició: L'usuari obté la llista ordenada per cognom/nom de tots els seus contactes que són persones

Procés normal principal:

1. L'aplicació recupera i mostra la llista de tots els contactes de l'usuari que són persones

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de veure el detall d'un contacte determinat

1a1 L'aplicació passa a executar el cas d'ús "Veure detall contacte"

1b. L'usuari escull l'opció d'afegir un nou contacte

1b1 L'aplicació passa a executar el cas d'ús "Afegir contacte"

1c. L'usuari escull l'opció d'esborrar un contacte determinat

1c1 L'aplicació passa a executar el cas d'ús "Esborrar contacte"

### **Cas d'ús número 10. "Afegir contacte".**

Resum de la funcionalitat: dona d'alta un nou contacte de l'usuari. Només serà possible si no existeix a la llista de contactes de l'usuari una entrada amb el mateix número de telèfon que el que s'intenta afegir

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: Cap

Postcondició: S'afegeix una nova entrada a la llista de contactes de l'usuari

Procés normal principal:

1. L'aplicació demana el nom complet, un nom curt i el número de telèfon del nou contacte
2. L'usuari introdueix les dades indicades al pas 1
3. L'aplicació verifica les condicions per procedir a la creació
4. L'aplicació enregistra les dades del nou contacte

Alternatives de procés i excepcions:

- 3a. El número de telèfon introduït ja està associat a un contacte existent
- 3a1 L'aplicació informa del problema a l'usuari
- 3a2 Es torna al pas 1

**Cas d'ús número 11. "Esborrar contacte".**

Resum de la funcionalitat: eliminació de totes les dades que l'aplicació emmagatzema d'un contacte. Només serà possible si no existeix cap trobada organitzada per l'usuari que tingui com assistent el contacte que s'intenta esborrar.

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: El contacte existeix

Postcondició: Totes les dades del contacte desapareixen (incloent la seva presència com a membre de qualsevol grup)

Procés normal principal:

1. L'aplicació demana confirmació
2. L'usuari confirma l'esborrament
3. El sistema verifica les condicions per procedir a l'esborrament
4. Les dades del contacte desapareixen de l'aplicació

Alternatives de procés i excepcions:

- 3a. El contacte apareix com assistent d'alguna futura trobada organitzada per l'usuari
- 3a1 L'aplicació informa del problema a l'usuari
- 3a2 L'execució finalitza

**Cas d'ús número 12. "Veure detall contacte".**

Resum de la funcionalitat: recuperació de totes les dades que l'aplicació té d'un contacte

Actors: Usuari

Casos d'ús relacionats: Modificar contacte, Esborrar contacte

Precondició: El contacte existeix

Postcondició: S'obtenen i mostren totes les dades d'un contacte

Procés normal principal:

1. L'aplicació recupera totes les dades del contacte

Alternatives de procés i excepcions:

- 1a. L'usuari escull l'opció de modificar el contacte seleccionat

1a1 L'aplicació passa a executar el cas d'ús "Modificar contacte"

1b. L'usuari escull l'opció d'esborrar el contacte seleccionat

1b1 L'aplicació passa a executar el cas d'ús "Esborrar contacte"

### **Cas d'ús número 13. "Modificar contacte".**

Resum de la funcionalitat: Permet modificar la informació d'un contacte

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: El contacte existeix

Postcondició: S'actualitzen les dades del contacte

Procés normal principal:

1. L'aplicació recupera totes les dades (nom complet, nom curt i número de telèfon) del contacte

2. L'usuari introdueix les noves dades

3. L'aplicació verifica les condicions per procedir a la modificació

4. L'aplicació emmagatzema les noves dades del contacte

Alternatives de procés i excepcions:

3a. El número de telèfon introduït ja està associat a un altre contacte

3a1 L'aplicació informa del problema a l'usuari

3a2 Es torna al pas 2

### **Cas d'ús número 14. "Llistar grups".**

Resum de la funcionalitat: L'usuari obté una llista de tots els seus grups de contactes

Actors: Usuari

Casos d'ús relacionats: Crear grup, Veure detall grup, Esborrar grup

Precondició: Cap

Postcondició: L'usuari obté la llista ordenada per nom de tots els seus grups de contactes

Procés normal principal:

1. L'aplicació recupera i mostra la llista de tots els grups de contactes de l'usuari

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de veure el detall d'un grup determinat

1a1 L'aplicació passa a executar el cas d'ús "Veure detall grup"

1b. L'usuari escull l'opció d'afegir un nou grup

1b1 L'aplicació passa a executar el cas d'ús "Crear grup"

1c. L'usuari escull l'opció d'esborrar un grup determinat

1c1 L'aplicació passa a executar el cas d'ús "Esborrar grup"

### **Cas d'ús número 15. "Crear grup".**

Resum de la funcionalitat: Permet crear un grup (buit) de contactes de forma que es faciliti l'enviament de convocatòries. Només serà possible si no existeix a la llista de contactes de l'usuari cap entrada amb el mateix nom que el que s'intenta afegir

Actors: Usuari

Casos d'ús relacionats: Llistar membres

Precondició: Cap

Postcondició: S'afegeix una nova entrada a la llista de contactes de l'usuari

Procés normal principal:

1. L'aplicació demana el nom del nou grup i una descripció
2. L'usuari introdueix les dades indicades al pas 1
3. L'aplicació verifica les condicions per procedir a la creació
4. L'aplicació enregistra les dades del nou contacte
5. L'aplicació passa a executar el cas d'ús "Llistar membres"

Alternatives de procés i excepcions:

- 3a. El nom del grup introduït ja existeix
  - 3a1 L'aplicació informa del problema a l'usuari
  - 3a2 Es torna al pas 1

### **Cas d'ús número 16. "Llistar membres".**

Resum de la funcionalitat: L'usuari obté una llista de tots els membres d'un grup

Actors: Usuari

Casos d'ús relacionats: Afegir membre, Veure detall contacte, Esborrar membre

Precondició: El grup existeix

Postcondició: L'usuari obté la llista ordenada per cognom/nom de tots els membres del grup

Procés normal principal:

1. L'aplicació recupera i mostra la llista de tots els membres del grup

Alternatives de procés i excepcions:

- 1a. L'usuari escull l'opció de veure el detall d'un membre determinat
  - 1a1 L'aplicació passa a executar el cas d'ús "Veure detall contacte"
- 1b. L'usuari escull l'opció d'afegir un nou membre
  - 1b1 L'aplicació passa a executar el cas d'ús "Afegir membre"
- 1c. L'usuari escull l'opció d'esborrar un membre determinat
  - 1c1 L'aplicació passa a executar el cas d'ús "Esborrar membre"

### **Cas d'ús número 17. "Afegir membre".**

Resum de la funcionalitat: Permet afegir contactes a un grup. Un grup no pot contenir altres grups, només persones. En el cas d'afegir un contacte que ja consta com a membre s'ignora la petició



Actors: Usuari

Casos d'ús relacionats: Llistar contactes

Precondició: El grup existeix

Postcondició: S'actualitzen els membres del grup

Procés normal principal:

1. include::Llistar contactes
2. L'usuari selecciona un/s contacte/s del llistat obtingut al pas 1
3. L'aplicació afegeix com a membre/s del grup el/s contacte/s seleccionat/s al pas 2

#### **Cas d'ús número 18. "Esborrar membre".**

Resum de la funcionalitat: Permet eliminar un contacte d'un grup

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: El grup existeix i el contacte n'és un membre

Postcondició: El contacte ja no forma part del grup

Procés normal principal:

1. L'aplicació demana confirmació
2. L'usuari confirma l'esborrament
3. L'aplicació elimina el contacte del grup

#### **Cas d'ús número 19. "Esborrar grup".**

Resum de la funcionalitat: eliminació de totes les dades que l'aplicació emmagatzema d'un grup de contactes

Actors: Usuari

Casos d'ús relacionats: Cap

Precondició: El grup existeix

Postcondició: Totes les dades del grup desapareixen

Procés normal principal:

1. L'aplicació demana confirmació
2. L'usuari confirma l'esborrament
3. Les dades del grup de contactes desapareixen de l'aplicació

#### **Cas d'ús número 20. "Veure detall grup".**

Resum de la funcionalitat: recuperació de totes les dades que l'aplicació té d'un grup de contactes

Actors: Usuari

Casos d'ús relacionats: Modificar grup, Esborrar grup

Precondició: El grup existeix

Postcondició: S'obtenen i mostren totes les dades d'un grup de contactes

Procés normal principal:

1. L'aplicació recupera totes les dades del grup

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de modificar el grup seleccionat

1a1 L'aplicació passa a executar el cas d'ús "Modificar grup"

1b. L'usuari escull l'opció d'esborrar el grup seleccionat

1b1 L'aplicació passa a executar el cas d'ús "Esborrar grup"

### **Cas d'ús número 21. "Modificar grup".**

Resum de la funcionalitat: Permet modificar la informació d'un grup de contactes

Actors: Usuari

Casos d'ús relacionats: Llistar membres

Precondició: El grup de contactes existeix

Postcondició: S'actualitzen les dades del grup

Procés normal principal:

1. L'aplicació recupera totes les dades (nom, descripció) del grup

2. L'usuari introdueix les noves dades

3. L'aplicació emmagatzema les noves dades del grup

Alternatives de procés i excepcions:

2a. L'usuari escull l'opció de modificar els membres

2a1 L'aplicació passa a executar el cas d'ús "Llistar membres"

### **Cas d'ús número 22. "Llistar trobades/reunions".**

Resum de la funcionalitat: L'usuari obté una llista de totes les trobades reunions on apareix com a organitzador

Actors: Organitzador

Casos d'ús relacionats: Organitzar trobada/reunió, Veure detall trobada/reunió, Cancel·lar trobada/reunió

Precondició: Cap

Postcondició: L'usuari obté la llista ordenada de forma descendent per data de creació, de totes les seves trobades

Procés normal principal:

1. L'aplicació recupera i mostra la llista de totes les trobades organitzades per l'usuari

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de veure el detall d'un trobada/reunió determinada

1a1 L'aplicació passa a executar el cas d'ús "Veure detall trobada/reunió"

1b. L'usuari escull l'opció d'organitzar una nova trobada/reunió

1b1 L'aplicació passa a executar el cas d'ús "Organitzar trobada/reunió"

1c. L'usuari escull l'opció de suspendre una trobada/reunió

1c1 L'aplicació passa a executar el cas d'ús "Cancel·lar trobada/reunió"

### **Cas d'ús número 23. "Organitzar trobada/reunió".**

Resum de la funcionalitat: Permet crear una nova trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Llistar assistents

Precondició: Cap

Postcondició: S'afegeix una nova entrada a la llista de trobades/reunions de l'usuari

Procés normal principal:

1. L'aplicació demana el nom, el motiu, la data i hora, la descripció de la ubicació de la trobada/reunió. Opcionalment es podrà introduir una durada i uns comentaris
2. L'usuari introdueix les dades indicades al pas 1
3. L'aplicació demana el mode de localització
4. L'usuari introdueix les dades indicades al pas 3
5. L'aplicació enregistra les dades de la nova trobada/reunió
6. L'aplicació passa a executar el cas d'ús "Llistar assistents"

Alternatives de procés i excepcions:

- 4a. L'usuari introdueix les coordenades GPS de la ubicació
- 4b. L'usuari defineix la ubicació com "segueix-me"

### **Cas d'ús número 24. "Llistar assistents".**

Resum de la funcionalitat: L'usuari obté una llista de tots els assistents a una trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Convocar assistent, Veure detall contacte, Esborrar assistent

Precondició: La trobada/reunió existeix

Postcondició: L'usuari obté la llista ordenada per cognom/nom de tots els assistents a la trobada/reunió

Procés normal principal:

1. L'aplicació recupera i mostra la llista de tots els assistents a la trobada/reunió

Alternatives de procés i excepcions:

- 1a. L'usuari escull l'opció de veure el detall d'un assistent
  - 1a1 L'aplicació passa a executar el cas d'ús "Veure detall contacte"
- 1b. L'usuari escull l'opció d'afegir un nou assistent
  - 1b1 L'aplicació passa a executar el cas d'ús "Convocar assistent"
- 1c. L'usuari escull l'opció d'esborrar un assistent
  - 1c1 L'aplicació passa a executar el cas d'ús "Esborrar assistent"

**Cas d'ús número 25. "Convocar assistent".**

Resum de la funcionalitat: Permet afegir assistents a una trobada/reunió. En el cas d'afegir un contacte que ja consta com a assistent s'ignora la petició

Actors: Organitzador

Casos d'ús relacionats: Llistar contactes

Precondició: La trobada/reunió existeix

Postcondició: S'actualitza la llista d'assistents de la trobada/reunió

Procés normal principal:

1. include::Llistar contactes
2. L'usuari selecciona un/s contacte/s del llistat obtingut al pas 1
3. L'aplicació afegeix com a assistent/s de la trobada/reunió el/s contacte/s seleccionat/s al pas 2
4. L'aplicació notifica al/s nou/s assistent/s de la trobada/reunió

**Cas d'ús número 26. "Esborrar assistent".**

Resum de la funcionalitat: Permet eliminar un assistent a una trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Cap

Precondició: La trobada/reunió existeix i el contacte hi consta com a assistent

Postcondició: L'assistent és eliminat de la trobada/reunió

Procés normal principal:

1. L'aplicació demana confirmació
2. L'usuari confirma l'esborrament
3. L'aplicació elimina l'assistent
4. L'aplicació notifica el canvi a l'assistent eliminat

**Cas d'ús número 27. "Cancel·lar trobada/reunió".**

Resum de la funcionalitat: Permet eliminar totes les dades que l'aplicació emmagatzema d'una trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Cap

Precondició: La trobada/reunió existeix i ha estat organitzada per l'usuari

Postcondició: Totes les dades de la trobada/reunió desapareixen

Procés normal principal:

1. L'aplicació demana confirmació
2. L'usuari confirma l'esborrament
3. Les dades de la trobada/reunió desapareixen de l'aplicació
4. L'aplicació notifica de la cancel·lació a tots els contactes que constaven com assistents

**Cas d'ús número 28. "Veure detall trobada/reunió".**

Resum de la funcionalitat: recuperació de totes les dades que l'aplicació té d'una trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Actualitzar trobada/reunió, Cancel·lar trobada/reunió

Precondició: La trobada/reunió existeix

Postcondició: S'obtenen i mostren totes les dades d'una trobada/reunió

Procés normal principal:

1. L'aplicació recupera totes les dades de la trobada/reunió

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de modificar la informació de la trobada/reunió

1a1 L'aplicació passa a executar el cas d'ús "Actualitzar trobada/reunió"

1b. L'usuari escull l'opció de cancel·lar la trobada/reunió

1b1 L'aplicació passa a executar el cas d'ús "Cancel·lar trobada/reunió"

**Cas d'ús número 29. "Actualitzar trobada/reunió".**

Resum de la funcionalitat: Permet modificar la informació d'una trobada/reunió

Actors: Organitzador

Casos d'ús relacionats: Llistar assistents

Precondició: La trobada/reunió existeix

Postcondició: S'actualitzen les dades de la trobada/reunió

Procés normal principal:

1. L'aplicació recupera totes les dades de la trobada/reunió seleccionada

2. L'usuari introdueix les noves dades

3. L'aplicació emmagatzema les noves dades de la trobada

4. L'aplicació envia una notificació informant del canvi a tots els assistents

Alternatives de procés i excepcions:

2a. L'usuari escull l'opció de modificar els assistents

2a1 L'aplicació passa a executar el cas d'ús "Llistar assistents"

**Cas d'ús número 30. "Llistar convocatòries".**

Resum de la funcionalitat: L'usuari obté una llista de totes les convocatòries a trobades a les que encara no ha respòs

Actors: Assistent

Casos d'ús relacionats: Veure detall convocatòria

Precondició: Cap

Postcondició: L'usuari obté la llista ordenada de forma descendent per data de creació, de totes les seves convocatòries

Procés normal principal:

1. L'aplicació recupera i mostra la llista de totes les convocatòries rebudes per l'usuari

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció de veure el detall d'un trobada/reunió determinada

1a1 L'aplicació passa a executar el cas d'ús "Veure detall convocatòria"

**Cas d'ús número 31. "Veure detall convocatòria".**

Resum de la funcionalitat: recuperació de totes les dades que l'aplicació té d'una trobada/reunió

Actors: Assistent

Casos d'ús relacionats: Acceptar convocatòria, Rebutjar convocatòria

Precondició: La trobada/reunió existeix

Postcondició: S'obtenen i mostren totes les dades d'una trobada/reunió

Procés normal principal:

1. L'aplicació recupera totes les dades de la trobada/reunió

Alternatives de procés i excepcions:

1a. L'usuari escull l'opció d'acceptar la convocatòria

1a1 L'aplicació passa a executar el cas d'ús "Acceptar convocatòria"

1b. L'usuari escull l'opció de rebutjar la convocatòria

1b1 L'aplicació passa a executar el cas d'ús "Rebutjar convocatòria"

**Cas d'ús número 32. "Acceptar convocatòria".**

Resum de la funcionalitat: Permet que un assistent a una trobada confirmi la seva presència

Actors: Assistent

Casos d'ús relacionats: Cap

Precondició: La trobada/reunió existeix i l'organitzador de la trobada/reunió ha proposat a l'usuari com a assistent

Postcondició: L'usuari passa a formar part dels assistents confirmats d'una trobada/reunió

Procés normal principal:

1. L'aplicació demana confirmació

2. L'usuari confirma l'assistència

3. L'aplicació emmagatzema el canvi

4. L'aplicació notifica l'organitzador

**Cas d'ús número 33. "Rebutjar convocatòria".**

Resum de la funcionalitat: Permet que un assistent a una trobada descarti la seva presència

Actors: Assistent

Casos d'ús relacionats: Cap

Precondició: La trobada/reunió existeix i l'organitzador de la trobada/reunió ha proposat a l'usuari com a assistent

Postcondició: L'usuari passa a formar part dels assistents descartats d'una trobada/reunió

Procés normal principal:

1. L'aplicació demana el motiu pel qual el contacte no assistirà a la trobada/reunió
2. L'usuari introdueix la informació requerida en el punt 1
3. L'aplicació emmagatzema les dades
4. L'aplicació notifica l'organitzador

#### **Cas d'ús número 34. "Recordar trobada/reunió".**

Resum de la funcionalitat: L'usuari rep una notificació informant que s'acosta la data d'una trobada/reunió on ell apareix com organitzador o assistent

Actors: Rellotge

Casos d'ús relacionats: Cap

Precondició: L'usuari consta com organitzador o assistent d'una trobada/reunió planificada per a dintre de M minuts

Postcondició: L'aplicació envia la notificació d'avís

Procés normal principal:

1. El rellotge dispara el procés d'avís
2. L'aplicació envia la notificació a l'usuari

#### **Cas d'ús número 35. "Informar posició".**

Resum de la funcionalitat: L'aplicació registra de forma periòdica la posició GPS de l'usuari

Actors: Rellotge

Casos d'ús relacionats: Cap

Precondició: A partir del telèfon es pot obtenir la posició GPS de l'usuari

Postcondició: El sistema emmagatzema la posició GPS actual de l'usuari

Procés normal principal:

1. El rellotge dispara el procés de publicació de la posició
2. L'aplicació obté del telèfon la posició GPS de l'usuari
3. El sistema emmagatzema les dades obtingudes en el punt 2

#### **Cas d'ús número 36. "Situar trobada/reunió".**

Resum de la funcionalitat: L'aplicació mostra, conjuntament a la imatge capturada per la càmera del telèfon, informació de la trobada, de com arribar a la seva ubicació.

Actors: Assistent

Casos d'ús relacionats: Cap

Precondició: A partir del telèfon es pot obtenir la posició GPS de l'usuari i també pot utilitzar-se la càmera del dispositiu

Postcondició: L'usuari obté la informació indicada anteriorment fent servir tècniques de realitat augmentada

Procés normal principal:

1. L'aplicació obté del telèfon la posició GPS de l'usuari
2. L'aplicació obté la posició GPS de la ubicació de la trobada/reunió segons el que ha definit l'organitzador
3. L'aplicació calcula quin és el millor camí per arribar a la ubicació de la trobada/reunió
4. L'aplicació obté totes les dades de la trobada/reunió
5. L'aplicació mostra, conjuntament a la imatge capturada per la càmera del telèfon, un resum de les dades obtingudes en els punts 3 i 4

Alternatives de procés i excepcions:

- 2a. L'organitzador va introduir les dades en el moment de la creació de la trobada/reunió
- 2a1 Es consulten les coordenades GPS emmagatzemades en l'aplicació
- 2b. L'organitzador va definir la ubicació del tipus "segueix-me"
- 2b1 La posició GPS de la trobada/reunió serà la de l'organitzador, la qual s'obté a partir del que hi ha emmagatzemat en el sistema

### **Cas d'ús número 37. "Situar assistent".**

Resum de la funcionalitat: L'aplicació indica en un mapa 2D, a escala adequada, la posició de l'assistent a una trobada/reunió

Actors: Assistent

Casos d'ús relacionats: Cap

Precondició: L'assistent d'una trobada/reunió encara no ha arribat al lloc designat per l'organitzador

Postcondició: L'usuari visualitza en un mapa 2D la posició de l'assistent

Procés normal principal:

1. L'aplicació obté la posició GPS de l'assistent emmagatzemada en el sistema
2. L'aplicació dibuixa el mapa 2D i hi situa la posició obtinguda en el punt 2

### **Cas d'ús número 38. "Comunicar arribada".**

Resum de la funcionalitat: Un assistent deixa constància en el sistema de la seva arribada al lloc de la trobada

Actors: Assistent

Casos d'ús relacionats: Cap

Precondició: Cap

Postcondició: L'estat del sistema reflecteix que l'usuari ja ha arribat al lloc de la trobada

Procés normal principal:

1. L'usuari envia una notificació per informar de la seva arribada
2. El sistema emmagatzema les dades obtingudes en el punt 1



Donat l'elevat nombre de casos d'ús resultants de l'anàlisi, la implementació dels casos d'ús es farà de forma iterativa començant per aquells que es consideren crítics i acabant pels menys rellevants (aquests últims seran considerats com opcionals):

<p><b>Crítics (obligatoris)</b></p>	<p>Cas d'ús número 1. "Instal·lar aplicació".                  Cas d'ús número 2. "Registrar-se en el sistema".                  Cas d'ús número 3. "Importar contactes".                  Cas d'ús número 8. "Veure contactes".                  Cas d'ús número 9. "Llistar contactes".                  Cas d'ús número 10. "Afegir contacte".                  Cas d'ús número 11. "Esborrar contacte".                  Cas d'ús número 12. "Veure detall contacte".                  Cas d'ús número 13. "Modificar contacte".                  Cas d'ús número 22. "Llistar trobades/reunions".                  Cas d'ús número 23. "Organitzar trobada/reunió".                  Cas d'ús número 24. "Llistar assistents".                  Cas d'ús número 25. "Convocar assistent".                  Cas d'ús número 26. "Esborrar assistent".                  Cas d'ús número 27. "Cancel·lar trobada/reunió".                  Cas d'ús número 28. "Veure detall trobada/reunió".                  Cas d'ús número 29. "Actualitzar trobada/reunió".                  Cas d'ús número 30. "Llistar convocatòries".                  Cas d'ús número 31. "Veure detall convocatòria".                  Cas d'ús número 32. "Acceptar convocatòria".                  Cas d'ús número 33. "Rebutjar convocatòria".</p>
<p><b>Importants (recomanables)</b></p>	<p>Cas d'ús número 35. "Informar posició".                  Cas d'ús número 36. "Situar trobada/reunió".                  Cas d'ús número 37. "Situar assistent".                  Cas d'ús número 38. "Comunicar arribada".</p>
<p><b>Poc rellevants (opcionals)</b></p>	<p>Cas d'ús número 5. "Sincronitzar contactes".                  Cas d'ús número 6. "Llistar agenda".                  Cas d'ús número 7. "Veure detall agenda".                  Cas d'ús número 4. "Convidar".                  Cas d'ús número 14. "Llistar grups".                  Cas d'ús número 15. "Crear grup".                  Cas d'ús número 16. "Llistar membres".</p>

	Cas d'ús número 17. "Afegir membre". Cas d'ús número 18. "Esborrar membre". Cas d'ús número 19. "Esborrar grup". Cas d'ús número 20. "Veure detall grup". Cas d'ús número 21. "Modificar grup". Cas d'ús número 34. "Recordar trobada/reunió".
--	---

## 2.5 Requisits no funcionals

Tot i que cada cop els dispositius mòbils són més potents en termes de capacitat de procés i emmagatzematge, no s'ha d'oblidar que es tracta d'aparells amb recursos limitats.

El sistema desenvolupat ha de preveure el ràpid creixement del volum d'usuaris adaptant dinàmicament les seves capacitats tecnològiques per continuar operant sense cap mena d'impacte en el seu rendiment.

S'ha d'extremar la portabilitat de la solució, així com l'ús de components i tecnologies estàndard.

## 2.6 Requisits de la UI

La interfície ha de resultar gràficament atractiva i al mateix temps molt còmoda d'utilitzar: la navegació per les diferents pantalles ha de ser el més intuïtiva possible.

Totes les llistes funcionaran de la mateixa manera:

- Permetran el desplaçament (*scroll*) vertical
- La primera entrada correspondrà a un camp de text buit que permetrà filtrar els elements de la llista aplicant l'operació "conté" al nom de l'element
- Des de la mateixa llista s'ha de permetre afegir un nou element
- Selecció d'una entrada s'anirà al detall de l'element

Tots els detalls funcionaran de la mateixa manera:

- Es mostraran les dades de l'element en qüestió
- Si l'usuari és el propietari/creador de l'element podrà modificar/esborrar l'element
- En el cas de les trobades els assistents podran acceptar/rebutjar la convocatòria

## 3 Anàlisi de Requisits

### 3.1 Revisió de casos d'ús

La base de partida per a l'anàlisi és la documentació sobre els casos d'ús elaborada en l'etapa anterior. El detall actual ja es considera suficient i per tant ja es poden utilitzar com a base per als passos posteriors.

### 3.2 Paquets d'anàlisi i serveis

Tots els casos d'ús descrits poden agrupar-se en els següents paquets d'anàlisi:

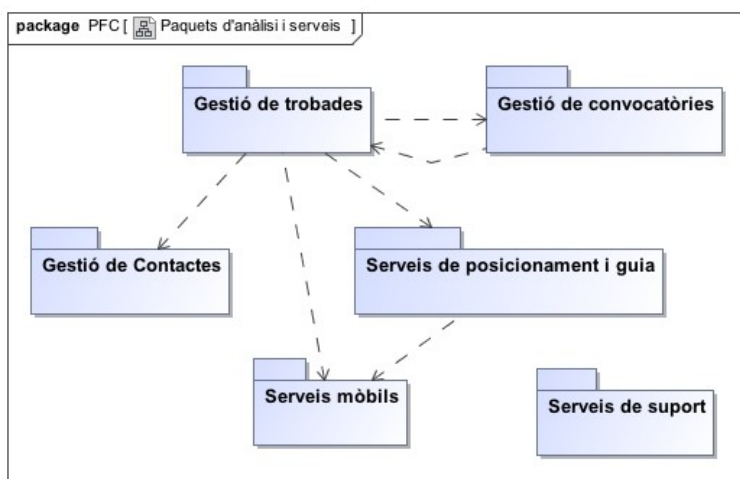


Figura 9: Diagrama de paquets d'anàlisi

Paquet “**Serveis de suport**”. Inclou els casos d'ús: Instal·lar aplicació, Registrar-se en el sistema

Paquet “**Serveis mòbils**”. Inclou els casos d'ús: Informar posició, Comunicar arribada

Paquet “**Serveis de posicionament i guia**”. Inclou els casos d'ús: Situar trobada/reunió, Situar assistent

Paquet “**Gestió de contactes**”. Inclou els casos d'ús: Veure contactes, Llistar contactes, Afegir contacte, Esborrar contacte, Veure detall contacte, Modificar contacte, Importar contactes, Sincronitzar contactes, Llistar agenda, Veure detall agenda, Convidar, Llistar grups, Crear grup, Llistar membres, Afegir membre, Esborrar membre, Esborrar grup, Veure detall grup, Modificar grup, Recordar trobada/reunió

Paquet “**Gestió de convocatòries**”. Inclou els casos d'ús: Llistar convocatòries, Veure detall convocatòria, Acceptar convocatòria, Rebutjar convocatòria

Paquet “**Gestió de trobades**”. Inclou els casos d'ús: Llistar trobades/reunions, Organitzar trobada/reunió, Llistar assistents, Convocar assistent, Esborrar assistent, Cancel·lar trobada/reunió, Veure detall trobada/reunió, Actualitzar trobada/reunió

Degut a la seva simplicitat, no s'ha considerat necessari afegir cap paquet de servei dins dels paquets d'anàlisi.

### 3.3 Classes d'entitat

De la lectura de tots els casos d'ús s'han identificat les següents classes d'entitat i les relacions entre elles:

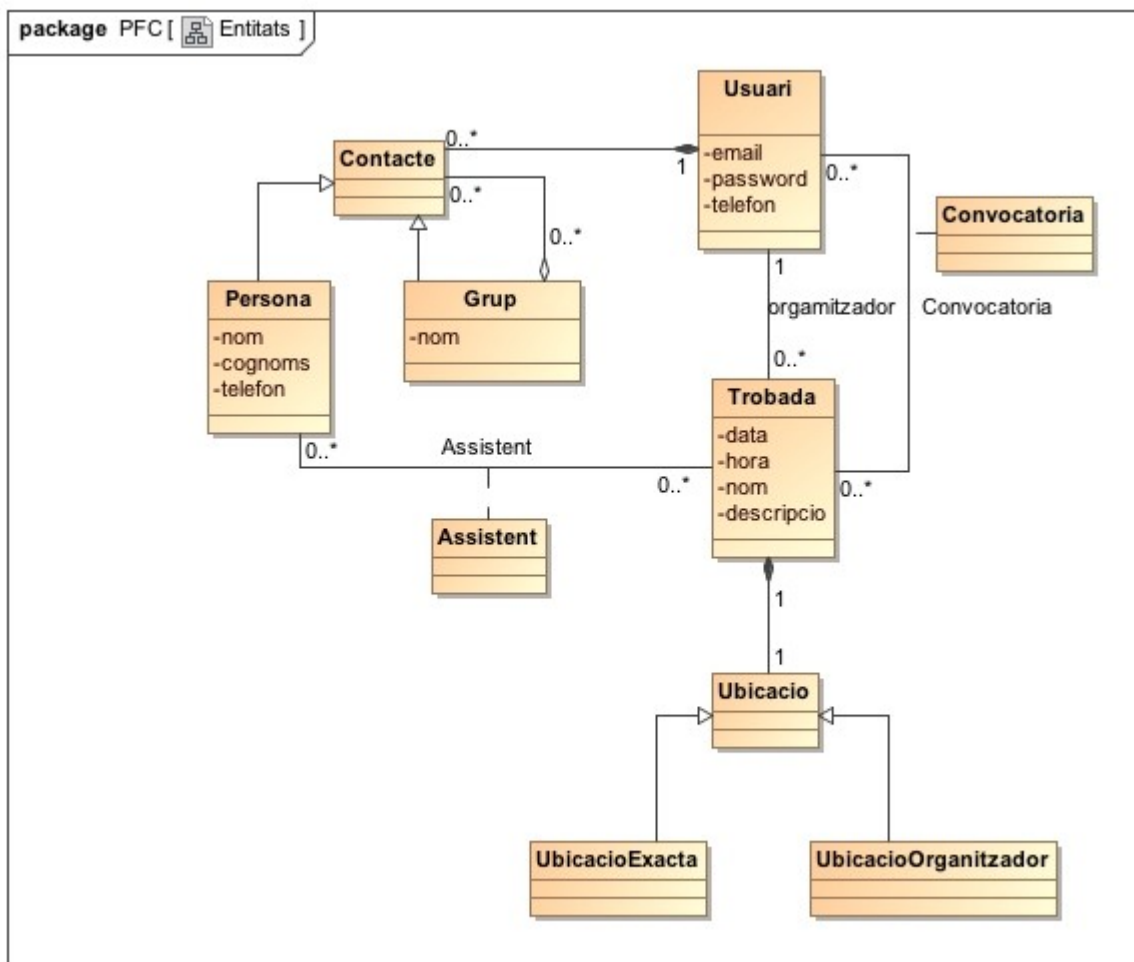


Figura 10: Diagrama de classes d'entitat

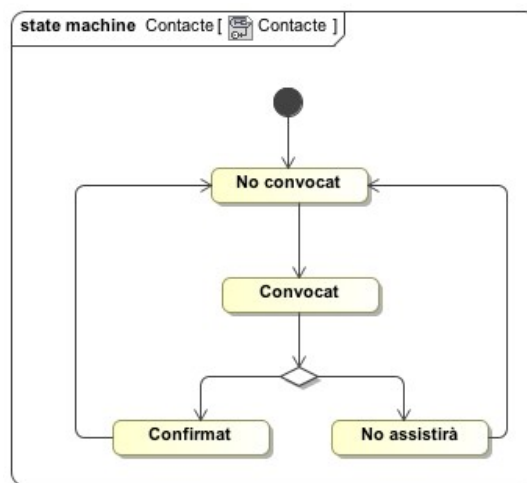
I el detall de cada entitat és:

Usuari	Classe que representa un usuari registrat del sistema.
Contacte	Classe abstracta que modela una entrada en l'agenda de l'aplicació. Es tracta d'una classe abstracta.
Persona	Tipus de contacte corresponent a una persona física.
Grup	Agrupació de contactes amb l'objectiu de facilitar la selecció dels assistents a una trobada.
Trobada	Entitat que representa una trobada o reunió.
Assistent	Classe que modela la relació M:N existent entre una trobada i els

	contactes de l'organitzador que han estat convocats.
Ubicació	Representa el punt de trobada d'una reunió. Es tracta d'una classe abstracta.
UbicacióExacta	Tipus d'ubicació corresponent a una adreça exacta.
UbicacióOrganitzador	Tipus d'ubicació segons la qual el punt de trobada s'estableix a partir de la posició GPS de l'organitzador.
Convocatòria	Classe que modela la relació M:N existent entre una trobada i els usuaris que han estat convocats.

### 3.4 Diagrama d'estats

Els estats associats a contacte en relació a una convocatòria són:



**Figura 11: Diagrama d'estats d'un contacte/assistent**

Del diagrama anterior cal destacar que un assistent que hagi denegat una convocatòria pot tornar a ser convocat eliminant i tornant a afegir-lo a la llista d'assistents (i el mateix per un assistent confirmat).

### 3.5 Classes de frontera, de control i de les operacions

Per a cada cas d'ús de la gestió de trobades es farà un diagrama de seqüències.

A tots els casos d'ús s'ha posat que l'actor demana una opció a la classe de frontera Menu, que correspon al punt d'entrada de l'aplicació, i aquesta passa a la classe de control GestorMenu, que crida a la classe de control principal del cas d'ús. Els noms dels missatges seran operacions de les classes destinatàries; aquells que van de les classes de frontera a l'actor no li demanen cap operació.

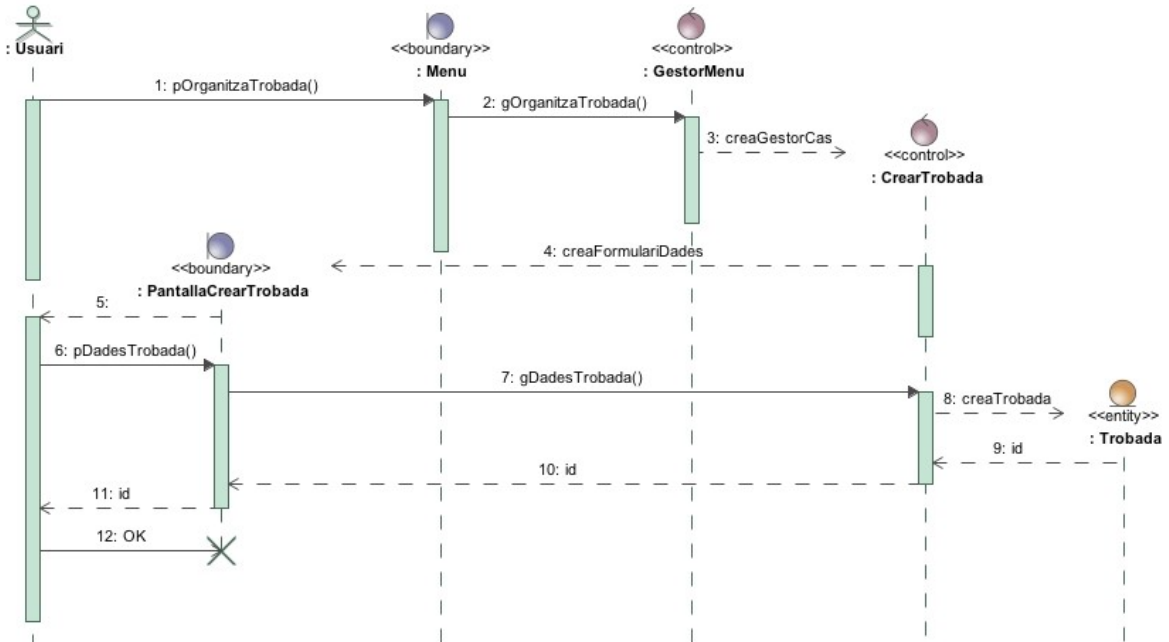


Figura 12: Diagrama de Seqüència "Organitzar trobada"

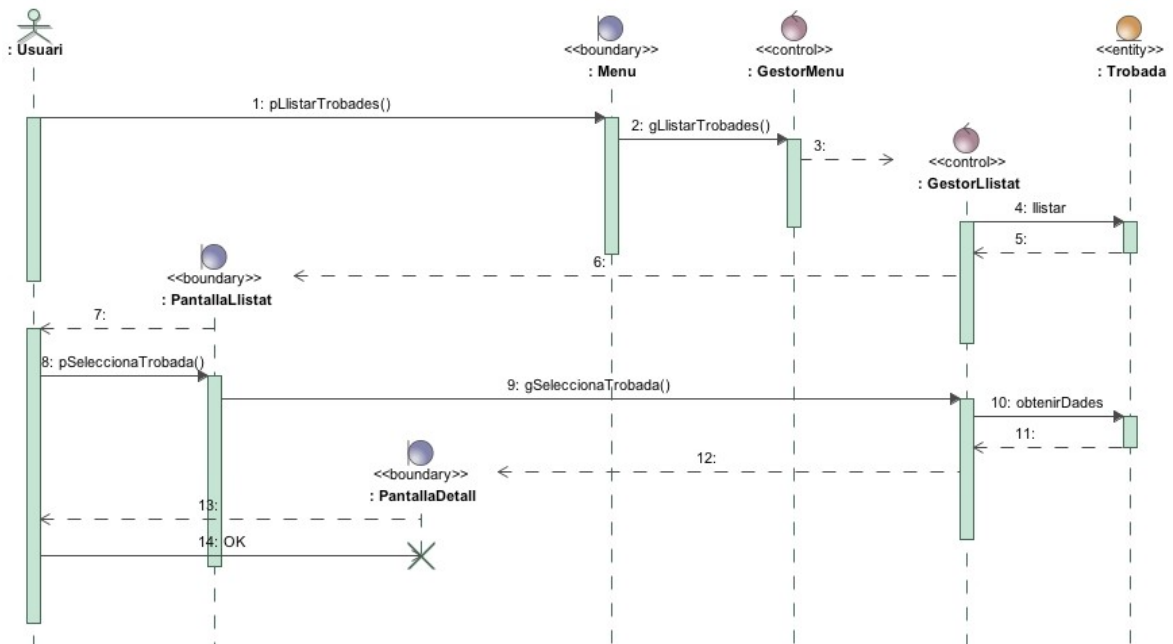


Figura 13: Diagrama de Seqüència de "Detall trobada"

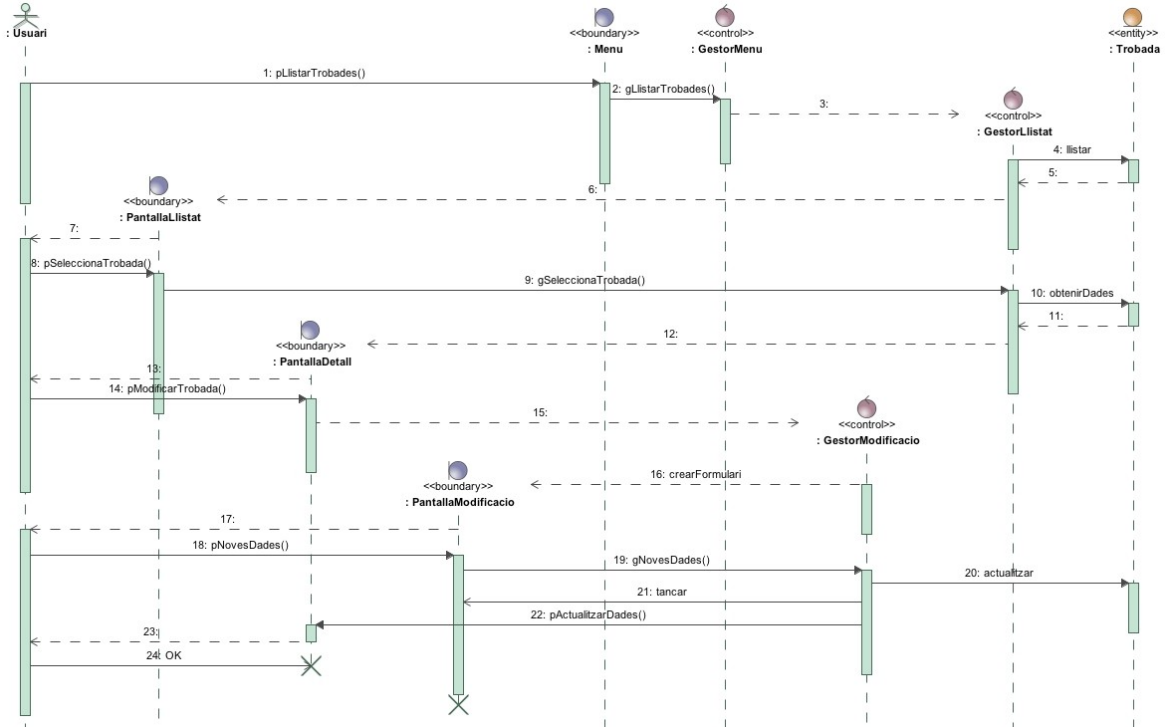


Figura 14: Diagrama de Seqüència de "Modificar trobada"

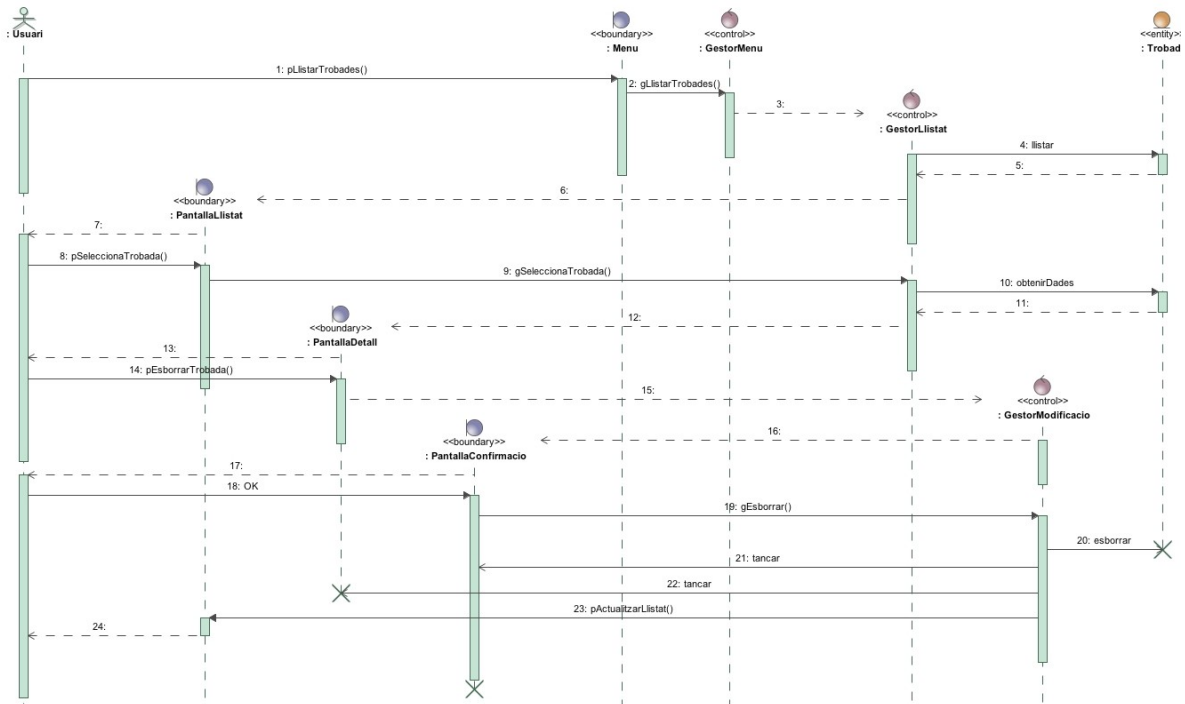


Figura 15: Diagrama de Seqüència de "Cancel·lar trobada"

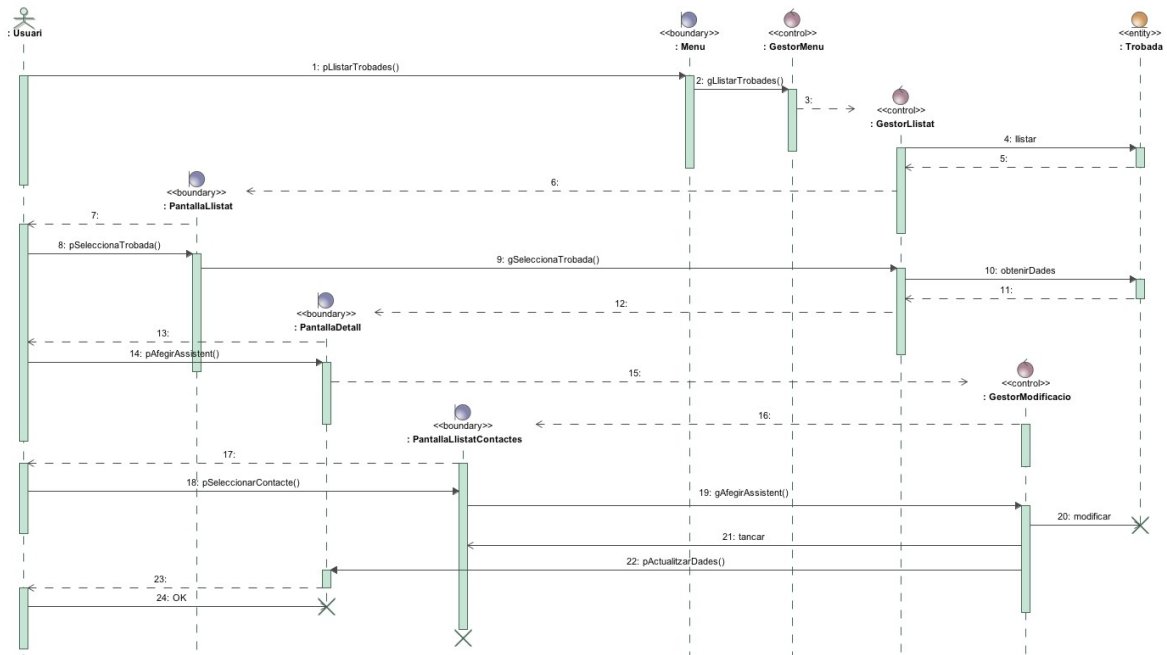


Figura 16: Diagrama de Seqüència corresponent a “Afegir assistent”

La resta de casos d'ús, inclosos els referents a la gestió de contactes i de convocatòries són totalment anàlegs als presentats fins ara i per aquest motiu no s'inclouen els seus diagrames de seqüència.



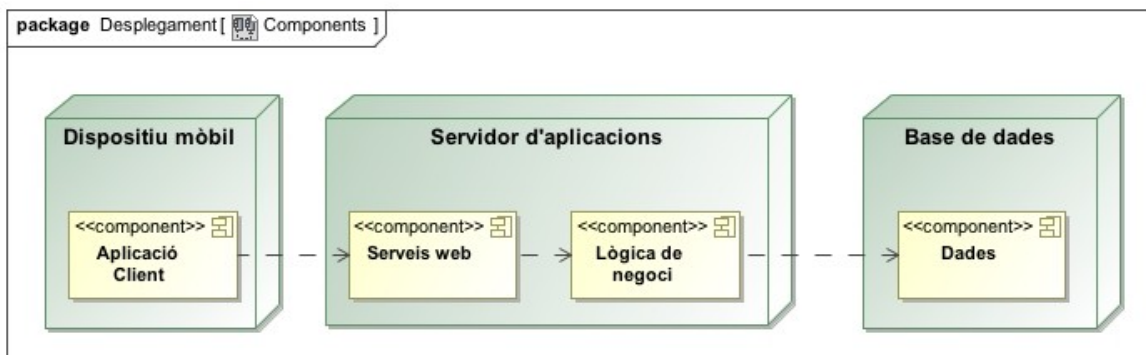
## 4 Disseny

### 4.1 Arquitectura

El sistema desenvolupat es divideix en dos subsistemes clarament identificables:

1. **Subsistema client.** Instal·lat en el dispositiu mòbil de l'usuari, inclou la interfície gràfica amb la que s'interactua. Les seves característiques tècniques estan condicionades per la plataforma de desenvolupament utilitzada, en aquest cas WAC.
2. **Subsistema servidor/remot.** Desplegat en un servidor d'aplicacions és l'encarregat de l'accés a les dades. Tindrà una arquitectura JEE i segons aquesta especificació es dividirà en capes intentant que siguin el més desacoblades possible:
  - **Presentació.** Com la UI de l'aplicació es trasllada al dispositiu client serà substituïda per una capa d'accés a la lògica de negoci implementada mitjançant serveis web.
  - **Lògica de negoci.** Implementació de les funcionalitats.
  - **Persistència.** Enregistrament perdurable de les dades.

En el següent diagrama s'il·lustra la ubicació de cada component:



**Figura 17: Diagrama de desplegament**

En cap cas s'utilitzaran les capacitats d'emmagatzematge local dels telèfons per lo que totes les dades hauran de ser manipulades per mitjà dels serveis del subsistema remot.

Per comunicar els dos subsistemes es recorrerà a l'ús del protocol REST/HTTP enviant les dades serialitzades en format JSON. El criteri seguit en la definició de les operacions ha estat:

- Si es tracta d'afegir una entitat nova, el mètode HTTP escollit és POST
- Si en canvi es tracta d'una modificació el mètode és PUT
- En el cas d'un esborrament, el mètode és DELETE
- I finalment, tots les operacions de consulta es fan mitjançant el mètode GET

Cada petició que generi el client incorporarà un *token* d'un sol ús que impossibiliti que un usuari malintencionat pugui capturar la petició i accedir a les dades del servidor.

L'especificació detalla del protocol és la que es presenta a continuació:

<b>Serveis generals dels usuaris</b>	
<b>URL base</b>	<b>/services/users</b>
	POST /register
	POST /login
	POST /location/{owner}/{token}
	GET /contactlocation/{contact}/{owner}/{token}

<b>Serveis per a la gestió de contactes</b>	
<b>URL base</b>	<b>/services/contacts</b>
	POST /add/{token}
	PUT /change/{token}
	DELETE /remove/{contact}/{owner}/{token}
	POST /addgroup/{token}
	PUT /change group/{token}
	DELETE /removegroup/{group}/{owner}/{token}
	POST /addtogroup/{group}/{owner}/{token}
	POST /addtogroup/{group}/{owner}/{contact}/{token}
	DELETE /removefromgroup/{group}/{owner}/{token}
	DELETE /removefromgroup/{group}/{owner}/{contact}/{token}"
	GET /get/{contact}/{owner}/{token}
	GET /list/{owner}/{token}
	GET /getgroup/{group}/{owner}/{token}
	GET /listgroups/{owner}/{token}
	GET /members/{group}/{owner}/{token}

<b>Serveis per a la gestió de trobades</b>	
<b>URL base</b>	<b>/services/meetings</b>
	POST /organize/{token}
	PUT /change/{token}
	DELETE /cancel/{meeting}/{owner}/{token}
	POST /invite/{meeting}/{owner}/{token}
	POST /invite/{meeting}/{owner}/{contact}/{token}
	DELETE /calloff/{meeting}/{owner}/{token}
	DELETE /calloff/{meeting}/{owner}/{contact}/{token}
	POST /accept/{meeting}/{user}/{token}
	POST /reject/{meeting}/{user}/{token}
	POST /arrived/{meeting}/{user}/{token}
	GET /get/{meeting}/{token}
	GET /list/{owner}/{token}
	GET /called/{user}/{token}
	GET /pending/{user}/{token}
	GET /attendees/{meeting}/{token}
	POST /geocode
	POST /ownerloc/{meeting}/{owner}/{token}

## 4.2 Frameworks i programari base

Els *frameworks* que s'utilitzaran en el desenvolupament de l'aplicació són:

- Interfície gràfica. **Sencha Touch**
- Interacció amb els serveis del telèfon: **WAC 2.0**
- Passarel·la de serveis web: **Apache CXF 2.5**
- Validacions de les entitats de negoci: **JSR 303 (Hibernate Validator)**
- Accés a dades. **JPA**

En el client, el software base a utilitzar serà:

- L'entorn d'execució **Borqs (Borqs Market)**.

I en el cas del servidor:

- Màquina virtual: **Sun JDK 1.6**
- Servidor web: **Apache 2.2.21 + mod\_jk 1.2.31**
- Servidor d'aplicacions: **glassfish 3.1.1 + postgresql JDBC driver 9.1**
- Servidor BBDD: **Postgres 8.4.7**

## 4.3 Patrons

Els patrons utilitzats són:

- **MVC (Model/View/Controller)**. La principal avantatge que s'obté és la separació clara de les diferents capes de l'aplicació i de les responsabilitats de cadascuna.
- **Factory**. La gestió de cada entitat de negoci s'agruparà en una classe que permetrà crear, cercar, modificar i esborrar-les.
- **DAO**. Encapsulen tots aquells accessos a components d'infraestructura o repositoris que no es facin mitjançant una Factory (entitats del domini de l'aplicació), com per exemple l'accés a un servidor de correu SMTP.
- **Façade**. Cada subsistema oferirà els seus serveis per mitjà d'una classe que amagarà si la informació s'obté per mitjà d'una Factory o d'un DAO. Quan aquesta "Façade" s'utilitzi per publicar els mètodes de negoci, en direm "**Business Façade**".
- **Singleton**. A utilitzar en aquelles classes que només contenen lògica i no representen cap entitat del domini.

## 4.4 Persistència. Entitats i atributs

### **USUARI\_REGISTRAT**

id\_usuari, telèfon, nom, cognoms, email, contrasenya, pregunta\_secreta, resposta  
{telèfon} és clau alternativa

### **CONTACTE** (entitat dèbil de USUARI\_REGISTRAT)

id\_contacte, id\_usuari, telèfon, nom, cognoms, adreça, població, codi\_postal, país, fax, email  
{id\_usuari } és clau forana a USUARI\_REGISTRAT  
{telèfon} és clau forana a USUARI\_REGISTRAT  
{id\_usuari, telèfon} és clau alternativa

### **GRUP** (entitat dèbil de USUARI\_REGISTRAT)

id\_grup, id\_usuari, nom, descripció  
{id\_usuari } és clau forana a USUARI\_REGISTRAT

### **GRUP\_MEMBRES** (materialització de la relació M:N entre CONTACTE i GRUP)

id\_grup, id\_usuari, id\_contacte  
{id\_usuari, id\_grup } és clau forana a GRUP  
{id\_usuari, id\_contacte } és clau forana a CONTACTE

### **TROBADA**

id\_trobada, descripció, organitzador, motiu, ubicació, durada, comentaris, data\_hora  
{ organitzador } és clau forana a USUARI\_REGISTRAT  
{ ubicació } és clau forana a UBICACIO

### **ASSISTENT** (entitat associativa entre TROBADA i CONTACTE)

id\_contacte, id\_usuari, id\_trobada, resposta, acceptat, arribat  
{id\_usuari, id\_contacte } és clau forana a CONTACTE  
{id\_trobada } és clau forana a TROBADA

### **CONVOCATORIA** (entitat associativa entre TROBADA i USUARI\_REGISTRAT)

id\_usuari, id\_trobada, resposta  
{id\_usuari } és clau forana a USUARI\_REGISTRAT  
{id\_trobada } és clau forana a TROBADA

## UBICACIO

id\_ubicació, posició\_gps

### UBI\_EXACTA (subclasse d'UBICACIO)

id\_ubicació, adreça

{ id\_ubicació } és clau forana a UBICACIO

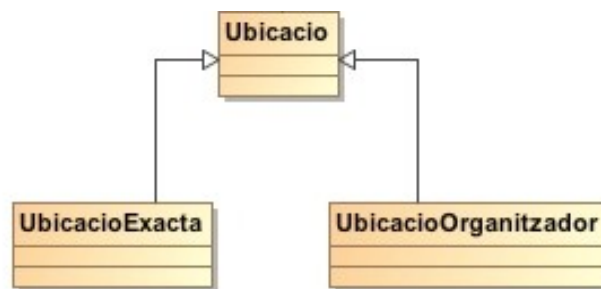
### UBI\_SEGUEIXME (subclasse d'UBICACIO)

id\_ubicació

{ id\_ubicació } és clau forana a UBICACIO

## 4.5 Revisió model estàtic

Respecte a les classes del domini identificades durant la fase d'anàlisi, es proposa eliminar la jerarquia de les ubicacions:



**Figura 18: Jerarquia de classes de les ubicacions**

En el seu lloc es proposa utilitzar una única classe que inclogui tots els atributs possibles de qualsevol tipus, més un que indiqui el tipus. En el cas que algun atribut no tingui sentit per a un tipus concret cal assignar-li el valor "null".

## 4.6 Classes de disseny

Aplicant al component servidor els patrons Façade, DAO i Factory les classes resultants tindran un dels següents rols:

Servei web	Classes que ofereixen els mètodes públics "de negoci" al component client. Són les encarregades de realitzar les validacions i de generar els missatges d'error adients.
Component de negoci	Internament utilitzen factories i DAOs per obtenir i manipular les entitats del domini. Són els encarregats d'obrir i tancar connexions i transaccions.
Factoria	Encapsula la gestió de les entitats de negoci. En la implementació realitzada aquest rol s'ha delegat totalment en la classe EntityManager de JPA.
DAO	Encapsula els accessos a aquells components d'infraestructura o

repositoris que no es facin mitjançant una Factoria. Només s'han utilitzat a mode didàctic i només quan l'operació a realitzar era molt senzilla.

Si representem gràficament les dependències entre aquestes classes:

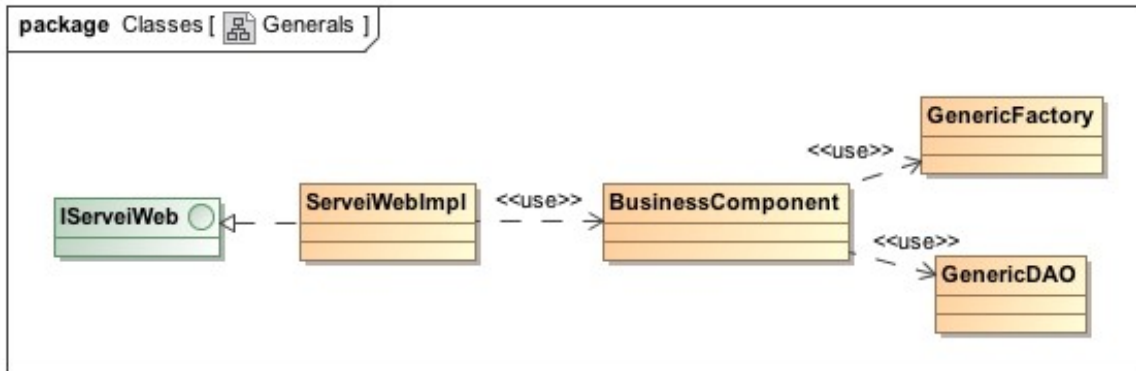


Figura 19: Estructura de classes del component servidor

I aplicant aquest mateix criteri als diferents serveis:

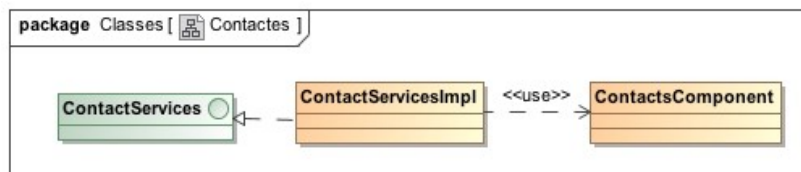


Figura 20: Diagrama de classes de servidor associades a la gestió de contactes

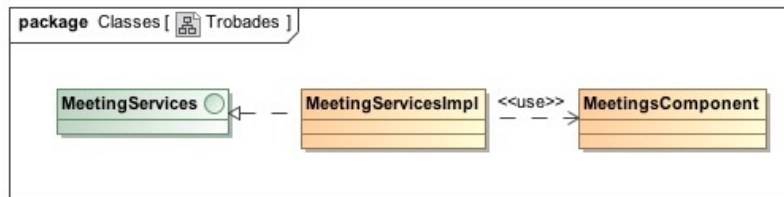


Figura 21: Diagrama de classes de servidor associades a la gestió de trobades

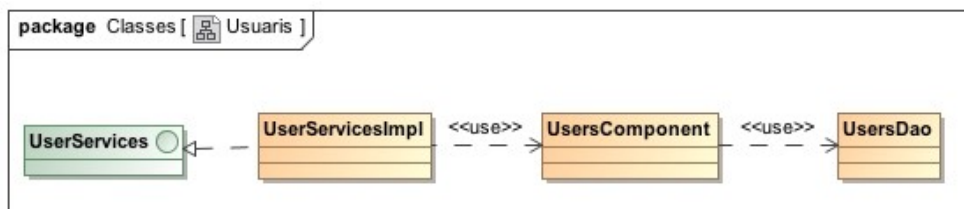


Figura 22: Diagrama de classes de servidor associades als serveis d'usuaris

## 4.7 Disseny de la UI

Tot seguit es presenta un recull de les principals finestres de l'aplicació. Pot consultar-se la llista completa en el manual d'usuari de l'annex 1.





Figura 23: Exemple de disseny gràfic de la interfície d'usuari



## 5 Implementació

### 5.1 Introducció

Respecte a llista de casos d'ús presentada en el capítol 2, cal destacar els següents canvis:

- Cas d'ús número 4: "Convidar". En la pantalla d'enviament del SMS d'invitació no s'accedeix a l'agenda del telèfon i cal introduir el número destí manualment.
- Cas d'ús número 5: "Sincronitzar contactes". No hi ha BBDD local de contactes, tota la informació únicament està al component servidor, per tant, s'ha eliminat aquesta funcionalitat.
- Cas d'ús número 6: "Llistar agenda". Considerat opcional i no implementat.
- Cas d'ús número 7: "Veure detall agenda". Considerat opcional i no implementat.
- Cas d'ús número 34: "Recordar trobada/reunió". Considerat opcional i no implementat.
- Cas d'ús número 35: "Informar posició". S'ha substituït la publicació automàtica de la posició per una opció de publicació manual sota demanda.
- La gestió de grups també s'ha considerat opcional i no s'ha implementat. Això afecta als casos d'ús: 14,15,16,17,18,19,20 i 21

En conclusió s'han implementat tots els casos d'ús considerats obligatoris i els recomanables, deixant de banda la majoria dels opcionals.

Pel que fa a les decisions de disseny, tot i que l'aplicació genera i intercanvia correctament el *token* de seguretat en cada invocació a un servei web, aquest no es valida i per tant, en la situació actual, la seva utilitat és quasi nul·la.

### 5.2 Aspectes a destacar de la solució

#### 5.2.1 Utilització de Sencha Touch segons el patró MVC

Tot el desenvolupament de la interfície gràfica s'ha realitzat aplicant el patró MVC al codi Javascript. En la següent figura pot observar-se com l'estructura definida s'adapta perfectament a aquest patró:

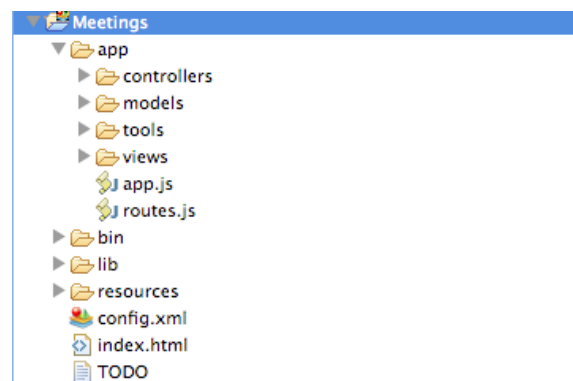


Figura 24: Aplicació del patró MVC en la UI

A més a més de les invocacions al component remot, també la definició dels esdeveniments a gestionar està en el codi del controlador de forma que les vistes són exclusivament components de visualització:

```
Ext.regController("meetings.controllers.Contacts", {
    reloadStore: false,

    index: function(back) {
        if (!this.card || this.reloadStore){
            this.card = this.application.viewport.getActiveItem().getActiveItem();

            var url=this.application.serverURL+'/services/contacts/List/'+
                this.application.currentUser.id+'/'+this.application.currentUser.token;
            this.card.list.store.proxy.url = url;
            this.card.list.store.load();

            this.card.list.on({
                select: this.show,
                scope: this
            });

            this.card.listpanel.query('#filterField')[0].on({
                keyup: this.filter,
                scope: this
            });

            this.card.listpanel.query('#addButton')[0].on('tap', this.compose, this, {
                operation: 'add'
            });
        }
    }
});
```

Figura 25: Definició de la gestió d'esdeveniments

```
this.dockedItems = [{
    dock: 'top',
    xtype: 'toolbar',
    title: 'Contact',
    items: [{
        text: 'Back',
        ui: 'back',
        itemId: 'backButton'
    }, {xtype: 'spacer'}, {
        text: 'Edit',
        ui: 'edit',
        itemId: 'editButton'
    }
    ]
}, {
    dock: 'bottom',
    xtype: 'toolbar',
    items: [{xtype: 'spacer'},
    {
        iconCls: 'trash',
        iconMask: true,
        itemId: 'delButton'
    }
    ]
}];

meetings.views.ContactDetails.superclass.initComponent.apply(this, arguments);
```

Figura 26: Exemple de vista

## 5.2.2 Reutilització de components

Totes aquelles pantalles de l'aplicació que són susceptibles de ser visualitzades molts cops només es construeixen una vegada i posteriorment es reutilitzen tants cops com faci falta. Ara bé, quan l'usuari realitza alguna operació que origina la necessitat de canviar-ne el contingut existeixen dues possibilitats d'actualitzar la informació:

1. Manipulant directament l'objecte que conté la informació que es visualitza. Per exemple, en el cas dels llistats els registres que el formen s'emmagatzemen en una estructura pròpia de Sencha Touch anomenada *store*; doncs bé, si un usuari afegeix un nou element enlloc de tornar a demanar totes les dades al servidor el que es fa és afegir directament l'element al *store* en qüestió. (veure figura 28)

2. I quan la càrrega directa descrita en el punt anterior no és possible, s'indica al controlador responsable que cal actualitzar la vista i és aquest l'encarregat de reconstruir-la.

```
doAccept: function(object, event, options){
    var req = Ext.Ajax.request({
        url: this.application.serverURL+"/services/meetings/accept/"+options.record.get('id')+"/"+
            this.application.currentUser.id+"/"+this.application.currentUser.token,
        method: 'POST',
        disableCaching: true,
        scope: this,
        success: function(response, opts) {
            console.log('meeting ('+options.record.get('id')+') accepted');
            this.card.list.store.remove(options.record);
            Ext.ControllerManager.get('meetings.controllers.Meetings').reloadStore = true;
            Ext.ControllerManager.get('meetings.controllers.Notifications').reloadStore = true;
            this.index();
        },
    },
```

Figura 27: Exemple de petició de refresc

### 5.2.3 Invocació de l'aplicació client als serveis remots

Totes es realitzen mitjançant una petició AJAX invocant el mètode *Ext.Ajax.request*:

```
create: function() {
    var record = this.form.getRecord();
    this.form.updateRecord(record, true);
    this.updateRecordPhones(record, this.form.getValues());

    var validator = new Validator(this.form, '#contactFormFieldset');
    if(validator.validateRecord(record)){
        var req = Ext.Ajax.request({
            url: this.application.serverURL+"/services/contacts/add/"+this.application.currentUser.token
            method: 'POST',
            headers: {'Content-Type': 'application/json', 'Accept': 'text/xml'},
            jsonData: Ext.encode({contact: this.getRecordData(record)}),
            disableCaching: true,
            scope: this,
            success: function(response, opts) {
                console.log('contact created with internal id: ' + response.responseText);
                record.setId(response.responseText);
                record.commit();
                this.card.list.store.add(record);

                this.index(true);
            },
            failure: this.failureCallback
        });
    }
},
```

Figura 28: Exemple d'invocació al component remot

### 5.2.4 Minimització de les dades enviades

En el diàleg client/servidor sempre s'ha intentat minimitzar al màxim les dades intercanviades: d'una banda, definint totes les relacions de les entitats de servidor en mode LAZY i no serialitzant-les quan es retorna l'entitat que les conté, i d'altra especialitzant molt els mètodes públics dels serveis web (cadascun d'ells només retorna una informació molt concreta).

## 5.2.5 Utilització de Google® Maps API

Per a la representació de la posició d'una trobada o d'un assistent en un mapa 2D, així com per obtenir les indicacions de com arribar a un lloc concret, s'ha integrat l'API de Google® Maps:

```
route.query('#routeButton')[0].on({
  tap: function(button){
    Ext.getBody().mask('');
    var directionsDisplay = new google.maps.DirectionsRenderer();
    var directionsService = new google.maps.DirectionsService();
    directionsDisplay.setMap(route.mapPanel.map);
    directionsDisplay.setPanel(route.tmpCont.getEl().dom);

    var request = {
      origin: location.latitude+', '+location.longitude,
      destination: meetingLocation,
      travelMode: google.maps.DirectionsTravelMode.DRIVING
    };
    directionsService.route(request, function(response, status) {
      if (status == google.maps.DirectionsStatus.OK) {
        directionsDisplay.setDirections(response);
        route.directionsServiceResponse = 'ok';
      }
      Ext.getBody().unmask();
    });
  },
  scope: scope
});
```

Figura 29: Càlcul de la ruta per anar a una ubicació

## 5.2.6 Georeferenciació

Per a la traducció d'adreces a posicions GPS s'utilitza el servei de georeferenciació de Google®:

```
public String getAddressCoordinates(String address, String authToken) throws
WebApplicationException {

    StringBuffer response = new StringBuffer();
    String googleService =
    "http://maps.google.com/maps/api/geocode/json?sensor=false&address=";

    try {

        String call =
            googleService+URLEncoder.encode(address, "UTF-8");
        URL url = new URL(call);
        BufferedReader in =
            new BufferedReader(new InputStreamReader(url.openStream()));

        String inputLine;
        while ((inputLine = in.readLine()) != null)
            response.append(inputLine);

        in.close();

        return response.toString();
    } catch (MalformedURLException e) {
        log.error(e);
        throw new ServerException();
    } catch (IOException e) {
        log.error(e);
        throw new ServerException();
    }
}
```

### 5.2.7 Operacions amb un elevat temps de resposta

En tots els casos que l'acció de l'usuari és asíncrona i té un temps de resposta elevat, l'aplicació bloqueja la interfície mitjançant una màscara. Aquest és el cas de totes les invocacions al component servidor i als serveis de l'API de Google®:

```
meetings.serverMask = new Ext.LoadMask(Ext.getBody(), { msg: "Server process" });
Ext.Ajax.on('beforerequest', function(connection,options){
    meetings.serverMask.show();
});
Ext.Ajax.on('requestcomplete', function(connection,options){
    meetings.serverMask.hide();
});
Ext.Ajax.on('requestexception', function(connection,options){
    meetings.serverMask.hide();
});
```

Figura 30: Bloqueig de la UI en les invocacions remotes

### 5.2.8 Emmagatzematge local de les credencials

En el cas que l'entorn d'execució ho suporti, l'aplicació utilitza el servei de SQLite d'HTML5 per emmagatzemar les credencials de l'usuari.

```
function MeetingsLocalDb() {
    this.meetingsDb = null;
    if (window.openDatabase){
        this.meetingsDb = openDatabase('meetingsDb', '1.0', 'Meetings local database', 2 * 1024 * 1024);
    }

    this.isSupported = function(){
        return (null!=this.meetingsDb);
    };

    this.open = function(){
        this.meetingsDb.transaction(function (tx) {
            tx.executeSql('CREATE TABLE IF NOT EXISTS configuration (id unique, property unique, value)');
        });
    };

    this.getProperty = function(property, callback){
        this.meetingsDb.transaction(function (tx) {
            tx.executeSql('SELECT value FROM configuration WHERE property = ?', [property], function (tx, results) {
                var len = results.rows.length;
                if (len>0){
                    callback(results.rows.item(0).value);
                } else {
                    callback(null);
                }
            });
        });
    };
};
```

Figura 31: Utilització de SQLite

### 5.2.9 Servidor: Sessions

La responsabilitat d'obrir i tancar les sessions JPA recau en un únic punt: els components de negoci. Tots aquests que treballin amb JPA segueixen el model:

```
EntityManager em = JPAHelper.getEntityManager();

    [accés a entitats]

em.close();
```

Del codi anterior es desprèn que en realitat l'obertura de les sessions es troba centralitzada en la classe *edu.uoc.pfc.wac.persistence.JPAHelper*.

```
public class JPAHelper {  
  
    private static Log log = LogFactory.getLog(JPAHelper.class);  
    private static final EntityManagerFactory emf;  
  
    static {  
        try {  
            // Use persistence.xml configuration  
            emf = Persistence.createEntityManagerFactory("meetingsJPA");  
        } catch (Throwable ex) {  
            log.error("Initial SessionFactory creation failed", ex);  
            throw new ExceptionInInitializerError(ex);  
        }  
    }  
  
    public static EntityManager getEntityManager() {  
        return emf.createEntityManager();  
        // Retrieve an application managed entity manager  
    }  
  
}
```

### 5.2.10 Servidor: Transaccionalitat

La responsabilitat de gestionar-les únicament estarà ubicada a nivell de component de negoci, de fet tots els mètodes de negoci que treballin amb transaccions han de seguir el model:

```
EntityManager em = JPAHelper.getEntityManager();  
em.getTransaction().begin();  
  
User user = em.find(User.class, contact.getOwner().getId());  
if (null!=user){  
    contact.setOwner(user);  
    em.persist(contact);  
}  
  
em.getTransaction().commit();  
em.close();
```

### 5.2.11 Servidor: Desconnexió dels objectes de negoci

El fet que dins del component de negoci es tanqui la sessió JPA implica que els objectes retornats al servei web es troben desconectats de la base de dades i per tant qualsevol intent d'accedir a un atribut que s'hagi definit amb un mode càrrega LAZY provocarà un error. Això és especialment important en el cas de les relacions entre entitats, com per exemple la llista dels assistents a una trobada.

La forma d'accedir a aquesta informació sempre serà mitjançant una segona crida al component de negoci per retornar únicament les col·leccions o altres entitats LAZY. De fet això encaixa perfectament amb el disseny de la interfície gràfica on les llistes d'elements sempre es mostren independentment de l'entitat a la que pertanyen: per exemple, la pantalla de detall d'una trobada mai mostra els assistents.

### 5.2.12 Servidor: connexions JDBC

En el cas de no utilitzar JPA i accedir directament per JDBC el mecanisme utilitzat inclou:

- Un *servlet* (*edu.uoc.pfc.wac.services.InstancesManager*) que s'executa en el moment que arranca l'aplicació i que carrega la configuració de connexió dels paràmetres definits en el fitxer *web.xml*
- Una classe de suport (*edu.uoc.pfc.wac.persistence.DbHelper*) que implementa el patró Singleton, instanciada pel *servlet* anterior i que conté un pool de connexions a la base de dades.

A partir d'aquests elements un component de negoci que vulgui accedir a la base de dades directament per JDBC seguirà el model:

```
Connection conn = InstancesManager.getDbHelper().getConnection();
try{
    UsersDao.insertUserPosition(conn, userId, location);
    conn.commit();
}
finally{
    if (null!=conn){ try {conn.close();} catch (SQLException e) {log.warn(e);}
    }
}
```

Destacar que en aquests casos la transacció no es gestiona a nivell JTA sinó directament amb la connexió.

### 5.2.13 Servidor: Log

El sistema de traces de l'aplicació s'ha desenvolupat a partir de la llibreria *log4j* i utilitzant la configuració del servidor d'aplicacions.

## 6 Evolució

Algunes propostes de millora que en la versió actual no s'han desenvolupat per manca de temps són:

### Component servidor

- Activar la validació del *token* de seguretat.
- Donada la sensibilitat de la informació involucrada (codi i contrasenya de l'usuari), caldria activar el protocol HTTPS en les invocacions al servei web de registre d'un nou usuari i també en el d'autenticació.
- Millorar la qualitat dels missatges d'error.

### Component client

- Afegir la gestió de grups de contactes.
- Afegir el control periòdic de la proximitat temporal de les trobades i l'enviament de les notificacions corresponents.
- Afegir la publicació automàtica de la posició GPS dels usuaris.
- Utilitzar l'agenda del telèfon per a l'enviament de missatges
- Activar la paginació dels llistats
- Revisió de la correcta eliminació dels objectes no utilitzats



## 7 Conclusions

Abans de tot cal recordar els objectius que ens havíem marcat a l'inici del PFC:

- Aplicar de forma pràctica els coneixements de programació orientada a objectes per tal de resoldre un problema real, i de forma anàloga en l'ús de la notació UML per especificar i dissenyar la proposta d'una solució
- Aprofundir en el coneixement de la tecnologia de serveis web i en l'ús de la tecnologia JPA
- Aprendre l'ús de *frameworks* de presentació basats en el llenguatge Javascript: Sencha Touch, Google® Maps API
- Aprendre les possibilitats que ofereix la plataforma WAC per al desenvolupament d'aplicacions mòbils
- Aplicar de forma pràctica molts dels patrons de disseny més populars

Un cop acabada la feina, estem en condicions de dir que tots ells han estat aconseguits i per tant en aquest sentit podem considerar el PFC com un èxit.

Ara bé, més enllà d'aquests objectius m'agradaria destacar unes lliçons més apreses durant l'elaboració del PFC:

- La dificultat de realitzar una planificació realista de la feina, tant pel que fa a la identificació de tasques com a l'esforç a dedicar a cadascuna d'elles. Segurament aquest és un aspecte que està en funció de l'experiència de la persona que fa la planificació en temes semblants i que per tant, en el meu cas, és quelcom a treballar.
- La dificultat de codificar en Javascript. D'una banda, per la manca d'entorns integrats de desenvolupament que facilitin la tasca, i d'altra per les característiques intrínseques del propi llenguatge: al ser interpretat tots els errors s'han de detectar en temps d'execució i a més no hi ha cap mena de comprovació del nombre d'arguments passats a les funcions ni dels seus tipus.
- Finalment també voldria destacar la poca qualitat de la documentació proporcionada per Sencha Touch. La major part dels problemes han hagut de ser resolts mitjançant fòrums, blogs o pàgines d'altres usuaris que ja s'havien trobat abans amb la mateixa problemàtica.

## 8 Glossari

A continuació es presenta el glossari del model de negoci:

**Agenda:** llista o conjunt de contactes.

**Assistent:** cadascun dels usuaris registrats en el sistema que han estat convidats per l'organitzador a una trobada/reunió.

**Contacte:** informació que un usuari guarda del telèfon d'una altra persona, tot i que aquesta no sigui un usuari registrat ni disposi de l'aplicació de gestió de trobades instal·lada en el seu dispositiu.

**Georeferenciació:** Informació geogràfica en què s'expressa la posició d'un objecte sobre la superfície terrestre mitjançant un sistema de referència. Font: <http://www.termcat.cat>

**GPS:** Sistema de localització de la posició de persones i de vehicles per mitjà de satèl·lits geostacionaris disposats al voltant de la Terra. La sigla GPS correspon a la denominació anglesa *global positioning system*. Font: <http://www.termcat.cat>

**Organitzador:** usuari registrat en el sistema que crea una trobada i hi convida altres usuaris.

**Reunió:** veure Trobada

**SMS:** Sistema que permet l'intercanvi de missatges en format de text mitjançant els telèfons mòbils i altres aparells electrònics. La sigla prové de l'anglès *short message service* (servei de missatges curts). Font: <http://www.termcat.cat>

**Trobada:** Esdeveniment organitzat per un usuari registrat, l'objectiu del qual es trobar-se físicament amb altres usuaris en un lloc i a una data i hora concreta.

**URL:** Adreça que indica la localització d'un recurs a Internet i que permet d'accedir-hi. La sigla prové de l'anglès *uniform resource locator* i *universal resource locator* ('localitzador uniforme de recursos' i 'localitzador universal de recursos'). Font: <http://www.termcat.cat>

## Bibliografia

L'única informació consultada han estat els manuals i exemples en línia següents:

[http://en.wikibooks.org/wiki/Java\\_Persistence](http://en.wikibooks.org/wiki/Java_Persistence)

<http://cxf.apache.org/docs/>

<http://www.sencha.com/learn/touch>

<http://blog.wolfgang-werner.net/restful-http-with-jax-rs/>

<http://www.wacapps.net/wiki>

<http://specs.wacapps.net/>

## Annex 1: Manual d'Usuari

Veure el document adjunt "jtorrentet\_ManualUsuari.doc".