



IMPLEMENTATION OF A QUALIFIED VOTING ALGORITHM BASED ON LEARNING ANALYTICS WITHIN A SYNCHRONIZED EVENTS CALENDAR.

Salvador Salas Truyols

[Grau d'Enginyeria Informàtica](#)

Fatos Xhafa

14/01/2018



Aquesta obra està subjecta a una llicència de [Reconeixement-Compartir Igual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-sa/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>IMPLEMENTATION OF A QUALIFIED VOTING ALGORITHM BASED ON LEARNING ANALYTICS WITHIN A SYNCHRONIZED EVENTS CALENDAR.</i>
Nom de l'autor:	<i>Salvador Salas Truyols</i>
Nom del consultor:	<i>Fatos Xhafa</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Àrea del Treball Final:	<i>Eines per a treball col·laboratiu</i>
Titulació:	Grau d'Enginyeria Informàtica
Abstract:	
<p>While almost all of the world's most prominent companies make an efficient use of cutting-edge technologies in aspects like manufacturing, operations, marketing and other divisions, many organizations don't yet fully accommodate available technologies to empower their teams, make them truly effective and coordinate their efforts. Instead, they choose to go fully analog or use decades old technologies to coordinate team efforts.</p> <p>In this thesis, our intention is to expand previous work done by Palou¹ et al. with respect to collaborative calendars and incorporate the ability to vote on calendar events in a way that members have unequal voting rights depending on their</p>	

¹ Palou, Daniel; Xhafa, Fatos (2013). *Implementació d'un algorisme de sincronització de calendari d'esdeveniments en un sistema Android*. Treball de Final de Grau presentat al Departament de Llenguatges i Sistemes Informàtics de la Universitat Politècnica de Catalunya. Consultat a <http://upcommons.upc.edu/handle/2099.1/16937>

activity. We will build a learning analytics system to track member activity per project and construct a model of voting rights based on this activity database.

The voting method will be a straightforward assessment of member activity but it also has the advantage to incorporate more data and weigh in more factors to determine the voting rights of each member.

Paraules clau:

Learning Analytics Weighed Voting Collaborative Open Source Solutions

Índex

1. Introducció	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball	1
1.3 Enfocament i mètode seguit	2
1.4 Planificació del Treball	3
1.5 Breu resumari de productes obtinguts	6
1.6 Breu descripció dels altres capítols de la memòria	6
2. Tecnologies usades:	7
3. Anàlisi de requeriments:	7
3.1 <i>Requisits funcionals:</i>	7
3.2 <i>Requisits no funcionals</i>	14
4. Disseny	14
Patró Arquitectònic:	15
Diagrama de classes:	15
Model Local	15
Model Remot	16
Diagrames de Sequència	16
Aplicació en Java	23
5. Implementació	23
Aspectes de codificació:	23
Desplegament	24
6. Experimentació	24
Eficiència del càlcul de la quota envers un nombre variable de votants	24
Eficiència del càlcul segons el tipus de votació:	26
7. Instruccions per instal·lació	27
8. Conclusions	29
Objectius aconseguits:	29
Objectius no aconseguits.	29
Futures ampliacions.	29
Valoració personal	30
9. Bibliografia	31
10. Annexos	32

1.Introducció

1.1 Context i justificació del Treball

Durant les últimes dècades, s'ha anat transformant el model de feina des d'un punt de vista tecnològic en molts de sentits però posant èmfasi en les comunicacions. Servidors, xarxes virtuals privades, comunicacions electròniques i tecnologies mòbils han descentralitzat la oficina i han facilitat la millora de les comunicacions en rapidesa i eficiència.

Tot i això el vessant col·laboratiu d'aquestes tecnologies no ha estat adoptat amb el mateix èxit per les empreses. Poques d'elles fins ara han adoptat eines que permetin als equips coordinar-se, prendre decisions i treballar conjuntament en un mateix projecte minimitzant la dispersió de dades i maximitzant la integració.

En paraules de Marie Puybaraud, cap d'investigació de solucions corporatives de JLL, els 3 aspectes clau que determinaran els resultats d'aquestes són la experiència humana, la tecnologia i els canvis en col·laboració i cooperació dins aquesta². En aquest sentit la feina que es descriurà a continuació vol ampliar aquest aspecte col·laboratiu d'una eina tecnològica open-source per integrar encara millor la presa de decisions.

La presa de decisions segueix sent un camp a on el suport de les eines digitals no ha tingut una gran acceptació. L'eina que desenvoluparem esten la ja presentada (Palou, 2016) amb la possibilitat de prendre decisions grupals que reflecteixin la participació dels membres i a seva implicació en el projecte, tot això en el context d'un sistema d'events compartits a un calendari grupal.

1.2 Objectius del Treball

L'objectiu principal d'aquest projecte consisteix en implementar un sistema de votació grupal qualificada al sistema desenvolupat per Palou consistent en la aplicació SweetSyncal com a client, combinada amb el servidor a on s'allotja SugarCRM Community Edition.

² Kirk, P. (2017). What new technologies mean for the future of office space. *National Real Estate Investor*, Retrieved from <https://search.proquest.com/docview/1902622138?accountid=15299>

Un altre objectiu del treball és incorporar els Learning Analytics al sistema. En la actualitat el sistema té incorporat un sistema de votació d'esdeveniments grupal basat en majoria de vot. No obstant, aquest sistema es podria millorar significativament incorporant dades de la activitat dels participants: Aquest conjunt de dades correctament interpretats ens podrien ajudar a desenvolupar un sistema de pesos en les votacions que reflectissin d'una manera més justa la participació de cada membre en el grup de treball i veure així millor recompensada la activitat realitzada.

D'una altra manera aquesta implementació pretén reforçar els incentius per treballar millor en equip. Si lliguem d'una banda una eina per millorar el treball en grup i d'altra els incentius per participar-hi podem ampliar i millorar l'ús que s'en fa i possibilitar la millora d'aquesta aplicació en un futur per altres membres de la comunitat open-source.

1.3 Enfocament i mètode seguit

El present treball prendrà com a punt de partida el treball de Daniel Palou *Implementation of a synchronization algorithm events calendar in an Android System* desenvolupat sota la direcció de Fatos Xhafa per la Universitat Politècnica de Catalunya.

En primer lloc instal·larem la part servidora (SugarCRM sobre Apache i MySQL) a la que afegirem els mòduls prèviament desenvolupats per Palou. Seguidament, instal·larem l'API a l'IDE per poder manipular-la des d'aquest entorn.

Un cop instal·lat tot caldrà fer les proves necessàries per comprovar que el sistema funciona. Per això es dissenyaran proves semblants a les que usarem un cop desenvolupada la funcionalitat, si bé no tan exhaustives.

Després passarem a especificar un sistema de votació qualificada. Aquí entrarem en el concepte matemàtic de majoria qualificada. Usarem la notació $[q: w_1, w_2, w_3, \dots, w_n]$ a on q és la quota necessària per aconseguir una majoria i w_1, w_2, w_3, \dots els pesos de cadascun dels n membres. Dissenyarem un sistema basat en les participacions grupals de cada membre que incentivaran la participació, però amb uns mínims i màxims que evitaran els casos extrems en els que un sol membre es pot convertir en un "dictador" (la quota q és inferior al pes del membre) o una "comparsa" (no hi ha cap combinació de pesos que sigui inferior a q **sense** la comparsa i igual o superior a q **amb** la comparsa). Per definició si hi ha un dictador la resta de membres són comparses.³ D'aquesta manera aconseguirem un sistema que incentivi els membres, sense arribar al cas d'una absoluta dominància i que al mateix temps no desincentivi completament els membres que no hagin participat prou.

³ Tannenbaum, Peter. *Excursions in Modern Mathematics*. 6th ed. Upper Saddle River: Prentice Hall, 2006. 48–83.

D'altra banda caldrà dissenyar el sistema que enregistrarà i analitzarà les participacions grupals. La base d'aquest sistema serà la mateixa utilitzada per el treball desenvolupat per Palou a la que s'afegirà aquesta funcionalitat. Es partirà d'un recompte simple de participacions amb la possibilitat d'ampliar-hi la sofisticació en futures revisions.

Així una primera divisió de la feina en fases:

FASE	DESCRIPCIÓ
FASE 1	Planificació i disseny del treball
FASE 2	Instal·lació del software requerit i proves inicials
FASE 3	Especificació del sistema de votació qualificada
FASE 4	Disseny del sistema de votació qualificada
FASE 5	Integració/ Implementació del sistema de votació qualificada
FASE 6	Proves i avaluacions
FI DEL PROJECTE	Presentació virtual del projecte

1.4 Planificació del Treball

A la següent graella es poden observar les diferents accions planificades per a cada fase, i una breu descripció dels recursos necessaris, així com el grau d'aconseguit de cada punt. A més, al final del punt es pot observar la planificació seguint el diagrama de Gantt.

Cada una de les PAC està integrada en alguna de les fases del projecte, ja que són part important del treball a realitzar.

	Dates	Fites	Finalitzat
FASE 1			
1r contacte amb el tutor del TFG	23/09/17	Missatge a través del correu electrònic	OK
Entrega del resum i la proposta	8/10/16	Enviament del pla de treball	OK
FASE 2			
Instal·lació de SugarCRM i altres components	23/9 – 30/10	S'han fet les proves i s'està redissenyant la aplicació	Executat amb retards entre 2/12 i 4/12
Proves de funcionament	9/10 – 30/10	Proves correctes. Detectat un problema amb JSON corregit amb una classe nova (veure apartat 'disseny')	Executat 4-10 Dec
FASE 3			
Especificació del sistema d'enregistrament de participació i learning analytics	15/10 - 19/10	Enviat	OK
Especificació de l'algorisme de votació	20/10 - 30/10	Completat en java i en PHP	Executat 13-15 Dec
Especificació de la integració amb els altres components del sistema	22/10 – 26/10	Enviat	OK
Entrega Especificació	26/10	Enviat	OK
FASE 4			
Disseny del sistema d'enregistrament de participació i learning analytics	27/10 – 31/10		Finalitzat 15-19 Dec
Disseny de l'algorisme de votació	2/11 – 3/11		Finalitzat 15-19 Dec
Disseny de la integració amb els altres sistemes	4/11 – 11/10		Finalitzat 15-19 Dec
FASE 5			
Desenvolupament del sistema d'enregistrament / Learning Analytics	12/11 – 20/11		Desenvolupat 20 Dec
Desenvolupament de l'algorisme de votació	21/11 - 30/11		Desenvolupat 21-22 Dec
Integració amb SugarCRM	1/12 – 8 /12		Desenvolupat 27 Dec – 2 Gen
Desenvolupament/ Modificació del Client amb Android Studio	9/12		AJORNAT a una futura ampliació. Únicament actualitzada la API
FASE 6			
Proves i avaluacions	11/12 – 20/12		Executades 7 Gen

Modificacions a la 1 ^a versió de la entrega	21/12 – 1/1		Desenvolupat 3-7 Gen
Actualització documentació i entrega final	2/1 – 12/1		OK
FI DEL PROJECTE			
Defensa virtual	15/01 - 21/01		

1.5 Breu sumari de productes obtinguts

El producte principal entregat serà el conjunt de programari format per un client Java (l'API de la aplicació Android actualitzada) i servidor SugarCRM degudament modificats i funcionant amb un sistema de votació qualificada modificat per fomentar la participació.

Com a productes auxiliars es lliuraran:

- Una memòria que recollirà el compendi d'exposició de motius i objectius a més de la redacció d'especificacions, comentaris a la implementació i annexes d'utilitat
- Un vídeo explicatiu de les noves funcionalitats explicant pas a pas com s'aconsegueixen els objectius amb les modificacions implementades

1.6 Breu descripció dels altres capítols de la memòria

A aquesta memòria s'inclouen

- Un comentari de les tecnologies usades (capítol 2)
- Un recull de requeriments per fer el disseny (capítol 3)
- El disseny de la aplicació amb diagrames de classes, de seqüència per explicar operacions i les noves taules emprades a la BD (capítol 4)
- Com s'han abordat problemes tècnics de codificació i les decisions preses pel desplegament de l'aplicatiu (capítol 5)
- Els resultats dels experiments fets en matèries de eficiència pel nou sistema de votació (capítol 6)
- El manual per instal·lar servidor i client (capítol 7)
- Conclusions i reflexions sobre el producte finalment desenvolupat (capítol 8)
- Bibliografia i annexos

2. Tecnologies usades:

Les tecnologies usades venen definides pel projecte que s'està estenent i només inclouen canvis en les versions dels paquets de software per motius de descatalogació o falta de suport:

-Java: El nostre sistema està dissenyat per funcionar en dispositius Android, el llenguatge de programació dels quals és el Java. Es dissenyarà un API en Java i es deixarà per una futura ampliació del projecte el desenvolupament d'una aplicació Android actualitzada que faci ús d'aquest API.

-SugarCRM: Continuant amb la feina de Palou es seguirà usant SugarCRM CE en la seva versió actual (6.5.26 – *build* 378, en contraposició a la versió 6.4. usada per Palou). D'aquesta manera el projecte queda situat dins d'un entorn empresarial i respectant la voluntat dels desenvolupadors originals, simularà un àmbit d'aquest tipus i no pas un d'acadèmic.

-PHP: SugarCRM està implementat en aquest llenguatge i aleshores s'haurà d'usar aquest per estendre les funcionalitats que li afegirem. La versió de PHP que usarem en aquest cas és la més moderna que igualment és compatible amb versions anteriors (5.6.30).

3. Anàlisi de requeriments:

Es definiran 2 tipus de requeriments: **funcionals** que defineixen el comportament del programari i **no funcionals**, que descriuen altres exigències de qualitat i d'altres aspectes (legal, accessibilitat,...). Els requisits funcionals es defineixen a la especificació que es fa a l'apartat 3.1. i els no funcionals a l'apartat 3.2 Per simplicitat no es repetiran els casos d'ús que es descriuen a Palou (2013) quant a que gran part del sistema no canvia els seus requeriments, només la part que afecta a la votació en el context d'una modificació de calendaris. En aquest cas afegirem un nou element a la base de dades anomenat **comptador de participació** ("*activity tracker*") que mesurarà el grau d'implicació dels participants mitjançant les seves intervencions. D'altra banda s'indicarà amb **color blau** les diferències amb els casos d'ús originals.

3.1 Requisits funcionals:

A continuació es farà la especificació de la resolució de canvis al calendari mitjançant una votació qualificada.

La votació, dins el sistema de calendaris compartits que usa la aplicació, és el sistema escollit per abordar modificacions als elements del calendari. A més, es requereixen votacions per aprovar determinats tipus d'elements, com ara una data d'entrega d'un projecte en la que tots els membres han de participar. Així la especificació tractarà aquestes situacions, i en particular la situació on els canvis

fets per un membre de l'equip no es poden fer persistents fins que els aprovin els altres membres de l'equip

Aleshores els casos d'ús que ens trobarem descriuen les situacions a les que els usuaris del sistema es poden trobar específicament dins aquest context i el comportament esperat del sistema:

Gestió de tasques:

Cas d'ús	Modificar Calendari
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació calendari
Descripció	L'usuari modifica alguna de les dades del calendari.
Precondició	L'usuari està loguejat i participa al calendari
Postcondició	El calendari es modifica amb les noves dades.
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica que vol modificar un calendari. 2. El sistema demana les noves dades a l'usuari. 3. L'usuari introdueix la informació. 4. El sistema verifica les dades introduïdes. 5. El sistema modifica el calendari i genera un esdeveniment per notificar als usuaris participants. El comptador de participació s'inicialitza pels nous membres, cas de que s'afegeixin
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>4A. Les dades no són correctes. 4A.1 El sistema notifica de l'error a l'usuari. 4A.2. Torna al pas 2</p> <p>5A. El calendari té més usuaris participants 5A.1. El sistema crea un <i>pending approval</i> per tal que la resta de participants pugin acceptar o rebutjar els canvis i notifica als participants.</p>

Cas d'ús	Eliminar Calendari
Actor	Usuari loguejat
Casos d'ús relacionats	Llistar calendaris
Descripció	L'usuari elimina un dels seus calendaris.
Precondició	L'usuari està loguejat i participa al calendari
Postcondició	El calendari, les seves tasques i el comptador de participació s'eliminen del sistema.
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica que vol eliminar un calendari. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema elimina el calendari, les seves tasques i el comptador de participació per aquell calendari. El sistema elimina els grups que no sigui necessari desar a nivell local.
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A L'usuari no confirma l'acció. 3A.1 Fi del cas d'ús.</p> <p>4A. El calendari té més usuaris participants</p>

	4A.1. El sistema crea un <i>pending approval</i> per tal que la resta de participants pugin acceptar o rebutjar la eliminació i notifica als usuaris participants.
--	--

Cas d'ús	Afegir Participant al Calendari
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació calendari
Descripció	L'usuari afegeix un o més grups/individus com a participants del calendari.
Precondició	L'usuari està loguejat i participa al calendari
Postcondició	Els grups/individus passen a participar al calendari
Curs principal	<ol style="list-style-type: none"> 6. L'usuari indica que vol afegir participants a un calendari. 7. El sistema demana els nous participants a l'usuari. 8. L'usuari introdueix la informació. 9. El sistema verifica les dades introduïdes. 10. El sistema afegeix els nous participants al calendari i genera un esdeveniment per notificar als nous usuaris. El comptador de participació inicialitza el comptador pels nous usuaris del calendari.
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>4A. Les dades no són correctes. 4A.1 El sistema notifica de l'error a l'usuari. 4A.2. Torna al pas 2</p> <p>5A. El calendari té més usuaris participants 5A.1. El sistema crea un <i>pending approval</i> per tal que la resta de participants pugin acceptar o rebutjar la adició de participants i notifica a aquests usuaris ja participants.</p>

Cas d'ús	Eliminar Participant al Calendari
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació calendari
Descripció	L'usuari elimina un o més grups/individus com a participants del calendari.
Precondició	L'usuari està loguejat i participa al calendari
Postcondició	Els grups/individus deixen de participar al calendari
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica que vol eliminar grups o participants de un calendari. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma la acció. 4. El sistema elimina els participants del calendari. El comptador de participació elimina les entrades dels corresponents participants.
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1 Fi del cas d'ús.</p>

Cas d'ús	Crear tasca
Actor	Usuari loguejat
Casos d'ús relacionats	Llistar tasques
Descripció	L'usuari crea una tasca.
Precondició	L'usuari està loguejat
Postcondició	La tasca es crea correctament al sistema local
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica que vol crear una nova tasca. 2. El sistema demana les dades a l'usuari.

	<p>3. L'usuari introdueix la informació de la tasca.</p> <p>4. El sistema verifica les dades introduïdes.</p> <p>5. El sistema crea la nova tasca a nivell local i, si cal, crea la alerta corresponent. Crida el cas d'ús <i>enviar tasques</i>. El comptador de participació suma pes a l'autor de la operació.</p>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació.</p> <p>1A.1 Fi del cas d'ús.</p> <p>4A. Les dades no són correctes.</p> <p>4A.1 El sistema notifica de l'error a l'usuari.</p> <p>4A.2. Torna al pas 2</p>

Cas d'ús	Modificar tasca
Actor	Usuari loguejat
Casos d'ús relacionats	Llistar tasques
Descripció	L'usuari modifica alguna de les dades de la tasca.
Precondició	L'usuari està loguejat i participa a la tasca.
Postcondició	La informació es modifica correctament al sistema local
Curs principal	<p>1. L'usuari indica que vol modificar una tasca.</p> <p>2. El sistema demana les noves dades a l'usuari.</p> <p>3. L'usuari introdueix la informació.</p> <p>4. El sistema verifica les dades introduïdes.</p> <p>5. El sistema modifica la tasca a nivell local i, si cal, crea, modifica o elimina l'alerta corresponent. Crida el cas d'ús <i>enviar tasques</i>. El comptador de participació suma pes a l'autor de la operació.</p>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació.</p> <p>1A.1 Fi del cas d'ús.</p> <p>4A. Les dades no són correctes.</p> <p>4A.1 El sistema notifica de l'error a l'usuari.</p> <p>4A.2. Torna al pas 2</p>

Cas d'ús	Afegir precedent a una tasca*
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació d'una tasca
Descripció	L'usuari afegeix una o més tasques com a precedents d'una tasca
Precondició	L'usuari està loguejat i participa a la tasca
Postcondició	La tasca s'afegeixen com a precedents a nivell local
Curs principal	<p>1. L'usuari indica que vol afegir precedents a una tasca.</p> <p>2. El sistema mostra a l'usuari les possibles tasques precedents.</p> <p>3. L'usuari escull les tasques precedents.</p> <p>4. El sistema afegeix les tasques com a precedents a nivell local. Crida el cas d'ús <i>Enviar tasques</i>. El comptador de participació suma pes a l'autor de la operació.</p>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació.</p> <p>1A.1 Fi del cas d'ús.</p> <p>2A. No hi ha possibles tasques precedents.</p> <p>2A.1 El sistema notifica a l'usuari que no hi ha tssques que puguin ser precedents de la tasca que es tracta.</p> <p>2A.2 Fi del cas d'ús.</p>

Cas d'ús	Eliminar precedent de una tasca*
Actor	Usuari loguejat

Casos d'ús relacionats	Veure informació d'una tasca
Descripció	L'usuari elimina una tasca com a precedent d'una altra tasca
Precondició	L'usuari està loguejat i participa a la tasca.
Postcondició	La tasca és eliminada com a precedent a nivell local
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica que vol eliminar una determinada tasca com a precedent. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema elimina la precedència a nivell local. Crida el cas d'ús <i>Enviar tasques</i>. El comptador de participació suma pes a l'autor de la operació.
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1 Fi del cas d'ús.</p>

Gestió d'esdeveniments i votacions:

Cas d'ús	Votar en un Meeting*
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació meeting
Descripció	L'usuari vol votar en un meeting al que participa
Precondició	L'usuari està loguejat, participa en el meeting i no ha votat encara.
Postcondició	Es desfa el vot de l'usuari si la votació ha finalitzat, es marca el meeting com a inactiu i es realitzen els canvis relacionats amb el resultat obtingut.
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema el seu vot. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma el vot. 4. El sistema desfa el vot. Es crida al cas "determinar majoria". Si tots els participants han votat o ja hi ha una majoria guanyadora d'acord amb els pesos establerts a cada vot, es marca el meeting com a inactiu, es marquen els canvis determinats pel resultat i es notifica als usuaris participants. Si hi ha un empat i ja han votat tots els participants, es repeteix el meeting.
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1 Fi del cas d'ús.</p> <p>4A. El meeting està inactiu. 4A.1. El sistema informa de l'error a l'usuari. 4A.2. Fi del cas d'ús.</p>

Cas d'ús	Acceptar/Rebutjar pending approval
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació pending approval.
Descripció	L'usuari vol acceptar o rebutjar un pending approval en el que participa
Precondició	L'usuari està loguejat, participa en el pending approval i no l'ha acceptat/rebutjat encara.
Postcondició	Es desfa el vot de l'usuari si la votació ha finalitzat, s'elimina el pending approval i es realitzen els canvis relacionats en funció del resultat.
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol acceptar o rebutjar un pending approval. 2. El sistema demana confirmació a l'usuari.

	<p>3. L'usuari confirma el vot.</p> <p>4. El sistema desfa la decisió. <i>Es crida al cas "determinar majoria". Si tots els participants han votat o ja hi ha una majoria guanyadora d'acord amb els pesos establerts a cada vot, es marca el meeting com a inactiu, es marquen els canvis determinats pel resultat i es notifica a l'usuari participants. Si hi ha un empat i ja han votat tots els participants, es rebutja el pending approval.</i></p>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1 Fi del cas d'ús.</p>

Cas d'ús	Afegir participant a un grup
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació grup
Descripció	L'usuari afegeix un o més usuaris com a membres d'un grup
Precondició	L'usuari està loguejat, i és l'administrador del grup.
Postcondició	Els usuaris escollits passen a formar part del grup
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol afegir usuaris al grup. 2. El sistema demana a l'usuari els nous participants. 3. L'usuari introdueix la informació demanada. 4. El sistema verifica les dades introduïdes. 5. El sistema afegeix els usuaris, els notifica que han estat afegits i també notifica als anteriors membres del grup. <i>S'inicialitza el comptador d'activitat per tots els nous membres a tots els calendaris als que pertany el grup.</i>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>4A. Les dades no són correctes. 4A.1. El sistema informa de l'error a l'usuari. 4A.2. Torna al pas 2.</p>

Cas d'ús	Eliminar participant a un grup
Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació grup
Descripció	L'usuari expulsa un membre d'un grup
Precondició	L'usuari està loguejat, i té permisos d'administrador del grup.
Postcondició	L'usuari escollit deixa de formar part del grup
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol eliminar un determinat participant d'un grup. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema verifica les dades introduïdes. 5. El sistema elimina l'usuari com membre del grup, el notifica que n'ha estat eliminat i també notifica a la resta de membres del grup. <i>S'elimina el comptador d'activitat d'aquell usuari per tots els calendaris als que pertany el grup.</i>
Cursos Alternatius	<p>1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1. Fi del cas d'ús.</p>

Cas d'ús	Sortir d' un grup
-----------------	-------------------

Actor	Usuari loguejat
Casos d'ús relacionats	Veure informació grup
Descripció	L'usuari surt d'un grup. Si és l'últim membre del grup el grup s'elimina el grup. Si no n'és l'últim membre però n'és l'administrador, s'escull un de nou.
Precondició	L'usuari està loguejat.
Postcondició	L'usuari deixa de formar part del grup
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol sortir d'un grup. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema elimina l'usuari del grup. Si és l'últim membre del grup el grup s'elimina el grup. Si no n'és l'últim membre però n'és l'administrador, s'escull un de nou. 5. S'elimina el comptador d'activitat d'aquell usuari per tots els calendaris als que pertany el grup.
Cursos Alternatius	3A. L'usuari cancel·la la operació. 3A.1 Fi del cas d'ús.

Cas d'ús	Acceptar/rebutjar sol·licitud*
Actor	Usuari loguejat
Casos d'ús relacionats	Llistar sol·licituds
Descripció	L'usuari vol acceptar o rebutjar una certa sol·licitud d'entrada a un grup
Precondició	L'usuari està loguejat.
Postcondició	La sol·licitud s'elimina correctament i, si és acceptada, l'usuari passa a ser membre del grup.
Curs principal	<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol acceptar o rebutjar una sol·licitud d'entrada a un grup. 2. El sistema elimina la sol·licitud i notifica de la decisió al propietari de la sol·licitud. Si és acceptada, s'afegeix l'usuari al grup i es notifica a tots els membres. 3. S'inicialitza el comptador d'activitat d'aquell usuari per tots els calendaris als que pertany el grup.
Cursos Alternatius	1A. L'usuari cancel·la la operació. 1A.1 Fi del cas d'ús.

Gestió de activitat d'usuaris i pesos en votacions:

Cas d'ús	Determinar majoria*
Actor	Sistema
Casos d'ús relacionats	Votar en un Meeting, PendingApproval
Descripció	El sistema calcula el pes del vot de cada usuari en un Meeting/PA i determina si hi ha una majoria.
Precondició	Un usuari acaba d'emetre un vot vàlid en un Meeting/PA.-
Postcondició	El resultat de la votació segons el pes de cada vot. Si s'aconsegueix una majoria es notifica al sistema per que declari el Meeting/PA com a inactiu.
Curs principal	<ol style="list-style-type: none"> 1. En emetre's un vot el sistema el contabilitza. 2. El sistema accedeix al comptador de participació ("<i>activity tracker</i>"), calcula el pes de tots els vots emesos fins al moment. 3. El sistema calcula la quota necessària per aprovar la moció 4. El sistema aplica els pesos al sentit de cada vot i determina si hi ha una majoria 5. En haver una majoria, el sistema es notificat per inactivar el Meeting/PA
Cursos Alternatius	4A. No s'arriba a una majoria i no han votat tots els participants 4A.1. Fi del cas d'ús.

	4B. No s'arriba a la quota i han votat tots els participants al Meeting/PA. 4B.1. El meeting es repeteix. En el cas del PA els canvis no s'enregistren i no es repeteix. 4B.2. En el cas del PA, aquest s'inactiva, juntament amb els vots dels participants. 4B.3. Fi del cas d'ús
--	--

NOTA: els casos marcats amb un asterisc i en vermell no han estat implantats. S'ha implantat una versió molt reduïda de l'algorisme per raons de temps i de disseny del codi de la extensió de les funcionalitats de SugarCRM a l'arxiu `SweetSyncaWebServiceImpl.php`. En una futura iteració aquestes funcionalitats se poden estendre molt fàcilment a aquests casos però requereixen d'un estudi en profunditat del codi al que ens referirem més endavant. En el cas "determinar majoria" s'ha implementat pels PA però no pels Meeting.

3.2 Requisits no funcionals

Requisit: fomentar la participació.

Descripció: El sistema de votació ha de ser dissenyat de manera que incentivi la participació dels membres al projecte sense desincentivar als que, per les circumstàncies que tinguin, no puguin participar. D'aquesta manera els pesos de la votació posaran especial cura a que el pes del vot augmenti amb la participació fins a un límit al que tots els participants tinguin prou pes per a determinar una majoria. Això vol dir que no hi haurà cap participant que votant en un sentit o en un altre no tindrà cap pes en la decisió (veure apartat 1.3)

Requisit: Seguretat

Descripció: Les dades de participació en el projecte han de estar degudament protegides i només un administrador hi ha de poder accedir.

Requisit: Transparència.

Descripció: El sistema ha de proporcionar informació del procés de presa de decisions basats en votacions qualificades. El sistema de notificacions pot recollir la informació necessària perquè els usuaris entenguin com s'ha pres la decisió

Requisit: Qualitat

Descripció: El sistema ha de garantir la robustesa de les votacions: manca de dobles votacions, pesos incorrectes, i notificacions correctes i puntuals als participants.

4. Disseny

El model local de la aplicació original no s'ha usat. Les funcionalitats afegides no canvien l'ús de SQLite per emmagatzemar la replica de les dades. Com no hem fet ús de la aplicació Android sinó només de la seva API hem prescindit de replicar el model en local. Quan s'han necessitat dades del servidor s'ha fet la petició REST corresponent i s'han emmagatzemat als beans del paquet 'model' que repliquen el diagrama anterior.

Model Remot

El model al servidor que allotja SugarCRM s'actualitza amb la nova taula "SS_ActivityTracker" que compta les intervencions "premiades" amb més influència de cada usuari dins el context d'un calendari. S'ha fet així perquè el calendari és el contenidor més general del disseny. Es podria haver fet un sol comptador d'activitat per usuari que s'usaria a tota la BD, però conceptualment el sistema no està enfocat així, sinó a la compartició de calendaris. Aleshores, aquesta nova taula tindrà un identificador de l'usuari, un identificador del calendari i un comptador d'activitat anomenat 'participation':

Taula SS_ActivityTracker

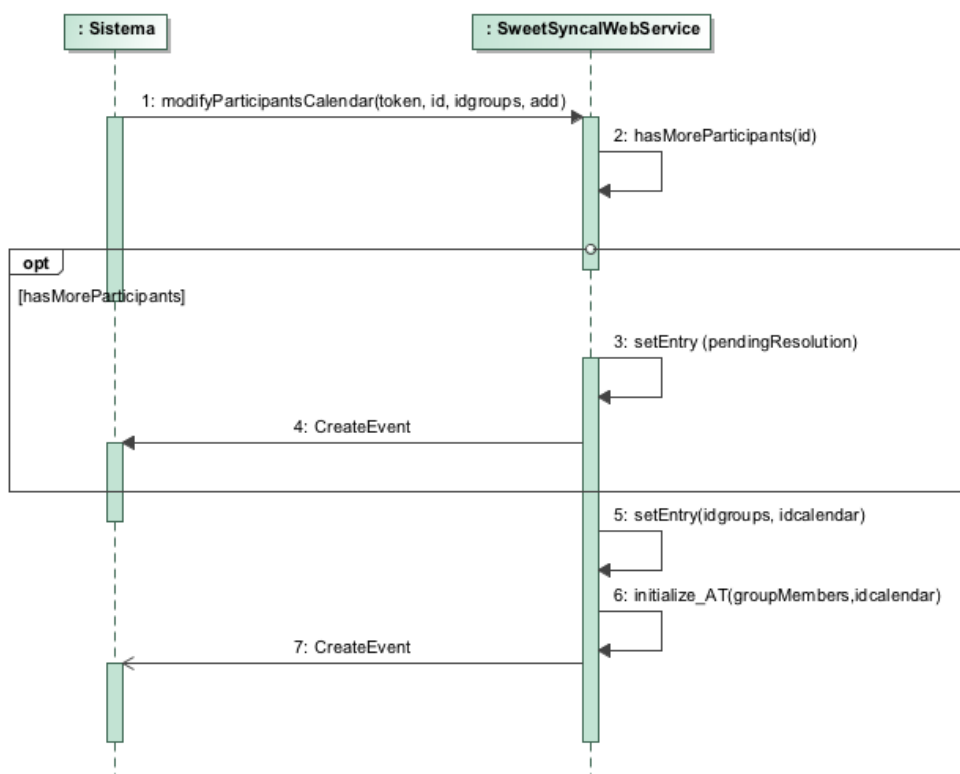
Columna	Tipus	Descripció
id	TEXT	Identificador comptador
iduser	TEXT	Identificador de l'usuari
idcalendar	TEXT	Identificador calendari
participation	INTEGER	Comptador activitat.

Diagrames de Seqüència

A continuació es mostren els diagrames de seqüència de les noves funcions. Aquí l'actor principal és el propi sistema perquè, si bé som els usuaris que fem ús directe de l'API a aquesta fase de desenvolupament, l'objectiu és incorporar aquesta API a una aplicació Android plenament funcional. Aleshores aquestes funcions seran completament transparents a l'usuari.

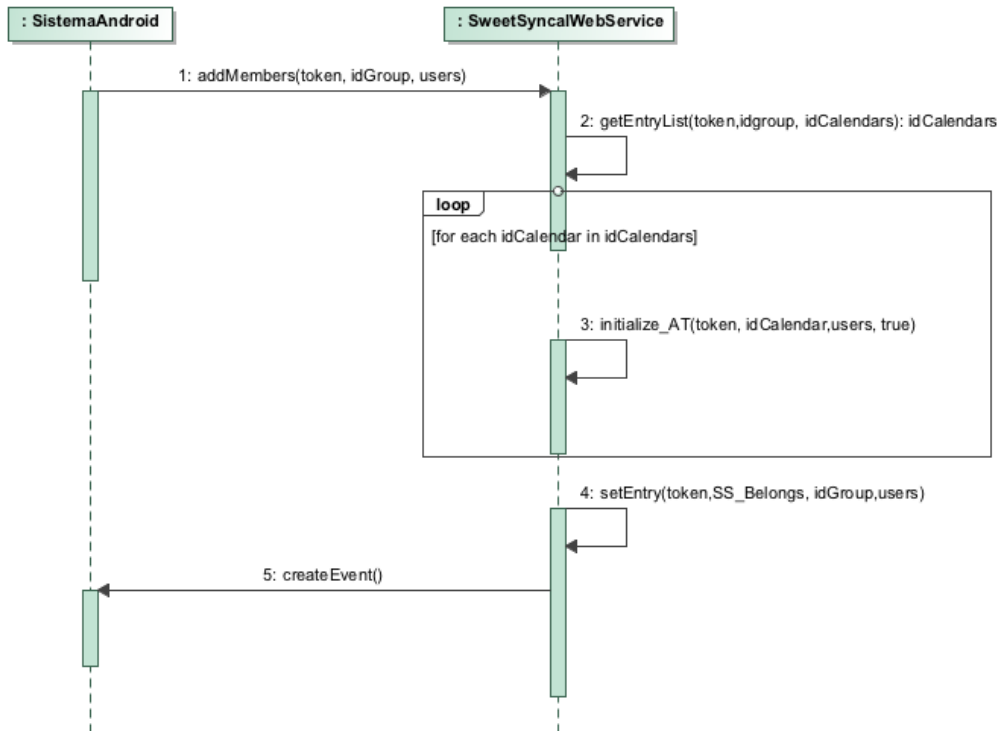
Operació modifyParticipantsCalendar(token, idCalendar, idGroups, add)

Aquesta operació associa o desassocia grups d'un calendari. La operació primer comprova si hi ha més participants al calendari i, en aquest cas, crea o modifica un pendingResolution. En cas contrari, la aplicació crea la associació a la taula SS_Shares i inicialitza els comptadors d'activitat per tots els membres del grup. En el cas de que es desassocii un grup d'un calendari s'esborra aquest comptador. En ambdós casos es notifica el resultat als membres actuals i als nous mitjançant events.



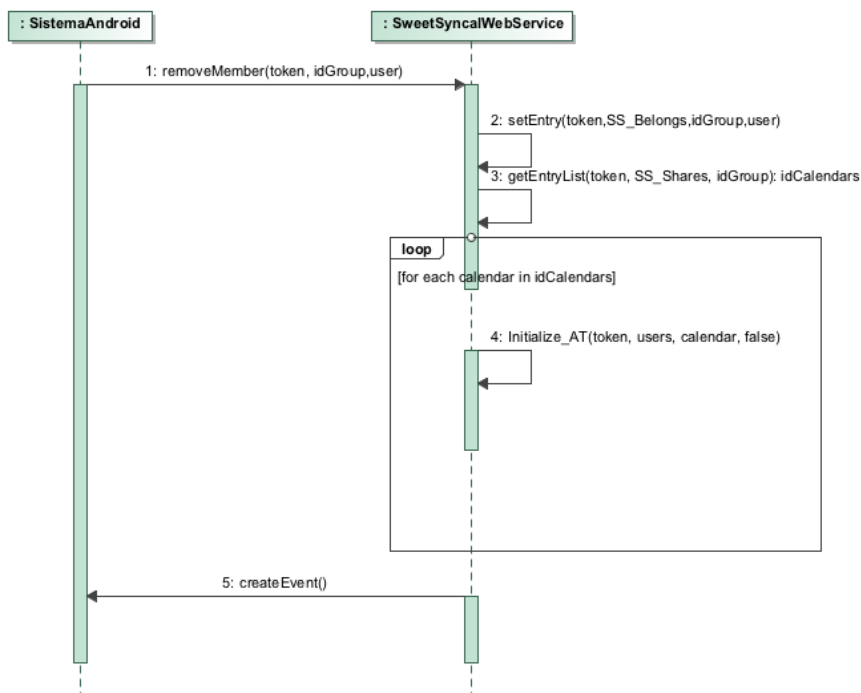
Operació addMembers(token, groupId, users)

Aquesta operació afegeix membres a un grup. El canvi que s'ha introduït és que el sistema comprova si el grup pertany a qualche calendari i, en cas afirmatiu, inicia el comptador d'activitat per als nous usuaris



Operació removeMember(token, groupId, user)

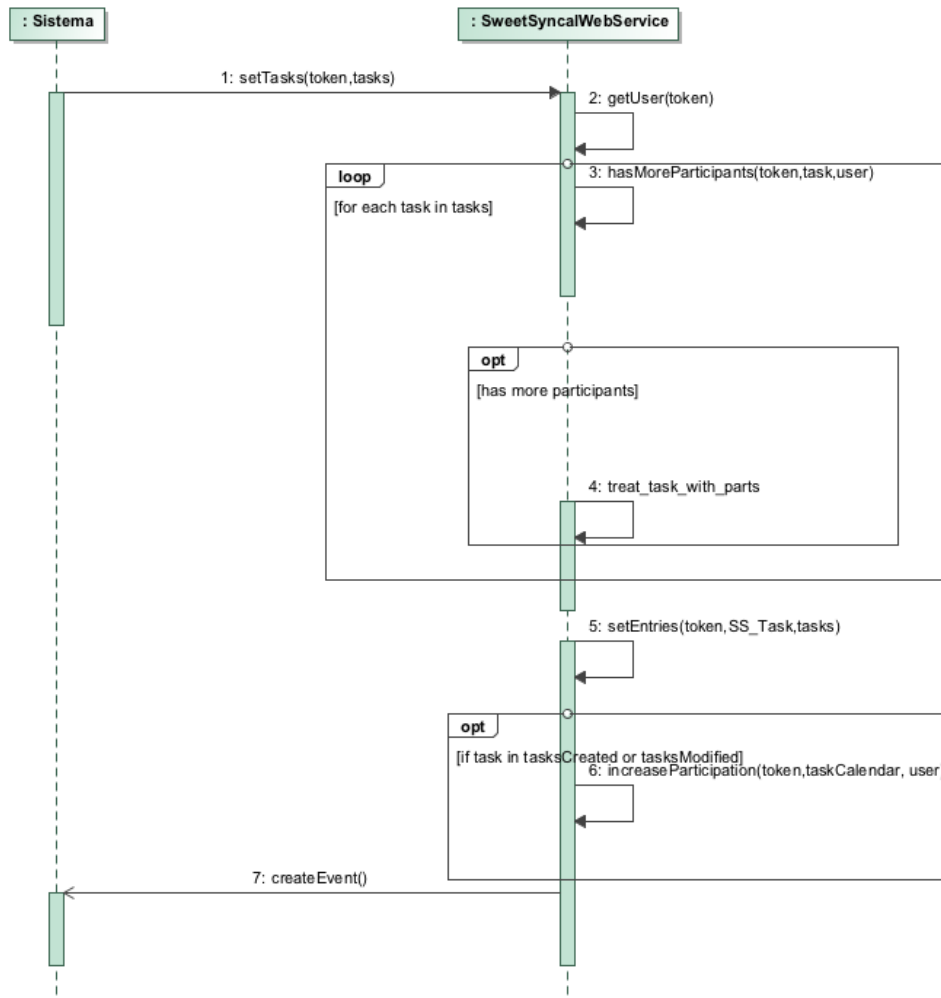
De manera similar a la operació anterior, aquesta funció lleva un membre d'un grup. A més d'això, ara la funció busca els calendaris als que el grup pertany i esborra els activity trackers corresponents als calendaris i usuari corresponent:



Operació setTasks(token, tasks)

Aquesta operació és la que s'ha usat per establir un augment dels ActivityTrackers. De la mateixa manera que s'ha fet per aquesta operació, en futures iteracions es podria afegir aquest augment del pes de votació a altres activitats, inclús variar el pes de cadascuna d'elles. S'ha fet només amb aquesta a mode de demostració i per una raó de temps: el codi és complex i en molts de casos hi ha diversos punts a on una acció s'acaba d'enregistrar (per exemple, accions a on, si hi ha més participants, s'estableix un pending approval i es completen a un altre lloc del codi quan tothom ha votat). Amb el temps limitat és arriscat modificar tot el codi responsable de cada activitat que es pot executar dins la aplicació. Aleshores, i servint aquesta operació per mostrar la dinàmica dels augments de pesos, s'ha optat per restringir només a aquesta operació el "premi" d'augmentar el pes de votació dins el context d'un calendari.

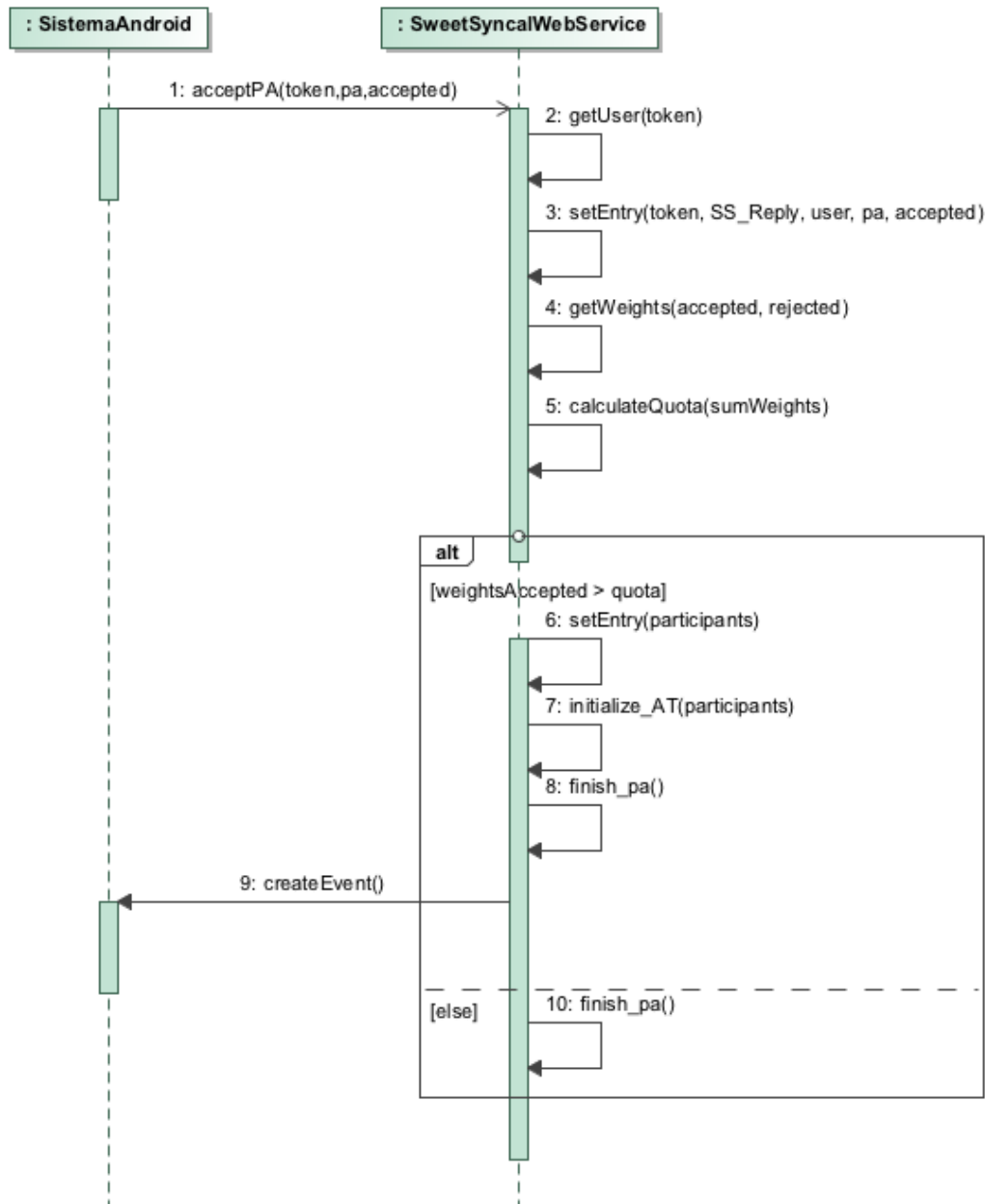
La operació crea, modifica o esborra tasques. Si hi ha altres participants o pending approvals els tracta separatament. Finalment dona a cada tasca el tractament apropiat (creació, modificació, cancel·lació). Les tasques creades o modificades criden el augment del comptador de participació, no així l'esborrament de les mateixes:



Operació acceptPA(token, pa, accepted)

Aquesta operació enregistra un vot i posteriorment fa un recompte dels vots ja emesos. Si no hi ha cap majoria qualificada i encara no han votat tots els membres, la operació no fa res més. Si hi ha una majoria o la totalitat dels vots han estat emesos sense arribar a la majoria qualificada (quota), el pending approval s'esborra i s'enregistren els canvis (quan s'ha arribat a la majoria de 'si') o es descarten.

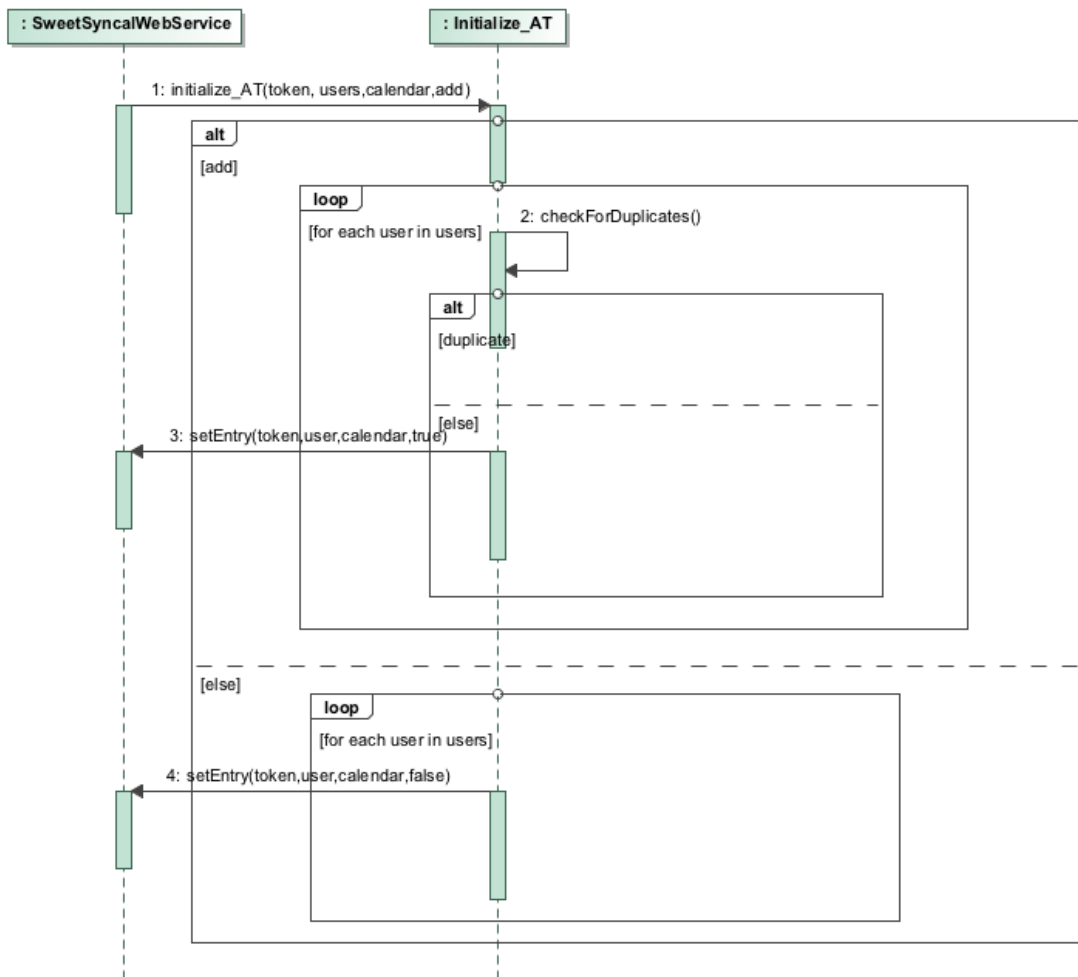
Les modificacions a aquesta operació, que bàsicament consisteixen en canviar la majoria simple per una qualificada amb recàlcul de quota guanyadora, es poden aplicar a la operació vote_meeting. Tampoc s'ha fet per les mateixes raons que la operació anterior. En ambdós casos es vol demostrar un sistema que en futures iteracions es pot implantar a la totalitat de la aplicació.



Operacions auxiliars:

Operació initialize_AT(token, users, calendar, add)

Com ja s'ha comentat aquesta operació inicialitza o esborra un comptador d'activitat a la taula SS_ActivityTracker per tots els usuaris presents a l'array users i el calendari passat per paràmetre. El booleà 'add' determina si la operació inicialitza (true) o esborra (false) el comptador.



Operació increase_participation (token, user, calendar):

Aquesta operació únicament incrementa en una unitat el valor de la participació (participation) per l'usuari i calendari especificats. És la operació que es crida a les activitats que es considera que contribueixen a augmentar la activitat dins el context d'un calendari compartit i aleshores "guanyen" influència a les votacions per l'usuari. En aquests moments i com hem comentat només està implementada a 'setTask' en les seves modalitats de creació i modificació

Operació calculate_quota(weights)

Aquesta operació pren com a paràmetre un conjunt de pesos corresponents als votants i determina una quota mínima que s'ha d'aconseguir per passar una moció. En principi la quota mínima està establerta en la meitat dels pesos (meitat sencera, de la manera que sortiria en una divisió euclidiana) més 1. El sistema calcula totes les combinacions de votacions entre els participants per determinar si hi ha al menys una combinació a on el votant amb menys pes (més vulnerable i principal candidat a ser una comparsa) és decisiu per aconseguir una majoria. Si no existeix cap combinació d'aquestes característiques, el sistema estableix la quota mínima per passar la moció en la majoria més petita que s'ha trobat d'entre les combinacions possibles més el pes del més petit. Així ens assegurem de que no existeixen comparses (dummies).

Operació `get_activity(token,participants, idcalendar)`

Aquesta operació extreu el valor de la participació per un array de users i un calendari determinat i els torna en forma d'array indexat per iduser.

Aplicació en Java

Enlloc de la aplicació Android s'ha actualitzat el codi en Java de la API que es comunica amb el servidor SugarCRM perquè actualitzar el codi Android excedeix el temps del que es disposa. En una futura iteració caldrà aplicar aquest nou codi a la façana d'una aplicació Android actualitzada.

La API es comunica mitjançant el protocol REST amb el servidor i envia la informació seguint l'estàndard JSON. La versió 6.5 Community Edition de SugarCRM presenta un bug⁴ no solucionat: Tot i que el format JSON no és ordenat (com és un diccionari clau-valor, l'ordre no és important perquè es pot fer la cerca per clau) l'entry point de REST requereix que les dades en JSON segueixin un ordre determinat. L'aplicació conté unes classes "OrderedJSON" que solucionen aquest problema. Aquest ordre el simularem amb les classes "OrderedJSONObject" i "OrderedJSONArray". Aquestes classes formen un text seguint l'estàndard JSON que assegura aquest ordre.

5. Implementació

Aspectes de codificació:

2 problemes han marcat la codificació d'aquest sistema de votació. El primer ha estat, com ja hem comentat, el problema de que SugarCRM només accepta la comunicació mitjançant JSON ordenat. Les proves per aconseguir una comunicació RESTful - JSON amb Sugar de manera estable van portar molt de temps. Hi ha molt poca literatura al respecte. Això va determinar que al codi Java s'introduïssin les 2 classes noves i es revisés el codi sencer de l'API perquè amb les classes estàndard no funcionava de manera estable (únicament quan per casualitat la ordenació era la correcta).

El segon problema ha estat la encapsulació del codi php. Al codi de la classe que s'usa per expandir les funcionalitats de l'entry point REST de SugarCRM, anomenada `SweetSyncaWebServiceImpl.php` no es fa un ús molt extens d'aquest principi. El codi està molt ben escrit i comentat però té un problema: moltes accions, com enregistrar una nova tasca, canvis a un calendari o moltes d'altres no s'executen a una funció que es crida a l'efecte i que encapsula la funcionalitat, sinó que s'executen a diferents punts del codi segons hi hagi més participants o no, hi hagi pending resolutions, se'n creïn de noves,... Això complica molt per exemple augmentar l'Activity Tracker cada cop que un usuari enregistra una nova tasca o fa una acció, perquè aquella acció pot passar a bastants de punts diferents del codi i tots s'han de modificar. En la meua opinió, això fa el codi més difícil de mantenir. Aquesta és la raó per la qual només s'ha

⁴ Veure <https://stackoverflow.com/questions/18863849/invalid-login-for-sugarcrm-over-rest>

implementat la funcionalitat a una petita part dels casos d'ús. S'ha volgut fer un model reduït per demostrar la funcionalitat amb la intenció d'ampliar-lo en un futur a totes les activitats que mereixin "premiar" amb més influència l'usuari.

Desplegament

Com s'ha hagut de modificar el servidor SugarCRM per afegir-hi les noves funcionalitats, no s'ha pogut usar la còpia ja desplegada al RDLab de la UPC que encara està actiu. Enlloc d'això s'ha optat per crear una còpia sobre una màquina virtual OpenSuSE Linux 42.3 Leap dins un Mac. La decisió de instal·lar el servidor SugarCRM a una VM Linux ha vingut perquè el Sistema SugarCRM ha estat lligat a LAMP (l'ús conjunt de Linux, Apache2 web server, MariaDB, i PHP) des dels seus inicis i la instal·lació era molt més estable que sobre el propi Mac.

El client/API en Java ha estat desenvolupat a un IDE Eclipse dins l'entorn MacOS, tot i que el desenvolupament en Java fa que sigui portable a qualsevol plataforma.

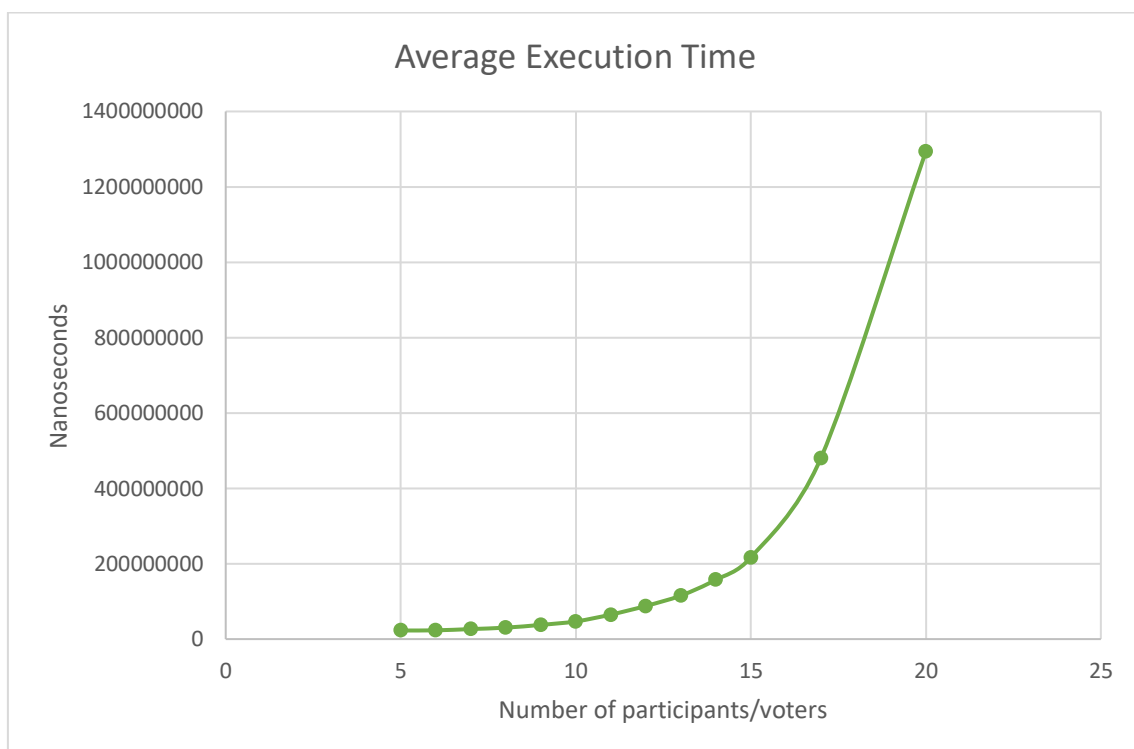
6. Experimentació

Eficiència del càlcul de la quota envers un nombre variable de votants

El càlcul de la quota mínima per a n votants que determina si una moció en la forma de pending resolution s'aprova o no depèn de calcular totes les combinacions de qualsevol nombre de participants des de 1 fins a n . Aquest càlcul té una complexitat exponencial $O(n)=2^n$. La possibilitat que aquests càlculs siguin computacionalment molt costosos s'ha de tenir en compte per evitar problemes majors a la pròpia màquina.

Es per això que s'ha fet un càlcul del cost temporal, com a proxy del cost computacional total, de l'execució d'aquesta aplicació en Java usant un nombre variable de votants. S'ha executat la funció del càlcul de quota 5 cops per cada nombre de votants i se n'ha calculat la mitjana. En tots els casos s'introduïa una combinació de pesos que es sabia que no produiria cap coalició guanyadora amb el votant de pes més petit, aleshores obligàvem a la funció a calcular totes les combinacions de números (la funció s'atura si troba una coalició d'aquestes característiques i deixa de calcular les combinacions restants). El resultats han estat els següents:

# Players	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
5	22032111	22430622	22500902	23948163	23279283	22838216
6	24999903	21661415	23600859	23595439	22576099	23286743
7	27476446	25815414	27568393	26942916	25370312	26634696
8	28724759	32261370	30683728	29625319	30937185	30446472
9	34873616	36703787	38118409	42078655	36760113	37706916
10	49993899	48510704	46280896	45135755	44583065	46900864
11	67871197	66757994	64727150	63685469	62169632	65042288
12	88931182	90050757	83129180	83743316	93564028	87883693
13	108148576	118023916	116291898	117406295	117831164	115540370
14	147369037	177617606	154497077	150957565	156797704	157447798
15	231342219	221471585	206640838	212124551	211711917	216658222
17	503014567	496644671	472024250	470612493	457285289	479916254
20	1268881876	1245372877	1378218149	1326194760	1251004506	1293934434



Com es veu, el cost es dispara a partir de 15 membres i dobla el seu valor entre 12 i 15. És per això que s'ha incorporat un límit superior de 12 a la funció. Aquests són els resultats de la experimentació amb la aplicació en Java. Tanmateix, aquesta no és la funció que es crida a l'entorn SugarCRM, sinó la

funció `calculate_quota` incorporada al codi PHP de la extensió. Sent operacions funcionalment idèntiques és d'esperar com a mínim que si bé una d'elles sigui més eficient que l'altra, la complexitat asimptòtica i la relació entre la eficiència dels diferents punts de la corba no presentin gaires variacions.

Aquest límit superior de 12 representa un nombre prou gran de votants com per incloure la majoria d'equips que podrien usar un sistema d'aquestes característiques, pel que la utilitat del sistema de votació no es veu compromesa. D'altra banda, cal pensar en la alternativa si el nombre de votants supera aquesta quantitat. La opció més senzilla, i la que s'ha adoptat al codi, és que es calculi una majoria qualificada simple. D'aquesta manera s'abandona la protecció dels votants amb pesos més petits però amb un nombre gran de votants les coalicions augmenten exponencialment i també les possibilitats de que els vots petits tinguin importància. Una altra possibilitat és usar regles heurístiques. Aquí hi ha diferents possibilitats, totes imperfectes, que fan un número reduït de càlculs combinant els votants més grans i més petits. Els resultats obtinguts pareixen indicar que no milloren gaire la detecció de comparses o *dummies*, i aleshores no s'han incorporat.

Eficiència del càlcul segons el tipus de votació:

Hi ha 2 tipus de votacions previstes al sistema. A la primera, implementada per la funció `accept_pa`, s'accepta o no un canvi a un calendari o tasca. La segona, implementada a `vote_meeting`, s'escull un canvi entre un nombre d'opcions. A aquesta segona, per raons de temps, no s'ha implementat el sistema de votació qualificada però posa un problema addicional al nou sistema: el fet que hi hagi varies opcions i no només una resposta si/no dificulta la implantació d'un sistema que té tendència a augmentar la quota i per tant dificulta l'aconseguir majories qualificades. Aquest problema es pot tractar de diferents maneres i una d'elles és la que usa el codi: En cas de no aconseguir una majoria en cap sentit és repeteix la votació. Una altra solució seria incorporar rondes de votació: a cada ronda, si cap opció passa la moció, la opció menys votada és descartada. Una tercera seria determinar el guanyador per majoria simple qualificada. En aquest cas la solució passa per un equilibri entre un sistema protector de minories, i l'augment de complexitat que això comporta.

7. Instruccions per instal·lació

La instal·lació consta de les següents parts:

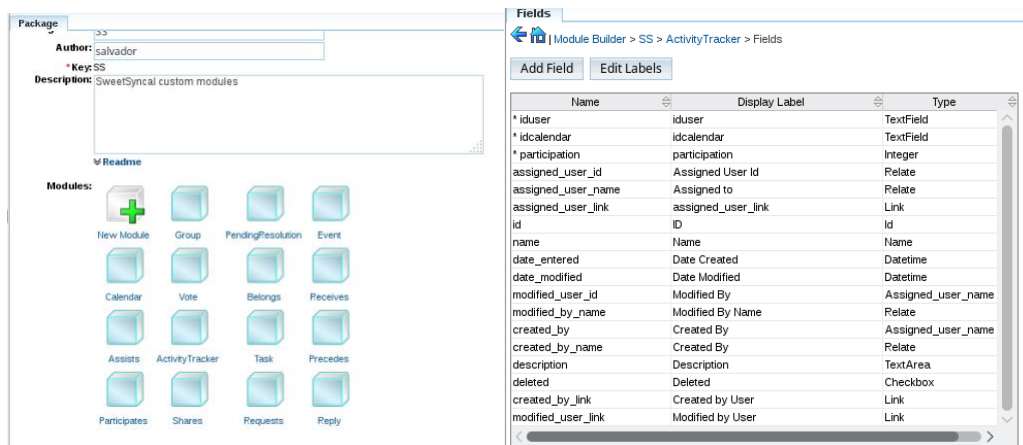
- Un projecte Java 'New SugarCRMClient' que recull la classe que implementa l'API de comunicació amb el servidor SugarCRM, a més d'altres classes auxiliars.
- A aquest projecte Java s'inclou la classe 'WeighedVotingScenario' que calcula la quota mínima per tenir un sistema de votacions sense 'dictadors' ni 'comparses' i per tant fomenti la participació i el treball en equip. Aquesta classe s'usa a l'efecte de fer demostracions pràctiques de com funciona la votació. En realitat el mateix sistema està implantat al model remot en PHP.
- D'altra banda, també s'inclou un paquet de tests que s'usen per les demostracions.
- L'arxiu modificat SweetSyncaWebServiceImpl.php que inclou les següents modificacions sobre l'arxiu original de Palou:
 - La inicialització de l'*ActivityTracker* quan s'afegeix un grup a un calendari o quan es modifiquen els grups (funció 'initialize_AT').
 - La funció 'increase_participation' que incrementa el comptador de participacions cada cop que un usuari fa una activitat que es 'premia' amb participació.
 - La funció 'accept_pa' recull un vot relacionat amb un pending approval determinat, i, si s'ha arribat a una majoria, esborra el PA i implementa els canvis. Aquesta funció s'ha modificat perquè enlloc d'una majoria simple calculi, si té els elements necessaris, la quota mínima i determini si s'ha arribat a una majoria qualificada.
 - La funció get_activity que extreu el comptador d'activitat d'un conjunt d'usuaris per un calendari determinat
 - Modificacions a altres funcions per incorporar les funcions anteriors

Les passes per instal·lar aquest conjunt són:

1. Importar el projecte Java 'New SugarCRMClient' a Eclipse.
2. Instal·lar a la pròpia màquina o a una altra una còpia de SugarCRM CE 6.5.26 seguint les opcions estàndard indicades a la web de suport.
3. Usant el Module Builder de SugarCRM, afegir el paquet SS i dins ell el mòdul 'ActivityTracker' a més dels mòduls del model remot descrits per Palou a la seva memòria. És important que els noms dels mòduls i els noms dels camps coincideixin exactament. Els camps del mòdul ActivityTracker són:

Taula SS_ActivityTracker

Columna	Tipus	Descripció
id	TEXT	Identificador comptador
iduser	TEXT	Identificador de l'usuari
idcalendar	TEXT	Identificador calendari
participation	INTEGER	Comptador activitat.



4. Incorporar l'arxiu *SweetSyncaWebServiceImpl.php* a la carpeta */sugarcrm/service/v4_1*
5. Dins aquesta mateixa carpeta obrir l'arxiu *rest.php* i modificar les següents línies:

```
require_once(' Sugar WebServiceImpl v4_1.php');
$webserviceimpl_dass = ' Sugar WebServiceImpl v4_1';
```

per

```
require_once(' SweetSyncaWebServiceImpl.php');
$webserviceimpl_dass = ' SweetSyncaWebServiceImpl';
```

6. Reiniciar el servidor apache.
7. Per que els tests funcionin, cal entrar com a administrador a SugarCRM i, amb la opció "User Management" afegir 6 usuaris normals al sistema. Si aquests usuaris no tenen l'username i contrasenya descrits als tests s'hauran de modificar. Si no es vol canviar els usuaris codificats als tests cal crear 6 usuaris amb els username: 'salvasalas', 'taguilo', 'jdoe', 'jking', 'mpowers' i 'tgriffin', tots ells amb la contrasenya 'cont3065'. Recordem que quan es crea un usuari nou Sugar envia un e-mail de confirmació a la direcció especificada amb una contrasenya temporal. Cal canviar-la per aquesta.
8. Finalment als tests que es comuniquin amb el servidor Sugar, cal substituir la URL 192.168.1.46 per la URL del servidor Apache a on està allotjat SugarCRM.

És important indicar que s'ha deshabilitat la funció "notify_users" i les funcions per incorporar o desincorporar aparells amb GCM perquè usa un mòdul no present ("SS_GCM"). Al no usar, per ara, l'aplicació d'Android, aquestes funcions deixen de ser necessàries. Com no podem comptar amb el Google Cloud Messaging hem d'anar a cercar les notificacions, PAs, i alertes amb una crida que ens retorni els beans.

8. Conclusions

Objectius aconseguits:

Aquest treball ha pogut establir un sistema de votació qualificada per la aplicació desenvolupada per Seguí i Palou.

S'ha mantingut el servidor SugarCRM com a repositori central de dades i de la lògica de la aplicació.

S'han fet proves sobre la escalabilitat d'aquest nou sistema de votació i les seves alternatives cas de que sigui massa costós.

S'ha iniciat un model de learning analytics, que, si bé molt primitiu, es pot fer més sofisticat per incorporar altres aspectes de la participació d'un usuari al sistema.

En definitiva, s'ha mostrat un model més sofisticat de votació que dona més poder als usuaris que més s'involucren sense deixar de banda a la resta.

Objectius no aconseguits.

L'objectiu original d'aquest treball era desenvolupar una aplicació Android completa que usés la funcionalitat descrita en aquest treball. Tot i que la major part del desenvolupament s'ha fet al servidor SugarCRM, no he aconseguit actualitzar la aplicació. La raó és la falta de experiència en desenvolupar per aquesta plataforma. En principi la aplicació de Palou era plenament funcional, però la versió d'Android que usava era anterior a la darrera que es donava suport amb Android Studio. Després de mesos intentant-ho vaig arribar a la conclusió que no aconseguiria aquesta fita que a més em desviava de l'objectiu original consistent en desenvolupar el sistema de votació qualificada.

L'altre objectiu no aconseguit és implantar el sistema de votació qualificada a tot el sistema. En la actualitat només funciona a una part. La idea era tenir un "working model" que mostres el camí a seguir per la resta de la aplicació. La falta de temps provocada pels mesos intentant sense èxit actualitzar la aplicació em va deixar sense masses opcions i vaig optar per fer una prova de concepte, ampliable d'una manera senzilla a la resta del codi. El treball d'ampliació és senzill, es tracta d'introduir les funcions ja desenvolupades a la resta d'accions, però, com hem comentat, el codi és prou complex que una mateixa acció s'executa a diferents punts d'aquest i per això s'ha optat per contenir el treball a un mínim sense errors.

Futures ampliacions.

Les futures iteracions del sistema passen en primer lloc per acabar els objectius no aconseguits en aquest treball:

- Ampliar el sistema de votació qualificada a la resta d'accions del codi
- Actualitzar la aplicació d'Android i incloure-hi la API ja actualitzada

En segon lloc cal desenvolupar un sistema de learning analytics més sofisticat. A l'actualitat el sistema només manté un comptador d'activitat que augmenta cada cop que l'usuari realitza una determinada acció (crear o modificar una tasca). Aquest comptador és necessari per calcular els pesos dels usuaris a cada votació, però els càlculs per incrementar aquest nombre és el que pot ser molt més complex. La base de dades pot tenir en compte moltes coses per calcular la activitat: permanència del login, respostes a altres membres, comunicacions, grau i temps de compleció de les tasques... la idea és tenir un sistema de learning analytics que esbrini realment que fa d'un usuari una persona activa o eficient i aquests càlculs es traslladin al comptador d'activitat.

Finalment, i tot i que ara no sembla un problema greu, en un futur caldrà solucionar la escalabilitat del sistema de votació qualificada d'una manera que segueixi protegint votants minoritaris. Potser la solució passarà per fer diferents categories de votants amb diferents status però sense un nombre de vots: Els usuaris més actius podrien tenir poder de aturar una moció (es necessari el vot de tots ells per passar una moció) i els votants menys actius hauran de fer coalicions entre ells. En qualsevol cas, la filosofia seria avançar en eficiència sense perdre la protecció dels usuaris menys actius que garanteix tothom un incentiu a participar.

Valoració personal

Els primers mesos de feina d'aquest treball em varen suposar molta frustració. Convençut de que havia de, i podia, assolir els objectius, el temps passava i no avançava. Vaig començar pensant que tindria una aplicació Android funcional i que amb pocs canvis podria seguir amb la part important de la operació que es desenvolupava al propi servidor. En realitat em vaig passar massa temps intentant fer funcionar la aplicació i, en retrospectiva, hauria d'haver canviat els objectius molt abans. Això és la gran lliçó que he après d'aquesta feina. Quan es veu que els objectius no seran aconseguits, plantejar-se uns objectius més assequibles i, a partir d'aquest punt, treballar de manera incremental. Aquest canvi el vaig fer a finals de novembre, quan podria haver fet el mateix un mes antes o inclús abans. Tot i així, quan el tutor em va proposar fer el model en Java, vaig pensar que encara podia fer el sistema una mica més real i usar un servidor SugarCRM i actualitzar únicament la part de la aplicació que era la API. D'aquesta manera podria simular les comunicacions entre dispositiu mòbil i servidor. Així ho vaig fer i al final he pogut tenir una prova de concepte del nou sistema de votació a l'entorn original. En definitiva, s'han aconseguit una part dels objectius tot i que el calendari s'ha endarrerit dos mesos, i l'objectiu central que era demostrar el nou sistema de votació s'ha assolit.

9. Bibliografia

- Palou, Daniel; Xhafa, Fatos (2013). **Implementació d'un algorisme de sincronització de calendari d'esdeveniments en un sistema Android.** Treball de Final de Grau presentat al Departament de Llenguatges i Sistemes Informàtics de la Universitat Politècnica de Catalunya. Consultat a <http://upcommons.upc.edu/handle/2099.1/16937>
- Kirk, P. (2017). What new technologies mean for the future of office space. *National Real Estate Investor*, Retrieved from <https://search.proquest.com/docview/1902622138?accountid=15299>
- Tannenbaum, Peter. *Excursions in Modern Mathematics*. 6th ed. Upper Saddle River: Prentice Hall, 2006.

10. Annexos

S'inclou com a annex la memòria del projecte original desenvolupat per Palou que s'ha usat en gran mesura per determinar quines funcionalitats del projecte calia modificar.