

**Títol:** Implementació d'un algorisme de sincronització de calendari d'esdeveniments en un sistema Android.

**Autor:** Daniel Palou Sala

**Data:** 21 de gener de 2013

**Director/Ponent:** Fatos Xhafa Xhafa

**Departament:** Llenguatges i Sistemes Informàtics

**Titulació:** Enginyeria informàtica (2003)

**Centre:** Facultat d'Informàtica de Barcelona (FIB)

**Universitat:** Universitat Politècnica de Catalunya (UPC)  
BarcelonaTech

---

## **DADES DEL PROJECTE**

*Títol del projecte:* Implementació d'un algorisme de sincronització de calendari d'esdeveniments en un sistema Android.

*Nom de l'estudiant:* Daniel Palou Sala

*Titulació:* Enginyeria informàtica (2003)

*Crèdits:* 37,5

*Director/Ponent:* Fatos Xhafa Xhafa

*Departament:* Llenguatge i Sistemes Informàtics

---

## **MEMBRES DEL TRIBUNAL (nom i signatura)**

*President:* Josefina López Herrera

*Vocal:* Cristina Barrado Muxi

*Secretari/a:* Fatos Xhafa Xhafa

---

## **QUALIFICACIÓ**

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:* 21 de gener de 2013

---

## ÍNDEX

Agraïments .....	15
<b>Descripció del projecte .....</b>	<b>16</b>
1. Context i motivació del projecte .....	17
2. Descripció del projecte .....	17
3. Objectius .....	18
4. Decisió de la tecnologia .....	18
5. Requisits no funcionals .....	19
6. Anàlisi de riscos .....	20
7. Conceptes importants per al projecte .....	22
7.1 Sincronització .....	22
7.2 Treball en grup .....	24
7.3 Peer-to-Peer .....	25
7.4 Dispositius mòbils .....	25
<b>Especificació .....</b>	<b>27</b>
1. Especificació casos d'ús .....	28
1.1 Gestió de comptes .....	28
1.2 Gestió de calendaris .....	29
1.3 Gestió de tasques .....	32
1.4 Sincronització .....	36
1.5 Gestió d'esdeveniments, votacions i preferències .....	40
1.6 Gestió de grups .....	43
2. Definició dels actors .....	48
3. Diagrames de casos d'ús .....	50
3.1 Gestió de comptes i usuaris .....	50
3.2 Gestió de calendaris .....	50
3.3 Gestió de tasques .....	51
3.4 Sincronització .....	51
3.5 Gestió d'esdeveniments, votacions i preferències .....	52
3.6 Gestió de grups .....	53
4. Diagrames de seqüència .....	54
4.1 Gestió de comptes .....	54
4.2 Gestió de calendaris .....	55
4.3 Gestió de tasques .....	57

4.4 Sincronització .....	59
4.5 Gestió d'esdeveniments, votacions i preferències .....	61
4.6 Gestió de grups .....	63
5. Contractes de les operacions .....	67
5.1 Gestió de comptes .....	67
5.2 Gestió de calendaris.....	68
5.3 Gestió de tasques .....	72
5.4 Sincronització .....	76
5.5 Gestió d'esdeveniments, votacions i preferències .....	77
5.6 Gestió de grups .....	81
6. Model conceptual .....	86
6.1 Diagrama de classes.....	86
6.2 Atributs de les classes.....	87
6.3 Diagrama de classes complet.....	91
<b>Disseny .....</b>	<b>93</b>
1. Patró arquitectònic .....	94
1.1 Vista.....	95
1.2 ViewModel.....	96
1.3 Model .....	96
2. Diagrames de seqüència .....	110
2.1 Gestió de comptes.....	111
2.2 Gestió de calendaris.....	115
2.3 Gestió de tasques .....	125
2.4 Sincronització .....	133
2.5 Gestió d'esdeveniments, votacions i preferències .....	141
2.6 Gestió de grups .....	151
3. Disseny de navegació .....	161
<b>Implementació .....</b>	<b>163</b>
1. Aspectes importants .....	164
2. Desplegament a la infraestructura RDLab .....	164
3. Principals problemes trobats .....	165
4. Proves i correcció d'errors .....	170
<b>Experimentació .....</b>	<b>171</b>
1. Client-Servidor vs Peer-to-peer .....	172

1.1	Estudi previ .....	173
1.2	Experimentació .....	180
2.	Relationships vs Modules .....	183
2.1	Estudi previ .....	184
2.2	Experimentació .....	187
3.	Bateria: SyncAdapter vs GCM .....	190
3.1	Estudi previ .....	191
3.2	Experimentació .....	194
4.	Desar només tasques vs desar-ho tot .....	197
4.1	Estudi previ .....	198
4.2	Experimentació .....	201
<b>Planificació i costos .....</b>		<b>203</b>
1.	Planificació temporal .....	204
1.1	Planificació inicial .....	204
1.2	Planificació final .....	206
2.	Anàlisi econòmic .....	206
2.1	Costos de hardware .....	207
2.2	Costos de software .....	207
2.3	Costos fixes .....	207
2.4	Costos de personal .....	208
2.5	Cost total del projecte .....	209
<b>Manual d'instal·lació i d'usuari .....</b>		<b>210</b>
1.	Introducció .....	211
2.	Instal·lació .....	211
2.1	Instal·lació de l'aplicació mòbil .....	211
2.2	Instal·lació del servidor .....	211
3.	Accés a l'aplicació .....	212
4.	Gestió de tasques .....	213
4.1	Llistat de tasques .....	213
4.2	Filtrar tasques .....	214
4.3	Informació tasca .....	215
4.4	Crear/modificar tasca .....	215
4.5	Afegir participants i Afegir precedents .....	216
5.	Gestió de calendaris .....	217

5.1 Llistat de calendaris .....	217
5.2 Informació calendari .....	217
5.3 Crear/modificar calendari .....	218
5.4 Afegir participants .....	218
6. Gestió de grups.....	219
6.1 Llistat de grups.....	219
6.2 Cercar grup .....	219
6.3 Informació grup.....	220
6.4 Crear grup .....	221
6.5 Afegir membres.....	221
6.6 Llistat de sol·licituds .....	222
7. Gestió d'esdeveniments, votacions i preferències.....	222
7.1 Llistat d'esdeveniments.....	222
7.2 Llistat de pending approvals.....	222
7.3 Informació pending approval.....	223
7.4 Llistat de meetings.....	223
7.5 Informació meeting.....	223
7.6 Estadístiques d'usuari.....	224
7.7 Preferències .....	224
7.8 Altres opcions del menú principal.....	224
<b>Conclusions.....</b>	<b>225</b>
1. Objectius aconseguits .....	226
2. Futures ampliacions.....	226
3. Valoració personal .....	227
<b>Annex.....</b>	<b>228</b>
1. Resultats detallats de l'experimentació .....	229
1.1 Client-servidor vs P2P .....	229
1.2 Relationships vs Modules .....	229
1.3 Desar només tasques vs desar-ho tot .....	231
2. Bibliografia .....	232

## ÍNDIX DE TAULES

<i>Taula 1: Risc 1</i> .....	21
<i>Taula 2: Risc 2</i> .....	21
<i>Taula 3: Risc 3</i> .....	21
<i>Taula 4: Risc 4</i> .....	22
<i>Taula 5: Risc 5</i> .....	22
<i>Taula 6: Cas d'ús Login</i> .....	28
<i>Taula 7: Cas d'ús Logout</i> .....	29
<i>Taula 8: Cas d'ús Veure estadístiques</i> .....	29
<i>Taula 9: Cas d'ús Crear calendari</i> .....	29
<i>Taula 10: Cas d'ús Modificar calendari</i> .....	30
<i>Taula 11: Cas d'ús Eliminar calendari</i> .....	30
<i>Taula 12: Cas d'ús Veure informació calendari</i> .....	31
<i>Taula 13: Cas d'ús Afegir participant al calendari</i> .....	31
<i>Taula 14: Cas d'ús Eliminar participant del calendari</i> .....	32
<i>Taula 15: Cas d'ús Eliminar participant del calendari</i> .....	32
<i>Taula 16: Cas d'ús Crear tasca</i> .....	32
<i>Taula 17: Cas d'ús Modificar tasca</i> .....	33
<i>Taula 18: Cas d'ús Eliminar tasca</i> .....	33
<i>Taula 19: Cas d'ús Veure informació tasca</i> .....	33
<i>Taula 20: Cas d'ús Afegir participant a una tasca</i> .....	34
<i>Taula 21: Cas d'ús Eliminar participant d'una tasca</i> .....	34
<i>Taula 22: Cas d'ús Afegir precedent a una tasca</i> .....	34
<i>Taula 23: Cas d'ús Eliminar precedent d'una tasca</i> .....	35
<i>Taula 24: Cas d'ús Llistar tasques</i> .....	35
<i>Taula 25: Cas d'ús Filtrar llistat de tasques</i> .....	35
<i>Taula 26: Cas d'ús Recordar tasca</i> .....	36
<i>Taula 27: Cas d'ús Sincronitzar calendaris</i> .....	37
<i>Taula 28: Cas d'ús Sincronització total</i> .....	38
<i>Taula 29: Cas d'ús Enviar tasques</i> .....	39
<i>Taula 30: Cas d'ús Demanar canvis</i> .....	39
<i>Taula 31: Cas d'ús Llistar esdeveniments</i> .....	40
<i>Taula 32: Cas d'ús Llistar meetings (votacions)</i> .....	40
<i>Taula 33: Cas d'ús Veure informació meeting</i> .....	40
<i>Taula 34: Cas d'ús Votar en un meeting</i> .....	41
<i>Taula 35: Cas d'ús Eliminar vot d'un meeting</i> .....	41
<i>Taula 36: Cas d'ús Llistar pending approvals</i> .....	42
<i>Taula 37: Cas d'ús Veure informació pending approval</i> .....	42
<i>Taula 38: Cas d'ús Acceptar/Rebutjar pending approval</i> .....	42
<i>Taula 39: Cas d'ús Veure preferències</i> .....	43
<i>Taula 40: Cas d'ús Modificar prederències</i> .....	43
<i>Taula 41: Cas d'ús Crear grup</i> .....	44
<i>Taula 42: Canviar nom del grup</i> .....	44
<i>Taula 43: Afegir participant a un grup</i> .....	44

Taula 44: Cas d'ús Eliminar participant d'un grup .....	45
Taula 45: Cas d'ús Sortir d'un grup.....	45
Taula 46: Cas d'ús Veure informació d'un grup .....	45
Taula 47: Cas d'ús Llistar grups .....	46
Taula 48: Cas d'ús Nomenar administrador .....	46
Taula 49: Cas d'ús Cercar grup .....	46
Taula 50: Cas d'ús Enviar/Cancel·lar sol·licitud .....	47
Taula 51: Cas d'ús Llistar sol·licituds .....	47
Taula 52: Cas d'ús Acceptar/Rebutjar sol·licitud .....	47
Taula 53: Operació obrirAplicacio .....	67
Taula 54: Operació validar .....	67
Taula 55: Operació confirmarLogin .....	67
Taula 56: Operació logout .....	68
Taula 57: Operació confirmarLogout.....	68
Taula 58: Operació veureEstadistiques .....	68
Taula 59: Operació altaCalendari .....	68
Taula 60: Operació crearCalendari .....	69
Taula 61: Operació modificarCalendari .....	69
Taula 62: Operació modificarDades .....	69
Taula 63: Operació eliminarCalendari .....	70
Taula 64: Operació confirmarEliminarCalendari .....	70
Taula 65: Operació mostrarInfoCalendari .....	70
Taula 66: Operació afegirParticipantCalendari .....	70
Taula 67: Operació enviarParticipantsCalendaris.....	71
Taula 68: Operació eliminarParticipantCalendari .....	71
Taula 69: Operació confirmarElimPartCal .....	71
Taula 70: Operació llistarCalendari .....	72
Taula 71: Operació altaTasca .....	72
Taula 72: Operació crearTasca .....	72
Taula 73: Operació modificarTasca .....	73
Taula 74: Operació modificarDadesTasca .....	73
Taula 75: Operació eliminarTasca .....	73
Taula 76: Operació confirmarElimTasca.....	73
Taula 77: Operació mostrarInfoTasca .....	73
Taula 78: Operació afegirParticipantTasca .....	74
Taula 79: Operació enviarParticipantsTasca .....	74
Taula 80: Operació eliminarParticipantTasca.....	74
Taula 81: Operació confirmarElimPartTasca .....	74
Taula 82: Operació afegirPrecedentTasca .....	75
Taula 83: Operació enviarPrecedentsTasca.....	75
Taula 84: Operació eliminarPrecedentTasca .....	75
Taula 85: Operació confirmarElimPartTasca .....	75
Taula 86: Operació llistarTasques .....	76
Taula 87: Operació filtrarLlistat.....	76
Taula 88: Operació filtrar .....	76



<i>Taula 89: Operació enviarNotificacio</i>	76
<i>Taula 90: Operació demanarCanvis</i>	77
<i>Taula 91: Operació enviarCanvis</i>	77
<i>Taula 92: Operació sincronitzacioCompleta</i>	77
<i>Taula 93: Operació llistarEsdeveniments</i>	77
<i>Taula 94: Operació llistarMeetings</i>	78
<i>Taula 95: Operació veureInfoMeeting</i>	78
<i>Taula 96: Operació votar</i>	78
<i>Taula 97: Operació confirmarVot</i>	78
<i>Taula 98: Operació eliminarVot</i>	79
<i>Taula 99: Operació confirmarEliminarVot</i>	79
<i>Taula 100: Operació llistarPendingApprovals</i>	79
<i>Taula 101: Operació veureInfoPendingApproval</i>	79
<i>Taula 102: Operació respondrePendingApproval</i>	80
<i>Taula 103: Operació confirmarResposta</i>	80
<i>Taula 104: VeurePreferencies</i>	80
<i>Taula 105: Operació modificarPreferencies</i>	80
<i>Taula 106: Operació modificarDadesPref</i>	81
<i>Taula 107: Operació altaGrup</i>	81
<i>Taula 108: Operació crearGrup</i>	81
<i>Taula 109: Operació modificarNomGrup</i>	81
<i>Taula 110: Operació canviarNom</i>	82
<i>Taula 111: Operació veureInfogGrup</i>	82
<i>Taula 112: Operació afegirParticipantGrup</i>	82
<i>Taula 113: Operació enviarParticipantsGrups</i>	82
<i>Taula 114: Operació eliminarParticipantGrup</i>	83
<i>Taula 115: Operació confirmarElimPartGrup</i>	83
<i>Taula 116: Operació sortirGrup</i>	83
<i>Taula 117: Operació confirmarSortirGrup</i>	83
<i>Taula 118: Operació llistarGrups</i>	84
<i>Taula 119: Operació nomenarAdministrador</i>	84
<i>Taula 120: Operació confirmarNomenar</i>	84
<i>Taula 121: Operació cercarGrup</i>	84
<i>Taula 122: Operació cerca</i>	85
<i>Taula 123: Operació enviarOCancelarSolicitud</i>	85
<i>Taula 124: Operació llistarSolicituds</i>	85
<i>Taula 125: Operació respondreSolicitud</i>	85
<i>Taula 126: Taula User local</i>	98
<i>Taula 127: Taula Group local</i>	98
<i>Taula 128: Taula Calendar local</i>	98
<i>Taula 129: Taula Task local</i>	99
<i>Taula 130: Taula Event local</i>	99
<i>Taula 131: Taula Pending Approval local</i>	100
<i>Taula 132: Taula Meeting local</i>	101
<i>Taula 133: Taula Vote local</i>	101

<i>Taula 134: Taula Belongs local</i> .....	101
<i>Taula 135: Taula Requests local</i> .....	101
<i>Taula 136: Taula Shares local</i> .....	102
<i>Taula 137: Taula Participates local</i> .....	102
<i>Taula 138: Taula Assists local</i> .....	102
<i>Taula 139: Taula Precedes local</i> .....	102
<i>Taula 140: Mòdul Users</i> .....	103
<i>Taula 141: Mòdul SS_Group</i> .....	103
<i>Taula 142: Mòdul SS_Calendar</i> .....	104
<i>Taula 143: Mòdul SS_Task</i> .....	104
<i>Taula 144: Mòdul SS_Event</i> .....	105
<i>Taula 145: Mòdul SS_PendingResolution</i> .....	106
<i>Taula 146: Mòdul SS_Vote</i> .....	106
<i>Taula 147: Mòdul SS_Belongs</i> .....	107
<i>Taula 148: Mòdul SS_Requests</i> .....	107
<i>Taula 149: Mòdul SS_Shares</i> .....	107
<i>Taula 150: Mòdul SS_Participates</i> .....	108
<i>Taula 151: Mòdul SS_Assists</i> .....	108
<i>Taula 152: Mòdul SS_Precedes</i> .....	108
<i>Taula 153: Mòdul SS_Reply</i> .....	109
<i>Taula 154: Mòdul SS_Receive</i> .....	109
<i>Taula 155: Avantatges i inconvenients mòduls i relationships</i> .....	168
<i>Taula 156: Resum experiment 1</i> .....	173
<i>Taula 157: Estudi previ experiment 1</i> .....	180
<i>Taula 158: Resultats experiment 1</i> .....	181
<i>Taula 159: Resum experiment 2</i> .....	184
<i>Taula 160: Estudi previ experiment 2</i> .....	186
<i>Taula 161: Resultats experiment 2</i> .....	187
<i>Taula 162: Resum experiment 3</i> .....	190
<i>Taula 163: Estudi previ experiment 3</i> .....	193
<i>Taula 164: Resultats experiment 3</i> .....	194
<i>Taula 165: Resum experiment 4</i> .....	198
<i>Taula 166: Estudi previ experiment 4</i> .....	200
<i>Taula 167: Resultats experiment 4</i> .....	201
<i>Taula 168: Planificació final</i> .....	206
<i>Taula 169: Costos de hardware</i> .....	207
<i>Taula 170: Costos de software</i> .....	207
<i>Taula 171: Costos fixes</i> .....	208
<i>Taula 172: Hores personal</i> .....	208
<i>Taula 173: Costos de personal</i> .....	208
<i>Taula 174: Cost total del projecte</i> .....	209
<i>Taula 175: Resultats detallats experiment 1</i> .....	229
<i>Taula 176: Resultats detallats temps total experiment 2</i> .....	229
<i>Taula 177: Resultats detallats temps rèpliques 5, 6 i 7 experiment 2</i> .....	230
<i>Taula 178: Resultats detallats temps experiment 4</i> .....	231

## ÍNDEX DE FIGURES

<i>Figura 1: Jerarquia d'actors</i> .....	48
<i>Figura 2: Actor Usuari</i> .....	48
<i>Figura 3: Actor Usuari no loguejat</i> .....	48
<i>Figura 4: Actor Usuari loguejat</i> .....	48
<i>Figura 5: Actor Sistema local</i> .....	49
<i>Figura 6: Actor Sistema remot</i> .....	49
<i>Figura 7: Diagrama casos d'ús comptes</i> .....	50
<i>Figura 8: Diagrama casos d'ús calendaris</i> .....	50
<i>Figura 9: Diagrama casos d'ús tasques</i> .....	51
<i>Figura 10: Diagrama casos d'ús sincronització</i> .....	51
<i>Figura 11: Diagrama casos d'ús sincronització corregit</i> .....	52
<i>Figura 12: Diagrama casos d'ús esdeveniments i votacions</i> .....	52
<i>Figura 13: Diagrama casos d'ús grups</i> .....	53
<i>Figura 14: Diagrama seqüència especificació login</i> .....	54
<i>Figura 15: Diagrama seqüència especificació logout</i> .....	54
<i>Figura 16: Diagrama seqüència especificació veure estadístiques</i> .....	54
<i>Figura 17: Diagrama seqüència especificació crear calendari</i> .....	55
<i>Figura 18: Diagrama seqüència especificació modificar calendari</i> .....	55
<i>Figura 19: Diagrama seqüència especificació eliminar calendari</i> .....	55
<i>Figura 20: Diagrama seqüència especificació info calendari</i> .....	55
<i>Figura 21: Diagrama seqüència especificació afegir participant calendari</i> .....	56
<i>Figura 22: Diagrama seqüència especificació eliminar participant calendari</i> .....	56
<i>Figura 23: Diagrama seqüència especificació llistar calendaris</i> .....	56
<i>Figura 24: Diagrama seqüència especificació crear tasca</i> .....	57
<i>Figura 25: Diagrama seqüència especificació modificar tasca</i> .....	57
<i>Figura 26: Diagrama seqüència especificació eliminar tasca</i> .....	57
<i>Figura 27: Diagrama seqüència especificació info tasca</i> .....	57
<i>Figura 28: Diagrama seqüència especificació afegir participant tasca</i> .....	58
<i>Figura 29: Diagrama seqüència especificació eliminar participant tasca</i> .....	58
<i>Figura 30: Diagrama seqüència especificació afegir precedent tasca</i> .....	58
<i>Figura 31: Diagrama seqüència especificació eliminar precedent tasca</i> .....	58
<i>Figura 32: Diagrama seqüència especificació llistar tasques</i> .....	59
<i>Figura 33: Diagrama seqüència especificació filtrar llistat</i> .....	59
<i>Figura 34: Diagrama seqüència especificació recordar tasca</i> .....	59
<i>Figura 35: Diagrama seqüència especificació sincronitzar calendaris</i> .....	59
<i>Figura 36: Diagrama seqüència especificació sincronització total</i> .....	60
<i>Figura 37: Diagrama seqüència especificació enviar tasques</i> .....	60
<i>Figura 38: Diagrama seqüència especificació demanar canvis</i> .....	60
<i>Figura 39: Diagrama seqüència especificació llistar esdeveniments</i> .....	61
<i>Figura 40: Diagrama seqüència especificació llistar meetings</i> .....	61
<i>Figura 41: Diagrama seqüència especificació info meeting</i> .....	61
<i>Figura 42: Diagrama seqüència especificació votar en un meeting</i> .....	61
<i>Figura 43: Diagrama seqüència especificació eliminar vot meeting</i> .....	62

<i>Figura 44: Diagrama seqüència especificació llistar pending approvals</i> .....	62
<i>Figura 45: Diagrama seqüència especificació info pending approval</i> .....	62
<i>Figura 46: Diagrama seqüència especificació acceptar/rebutjar PA</i> .....	62
<i>Figura 47: Diagrama seqüència especificació veure preferències</i> .....	63
<i>Figura 48: Diagrama seqüència especificació modificar preferències</i> .....	63
<i>Figura 49: Diagrama seqüència especificació crear grup</i> .....	63
<i>Figura 50: Diagrama seqüència especificació modificar nom grup</i> .....	63
<i>Figura 51: Diagrama seqüència especificació afegir participant grup</i> .....	64
<i>Figura 52: Diagrama seqüència especificació eliminar participant grup</i> .....	64
<i>Figura 53: Diagrama seqüència especificació sortir grup</i> .....	64
<i>Figura 54: Diagrama seqüència especificació info grup</i> .....	64
<i>Figura 55: Diagrama seqüència especificació llistar grups</i> .....	65
<i>Figura 56: Diagrama seqüència especificació nomenar administrador</i> .....	65
<i>Figura 57: Diagrama seqüència especificació cercar grup</i> .....	65
<i>Figura 58: Diagrama seqüència especificació enviar/cancel·lar sol·licitud</i> .....	65
<i>Figura 59: Diagrama seqüència especificació llistar sol·licituds</i> .....	66
<i>Figura 60: Diagrama seqüència especificació acceptar/rebutjar sol·licitud</i> .....	66
<i>Figura 61: Model conceptual local sense atributs</i> .....	86
<i>Figura 62: Model conceptual servidor sense atributs</i> .....	86
<i>Figura 63: Classe Usuari</i> .....	87
<i>Figura 64: Classe Grup</i> .....	87
<i>Figura 65: Classe Calendari i subclasses</i> .....	87
<i>Figura 66: Classe Tasca</i> .....	88
<i>Figura 67: Classe Objecte amb esdeveniments</i> .....	88
<i>Figura 68: Classe Esdeveniment</i> .....	88
<i>Figura 69: Classe Alerta</i> .....	89
<i>Figura 70: Classe Resol. Pend.</i> .....	89
<i>Figura 71: Classe Meeting</i> .....	89
<i>Figura 72: Classe Vot</i> .....	90
<i>Figura 73: Classe Pending Approval</i> .....	90
<i>Figura 74: Classe Resposta</i> .....	90
<i>Figura 75: Classe Preferències</i> .....	90
<i>Figura 76: Model conceptual local amb atributs</i> .....	91
<i>Figura 77: Model conceptual servidor amb atributs</i> .....	91
<i>Figura 78: Esquema patró MVVM</i> .....	94
<i>Figura 79: Diagrama seqüència disseny obrirAplicacio</i> .....	111
<i>Figura 80: Diagrama seqüència disseny validar</i> .....	111
<i>Figura 81: Diagrama seqüència disseny confirmarLogin</i> .....	112
<i>Figura 82: Diagrama seqüència disseny logout</i> .....	113
<i>Figura 83: Diagrama seqüència disseny confirmarLogout</i> .....	114
<i>Figura 84: Diagrama seqüència disseny veureEstadistiques</i> .....	115
<i>Figura 85: Diagrama seqüència disseny altaCalendari</i> .....	116
<i>Figura 86: Diagrama seqüència disseny crearCalendari</i> .....	117
<i>Figura 87: Diagrama seqüència disseny modificarCalendari</i> .....	118
<i>Figura 88: Diagrama seqüència disseny modify_calendar</i> .....	119

<i>Figura 89: Diagrama seqüència disseny confirmarEliminarCalendari</i> .....	120
<i>Figura 90: Diagrama seqüència disseny mostrarInfoCalendari</i> .....	121
<i>Figura 91: Diagrama seqüència disseny enviarParticipantsCalendari</i> .....	122
<i>Figura 92: Diagrama seqüència disseny modify_participants_calendar</i> .....	123
<i>Figura 93: Diagrama seqüència disseny confirmarElimPartCal</i> .....	124
<i>Figura 94: Diagrama seqüència disseny llistarCalendaris</i> .....	125
<i>Figura 95: Diagrama seqüència disseny crearTasca</i> .....	126
<i>Figura 96: Diagrama seqüència disseny modificarTasca</i> .....	127
<i>Figura 97: Diagrama seqüència disseny confirmarElimTasca</i> .....	128
<i>Figura 98: Diagrama seqüència disseny mostrarInfoTasca</i> .....	128
<i>Figura 99: Diagrama seqüència disseny enviarParticipantsTasca</i> .....	129
<i>Figura 100: Diagrama seqüència disseny confirmarElimPartTasca</i> .....	129
<i>Figura 101: Diagrama seqüència disseny enviarPrecedentsTasca</i> .....	130
<i>Figura 102: Diagrama seqüència disseny confirmarElimPrecTasca</i> .....	130
<i>Figura 103: Diagrama seqüència disseny llistarTasques</i> .....	131
<i>Figura 104: Diagrama seqüència disseny llistar</i> .....	132
<i>Figura 105: Diagrama seqüència disseny enviarNotificacio</i> .....	133
<i>Figura 106: Diagrama seqüència disseny sincronizacioCompleta</i> .....	134
<i>Figura 107: Diagrama seqüència disseny get_info</i> .....	135
<i>Figura 108: Diagrama seqüència disseny set_tasks</i> .....	136
<i>Figura 109: Diagrama seqüència disseny treat_task</i> .....	137
<i>Figura 110: Diagrama seqüència disseny set_relationships</i> .....	139
<i>Figura 111: Diagrama seqüència disseny treat_relationship</i> .....	140
<i>Figura 112: Diagrama seqüència disseny llistarEsdeveniments</i> .....	141
<i>Figura 113: Diagrama seqüència disseny llistarMeetings</i> .....	141
<i>Figura 114: Diagrama seqüència disseny veureInfoMeeting</i> .....	142
<i>Figura 115: Diagrama seqüència disseny confirmarVot</i> .....	143
<i>Figura 116: Diagrama seqüència disseny vote_meeting</i> .....	144
<i>Figura 117: Diagrama seqüència disseny confirmarEliminarVot</i> .....	145
<i>Figura 118: Diagrama seqüència disseny llistarPendingApprovals</i> .....	146
<i>Figura 119: Diagrama seqüència disseny veureInfopendingApproval</i> .....	146
<i>Figura 120: Diagrama seqüència disseny confirmarResposta</i> .....	147
<i>Figura 121: Diagrama seqüència disseny accept_pa, pa_accepted i pa_rejected</i> .....	148
<i>Figura 122: Diagrama seqüència disseny finish_pa</i> .....	149
<i>Figura 123: Diagrama seqüència disseny veurePreferencies</i> .....	150
<i>Figura 124: Diagrama seqüència disseny enviarPreferencies</i> .....	150
<i>Figura 125: Diagrama seqüència disseny crearGrup</i> .....	151
<i>Figura 126: Diagrama seqüència disseny canviarNom</i> .....	152
<i>Figura 127: Diagrama seqüència disseny enviarParticipantsGrup</i> .....	153
<i>Figura 128: Diagrama seqüència disseny confirmarElimPartGrup</i> .....	153
<i>Figura 129: Diagrama seqüència disseny confirmarSortirGrup</i> .....	154
<i>Figura 130: Diagrama seqüència disseny leave_group</i> .....	155
<i>Figura 131: Diagrama seqüència disseny veureInfoGrup</i> .....	156
<i>Figura 132: Diagrama seqüència disseny getGroupServer</i> .....	157
<i>Figura 133: Diagrama seqüència disseny llistarGrups</i> .....	157

<i>Figura 134: Diagrama seqüència disseny confirmarNomenar</i> .....	158
<i>Figura 135: Diagrama seqüència disseny cerca</i> .....	158
<i>Figura 136: Diagrama seqüència disseny enviarOCancelarSolicitud</i> .....	159
<i>Figura 137: Diagrama seqüència disseny llistarSolicitud</i> .....	160
<i>Figura 138: Diagrama seqüència disseny respondreSolicitud</i> .....	160
<i>Figura 139: Disseny de navegació 1</i> .....	161
<i>Figura 140: Disseny de navegació 2</i> .....	162
<i>Figura 141: Esquema enviar dades client-servidor</i> .....	174
<i>Figura 142: Esquema demanar canvis client-servidor</i> .....	174
<i>Figura 143: Esquema enviar dades P2P</i> .....	174
<i>Figura 144: Esquema demanar canvis P2P</i> .....	175
<i>Figura 145: Gràfica enviar dades</i> .....	175
<i>Figura 146: Gràfica reconnexió només enviar</i> .....	176
<i>Figura 147: Gràfica reconnexió no s'envia ni es rep</i> .....	177
<i>Figura 148: Gràfic comparació temps experiment 1</i> .....	181
<i>Figura 149: Gràfic comparació temps experiment 2</i> .....	188
<i>Figura 150: Gràfic comparació consum experiment 3</i> .....	195
<i>Figura 151: Gràfic comparació temps experiment 4</i> .....	202
<i>Figura 152: Planificació inicial en llista de tasques</i> .....	204
<i>Figura 153: Planificació inicial en diagrama de Gantt</i> .....	205
<i>Figura 154: Vista login</i> .....	213
<i>Figura 155: Vista llistar tasques</i> .....	213
<i>Figura 156: Vista filtrar tasques</i> .....	214
<i>Figura 157: Vista informació tasca</i> .....	215
<i>Figura 158: Vista crear tasca</i> .....	215
<i>Figura 159: Vistes afegir participants i Afegir precedents</i> .....	216
<i>Figura 160: Vista llistar calendaris</i> .....	217
<i>Figura 161: Vista informació calendari</i> .....	217
<i>Figura 162: Vista crear calendari</i> .....	218
<i>Figura 163: Vista afegir participants</i> .....	218
<i>Figura 164: Vista llistar grups</i> .....	219
<i>Figura 165: Vista cercar grup</i> .....	219
<i>Figura 166: Vista informació grup</i> .....	220
<i>Figura 167: Vista crear grup</i> .....	221
<i>Figura 168: Vista afegir membres</i> .....	221
<i>Figura 169: Vista llistat de sol·licituds</i> .....	222
<i>Figura 170: Vista llistat d'esdeveniments</i> .....	222
<i>Figura 171: Vista llistat de pending approvals</i> .....	222
<i>Figura 172: Vista informació pending approval</i> .....	223
<i>Figura 173: Vista llistar meetings</i> .....	223
<i>Figura 174: Vista informació meeting</i> .....	223
<i>Figura 175: Vista estadístiques d'usuari</i> .....	224
<i>Figura 176: Vista preferències</i> .....	224

## Agraïments

Primerament vull agrair al tutor d'aquest projecte, Fatos Xhafa, per oferir-me la possibilitat de desenvolupar un projecte que he trobat molt interessant, guiar-me durant el seu desenvolupament i resoldre tots els dubtes que m'han anat sorgint.

Als membres de RDLab, que van instal·lar i configurar al servidor del LSI tots els elements necessaris per a l'elaboració del projecte. Vull agrair especialment a en Gabriel, que es va reunir amb mi en dues ocasions per entendre exactament què volia i ajudar-me a arreglar un parell d'assumptes.

A la comunitat de *Stackoverflow* i del fòrum de *SugarCRM*, sempre disposats a resoldre els meus dubtes i els de tothom que ho necessiti. Sense la seva ajuda segurament no hauria pogut finalitzar l'aplicació o hauria hagut d'optar per alternatives menys eficients i útils.

A Andy Tsui, desenvolupador d'Android Binding, que quan em vaig posar en contacte amb ell per intentar corregir un error relacionat amb la seva llibreria em va oferir tota la seva ajuda, descarregant-se els meus exemples i investigant pel seu compte per intentar arreglar-ho. Tot i que, com s'explica a l'apartat d'implementació, finalment no va ser possible corregir-ho, em va sorprendre enormement la seva dedicació envers una persona anònima d'Internet.

Als companys d'universitat i altres amics que m'han ajudat a seguir endavant amb la carrera i fer-la menys dura, sense ells potser no hauria arribat a escriure aquesta memòria.

Finalment, agraeixo a la meva família que m'hagi ofert la possibilitat de poder realitzar els estudis universitaris que volia i a on volia i tot el suport que m'han ofert durant aquests anys.

# **Descripció del projecte**

En aquest primer apartat de la documentació del projecte s'explica en què consisteix el projecte, les motivacions que hi ha darrere d'aquest i les primeres decisions preses per tal de començar-lo a desenvolupar.



## 1. Context i motivació del projecte

El quadrimestre de primavera del curs 2010-2011, l'estudiant Vicent Seguí Pascual de la UOC va realitzar el projecte *Sistemes Android i treball en grup* amb el professor Fatos Xhafa com a consultor. Aquest projecte pretenia implementar un sistema de sincronització de contactes i de calendaris per a sistemes Android. Degut a certs problemes durant la implementació d'aquest, es va prescindir de la sincronització de calendaris per manca de temps, deixant aquest objectiu com a possible extensió futura.

En aquest punt neix la idea d'aquest projecte, que es basarà per una banda en estendre el projecte mencionat anteriorment per tal d'incloure la sincronització de calendaris que no es va poder implementar en el seu moment, i per l'altra a estudiar el tema de la sincronització en sistemes Android mitjançant un conjunt d'experiments.

El sistema Android ja disposa d'un sistema de calendari que permet treballar en grup: Google Calendar. Durant l'anàlisi del projecte es va valorar la possibilitat de treballar amb els calendaris i esdeveniments de Google Calendar enlloc de tenir els nostres propis calendaris, però aquesta opció es va descartar pel fet que obligaríem als usuaris de la nostra aplicació a tenir un compte de Google, i això podria ser no desitjable des del seu punt de vista. A part, d'aquesta manera podem ser molt més flexibles en la implementació i intentar ampliar les funcionalitats i opcions d'aquest sistema, obtenint una aplicació més atractiva i més orientada al treball en grup.

El fet que els sistemes mòbils siguin una de les plataformes amb més expansió mundial van fer que m'interessés en aquest projecte. A més, si observem els sistemes web més populars actualment veiem que estan orientats a la interacció entre usuaris i el treball en grup: xarxes socials, webs per compartir vídeos i/o música, dropbox, etc. Per tant, crec que és important conèixer eines per tal d'incloure aquest treball en grup en els sistemes mòbils i estudiar el seu funcionament per tal de definir els avantatges i inconvenients de cada cas. Si a això s'hi afegeix el fet de poder treballar amb tecnologies poc (o gens) utilitzades durant la carrera, fet que em permetrà ampliar els meus coneixements, obtenim un projecte que trobo molt interessant.

## 2. Descripció del projecte

Com ja s'ha mencionat anteriorment, el projecte consisteix en la implementació d'una aplicació que treballa amb calendaris i la sincronització d'aquests.

El nostre software permetrà als usuaris la creació i modificació dels seus propis calendaris que contindran esdeveniments i tasques. Si l'usuari ho desitja podrà compartir certes tasques i esdeveniments amb altres usuaris perquè hi participin i apareguin en els seus calendaris, o directament compartir un calendari sencer. Si un usuari té permisos suficients podrà modificar un esdeveniment o tasca, i aquests canvis es propagaran a tots els usuaris que participin a l'esdeveniment o calendari.

Per altra banda, el projecte comptarà amb una fase d'experimentació en la qual s'estudiarà la sincronització d'esdeveniments, tasques i calendaris en diferents situacions i implementacions. Durant l'etapa d'implementació es programaran diversos algorismes per realitzar la sincronització o per modificar lleument alguna de les seves parts, i s'intentarà trobar quins són els més avantatjosos.

### 3. Objectius

A partir del context i motivació del projecte i de la documentació i codi del projecte que ens disposem a estendre podem definir els objectius del projecte, presentats a continuació:

- Donar continuïtat al projecte *Sistemes Android i treball en grup* d'en Vicent Seguí Pascual, reutilitzant totes les classes que sigui possible i mantenint la tecnologia utilitzada.
- Oferir als usuaris una aplicació que els permeti crear i modificar els seus propis calendaris, amb més diversitat i funcionalitats que el sistema actual de calendari d'Android i més orientat al treball en grup.
- Mitjançant l'experimentació, descobrir els mètodes d'implementació més avantatjosos segons els criteris marcats en el desenvolupament de l'aplicació.
- Programar l'aplicació de manera adequada per facilitar possibles ampliacions futures.

### 4. Decisió de la tecnologia

Les tecnologies del sistema venen definides principalment pel projecte que s'està estenent, *Sistemes Android i treball en grup*. No obstant, en les primeres etapes del projecte s'ha realitzat un anàlisi de totes elles per tal de determinar la seva viabilitat i si eren les més adequades per al sistema.

**-Java:** El nostre sistema està dissenyat per funcionar en dispositius Android, el llenguatge de programació dels quals és el Java.

**-SQLite:** Com en el cas anterior, els sistemes Android utilitzen SQLite com a base de dades local, ja que les seues recursos són bastant limitats. L'emmagatzemament de dades també es pot realitzar en fitxers, però s'ha preferit utilitzar la base de dades.

**-Android 2.2:** Tot i que el més usual actualment és que els smartphones actuals utilitzin la versió 2.3.3 de Android o posteriors, no és estrany trobar-ne que encara utilitzin la versió 2.2. Com que ens interessa que la nostra aplicació es pugui utilitzar en el màxim nombre de dispositius, s'ha decidit que s'implementarà l'aplicació perquè sigui compatible amb aquesta versió.

**-Android-Binding:** Com es detallarà en futures seccions, s'utilitzarà aquesta llibreria externa de codi lliure per tal d'aconseguir major independència entre les vistes i el nostre sistema, aplicant el patró MVVM.

**-SugarCRM:** En el projecte *Sistemes Android i treball en grup* es va decidir utilitzar SugarCRM degut a que l'autor del projecte treballava en una empresa que tenia un servidor amb aquest CRM instal·lat. D'aquesta manera el projecte quedava situat dins d'un entorn empresarial, simulant el cas d'un projecte real elaborat per aquesta empresa. Per tal de donar continuïtat al projecte, i degut a que aquesta tecnologia no suposa problemes afegits, s'ha decidit mantenir-la.

**-PHP:** Les funcionalitats del servidor s'hauran d'ampliar per tal de poder-lo adaptar al nostre sistema. Com que SugarCRM està implementat en PHP, s'utilitzarà aquest llenguatge de programació per ampliar-lo.

**-Memcached:** Memcached és un sistema distribuït que s'utilitza per a l'emmagatzematge en caché de dades o objectes, reduint així la necessitat d'accedir a un origen de dades extern. S'aplicarà al servidor ja que es necessitarà en certes funcionalitats.

**-JXTA:** JXTA (Juxtapose) és una plataforma P2P de codi lliure creada per Sun Microsystems l'any 2001 i que està definida com un conjunt de protocols basats en XML. Aquests protocols permeten que dispositius connectats a una xarxa intercanviïn missatges entre ells independentment de la topologia de la xarxa. La definició de JXTA és abstracte i està basat en una sèrie de protocols oberts de manera que, en teoria, pot ser portat a qualsevol llenguatge de programació modern. Hi ha diverses implementacions, la més avançada de les quals és JXSE, la versió per Java. La versió per Java Micro Edition s'anomena JXME.

**-PeerDroid:** PeerDroid és una llibreria de codi lliure que aplica els protocols de JXME a la plataforma Android. Permet crear aplicacions per la plataforma Android que utilitzin les propietats de JXTA, creant així una xarxa de mòbils i altres peers més tradicionals (ordinadors, per exemple).

## 5. Requisits no funcionals

Durant l'anàlisi de requisits s'obtenen requisits de dos tipus: funcionals, que defineixen el comportament intern del software, i no funcionals, que defineixen exigències de qualitat que s'imposen a projecte. Els requisits funcionals es definiran a l'apartat d'especificació ja que es corresponen als casos d'ús del sistema. A continuació es defineixen els requisits no funcionals.

**Requisit:** Sistema intuïtiu

**Descripció:** El sistema ha de ser suficientment intuïtiu per a l'usuari per tal de facilitar la utilització de la nostra aplicació.

**Requisit:** Sincronització transparent

**Descripció:** La sincronització de calendaris s'ha de realitzar de forma transparent a l'usuari amb l'excepció dels casos en que hi hagi un conflicte.

**Requisit:** Eficiència

**Descripció:** El programa implementat haurà de ser eficient tan en temps com en recursos, ja que s'utilitzarà en sistemes mòbils que disposen de recursos força limitats i amb notables diferències entre diferents dispositius.

**Requisit:** Seguretat

**Descripció:** El sistema haurà de ser el més segur possible, de manera que les dades i calendaris dels usuaris només puguin ser accedits pels usuaris amb autorització per fer-ho i les contrasenyes estiguin encriptades.

**Requisit:** Extensibilitat

**Descripció:** Durant la implementació del sistema s'haurà de tenir en compte que aquest haurà d'estar orientat a possibles extensions futures.

## 6. Anàlisi de riscos

A l'implantar un nou sistema hem de tenir en compte una sèrie de riscos. Quan parlem de riscos ens referim a allò que pot afectar negativament al projecte i que té certa probabilitat de passar. Així doncs, la única manera de minimitzar els efectes d'aquests riscos és detectar-los abans del desenvolupament del projecte, quantificar en quina mesura poden ser perjudicials i planificar solucions per actuar en cas que ens hi trobem.

A continuació definim possibles riscos als que estarà exposat el nostre projecte.

<b>Mala planificació temporal</b>	
<b>Descripció</b>	Un dels riscos més importants als que està exposat el nostre projecte és una mala planificació temporal, fet que implicarà que la duració del projecte augmenti, provocant un possible incompliment de la data d'entrega del projecte.
<b>Impacte</b>	Mig
<b>Probabilitat</b>	Alta
<b>Mesures</b>	Degut a que encara no hi ha una data d'entrega del projecte fixa, si les hores extra no són molt nombroses les podrem realitzar igualment sense afectar negativament el projecte. En cas que la

	quantitat d'hores extra sigui massa elevada, es prescindirà de les funcionalitats menys importants.
--	---

*Taula 1: Risc 1*

<b>No es disposa de suficient documentació per elaborar el projecte</b>	
<b>Descripció</b>	Ja que treballem amb certes tecnologies amb les quals no hem treballat mai, ens caldrà tenir suficient documentació per a poder complir tots els objectius.
<b>Impacte</b>	Alt
<b>Probabilitat</b>	Molt baixa
<b>Mesures</b>	Abans d'iniciar el projecte estudiarem totes les tecnologies que utilitzarem, intentant no incloure'n cap que estigui massa poc documentada.  En cas de trobar-nos en aquesta situació un cop iniciat el projecte, es substituiria la tecnologia en qüestió per alguna altra que ofereixi resultats similars.

*Taula 2: Risc 2*

<b>Manca de la infraestructura necessària per a realitzar el projecte</b>	
<b>Descripció</b>	La realització del projecte requereix disposar d'un servidor, smartphones, etc. Algun d'aquests elements podria no ser senzill d'obtenir.
<b>Impacte</b>	Mig
<b>Probabilitat</b>	Baixa
<b>Mesures</b>	Definir tots els elements necessaris per a realitzar el projecte abans de l'inici d'aquest i intentar disposar de tots ells prèviament mitjançant l'ajuda del director del projecte.

*Taula 3: Risc 3*

<b>Caiguda del servidor</b>	
<b>Descripció</b>	Un cop el projecte estigui en funcionament, el servidor podria deixar d'estar funcional en algun moment, provocant problemes en el funcionament del software.
<b>Impacte</b>	Mig
<b>Probabilitat</b>	Mitja

<b>Mesures</b>	<p>En cas de disposar d'un servidor extra, instal·lar-hi el software amb una còpia de les dades del servidor principal i realitzar la sincronització amb aquest. Quan el servidor principal torni a estar actiu, traslladar els canvis del servidor secundari al principal.</p> <p>En cas de no disposar d'un servidor extra, notificar als usuaris de l'error per tal que no perdin canvis fets.</p>
----------------	---

*Taula 4: Risc 4*

<b>Conflictes en la sincronització</b>	
<b>Descripció</b>	Dos usuaris poden realitzar canvis en un mateix element al mateix temps, provocant un conflicte que el sistema no sap com resoldre.
<b>Impacte</b>	Baix
<b>Probabilitat</b>	Molt alta
<b>Mesures</b>	S'implementarà un sistema de detecció i resolució de conflictes en el qual hauran de participar els usuaris afectats.

*Taula 5: Risc 5*

## **7. Conceptes importants per al projecte**

### **7.1 Sincronització**

La sincronització és un element clau sobre el qual es fonamenta el projecte. La sincronització de dades s'utilitza per aconseguir tenir exactament les mateixes dades en dues o més ubicacions diferents. En el cas del nostre projecte, ens interessa que les dades de l'aplicació dels telèfons mòbils coincideixin entre elles (sempre que no hi hagi conflictes). És a dir, quan un usuari faci un canvi que no provoqui cap conflicte, aquest s'ha de reflectir en tots els dispositius que continguin les mateixes dades.

Hi ha molts aspectes que influeixen en la sincronització, com per exemple l'estratègia d'emmagatzematge de dades. Es pot optar per tenir les dades distribuïdes en els diferents dispositius, de manera que quan un usuari realitzi un canvi s'haurà de comunicar directament amb els dispositius necessaris, o bé tenir-les centralitzades en un servidor i que tota la sincronització es faci a través d'aquest. Tot i que a l'apartat d'experimentació es posaran en pràctica totes dues estratègies per tal de comparar-les, a la versió definitiva del nostre sistema s'utilitza un servidor. És a dir, el servidor contindrà tota la informació del nostre sistema, i aquesta estarà replicada en els dispositius mòbils que sigui necessari. Òbviament, un dispositiu només desarà la informació necessària per a l'usuari que l'estigui utilitzant en aquell moment, no desarà totes les dades del sistema. Quan un usuari realitzi un canvi el dispositiu l'enviarà únicament al servidor, i aquest el desarà i l'enviarà a la resta de dispositius afectats.

Hi ha diverses estratègies per a realitzar la sincronització de les dades. Les més usades són la sincronització completa i la incremental, tot i que n'hi ha d'altres, i aquestes dues són les que s'utilitzaran en el projecte.

La sincronització completa consisteix en enviar sempre totes les dades. És a dir, si un dispositiu vol posar-se al dia amb la informació del servidor, aquest li enviarà sempre totes les dades corresponents a l'usuari, i serà el dispositiu qui haurà de desar o descartar la informació en funció de si ja disposa d'ella o no. Aquesta sincronització es pot considerar que és més "segura" en quant a errors de sincronització, ja que com que sempre rebem tota la informació podem estar segurs que el dispositiu contindrà exactament tota la informació del servidor, però això implica un cost temporal i de tràfic molt elevat, ja que estem recuperant i enviant informació que no és necessària. En el nostre projecte aquesta sincronització s'utilitzarà en el moment de fer login al sistema (ja que el dispositiu no contindrà cap dada, les haurà de rebre totes) i quan l'usuari ho indiqui expressament. En la resta de casos s'utilitzarà una sincronització incremental.

La sincronització incremental consisteix a enviar únicament les dades que s'han modificat a partir de l'última sincronització. Així doncs, un dispositiu rebrà únicament els canvis fets, no tota la informació relativa al seu usuari. En aquest cas és més fàcil que hi hagi algun error de sincronització. Per exemple, si per algun motiu la sincronització no es realitzés correctament però s'actualitzés la data de l'última sincronització, totes les dades que s'haurien d'haver rebut en aquesta sincronització ja no es rebran a no ser que tornin a ser modificades, ja que el dispositiu considera que està sincronitzat tot i que en realitat no és així. No obstant, si es va amb compte aquests errors es poden minimitzar. Per altra banda, aquesta sincronització és molt més eficient tant en espai com en tràfic, ja que només s'envia la informació necessària, no hi ha dades inútils. Com que mai es pot estar segur que no succeirà un error de sincronització com els esmentats anteriorment, s'ha decidit mantenir l'opció de sincronització completa quan l'usuari ho indiqui, ja que així es podrà recuperar fàcilment d'aquest tipus d'errors.

Amb els telèfons mòbils ens podem trobar sovint que els dispositius perden la connexió a Internet i no es poden sincronitzar. Si un usuari realitza canvis i no té connexió, aquests es desaran a nivell local i s'enviaran al servidor quan recuperi la connexió. Pel què fa als canvis dels altres usuaris, un cop es recuperi la connexió el dispositiu demanarà els canvis al servidor sempre i quan n'hi hagi algun.

Durant el procés de sincronització ens podem trobar amb conflictes, ja que dos usuaris poden estar modificant les mateixes dades. En aquests casos, quin valor s'ha de considerar correcte? Hi ha moltes estratègies per resoldre conflictes, i en aquest projecte hem optat per traslladar la decisió als usuaris. Quan es produeixi un conflicte, es demanarà als usuaris afectats que decideixin quin dels valors volen que es desi, i es prendrà una decisió mitjançant una votació amb majoria simple. En cas d'empat, es repetirà la votació.

## 7.2 Treball en grup

Com ja s'ha comentat anteriorment, les tecnologies actuals estan evolucionant cap al treball en grup. Compartir dades, experiències, recursos, etc. pot millorar enormement l'eficiència envers el fet de treballar individualment. Fins ara gran part d'aquest treball en grup s'havia de fer físicament: compartir documents, CDs, agendes, etc. Ara, en canvi, tot això es pot fer de manera virtual.

L'aparició dels anomenats *smartphones* ha revolucionat encara més el treball en grup, ja que ens permet aplicar-lo en qualsevol lloc on disposem de connexió a Internet, no ens veiem limitats a disposar d'un ordinador per poder-ho utilitzar. És habitual veure gent parlant, compartint fitxers i imatges o entrant a xarxes socials des del carrer o el metro.

Aquest és un altre concepte en el qual es basa la nostra aplicació. Tot i que ja existeix software de creació i modificació de tasques i calendaris, la majoria van ser pensats per a ser utilitzats de forma individual (com si fos una agenda personal) i a partir d'aquí es van ampliar per poder aprofitar més o menys el treball en grup. La nostra aplicació, en canvi, estarà pensada i dissenyada per tal de ser utilitzada principalment pel treball en grup, tot i que també permetrà tenir calendaris i tasques individuals que no podrà consultar cap altre usuari.

Tot i que els *smartphones* permeten connectar-se a la xarxa des de molts llocs, encara no és possible disposar de connexió permanent, sempre hi ha punts on es perd la connexió. El fet d'obligar a disposar de connexió per poder utilitzar una aplicació de treball en grup pot ser un inconvenient molt important per als usuaris, especialment en el nostre cas en què la quantitat de canvis pot ser elevada. A part, també pot passar que el servidor falli, de manera que tot i tenir connexió l'usuari es trobaria que no pot fer res, provocant-li frustració i malestar. El fet de poder treballar *offline* ofereix a l'usuari un conjunt d'avantatges molt atractius: una millor organització del seu temps (ja que pot utilitzar l'aplicació sempre que vulgui, sense restriccions), comoditat i despreocupació del fet de tenir connexió, entre d'altres. No obstant, aquesta alternativa ofereix alguns inconvenients de cara al programador que s'han de tractar.

Permetre treballar *offline* fa que augmentin la quantitat de conflictes, ja que al no rebre els canvis fets per altres usuaris no es pot saber si la tasca que s'està modificant ja ha estat modificada o no. A part, cal decidir una estratègia de com i quan enviar i rebre les dades. Android disposa d'un sistema que notifica quan es recupera la connexió a Internet, de manera que en la nostra aplicació s'enviaran els canvis fets *offline* tan bon punt el dispositiu torni a estar connectat a Internet. Si el problema és del servidor, però, ens haurèm de limitar a anar-ho intentant fins que funcioni, però tot aquest procés serà transparent a l'usuari.

No obstant, cal limitar les funcionalitats que un usuari pot fer *offline*, ja que mantenir tot el sistema per tal que es pugui fer qualsevol operació de manera *offline* no és trivial. Així doncs, s'ha decidit que les operacions relacionades directament amb les tasques es podran fer *offline* (crear, modificar, eliminar, afegir participants, etc.), però la resta no (operacions amb grups, calendaris, pending approvals i meetings).



### 7.3 Peer-to-Peer

Les xarxes peer-to-peer, a diferència de les xarxes que utilitzen una arquitectura client-servidor, permet als dispositius o nodes comunicar-se directament sense utilitzar intermediaris. Una xarxa P2P pura no utilitza cap servidor, sinó que tota la comunicació es fa entre clients o *peers* i tots ells tenen la mateixa responsabilitat. Aquest tipus de xarxa no és gens pràctica, ja que si no s'organitza el tràfic entre *peers* utilitzant mètodes eficients el resultat és un sistema amb un rendiment molt baix. Per tal de solucionar-ho cal que alguns *peers* tinguin més responsabilitats que altres i els ajudin a trobar altres *peers*. Aquest tipus de "*super peers*" s'anomenen *rendezvous peers*.

Hi ha un altre tipus especial de *peers* anomenats *relay peers*. Aquests s'utilitzen quan un *peer* no pot ser accedit directament (per exemple, si es troba darrere d'un firewall o NAT) i s'utilitza com una espècie de *proxy*: la resta de *peers* envien els missatges al *relay peer* i aquest els redirigeix cap al *peer* que representa.

Una xarxa P2P sol estar formada per un conjunt de *rendezvous peers*, i cadascun d'ells s'encarrega d'organitzar un grup de *peers*. També hi sol haver tot un seguit de *relay peers* que s'encarreguen de garantir l'accés a tots els *peers* "amagats".

En el nostre sistema només utilitzarem l'arquitectura P2P per realitzar experiments, no estarà present a la versió final del sistema, tot i que en ampliacions futures es podria considerar introduir-la. Així doncs, serà suficient implementar una versió senzilla d'una xarxa P2P que contingui els elements suficients per a poder realitzar les proves: un *relay peer* i un *rendezvous peer*, tots dos ubicats al mateix servidor. La responsabilitat del *rendezvous peer* es limitarà al màxim per tal d'acostar-se tant com es pugui a una xarxa P2P pura; el *rendezvous peer* es limitarà a mantenir la llista de *peers* amb les seves adreces i a transmetre-la als nous *peers* que s'afegeixin a la xarxa. Tota la comunicació de dades de tasques es farà directament entre *peers*. Com que hem limitat les funcionalitats del *rendezvous peer* no disposarem de la informació centralitzada al servidor, fet que complica enormement el tractament de conflictes. Per simplificar, a l'experimentació s'utilitzarà un sistema P2P optimista, suposant que no hi ha conflictes entre els usuaris.

Tot i que a priori les xarxes P2P ofereixen una millor eficiència temporal degut a la manca d'intermediaris, és usual que el tràfic de la xarxa augmenti. A l'apartat d'experimentació s'analitzarà en detall el temps i el tràfic de totes dues xarxes per a poder comparar-les aplicades al nostre cas i observar-ne els avantatges i inconvenients.

### 7.4 Dispositius mòbils

Els *smartphones* han suposat una revolució en el món de les noves tecnologies. Ens ofereixen la possibilitat de disposar d'una espècie d'ordinadors portàtils, de manera que podem realitzar a on vulguem moltes coses que abans només podíem fer si ens trobàvem davant d'un ordinador. Els avantatges que això ens ofereix són innumerables: accés a tota la informació d'Internet en qualsevol lloc, gran millora

de treball en grup, possibilitat de realitzar operacions *online*, comunicació "gratuïta", etc. Però els *smartphones* també tenen els seus inconvenients.

Els inconvenients dels *smartphones* en general, com la saturació de la xarxa, cauen fora de l'àmbit del projecte. Ens interessen els inconvenients específics del sistema Android i del hardware dels dispositius, en concret el seu principal: la limitació de recursos. És obvi que un *smartphone* no pot disposar dels mateixos recursos que un ordinador, i això limita la seva capacitat d'operació.

Si s'observen els telèfons de fa uns anys i els *smartphones* actuals es pot veure que s'està avançant cap a una ampliació de les prestacions i una millor eficiència temporal en detriment d'un element molt important: la bateria. Actualment és difícil trobar usuaris d'*smartphones* que puguin passar més de 24 hores sense haver de carregar la bateria. El consum de bateria va estretament lligat a l'ús del telèfon i, en concret, de la CPU, així que quan s'implementen aplicacions per a aquests sistemes s'ha de tenir bastant en compte la bateria. Per exemple, en la nostra aplicació inicialment s'havia decidit fer sincronitzacions periòdiques, una opció viable en ordinadors, però finalment es va haver de canviar i fer sincronitzacions únicament quan són necessàries ja que en cas contrari el consum de bateria era excessiu.

Això és només un exemple de com ens limiten els recursos dels telèfons. El rendiment del processador, la capacitat del dispositiu i el tràfic utilitzat s'han de tenir molt en compte quan es programen aplicacions Android, ja que ens podem trobar que l'aplicació funciona lentament o es bloqueja, que la memòria del dispositiu s'omple ràpidament o que en pocs dies l'usuari arriba al límit de tràfic contractat amb el seu operador. Algunes d'aquestes limitacions de recursos també són presents en ordinadors, és clar, però en molta menor mesura.

# Especificació

A l'especificació es descriu en detall el comportament extern del sistema descrivint els requisits funcionals que el software ha de complir. Per fer l'especificació s'ha utilitzat el llenguatge UML (*Unified Modeling Language*).

## 1. Especificació casos d'ús

A l'especificació dels casos d'ús descriurem textualment cadascun dels casos d'ús presentats a l'apartat anterior. Cal tenir en compte que, en les interaccions entre l'actor i el sistema dels casos d'ús on l'actor és l'usuari, es realitzen accions al sistema local i al servidor, i aquestes segones es realitzaran quan es faci la sincronització corresponent tot i no estar indicat explícitament al cas d'ús. Per exemple, tota la gestió d'esdeveniments es realitza a nivell de servidor, a nivell local només es tindran els esdeveniments dirigits a l'usuari que utilitza l'aplicació.

### 1.1 Gestió de comptes

<b>Cas d'ús</b>	Login
<b>Actor</b>	Usuari no loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari entra al seu compte per utilitzar l'aplicació.
<b>Precondició</b>	L'usuari no està loguejat.
<b>Postcondició</b>	L'usuari passa a ser usuari loguejat.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari inicia l'aplicació.</li> <li>2. El sistema comprova si està loguejat.</li> <li>3. El sistema demana la informació necessària a l'usuari.</li> <li>4. L'usuari introdueix la informació.</li> <li>5. El sistema verifica les dades introduïdes i demana confirmació a l'usuari.</li> <li>6. L'usuari confirma.</li> <li>7. El sistema desa la informació necessària, crida al cas d'ús <i>Demandar canvis i, a continuació, Llistar tasques</i>.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p> <p>2A. L'usuari està loguejat.</p> <p>2A.1. Crida el cas d'ús <i>Llistar tasques</i>.</p> <p>2A.2. Fi del cas d'ús.</p> <p>5A. Les dades no són correctes.</p> <p>5A.1. El sistema notifica l'error a l'usuari.</p> <p>5A.2. Torna al pas 3.</p> <p>6A. L'usuari no confirma l'acció.</p> <p>6A.1. Torna al pas 3.</p>

Taula 6: Cas d'ús Login

<b>Cas d'ús</b>	Logout
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari surt del seu compte i ja no podrà utilitzar l'aplicació.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	L'usuari passa a ser usuari no loguejat.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol fer logout.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> </ol>

	4. El sistema fa una última sincronització de tasques (crida al cas d'ús <i>Enviar tasques</i> ).
	5. El sistema elimina les dades desades al fer login.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Hi ha un error al sincronitzar les tasques. 4A.1. El sistema notifica l'error a l'usuari i li demana si vol seguir amb el cas d'ús. 4A.2. Si l'usuari indica que vol seguir, va al pas 5. Si no, fi del cas d'ús.

*Taula 7: Cas d'ús Logout*

<b>Cas d'ús</b>	Veure estadístiques
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari veu les estadístiques relacionades amb el seu compte.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra les estadístiques de l'usuari.
<b>Curs principal</b>	1. L'usuari indica que vol veure les estadístiques del seu compte. 2. El sistema mostra les estadístiques.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.

*Taula 8: Cas d'ús Veure estadístiques*

## 1.2 Gestió de calendaris

<b>Cas d'ús</b>	Crear calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar calendaris.
<b>Descripció</b>	L'usuari crea un calendari.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El calendari es crea correctament i es notifica als usuaris participants.
<b>Curs principal</b>	1. L'usuari indica que vol crear un nou calendari. 2. El sistema demana les dades a l'usuari. 3. L'usuari introdueix la informació del calendari i dels seus participants. 4. El sistema verifica les dades introduïdes. 5. El sistema crea el nou calendari i genera un esdeveniment per notificar als usuaris participants.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Les dades no són correctes. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Torna al pas 2.

*Taula 9: Cas d'ús Crear calendari*

<b>Cas d'ús</b>	Modificar calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació calendari.
<b>Descripció</b>	L'usuari modifica alguna de les dades del calendari.
<b>Precondició</b>	L'usuari està loguejat i participa al calendari.
<b>Postcondició</b>	La informació es modifica correctament.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol modificar un calendari.</li> <li>2. El sistema demana a l'usuari les noves dades.</li> <li>3. L'usuari introdueix la informació.</li> <li>4. El sistema verifica les dades introduïdes.</li> <li>5. El sistema modifica el calendari.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.</p> <p>4A. Les dades no són correctes. 4A.1. El sistema notifica 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 pending approval per tal que la resta de participants puguin acceptar o rebutjar els canvis i notifica als usuaris participants.</p>

*Taula 10: Cas d'ús Modificar calendari*

<b>Cas d'ús</b>	Eliminar calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar calendaris.
<b>Descripció</b>	L'usuari elimina un dels seus calendaris.
<b>Precondició</b>	L'usuari està loguejat i participa al calendari.
<b>Postcondició</b>	El calendari i les seves tasques s'eliminen del sistema.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol eliminar un calendari.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema elimina el calendari i les seves tasques i el sistema local elimina els grups que ja no sigui necessari desar a nivell local.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.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. 4A.1. El sistema crea un pending approval per tal que la resta de participants puguin acceptar o rebutjar l'eliminació i notifica als usuaris participants.</p>

*Taula 11: Cas d'ús Eliminar calendari*

<b>Cas d'ús</b>	Veure informació calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	

<b>Descripció</b>	L'usuari vol veure la informació corresponent a un calendari.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra la informació del calendari especificat.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure la informació d'un calendari. 2. El sistema mostra la informació del calendari.
<b>Cursos alternatius</b>	

*Taula 12: Cas d'ús Veure informació calendari*

<b>Cas d'ús</b>	Afegir participant al calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació calendari.
<b>Descripció</b>	L'usuari afegeix un o més grups com a participants del calendari.
<b>Precondició</b>	L'usuari està loguejat i participa al calendari.
<b>Postcondició</b>	Els grups passen a participar al calendari.
<b>Curs principal</b>	1. L'usuari indica que vol afegir participants al calendari. 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 grups com a participants i genera un esdeveniment per notificar als membres d'aquests grups que hi han estat afegits.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Les dades no són correctes. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Torna al pas 2.  5A. El calendari té més usuaris participants. 5A.1. El sistema crea un pending approval per tal que la resta de participants puguin acceptar o rebutjar l'addició de participants i notifica als actuals usuaris participants.

*Taula 13: Cas d'ús Afegir participant al calendari*

<b>Cas d'ús</b>	Eliminar participant del calendari
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació calendari.
<b>Descripció</b>	L'usuari elimina un grup com a participant del calendari.
<b>Precondició</b>	L'usuari està loguejat i participa al calendari.
<b>Postcondició</b>	El grup deixa de participar al calendari.
<b>Curs principal</b>	1. L'usuari indica que vol eliminar un determinat grup com a participant del calendari. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema elimina grup participant.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  3A. L'usuari no confirma l'acció. 3A.1. Fi del cas d'ús.

	<p>4A. El calendari té més usuaris participants.</p> <p>4A.1. El sistema crea un pending approval per tal que la resta de participants puguin acceptar o rebutjar l'eliminació del grup participant i notifica als actuals usuaris participants.</p>
--	--

*Taula 14: Cas d'ús Eliminar participant del calendari*

<b>Cas d'ús</b>	Llistar calendaris
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure el llistat de calendaris en què participa.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra el llistat de calendaris en què participa l'usuari.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema que vol veure el llistat de calendaris.</li> <li>2. El sistema mostra la llista de calendaris en què participa l'usuari.</li> </ol>
<b>Cursos alternatius</b>	

*Taula 15: Cas d'ús Eliminar participant del calendari*

### 1.3 Gestió de tasques

<b>Cas d'ús</b>	Crear tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar tasques.
<b>Descripció</b>	L'usuari crea una tasca.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	La tasca es crea correctament al sistema local.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol crear una nova tasca.</li> <li>2. El sistema demana les dades a l'usuari.</li> <li>3. L'usuari introdueix la informació de la tasca.</li> <li>4. El sistema verifica les dades introduïdes.</li> <li>5. El sistema crea la nova tasca a nivell local i, si és necessari, crea l'alerta corresponent. Crida el cas d'ús <i>Enviar tasques</i>.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p> <p>4A. Les dades no són correctes.</p> <p>4A.1. El sistema notifica l'error a l'usuari.</p> <p>4A.2. Torna al pas 2.</p>

*Taula 16: Cas d'ús Crear tasca*

<b>Cas d'ús</b>	Modificar tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació tasca.
<b>Descripció</b>	L'usuari modifica alguna de les dades de la tasca.
<b>Precondició</b>	L'usuari està loguejat i participa a la tasca.
<b>Postcondició</b>	La informació es modifica correctament al sistema local.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol modificar una tasca.</li> </ol>



	<p>2. El sistema demana a l'usuari les noves dades.</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 una alerta. Crida el cas d'ús <i>Enviar tasques</i>.</p>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p> <p>4A. Les dades no són correctes.</p> <p>4A.1. El sistema notifica l'error a l'usuari.</p> <p>4A.2. Torna al pas 2.</p>

*Taula 17: Cas d'ús Modificar tasca*

<b>Cas d'ús</b>	Eliminar tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar tasques.
<b>Descripció</b>	L'usuari elimina una tasca.
<b>Precondició</b>	L'usuari està loguejat i participa a la tasca.
<b>Postcondició</b>	La tasca s'elimina del sistema local.
<b>Curs principal</b>	<p>1. L'usuari indica que vol eliminar una tasca.</p> <p>2. El sistema demana confirmació a l'usuari.</p> <p>3. L'usuari confirma l'acció.</p> <p>4. El sistema elimina la tasca a nivell local, l'alerta corresponent i totes les seves relacions. Crida el cas d'ús <i>Enviar tasques</i>.</p>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció.</p> <p>3A.1. Fi del cas d'ús.</p>

*Taula 18: Cas d'ús Eliminar tasca*

<b>Cas d'ús</b>	Veure informació tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure la informació corresponent a una tasca.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra la informació de la tasca especificada.
<b>Curs principal</b>	<p>1. L'usuari indica al sistema que vol veure la informació d'una tasca.</p> <p>2. El sistema mostra la informació corresponent.</p>
<b>Cursos alternatius</b>	

*Taula 19: Cas d'ús Veure informació tasca*

<b>Cas d'ús</b>	Afegir participant a una tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació tasca.
<b>Descripció</b>	L'usuari afegeix un o més usuaris o grups com a participants d'una tasca.

<b>Precondició</b>	L'usuari està loguejat i participa a la tasca.
<b>Postcondició</b>	Els usuaris o grups s'afegeixen com a participants de la tasca a nivell local.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol afegir participants a la tasca.</li> <li>2. El sistema demana a l'usuari els nous participants.</li> <li>3. L'usuari introdueix la informació demanada.</li> <li>4. El sistema afegeix els usuaris i grups com a participants a nivell local. Crida el cas d'ús <i>Enviar tasques</i>.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p>

*Taula 20: Cas d'ús Afegir participant a una tasca*

<b>Cas d'ús</b>	Eliminar participant d'una tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació tasca.
<b>Descripció</b>	L'usuari elimina un usuari o grup com a participant d'una tasca.
<b>Precondició</b>	L'usuari està loguejat i participa a la tasca.
<b>Postcondició</b>	L'usuari o grup deixa de participar a la tasca a nivell local.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol eliminar un determinat usuari o grup de la tasca.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema elimina el participant a nivell local i, si no queden participants, s'elimina la tasca. Crida el cas d'ús <i>Enviar tasques</i>.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.1. Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció.</p> <p>3A.1. Fi del cas d'ús.</p>

*Taula 21: Cas d'ús Eliminar participant d'una tasca*

<b>Cas d'ús</b>	Afegir precedent a una tasca
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació tasca.
<b>Descripció</b>	L'usuari afegeix una o més tasques com a precedents d'una tasca.
<b>Precondició</b>	L'usuari està loguejat i participa a la tasca.
<b>Postcondició</b>	Les tasques s'afegeixen com a precedents a nivell local.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol afegir precedents a la tasca.</li> <li>2. El sistema mostra a l'usuari les possibles tasques precedents.</li> <li>3. L'usuari escull les tasques precedents.</li> <li>4. El sistema afegeix les tasques com a precedents a nivell local. Crida el cas d'ús <i>Enviar tasques</i>.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació.</p> <p>*A.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 tasques que puguin ser precedents de la tasca que es tracta.</p> <p>2A.2. Fi del cas d'ús.</p>

*Taula 22: Cas d'ús Afegir precedent a una tasca*

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

*Taula 23: Cas d'ús Eliminar precedent d'una tasca*

<b>Cas d'ús</b>	Llistar tasques
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure el llistat de tasques.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra el llistat de tasques de l'usuari.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari es logueja a l'aplicació o indica explícitament que vol veure les tasques (les seves o totes les dels seus calendaris).</li> <li>2. El sistema mostra la llista de tasques demanada.</li> </ol>
<b>Cursos alternatius</b>	

*Taula 24: Cas d'ús Llistar tasques*

<b>Cas d'ús</b>	Filtrar llistat de tasques
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure només certes tasques del llistat.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra el llistat de tasques filtrat pels paràmetres determinats per l'usuari.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema que vol filtrar el llistat.</li> <li>2. El sistema demana els filtres a l'usuari.</li> <li>3. L'usuari introdueix la informació.</li> <li>4. El sistema mostra el nou llistat.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.</p>

*Taula 25: Cas d'ús Filtrar llistat de tasques*

<b>Cas d'ús</b>	Recordar tasca
<b>Actor</b>	Sistema local
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	El sistema envia una notificació a l'usuari per recordar-li que una certa tasca és a prop de començar.
<b>Precondició</b>	La tasca té com a estat "No començada" i la data i hora d'inici és posterior o igual a la data i hora actual
<b>Postcondició</b>	L'usuari rep una notificació recordant-li que la tasca és a prop de començar.
<b>Curs principal</b>	1. El sistema local genera la notificació i l'envia a l'usuari. 2. El sistema operatiu mostra la notificació a la barra de notificacions.
<b>Cursos alternatius</b>	

Taula 26: Cas d'ús Recordar tasca

## 1.4 Sincronització

<b>Cas d'ús</b>	Sincronitzar calendaris
<b>Actor</b>	Sistema local
<b>Descripció</b>	El sistema ha de sincronitzar els canvis realitzats amb el servidor i la resta d'usuaris que participen en els calendaris.
<b>Precondició</b>	-
<b>Postcondició</b>	Els canvis realitzats es veuen reflectits al servidor i, eventualment, als dispositius de la resta d'usuaris participants. Els canvis realitzats pels altres usuaris es veuen reflectits al dispositiu mòbil.
<b>Curs principal</b>	1. El sistema local inicia el procés de sincronització. 2. El sistema local recupera els credencials del compte i demana al servidor la informació que s'ha de sincronitzar a nivell local (canvis des de l'última sincronització si és una sincronització normal, o bé tota la informació de l'usuari si és una sincronització completa). 3. El sistema remot recupera la informació demanada i la retorna al sistema local. 4. El sistema local verifica la informació rebuda i desa els canvis. 5. El sistema local recupera la informació modificada des de l'última sincronització i l'envia al servidor. 6. El sistema remot revisa i desa els canvis, genera els esdeveniments corresponents i, si troba algun conflicte, genera un <i>meeting</i> o un <i>pending approval</i> per resoldre'l. 7. El sistema remot retorna al sistema local la confirmació dels canvis fets i els <i>meetings</i> i <i>pending approvals</i> creats. 8. El sistema local marca els elements enviats com a no modificats i desa els <i>meetings</i> i <i>pending approvals</i> creats si és necessari. 9. El sistema local actualitza la data de l'última sincronització.
<b>Cursos alternatius</b>	*A. No es detecta connexió amb el servidor o hi ha un error de connexió. *A.1. Fi del cas d'ús.  2A. Els credencials de l'usuari no són correctes. 2A.1. El sistema remot notifica al sistema local.

	<p>2A.2. El sistema local invalida els credencials. 2A.3. Fi del cas d'ús.</p> <p>4A. La informació rebuda no és correcte o hi ha un error desant els canvis. 4A.1. Fi del cas d'ús.</p> <p>6A. La informació rebuda pel sistema remot no és correcta o hi ha algun error desant els canvis. 6A.1. El sistema remot notifica al sistema local. 6A.2. Fi del cas d'ús.</p> <p>7A. La informació retornada pel sistema remot no és correcta o bé indica que s'ha produït un error. 7A.1. Fi del cas d'ús.</p> <p>8A. Hi ha un error realitzant els canvis. 8A.1. Fi del cas d'ús.</p>
--	---

*Taula 27: Cas d'ús Sincronitzar calendaris*

## Correcció

Durant el desenvolupament del projecte s'han realitzat canvis en el cas d'ús de sincronitzar calendaris, convertint-se en tres casos d'ús diferents.

<b>Cas d'ús</b>	Sincronització total
<b>Actor</b>	Usuari
<b>Descripció</b>	L'usuari vol que es recuperi tota la informació relacionada amb ell (sincronització completa) i que s'enviïn els canvis fets per ell.
<b>Precondició</b>	-
<b>Postcondició</b>	Els canvis realitzats es veuen reflectits al servidor i, eventualment, als dispositius de la resta d'usuaris participants. Els canvis realitzats pels altres usuaris es veuen reflectits al dispositiu mòbil.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema local que vol que es realitzi una sincronització completa.</li> <li>2. El sistema local recupera els credencials del compte i demana al servidor tota la informació corresponent a l'usuari.</li> <li>3. El sistema remot recupera la informació demanada i la retorna al sistema local.</li> <li>4. El sistema local verifica la informació rebuda i desa els canvis.</li> <li>5. El sistema local recupera la informació modificada des de l'última sincronització i l'envia al servidor.</li> <li>6. El sistema remot revisa i desa els canvis, genera els esdeveniments corresponents i, si troba algun conflicte, genera un <i>meeting</i> o un <i>pending approval</i> per resoldre'l.</li> <li>7. El sistema remot retorna al sistema local la confirmació dels canvis fets i els <i>meetings</i> i <i>pending approvals</i> creats.</li> <li>8. El sistema local marca els elements enviats com a no modificats i desa els <i>meetings</i> i <i>pending approvals</i> creats si és necessari.</li> <li>9. El sistema local notifica a l'usuari que la sincronització s'ha realitzat correctament.</li> </ol>
<b>Cursos alternatius</b>	*A. No es detecta connexió amb el servidor o hi ha un error de connexió.

	<p>*A.1. El sistema local notifica a l'usuari. *A.1. Fi del cas d'ús.</p> <p>2A. Els credencials de l'usuari no són correctes. 2A.1. El sistema remot notifica al sistema local. 2A.2. El sistema local invalida els credencials i notifica a l'usuari. 2A.3. Fi del cas d'ús.</p> <p>4A. La informació rebuda no és correcta o hi ha un error desant els canvis. 4A.1. El sistema local notifica a l'usuari. 4A.2. Fi del cas d'ús.</p> <p>6A. La informació rebuda pel sistema remot no és correcta o hi ha algun error desant els canvis. 6A.1. El sistema remot notifica al sistema local. 6A.2. El sistema local notifica a l'usuari. 6A.3. Fi del cas d'ús.</p> <p>7A. La informació retornada pel sistema remot no és correcta o bé indica que s'ha produït un error. 7A.1. El sistema local notifica a l'usuari. 7A.2. Fi del cas d'ús.</p> <p>8A. Hi ha un error realitzant els canvis. 8A.1. El sistema local notifica a l'usuari. 8A.2. Fi del cas d'ús.</p>
--	---

*Taula 28: Cas d'ús Sincronització total*

<b>Cas d'ús</b>	Enviar tasques
<b>Actor</b>	Sistema local
<b>Descripció</b>	El sistema envia al servidor els canvis realitzats a nivell local per tal que aquests siguin sincronitzats amb la resta d'usuaris.
<b>Precondició</b>	-
<b>Postcondició</b>	Els canvis realitzats es veuen reflectits al servidor i, eventualment, als dispositius de la resta d'usuaris participants.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. El sistema local inicia el procés d'enviar tasques.</li> <li>2. El sistema local recupera els credencials del compte i la informació modificada des de l'última sincronització i l'envia al servidor.</li> <li>3. El sistema remot revisa i desa els canvis, genera els esdeveniments corresponents i, si troba algun conflicte, genera un <i>meeting</i> o un <i>pending approval</i> per resoldre'l.</li> <li>4. El sistema remot retorna al sistema local la confirmació dels canvis fets i els <i>meetings</i> i <i>pending approvals</i> creats.</li> <li>5. El sistema local marca els elements enviats com a no modificats i desa els <i>meetings</i> i <i>pending approvals</i> creats si és necessari.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. No es detecta connexió amb el servidor o hi ha un error de connexió.</p> <p>*A.1. El sistema local ajorna l'execució del cas d'ús. S'executarà quan es detecti connexió a Internet o, en cas que n'hi hagi, al cap de 30 segons.</p>

	<p>3A. Els credencials de l'usuari no són correctes.  3A.1. El sistema remot notifica al sistema local.  3A.2. El sistema local invalida els credencials i ajorna l'execució del cas d'ús durant 30 segons.</p> <p>3B. La informació rebuda pel sistema remot no és correcta o hi ha un error desant els canvis.  3B.1. El sistema remot notifica al sistema local.  3B.2. El sistema local ajorna l'execució del cas d'ús durant 30 segons.</p> <p>4A. La informació retornada pel sistema remot no és correcta o bé indica que s'ha produït un error.  4A.1. El sistema local ajorna l'execució del cas d'ús durant 30 segons.</p> <p>5A. Hi ha un error realitzant els canvis.  5A.1. El sistema local ajorna l'execució del cas d'ús durant 30 segons.</p>
--	--

*Taula 29: Cas d'ús Enviar tasques*

<b>Cas d'ús</b>	Demandar canvis
<b>Actor</b>	Sistema local
<b>Descripció</b>	El sistema local demana al servidor tots els canvis realitzats des de l'última sincronització.
<b>Precondició</b>	-
<b>Postcondició</b>	Els canvis realitzats pels altres usuaris es veuen reflectits al dispositiu mòbil.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. El sistema local inicia el procés de demanar canvis.</li> <li>2. El sistema local recupera els credencials del compte i l'última data de sincronització i demana la informació al servidor.</li> <li>3. El sistema remot verifica els credencials rebuts, recupera la informació demanada i la retorna al sistema local.</li> <li>4. El sistema local verifica la informació rebuda i desa els canvis.</li> <li>5. El sistema local actualitza la data de l'última sincronització.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. No es detecta connexió amb el servidor o hi ha un error de connexió.  *A.1. El sistema local ajorna l'execució del cas d'ús. S'executarà quan es detecti connexió a Internet o, en cas que n'hi hagi, al cap de 30 segons.</p> <p>3A. Els credencials de l'usuari no són correctes.  3A.1. El sistema remot notifica al sistema local.  3A.2. El sistema local invalida els credencials i ajorna l'execució del cas d'ús durant 30 segons.</p> <p>4A. La informació rebuda no és correcta o hi ha un error desant els canvis.  4A.1. El sistema local invalida els credencials i ajorna l'execució del cas d'ús durant 30 segons.</p>

*Taula 30: Cas d'ús Demandar canvis*

## 1.5 Gestió d'esdeveniments, votacions i preferències

<b>Cas d'ús</b>	Llistar esdeveniments
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure el llistat d'esdeveniments ocorreguts.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostra el llistat d'esdeveniments de l'usuari.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure el seu llistat d'esdeveniments. 2. El sistema mostra el llistat a l'usuari.
<b>Cursos alternatius</b>	

*Taula 31: Cas d'ús Llistar esdeveniments*

<b>Cas d'ús</b>	Llistar meetings (votacions)
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure els meetings en els que participa.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostra el llistat de meetings de l'usuari.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure el llistat dels seus meetings. 2. El sistema mostra el llistat a l'usuari.
<b>Cursos alternatius</b>	

*Taula 32: Cas d'ús Llistar meetings (votacions)*

<b>Cas d'ús</b>	Veure informació meeting
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure la informació d'un meeting en el que participa.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostra la informació del meeting especificat.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure la informació d'un meeting determinat. 2. El sistema mostra la informació a l'usuari.
<b>Cursos alternatius</b>	

*Taula 33: Cas d'ús Veure informació meeting*

<b>Cas d'ús</b>	Votar en un meeting
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació meeting.
<b>Descripció</b>	L'usuari vol votar en un meeting en el que participa.



<b>Precondició</b>	L'usuari està loguejat, participa en el meeting i no ha votat encara.
<b>Postcondició</b>	Es desfa el vot de l'usuari i, si la votació ha finalitzat, es marca el meeting com a inactiu i es realitzen els canvis relacionats amb el resultat obtingut.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema el seu vot.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema desfa el vot. Si tots els participants han votat o un dels resultats ha obtingut suficients vots per a ser determinada guanyadora, es marca el meeting com a inactiu, es realitzen els canvis determinats pel resultat i es notifica als usuaris participants. Si hi ha un empat, es repeteix el meeting.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.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>

*Taula 34: Cas d'ús Votar en un meeting*

<b>Cas d'ús</b>	Eliminar vot d'un meeting
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació meeting.
<b>Descripció</b>	L'usuari vol eliminar el seu vot en un meeting.
<b>Precondició</b>	L'usuari està loguejat, participa en el meeting i ha votat.
<b>Postcondició</b>	S'elimina el vot de l'usuari.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema que vol eliminar el seu vot en un determinat meeting.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema elimina el vot.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.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>

*Taula 35: Cas d'ús Eliminar vot d'un meeting*

<b>Cas d'ús</b>	Llistar pending approvals
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure els pending approvals en els que participa.

<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostra el llistat de pending approvals de l'usuari.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure el llistat dels seus pending approvals. 2. El sistema mostra el llistat a l'usuari.
<b>Cursos alternatius</b>	

*Taula 36: Cas d'ús Llistar pending approvals*

<b>Cas d'ús</b>	Veure informació pending approval
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure la informació d'un pending approval en el que participa.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostra la informació del pending approval especificat.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure la informació d'un pending approval determinat. 2. El sistema mostra la informació a l'usuari.
<b>Cursos alternatius</b>	

*Taula 37: Cas d'ús Veure informació pending approval*

<b>Cas d'ús</b>	Acceptar/Rebutjar pending approval
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació pending approval.
<b>Descripció</b>	L'usuari vol acceptar o rebutjar un pending approval en el que participa.
<b>Precondició</b>	L'usuari està loguejat, participa en el pending approval i no l'ha acceptat o rebutjat encara.
<b>Postcondició</b>	Es desfa la decisió de l'usuari i, si la votació ha finalitzat, s'elimina el pending approval i es fan els canvis necessaris en funció del resultat.
<b>Curs principal</b>	1. L'usuari indica que vol acceptar o rebutjar un pending approval. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema desfa la decisió. Si tots els participants han votat o un dels resultats ha obtingut una quantitat de vots superior al 50%, s'elimina el pending approval, es realitzen els canvis determinats pel resultat i es notifica a tots els participants. Si hi ha un empat, es rebutja el pending approval.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  3A. L'usuari no confirma l'acció. 3A.1. Fi del cas d'ús.

*Taula 38: Cas d'ús Acceptar/Rebutjar pending approval*

<b>Cas d'ús</b>	Veure preferències
<b>Actor</b>	Usuari loguejat.
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure les seves preferències.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	Es mostren les preferències.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure les preferències. 2. El sistema mostra les preferències a l'usuari.
<b>Cursos alternatius</b>	

*Taula 39: Cas d'ús Veure preferències*

<b>Cas d'ús</b>	Modificar preferències
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol definir les seves preferències a l'hora d'utilitzar l'aplicació.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema enregistra els canvis en les preferències i els comença a aplicar.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol modificar les preferències. 2. El sistema demana a l'usuari les dades pertinents. 3. L'usuari introdueix la informació demanada. 4. El sistema verifica les dades introduïdes. 5. El sistema enregistra els canvis.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Les dades no són correctes. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Torna al pas 2.

*Taula 40: Cas d'ús Modificar preferències*

## 1.6 Gestió de grups

<b>Cas d'ús</b>	Crear grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar grups.
<b>Descripció</b>	L'usuari crea un grup.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El grup es crea correctament.
<b>Curs principal</b>	1. L'usuari indica que vol crear un nou grup. 2. El sistema demana la informació necessària a l'usuari. 3. L'usuari introdueix la informació. 4. El sistema verifica les dades introduïdes. 5. El sistema crea el nou grup i notifica als membres afegits.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.

	4A. Les dades no són correctes. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Torna al pas 2.
--	---

*Taula 41: Cas d'ús Crear grup*

<b>Cas d'ús</b>	Canviar nom del grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari modifica el nom del grup.
<b>Precondició</b>	L'usuari està loguejat i és l'administrador del grup.
<b>Postcondició</b>	El nom es modifica correctament.
<b>Curs principal</b>	1. L'usuari indica que vol modificar el nom del grup. 2. El sistema demana a l'usuari el nou nom. 3. L'usuari introdueix la informació. 4. El sistema verifica el nom introduït. 5. El sistema modifica el nom del grup.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Ja existeix un grup amb aquest nom. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Fi del cas d'ús.

*Taula 42: Canviar nom del grup*

<b>Cas d'ús</b>	Afegir participant a un grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari afegeix un o més usuaris com a membres d'un grup.
<b>Precondició</b>	L'usuari està loguejat i és l'administrador del grup.
<b>Postcondició</b>	Els usuaris escollits passen a formar part del grup.
<b>Curs principal</b>	1. L'usuari indica 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 notifica a la resta de membres que s'han afegit usuaris.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  4A. Les dades no són correctes. 4A.1. El sistema notifica l'error a l'usuari. 4A.2. Torna al pas 2.

*Taula 43: Afegir participant a un grup*

<b>Cas d'ús</b>	Eliminar participant d'un grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari expulsa un membre d'un grup.
<b>Precondició</b>	L'usuari està loguejat i té permisos d'administrador del grup.
<b>Postcondició</b>	El membre escollit deixa de formar part del grup.

<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol eliminar un determinat usuari del grup.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema elimina l'usuari com a membre del grup, el notifica que n'ha estat eliminat i notifica a la resta de membres que s'ha eliminat aquest usuari.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.</p> <p>3A. L'usuari no confirma l'acció. 3A.1. Fi del cas d'ús.</p>

*Taula 44: Cas d'ús Eliminar participant d'un grup*

<b>Cas d'ús</b>	Sortir d'un grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari surt d'un grup. Si és l'últim membre del grup, el grup s'elimina. Si no és l'últim membre del grup però n'és l'administrador, s'escull un nou administrador.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	L'usuari deixa de participar al grup.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica que vol sortir del grup.</li> <li>2. El sistema demana confirmació a l'usuari.</li> <li>3. L'usuari confirma l'acció.</li> <li>4. El sistema elimina l'usuari del grup. Si és l'últim membre del grup, elimina el grup del sistema. Si no és l'últim membre del grup però n'és l'administrador, s'escull un nou administrador.</li> </ol>
<b>Cursos alternatius</b>	<p>*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.</p>

*Taula 45: Cas d'ús Sortir d'un grup*

<b>Cas d'ús</b>	Veure informació d'un grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure la informació corresponent a un grup.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra la informació del grup especificat.
<b>Curs principal</b>	<ol style="list-style-type: none"> <li>1. L'usuari indica al sistema que vol veure la informació d'un grup.</li> <li>2. El sistema mostra la informació del grup.</li> </ol>
<b>Cursos alternatius</b>	

*Taula 46: Cas d'ús Veure informació d'un grup*

<b>Cas d'ús</b>	Llistar grups
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	
<b>Descripció</b>	L'usuari vol veure el llistat de grups als que pertany.
<b>Precondició</b>	L'usuari està loguejat.

<b>Postcondició</b>	El sistema mostra el llistat de grups que l'usuari n'és membre.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure el llistat de grups. 2. El sistema mostra la llista de grups als que pertany l'usuari.
<b>Cursos alternatius</b>	

*Taula 47: Cas d'ús Llistar grups*

<b>Cas d'ús</b>	Nomenar administrador
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari vol nomenar administrador del grup un altre usuari.
<b>Precondició</b>	L'usuari està loguejat i és l'administrador actual del grup.
<b>Postcondició</b>	L'usuari indicat passa a ser l'administrador del grup i l'actual perd aquest rol.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol que un determinat usuari sigui l'administrador del grup. 2. El sistema demana confirmació a l'usuari. 3. L'usuari confirma l'acció. 4. El sistema desa com a administrador l'usuari indicat.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  3A. L'usuari no confirma l'acció. 3A.1. Fi del cas d'ús.

*Taula 48: Cas d'ús Nomenar administrador*

<b>Cas d'ús</b>	Cercar grup
<b>Actor</b>	Usuari loguejat
<b>Casos d'ús relacionats</b>	Llistar grups.
<b>Descripció</b>	L'usuari vol cercar un grup a partir d'una certa cadena de caràcters. Es mostraran els grups que continguin en el seu nom la cadena de caràcters determinada per l'usuari.
<b>Precondició</b>	L'usuari està loguejat.
<b>Postcondició</b>	El sistema mostra el llistat de grups que contenen una certa cadena de caràcters al seu nom.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol cercar un grup. 2. El sistema demana a l'usuari la cadena a cercar. 3. L'usuari introdueix la informació. 4. El sistema mostra el llistat de grups que contenen la cadena de caràcters al seu nom.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  3A. La cadena introduïda no és vàlida. 3A.1. El sistema notifica l'error a l'usuari. 3A.2. Torna al pas 2.

*Taula 49: Cas d'ús Cercar grup*

<b>Cas d'ús</b>	Enviar/Cancel·lar sol·licitud
<b>Actor</b>	Usuari loguejat.
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari vol enviar una sol·licitud per entrar a un cert grup o, en cas que ja n'hagi enviat una, vol cancel·lar-la.
<b>Precondició</b>	L'usuari està loguejat i no és membre del grup.
<b>Postcondició</b>	La sol·licitud es crea o s'elimina correctament.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol enviar o cancel·lar una sol·licitud. 2. El sistema desa o elimina la sol·licitud i notifica a l'administrador del grup.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.

*Taula 50: Cas d'ús Enviar/Cancel·lar sol·licitud*

<b>Cas d'ús</b>	Llistar sol·licituds
<b>Actor</b>	Usuari loguejat.
<b>Casos d'ús relacionats</b>	Veure informació grup.
<b>Descripció</b>	L'usuari vol veure les sol·licituds pendents d'entrada a un grup que administra.
<b>Precondició</b>	L'usuari està loguejat i és l'administrador del grup.
<b>Postcondició</b>	El sistema mostra el llistat de sol·licituds.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol veure el llistat de sol·licituds d'entrada a un cert grup. 2. El sistema mostra el llistat.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.  2A. No hi ha sol·licituds. 2A.1. El sistema notifica a l'usuari que no hi ha sol·licituds. 2A.2. Fi del cas d'ús.

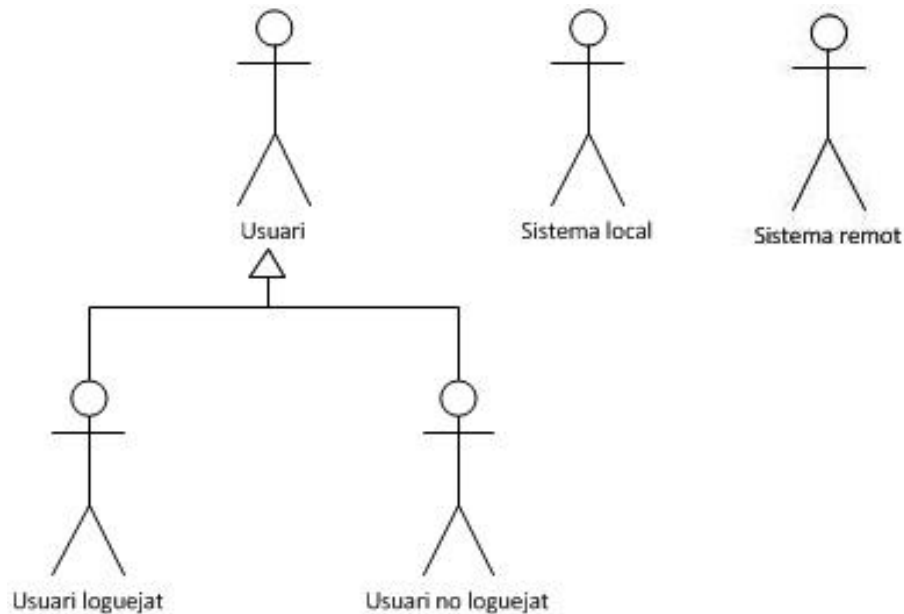
*Taula 51: Cas d'ús Llistar sol·licituds*

<b>Cas d'ús</b>	Acceptar/Rebutjar sol·licitud
<b>Actor</b>	Usuari loguejat.
<b>Casos d'ús relacionats</b>	Llistar sol·licituds.
<b>Descripció</b>	L'usuari vol acceptar o rebutjar una certa sol·licitud d'entrada a un grup.
<b>Precondició</b>	L'usuari està loguejat i és l'administrador del grup al qual pertany la sol·licitud.
<b>Postcondició</b>	La sol·licitud s'elimina correctament i, si és acceptada, l'usuari passa a ser membre del grup.
<b>Curs principal</b>	1. L'usuari indica al sistema que vol acceptar o rebutjar una sol·licitud. 2. El sistema elimina la sol·licitud i notifica al propietari de la sol·licitud si ha estat acceptada o rebutjada. Si ha estat acceptada, s'afegeix l'usuari com a membre del grup i es notifica a la resta de membres.
<b>Cursos alternatius</b>	*A. L'usuari cancel·la l'operació. *A.1. Fi del cas d'ús.

*Taula 52: Cas d'ús Acceptar/Rebutjar sol·licitud*

## 2. Definició dels actors

En aquest apartat es descriuen els diferents actors que intervenen al sistema i la seva jerarquia. Com que algunes tasques són cridades automàticament pel sistema quan es compleixen certes condicions, els sistemes local i remot també seran actors.



*Figura 1: Jerarquia d'actors*



**Usuari:** Actor que representa totes les persones que s'han instal·lat l'aplicació.

*Figura 2: Actor Usuari*



**Usuari no loguejat:** Actor que representa les persones que s'han descarregat l'aplicació però no han entrat amb el seu compte o, directament, encara no tenen compte.

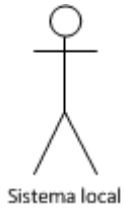
*Figura 3: Actor Usuari no loguejat*



**Usuari loguejat:** Actor que representa les persones que s'han descarregat l'aplicació i s'han identificat amb el seu compte.

*Figura 4: Actor Usuari loguejat*





**Sistema local:** Actor que representa el software de l'aplicació local que, com ja s'ha mencionat anteriorment, actuarà també d'actor en alguns casos d'ús.

*Figura 5: Actor Sistema local*



**Sistema remot:** Actor que representa el software del servidor que, com ja s'ha mencionat anteriorment, actuarà també d'actor en alguns casos d'ús.

*Figura 6: Actor Sistema remot*

### 3. Diagrames de casos d'ús

Els diagrames de casos d'ús descriuen la interacció entre l'usuari i el sistema. Els casos d'ús es classifiquen en els següents subsistemes:

- Gestió de comptes
- Gestió de calendaris
- Gestió de tasques
- Sincronització
- Gestió d'esdeveniments i preferències
- Gestió de grups

#### 3.1 Gestió de comptes i usuaris

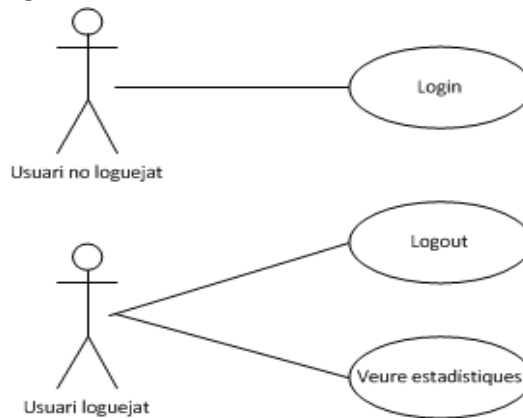


Figura 7: Diagrama casos d'ús comptes

#### 3.2 Gestió de calendaris

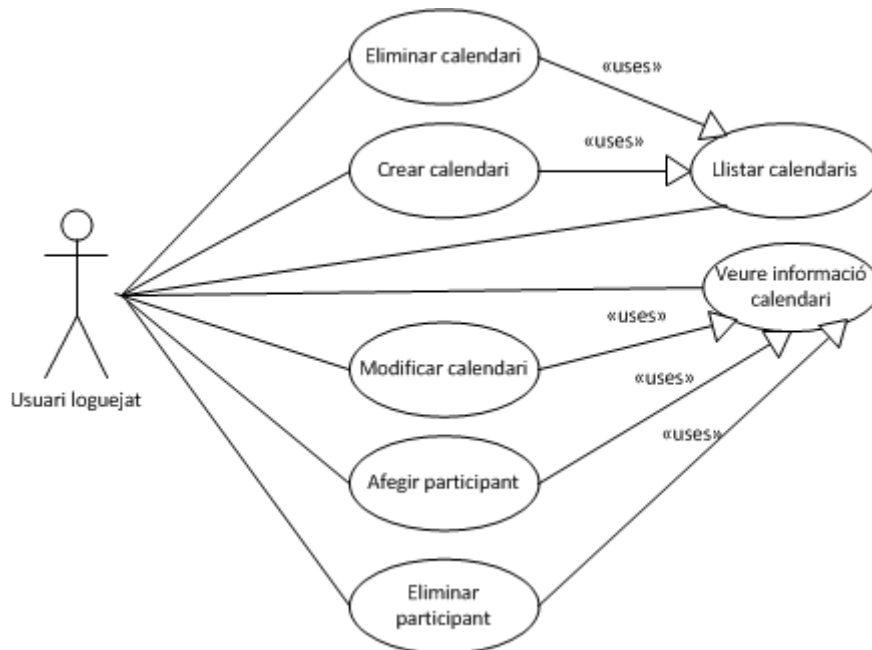


Figura 8: Diagrama casos d'ús calendaris

### 3.3 Gestió de tasques

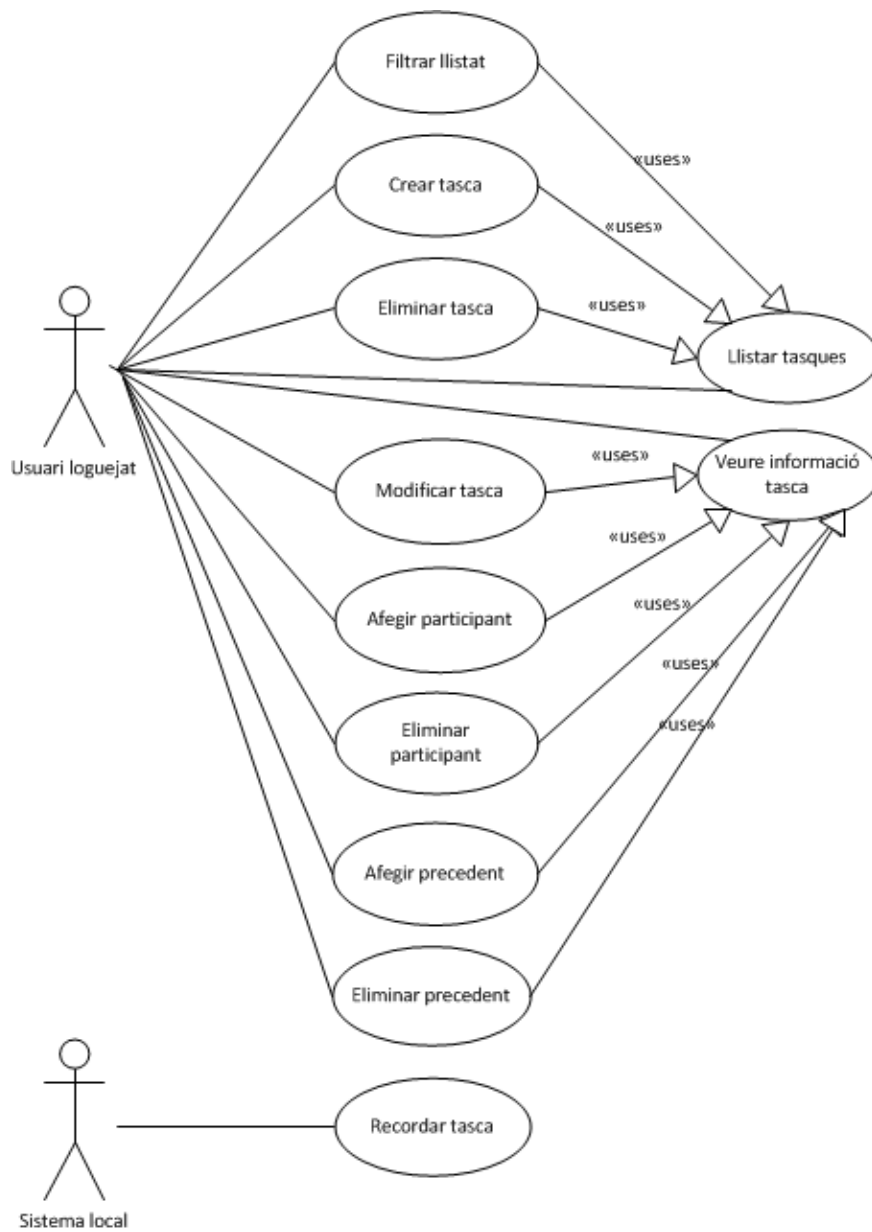


Figura 9: Diagrama casos d'ús tasques

### 3.4 Sincronització

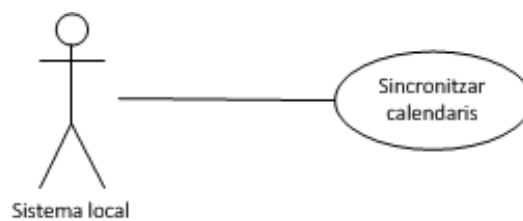


Figura 10: Diagrama casos d'ús sincronització

## Correcció

Durant el desenvolupament del projecte s'han realitzat canvis en el cas d'ús de sincronitzar calendaris, convertint-se en tres casos d'ús diferents.

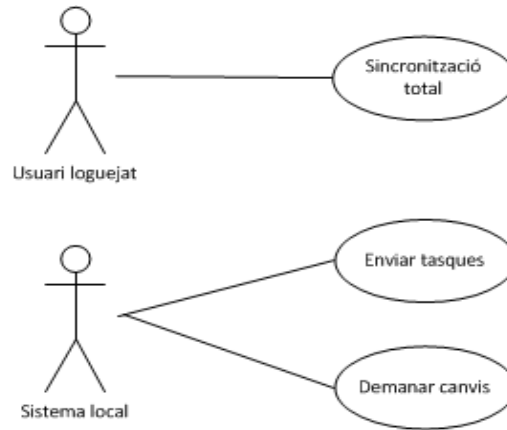


Figura 11: Diagrama casos d'ús sincronització corregit

## 3.5 Gestió d'esdeveniments, votacions i preferències



Figura 12: Diagrama casos d'ús esdeveniments i votacions

### 3.6 Gestió de grups

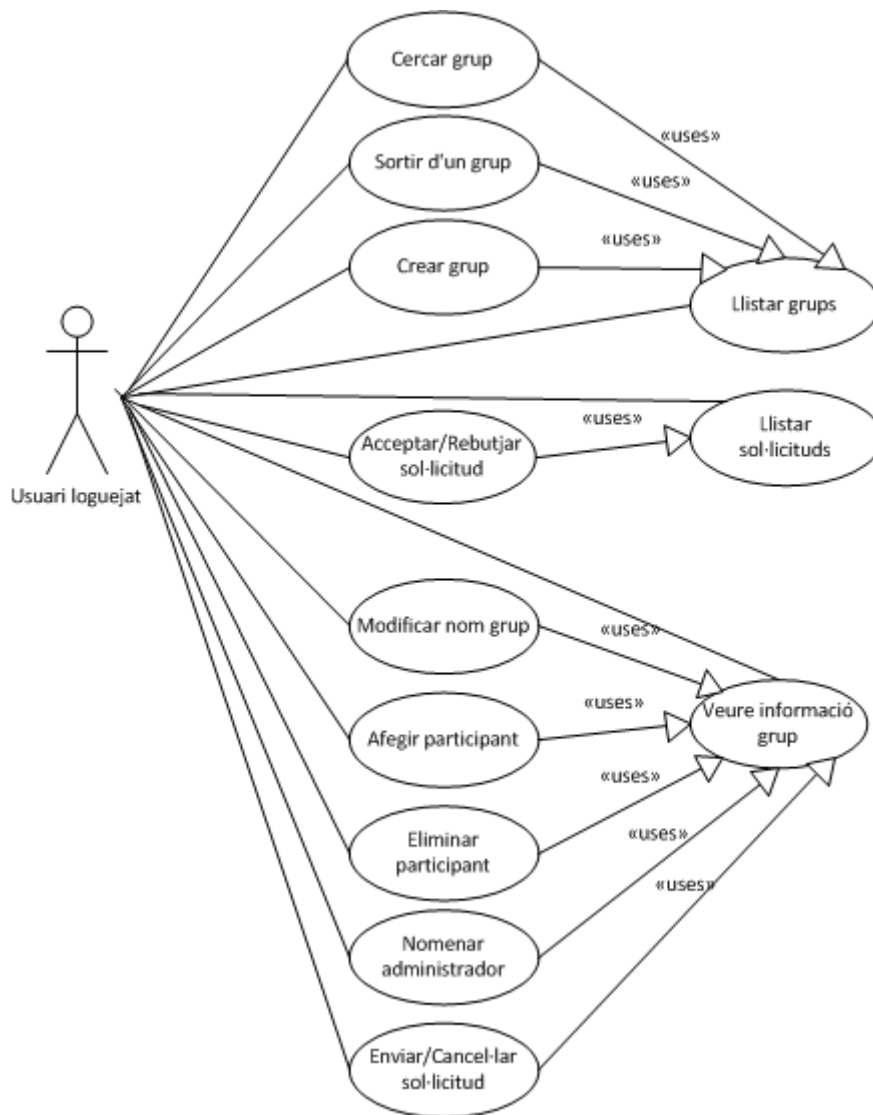


Figura 13: Diagrama casos d'ús grups

## 4. Diagrames de seqüència

Els diagrames de seqüència descriuen la interacció entre l'actor i el sistema de cada cas d'ús, permetent identificar les operacions del sistema en una seqüència temporal.

A continuació es presenten els diagrames de seqüència d'especificació dels casos d'ús presentats anteriorment. A l'etapa de disseny s'expandiran aquests diagrames de seqüència, fent-los més concrets i adaptats a la tecnologia utilitzada.

### 4.1 Gestió de comptes

#### Login

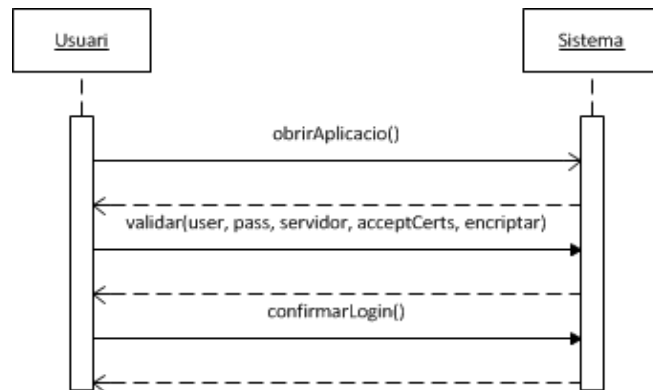


Figura 14: Diagrama seqüència especificació login

#### Logout

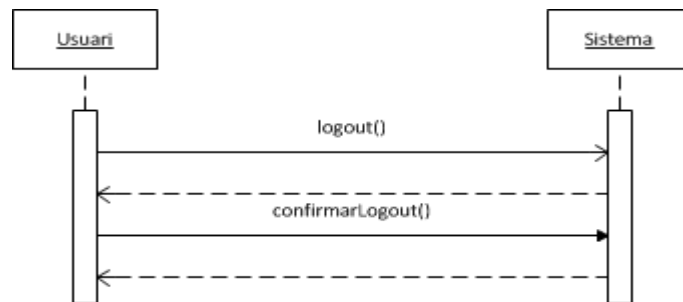


Figura 15: Diagrama seqüència especificació logout

#### Veure estadístiques



Figura 16: Diagrama seqüència especificació veure estadístiques

## 4.2 Gestió de calendaris

### Crear calendari

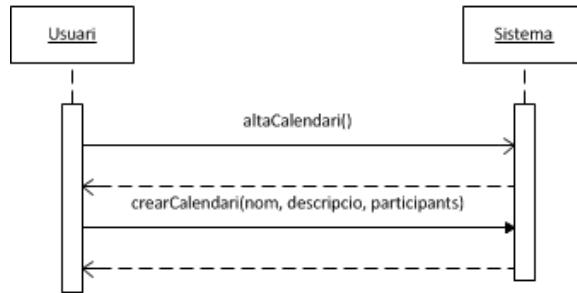


Figura 17: Diagrama seqüència especificació crear calendari

### Modificar calendari

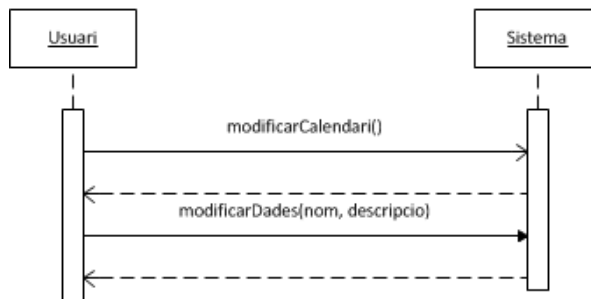


Figura 18: Diagrama seqüència especificació modificar calendari

### Eliminar calendari

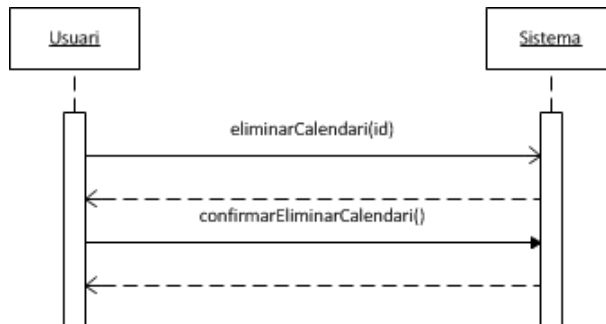


Figura 19: Diagrama seqüència especificació eliminar calendari

### Veure informació calendari



Figura 20: Diagrama seqüència especificació info calendari

### Afegir participant al calendari

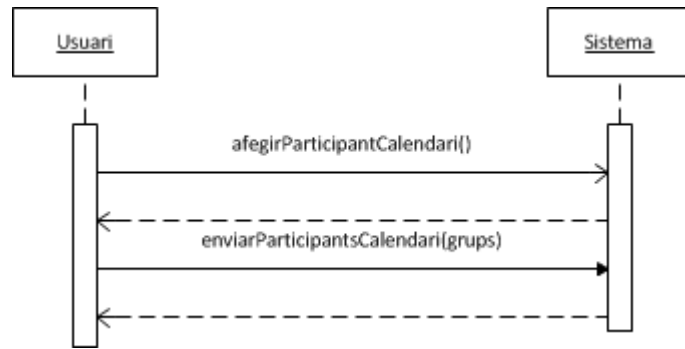


Figura 21: Diagrama seqüència especificació afegir participant calendari

### Eliminar participant del calendari

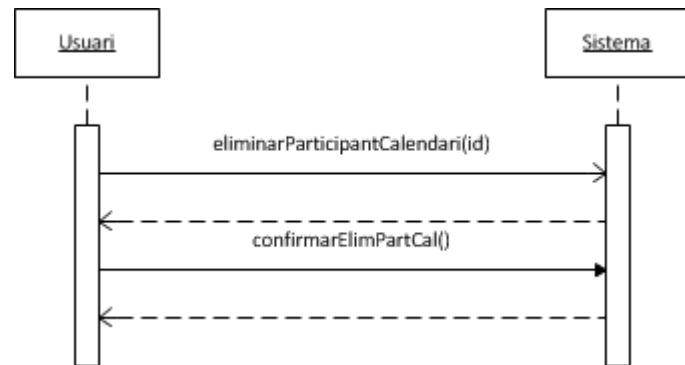


Figura 22: Diagrama seqüència especificació eliminar participant calendari

### Llistar calendaris

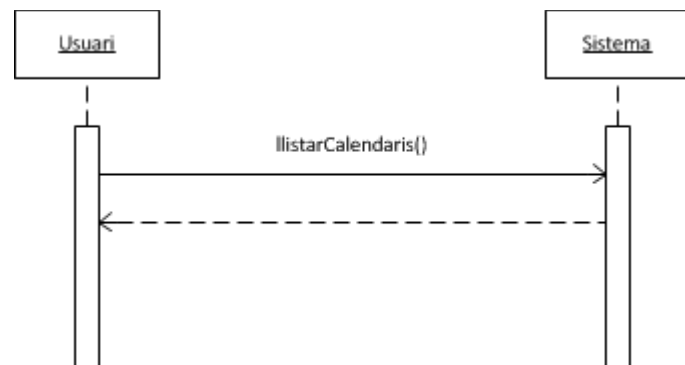


Figura 23: Diagrama seqüència especificació llistar calendaris



## 4.3 Gestió de tasques

### Crear tasca

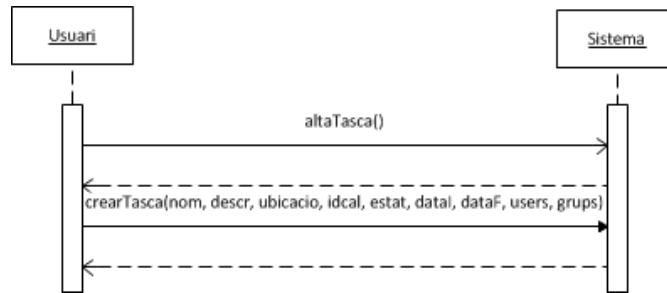


Figura 24: Diagrama seqüència especificació crear tasca

### Modificar tasca

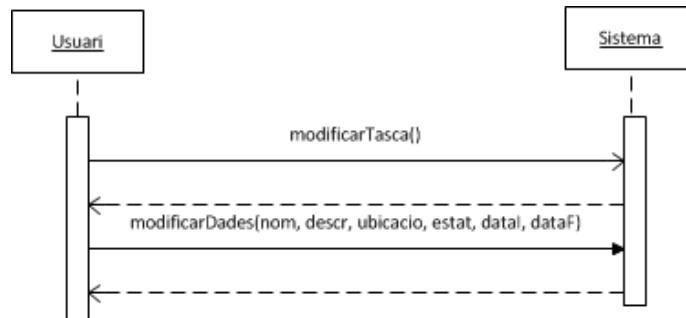


Figura 25: Diagrama seqüència especificació modificar tasca

### Eliminar tasca

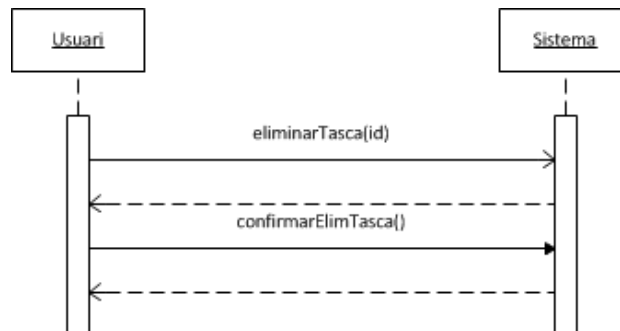


Figura 26: Diagrama seqüència especificació eliminar tasca

### Veure informació tasca

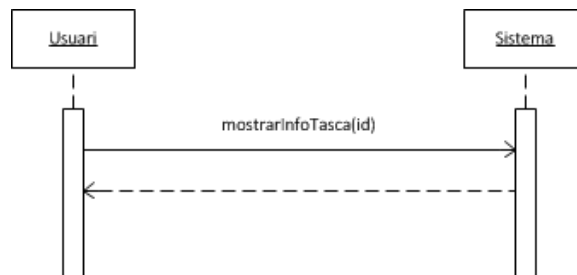


Figura 27: Diagrama seqüència especificació info tasca

### Afegir participant a tasca

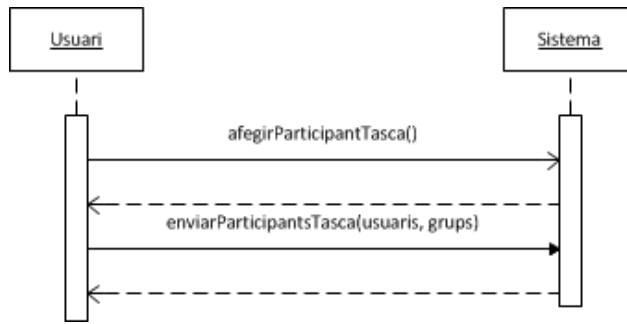


Figura 28: Diagrama seqüència especificació afegir participant tasca

### Eliminar participant d'una tasca

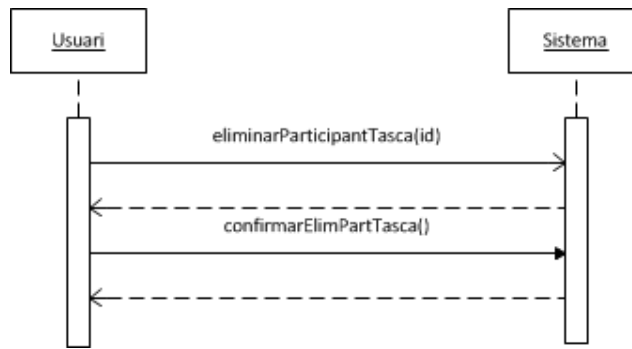


Figura 29: Diagrama seqüència especificació eliminar participant tasca

### Afegir precedent a tasca

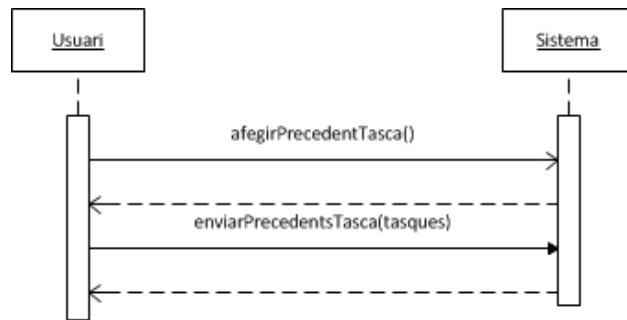


Figura 30: Diagrama seqüència especificació afegir precedent tasca

### Eliminar precedent d'una tasca

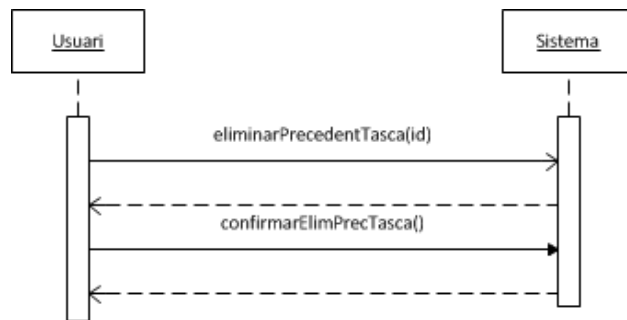


Figura 31: Diagrama seqüència especificació eliminar precedent tasca

### Llistar tasques



Figura 32: Diagrama seqüència especificació l·listar tasques

### Filtrar l·listat de tasques

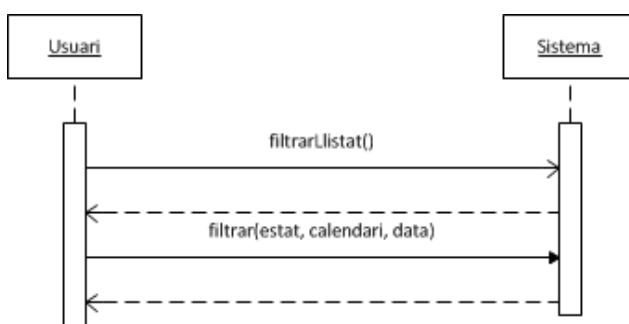


Figura 33: Diagrama seqüència especificació filtrar l·listat

### Recordar tasca

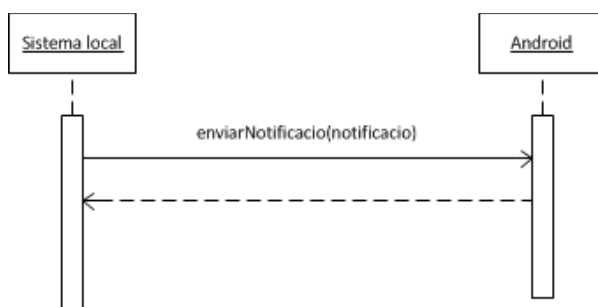


Figura 34: Diagrama seqüència especificació recordar tasca

## 4.4 Sincronització

### Sincronitzar calendaris

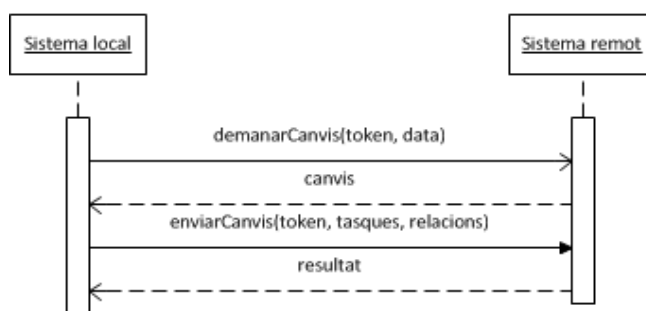


Figura 35: Diagrama seqüència especificació sincronitzar calendaris

## Correcció

Durant el desenvolupament del projecte s'han realitzat canvis en el cas d'ús de sincronitzar calendaris, convertint-se en tres casos d'ús diferents.

### Sincronització total

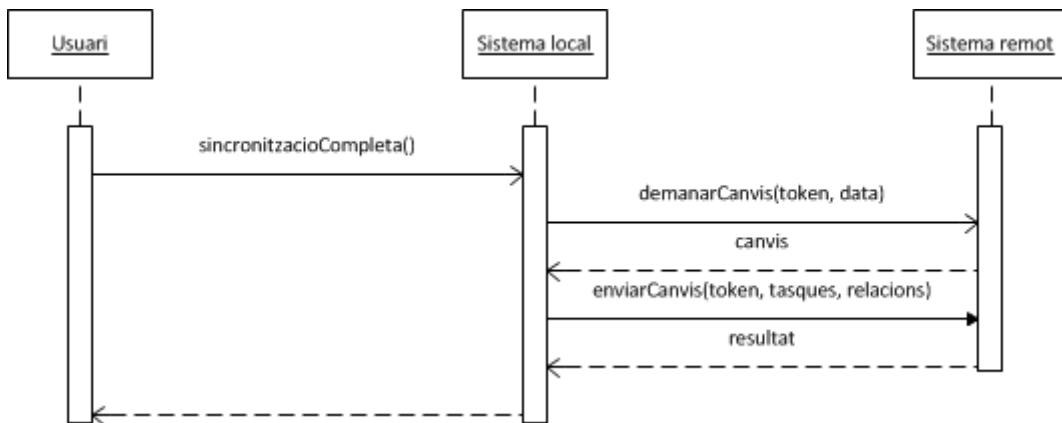


Figura 36: Diagrama seqüència especificació sincronització total

### Enviar tasques

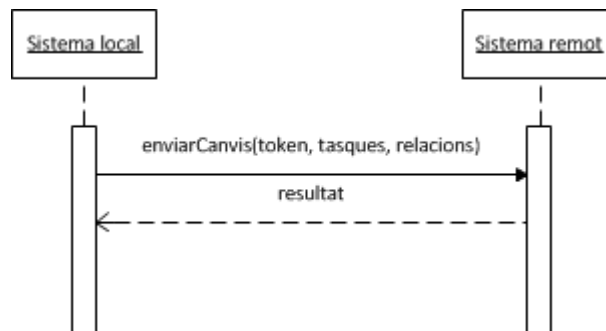


Figura 37: Diagrama seqüència especificació enviar tasques

### Demanar canvis

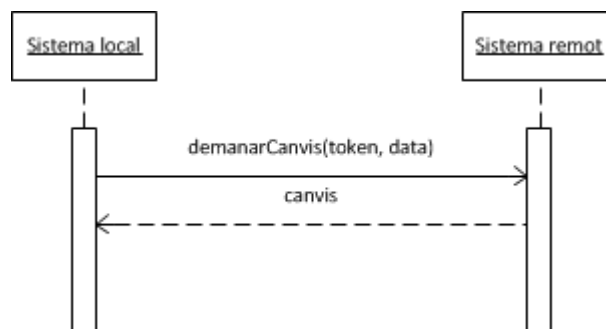


Figura 38: Diagrama seqüència especificació demanar canvis

## 4.5 Gestió d'esdeveniments, votacions i preferències

### Llistar esdeveniments



Figura 39: Diagrama seqüència especificació llistar esdeveniments

### Llistar meetings

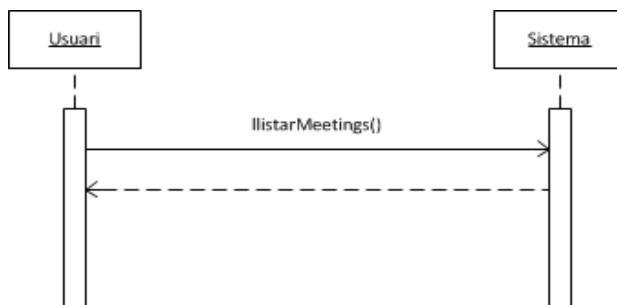


Figura 40: Diagrama seqüència especificació llistar meetings

### Veure informació meeting



Figura 41: Diagrama seqüència especificació info meeting

### Votar en un meeting

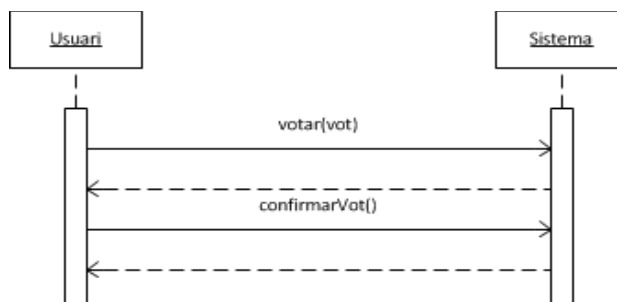


Figura 42: Diagrama seqüència especificació votar en un meeting

### Eliminar vot d'un meeting

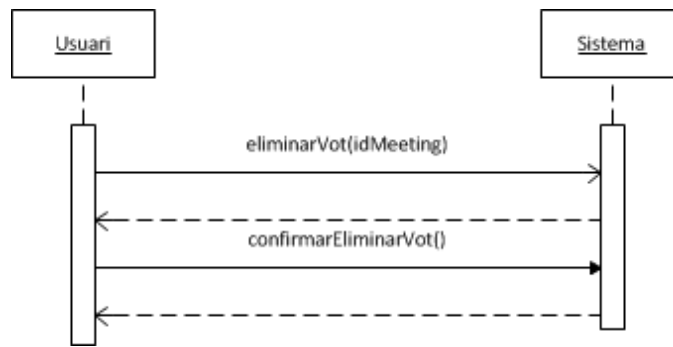


Figura 43: Diagrama seqüència especificació eliminar vot meeting

### Llistar pending approvals

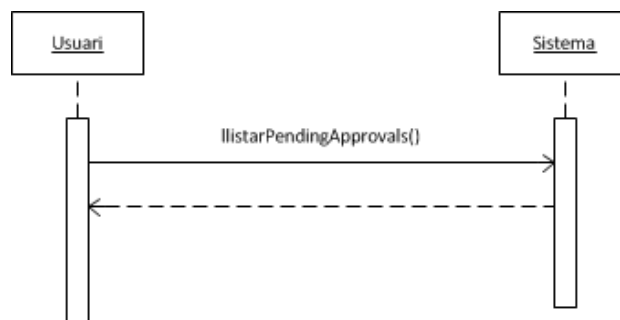


Figura 44: Diagrama seqüència especificació llistar pending approvals

### Veure informació pending approval



Figura 45: Diagrama seqüència especificació info pending approval

### Acceptar/Rebutjar pending approval

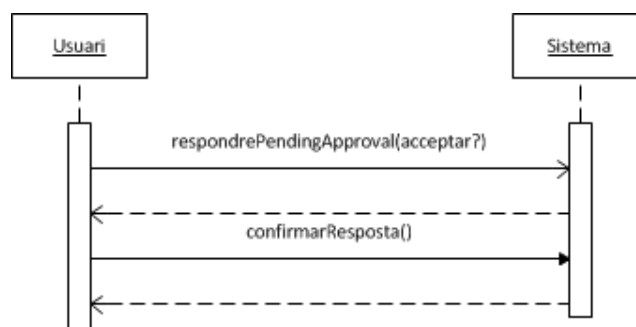


Figura 46: Diagrama seqüència especificació acceptar/rebutjar PA

### Veure preferències

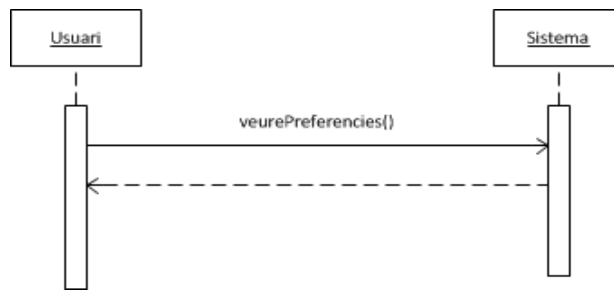


Figura 47: Diagrama seqüència especificació veure preferències

### Modificar preferències

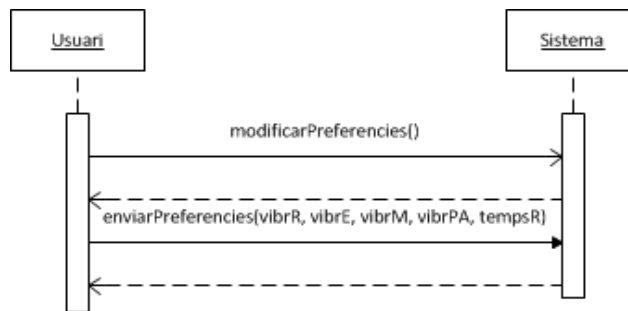


Figura 48: Diagrama seqüència especificació modificar preferències

## 4.6 Gestió de grups

### Crear grup

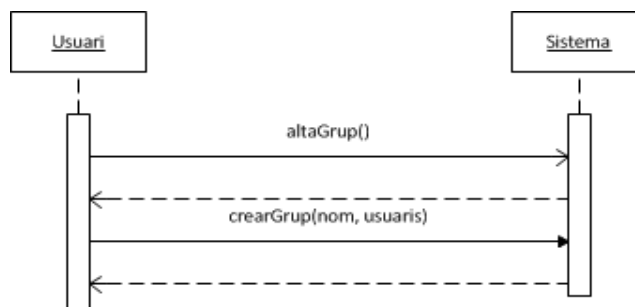


Figura 49: Diagrama seqüència especificació crear grup

### Modificar nom grup

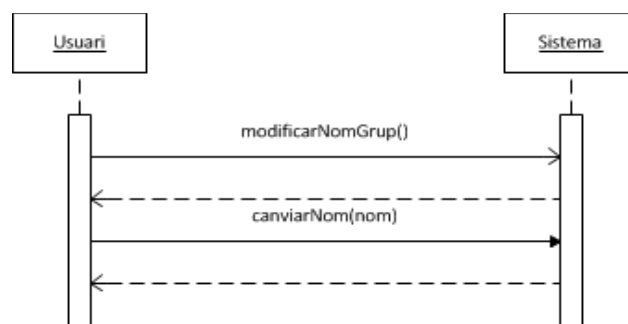


Figura 50: Diagrama seqüència especificació modificar nom grup

### Afegir participant a un grup

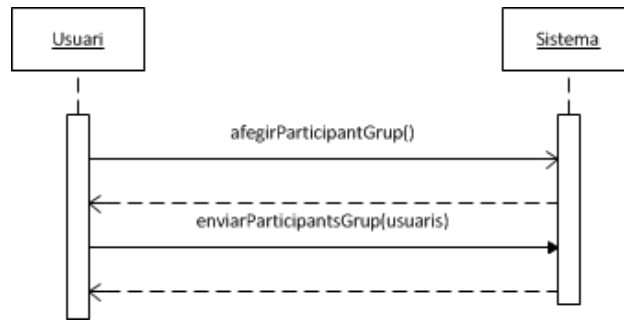


Figura 51: Diagrama seqüència especificació afegir participant grup

### Eliminar participant grup

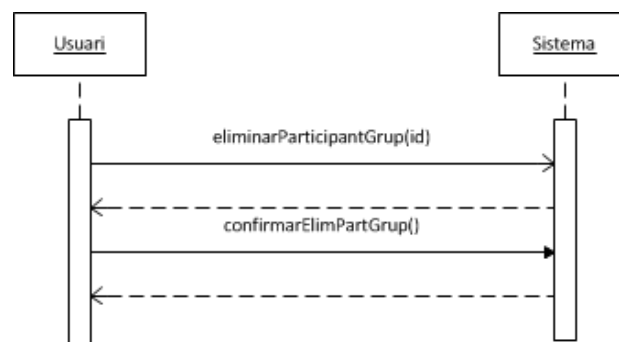


Figura 52: Diagrama seqüència especificació eliminar participant grup

### Sortir d'un grup

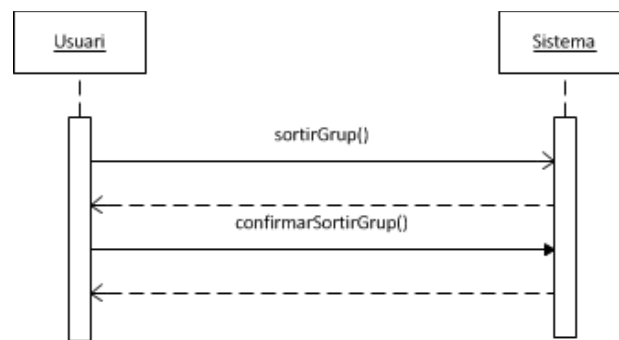


Figura 53: Diagrama seqüència especificació sortir grup

### Veure informació grup



Figura 54: Diagrama seqüència especificació info grup



### Llistar grups

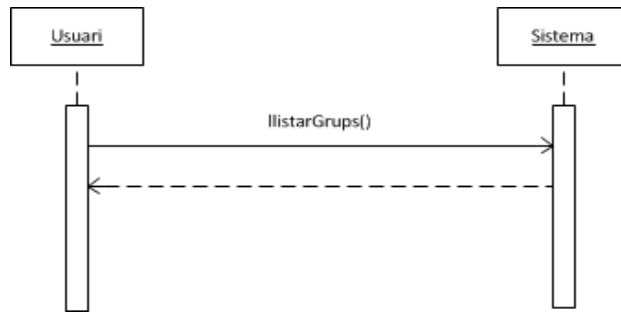


Figura 55: Diagrama seqüència especificació llistar grups

### Nomenar administrador

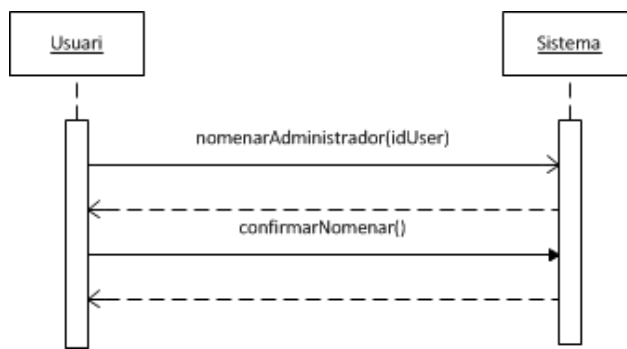


Figura 56: Diagrama seqüència especificació nomenar administrador

### Cercar grup

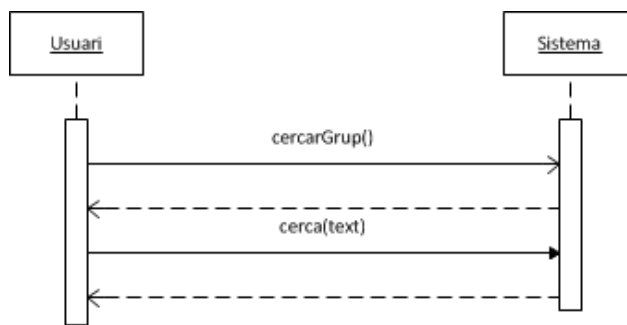


Figura 57: Diagrama seqüència especificació cercar grup

### Enviar/Cancel·lar sol·licitud



Figura 58: Diagrama seqüència especificació enviar/cancel·lar sol·licitud

### Llistar sol·licituds



Figura 59: Diagrama seqüència especificació l·listar sol·licituds

### Acceptar/Rebutjar sol·licitud

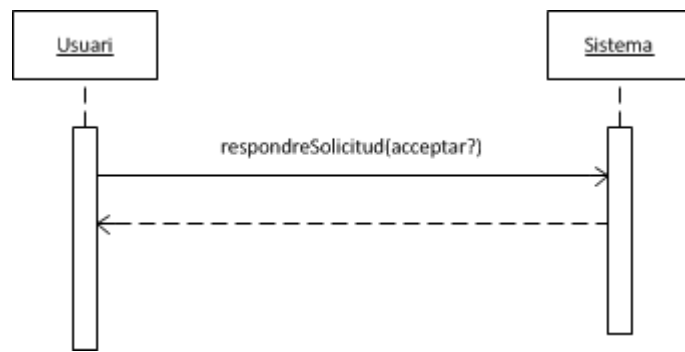


Figura 60: Diagrama seqüència especificació acceptar/rebutjar sol·licitud

## 5. Contractes de les operacions

Els contractes de les operacions descriuen el sistema en termes de quins són els canvis d'estat i quines són les sortides que el sistema proporciona quan s'invoca l'operació. Per cada operació es genera un contracte que informa sobre la semàntica, excepcions, sortides, precondicions i postcondicions.

### 5.1 Gestió de comptes

<b>Operació</b>	obrirAplicacio()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol iniciar l'aplicació.
<b>Cas d'ús</b>	Login.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de login o crida el cas d'ús <i>Llistar tasques</i> .
<b>Sortides</b>	

Taula 53: Operació *obrirAplicacio*

<b>Operació</b>	validar(user, pass, servidor, acceptCerts, encriptar)
<b>Paràmetres</b>	Es passa com a paràmetre la informació necessària per a fer el login i alguns paràmetres sobre com han de ser les connexions: nom d'usuari, contrasenya, URL del servidor i dos booleans indicant si s'han d'acceptar tots els certificats i si s'ha d'encriptar la contrasenya.
<b>Semàntica</b>	Comunica la informació necessària per a crear el nou compte per tal que el sistema la validi.
<b>Cas d'ús</b>	Login.
<b>Excepcions</b>	-Si <i>nom</i> , <i>pass</i> o <i>servidor</i> són buits s'indica que són obligatoris. -Si el servidor no troba l'usuari o la contrasenya és incorrecta s'indica a l'usuari que els valors són incorrectes. -Si no es pot connectar amb el servidor s'indica que hi ha hagut un error en la connexió i que s'hauria de revisar el valor del camp <i>servidor</i> .
<b>Precondicions</b>	
<b>Postcondicions</b>	Les dades han estat validades i es demana confirmació a l'usuari.
<b>Sortides</b>	

Taula 54: Operació *validar*

<b>Operació</b>	confirmarLogin()
<b>Paràmetres</b>	
<b>Semàntica</b>	Confirma al sistema que es pot continuar amb el login.
<b>Cas d'ús</b>	Login.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es desen les dades necessàries a nivell local i es criden els casos d'ús <i>Demandar canvis</i> i <i>Llistar tasques</i> .
<b>Sortides</b>	

Taula 55: Operació *confirmarLogin*

<b>Operació</b>	logout()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol sortir del compte (fer logout).
<b>Cas d'ús</b>	Logout.
<b>Excepcions</b>	
<b>Precondicions</b>	L'usuari està loguejat.
<b>Postcondicions</b>	Es demana confirmació a l'usuari.
<b>Sortides</b>	

*Taula 56: Operació logout*

<b>Operació</b>	confirmarLogout()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol fer logout.
<b>Cas d'ús</b>	Logout.
<b>Excepcions</b>	-Si hi ha un error al fer l'última sincronització de tasques es comunica a l'usuari que hi ha hagut un error i se li demana si vol continuar amb el cas d'ús o no.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'ha cridat el cas d'ús <i>Enviar tasques</i> i s'han eliminat les dades desades al fer login.
<b>Sortides</b>	

*Taula 57: Operació confirmarLogout*

<b>Operació</b>	veureEstadistiques()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es volen veure les estadístiques del compte: nombre de grups, calendaris, tasques, etc.
<b>Cas d'ús</b>	Veure estadístiques.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostren les estadístiques del compte.
<b>Sortides</b>	

*Taula 58: Operació veureEstadistiques*

## 5.2 Gestió de calendaris

<b>Operació</b>	altaCalendari()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol crear un nou calendari.
<b>Cas d'ús</b>	Crear calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de creació de calendaris.
<b>Sortides</b>	

*Taula 59: Operació altaCalendari*

<b>Operació</b>	crearCalendari(nom, descripcio, participants)
<b>Paràmetres</b>	Es passa com a paràmetre la informació necessària per a crear el calendari: nom, descripció i identificadors dels participants.
<b>Semàntica</b>	Comunica la informació necessària per a crear el nou calendari.
<b>Cas d'ús</b>	Crear calendari.

<b>Excepcions</b>	<ul style="list-style-type: none"> <li>-Si el nom és buit s'indica que és obligatori.</li> <li>-Si el nom conté caràcters invàlids s'indica l'error.</li> <li>-Si ja existeix un calendari amb aquest nom s'indica que introdueixi un altre nom.</li> <li>-Si no s'escull cap grup participant s'indica que és obligatori escollir-ne com a mínim un.</li> <li>-Si l'usuari no participa en el calendari creat s'indica que no pot crear un calendari on no hi participi.</li> <li>-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.</li> </ul>
<b>Precondicions</b>	
<b>Postcondicions</b>	Es dona d'alta al sistema un calendari amb les dades introduïdes i es notifiquen tots els participants.
<b>Sortides</b>	

*Taula 60: Operació crearCalendari*

<b>Operació</b>	modificarCalendari()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol modificar la informació d'un calendari.
<b>Cas d'ús</b>	Modificar calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de modificació d'un calendari.
<b>Sortides</b>	

*Taula 61: Operació modificarCalendari*

<b>Operació</b>	modificarDades(nom, descripcio)
<b>Paràmetres</b>	Es passa com a paràmetre la nova informació del calendari: nom i descripció.
<b>Semàntica</b>	Comunica al sistema les noves dades del calendari.
<b>Cas d'ús</b>	Modificar calendari.
<b>Excepcions</b>	<ul style="list-style-type: none"> <li>-Si el nom és buit s'indica que és obligatori.</li> <li>-Si el nom conté caràcters invàlids s'indica l'error.</li> <li>-Si ja existeix un calendari amb aquest nom s'indica que introdueixi un altre nom.</li> <li>-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.</li> <li>-Si un altre usuari està modificant el calendari s'indica a l'usuari que ho torni a intentar.</li> <li>-Si la informació del calendari té un pending approval d'un altre usuari o té un pending approval de l'usuari cridant l'operació que ha estat respost per algun usuari s'indica a l'usuari que ha d'esperar a que el pending approval sigui resultat.</li> </ul>
<b>Precondicions</b>	
<b>Postcondicions</b>	Es modifiquen les dades del calendari amb les dades introduïdes o es crea un pending approval per tal que la resta de participants puguin acceptar o rebutjar els canvis. Es notifiquen tots els participants del calendari.
<b>Sortides</b>	

*Taula 62: Operació modificarDades*

<b>Operació</b>	eliminarCalendari(id)
<b>Paràmetres</b>	Identificador del calendari que es vol eliminar.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar un calendari.
<b>Cas d'ús</b>	Eliminar calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del calendari és correcte.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 63: Operació eliminarCalendari*

<b>Operació</b>	confirmarEliminarCalendari()
<b>Paràmetres</b>	
<b>Semàntica</b>	Confirma al sistema que es pot continuar amb l'eliminació del calendari.
<b>Cas d'ús</b>	Eliminar calendari.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari. -Si un altre usuari està modificant el calendari s'indica a l'usuari que ho torni a intentar. -Si la informació del calendari té un pending approval d'un altre usuari o té un pending approval de l'usuari cridant l'operació que ha estat respost per algun usuari s'indica a l'usuari que ha d'esperar a que el pending approval sigui resolt.
<b>Precondicions</b>	
<b>Postcondicions</b>	El calendari i les seves tasques s'han eliminat del sistema o s'ha creat un pending approval per tal que la resta de participants puguin acceptar o rebutjar l'eliminació. Es notifiquen tots els participants del calendari.
<b>Sortides</b>	

*Taula 64: Operació confirmarEliminarCalendari*

<b>Operació</b>	mostrarInfoCalendari(id)
<b>Paràmetres</b>	Identificador del calendari que es vol veure.
<b>Semàntica</b>	Comunica al sistema que es vol veure la informació d'un determinat calendari.
<b>Cas d'ús</b>	Veure informació calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del calendari és correcte.
<b>Postcondicions</b>	Es visualitza la informació del calendari especificat.
<b>Sortides</b>	

*Taula 65: Operació mostrarInfoCalendari*

<b>Operació</b>	afegirParticipantCalendari()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol afegir un o més grups participants al calendari
<b>Cas d'ús</b>	Afegir participant al calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari d'addició de participants a calendaris.
<b>Sortides</b>	

*Taula 66: Operació afegirParticipantCalendari*

<b>Operació</b>	enviarParticipantsCalendari(grups)
<b>Paràmetres</b>	Es passa com a paràmetre els identificadors dels grups que s'afegiran com a participants.
<b>Semàntica</b>	Comunica al sistema quins participants s'afegiran al calendari.
<b>Cas d'ús</b>	Afegir participant al calendari.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari. -Si un altre usuari està modificant els participants del calendari s'indica a l'usuari que ho torni a intentar. -Si l'apartat de participants del calendari té un pending approval d'un altre usuari que conté algun dels grups enviats o bé un pending approval de l'usuari cridant l'operació que ha estat respost i que conté algun dels grups enviats s'indica a l'usuari que ha d'esperar a que el pending approval sigui resolt.
<b>Precondicions</b>	Els identificadors dels grups són correctes.
<b>Postcondicions</b>	S'han afegit al conjunt de participants els grups determinats per l'usuari o s'ha creat un pending approval per tal que la resta de participants puguin acceptar/rebutjar l'addició. Es notifiquen tots els participants del calendari i els membres dels grups afegits.
<b>Sortides</b>	

*Taula 67: Operació enviarParticipantsCalendari*

<b>Operació</b>	eliminarParticipantCalendari(id)
<b>Paràmetres</b>	Identificador del grup que es vol eliminar com a participant.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar un participant determinat.
<b>Cas d'ús</b>	Eliminar participant del calendari.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del participant és correcte.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 68: Operació eliminarParticipantCalendari*

<b>Operació</b>	confirmarElimPartCal()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que pot eliminar el participant del calendari.
<b>Cas d'ús</b>	Eliminar participant del calendari.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari. -Si un altre usuari està modificant els participants del calendari s'indica a l'usuari que ho torni a intentar. -Si l'apartat de participants del calendari té un pending approval d'un altre usuari que conté el grup enviat o bé un pending approval de l'usuari cridant l'operació que ha estat respost i que conté el grup enviat s'indica a l'usuari que ha d'esperar a que el pending approval sigui resolt.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'ha eliminat el grup del conjunt de participants del calendari o s'ha creat un pending approval per tal que la resta de participants puguin acceptar o rebutjar l'eliminació. Es notifiquen tots els participants del calendari.
<b>Sortides</b>	

*Taula 69: Operació confirmarElimPartCal*

<b>Operació</b>	l·listarCalendaris()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol veure la l·lista de calendaris en què participa l'usuari.
<b>Cas d'ús</b>	L·listar calendaris
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el l·listat de calendaris en què participa l'usuari.
<b>Sortides</b>	

*Taula 70: Operació l·listarCalendaris*

### 5.3 Gestió de tasques

<b>Operació</b>	altaTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol crear una nova tasca.
<b>Cas d'ús</b>	Crear tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de creació de tasques.
<b>Sortides</b>	

*Taula 71: Operació altaTasca*

<b>Operació</b>	crearTasca(nom, descr, ubicacio, idcal, estat, dataI, dataF, users, grups)
<b>Paràmetres</b>	Es passa com a paràmetre la informació necessària per a crear la tasca: nom, descripció, ubicació, identificador del calendari al qual pertany la tasca, estat, hora i data d'inici, hora i data de finalització, usuaris participants i grups participants.
<b>Semàntica</b>	Comunica la informació necessària per a crear la nova tasca.
<b>Cas d'ús</b>	Crear tasca.
<b>Excepcions</b>	<ul style="list-style-type: none"> <li>-Si el nom és buit s'indica que és obligatori.</li> <li>-Si el nom conté caràcters invàlids s'indica l'error.</li> <li>-Si l'any d'inici o finalització és inferior a 1990 o superior a 2200 s'indica que ha d'estar entre aquests dos valors.</li> <li>-Si la data i hora finals són anteriors a la data i hora inicials s'indica l'error a l'usuari.</li> <li>-Si no s'ha seleccionat cap participant s'indica que hi ha d'haver com a mínim un participant seleccionat.</li> </ul>
<b>Precondicions</b>	
<b>Postcondicions</b>	Es dona d'alta al sistema local una tasca amb les dades introduïdes i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 72: Operació crearTasca*

<b>Operació</b>	modificarTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol modificar la informació d'una tasca.
<b>Cas d'ús</b>	Modificar tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	



<b>Postcondicions</b>	Es visualitza el formulari de modificació de tasques.
<b>Sortides</b>	

*Taula 73: Operació modificarTasca*

<b>Operació</b>	modificarDadesTasca(nom, descr, ubicacio, estat, dataI, dataF)
<b>Paràmetres</b>	Es passa com a paràmetre la nova informació de la tasca: nom, descripció, ubicació, estat, hora i data d'inici i de finalització.
<b>Semàntica</b>	Comunica al sistema les noves dades de la tasca.
<b>Cas d'ús</b>	Modificar tasca.
<b>Excepcions</b>	-Si el nom és buit s'indica que és obligatori. -Si el nom conté caràcters invàlids s'indica l'error. -Si l'any d'inici o finalització és inferior a 1990 o superior a 2200 s'indica que ha d'estar entre aquests dos valors. -Si la data i hora finals són anteriors a la data i hora inicials s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es modifiquen les dades de la tasca amb les dades introduïdes a nivell local i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 74: Operació modificarDadesTasca*

<b>Operació</b>	eliminarTasca(id)
<b>Paràmetres</b>	Identificador de la tasca que es vol eliminar.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar una tasca.
<b>Cas d'ús</b>	Eliminar tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador de la tasca és correcte.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 75: Operació eliminarTasca*

<b>Operació</b>	confirmarElimTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma que es vol donar de baixa la tasca del sistema.
<b>Cas d'ús</b>	Eliminar tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es dona de baixa la tasca a nivell local i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 76: Operació confirmarElimTasca*

<b>Operació</b>	mostrarInfoTasca(id)
<b>Paràmetres</b>	Identificador de la tasca que es vol veure.
<b>Semàntica</b>	Comunica al sistema que es vol veure la informació d'una tasca.
<b>Cas d'ús</b>	Veure informació tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador de la tasca és correcte.
<b>Postcondicions</b>	Es visualitza la informació de la tasca especificada.
<b>Sortides</b>	

*Taula 77: Operació mostrarInfoTasca*

<b>Operació</b>	afegirParticipantTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol afegir un o més usuaris o grups participants a la tasca.
<b>Cas d'ús</b>	Afegir participant a tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari d'addició de participants a tasques.
<b>Sortides</b>	

*Taula 78: Operació afegirParticipantTasca*

<b>Operació</b>	enviarParticipantsTasca(usuaris, grups)
<b>Paràmetres</b>	Es passa com a paràmetre els identificadors dels usuaris i grups que s'afegiran com a participants.
<b>Semàntica</b>	Comunica al sistema quins participants s'afegiran a la tasca.
<b>Cas d'ús</b>	Afegir participant a tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	Els identificadors dels usuaris i grups són correctes.
<b>Postcondicions</b>	S'afegeixen al conjunt de participants els grups i/o usuaris determinats per l'usuari i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 79: Operació enviarParticipantsTasca*

<b>Operació</b>	eliminarParticipantTasca(id)
<b>Paràmetres</b>	Identificador de l'usuari o grup que es vol eliminar com a participant.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar un participant determinat.
<b>Cas d'ús</b>	Eliminar participant d'una tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del participant és correcte.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 80: Operació eliminarParticipantTasca*

<b>Operació</b>	confirmarElimPartTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que pot eliminar el participant de la tasca.
<b>Cas d'ús</b>	Eliminar participant d'una tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	S'elimina l'usuari o grup del conjunt de participants de la tasca i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 81: Operació confirmarElimPartTasca*

<b>Operació</b>	afegirPrecedentTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol afegir una o més tasques com a precedents de la tasca.
<b>Cas d'ús</b>	Afegir precedent a tasca.

<b>Excepcions</b>	-Si no hi ha cap tasca que pugui ser precedent de la tasca tractada es notifica a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari d'addició de tasques precedents.
<b>Sortides</b>	

*Taula 82: Operació afegirPrecedentTasca*

<b>Operació</b>	enviarPrecedentsTasca(tasques)
<b>Paràmetres</b>	Es passa com a paràmetre els identificadors de les tasques que s'afegiran com a precedents.
<b>Semàntica</b>	Comunica al sistema quines tasques s'afegiran com a precedents.
<b>Cas d'ús</b>	Afegir precedent a tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	Els identificadors de les tasques precedents són correctes.
<b>Postcondicions</b>	S'afegeixen al conjunt de precedents les tasques determinades per l'usuari i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 83: Operació enviarPrecedentsTasca*

<b>Operació</b>	eliminarPrecedentTasca(id)
<b>Paràmetres</b>	Identificador de la tasca que es vol eliminar com a precedent.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar una tasca precedent determinada.
<b>Cas d'ús</b>	Eliminar precedent d'una tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador de la tasca precedent és correcte.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 84: Operació eliminarPrecedentTasca*

<b>Operació</b>	confirmarElimPartTasca()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que pot eliminar la tasca com a precedent.
<b>Cas d'ús</b>	Eliminar precedent d'una tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	S'elimina la tasca del conjunt de precedents de la tasca i es crida el cas d'ús <i>Enviar tasques</i> .
<b>Sortides</b>	

*Taula 85: Operació confirmarElimPartTasca*

<b>Operació</b>	l·listarTasques(nomesMeves)
<b>Paràmetres</b>	Paràmetre indicant si s'han de mostrar només les tasques en què l'usuari participa o bé totes les tasques dels calendaris de l'usuari.
<b>Semàntica</b>	Comunica al sistema que es vol veure la llista de tasques en què l'usuari participa o bé la llista de tasques dels calendaris de l'usuari.
<b>Cas d'ús</b>	L·listar tasques

<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el llistat de tasques demanat.
<b>Sortides</b>	

*Taula 86: Operació llistarTasques*

<b>Operació</b>	filtrarLlistat()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol filtrar la llista de tasques.
<b>Cas d'ús</b>	Filtrar llistat de tasques
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de filtració.
<b>Sortides</b>	

*Taula 87: Operació filtrarLlistat*

<b>Operació</b>	filtrar(estat, calendari, data)
<b>Paràmetres</b>	Es passen com a paràmetre els criteris que es seguiran per filtrar el llistat: estat, calendari i data. No és obligatori filtrar utilitzant tots tres criteris.
<b>Semàntica</b>	Comunica al sistema els criteris de filtratge.
<b>Cas d'ús</b>	Filtrar llistat de tasques
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el llistat filtrat de tasques dels calendaris en els que participa l'usuari.
<b>Sortides</b>	

*Taula 88: Operació filtrar*

<b>Operació</b>	enviarNotificacio(notificacio)
<b>Paràmetres</b>	Notificació que es vol enviar.
<b>Semàntica</b>	El sistema local (aplicació) comunica al sistema operatiu que ha d'enviar una notificació a l'usuari.
<b>Cas d'ús</b>	Recordar tasca.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostra la notificació a la barra de notificacions.
<b>Sortides</b>	

*Taula 89: Operació enviarNotificacio*

## 5.4 Sincronització

<b>Operació</b>	demanarCanvis(token, data)
<b>Paràmetres</b>	Es passen com a paràmetre el token per identificar l'usuari al servidor i la data de l'última sincronització (data a partir de la qual s'agafarà la informació). Si la data és null es realitzarà una sincronització completa (agafar tota la informació).
<b>Semàntica</b>	Demana al sistema remot els canvis realitzats a partir d'una certa data.
<b>Cas d'ús</b>	Sincronització total i Demanar canvis.
<b>Excepcions</b>	

<b>Precondicions</b>	
<b>Postcondicions</b>	El sistema remot retorna la informació demanada al sistema local.
<b>Sortides</b>	Llistat de informació que cal emmagatzemar a nivell local (tasques, calendaris, grups, etc.).

*Taula 90: Operació demanarCanvis*

<b>Operació</b>	enviarCanvis(token, tasques, relacions)
<b>Paràmetres</b>	Es passen com a paràmetre tots els canvis realitzats per l'usuari a nivell local (tasques i relacions de les tasques). Per cada canvi, identificador de l'element i els seus camps.
<b>Semàntica</b>	Envia al sistema remot els canvis que s'han de sincronitzar i aquest els desa i, si és necessari, crea esdeveniments, <i>meetings</i> i/o <i>pending approvals</i> .
<b>Cas d'ús</b>	Sincronització total i Enviar tasques.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	El sistema remot enregistra els canvis i envia al sistema local la confirmació dels canvis fets i els elements creats.
<b>Sortides</b>	Llistat amb les tasques i relacions que s'han modificat al servidor i amb els <i>pending approvals</i> i <i>meetings</i> creats.

*Taula 91: Operació enviarCanvis*

<b>Operació</b>	sincronizacioCompleta()
<b>Paràmetres</b>	
<b>Semàntica</b>	S'indica al sistema local que realitzi una sincronització completa.
<b>Cas d'ús</b>	Sincronització total.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	El sistema remot desa els canvis fets a nivell local i el sistema local desa els canvis realitzats per altres usuaris.
<b>Sortides</b>	

*Taula 92: Operació sincronizacioCompleta*

## 5.5 Gestió d'esdeveniments, votacions i preferències

<b>Operació</b>	listarEsdeveniments()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol veure la llista d'esdeveniments de l'usuari.
<b>Cas d'ús</b>	Llistar esdeveniments
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostra el llistat d'esdeveniments de l'usuari.
<b>Sortides</b>	

*Taula 93: Operació llistarEsdeveniments*

<b>Operació</b>	listarMeetings()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol veure la llista de meetings en els que participa l'usuari.

<b>Cas d'ús</b>	Llistar meetings
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostra el llistat de meetings en què participa l'usuari.
<b>Sortides</b>	

*Taula 94: Operació llistarMeetings*

<b>Operació</b>	veureInfoMeeting(id)
<b>Paràmetres</b>	Identificador del meeting que es vol veure.
<b>Semàntica</b>	Comunica al sistema que es vol veure la informació d'un meeting determinat.
<b>Cas d'ús</b>	Veure informació meeting.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del meeting és correcte.
<b>Postcondicions</b>	Es visualitza la informació del meeting especificat.
<b>Sortides</b>	

*Taula 95: Operació veureInfoMeeting*

<b>Operació</b>	votar(vot)
<b>Paràmetres</b>	Es passa com a paràmetre les opcions preferides per l'usuari (una opció per cada camp afectat).
<b>Semàntica</b>	Comunica al sistema l'opció que vol votar l'usuari.
<b>Cas d'ús</b>	Votar en un meeting.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 96: Operació votar*

<b>Operació</b>	confirmarVot()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema el vot de l'usuari.
<b>Cas d'ús</b>	Votar en un meeting.
<b>Excepcions</b>	-Si algun camp afectat no té opció escollida s'indica a l'usuari que és obligatori votar una opció per cada camp afectat. -Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari. -Si el meeting està inactiu es notifica a l'usuari que ja no pot votar. -Si l'usuari ja ha votat en aquest meeting se li notifica.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es desa el vot de l'usuari i, si tots els participants han votat o un dels resultats ha obtingut suficients vots per a ser determinada guanyadora, es marca el meeting com a inactiu, es realitzen els canvis determinats pel resultat i es notifica als usuaris participants. Si hi ha un empat, es repeteix el meeting. Si el meeting finalitza, es notifiquen tots els seus participants.
<b>Sortides</b>	

*Taula 97: Operació confirmarVot*

<b>Operació</b>	eliminarVot(idMeeting)
<b>Paràmetres</b>	Es passa com a paràmetre l'identificador de meeting del qual es vol eliminar el vot.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar un vot.
<b>Cas d'ús</b>	Eliminar vot d'un meeting.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 98: Operació eliminarVot*

<b>Operació</b>	confirmarEliminarVot()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que es vol eliminar el vot.
<b>Cas d'ús</b>	Eliminar vot d'un meeting.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari. -Si el meeting està inactiu es notifica a l'usuari que ja no pot eliminar el vot. -Si l'usuari no ha votat en aquest meeting se li notifica.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'elimina el vot de l'usuari.
<b>Sortides</b>	

*Taula 99: Operació confirmarEliminarVot*

<b>Operació</b>	l·listarPendingApprovals()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol veure la llista de pending approvals en els que participa l'usuari.
<b>Cas d'ús</b>	L·listar pending approvals.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostra el llistat de pending approvals en què participa l'usuari.
<b>Sortides</b>	

*Taula 100: Operació l·listarPendingApprovals*

<b>Operació</b>	veureInfoPendingApproval(id)
<b>Paràmetres</b>	Identificador del pending approval que es vol veure.
<b>Semàntica</b>	Comunica al sistema que es vol veure la informació d'un pending approval determinat.
<b>Cas d'ús</b>	Veure informació pending approval.
<b>Excepcions</b>	
<b>Precondicions</b>	L'identificador del pending approval és correcte.
<b>Postcondicions</b>	Es visualitza la informació del pending approval especificat.
<b>Sortides</b>	

*Taula 101: Operació veureInfoPendingApproval*

<b>Operació</b>	respondrePendingApproval(acceptar?)
<b>Paràmetres</b>	El paràmetre indica si s'accepta o es rebutja el pending approval.
<b>Semàntica</b>	Comunica al sistema que es vol rebutjar o acceptar un pending

	approval.
<b>Cas d'ús</b>	Acceptar/Rebutjar pending approval.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 102: Operació respondrePendingApproval*

<b>Operació</b>	confirmarResposta()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que es vol respondre el pending approval.
<b>Cas d'ús</b>	Acceptar/Rebutjar pending approval.
<b>Excepcions</b>	-Si el pending approval ja no existeix es notifica a l'usuari. -Si l'usuari ja ha respost el pending approval se li notifica.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es desfa la decisió presa per l'usuari i, si tots els participants han votat o un dels resultats ha obtingut una quantitat de vots superior al 50%, s'elimina el pending approval, es realitzen els canvis determinats pel resultat i es notifica a tots els participants. Si hi ha un empat, es rebutja el pending approval. Si el pending approval finalitza, es notifiquen tots els seus participants.
<b>Sortides</b>	

*Taula 103: Operació confirmarResposta*

<b>Operació</b>	veurePreferencies()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es volen veure les preferències.
<b>Cas d'ús</b>	Veure preferències.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostren les preferències.
<b>Sortides</b>	

*Taula 104: VeurePreferencies*

<b>Operació</b>	modificarPreferencies()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es volen modificar les preferències de l'usuari.
<b>Cas d'ús</b>	Modificar preferències.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de modificació de preferències.
<b>Sortides</b>	

*Taula 105: Operació modificarPreferencies*

<b>Operació</b>	modificarDadesPref(vibrR, vibrE, vibrM, vibrPA, tempsR)
<b>Paràmetres</b>	Es passa com a paràmetre la informació de les noves preferències definides: si el mòbil ha de vibrar al rebre notificacions de recordatoris, esdeveniments, meetings o pending approvals, i el temps d'antelació amb el qual s'han de



	recordar les tasques.
<b>Semàntica</b>	Comunica al sistema les noves preferències.
<b>Cas d'ús</b>	Modificar preferències.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es modifiquen les preferències amb les dades introduïdes.
<b>Sortides</b>	

*Taula 106: Operació modificarDadesPref*

## 5.6 Gestió de grups

<b>Operació</b>	altaGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol crear un nou grup.
<b>Cas d'ús</b>	Crear grup.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el formulari de creació de grups.
<b>Sortides</b>	

*Taula 107: Operació altaGrup*

<b>Operació</b>	crearGrup(nom, usuaris)
<b>Paràmetres</b>	Es passa com a paràmetre la informació necessària per a crear el grup: nom i identificadors dels membres.
<b>Semàntica</b>	Comunica la informació necessària per a crear el nou grup.
<b>Cas d'ús</b>	Crear grup.
<b>Excepcions</b>	-Si el nom és buit s'indica que és obligatori. -Si el nom conté caràcters invàlids s'indica l'error. -Si ja existeix un grup amb aquest nom s'indica que introdueixi un altre nom. -Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es dona d'alta al sistema un grup amb les dades introduïdes i amb l'usuari com a administrador i es notifiquen tots els membres del grup.
<b>Sortides</b>	

*Taula 108: Operació crearGrup*

<b>Operació</b>	modificarNomGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol modificar el nom d'un grup.
<b>Cas d'ús</b>	Modificar nom grup.
<b>Excepcions</b>	
<b>Precondicions</b>	L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	Es visualitza el formulari de modificació de nom.
<b>Sortides</b>	

*Taula 109: Operació modificarNomGrup*

<b>Operació</b>	canviarNom(nom)
<b>Paràmetres</b>	Es passa com a paràmetre el nou nom del grup.
<b>Semàntica</b>	Comunica al sistema el nou nom del grup.
<b>Cas d'ús</b>	Modificar nom grup.
<b>Excepcions</b>	-Si el nom és buit s'indica que és obligatori. -Si el nom conté caràcters invàlids s'indica l'error. -Si ja existeix un grup amb aquest nom s'indica que introdueixi un altre nom. -Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es modifica el nom del grup i es notifiquen tots els membres del grup..
<b>Sortides</b>	

*Taula 110: Operació canviarNom*

<b>Operació</b>	veureInfoGrup(id)
<b>Paràmetres</b>	Identificador del grup que es vol veure.
<b>Semàntica</b>	Comunica al sistema que es vol veure la informació d'un determinat grup.
<b>Cas d'ús</b>	Veure informació grup.
<b>Excepcions</b>	-Si el grup no està desat a nivell local i hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	L'identificador del grup és correcte.
<b>Postcondicions</b>	Es visualitza la informació del grup especificat.
<b>Sortides</b>	

*Taula 111: Operació veureInfoGrup*

<b>Operació</b>	afegirParticipantGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol afegir un o més membres al grup.
<b>Cas d'ús</b>	Afegir participant a un grup.
<b>Excepcions</b>	
<b>Precondicions</b>	L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	Es visualitza el formulari d'addició de participants a grups.
<b>Sortides</b>	

*Taula 112: Operació afegirParticipantGrup*

<b>Operació</b>	enviarParticipantsGrup(usuaris)
<b>Paràmetres</b>	Es passa com a paràmetre els identificadors dels usuaris que s'afegiran com a participants.
<b>Semàntica</b>	Comunica al sistema quins participants s'afegiran al grup.
<b>Cas d'ús</b>	Afegir participant a un grup.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'afegeixen al grup els usuaris especificats i es notifiquen tots els membres del grup i tots els usuaris afegits.
<b>Sortides</b>	

*Taula 113: Operació enviarParticipantsGrup*

<b>Operació</b>	eliminarParticipantGrup(id)
<b>Paràmetres</b>	Identificador de l'usuari que es vol eliminar del grup.
<b>Semàntica</b>	Comunica al sistema que es vol eliminar un participant determinat.
<b>Cas d'ús</b>	Eliminar participant grup.
<b>Excepcions</b>	-Si l'usuari escull eliminar-se a ell mateix se li notifica que utilitzi l'opció "Sortir del grup".
<b>Precondicions</b>	L'identificador de l'usuari a eliminar és correcte. L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 114: Operació eliminarParticipantGrup*

<b>Operació</b>	confirmarElimPartGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que pot eliminar el participant del grup.
<b>Cas d'ús</b>	Eliminar participant grup.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'elimina l'usuari del conjunt de participants del grup i es notifiquen tots els membres del grup.
<b>Sortides</b>	

*Taula 115: Operació confirmarElimPartGrup*

<b>Operació</b>	sortirGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol sortir del grup.
<b>Cas d'ús</b>	Sortir d'un grup.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 116: Operació sortirGrup*

<b>Operació</b>	confirmarSortirGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que es vol sortir del grup.
<b>Cas d'ús</b>	Sortir d'un grup.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	S'elimina l'usuari del conjunt de participants del grup. Si és l'últim membre del grup, s'elimina el grup. Si no és l'últim membre del grup però n'és l'administrador, s'escull un nou administrador. Es notifiquen tots els membres del grup.
<b>Sortides</b>	

*Taula 117: Operació confirmarSortirGrup*

<b>Operació</b>	l·listarGrups()
<b>Paràmetres</b>	

<b>Semàntica</b>	Comunica al sistema que es vol veure la llista de grups als que pertany l'usuari.
<b>Cas d'ús</b>	Llistar grups
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza el llistat de grups als que pertany l'usuari.
<b>Sortides</b>	

*Taula 118: Operació llistarGrups*

<b>Operació</b>	nomenarAdministrador(idUser)
<b>Paràmetres</b>	Identificador de l'usuari que es vol nomenar administrador.
<b>Semàntica</b>	Comunica al sistema que es vol nomenar un nou administrador.
<b>Cas d'ús</b>	Nomenar administrador.
<b>Excepcions</b>	-Si l'usuari escull nomenar-se administrador a ell mateix se li indica que ja ho és.
<b>Precondicions</b>	L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	Es visualitza una vista de confirmació.
<b>Sortides</b>	

*Taula 119: Operació nomenarAdministrador*

<b>Operació</b>	confirmarNomenar()
<b>Paràmetres</b>	
<b>Semàntica</b>	Es confirma al sistema que pot canviar l'administrador del grup.
<b>Cas d'ús</b>	Nomenar administrador.
<b>Excepcions</b>	-Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	L'usuari indicat passa a ser l'administrador del grup i l'anterior administrador perd aquest rol. Es notifiquen tots els membres del grup.
<b>Sortides</b>	

*Taula 120: Operació confirmarNomenar*

<b>Operació</b>	cercarGrup()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol cercar un grup.
<b>Cas d'ús</b>	Cercar grup.
<b>Excepcions</b>	
<b>Precondicions</b>	
<b>Postcondicions</b>	Es visualitza la vista de cerca de grups.
<b>Sortides</b>	

*Taula 121: Operació cercarGrup*

<b>Operació</b>	cerca(text)
<b>Paràmetres</b>	Text que es vol cercar.
<b>Semàntica</b>	Es comunica al sistema la cadena de caràcters a cercar en els noms dels grups.
<b>Cas d'ús</b>	Cercar grup.
<b>Excepcions</b>	-Si el text és buit s'indica que no pot ser-ho. -Si el text conté caràcters invàlids s'indica l'error. -Si hi ha un error al connectar amb el servidor s'indica l'error a

	l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es mostra el llistat de grups que contenen al seu nom la cadena de caràcters introduïda per l'usuari.
<b>Sortides</b>	

*Taula 122: Operació cerca*

<b>Operació</b>	enviarOCancelarSolicitud()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol enviar una sol·licitud per entrar a un grup o, en cas que ja hi hagi una sol·licitud per entrar al grup, es vol eliminar la sol·licitud.
<b>Cas d'ús</b>	Enviar/Cancel·lar sol·licitud
<b>Excepcions</b>	-Si l'usuari ja és membre del grup no se li permet enviar sol·licitud. -Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	
<b>Postcondicions</b>	Es desa o s'elimina la sol·licitud i es notifica a l'administrador del grup.
<b>Sortides</b>	

*Taula 123: Operació enviarOCancelarSolicitud*

<b>Operació</b>	l·listarSolicitud()
<b>Paràmetres</b>	
<b>Semàntica</b>	Comunica al sistema que es vol veure la llista de sol·licituds d'entrada a un grup.
<b>Cas d'ús</b>	Llistar sol·licituds.
<b>Excepcions</b>	-Si no hi ha sol·licituds es notifica a l'usuari.
<b>Precondicions</b>	L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	Es mostra el llistat de sol·licituds per entrar al grup.
<b>Sortides</b>	

*Taula 124: Operació l·listarSolicitud*

<b>Operació</b>	respondreSolicitud(acceptar?)
<b>Paràmetres</b>	El paràmetre indica si s'accepta o es rebutja la sol·licitud.
<b>Semàntica</b>	Comunica al sistema que es vol acceptar o rebutjar una sol·licitud.
<b>Cas d'ús</b>	Acceptar/Rebutjar sol·licitud
<b>Excepcions</b>	-Si la sol·licitud no existeix es notifica l'error a l'usuari. -Si hi ha un error al connectar amb el servidor s'indica l'error a l'usuari.
<b>Precondicions</b>	L'usuari que crida l'operació és l'administrador del grup.
<b>Postcondicions</b>	S'elimina la sol·licitud i, en cas que sigui acceptada, s'afegeix l'usuari de la sol·licitud com a membre del grup. Es notifiquen els usuaris corresponents.
<b>Sortides</b>	

*Taula 125: Operació respondreSolicitud*

## 6. Model conceptual

### 6.1 Diagrama de classes

A continuació podem observar el diagrama de classes sense atributs. S'ha dividit en dos diagrames de classes: el que utilitzarà el client (aplicació Android) i el que utilitzarà el servidor (SugarCRM).

#### Client

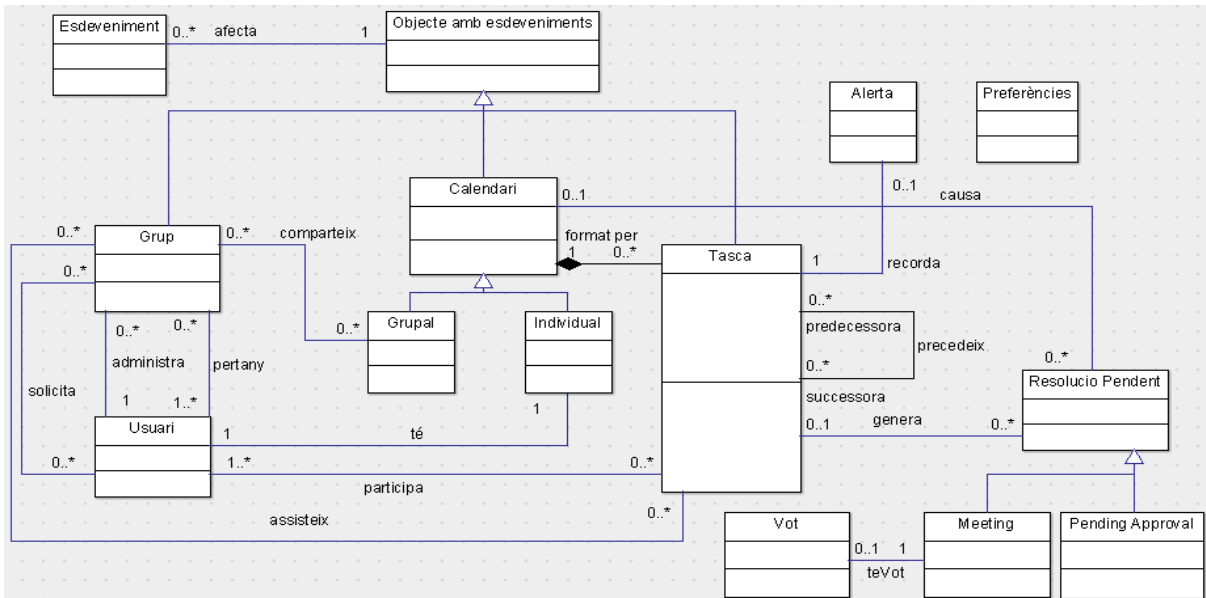


Figura 61: Model conceptual local sense atributs

#### Servidor

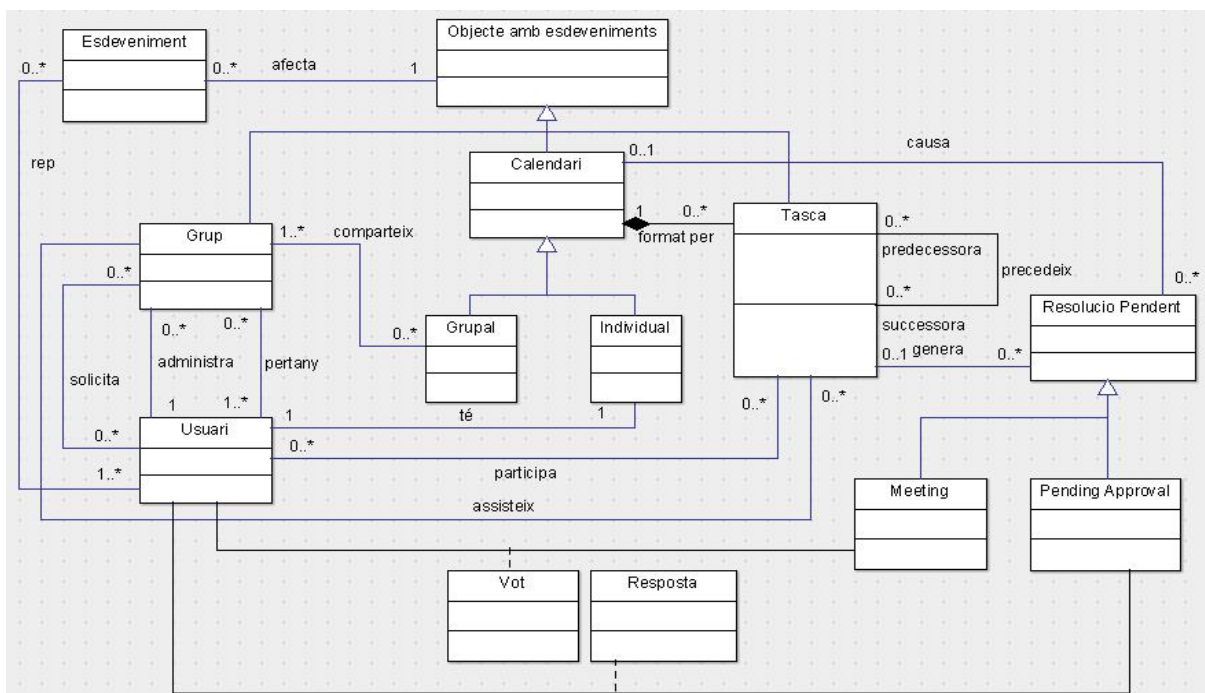


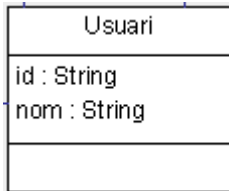
Figura 62: Model conceptual servidor sense atributs

## 6.2 Atributs de les classes

A continuació s'expliquen les classes amb els seus atributs.

### Classe Usuari

Classe que representa els usuaris que utilitzen l'aplicació. Per tal de poder utilitzar totes les funcionalitats que ofereix el sistema, l'usuari ha d'estar loguejat.



*id*: Identificador de l'usuari, assignat pel sistema.

*nom*: Nom que utilitza l'usuari per a identificar-se al sistema.

Figura 63: Classe Usuari

### Classe Grup

Classe que representa els grups d'usuaris, creats per tal de facilitar la feina de compartir tasques i calendaris.

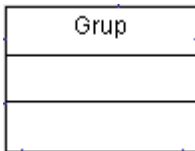
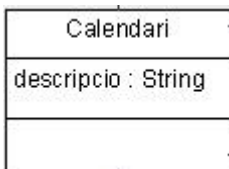


Figura 64: Classe Grup

### Classe Calendari

Classe que representa els calendaris dels usuaris. Cada usuari pot crear tants calendaris com vulgui, i es poden compartir amb altres usuaris. Tots els usuaris que participin en un calendari podran veure totes les tasques del calendari. Els calendaris poden ser de dos tipus: grupals o individuals. Un usuari té únicament un calendari individual que no es pot modificar ni s'hi poden afegir més participants. Els calendaris grupals, en canvi, es poden crear, modificar i eliminar, i s'hi poden afegir més participants.



*descripcio*: Breu descripció del calendari.



Figura 65: Classe Calendari i subclasses

## Classe Tasca

Classe que representa les tasques dels usuaris. Els usuaris que participin en una tasca tindran permisos per modificar la tasca.

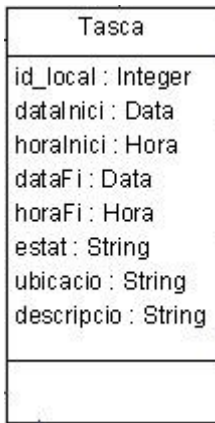


Figura 66: Classe Tasca

*id\_local*: Identificador de la tasca al client, assignat pel sistema. Només al model del client.

*dataInici*: Dia, mes i any en que s'inicia la tasca.

*horaInici*: Hora i minut en que s'inicia la tasca.

*dataFi*: Dia, mes i any en que s'acaba la tasca.

*horaFi*: Hora i minut en que s'acaba la tasca.

*estat*: Estat actual de la tasca: pendent, en curs, ajornada, cancel·lada, ...

*ubicació*: Ubicació de la tasca.

*descripció*: Breu descripció de la tasca.

## Classe Objecte amb esdeveniments

Classe auxiliar amb les següents subclasses: Grup, Calendari i Tasca. Totes les classes que heretin d'aquesta poden tenir esdeveniments.

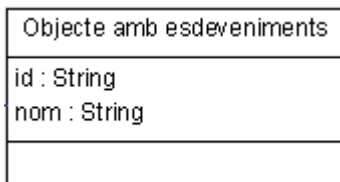


Figura 67: Classe Objecte amb esdeveniments

*id*: Identificador de l'objecte, assignat pel sistema.

*nom*: Nom de l'objecte.

## Classe Esdeveniment

Tipus de notificació que es crearà quan es realitzin certes accions sobre tasques, calendaris o grups. El seu objectiu es informar a l'usuari de les accions que s'han realitzat.

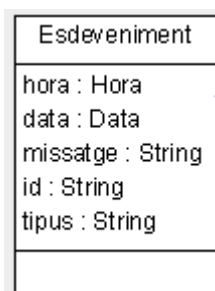


Figura 68: Classe Esdeveniment

*hora*: Hora i minut en què s'ha generat l'esdeveniment.

*data*: Dia, mes i any en què s'ha generat l'esdeveniment.

*missatge*: Missatge de l'esdeveniment.

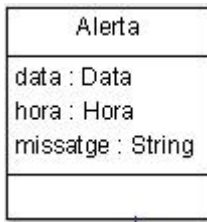
*id*: Identificador de l'esdeveniment, assignat pel sistema.

*tipus*: Tipus d'esdeveniment: Grup, Calendari o Tasca. Indica a quin tipus d'objecte afecta.



## Classe Alerta

Tipus de notificació que té com a objectiu recordar a l'usuari les tasques que s'iniciaran properament.



*data*: Dia, mes i any en què s'enviarà la notificació a l'usuari.

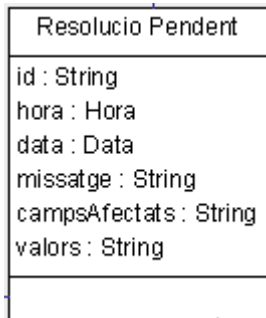
*hora*: Hora i minut en què s'enviarà la notificació a l'usuari.

*missatge*: Missatge de l'alerta.

Figura 69: Classe Alerta

## Classe Resolució Pendent

Quan un usuari vulgui modificar una tasca o calendari amb altres participants es generarà una Resolució Pendent per tal que la resta de participants puguin escollir què fer.



*id*: Identificador de la resolució pendent, assignat pel sistema.

*hora*: Hora i minut en què s'ha creat la resolució pendent.

*data*: Dia, mes i any en què s'ha creat la resolució pendent.

*missatge*: Missatge de la resolució pendent.

*campsAfectats*: Camps de la tasca/calendari afectats per la resolució pendent. Per exemple, *nom*.

Figura 70: Classe Resol. Pend.

*valors*: Valors que poden prendre cadascun dels camps afectats.

## Classe Meeting

Subclasse de Resolució Pendent que representa una "trobadura" online per tal de resoldre un conflicte. Es generarà cada cop que es detecti un conflicte en una tasca, i hi participaran tots els usuaris que participin en la tasca. Cada usuari podrà votar l'opció que prefereix per cadascun dels camps afectats.

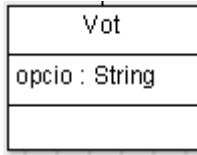


*actiu*: Indica si la votació ha finalitzat.

Figura 71: Classe Meeting

## Classe Vot

Classe que representa el vot d'un usuari en un meeting.



*opcio*: String amb els valors escollits per l'usuari per cadascun dels camps afectats del meeting.

Figura 72: Classe Vot

## Classe Pending Approval

Subclasse de Resolució Pendent que es generarà cada cop que un usuari vulgui modificar una tasca o calendari que tingui més participants i no hi hagi conflicte, i hi participaran tots els usuaris que participin en la tasca.

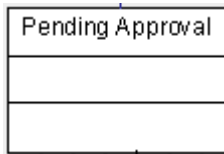
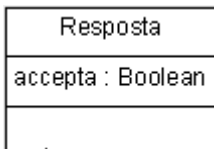


Figura 73: Classe Pending Approval

## Classe Reply

Classe que representa la resposta d'un usuari a un pending approval.

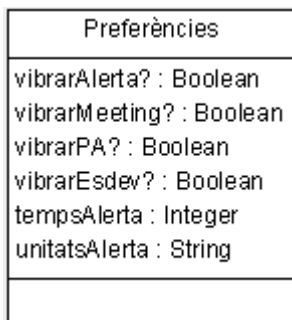


*accepta*: Indica si l'usuari accepta els canvis del pending approval o no.

Figura 74: Classe Resposta

## Classe Preferències

Classe que representa les preferències escollides per l'usuari.



*vibrarAlerta?*: Indica si el telèfon ha de vibrar al rebre una notificació d'alerta.

*vibrarMeeting?*: Indica si el telèfon ha de vibrar al rebre una notificació de meeting.

*vibrarPA?*: Indica si el telèfon ha de vibrar al rebre una notificació de pending approval.

*vibrarEsdev?*: Indica si el telèfon ha de vibrar al rebre una notificació d'esdeveniment.

*tempsAlerta*: Indica la quantitat de temps d'antelació amb què s'han d'avisar les alertes.

*unitatsAlerta*: Indica les unitats del temps d'antelació amb què s'han d'avisar les alertes. "minuts" o "hores".

Figura 75: Classe Preferències

## 6.3 Diagrama de classes complet

### Client

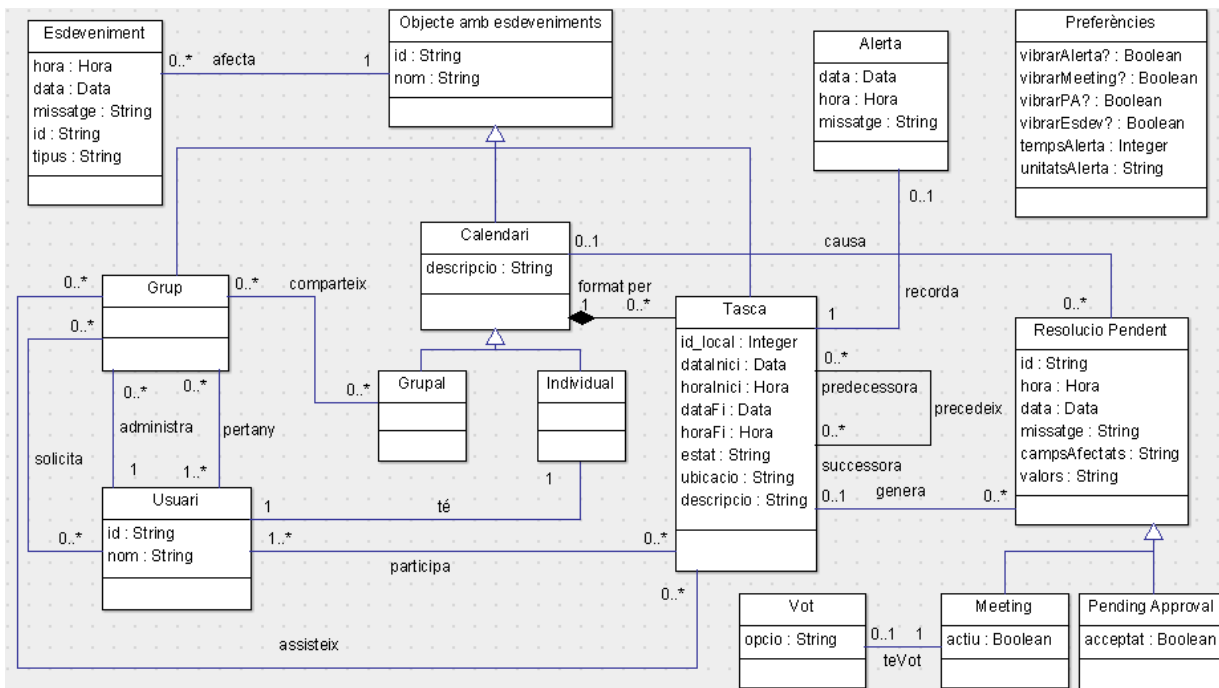


Figura 76: Model conceptual local amb atributs

### Servidor

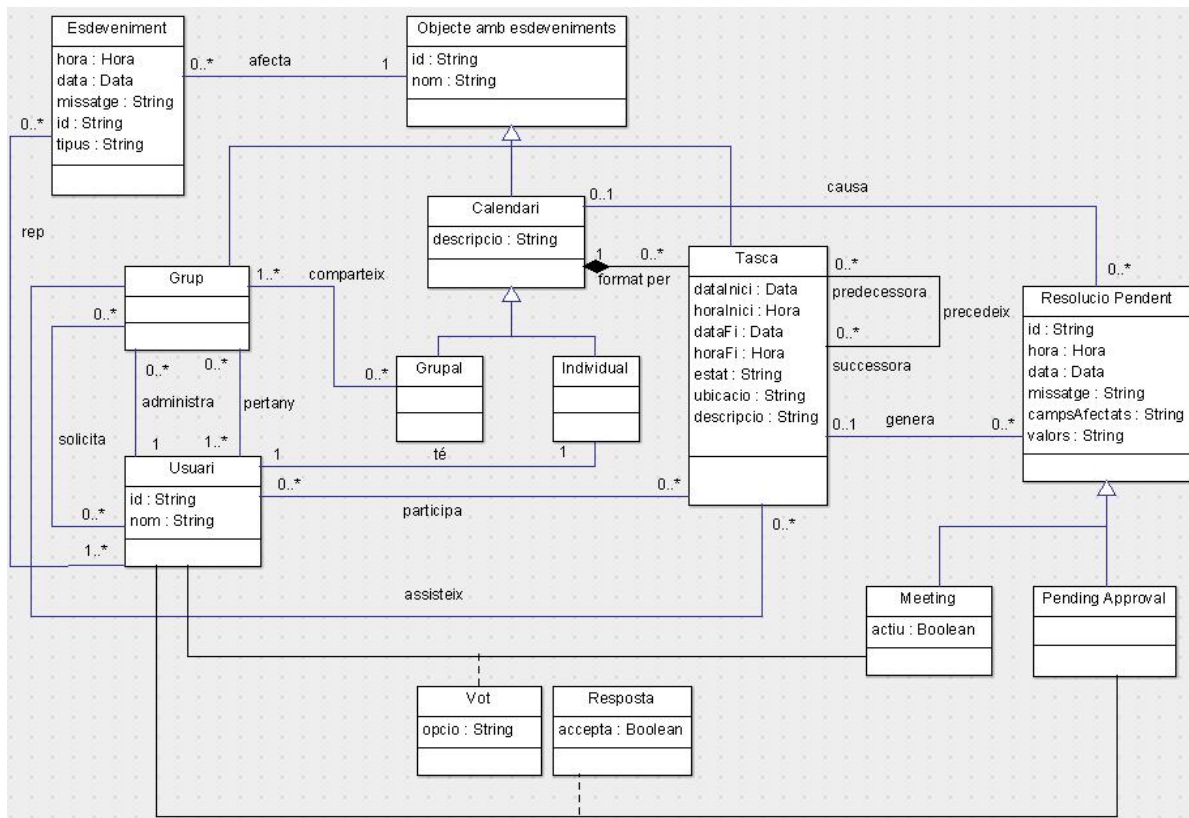


Figura 77: Model conceptual servidor amb atributs

## **Restriccions d'integritat textuais**

- 1-** Clau primària: (Objecte amb esdeveniments, id), (Resolucio Pendent, id), (Esdeveniment, id), (Vot, Usuari.id + Meeting.id), (Resposta, Usuari.id + PendingApproval.id).
- 2-** L'hora i data finals d'una tasca no poden ser anteriors a l'hora i data inicials d'aquesta mateixa tasca.
- 3-** L'hora i data inicials d'una tasca no poden ser anteriors a l'hora i data finals d'una tasca precedent.
- 4-** L'usuari que administra el grup està inclòs a l'associació *pertany*.
- 5-** Dos grups no poden tenir el mateix nom.
- 6-** Dos calendari no poden tenir el mateix nom.
- 7-** Dos usuaris no poden tenir el mateix nom.
- 8-** Els usuaris i grups que participen en una tasca han de participar en el calendari al qual pertany la tasca.
- 9-** El tipus de l'esdeveniment ha de ser igual al tipus de l'objecte amb esdeveniments al qual està associat.
- 10-** Una resolució pendent afecta a una tasca o a un calendari, mai a tots dos a la vegada ni a cap dels dos.
- 11-** En un telèfon no poden haver-hi dues tasques amb el mateix identificador local.
- 12-** No hi pot haver dues instàncies de la classe Preferències en un mateix dispositiu.

# Disseny

A l'etapa de disseny es presenta l'arquitectura del sistema, adaptant l'especificació de l'apartat anterior a la tecnologia específica que s'utilitzarà. D'aquesta manera, la documentació queda enllestida per a passar a l'etapa d'implementació.

## 1. Patró arquitectònic

El sistema Android està desenvolupat per tal que s'utilitzi per defecte una lleugera variació del patró MVVM (Model View ViewModel) per a desenvolupar les aplicacions, un patró arquitectònic creat per Microsoft com una especialització del patró Presentation Model. Tot i que això no implica que sigui obligatori utilitzar aquest patró (hi ha maneres de passar a MVC o a MVP, però no són intuïtives), per simplicitat i pels avantatges que ofereix aquest patró s'ha decidit utilitzar MVVM per a desenvolupar el projecte.

El patró MVVM proporciona una clara separació de la interfície gràfica i de la lògica del model de l'aplicació. El ViewModel serveix com a enllaç entre aquests dos elements, és el responsable de mostrar els objectes del model de manera que siguin fàcilment manipulables. Aquest patró aconsegueix tan els avantatges de separació funcional que proporciona el patró MVC (Model Vista Controlador) com la simplificació que proporciona utilitzar *data binding* per transferir dades des de i cap a la vista, que permet que el controlador només sigui responsable d'implementar el comportament de la vista.

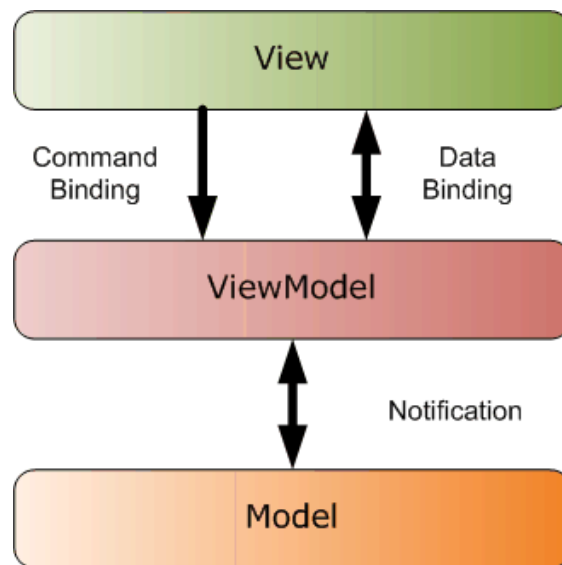


Figura 78: Esquema patró MVVM

Els elements del patró MVVM són:

-Model: Com en el patró MVC, representa la informació específica amb la qual el sistema opera.

-Vista: Com en el patró MVC, es refereix a tots els elements mostrats per la interfície gràfica.

-View Model (Model Vista): és el responsable d'implementar el comportament de la vista per a respondre a les accions de l'usuari i d'exposar les dades del model de forma que sigui senzill utilitzar *bindings* a la vista. Es podria veure com una especialització del que seria un controlador del patró MVC que converteix informació del model en informació de la vista i transmet accions de la vista al model.

En la implementació estàndard en Android no s'utilitza *binding*. Per tal d'incorporar-lo al nostre sistema, utilitzarem un framework de codi lliure anomenat *Android Binding*.

El flux més habitual que segueix el patró MVVM pel que fa a la interacció entre capes és el següent:

1. L'usuari interactua amb la interfície gràfica (vista).
2. L'acció de l'usuari invoca un Command al ViewModel.
3. El ViewModel accedeix al model per actualitzar o consultar les dades en funció de l'acció realitzada.
4. El model retorna la informació pertinent al ViewModel.
5. El ViewModel envia la informació necessària a la vista actual o a una nova vista.
6. Es genera la vista amb la informació corresponent.
7. La interfície gràfica espera noves interaccions.

Tot programador Android sap que la programació en aquest sistema es basa principalment en les *Activity*, que es podrien interpretar com les "pantalles" d'una aplicació. Amb el patró MVVM, la funcionalitat de les *Activity* s'ha reduït notablement. Inicialment la pròpia *Activity* realitzava la funció de ViewModel, però a les noves versions d'*Android Binding* es va decidir descartar aquesta opció i convertir l'*Activity* en una classe l'única funció de la qual és enllaçar la vista amb el ViewModel (aplicar el *binding*), traslladant el ViewModel a una classe a part.

A continuació s'expliquen amb més detall cadascun dels elements del patró.

## 1.1 Vista

Com ja s'ha explicat anteriorment, la vista és la interfície gràfica amb la que opera l'usuari. En el cas d'Android, les vistes es defineixen en format XML i s'anomenen *layouts*. Normalment un *layout* equival a una vista, però hi ha alguns casos especials en què una vista requereix més d'un *layout*. És el cas, per exemple, de les vistes que inclouen llistes (ListView), que solen requerir un *layout* per la vista en sí i un altre per definir els elements de la llista.

En la programació Android sense utilitzar *binding* el contingut present als *layouts* és estàtic, són les *Activity* les que s'encarreguen de modificar-ne el contingut i els atributs. Utilitzant *Android Binding* es pot indicar al *layout* d'on ha d'agafar la informació, de manera que el ViewModel no ha d'especificar el contingut de cada element de la vista, només cal que defineixi les variables que s'enllaçaran amb els diferents elements de la vista.

A l'hora de crear les vistes de l'aplicació s'ha intentat sempre obtenir vistes intuïtives i agradables a la vista dins de les limitacions del sistema Android. S'ha optat per una estètica senzilla però funcional.

## 1.2 ViewModel

El ViewModel és l'element que s'ha explicat més en l'explicació del patró arquitectònic ja que es tracta de l'element particular del patró MVVM que no és present a altres patrons.

El ViewModel conté, per una banda, totes les dades de la vista subjectes al *binding* i, per l'altra, els *Commands* que invocarà la vista. Com que tota aquesta part és una mica abstracta si no s'hi ha treballat mai, a continuació es presenta un senzill exemple per tal de facilitar la comprensió del patró.

L'exemple es basarà en programar un Hello World utilitzant MVVM amb *Android Binding*. La vista inicial mostrarà un missatge de benvinguda i un botó. Quan es premi el botó, el missatge de benvinguda canviarà. Degut a la simplicitat de l'exemple, aquest no comptarà amb la capa de model.

El contingut del ViewModel és molt senzill. Per una banda, cal declarar un *StringObservable* que es correspondrà amb el missatge de benvinguda mostrat a la vista. Un *StringObservable* no és res més que un *String* que és observable per altres objectes de manera que, si s'hi realitza algun canvi, es notificarà als observadors. En el nostre cas, aquesta variable s'enllaçarà amb l'element de la vista que mostra el missatge de benvinguda, de manera que cada cop que el contingut de la variable es modifiqui també ho farà el text mostrat a la vista. Per altra banda, el ViewModel contindrà un *Command* que s'encarregarà de canviar el missatge i que serà cridat quan es premi el botó. D'aquesta manera, el codi del ViewModel és el següent:

```
public class HelloViewModel {
    public StringObservable Greeting = new StringObservable("Hello from
    Android Binding");

    public Command ChangeGreeting = new Command(){
        public void Invoke(View arg0, Object... arg1){
            Greeting.set("Greeting changed by command");
        }
    };
}
```

Com ja s'ha comentat anteriorment, les *Activity* seran les encarregades d'enllaçar les vistes amb els ViewModel corresponents. Per tal de disposar de les funcions necessàries, les nostres *Activity* hauran d'heretar de la classe *BindingActivity*, obtenint així les funcions *setAndBindRootView*, que s'encarregarà de crear l'enllaç automàticament, i *setAndBindOptionsMenu*, necessària per si es vol afegir un Menú a la vista.

## 1.3 Model

El model conté la informació del sistema amb la qual treballaran els ViewModels. Una bona part de la informació amb la que treballarà l'aplicació es desarà a la base de dades local del telèfon. Degut a la limitació de recursos del sistema Android, els *smartphones* amb aquest sistema operatiu utilitzen SQLite, un SGBD compatible



amb ACID contingut en una llibreria relativament petita. El motor de SQLite no és un procés independent amb el que el programa principal es comunica, sinó que s'enllaça amb el programa passant a formar-ne part. Utilitza SQL com a llenguatge per treballar amb la BD.

Per altra banda, Android ens ofereix un altre sistema per guardar informació permanent d'un tipus molt específic: les preferències. Com s'ha vist als casos d'ús, la nostra aplicació comptarà amb un apartat de preferències on l'usuari escollirà les opcions amb les que prefereixi treballar. Ens podem estalviar haver de crear una taula a la base de dades per guardar-ho, ja que Android ofereix la classe *SharedPreferences* que s'encarrega de desar les preferències de forma permanent, fent aquesta tasca molt més còmode. No només l'utilitzarem per desar les preferències, també s'utilitzarà per emmagatzemar informació que es pugui necessitar en més d'una *Activity*.

Com ja s'ha vist a l'apartat d'especificació, el nostre sistema disposarà de dos models: el model del servidor, que serà únic i compartit per tots els usuaris, i el model local, que contindrà informació relacionada únicament amb l'usuari que utilitzi l'aplicació i que s'emmagatzemarà a la base de dades local del telèfon. La informació desada a nivell local serà una rèplica de la emmagatzemada al servidor (sempre i quan el dispositiu estigui sincronitzat), i ens permetrà poder consultar i modificar certa informació sense haver de dependre de la connexió a Internet.

### 1.3.1 Model local

A l'hora de traslladar el model conceptual a taules de la base de dades, un dels punts importants a decidir és com mapejar les jerarquies a la base de dades, és a dir, com representar les subclasses i superclasses. Com es pot observar al model conceptual, a nivell de client tenim tres jerarquies.

La classe *Objecte amb esdeveniments* no la traslladarem a la base de dades, utilitzarem una taula per cada subclasse i els atributs *id* i *nom* estaran repetits en cadascuna de les taules. El principal motiu de fer-ho així és que la superclasse només va ser creada per tal de simplificar el model conceptual, no té un significat important. Per tal de representar l'associació *afecta* afegirem un camp *id\_affected* a la taula de la classe *Esdeveniment* i un camp *type* per saber a quina classe es refereix aquest identificador. Com que les tres subclasse gairebé no comparteixen atributs i no hi ha solapament entre elles (un objecte no pot ser a la vegada grup i tasca), no hi ha cap inconvenient important que ens impedeixi utilitzar aquesta opció.

Per la classe *Calendari* i les seves subclasses, en canvi, utilitzarem una única taula per tota la jerarquia. S'ha pres aquesta decisió per dos motius: per una banda, les subclasses no contenen informació específica, és a dir, només caldrà afegir un camp *type* a la taula del calendari per identificar el tipus del calendari, de manera que no hi ha espai desaprofitat; per altra banda, no hi ha solapament entre subclasses, de manera que amb un únic atribut és suficient per identificar de quina subclasse es tracta.

Per últim, la classe *Resolució Pendent* tampoc es traslladarà a la base de dades. Com en el cas d'*Objecte amb esdeveniments*, no hi ha solapament entre subclasses. No obstant, en aquest cas hi ha més informació replicada, però no és

un volum massa gran (5 atributs de la base de dades), de manera que això no suposa massa inconvenient. Acceptar aquest inconvenient ens permet millorar l'eficiència a l'hora de consultar *meetings* o *pending approvals*, ja que l'usuari sempre els consultarà per separat i amb aquesta representació senzillament haurem d'agafar totes les files de la taula corresponent, no ens haurem de centrar en filtrar per tipus.

Partint del model conceptual **local** de l'especificació i de les decisions explicades anteriorment obtenim les taules següents:

#### Taula User

Columna	Tipus	Descripció
id	TEXT	Identificador de l'usuari.
name	TEXT	Nom de l'usuari.

*Taula 126: Taula User local*

#### Taula Group

Columna	Tipus	Descripció
id	TEXT	Identificador del grup.
name	TEXT	Nom del grup.
id_admin	TEXT	Identificador de l'usuari administrador.

*Taula 127: Taula Group local*

#### Taula Calendar

Columna	Tipus	Descripció
id	TEXT	Identificador del calendari.
name	TEXT	Nom del calendari.
description	TEXT	Descripció del calendari.
type	TEXT	Tipus del calendari: individual o grupal.

*Taula 128: Taula Calendar local*

#### Taula Task

Columna	Tipus	Descripció
_id	INTEGER	Identificador local de la tasca. Una mateixa tasca pot tenir un identificador local diferent per cada dispositiu on estigui desada.
id_server	TEXT	Identificador de la tasca al servidor. És el mateix a tots els dispositius.

name	TEXT	Nom de la tasca.
description	TEXT	Descripció de la tasca.
place	TEXT	Ubicació de la tasca.
start_date	INTEGER	Hora i data d'inici de la tasca representat en hora Unix.
end_date	INTEGER	Hora i data de finalització de la tasca representat en hora Unix.
status	TEXT	Estat de la tasca.
id_calendar	TEXT	Identificador del calendari al qual pertany la tasca.
dirty_bit	INTEGER	Bit que indica si la tasca ha estat modificada (1) o no (0).
deleted	INTEGER	Bit que indica si la tasca està eliminada (1) o no (0).

**Taula 129: Taula Task local**

### Taula Event

Columna	Tipus	Descripció
id	TEXT	Identificador de l'esdeveniment.
message	TEXT	Missatge de l'esdeveniment.
date	INTEGER	Hora i data de creació de l'esdeveniment en hora Unix.
type	TEXT	Tipus de l'esdeveniment. Serveix per saber si afecta a una tasca, a un calendari o a un grup.
id_affected	TEXT	Identificador de l'objecte afectat que pot ser una tasca, un calendari o un grup.

**Taula 130: Taula Event local**

### Taula Pending Approval

Columna	Tipus	Descripció
id	TEXT	Identificador del pending approval.
message	TEXT	Missatge del pending approval.
fields_affected	TEXT	Nom de tots els camps afectats del pending approval, separats per un caràcter separador. Per exemple, nom#descripcio#ubicacio.

fields_values	TEXT	Valors dels camps afectats del pending approval, separats per un caràcter separador. Per exemple, Tasca1#descr#Aquí.
ids_fields_values	TEXT	Identificadors dels valors del pending approval, separats per un caràcter separador. Utilitzat només quan el pending approval afecta relacions, ja que en aquest cas els valors són els noms dels objectes afegits o eliminats, i aquest camp contindrà els seus identificadors.
accepted	INTEGER	Indica si l'usuari ha acceptat (1) o rebutjat (0) el pending approval. Si encara no l'ha respost, el valor és -1.
id_task	TEXT	Identificador de la tasca afectada pel pending approval. Si afecta a un calendari, aquest valor és nul.
id_calendar	TEXT	Identificador del calendari afectat pel pending approval. Si afecta a una tasca, aquest valor és nul.
date	TEXT	Hora i data de creació del pending approval en hora Unix.
created_by_me	INTEGER	Bit que indica si el pending approval ha estat creat per l'usuari (1) o no (0).

**Taula 131: Taula Pending Approval local**

### Taula Meeting

Columna	Tipus	Descripció
id	TEXT	Identificador del meeting.
message	TEXT	Missatge del meeting.
fields_affected	TEXT	Nom de tots els camps afectats del meeting, separats per un caràcter separador. Per exemple, nom#descripcio#ubicacio.
fields_values	TEXT	Valors dels camps afectats del meeting. Cada camp afectat tindrà un conjunt de valors. Els conjunts de valors estan separats per un caràcter separador, i dins de cada conjunt els valors estan separats per un altre caràcter separador.  Per exemple, Tasca1 T1#d1 d2#Aquí Allà.

active	INTEGER	Indica si el meeting està actiu (1) o no (0).
id_task	TEXT	Identificador de la tasca afectada pel meeting. Si afecta a un calendari, aquest valor és nul.
id_calendar	TEXT	Identificador del calendari afectat pel meeting. Si afecta a una tasca, aquest valor és nul.
date	TEXT	Hora i data de creació del meeting en hora Unix.

*Taula 132: Taula Meeting local*

### Taula Vote

Columna	Tipus	Descripció
choice	TEXT	Opcions del meeting escollides per l'usuari, separades per un caràcter separador. Per exemple, T1#d1#Aquí.
id_meeting	TEXT	Identificador del meeting al qual pertany el vot.

*Taula 133: Taula Vote local*

### Taula Belongs (associació Pertany)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.
id_user	TEXT	Identificador de l'usuari relacionat.
id_group	TEXT	Identificador del grup relacionat.

*Taula 134: Taula Belongs local*

### Taula Requests (associació Solicita)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.
id_user	TEXT	Identificador de l'usuari relacionat.
id_group	TEXT	Identificador del grup relacionat.

*Taula 135: Taula Requests local*

### Taula Shares (associació Comparteix)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.

id_calendar	TEXT	Identificador del calendari relacionat.
id_group	TEXT	Identificador del grup relacionat.

*Taula 136: Taula Shares local*

### Taula Participates (associació Participa)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.
id_user	TEXT	Identificador de l'usuari relacionat.
id_task	TEXT	Identificador de la tasca relacionada.

*Taula 137: Taula Participates local*

### Taula Assists (associació Assisteix)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.
id_group	TEXT	Identificador del grup relacionat.
id_task	TEXT	Identificador de la tasca relacionada.

*Taula 138: Taula Assists local*

### Taula Precedes (associació Precedeix)

Columna	Tipus	Descripció
_ID	INTEGER	Identificador local de la relació.
id_previous	TEXT	Identificador de la tasca precedent.
id_following	TEXT	Identificador de la tasca següent.

*Taula 139: Taula Precedes local*

#### 1.3.2 Model remot

Com passa amb el model local, tenim tres jerarquies per tractar al model remot: *Objecte amb esdeveniments*, *Calendari* i *Resolució Pendent*. Les dues primeres jerarquies es representaran com a la base de dades local i pels mateixos motius.

En el cas de *Resolució Pendent*, però, s'ha decidit representar la jerarquia amb una única taula: *Pending Resolution*. Això és així perquè el principal avantatge utilitzat per prendre la decisió a nivell local, que és la millora d'eficiència en les consultes degut a que els meetings i els pending approvals sovint es consultaven per separat, en aquest cas no ens serveix perquè l'usuari sempre demanarà les noves dades sense fer distincions segons el tipus. A més, cal tenir en compte que al servidor la informació replicada en cas d'utilitzar una taula per cada subclasse seria major, ja que els mòduls de SugarCRM creats amb l'eina *ModuleBuilder* tenen un conjunt d'atributs que són obligatoris, com per exemple *description* o *modified\_by*. És a dir,

en cas d'utilitzar una taula per cada subclasse estaríem replicant més de 20 atributs sense aprofitar un dels principals avantatges d'utilitzar aquesta opció. Utilitzar una taula per cada classe (incloent la superclasse) augmentaria encara més la informació replicada degut als atributs obligatoris, així que el més lògic és utilitzar una única taula per tota la jerarquia.

SugarCRM disposa de mòduls per calendaris i tasques, entre altres. En aquest punt tenim dos alternatives: aprofitar algun dels mòduls que proporciona SugarCRM i afegir els que siguin necessaris o directament crear tots els mòduls (excepte Users) de zero. Per poder prendre la decisió correcta caldria realitzar un anàlisi de requisits en profunditat amb possibles usuaris de l'aplicació per tal de descobrir si prefereixen tenir juntament les seves tasques de SugarCRM i les de la nostra aplicació o que estiguin separades. Com que no disposem d'un grup de persones com el que necessitem, s'ha pres la decisió de crear els mòduls de zero ja que potser és l'opció que requereix més canvis. En cas que finalment es volgués aprofitar els mòduls de SugarCRM, només caldria canviar el nom dels mòduls a algunes crides de l'aplicació i a l'API de SugarCRM i afegir-hi els atributs que fossin necessaris.

Com ja s'ha comentat al paràgraf anterior, tots els mòduls tenen un conjunt d'atributs obligatoris. A les taules presents en aquest apartat només inclourem els atributs utilitzats per la nostra aplicació. Les associacions es representaran amb mòduls, a l'apartat d'implementació es pot trobar la justificació.

Partint del model conceptual **del servidor** de l'especificació i de les decisions explicades anteriorment obtenim les taules següents:

### Mòdul Users

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de l'usuari.
user_name	VARCHAR(60)	Nom de l'usuari.

*Taula 140: Mòdul Users*

### Mòdul SS\_Group

Columna	Tipus	Descripció
id	CHAR(36)	Identificador del grup.
name	VARCHAR(255)	Nom del grup.
id_admin	VARCHAR(36)	Identificador de l'usuari administrador.
date_modified	DATETIME	Data i hora en què s'ha modificat el grup. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si el grup està eliminat (1) o no (0).

*Taula 141: Mòdul SS\_Group*

## Mòdul SS\_Calendar

Columna	Tipus	Descripció
id	CHAR(36)	Identificador del calendari.
name	VARCHAR(255)	Nom del calendari.
description	TEXT	Descripció del calendari.
type	VARCHAR(10)	Tipus del calendari: individual o grupal.
date_modified	DATETIME	Data i hora en què s'ha modificat el calendari. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si el calendari està eliminat (1) o no (0).
assigned_user_id	CHAR(36)	Identificador de l'usuari al qual està assignat el calendari. Utilitzat en els calendaris individuals.

Taula 142: Mòdul SS\_Calendar

## Mòdul SS\_Task

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la tasca.
name	VARCHAR(255)	Nom de la tasca.
description	TEXT	Descripció de la tasca.
place	VARCHAR(40)	Ubicació de la tasca.
startdate	DATETIME	Hora i data d'inici de la tasca. Format: aaaa-mm-dd hh:mm:ss
enddate	DATETIME	Hora i data de finalització. Format: aaaa-mm-dd hh:mm:ss
status	VARCHAR(12)	Estat de la tasca.
id_calendar	VARCHAR(36)	Identificador del calendari al qual pertany la tasca.
date_modified	DATETIME	Data i hora en què s'ha modificat la tasca. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la tasca està eliminada (1) o no (0).

Taula 143: Mòdul SS\_Task



## Mòdul SS\_Event

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de l'esdeveniment.
message	VARCHAR(128)	Missatge de l'esdeveniment.
date_entered	DATETIME	Hora i data de creació de l'esdeveniment. Format: aaaa-mm-dd hh:mm:ss
type	VARCHAR(10)	Tipus de l'esdeveniment. Serveix per saber si afecta a una tasca, a un calendari o a un grup.
id_affected	VARCHAR(36)	Identificador de l'objecte afectat que pot ser una tasca, un calendari o un grup.

Taula 144: Mòdul SS\_Event

## Mòdul SS\_PendingResolution

Columna	Tipus	Descripció
id	CHAR(36)	Identificador del pending resolution.
message	VARCHAR(128)	Missatge del pending resolution.
affectedfields	TEXT	Nom de tots els camps afectats del pending resolution, separats per un caràcter separador.  Per exemple, nom#descripcio#ubicacio.
fieldsvalues	TEXT	Valors dels camps afectats del pending resolution, separats per un caràcter separador.  Per exemple, Tasca1#descr#Aquí.
idsfieldsvalues	TEXT	Identificadors dels valors del pending resolution, separats per un caràcter separador. Utilitzat només quan el pending resolution afecta relacions, ja que en aquest cas els valors són els noms dels objectes afegits o eliminats, i aquest camp contindrà els seus identificadors.
type	VARCHAR(16)	Indica el tipus del pending resolution: Meeting o Pending Approval.
active	TINYINT	Si és un meeting, indica si està actiu (1) o no (0).

id_task	VARCHAR(36)	Identificador de la tasca afectada pel pending resolution. Si afecta a un calendari, aquest valor és nul.
id_calendar	VARCHAR(36)	Identificador del calendari afectat pel pending resolution. Si afecta a una tasca, aquest valor és nul.
date_entered	DATETIME	Hora i data de creació del pending resolution. Format: aaaa-mm-dd hh:mm:ss
date_modified	DATETIME	Hora i data de modificació del pending resolution. Format: aaaa-mm-dd hh:mm:ss
created_by	CHAR(36)	Identificador de l'usuari que ha creat el pending resolution.
deleted	TINYINT	Indica si el pending resolution està eliminat (1) o no (0).
chosenoption	VARCHAR(2)	Indica si el pending approval ha estat acceptat (1), rebutjat (0) o encara no ha finalitzat (-1).

*Taula 145: Mòdul SS\_PendingResolution*

### Mòdul SS\_Vote

Columna	Tipus	Descripció
id	CHAR(36)	Identificador del vot.
choice	TEXT	Opcions del meeting escollides per l'usuari, separades per un caràcter separador. Per exemple, T1#d1#Aquí.
id_meeting	VARCHAR(36)	Identificador del meeting al que pertany el vot.
id_user	VARCHAR(36)	Identificador de l'usuari al que pertany el vot.

*Taula 146: Mòdul SS\_Vote*

### Mòdul SS\_Belongs (associació Pertany)

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
iduser	VARCHAR(36)	Identificador de l'usuari relacionat.
idgroup	VARCHAR(36)	Identificador del grup relacionat.

date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 147: Mòdul SS\_Belongs*

### **Mòdul SS\_Requests (associació Solicita)**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
iduser	VARCHAR(36)	Identificador de l'usuari relacionat.
idgroup	VARCHAR(36)	Identificador del grup relacionat.
date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 148: Mòdul SS\_Requests*

### **Mòdul SS\_Shares (associació Comparteix)**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
idcalendar	VARCHAR(36)	Identificador del calendari relacionat.
idgroup	VARCHAR(36)	Identificador del grup relacionat.
date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 149: Mòdul SS\_Shares*

### **Mòdul SS\_Participates (associació Participa)**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
iduser	VARCHAR(36)	Identificador de l'usuari relacionat.
idtask	VARCHAR(36)	Identificador de la tasca relacionada.

date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 150: Mòdul SS\_Participates*

### **Mòdul SS\_Assists (associació Assisteix)**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
idtask	VARCHAR(36)	Identificador de la tasca relacionada.
idgroup	VARCHAR(36)	Identificador del grup relacionat.
date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 151: Mòdul SS\_Assists*

### **Mòdul SS\_Precedes (associació Precedeix)**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
idprevious	VARCHAR(36)	Identificador de la tasca precedent.
idfollowing	VARCHAR(36)	Identificador de la tasca següent.
date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 152: Mòdul SS\_Precedes*

### **Mòdul SS\_Reply**

Columna	Tipus	Descripció
id	CHAR(36)	Identificador de la relació.
iduser	VARCHAR(36)	Identificador de l'usuari relacionat.
idpendingapproval	VARCHAR(36)	Identificador del pending approval relacionat.

accepted	TINYINT	Indica si l'usuari ha acceptat (1) o rebutjat (0) el pending approval.
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 153: Mòdul SS\_Reply*

### **Mòdul SS\_Receive**

<b>Columna</b>	<b>Tipus</b>	<b>Descripció</b>
id	CHAR(36)	Identificador de la relació.
iduser	VARCHAR(36)	Identificador de l'usuari relacionat.
idevent	VARCHAR(36)	Identificador de l'esdeveniment relacionat.
date_modified	DATETIME	Hora i data de modificació de la relació. Format: aaaa-mm-dd hh:mm:ss
deleted	TINYINT	Indica si la relació està eliminada (1) o no (0).

*Taula 154: Mòdul SS\_Receive*

## 2. Diagrames de seqüència

A continuació es mostren els diagrames de seqüència de disseny del sistema. Cada diagrama on l'actor és l'usuari comença quan l'usuari realitza alguna acció a la interfície gràfica, invocant així una operació al ViewModel corresponent, amb l'excepció dels casos d'ús de login ja que, com s'explicarà a l'apartat d'implementació, les accions de l'usuari invoquen operacions de la pròpia Activity, no hi ha ViewModel..

Als diagrames de seqüència d'especificació es pot observar que molts dels casos d'ús requereixen dues accions per part de l'usuari: la primera carrega la vista corresponent i la segona és la que realitza l'operació en sí. En aquests casos el diagrama de seqüència de la primera operació és trivial i molt semblant en la majoria de casos d'ús. Per aquest motiu només presentarem el diagrama de la primera operació del cas d'ús *Crear calendari*, on la nova vista és una nova Activity, i la primera operació del cas d'ús *Logout*, on la nova vista és dins de la mateixa Activity. En la resta de casos d'ús on la primera operació sigui d'aquest tipus només es realitzarà el diagrama de la segona operació.

A més, per simplificar els diagrames es suposarà que les operacions *Invoke*, que són les que es criden quan l'usuari realitza una acció, pertanyen al ViewModel enlloc de a la classe Command corresponent, ja que sempre serà el ViewModel qui contindrà tots els Commands. Mostrar els Commands als diagrames de seqüència només aconseguiria complicar-los a canvi d'uns avantatges pràcticament nuls.

Finalment, quan es creï una nova Activity des d'un ViewModel es cridarà una operació de la classe Context. En realitat, aquest Context serà sempre una Activity (l'actual), ja que la classe Activity és subclasse de Context, però per evitar confusions en els diagrames s'ha decidit anomenar Context a aquesta classe enlloc de, per exemple, *CreateTaskActivity*. Això és degut al fet que, per iniciar una nova Activity, s'ha de cridar l'operació *startActivity* a l'Activity actual i indicar per paràmetre l'Activity que es vol iniciar, però a efectes del diagrama de seqüència podria semblar que l'operació *startActivity* estigués creant l'Activity actual enlloc de la que volem crear en realitat. Per exemple, si des de la vista del llistat de tasques volem crear una nova tasca, es cridarà l'operació *TaskListActivity::startActivity*, i aquesta operació tindrà com a paràmetre un Intent que indicarà que l'Activity que s'ha d'obrir és *CreateTaskActivity*.

## 2.1 Gestió de comptes

### Login: operació obrirAplicacio()

Quan s'iniciï l'aplicació es comprovarà si existeix un compte d'usuari. Si existeix, es mostrarà la vista de llistar tasques. En cas contrari, s'iniciarà el procés de login.

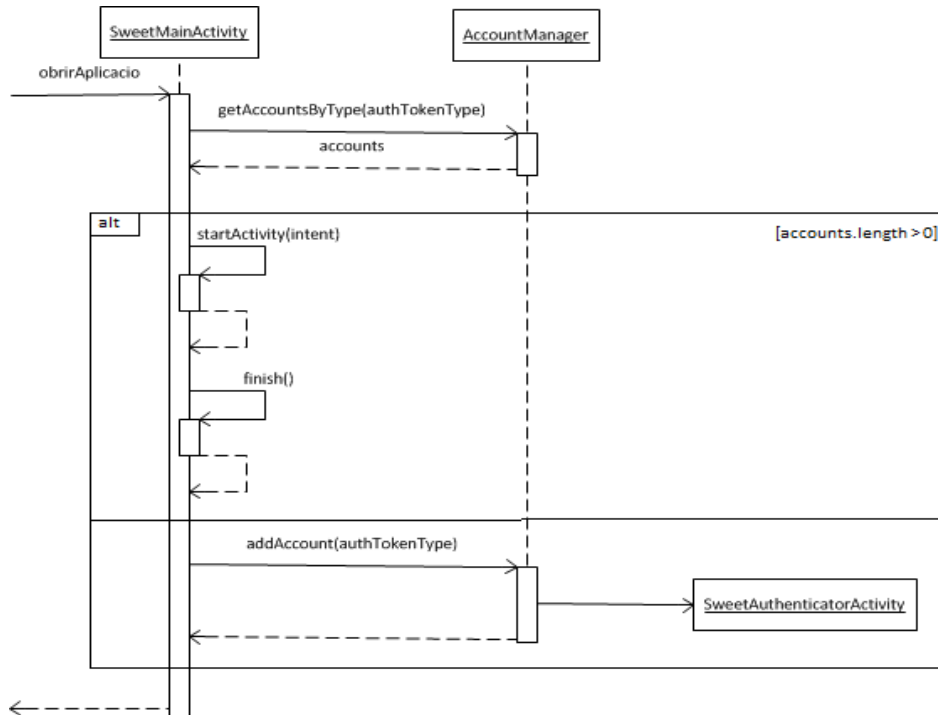


Figura 79: Diagrama seqüència disseny obrirAplicacio

### Login: operació validar(user, pass, servidor, acceptCerts, encriptar)

Per validar les dades aconseguim una instància de *SugarAPI* (classe utilitzada per comunicar-se amb el servidor) i s'envien les dades al servidor per què les validi.

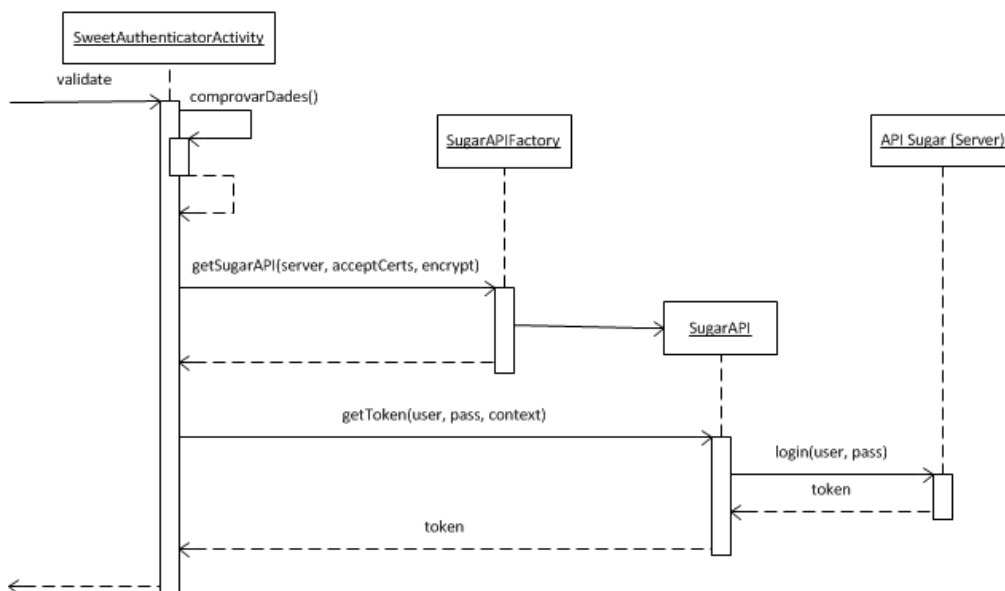


Figura 80: Diagrama seqüència disseny validar

## Login: operació confirmarLogin()

Quan l'usuari confirma que vol fer login després d'haver validat les dades introduïdes cal crear un compte a nivell local per tal de gestionar la renovació de tokens de manera transparent a l'usuari. Com que l'usuari acaba de fer login, hem de demanar al servidor les dades corresponents a l'usuari i desar-les a nivell local. Per fer-ho utilitzem el servei *SweetComService*. Per desar les dades utilitzem el controlador de model, *SweetSyncalModel*.

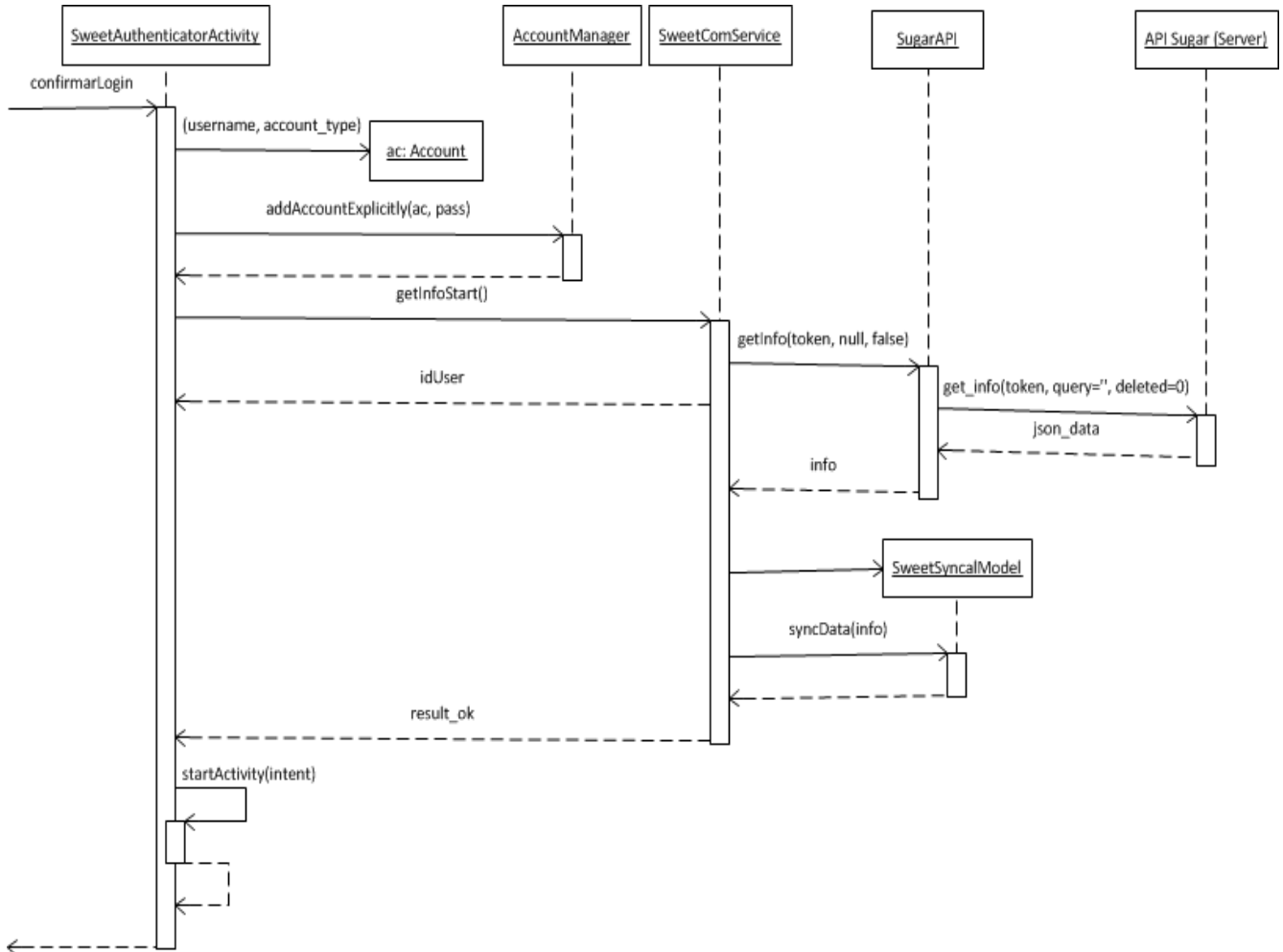


Figura 81: Diagrama seqüència disseny confirmarLogin

L'operació *get\_info* de l'API de Sugar del servidor es pot veure en detall als diagrames de l'operació *Demandar canvis* de l'apartat 2.4.

## Logout: operació logout()

Quan l'usuari indica que vol fer logout únicament cal mostrar una vista de confirmació (*dialog*).



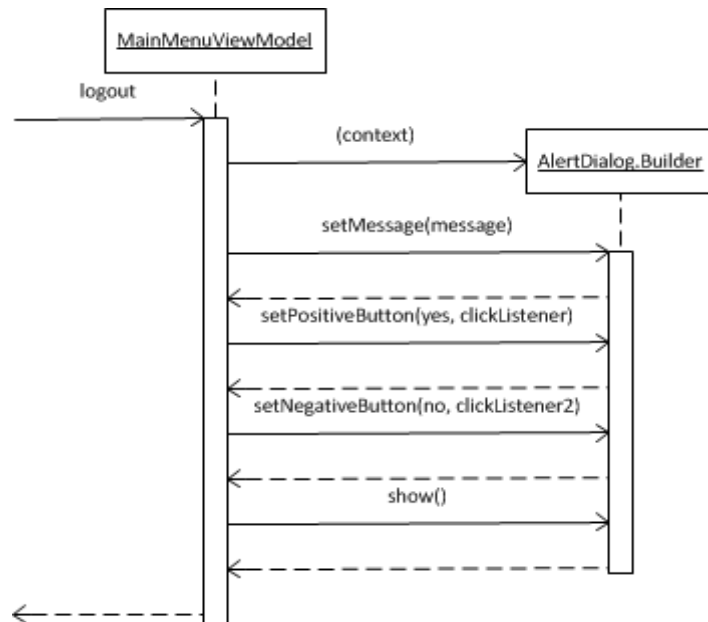


Figura 82: Diagrama seqüència disseny logout

### Logout: operació confirmarLogout()

L'usuari ha confirmat que vol fer logout. Com que pot ser que hi hagi tasques no sincronitzades amb el servidor que es perdrien en cas de fer logout intentem sincronitzar-les. El procés de sincronitzar les tasques és com a l'operació *Enviar tasques* de l'apartat 2.4.

Si hi ha algun error durant la sincronització d'aquestes tasques o bé encara hi ha tasques per sincronitzar després de finalitzar l'enviament (algunes tasques poden no ser tractades per diferents motius sense que això impliqui que hi ha hagut un error durant la sincronització) cal demanar a l'usuari si vol continuar amb l'operació de logout encara que això impliqui perdre informació.

Si no hi ha cap error es procedirà a eliminar el compte a nivell local (Account d'Android) per tal que l'usuari no pugui seguir interaccionant amb el servidor a no ser que torni a fer login. Aquesta operació és asíncrona, i quan finalitza es crida un procés dins de la classe AccountManagerCallback que acabarà de realitzar l'operació de logout: cancel·larà totes les alarmes, buidarà la base de dades i eliminarà totes les notificacions que es puguin estar mostrant a l'usuari. Finalment es tancarà la vista de l'aplicació.

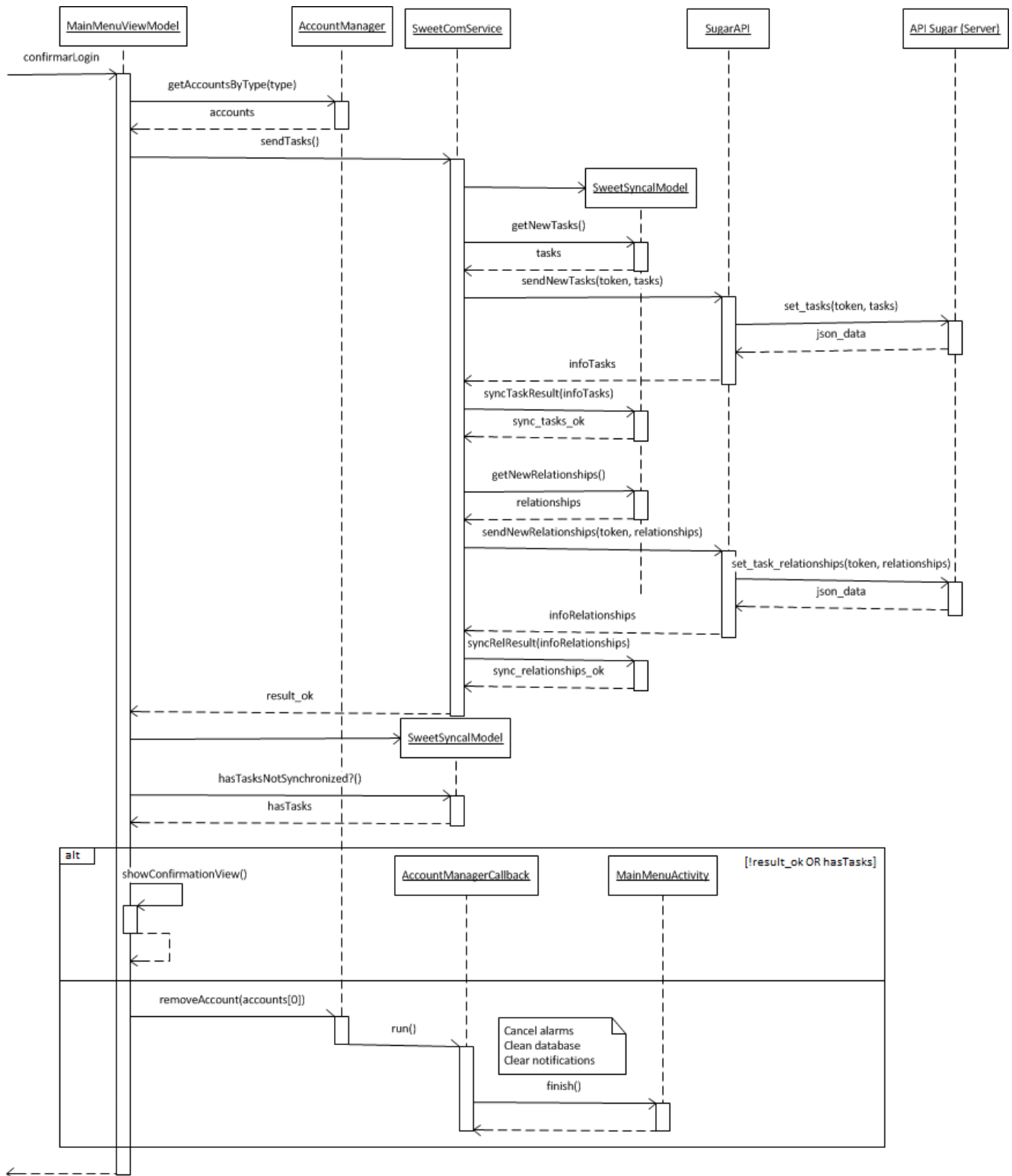


Figura 83: Diagrama seqüència disseny confirmarLogout

Les operacions *set\_tasks* i *set\_task\_relationships* de l'API de Sugar del servidor es poden veure en detall als diagrames de l'operació *Enviar tasques* de l'apartat 2.4.

El diagrama de l'operació *showConfirmationView()* és exactament igual al de l'operació *logout()*. Si l'usuari confirma, es crida *removeAccount(accounts[0])*.

## Veure estadístiques: operació veureEstadistiques()

Per veure les estadístiques s'inicia l'Activity corresponent, que recuperarà les dades de la base de dades local i les mostrarà per pantalla. La major part de les operacions les realitza el ViewModel enloc de l'Activity ja que estem utilitzant el patró MVVM.

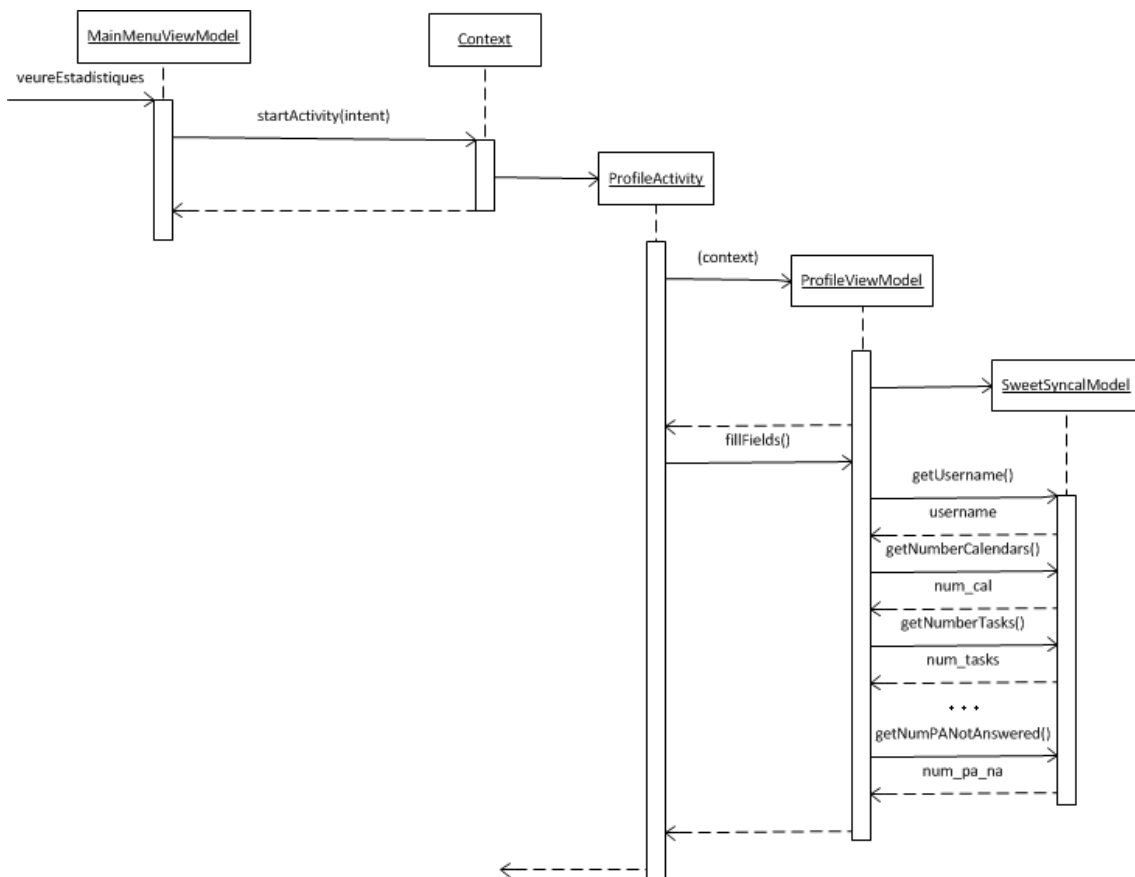


Figura 84: Diagrama seqüència disseny veureEstadistiques

El ProfileViewModel realitza més de 8 crides al model. Per evitar ampliar el diagrama amb un seguit de crides iguals s'ha optat per representar-ne només quatre per tal que es vegi el format d'aquestes.

## 2.2 Gestió de calendaris

### Crear calendari: operació altaCalendari()

Quan l'usuari indica que vol crear un calendari senzillament es mostrarà la vista de creació de calendaris. Aquesta vista contindrà un llistat de possibles grups participants que s'hauran recuperat de la base de dades local.

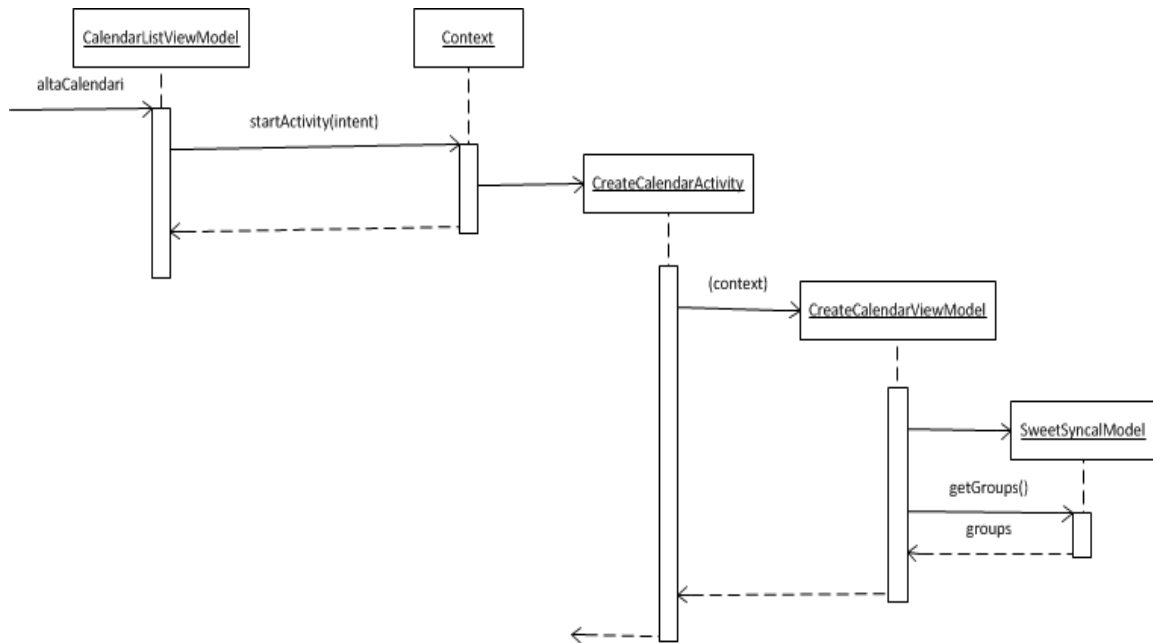


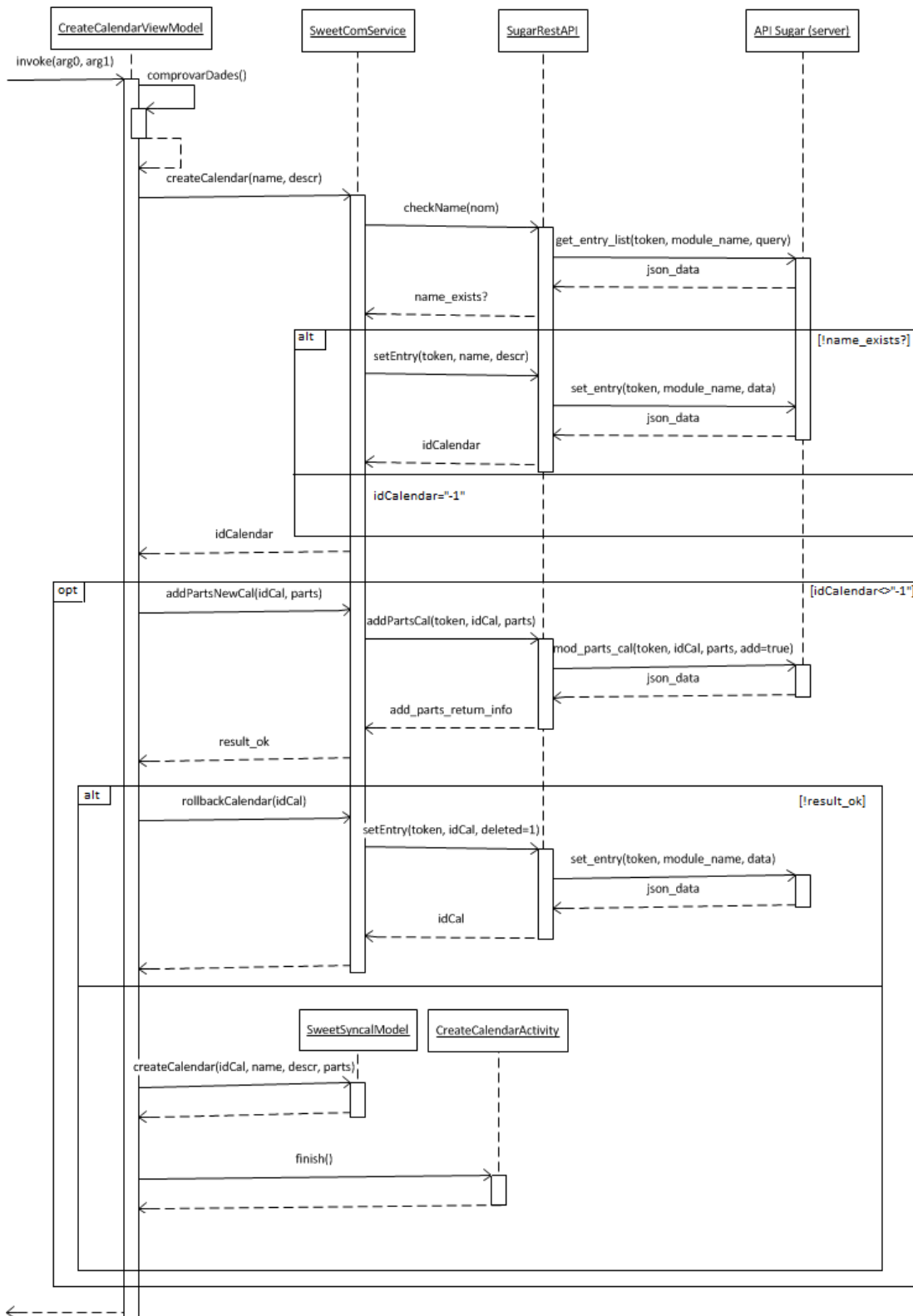
Figura 85: Diagrama seqüència disseny altaCalendari

### Crear calendari: operació crearCalendari(nom, descripció, participants)

Per a crear un calendari primer de tot cal comprovar que les dades siguin correctes. Si ho són, intentem crear el calendari al servidor. Els noms dels calendaris són únics, així que abans de crear-lo haurem de comprovar que no existeix ja un calendari amb aquest nom.

Si el calendari s'ha creat correctament intentarem afegir-li els participants al servidor. Si aquesta operació falla haurem de fer un rollback, ja que haurem de cancel·lar tota l'operació de crear el calendari. Com que Sugar no té un sistema de transaccions i *rollbacks* hem d'eliminar el calendari manualment.

Si, en canvi, els participants s'afegeixen correctament, procedirem a crear el calendari a nivell local i a afegir-li els participants. Si no hi ha cap error podem considerar que el calendari ha estat creat, ja que existeix tan al servidor com a nivell local. Així doncs, podem finalitzar la vista de creació de calendaris.



**Figura 86: Diagrama seqüència disseny crearCalendar**

L'operació `set_entry` de l'API de Sugar del servidor és una operació interna de Sugar, no creada per nosaltres.

L'operació `mod_parts_cal` de l'API de Sugar del servidor es pot veure en detall als diagrames de l'operació *Afegir participants calendari* d'aquest mateix apartat.

## Modificar calendari: operació modificarCalendari(nom, descripció)

L'operació de modificar calendari és molt semblant a la de crear, amb la diferència que no es modifiquen els participants i que cal tractar els pending approvals.

Com s'explica a l'especificació dels casos d'ús, quan s'intenta modificar un calendari que té més participants s'intenta crear o modificar un pending approval. És per aquest motiu que la variable *return\_info* que retorna la crida *modify\_calendar* pot contenir o bé un conjunt de pending approvals o bé únicament si el resultat ha estat correcte o no. En cas de retornar pending approvals, aquests s'han de desar o modificar a la base de dades local. Si *result\_ok = 0* vol dir que no hi ha més participants, així que modificarem el calendari a nivell local.

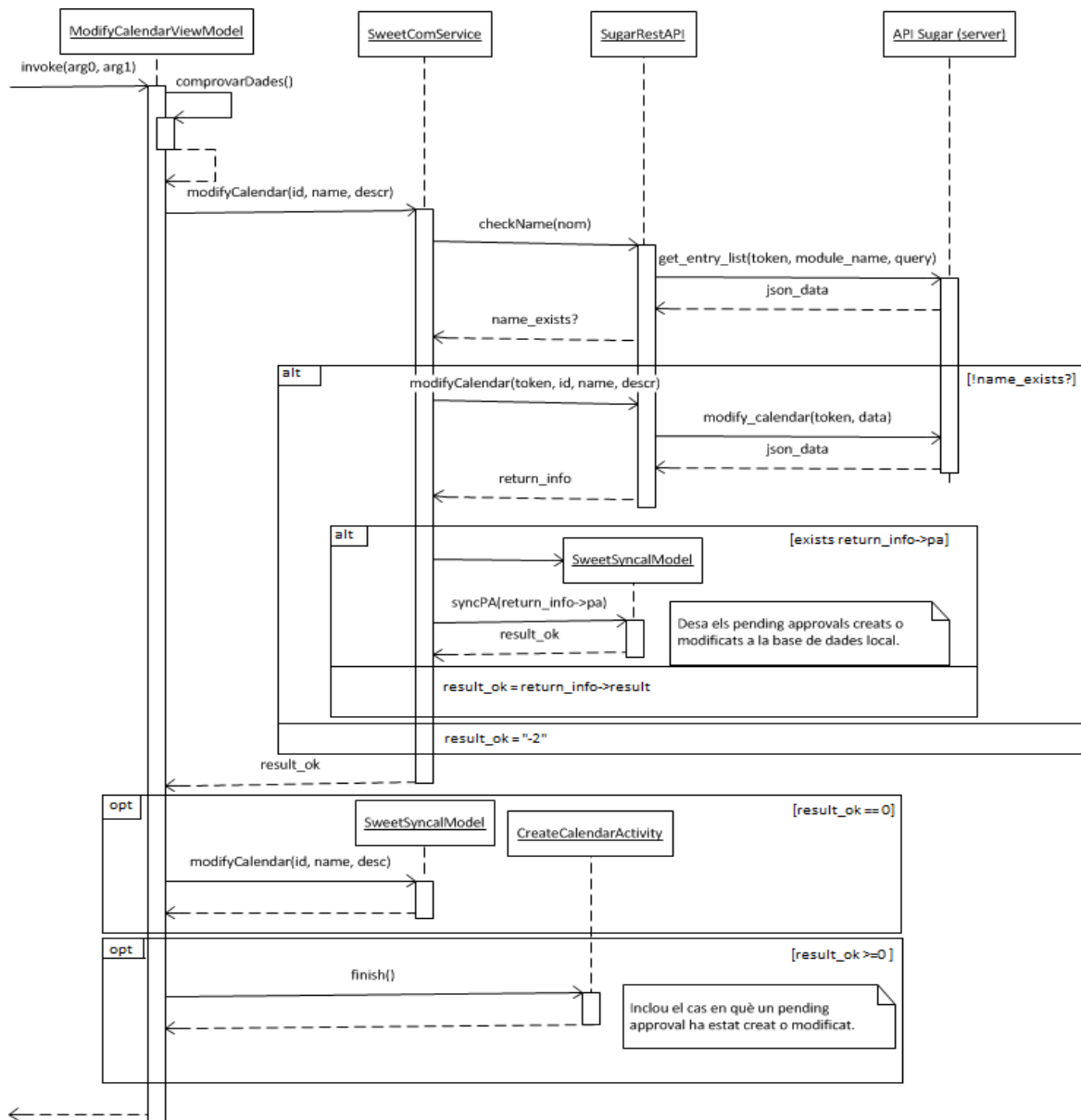


Figura 87: Diagrama seqüència disseny modificarCalendari

A continuació es pot observar en detall la funció *modify\_calendar* de l'API de Sugar localitzada al servidor.

## Sugar API: operació modify\_calendar

L'explicació de la funcionalitat es pot trobar després de la figura.

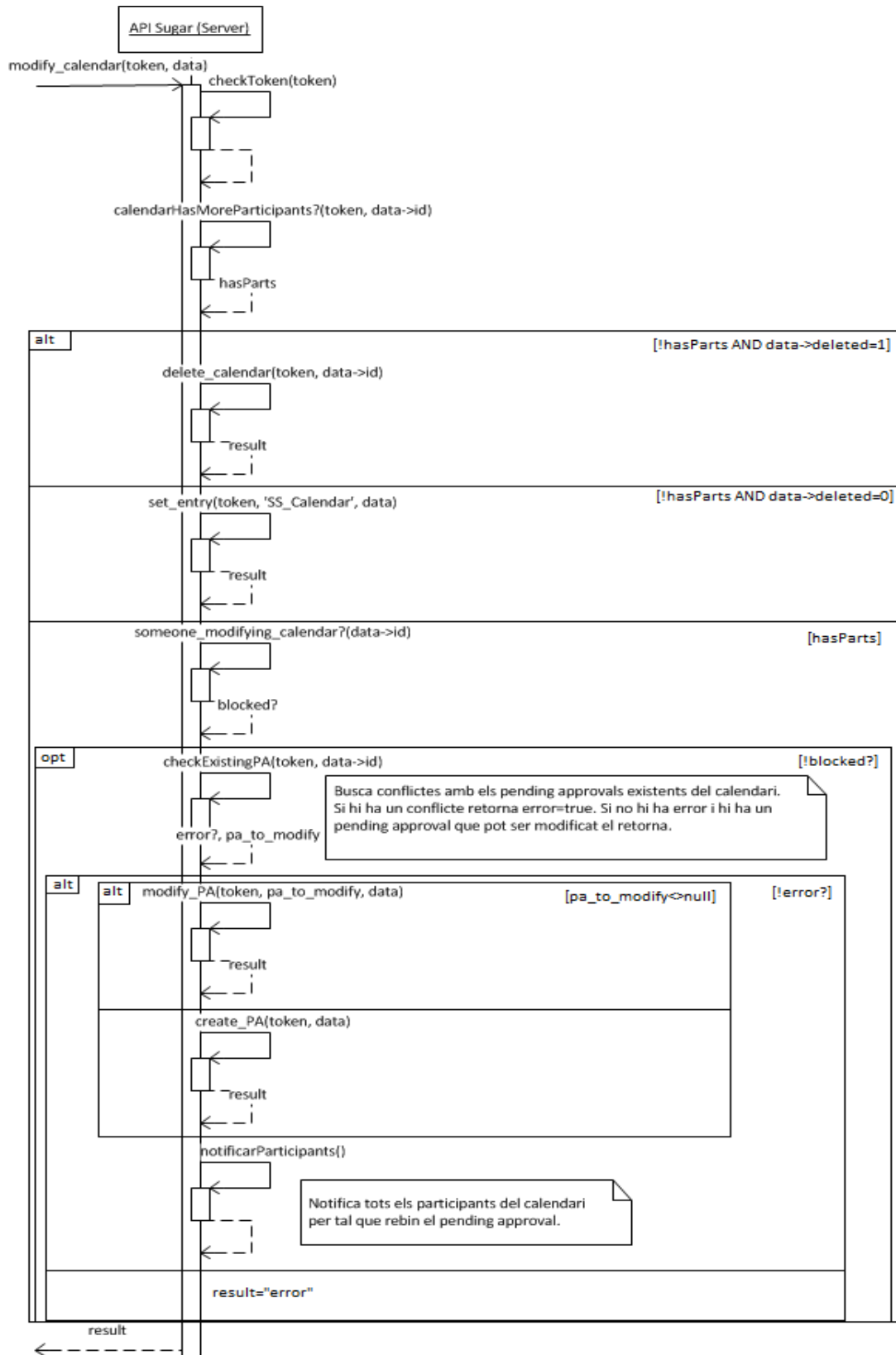


Figura 88: Diagrama seqüència disseny modify\_calendar

Totes les operacions al servidor comencen comprovant que el token de l'usuari sigui correcte i tingui accés als mòduls. A continuació cal comprovar si el calendari té més participants. Si no en té realitzem l'operació pertinent (eliminar o modificar). Si en té cal comprovar que no hi hagi algú modificant el calendari en aquest moment.

Si ningú està modificant el calendari haurem de revisar els pending approvals existents. Si hi ha conflicte amb algun pending approval l'operació acaba i es retorna error. Si no hi ha conflicte però hi ha un pending approval que pot ser modificat (va ser creat pel mateix usuari) el modifiquem introduint-hi la nova informació enviada. Si no hi ha conflicte i tampoc hi ha cap pending approval per modificar, en creem un de nou.

### Eliminar calendari: operació confirmarEliminarCalendari()

La funcionalitat d'eliminar calendari funciona pràcticament igual que la de modificar, l'única diferència és que com que ens trobem a la vista de veure informació només la finalitzarem si no hi ha més participants.

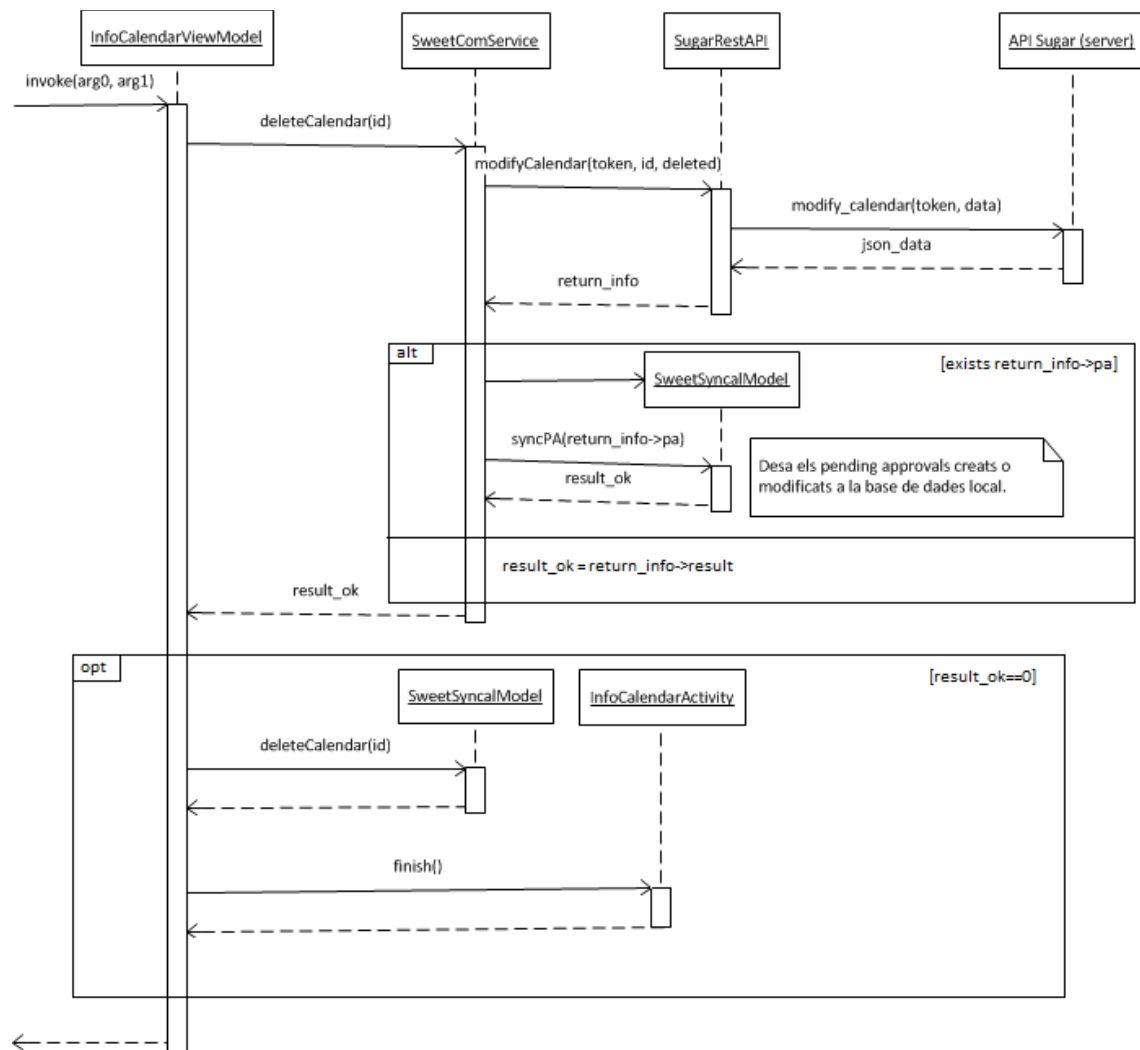


Figura 89: Diagrama seqüència disseny confirmarEliminarCalendari



## Veure informació calendari: operació mostrarInfoCalendari(id)

Operació molt senzilla. Inicia l'Activity corresponent, que crea el ViewModel i aquest recupera la informació necessària de la base de dades local. En aquest cas, la informació del calendari i els seus participants.

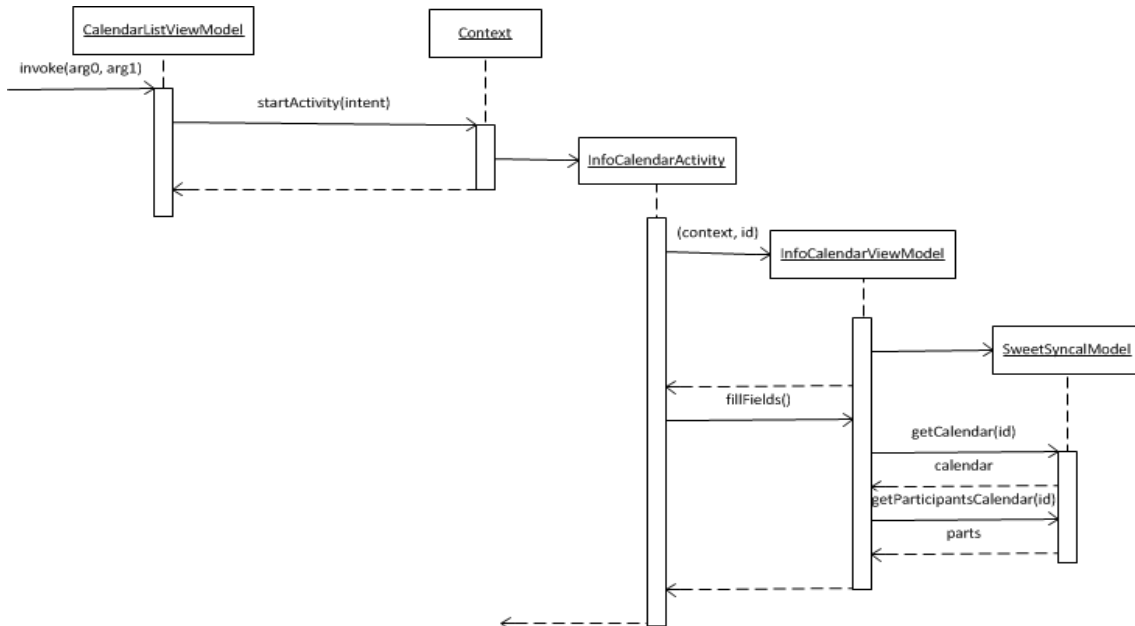


Figura 90: Diagrama seqüència disseny mostrarInfoCalendari

## Afegir participant calendari: operació enviarParticipantsCalendari(grups)

Primer de tot cal comprovar que la llista de grups participants no sigui buida. Si no ho és, intentarem afegir els participants al servidor. Si el calendari ja té més usuaris participants abans d'afegir aquests grups haurem de tractar els pending approvals corresponents.

Com s'explica a l'especificació dels casos d'ús, quan s'intenten modificar els participants d'un calendari que té més participants s'intenta crear o modificar un pending approval. És per aquest motiu que la crida *modifyPartsCalendar* pot retornar o bé un conjunt de pending approvals o bé únicament si el resultat ha estat correcte o no. En cas de retornar pending approvals, aquests s'han de desar o modificar a la base de dades local.

Si `result_ok = 0` vol dir que no hi ha més participants, així que els podem afegir a la base de dades local. Finalment, si no hi ha hagut cap error, finalitzem la vista d'addició de participants.

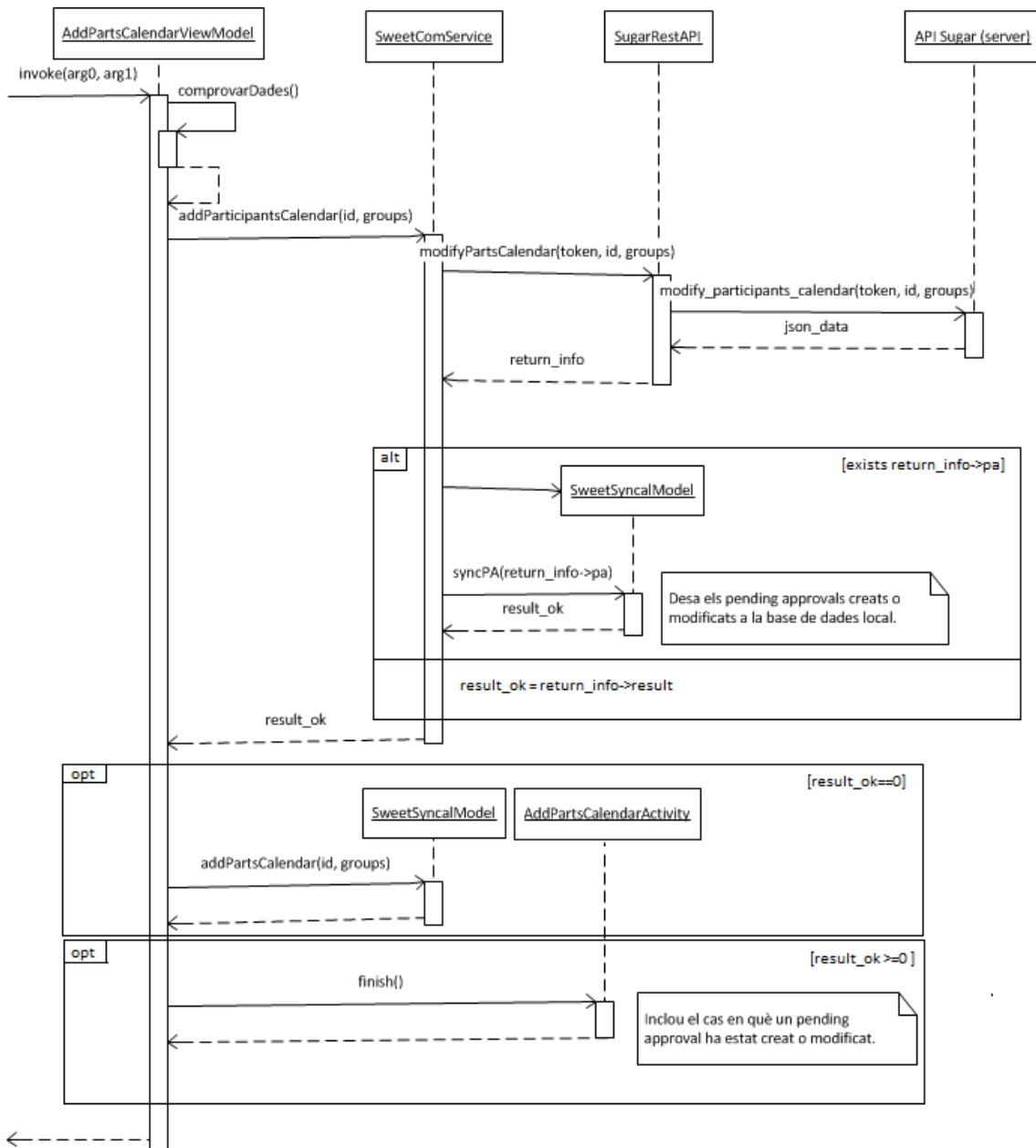


Figura 91: Diagrama seqüència disseny enviarParticipantsCalendar

A continuació es pot observar en detall la funció *modify\_participants\_calendar* de l'API de Sugar localitzada al servidor.

### Sugar API: operació *modify\_participants\_calendar*

Aquesta operació és molt semblant a *modify\_calendar*. Comprovem el token i mirem si el calendari té més participants. Si no en té, afegim o eliminem els participants directament (la funció serveix per totes dues coses). Cada canvi fet el desarem a la variable *entries* per retornar-ho a l'usuari i que pugui saber quins participants s'han afegit o eliminat.

Si el calendari té més participants cal fer el mateix que a *modify\_calendar*: comprovar que no estigui sent modificat, buscar conflictes amb pending approvals existents i crear o modificar un pending approval. Finalment, cal notificar als usuaris participants que hi ha hagut canvis perquè rebin el nou pending approval.

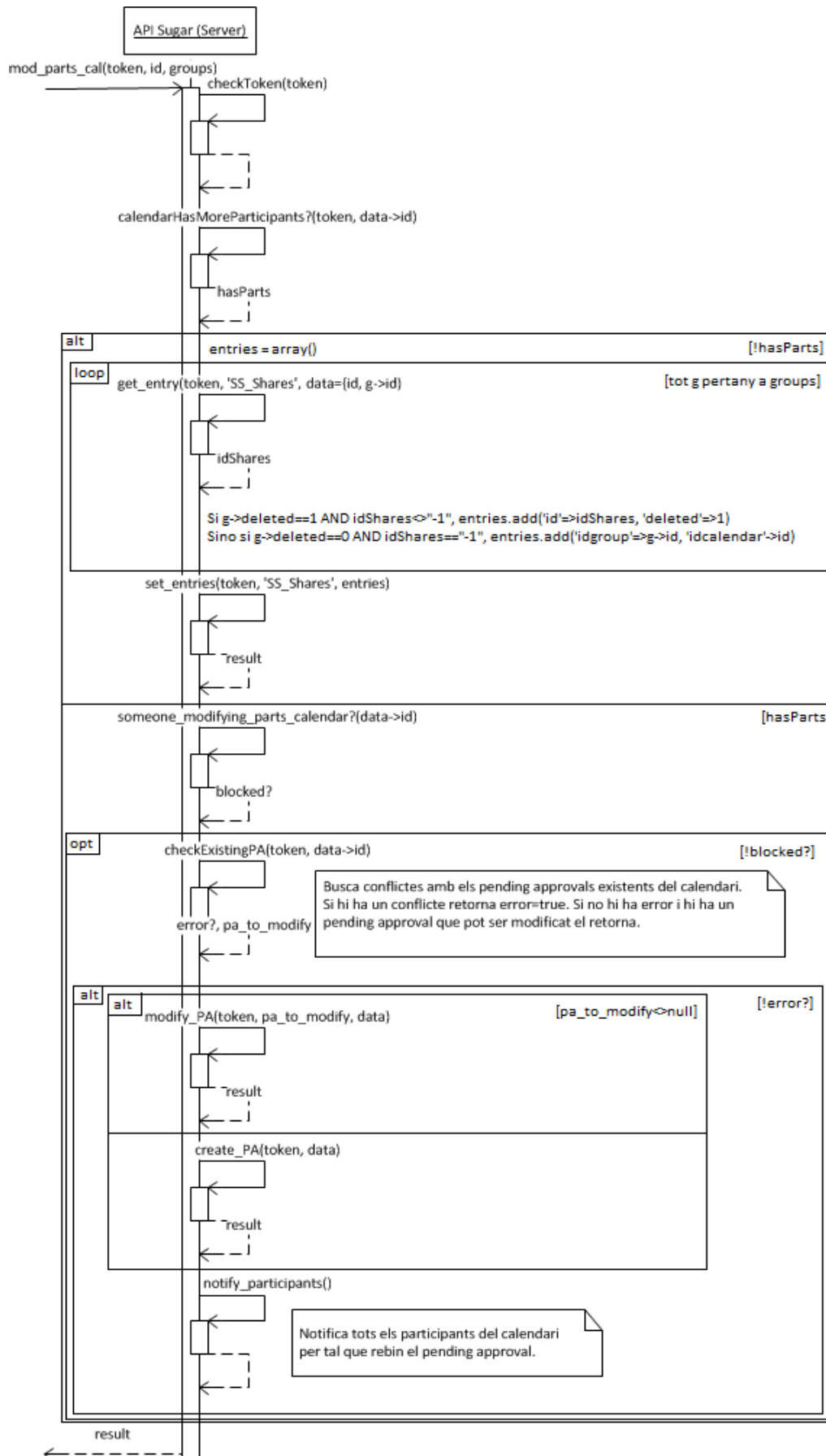


Figura 92: Diagrama seqüència disseny modify\_participants\_calendar

## Eliminar participant calendari: operació confirmarElimPartCal()

Aquesta operació és molt semblant a afegir participants. L'única diferència és que no permetrem realitzar l'operació si el grup participant eliminat és l'últim del calendari, en aquest cas caldria eliminar el calendari directament.

L'operació *modify\_participants\_calendar* es pot veure a la pàgina anterior.

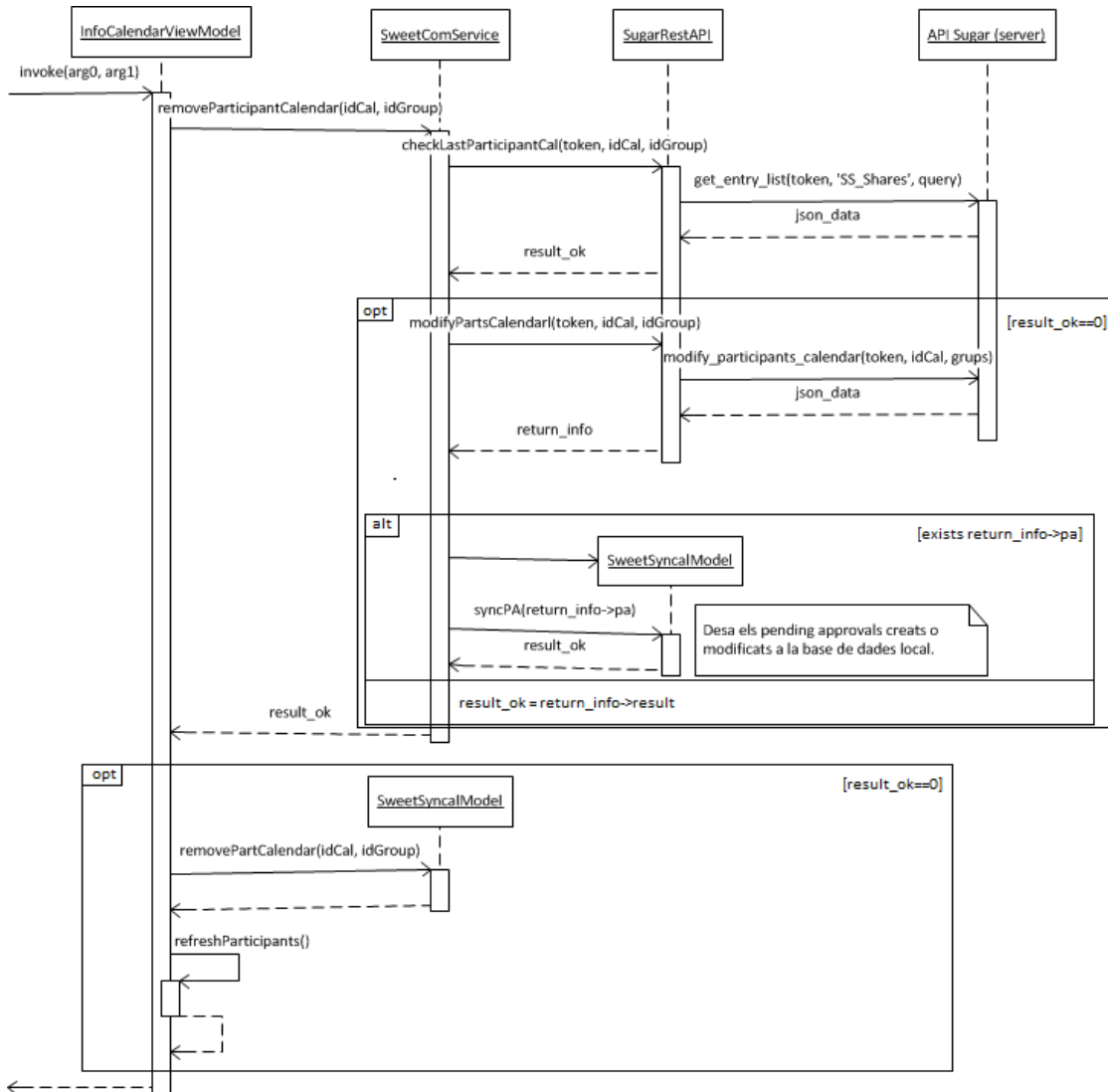


Figura 93: Diagrama seqüència disseny confirmarElimPartCal

## Llistar calendaris: operació llistarCalendaris()

Operació molt senzilla. Crea l'Activity corresponent, que crea el ViewModel i aquest recupera el llistat de calendaris de la base de dades local i els assigna a la vista.

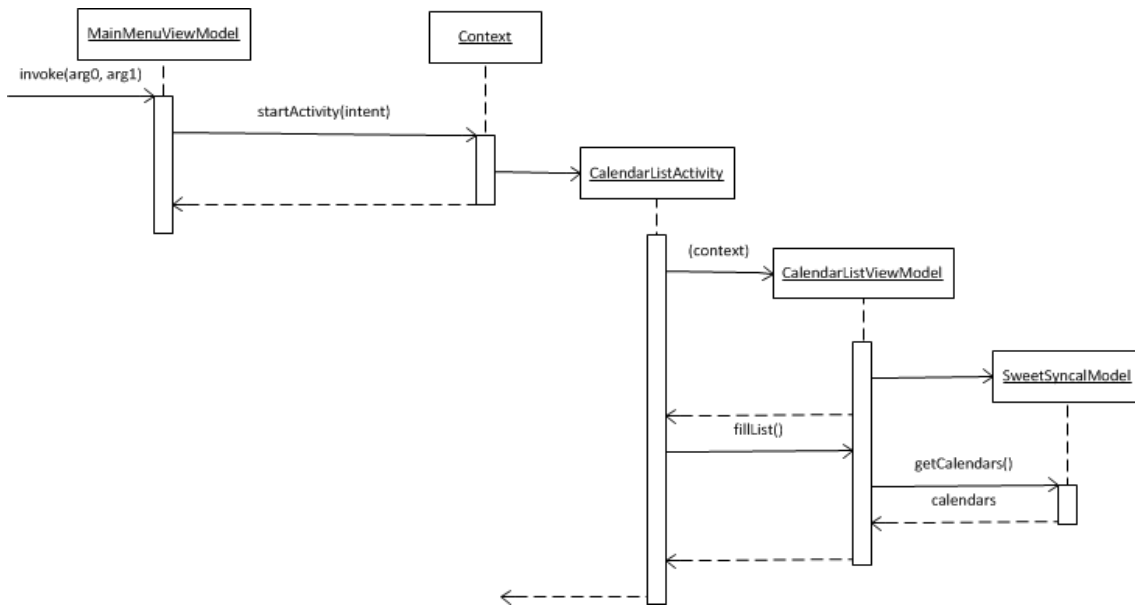


Figura 94: Diagrama seqüència disseny llistarCalendaris

## 2.3 Gestió de tasques

**Crear tasca: operació crearTasca(nom, descr, ubicacio, idcal, estat, dataI, dataF, users, grups)**

Com que les funcionalitats directament relacionades amb tasques es poden realitzar *offline* no hi ha comunicació amb el servidor, es farà durant la sincronització.

Quan l'usuari envia les dades de la tasca comprovem que siguin correctes i, si és així, creem la tasca a nivell local.

En aquest cas d'ús es creen com a molt dues alarmes: la primera s'utilitza per enviar les tasques modificades (cas d'ús *Enviar tasques*), l'altra és el recordatori de la tasca (classe Alarma del model conceptual) i només es programarà si l'estat de la tasca és no començat i l'usuari del dispositiu participa a la tasca creada. Com que el recordatori depèn de les preferències indicades per l'usuari, caldrà recuperar aquestes preferències.

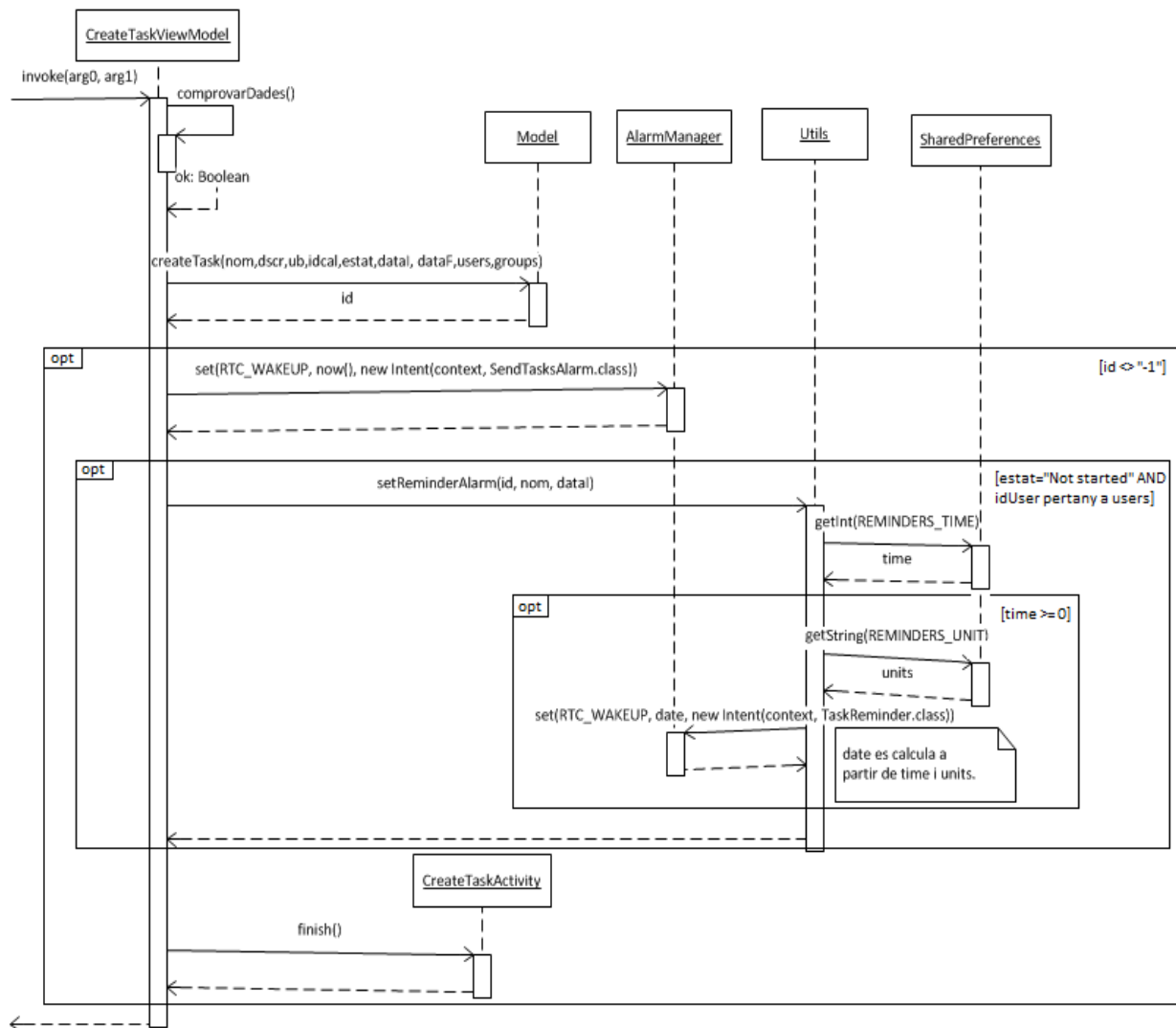


Figura 95: Diagrama seqüència disseny crearTasca

### Modificar tasca: operació `modificarTasca(nom, descr, ubicacio, estat, dataI, dataF)`

L'operació de modificar tasca és molt semblant a la de crear. Comprovem que les dades siguin correctes i, si ho són, modifiquem la tasca. Si no hi ha hagut cap error programem una alarma per enviar les dades (cas d'ús *Enviar tasques*).

En alguns casos caldrà actualitzar l'alarma de recordatori. Si el nou estat de la tasca és "no començat" podem eliminar els recordatoris que hi hagi programats per la tasca, ja que només tenen recordatori les tasques que encara no s'hagin iniciat. Si és el cas contrari, que ara l'estat nou és "no començat", crearem el recordatori.

El diagrama en detall de l'operació `setReminderAlarm` es pot veure al diagrama de l'operació anterior, `crearTasca`.

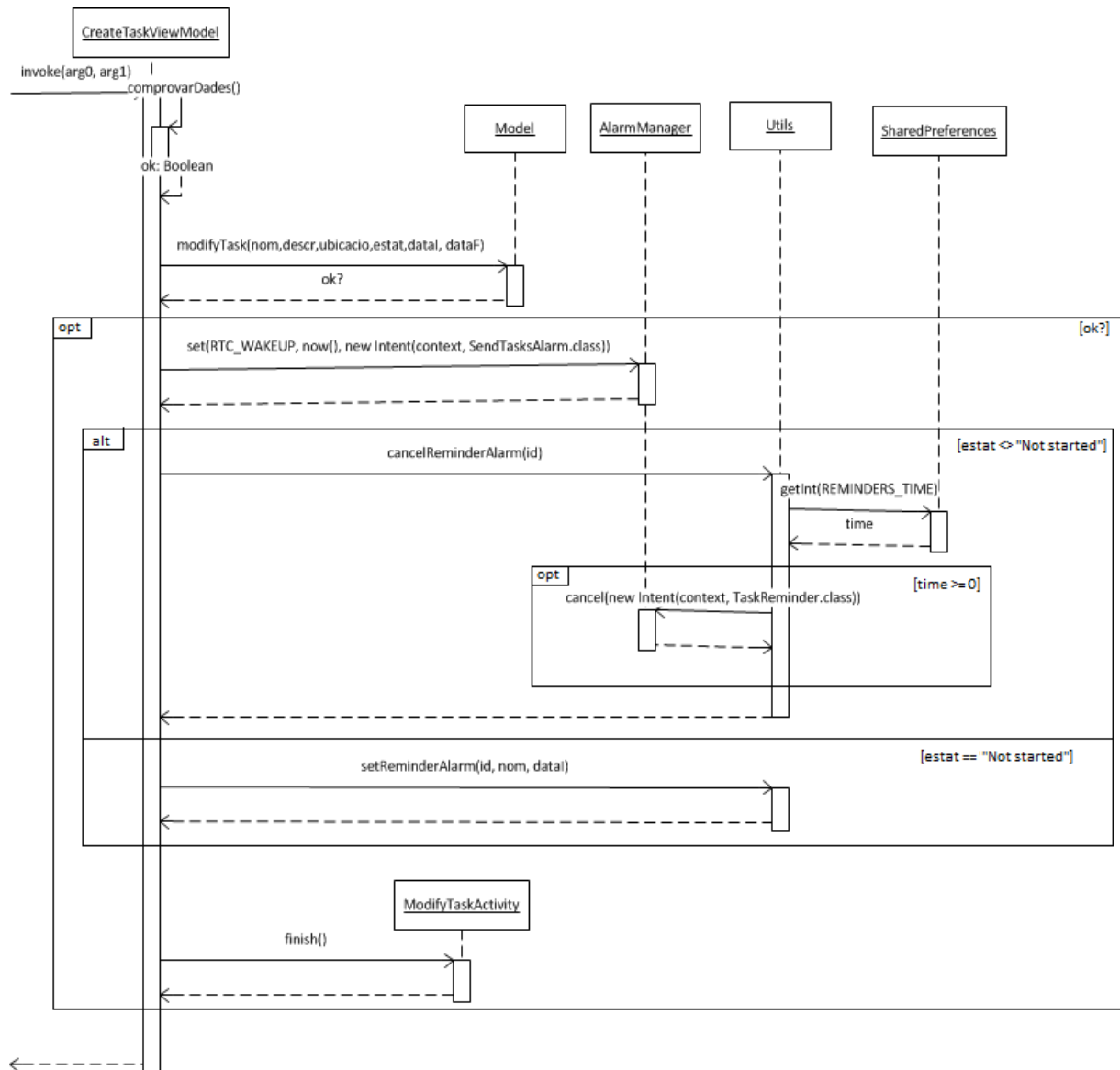


Figura 96: Diagrama seqüència disseny modificarTasca

### Eliminar tasca: operació confirmarElimTasca()

Aquesta operació també és molt semblant a les anteriors. Es marca la tasca com a eliminada a nivell local, es programa l'enviament de tasques i s'elimina l'alarma corresponent. És important remarcar que es marca la tasca com a eliminada, no s'elimina encara, ja que primer s'ha de sincronitzar amb el servidor. Si finalment es confirmés l'eliminació (ja sigui perquè no hi ha més participants com perquè s'ha acceptat el pending approval corresponent) llavors s'eliminaria de la base de dades local. Òbviament, les tasques que estan marcades com a eliminades no seran visibles per a l'usuari.

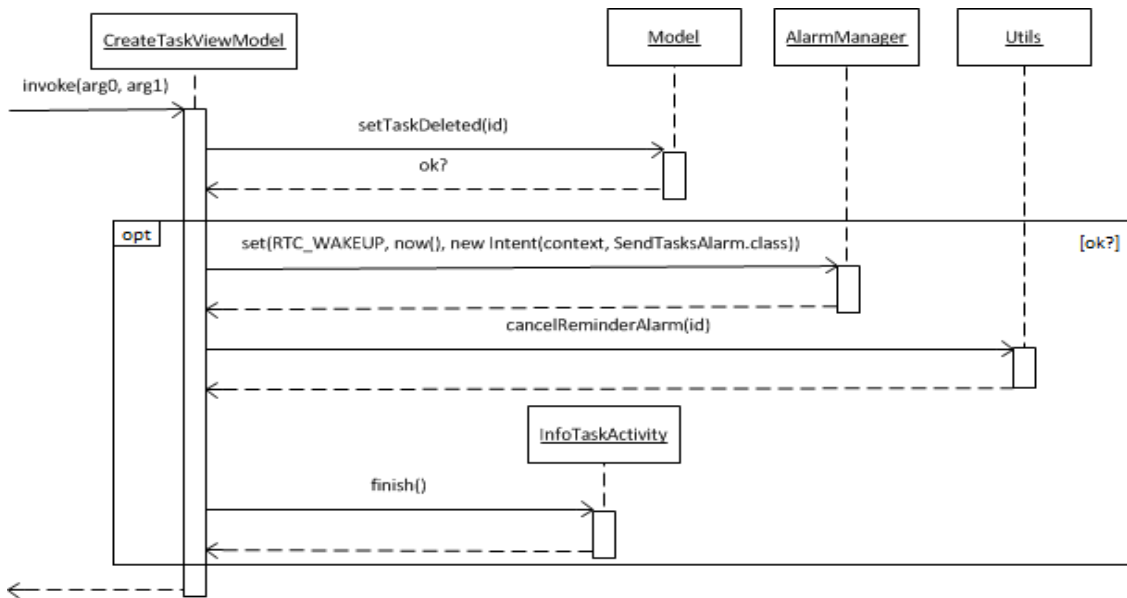


Figura 97: Diagrama seqüència disseny confirmarElimTasca

### Veure informació tasca: operació mostrarInfoTasca(id)

Operació de veure informació. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

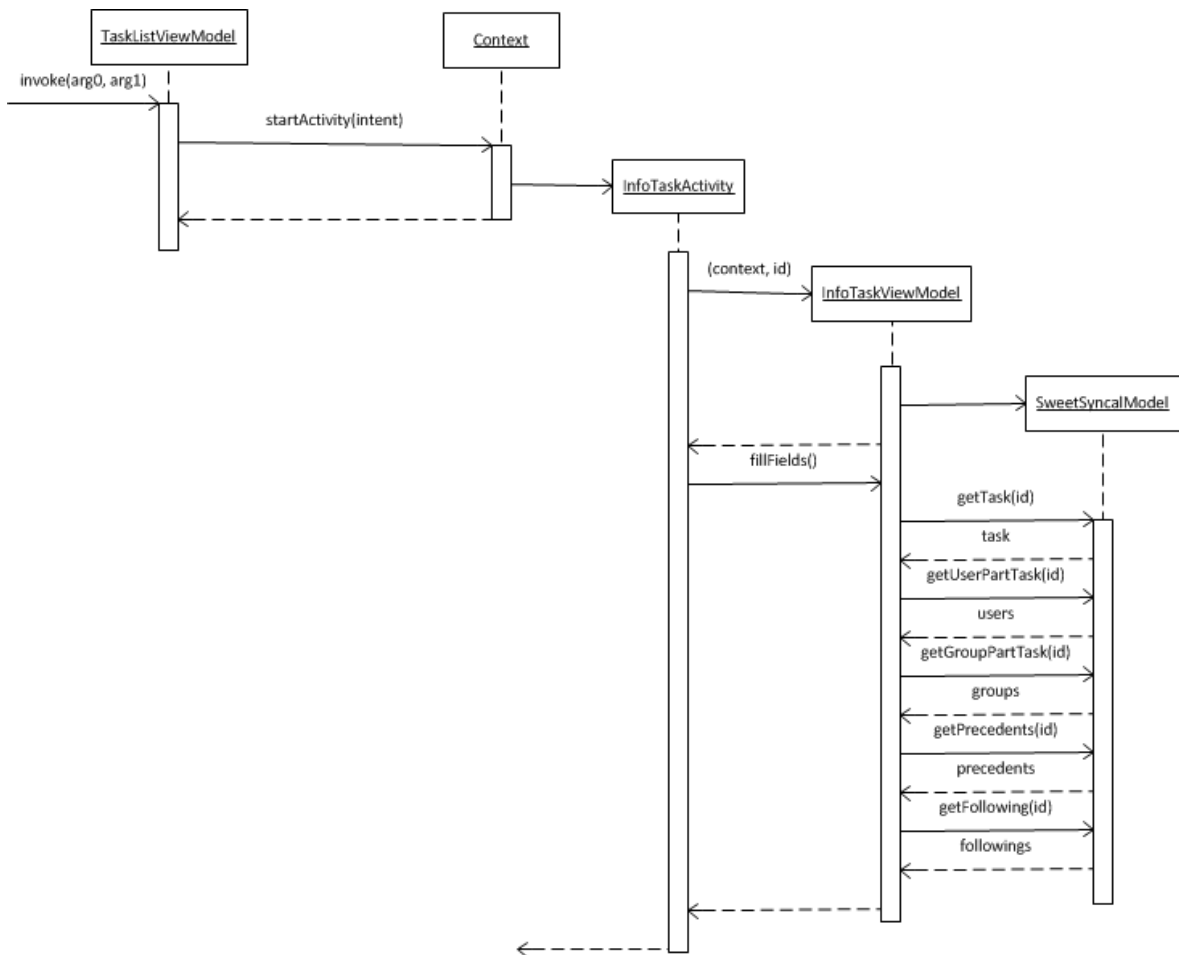


Figura 98: Diagrama seqüència disseny mostrarInfoTasca



### Afegir participant tasca: operació enviarParticipantsTasca(usuaris, grups)

Operació molt semblant a crear/modificar. Es comproven les dades, es fan els canvis a nivell local i es programa l'enviament de dades al servidor.

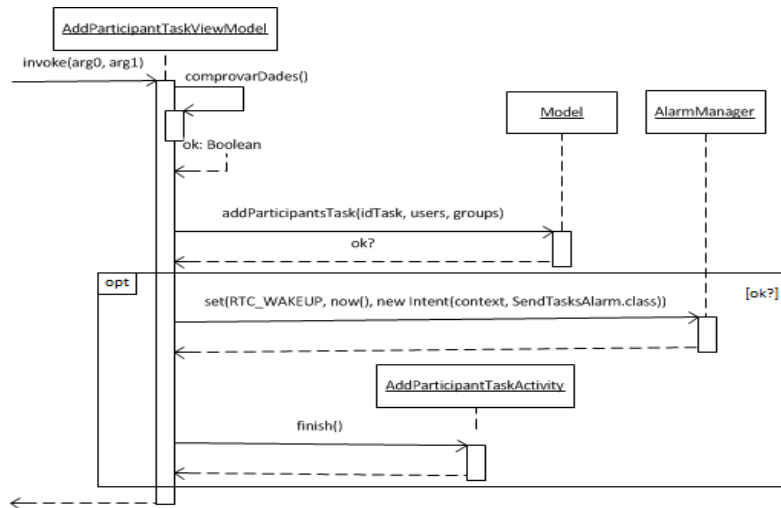


Figura 99: Diagrama seqüència disseny enviarParticipantsTasca

### Eliminar participant tasca: operació confirmarElimPartTasca()

El funcionament és molt semblant a afegir participant, però amb la diferència que si el participant eliminat és l'últim de la tasca es marca la tasca com a eliminada.

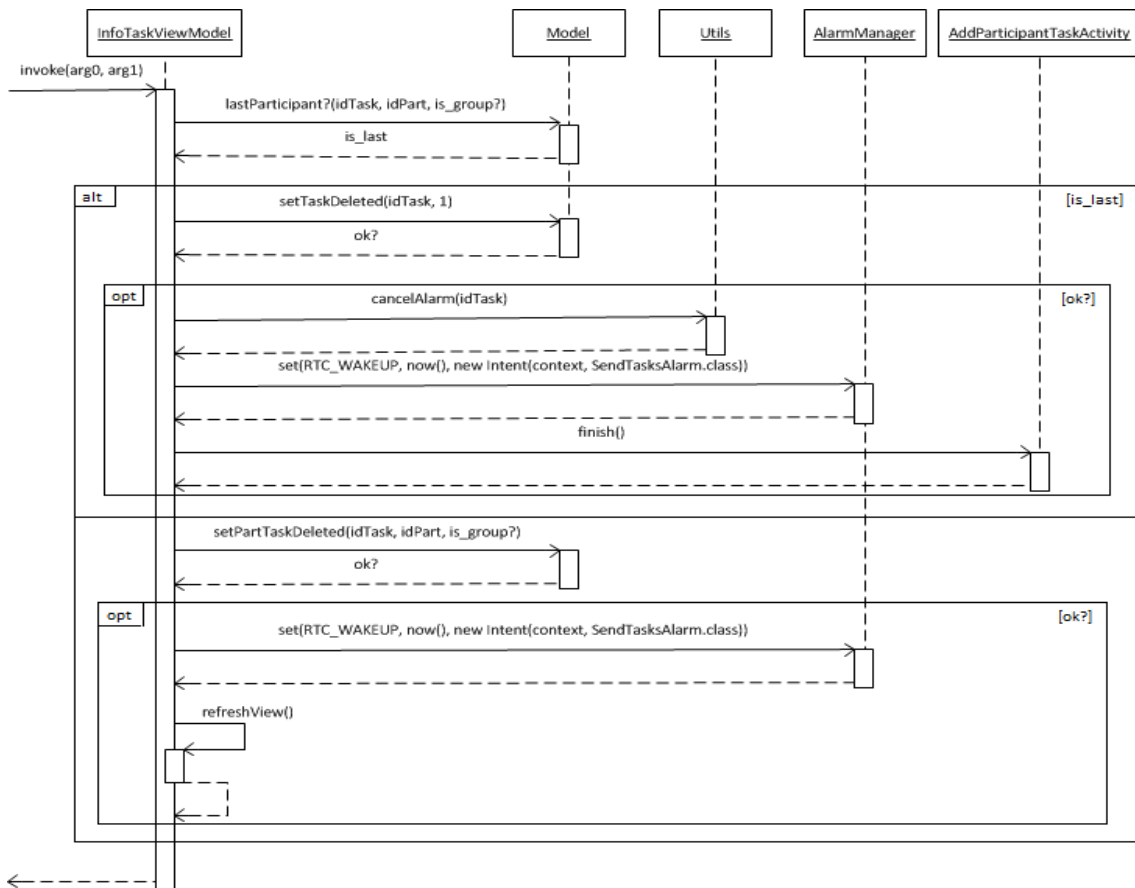


Figura 100: Diagrama seqüència disseny confirmarElimPartTasca

## Afegir precedent tasca: operació enviarPrecedentsTasca(tasques)

Funcionament igual que l'operació d'afegir participant.

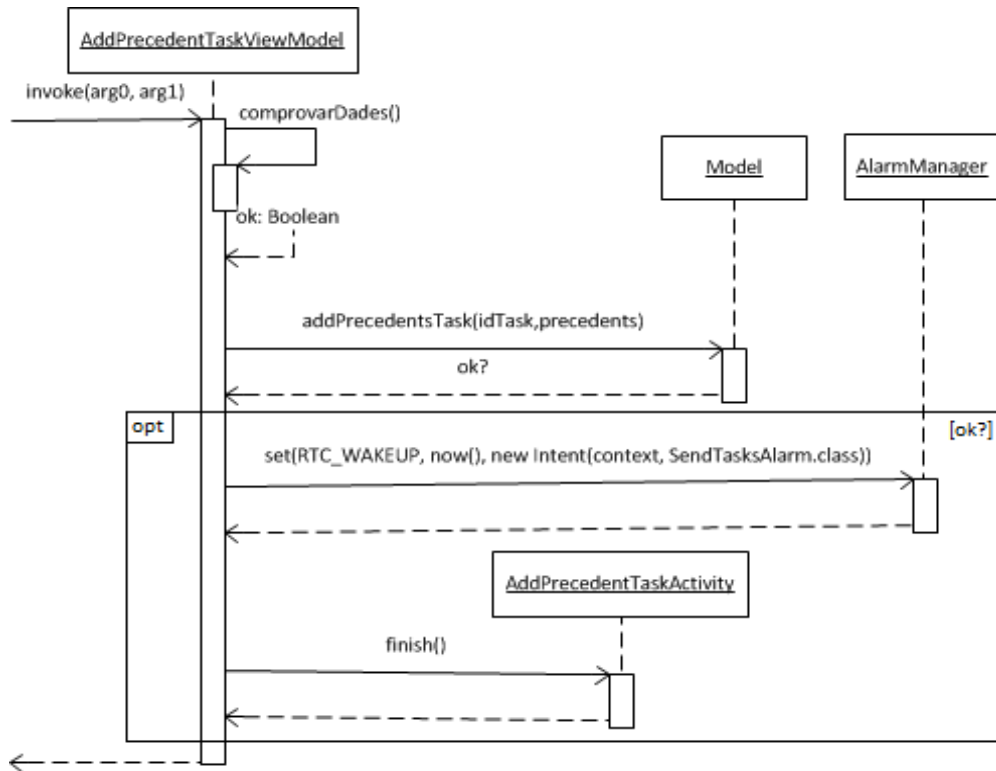


Figura 101: Diagrama seqüència disseny enviarPrecedentsTasca

## Eliminar precedent tasca: operació confirmarElimPrecTasca()

Funcionament igual que afegir precedent, amb la diferència que els precedents es marquen com a eliminats enlloc de ser eliminats directament (veure operació eliminar tasca).

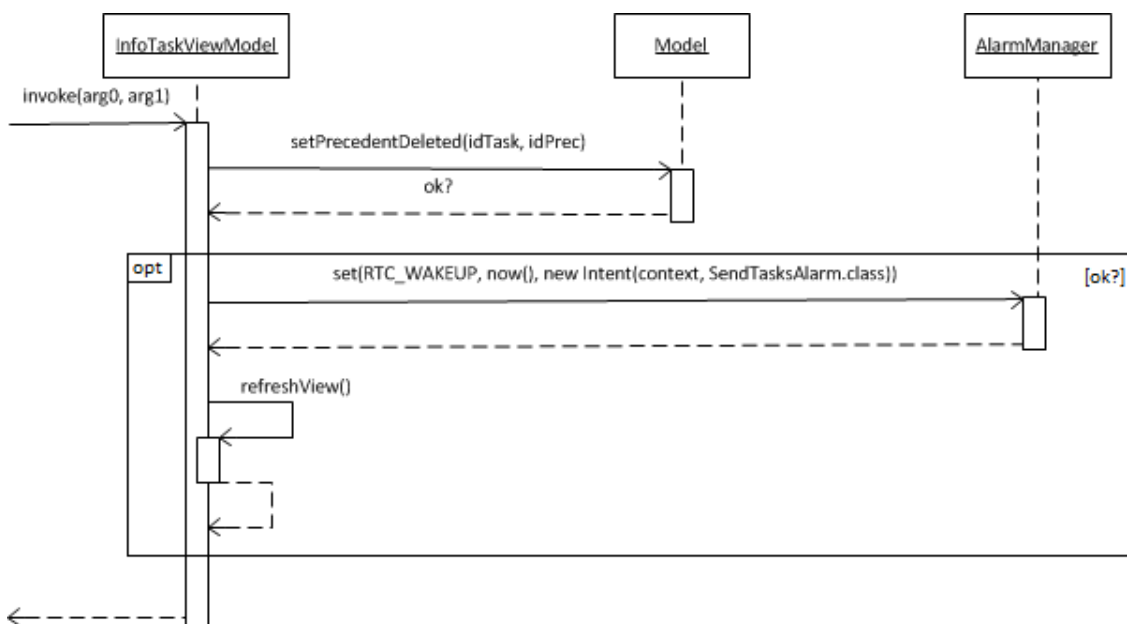


Figura 102: Diagrama seqüència disseny confirmarElimPrecTasca

## Llistar tasques: operació llistarTasques(nomesMeves)

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

El paràmetre *only\_users* que es passa al ViewModel quan és creat indica si s'han de mostrar totes les tasques o només les que hi participa l'usuari.

A la funció *getTasks* se li passa per paràmetre una data inicial i una data final i retorna les tasques que es troben dins d'aquest període. No cal que tot el temps de duració de la tasca estigui entre aquestes dates, és suficient amb que en algun moment la tasca es trobi dins d'aquest període. També se li passa com a paràmetre els filtres d'estat i de calendari i el booleà que indica si s'han de retornar totes les tasques o només les que hi participa l'usuari.

Es demanen les tasques dues vegades al model perquè les dates d'inici i final de la vista de calendari i de la llista no són les mateixes.

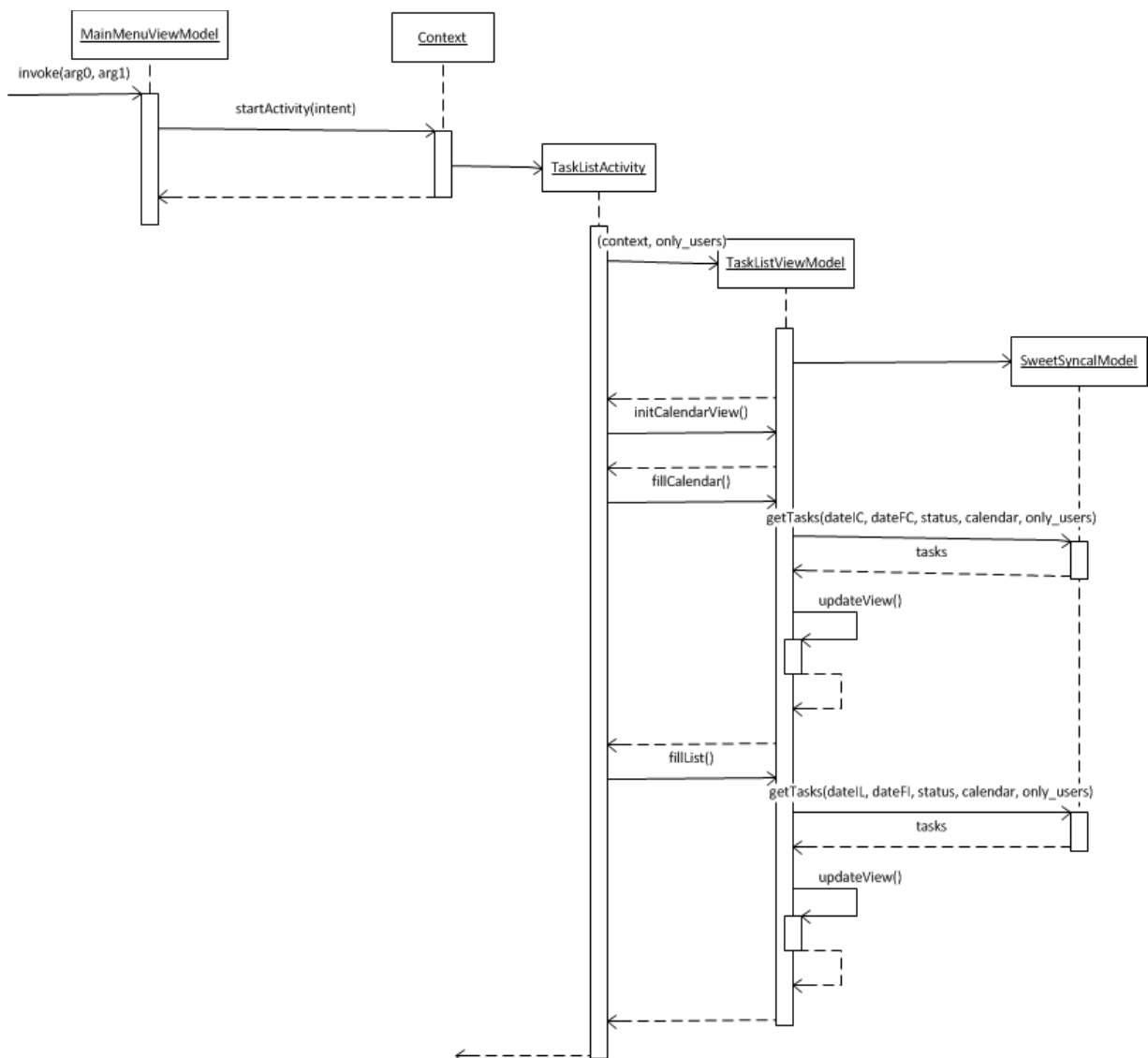


Figura 103: Diagrama seqüència disseny llistarTasques

### Filtrar llistat tasques: operació filtrar(estat, calendari, data)

El filtrat de tasques es determina en una vista nova, però s'aplicarà a la vista de llistat de tasques. Quan s'envien les dades únicament cal retornar-les a la vista anterior, assignant-les com a resultat de l'Activity.

Quan l'Activity de llistar tasques rep el filtre el desa i actualitza les vistes de calendari i llista per tal d'aplicar-hi el filtre.

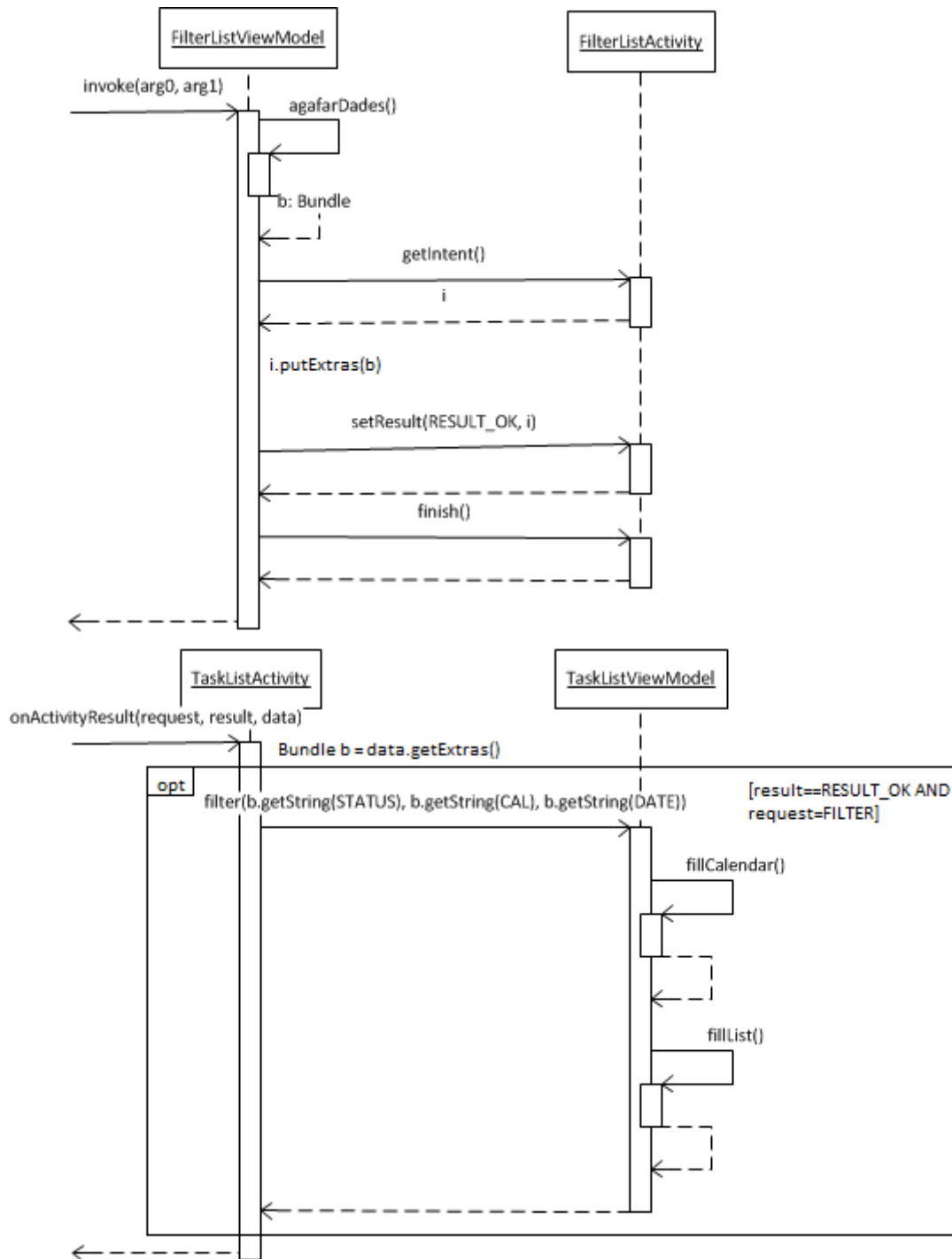


Figura 104: Diagrama seqüència disseny llistar

## Recordar tasca: operació enviarNotificacio(notificacio)

Tot i que l'operació *enviarNotificacio* correspondria únicament a l'última operació del diagrama, *notify*, s'ha cregut convenient il·lustrar tot el procés que s'executa quan s'ha de recordar una tasca.

L'alarma de recordatori rep les dades de la tasca que s'ha de recordar i les utilitza per crear la notificació corresponent.

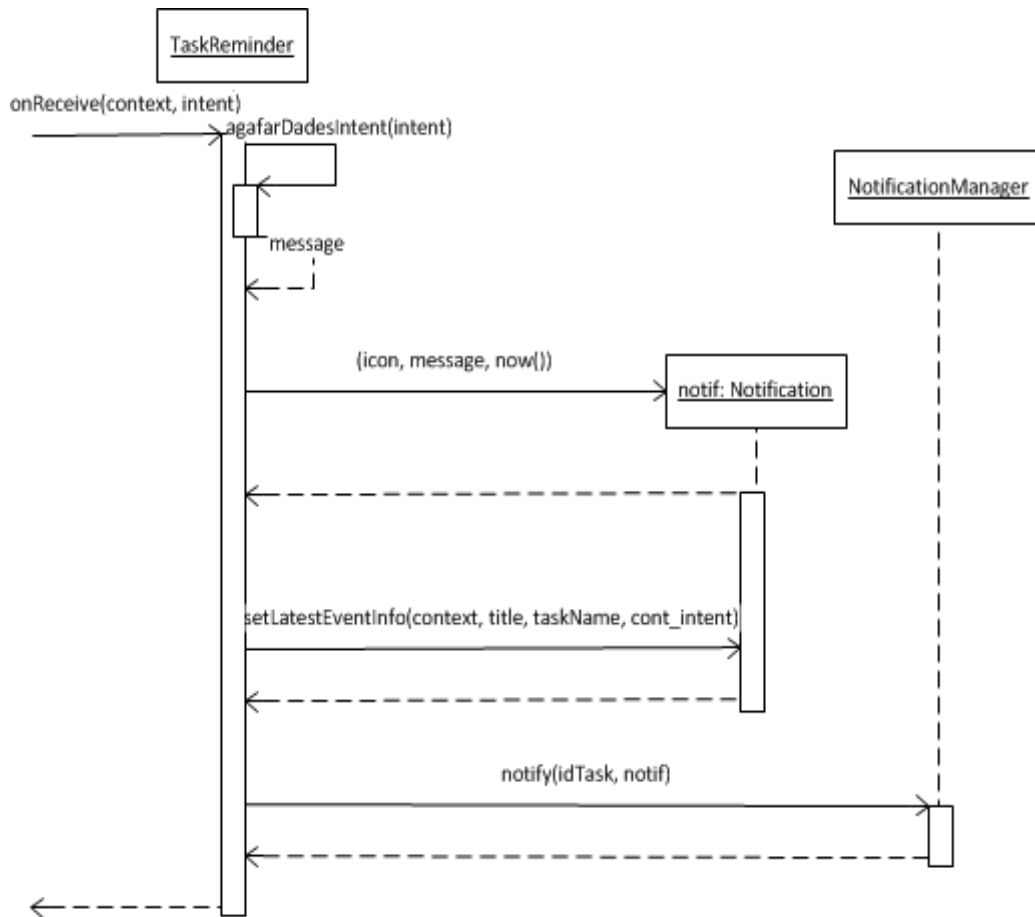


Figura 105: Diagrama seqüència disseny enviarNotificacio

## 2.4 Sincronització

Degut a la gran similitud entre els diagrames del cas d'ús *Sincronitzar calendaris* i els del cas d'ús *Sincronització total* (un dels casos d'ús resultants de la correcció del primer), els diagrames del cas d'ús anterior a la correcció no es mostraran. A més, com que el cas d'ús *Sincronització total* inclou les operacions dels casos d'ús *Enviar tasques* i *Demandar canvis*, els diagrames d'aquests dos casos d'ús tampoc es mostraran ja que seria redundants. Només cal remarcar que, en el cas d'ús *Demandar tasques*, cal recuperar i actualitzar la data de l'última sincronització, mentre que en la part corresponent del cas d'ús *Sincronització total* això no es fa.

## Sincronització total: operació sincronizacioCompleta()

El procés de sincronització total consisteix en demanar dades del servidor i enviar les dades modificades a nivell local. Primer de tot es demanaran els canvis fets al servidor (*getInfo*) i es desaran els resultats a nivell local amb l'operació *syncData*.

A continuació, si no hi ha hagut cap error es recuperaran els canvis fets a nivell local i, si n'hi ha algun, s'enviaran al servidor amb l'operació *sendData*. Aquesta operació retornarà les tasques i relacions que han estat tractades per tal que es puguin desmarcar com a modificades a nivell local. També retorna el llistat de pending approvals creats o modificats.

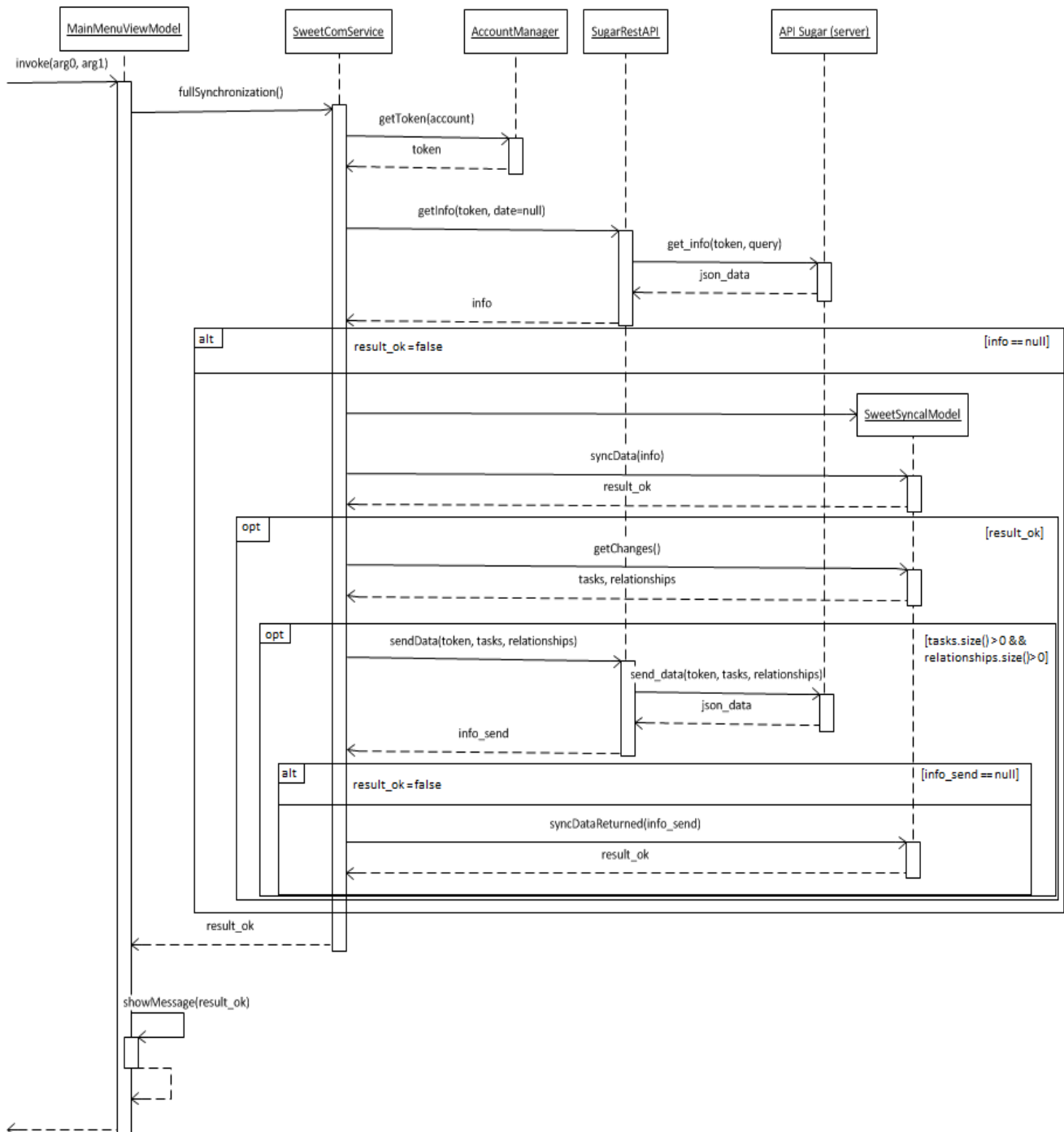


Figura 106: Diagrama seqüència disseny sincronizacioCompleta

## Sugar API: operació get\_info

L'operació `get_info` en general no és difícil d'entendre. Es tracta d'anar recuperant la informació que ha de rebre l'usuari: grups, calendaris, tasques, relacions, etc. El paràmetre `query` indica si és sincronització completa o incremental.

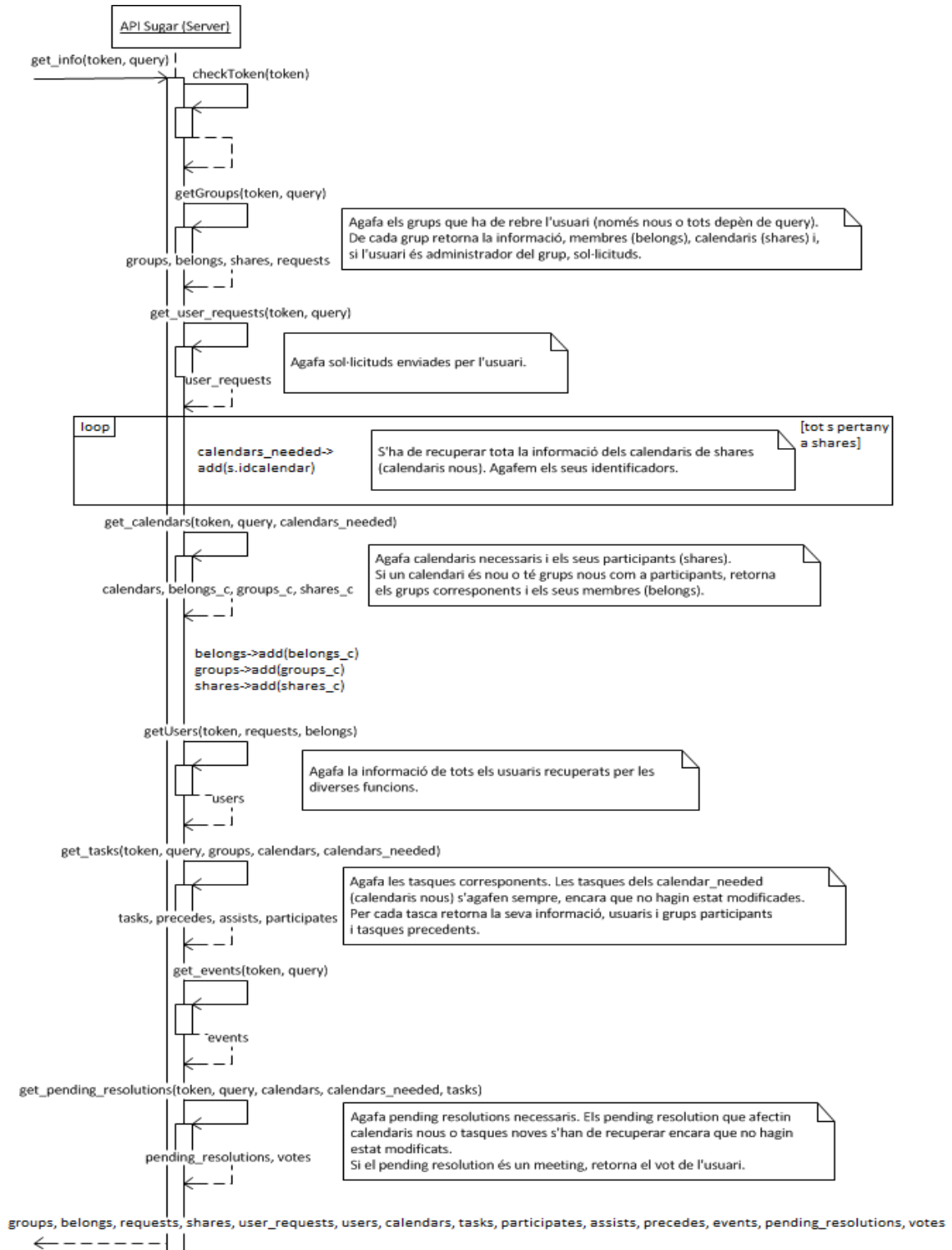


Figura 107: Diagrama seqüència disseny `get_info`

## Sugar API: operació send\_data

La funció `send_data(token, tasks, relationships)` del diagrama de `sincronitzacioCompleta()` únicament crida les funcions `set_tasks(token, tasks)` i `set_relationships(token, relationships)`. Degut a la seva senzillesa no es mostrarà el seu diagrama, però si el de les dues funcions esmentades.

La funció `set_tasks` s'encarrega de tractar les tasques. Els canvis de les tasques que no tenen més participants es faran tots al final perquè l'operació sigui més ràpida. Per exemple, totes les tasques que s'han de crear es desaran a `new_tasks` i al final es crearan totes de cop.

Si l'identificador de la tasca és nul vol dir que la tasca no existeix i s'ha de crear. Si la tasca existeix i no té més participants la desem al grup corresponent. Si té més participants, però, caldrà tractar els pending approvals. La funció `treat_task` es pot veure a continuació de al figura següent. Les tasques tractades es desaran per a ser retornades a l'usuari i que aquest sàpiga que han estat tractades.

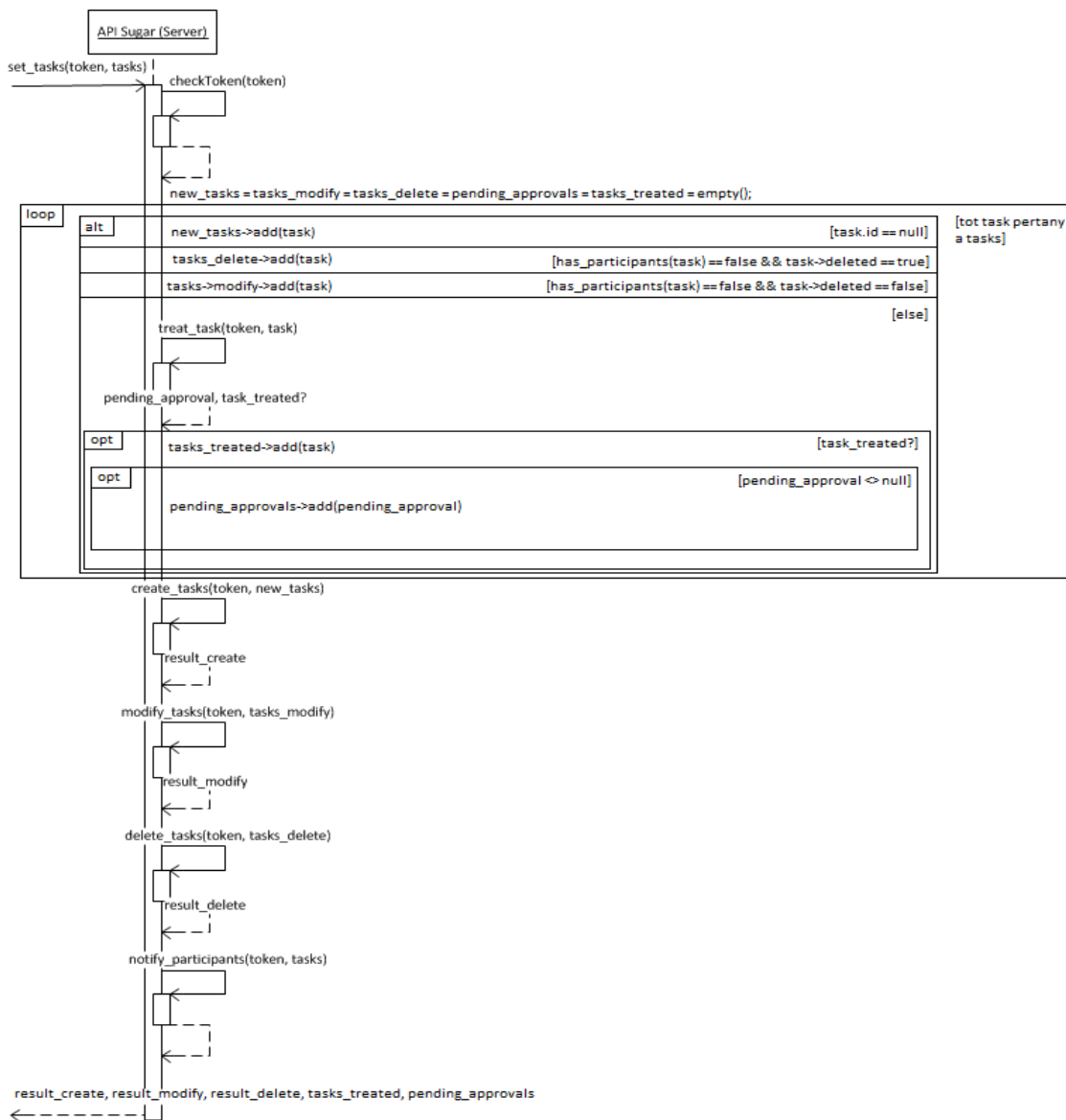


Figura 108: Diagrama seqüència disseny set\_tasks



A la funció *treat\_task* cal comprovar que no hi hagi ningú modificant la tasca. Si la tasca està sent modificada s'haurà de tractar en la següent sincronització.

Si la tasca no està sent modificada procedim a revisar els pending approvals i meetings existents. Si hi ha algun conflicte que no es pot solucionar en aquests moments la tasca no es tractarà. Si hi ha un pending approval que pot ser modificat (creat pel mateix usuari), es modificarà més endavant.

Si es troba un pending approval **no respost** que afecta algun dels camps enviats per l'usuari, s'elimina el camp corresponent del pending approval (el pending approval s'elimina si queda buit) i es crea un meeting amb els possibles valors que poden prendre cadascun dels camps en conflicte. Si, en canvi, el camp es troba en un meeting, s'afegeix el valor al llistat de valors del meeting. Òbviament, tot això es farà només si el valor present al pending approval o meeting és diferent al valor enviat per l'usuari, en cas contrari es considerarà que el camp ha estat tractat.

La variable *data\_left* contindrà els camps no tractats. Si no hi ha cap conflicte es modificarà o crearà un pending approval amb els camps no tractats (*data\_left*).

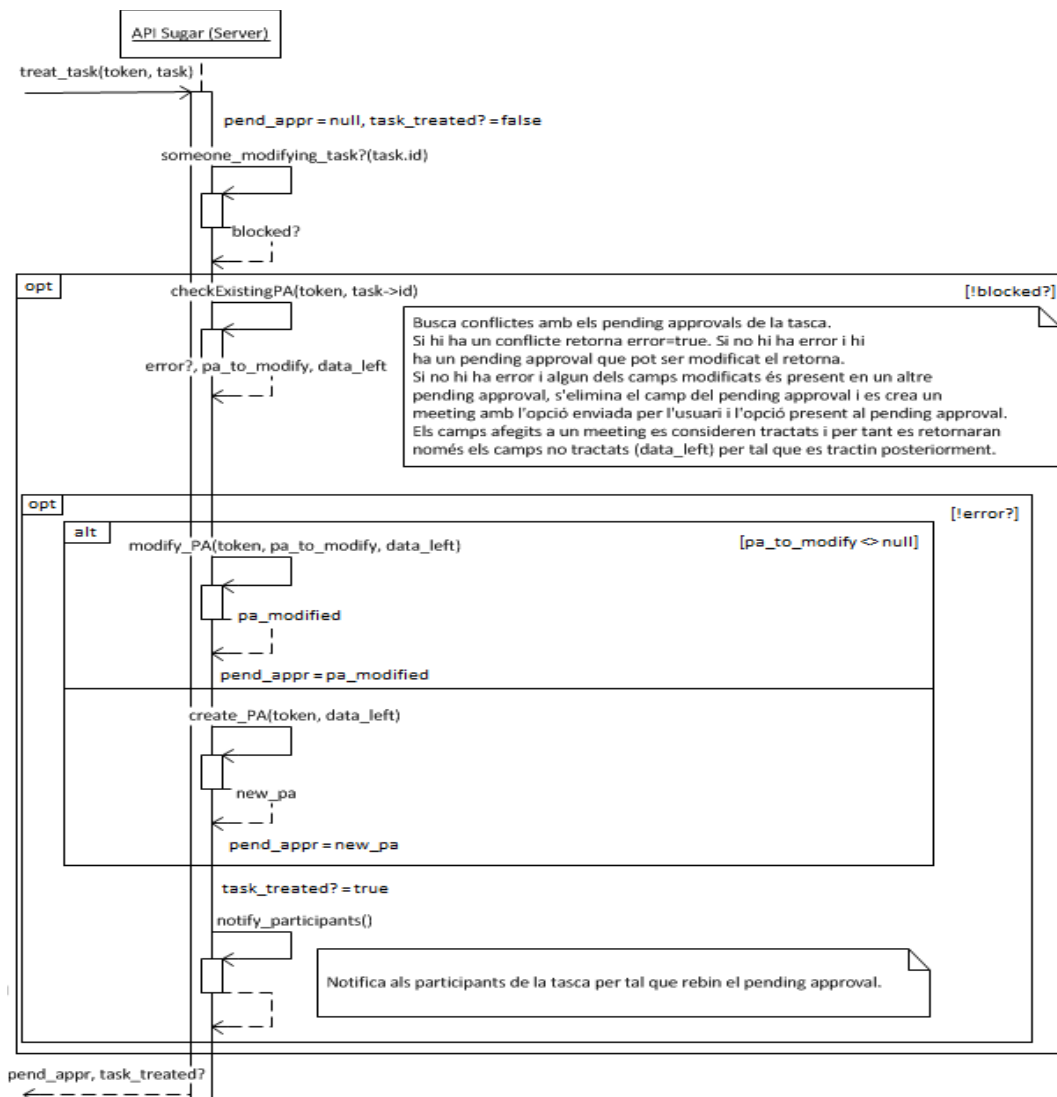


Figura 109: Diagrama seqüència disseny *treat\_task*

Amb això podem donar per finalitzada la funció *set\_tasks*. Procedim ara a dissenyar la funció *set\_relationships*.

El paràmetre *relationships* de la funció *set\_relationships*, el diagrama de la qual es mostra a continuació, conté les relacions enviades al servidor. Aquest paràmetre és un array que contindrà diverses relacions en cada posició (array d'arrays). Per evitar el tractament de tasques repetides, les relacions del mateix tipus que afectin a una mateixa tasca s'agruparan en una sola posició. D'aquesta manera, en cada posició hi haurà l'identificador de la tasca a la que afecta la relació (la tasca posterior en el cas de relació de precedència), el tipus de la relació (assisteix, participa o precedeix) i un array amb els identificadors relacionats (identificadors d'usuaris, grups o tasques).

Per exemple, si afegim *usuari1* i *usuari2* com a participants de *tasca1*, tindrem una sola posició a *relationships* amb l'identificador de *tasca1*, tipus "participa" i com a relacions un array amb els identificadors d'*usuari1* i *usuari2*.

Les modificacions fetes a tasques que no tinguin més participants es faran totes al final per tal de millorar l'eficiència temporal de la funció. Així doncs, totes les relacions que s'hagin d'afegir o eliminar que afectin una tasca sense més participants es desaran a la variable *rels\_add* o *rels\_delete* segons correspongui i al final de tot s'afegiran o eliminaran.

El contingut de *relationships* es tractarà de forma iterativa. Per cada posició primer de tot comprovarem si la tasca afectada té més participants o no. Si no té més participants afegim les relacions a la variable corresponent en funció de si s'ha d'afegir la relació o eliminar-la. Si té més participants, en canvi, caldrà tractar els pending approvals. La funció *treat\_relationships* retorna els pending approvals creats o modificats i la llista de relacions tractades. Aquestes relacions tractades s'hauran de retornar a l'aplicació local per tal que puguin ser marcades com a tractades.

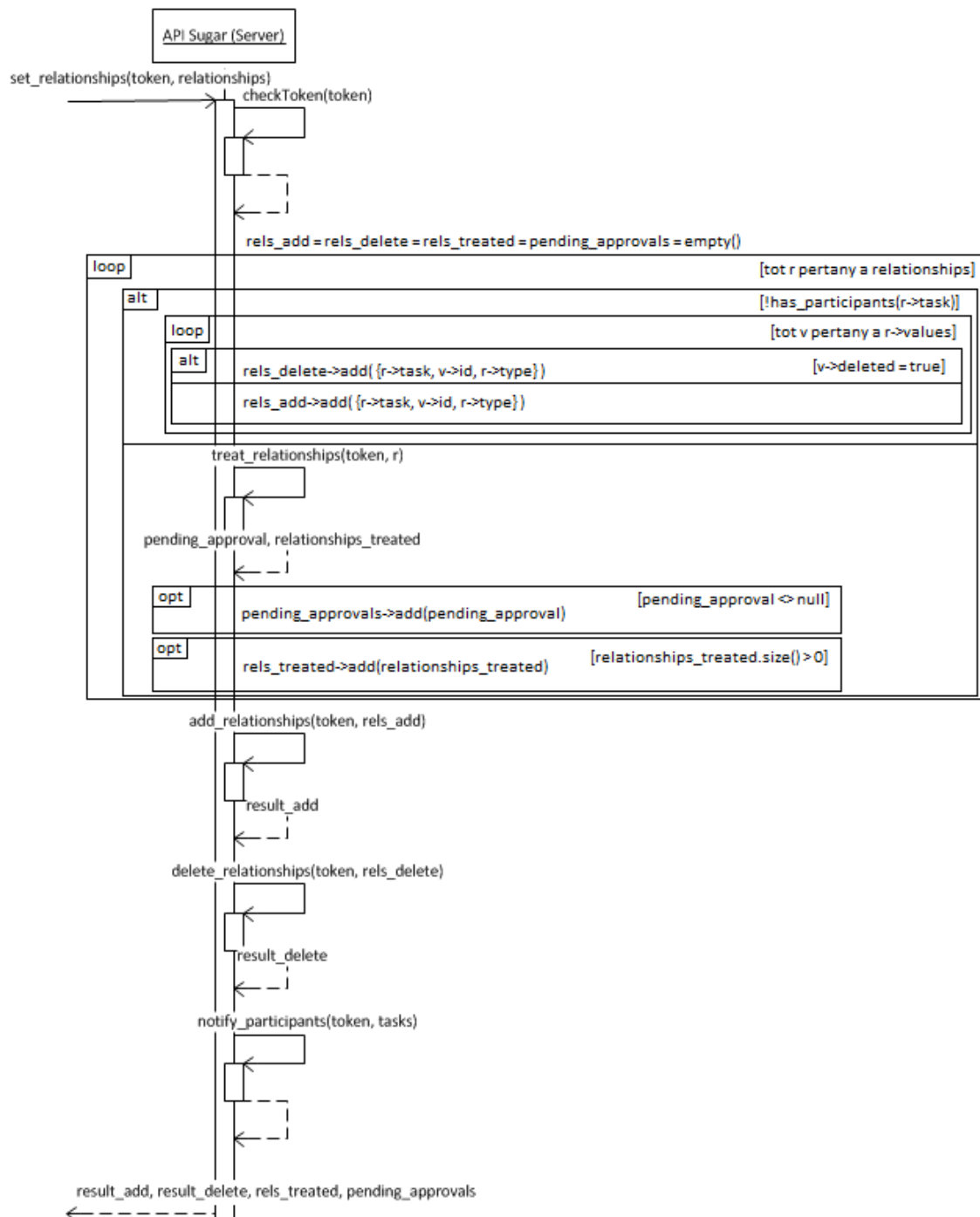


Figura 110: Diagrama seqüència disseny set\_relationships

La funció *treat\_relationship(token, relationship)* s'encarrega de tractar les relacions de les tasques que tenen més participants, creant els *pending approvals* que siguin necessaris. El diagrama es troba a continuació.

Primer de tot comprovem si la tasca afectada està sent modificada. Si no ho està, procedim a revisar els pending approvals existents. Si es troba un pending approval que ja conté alguna de les relacions enviades amb el mateix "valor" (afegir o eliminar) es considerarà la relació com a tractada, com si s'hagués creat o modificat un pending approval per la relació. Si hi ha un pending approval que pot ser modificat (creat pel mateix usuari), es modificarà més endavant.

La variable *rels\_left* conté les relacions que no han estat tractades. A continuació es crearà o modificarà un pending approval per tractar aquestes relacions.

La funció retorna el pending approval creat o modificat i les relacions que han estat tractades sense necessitat de crear o modificar un pending approval.

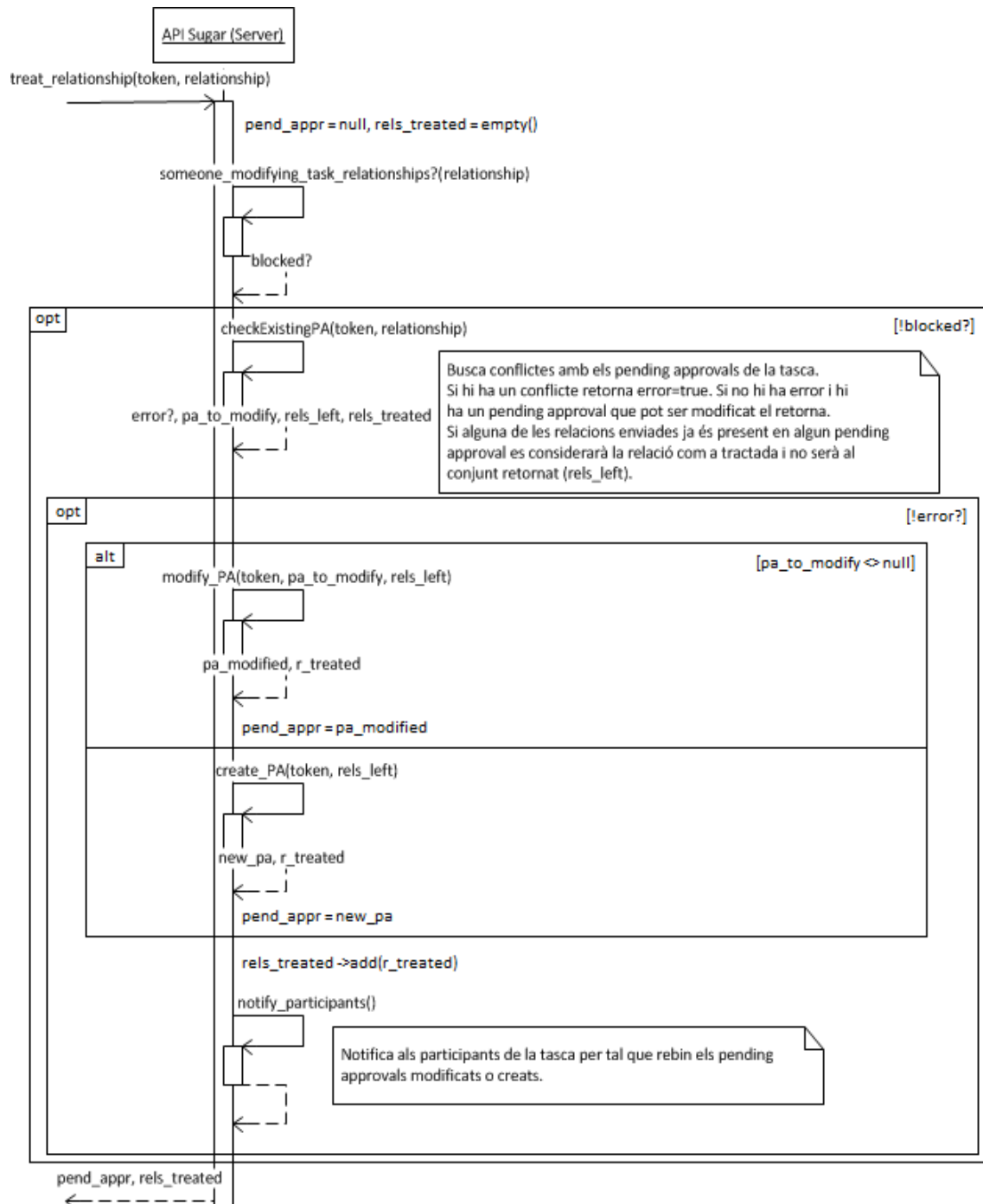


Figura 111: Diagrama seqüència disseny *treat\_relationship*

## 2.5 Gestió d'esdeveniments, votacions i preferències

### Llistar esdeveniments: operació llistarEsdeveniments()

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

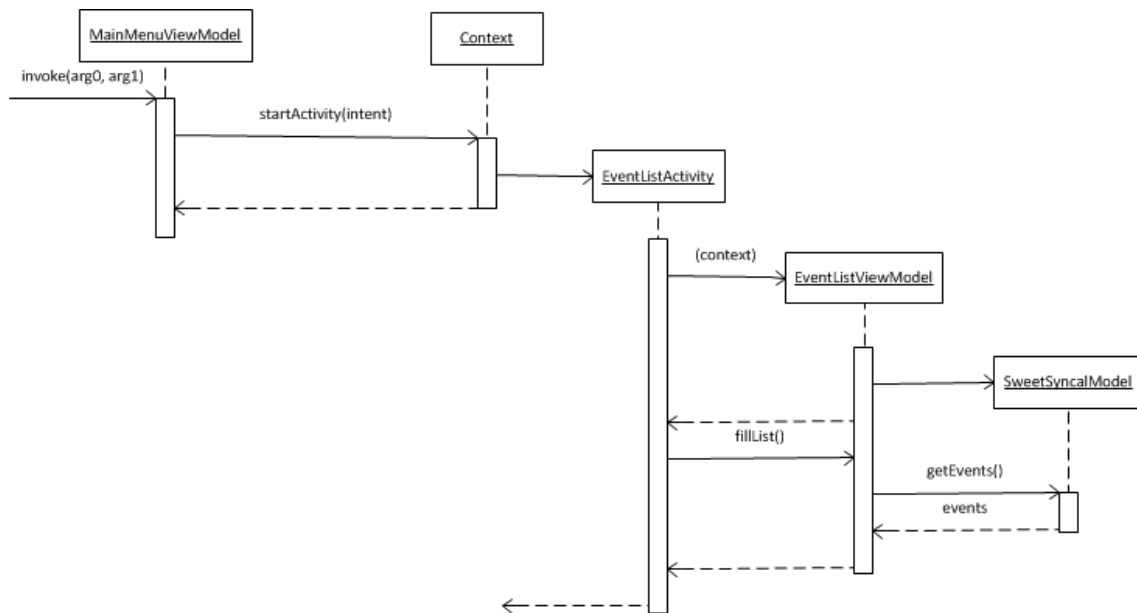


Figura 112: Diagrama seqüència disseny llistarEsdeveniments

### Llistar meetings: operació llistarMeetings()

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

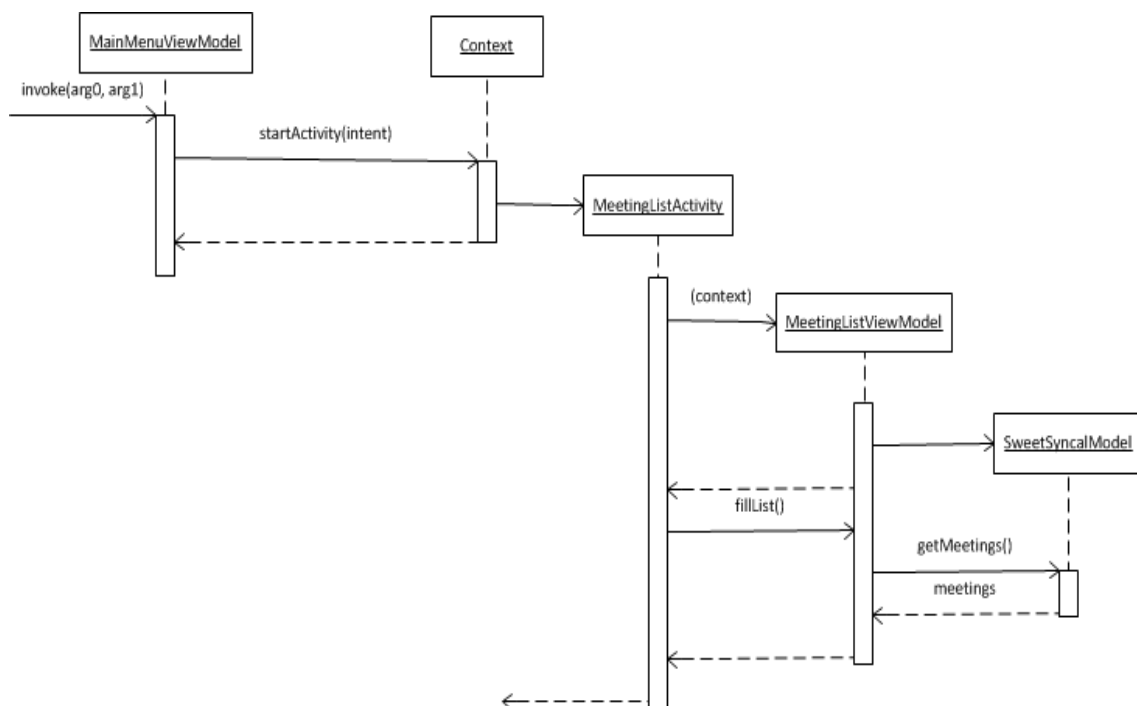


Figura 113: Diagrama seqüència disseny llistarMeetings

## Veure informació meeting: operació veureInfoMeeting(id)

Operació de veure informació. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

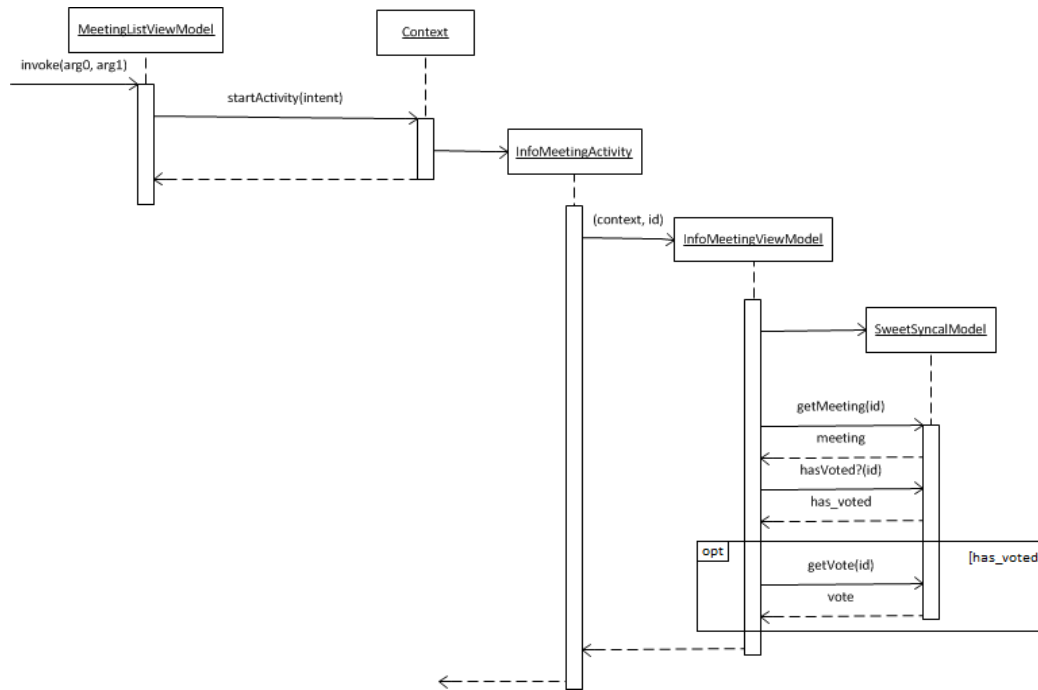


Figura 114: Diagrama seqüència disseny veureInfoMeeting

## Votar en un meeting: operació confirmarVot()

Quan l'usuari envia les dades per votar en un meeting primer de tot comprovem que les dades enviades són correctes i, si ho són, les enviem al servidor per tal que aquest desi el vot. La funció *vote\_meeting* es pot veure a continuació de la següent figura.

Si el vot es desa correctament al servidor (`result_code == "1"`), el desem també a nivell local i actualitzem la vista.

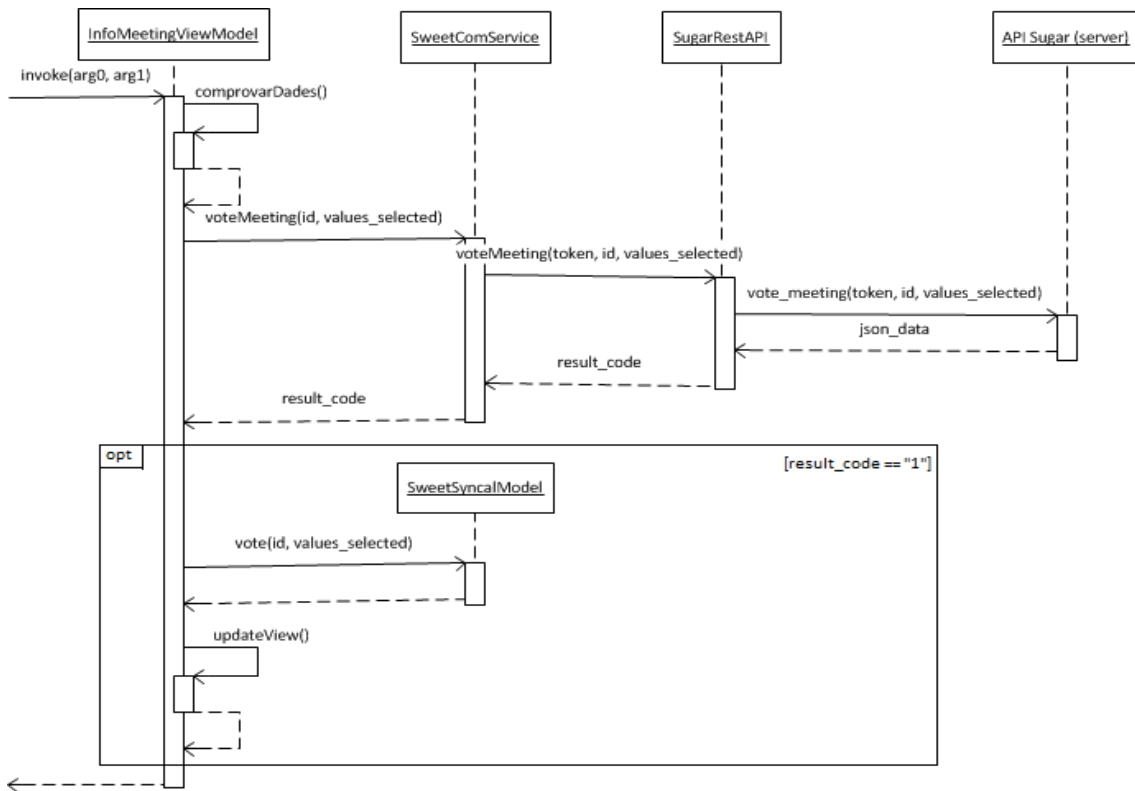


Figura 115: Diagrama seqüència disseny confirmarVot

### Sugar API: operació `vote_meeting`

La funció `vote_meeting` primer de tot recupera la informació del meeting per comprovar que estigui actiu. Si no ho està s'indicarà a l'usuari que ja no es pot votar el meeting.

Si el meeting està actiu comprovem si l'usuari ha votat ja per evitar tenir vots duplicats. Si l'usuari ja ha votat li comunicarem l'error. Si no, desem el vot.

Un cop hem desat el vot de l'usuari cal comprovar si el meeting ha finalitzat. Per saber-ho hem d'agafar tots els vots del meeting i els usuaris que poden votar-hi. Si la quantitat de vots no arriba a la meitat d'usuaris participants és impossible que el meeting hagi acabat.

La funció `isVotingOver?` s'encarrega de comprovar si el meeting ha acabat. Un meeting ha acabat si alguna opció té suficients vots per ser considerada guanyadora o si tots els usuaris han votat. Aquesta funció retorna dos booleans que indiquen si el meeting ha acabat i si hi ha hagut un empat.

Si hi ha un empat s'eliminen tots els vots i es repeteix el meeting. Si el meeting s'ha acabat i no hi ha cap empat es modificarà la tasca amb els valors que hagin guanyat el meeting i es marcarà el meeting com a inactiu. A més, si es dona algun d'aquests dos casos es crearà un esdeveniment per notificar als usuaris que participin al meeting.

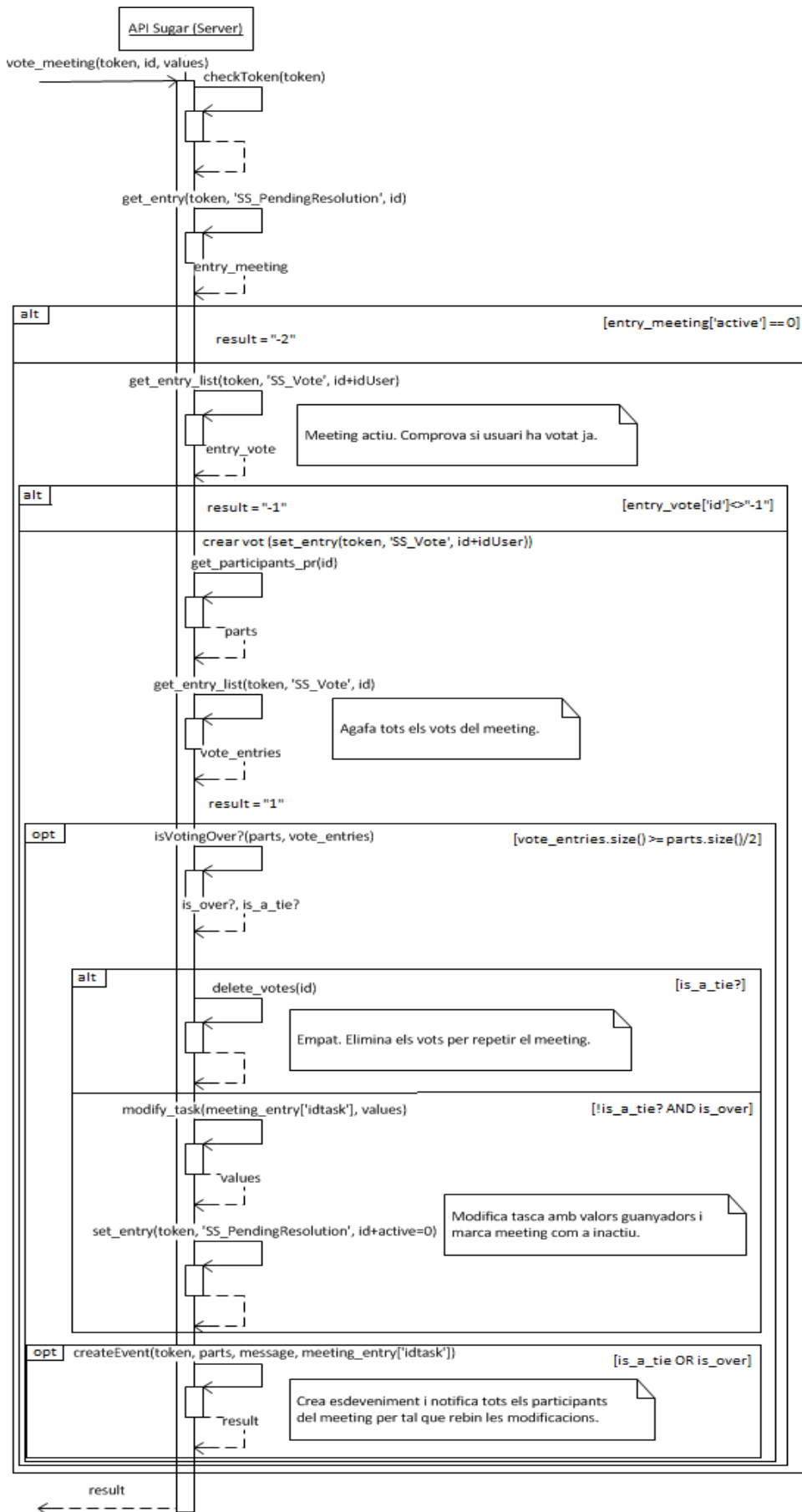


Figura 116: Diagrama seqüència disseny vote\_meeting



## Eliminar vot d'un meeting: operació confirmarEliminarVot()

Per eliminar un vot d'un meeting primer de tot cal comprovar si el meeting està actiu. Si ho està, recuperem el vot de l'usuari i l'eliminem. Si l'usuari no ha votat al meeting es considerarà que el vot ja ha estat eliminat.

Si el vot s'ha eliminat correctament del servidor l'eliminem també a nivell local i actualitzem la vista.

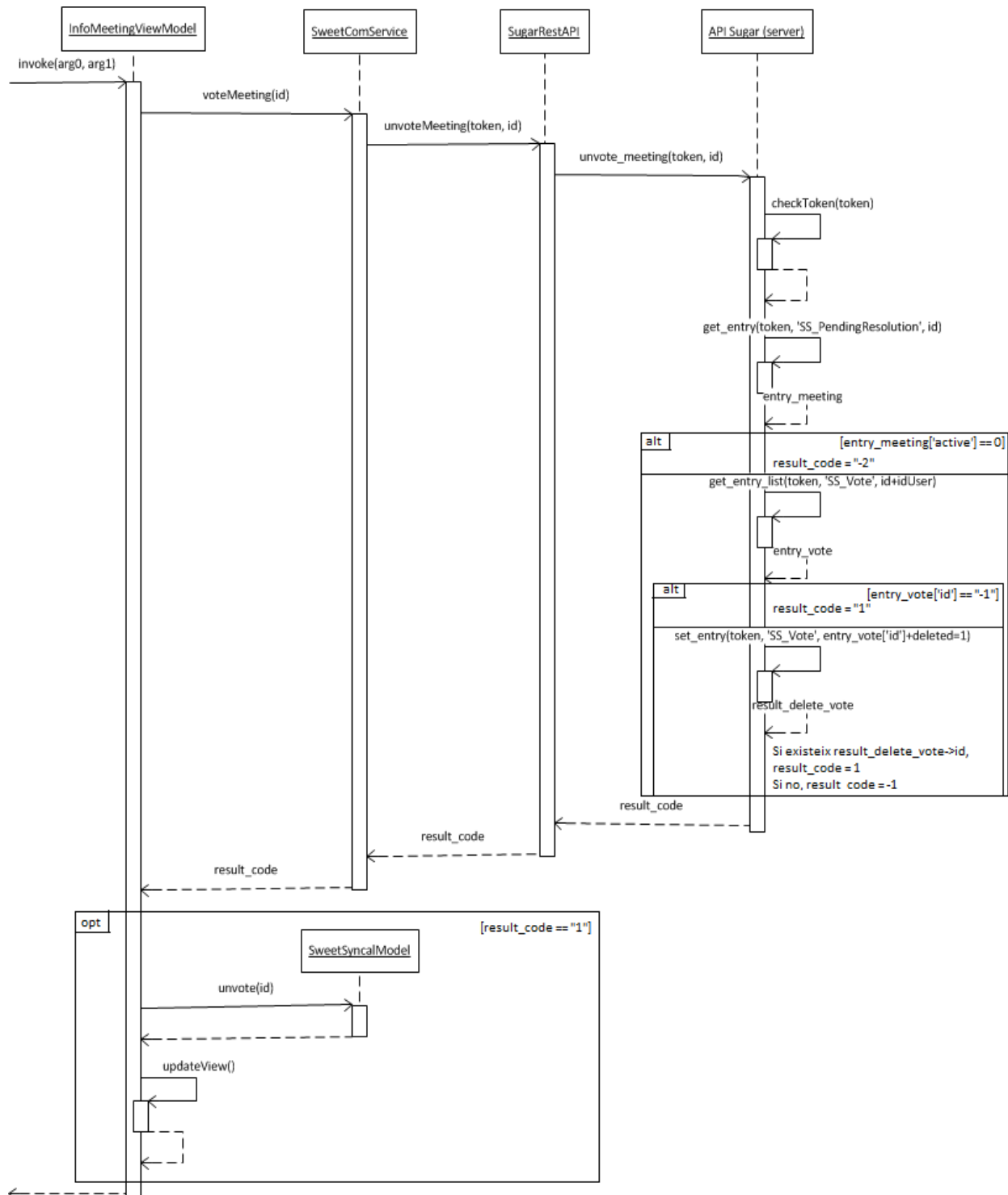


Figura 117: Diagrama seqüència disseny confirmarEliminarVot

## Llistar pending approvals: operació llistarPendingApprovals()

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

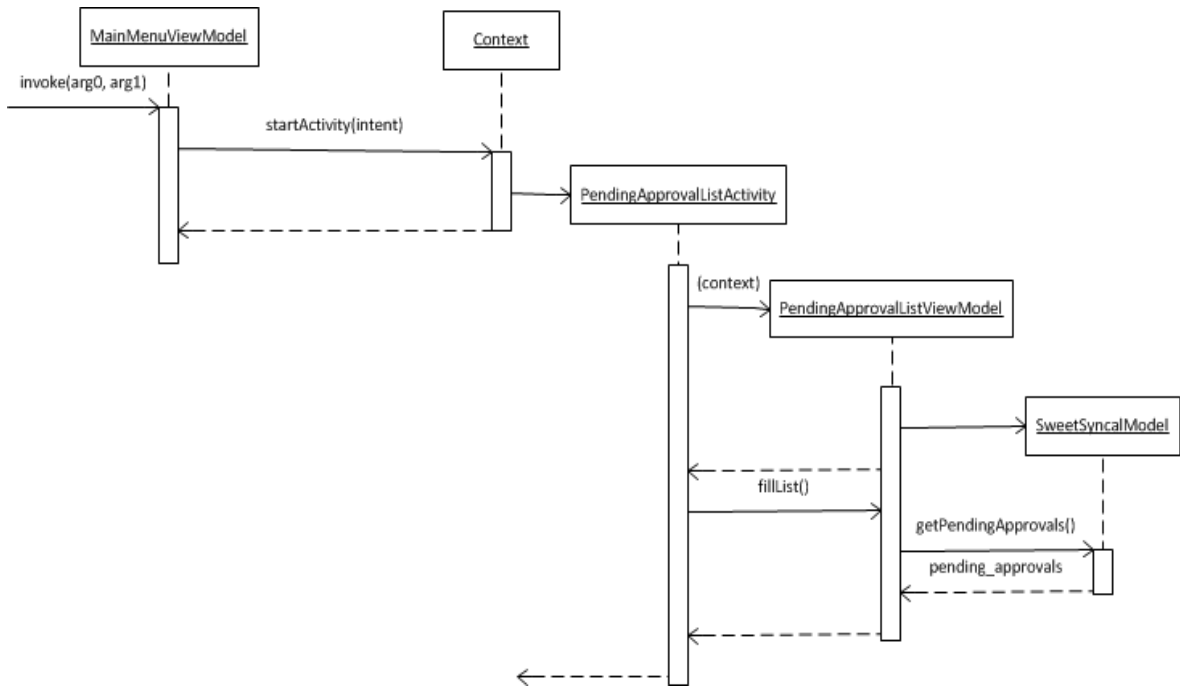


Figura 118: Diagrama seqüència disseny llistarPendingApprovals

## Veure info pending approval: operació veureInfoPendingApproval(id)

Operació de veure informació. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades.

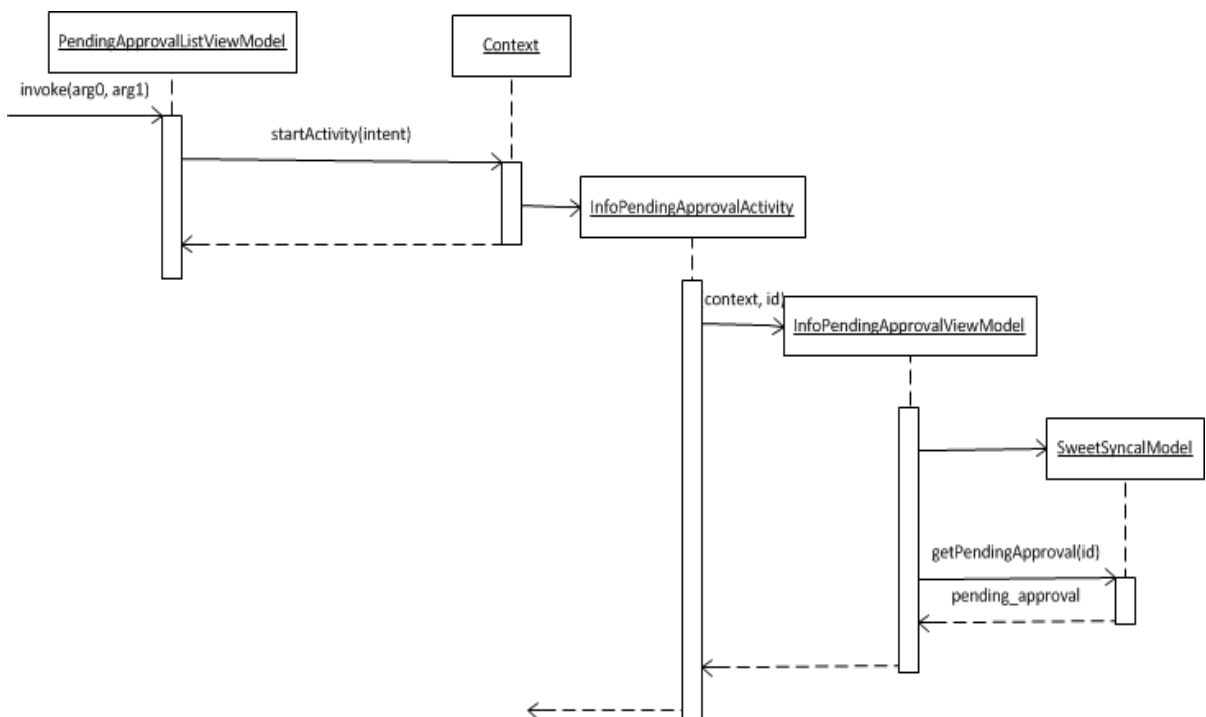


Figura 119: Diagrama seqüència disseny veureInfoPendingApproval

## Acceptar/Rebutjar pending approval: operació confirmarResposta()

La part important de l'operació es troba a l'operació *accept\_pa*, la resta és bastant trivial. Es fa la petició d'acceptar o rebutjar un pending approval al servidor i si no hi ha cap error fem les modificacions a nivell local.

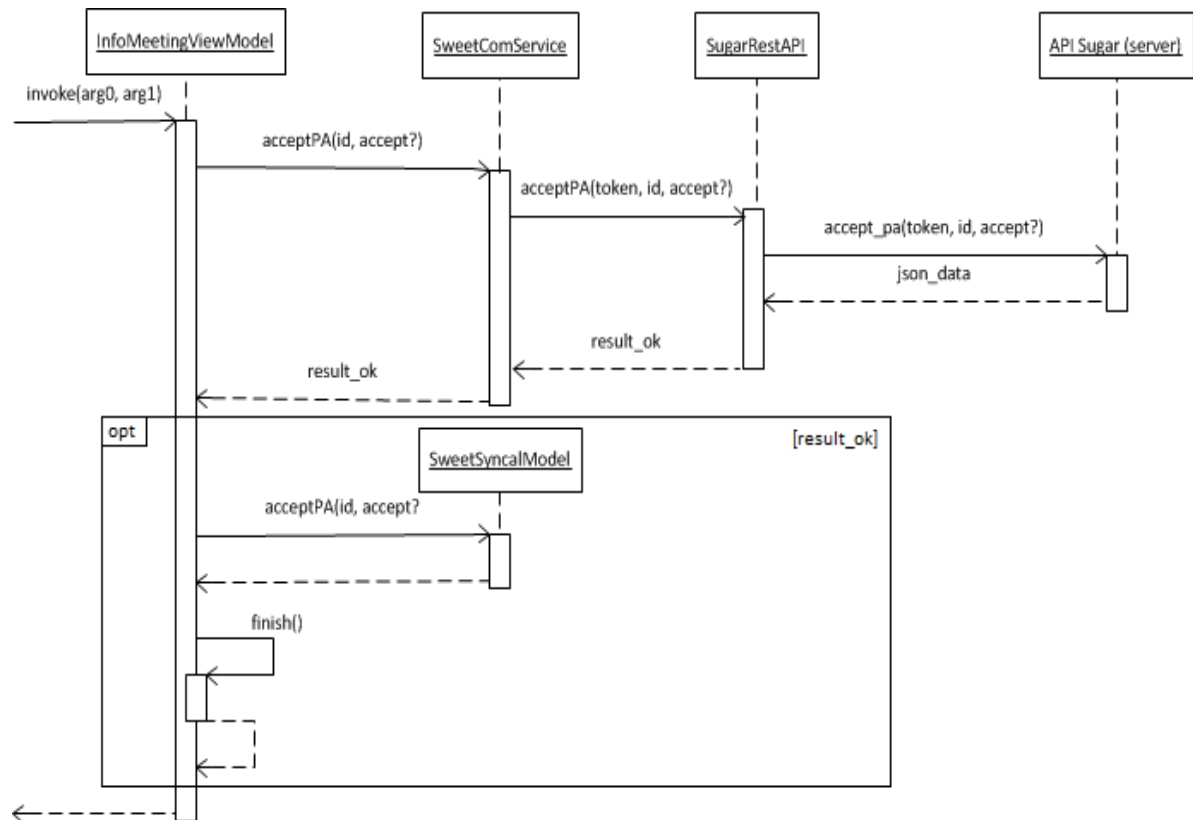


Figura 120: Diagrama seqüència disseny confirmarResposta

## Sugar API: operació accept\_pa

Primer de tot cal comprovar si l'usuari ja ha respost el pending approval. Si no ho ha fet, desm la resposta i, a continuació, caldrà comprovar si el pending approval ha finalitzat. Per fer-ho cal recuperar els participants del pending approval i totes les seves respostes. Si la quantitat de respostes d'acceptar és superior a la meitat dels participants el pending approval finalitza com a acceptat. Si la quantitat de respostes de rebutjar és superior o igual a la meitat dels participants el pending approval finalitza com a rebutjat.

Si el pending approval es finalitza com a rebutjat només cal fer un *touch* (actualitzar data de modificació) de la tasca afectada (si el pending approval afectava una tasca) i finalitzar el pending approval. És necessari actualitzar la data per tal que l'usuari que ha creat el pending approval rebi la tasca amb les dades correctes, ja que ell té la tasca modificada. Si finalitza com a acceptat, en canvi, caldrà realitzar els canvis que conté el pending approval: modificar o eliminar tasca/calendari o afegir/eliminar participants o precedents.

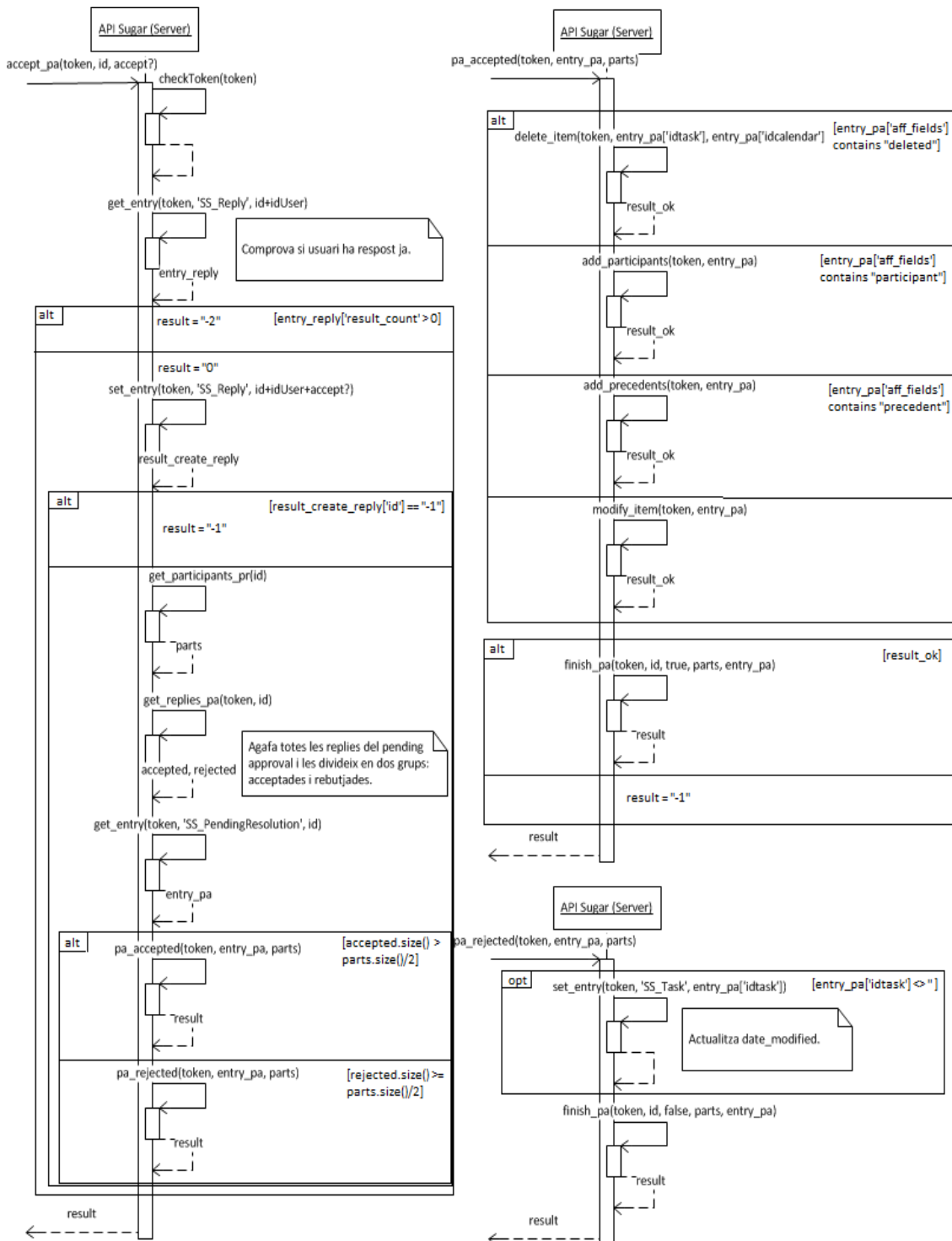


Figura 121: Diagrama seqüència disseny `accept_pa`, `pa_accepted` i `pa_rejected`

Per finalitzar el pending approval (operació `finish_pa`, el diagrama es troba a continuació) només cal eliminar-lo a ell i totes les seves respostes i crear un esdeveniment per notificar als usuaris participants.

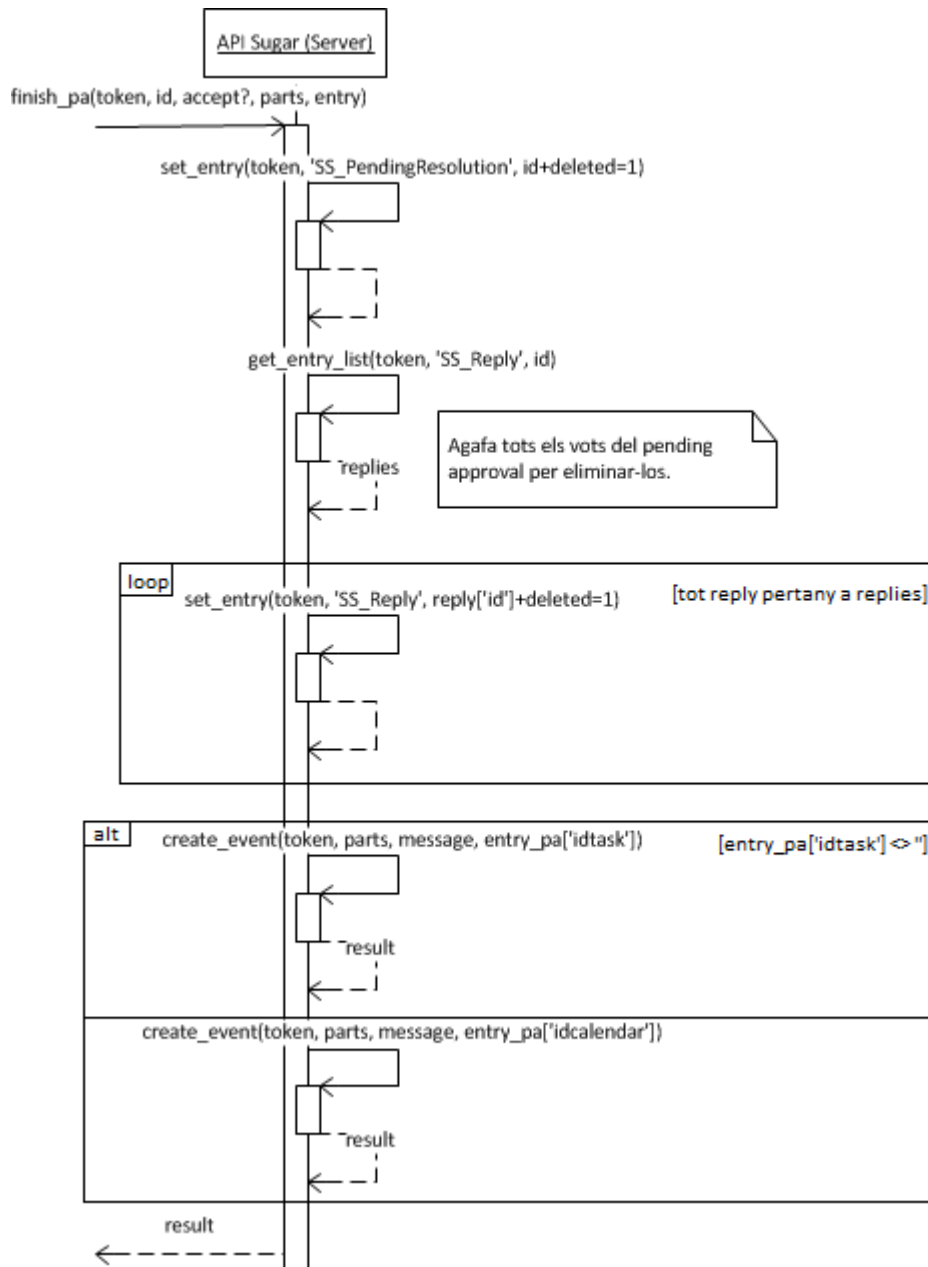


Figura 122: Diagrama seqüència disseny `finish_pa`

### Veure preferències: operació `veurePreferencies()`

Operació de veure informació. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la classe `SharedPreferences`.

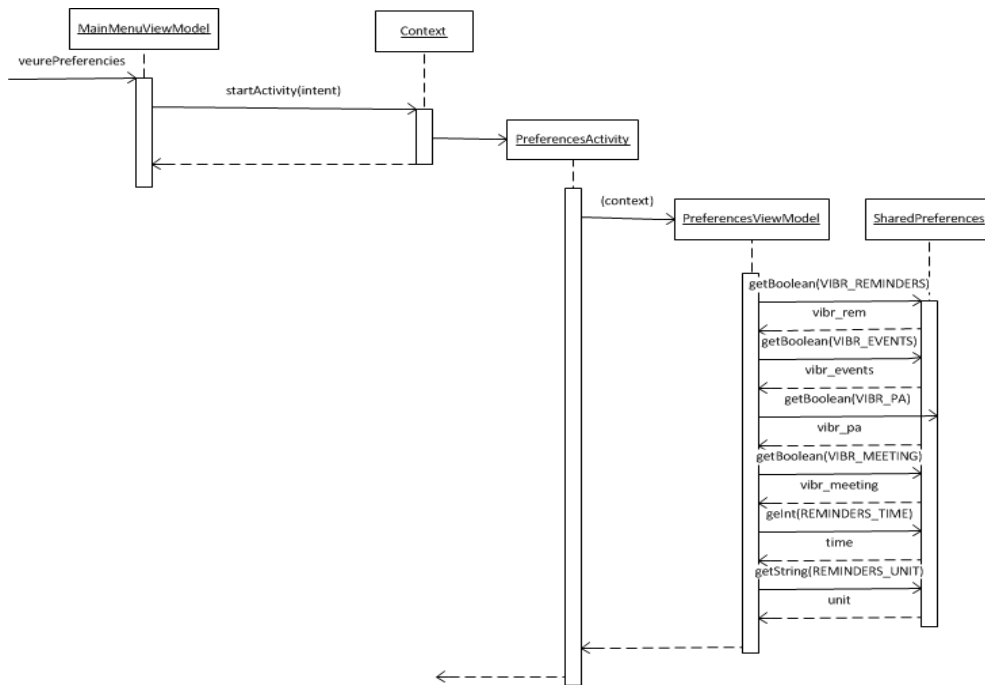


Figura 123: Diagrama seqüència disseny veurePreferencies

### Modificar preferències: operació enviarPreferencies(vibrR, vibrE, vibrM, vibrPA, tempsR)

Per modificar les preferències només cal introduir les dades a l'editor de *SharedPreferences* i fer un commit.

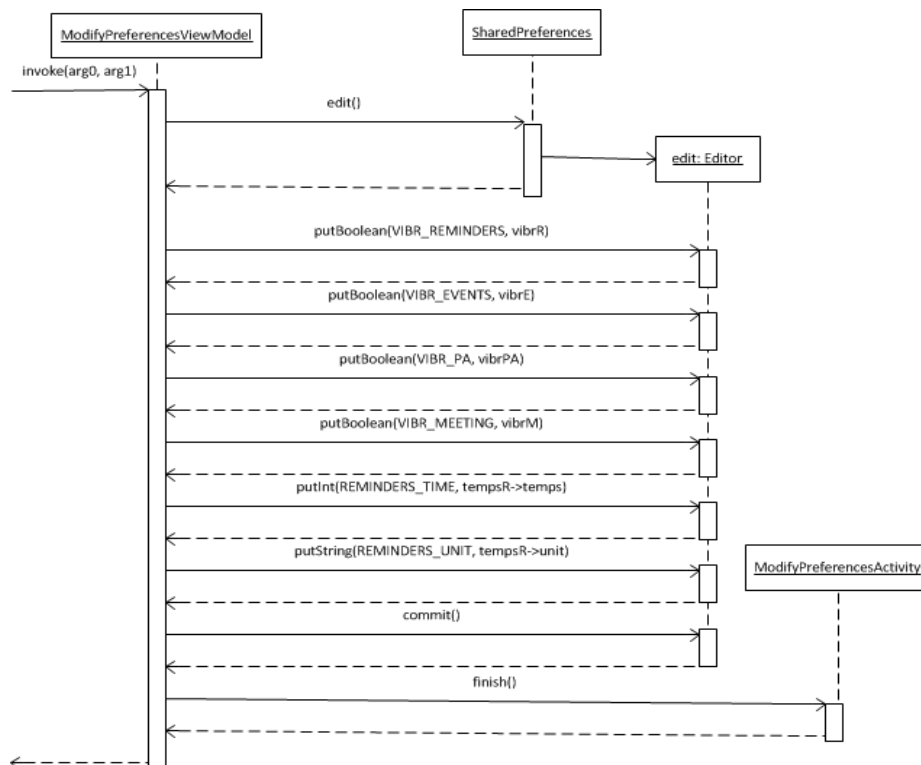


Figura 124: Diagrama seqüència disseny enviarPreferencies

## 2.6 Gestió de grups

### Crear grup: operació crearGrup(nom, usuaris)

L'operació de crear grup és bastant semblant a la de crear calendari de l'apartat 2.2. Es comproven les dades i s'intenta crear el grup al servidor (cal comprovar que no existeixi un grup amb el mateix nom). Si el grup es crea correctament intentem afegir-hi els participants. Si s'afegeixen correctament, els afegim també a nivell local. Si no, fem rollback i eliminem el grup que acabem de crear.

La funció *add\_members* de l'API de Sugar del servidor és senzilla: afegeix els usuaris enviats com a membres del grup (si no ho són ja) i crea esdeveniments per notificar els usuaris afegits i els actuals membres del grup.

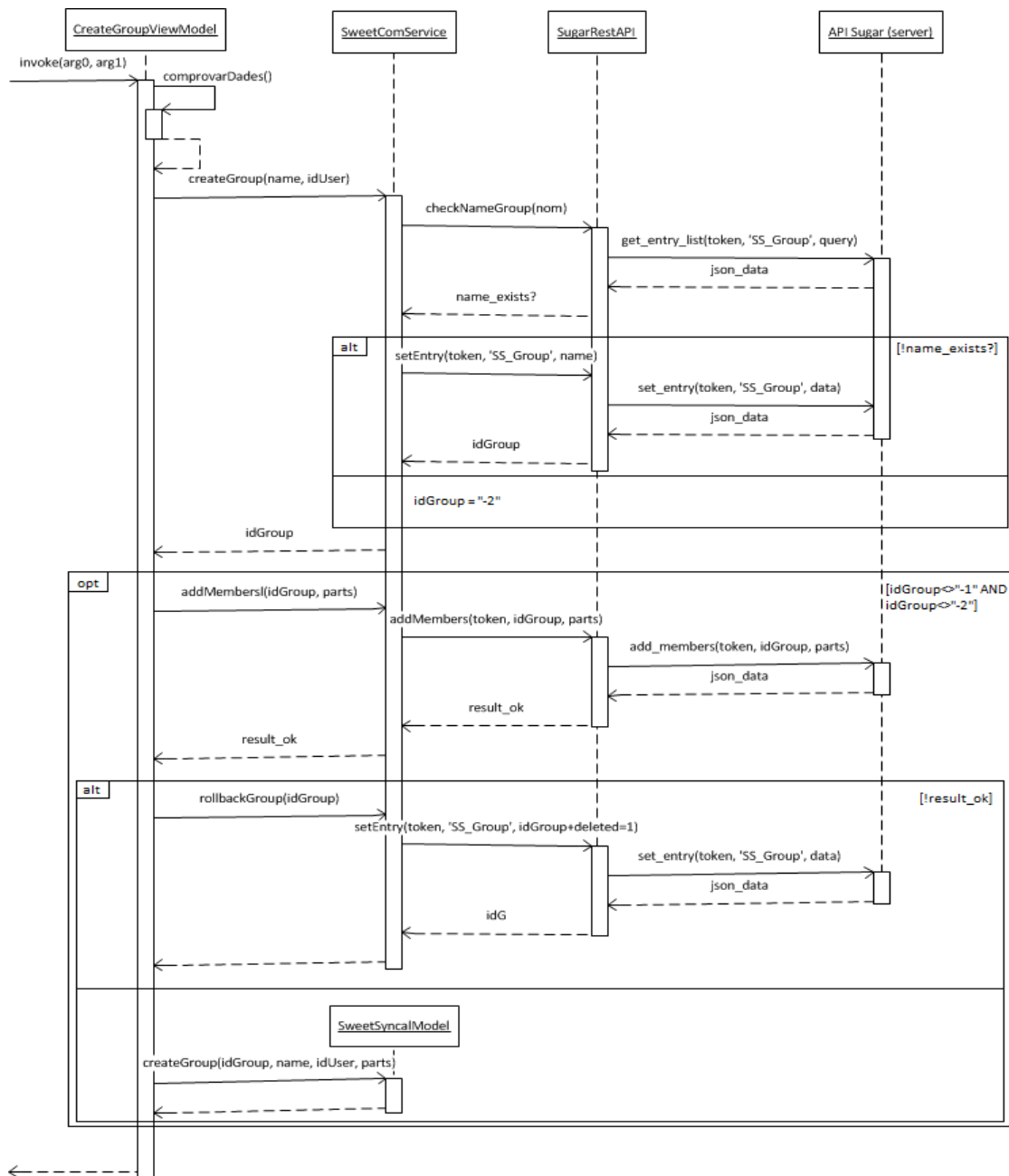


Figura 125: Diagrama seqüència disseny crearGrup

## Modificar nom grup: operació canviarNom(nom)

Per canviar el nom del grup comprovem si ja hi ha algun grup amb el mateix nom i, si no és així, el modifiquem. A continuació hem de crear un esdeveniment per notificar als membres del grup ja que la modificació del nom es fa amb la funció *set\_entry*, que és una operació pròpia de Sugar i que, per tant, no crea cap esdeveniment. Si no hi ha cap error, modifiquem el nom a nivell local.

La funció *set\_event* de l'API de Sugar del servidor únicament crea els esdeveniments necessaris en funció de l'operació realitzada (en aquest cas, operació 0) per tal de notificar els usuaris corresponents.

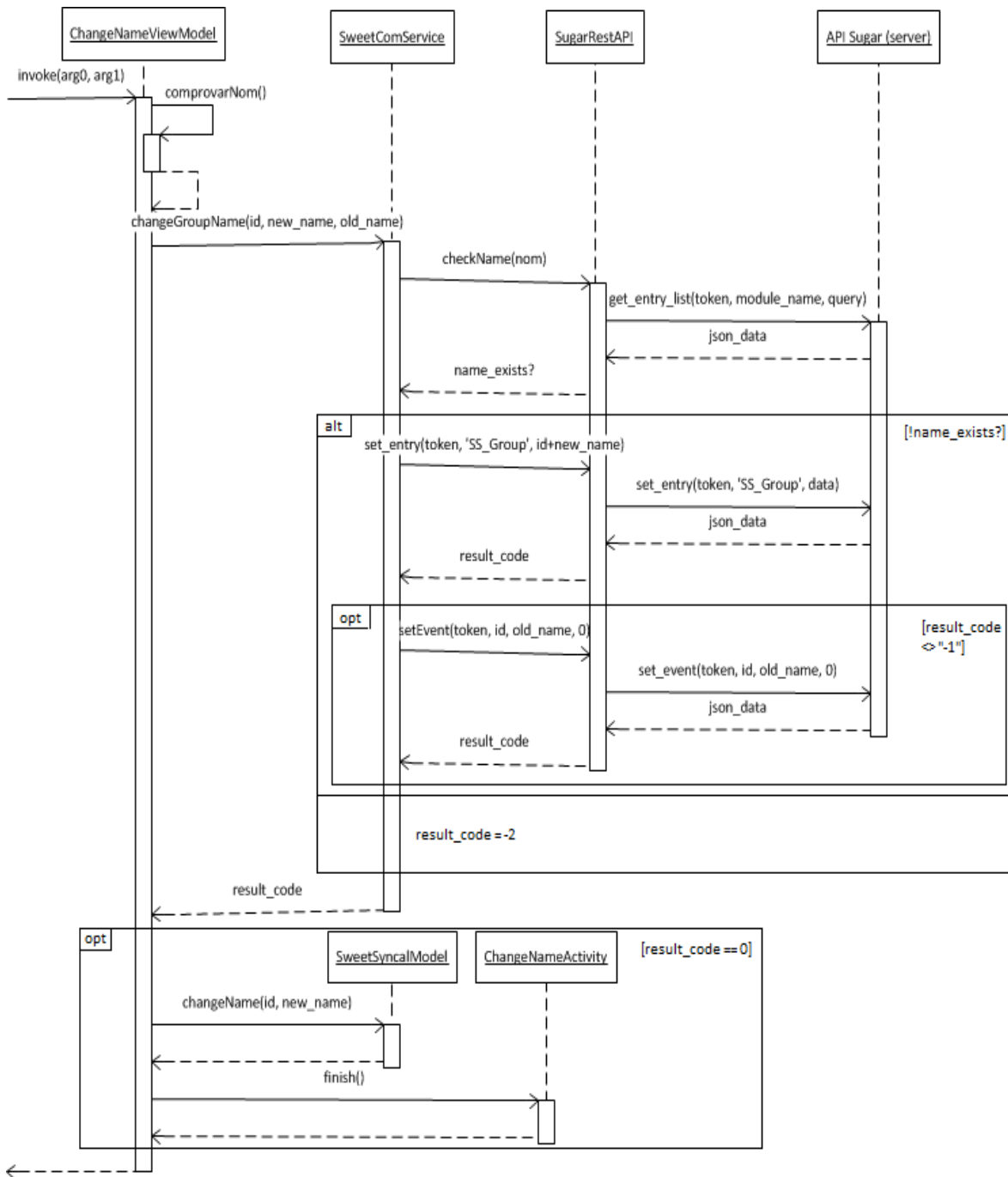


Figura 126: Diagrama seqüència disseny canviarNom



## Afegir participant grup: operació enviarParticipantsGrup(usuari)

La funció *add\_members* s'explica breument dins de l'operació *crearGrup* d'aquest mateix apartat. Si els membres s'afegeixen correctament al servidor, els afegim també a nivell local. La funció *cleanRequests* elimina de la base de dades local les sol·licituds dels usuaris afegits, com si haguessin estat acceptades. La funció *add\_members* també realitza aquesta funció a la base de dades del servidor.

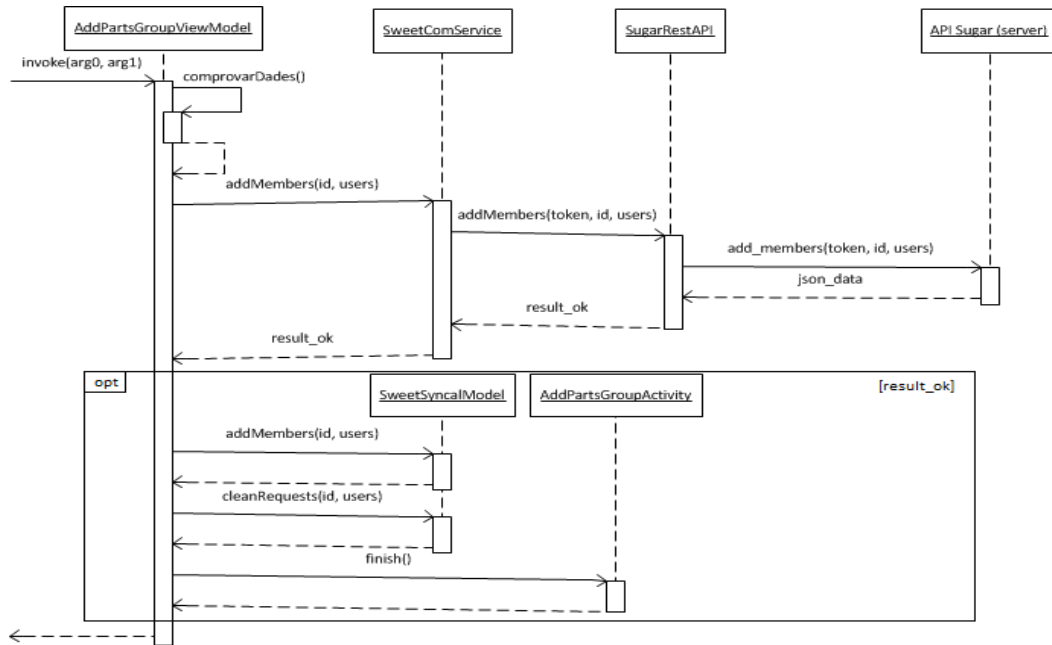


Figura 127: Diagrama seqüència disseny enviarParticipantsGrup

## Eliminar participant grup: operació confirmarElimPartGrup()

La funció *remove\_member* elimina el membre del grup i de totes les tasques en què participi a través del grup i crea els esdeveniments corresponents. Si finalitza correctament, eliminem el membre de la base de dades local.

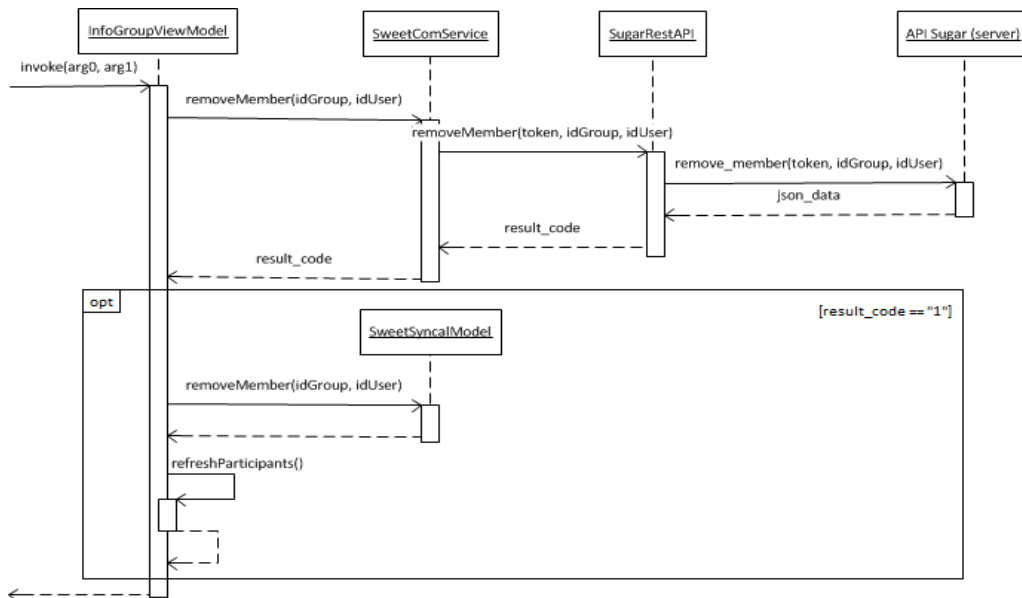


Figura 128: Diagrama seqüència disseny confirmarElimPartGrup

## Sortir d'un grup: operació confirmarSortirGrup()

La part important d'aquesta operació és la funció *leave\_group*, la resta és bastant trivial. L'explicació i diagrama d'aquesta funció es troben a continuació de la següent figura.

Si la funció *leaveGroup* finalitza correctament eliminem l'usuari del grup a nivell local. L'operació *leaveGroup* comprova si el grup és necessari (només guardem els grups als quals pertany l'usuari o els grups que participin en algun calendari en què participi l'usuari) i, en cas que no ho sigui, és eliminat. També s'eliminen els calendaris que ja no siguin necessaris perquè l'usuari ja no hi participa.

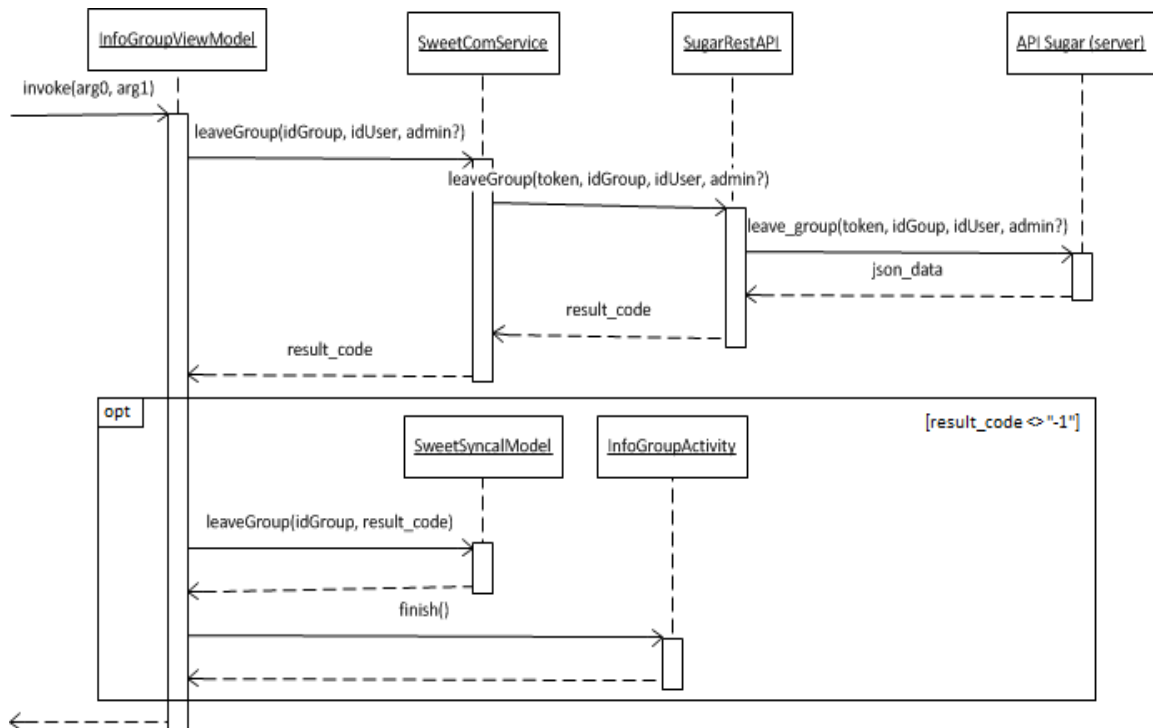


Figura 129: Diagrama seqüència disseny confirmarSortirGrup

## Sugar API: operació leave\_group

Primer de tot recuperem la relació que uneix el membre i el grup i l'eliminem. També cal eliminar l'usuari com a participant de totes les tasques en les que participava per ser membre del grup (funció *remove\_user\_from\_tasks*).

A continuació comprovem si queden més membres al grup. Si no en queden, podem eliminar el grup. Si en queda algun, crearem un esdeveniment per notificar-los que l'usuari ha sortit del grup. A part, si l'usuari que ha sortit del grup n'era l'administrador, s'assignarà un nou administrador al grup.

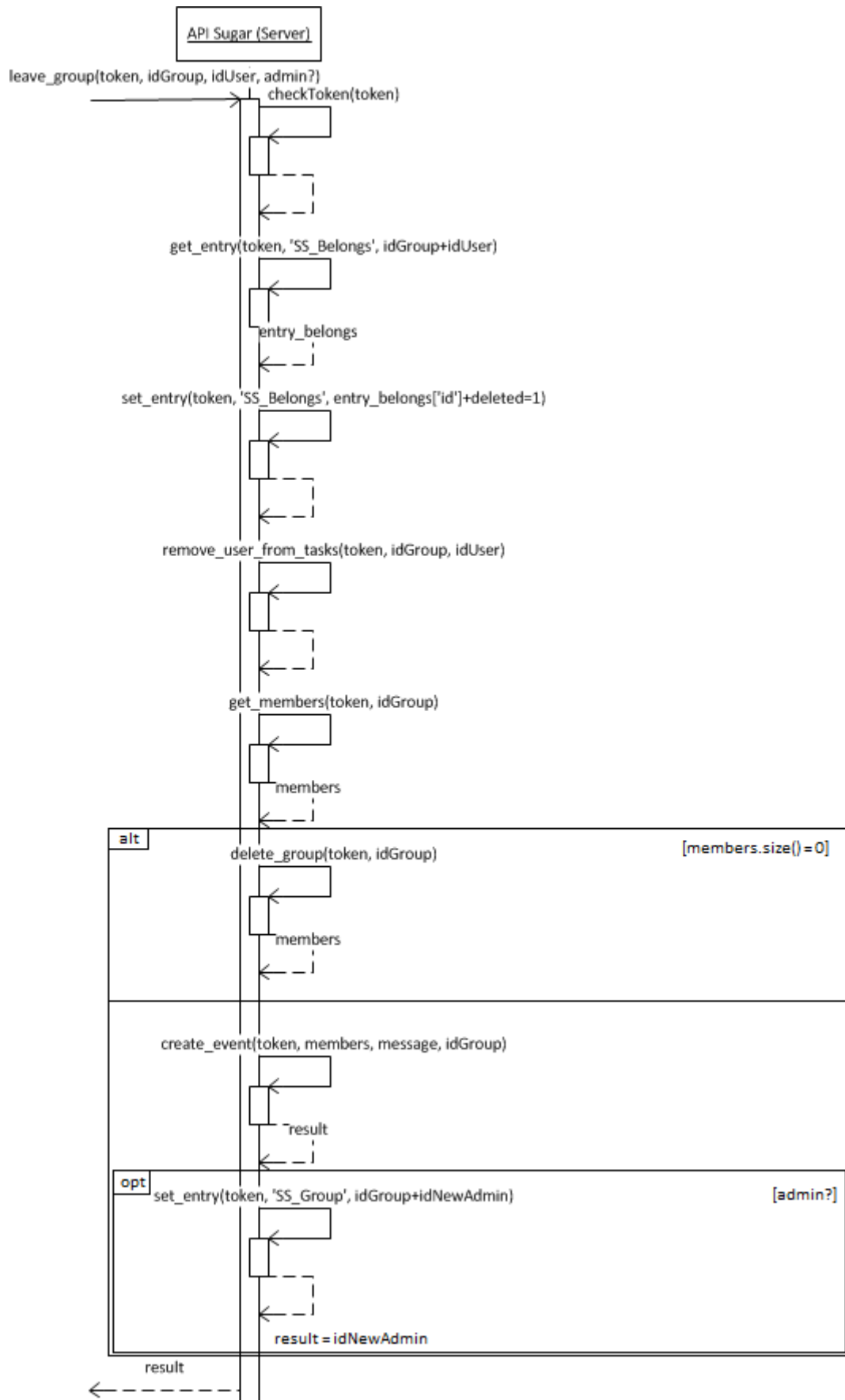


Figura 130: Diagrama seqüència disseny leave\_group

## Veure informació grup: operació veureInfoGrup(id)

Operació de veure informació. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària. Aquesta operació, però, és una mica diferent, ja que l'usuari pot veure la informació de grups als quals no pertany i que, per tant, no estaran a la base de dades local. Així doncs, si el grup no és a la base de dades local s'haurà de recuperar la seva informació del servidor. Això s'ha representat amb l'operació *getGroupServer*, el diagrama i explicació de la qual es pot trobar a continuació de la següent figura.

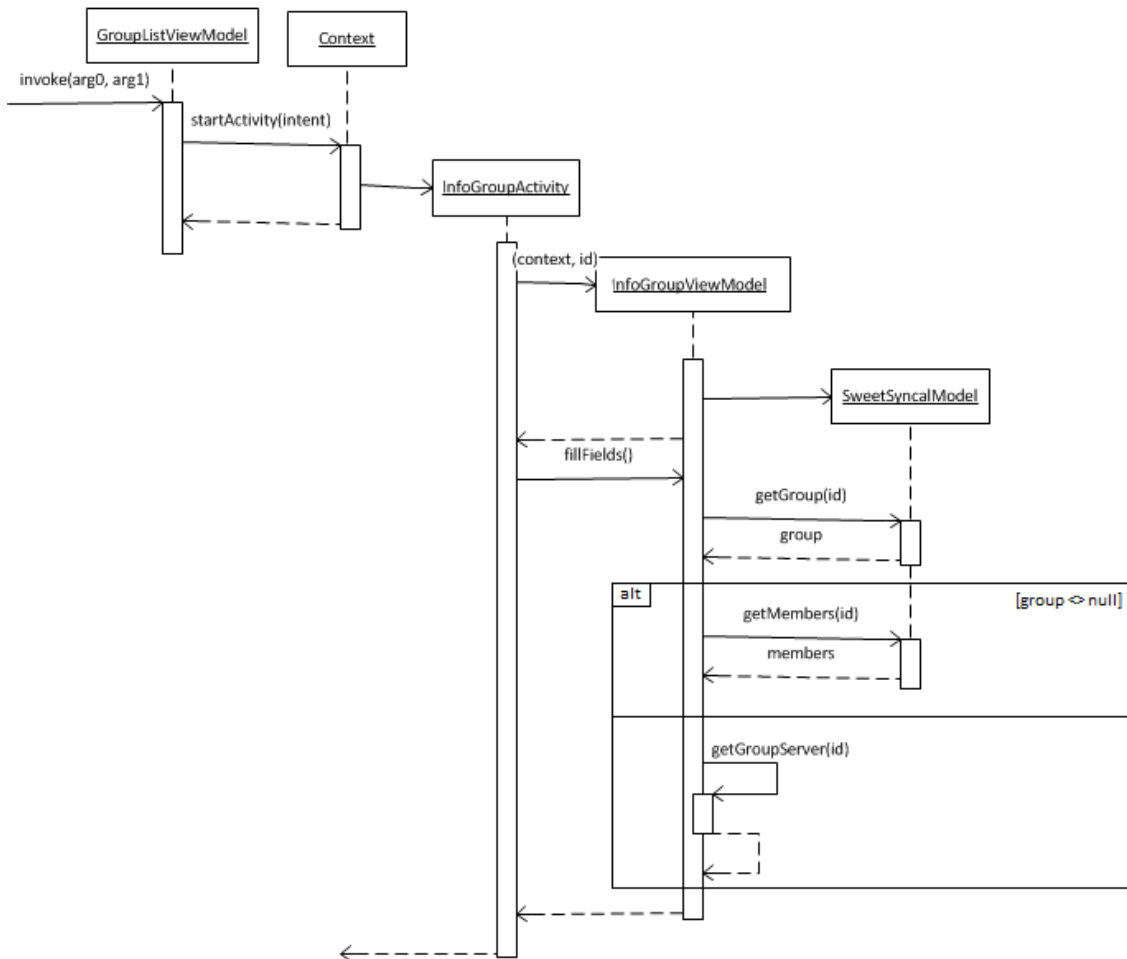


Figura 131: Diagrama seqüència disseny veureInfoGrup

La funció *getGroupServer* fa dues peticions al servidor: una per recuperar la informació del grup i una altra per recuperar els seus membres.

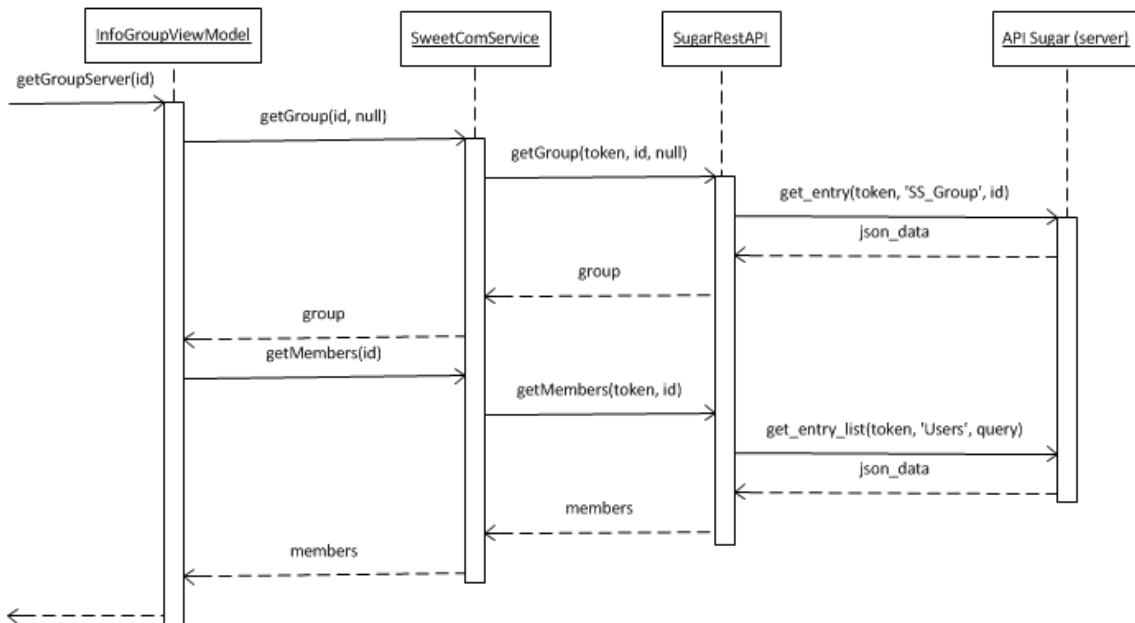


Figura 132: Diagrama seqüència disseny getGroupServer

### Llistar grups: operació llistarGrups()

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades local.

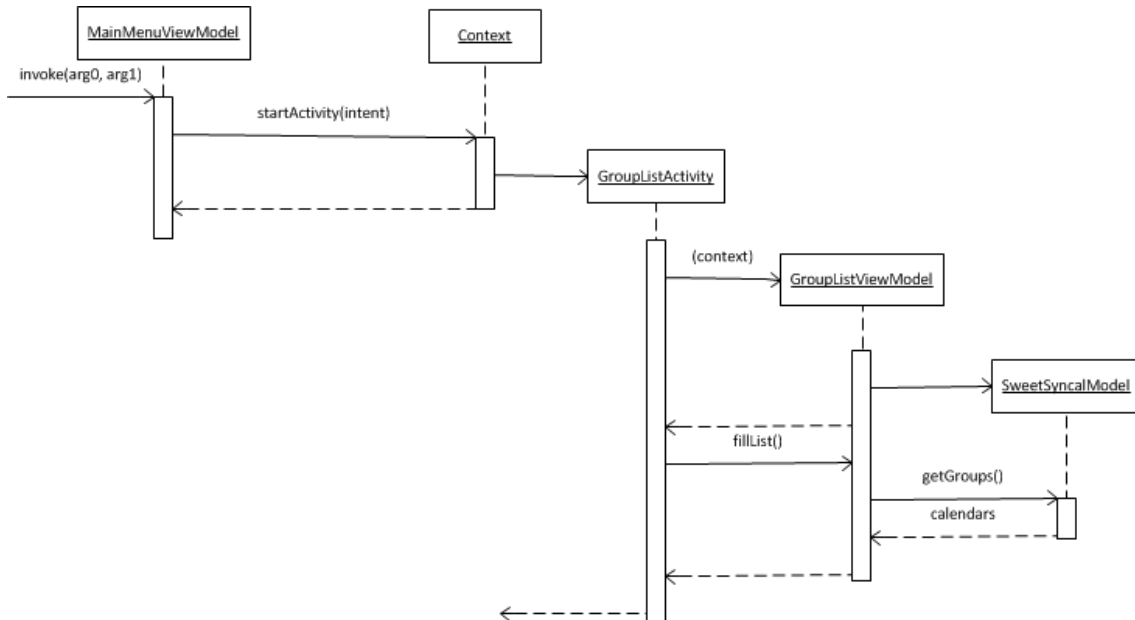


Figura 133: Diagrama seqüència disseny llistarGrups

### Nomenar administrador: operació confirmarNomenar()

Quan l'usuari nomena un nou administrador es modifica la informació del grup al servidor per tal que contingui el nou administrador i, a continuació, es crea un esdeveniment per notificar tots els membres del grup.

Si la funció *makeAdmin* finalitza correctament es fan els canvis a nivell local i s'actualitza la vista.

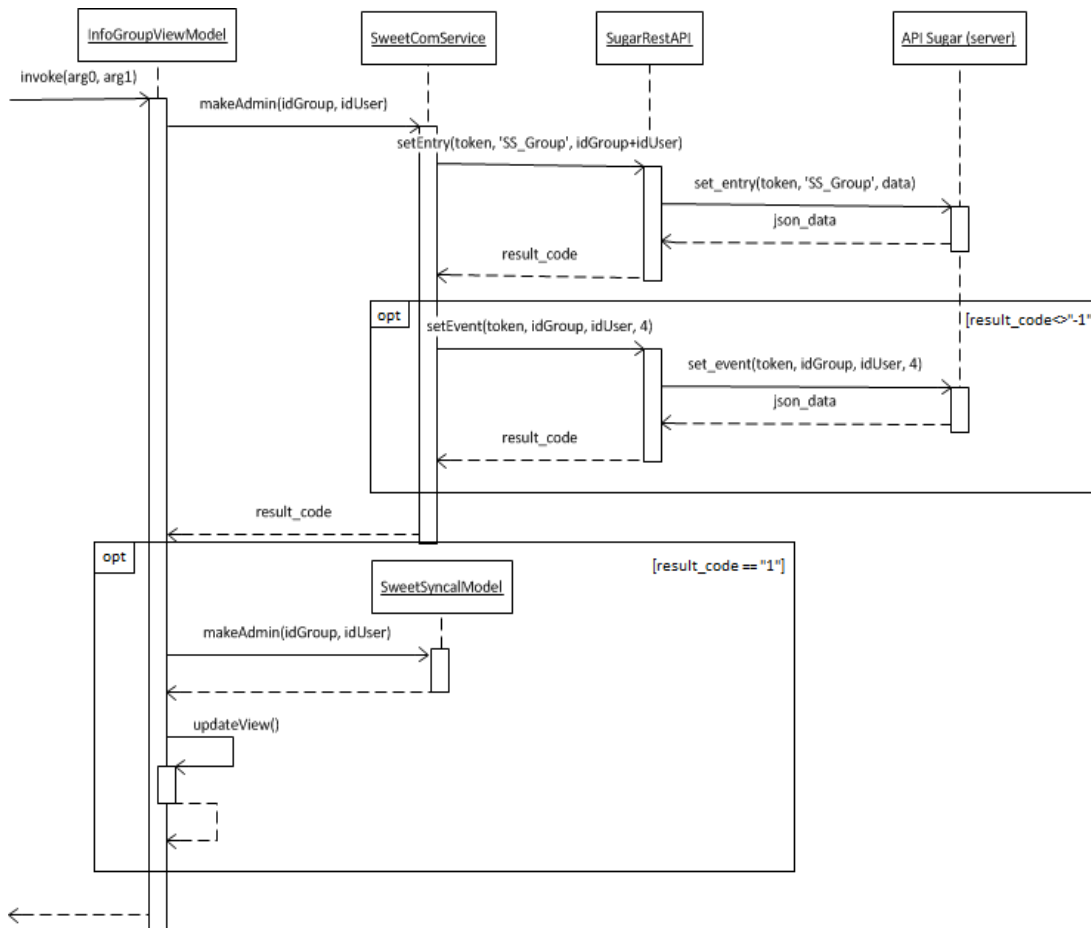


Figura 134: Diagrama seqüència disseny confirmarNomenar

### Cercar grup: operació cerca(text)

L'operació de cercar un grup és bastant senzilla. Únicament cal fer una query al servidor perquè retorni tots els grups que continguin en el seu nom la cadena introduïda i tractar la informació retornada. Cal actualitzar la vista perquè contingui els grups retornats.

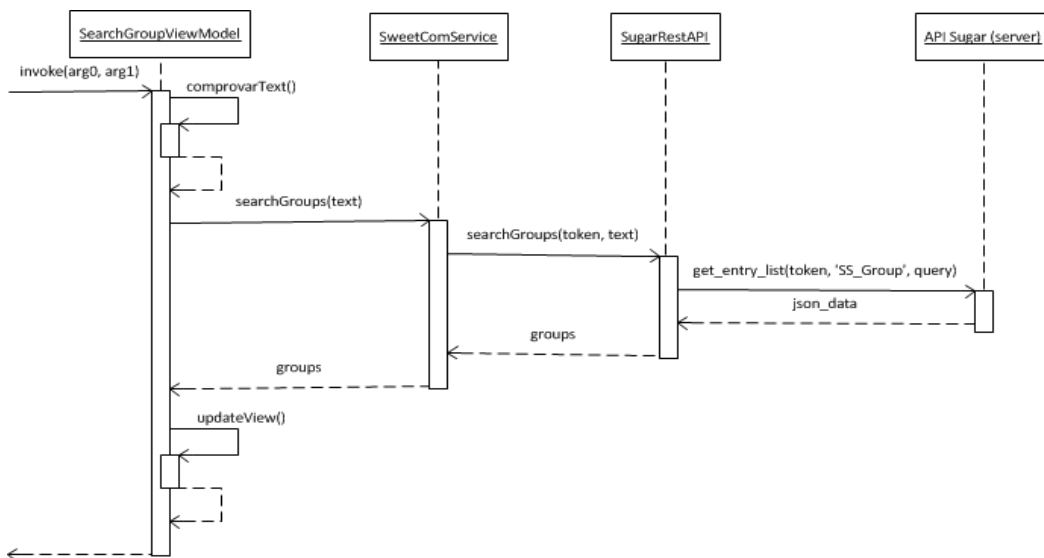


Figura 135: Diagrama seqüència disseny cerca

## Enviar/Cancel·lar sol·licitud: enviarOCancelarSolicitud()

Per enviar o cancel·lar una sol·licitud primer de tot cal intentar recuperar la sol·licitud del servidor. Si la sol·licitud existeix i es vol eliminar, s'elimina del servidor. Si la sol·licitud no existeix i es vol crear, es crea al servidor. En tots dos casos es crea un esdeveniment per notificar a l'administrador del grup. Si la sol·licitud es vol crear i ja existeix o es vol eliminar i no existeix es notificarà l'error a l'usuari. La funció *setEvent* no es mostra en detall al diagrama perquè ja ha aparegut repetides vegades en diagrames anteriors.

Si l'operació *sendRequest* finalitza sense error es crearà o eliminarà la sol·licitud a nivell local.

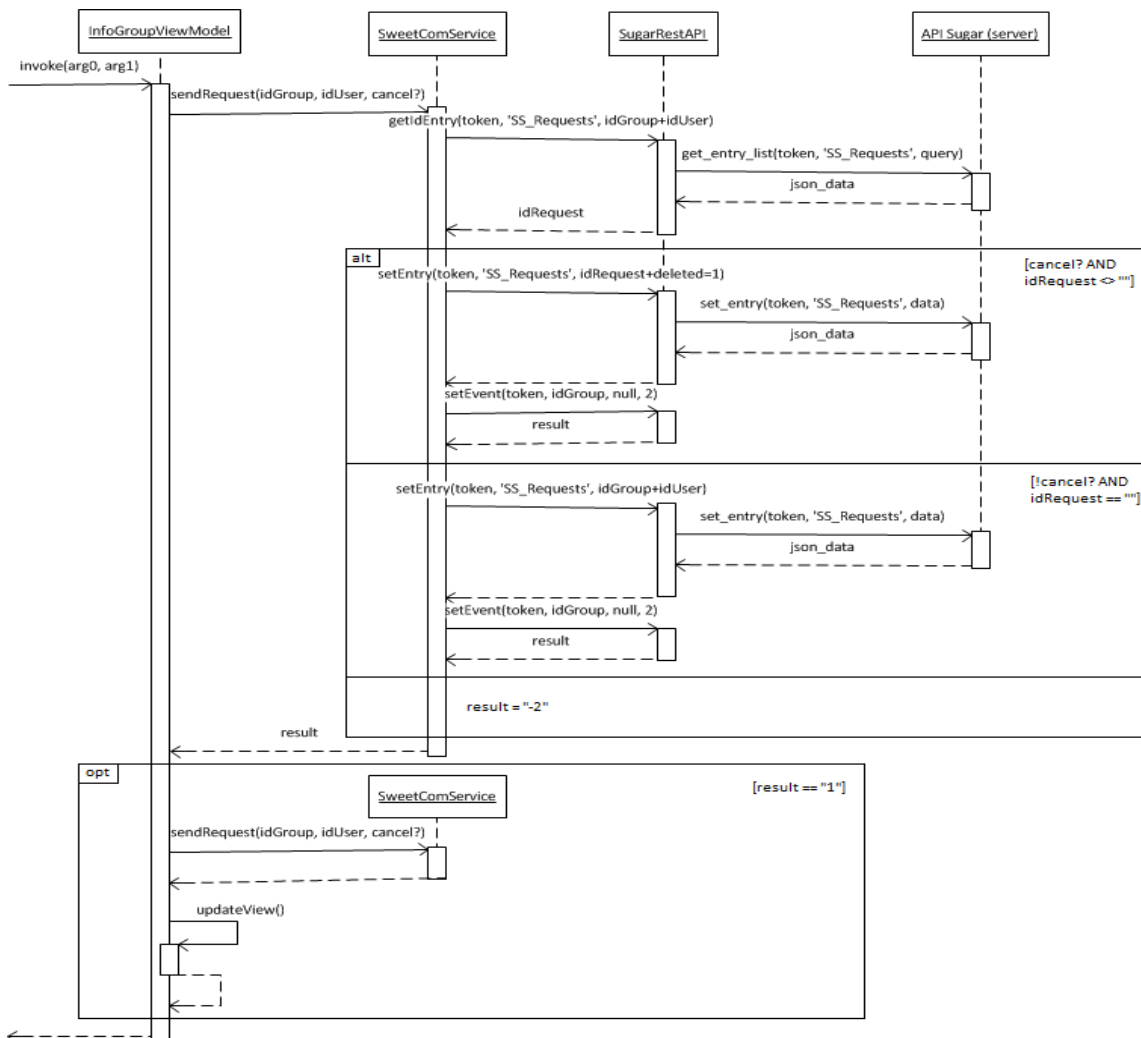


Figura 136: Diagrama seqüència disseny enviarOCancelarSolicitud

## Llistar sol·licituds: operació llistarSolicitudes()

Operació de llistar. Es crea l'Activity i el ViewModel corresponent i es recupera la informació necessària de la base de dades local.

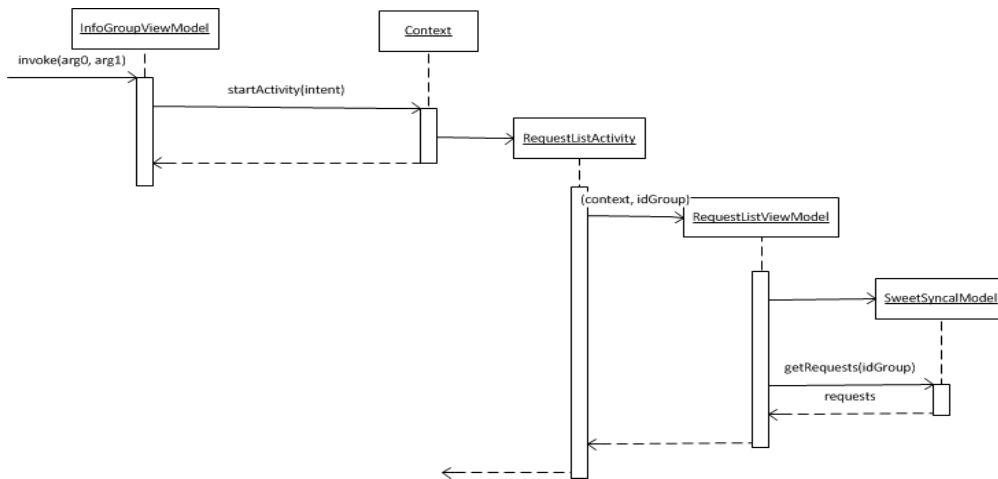


Figura 137: Diagrama seqüència disseny llistarSolicitudes

### Acceptar/Rebutjar sol·licitud: operació respondreSolicitud(acceptar?)

Per acceptar o rebutjar una sol·licitud primer cal recuperar l'identificador de la sol·licitud en qüestió i eliminar-la. En cas que la sol·licitud s'hagi acceptat, l'usuari que la va enviar és afegit com a membre del grup. Cal crear un esdeveniment per notificar als usuaris afectats. La funció *setEvent* no es mostra en detall al diagrama perquè ja ha aparegut repetides vegades en diagrames anteriors.

Si la funció *resolveRequest* finalitza sense error es fan els canvis a nivell local.

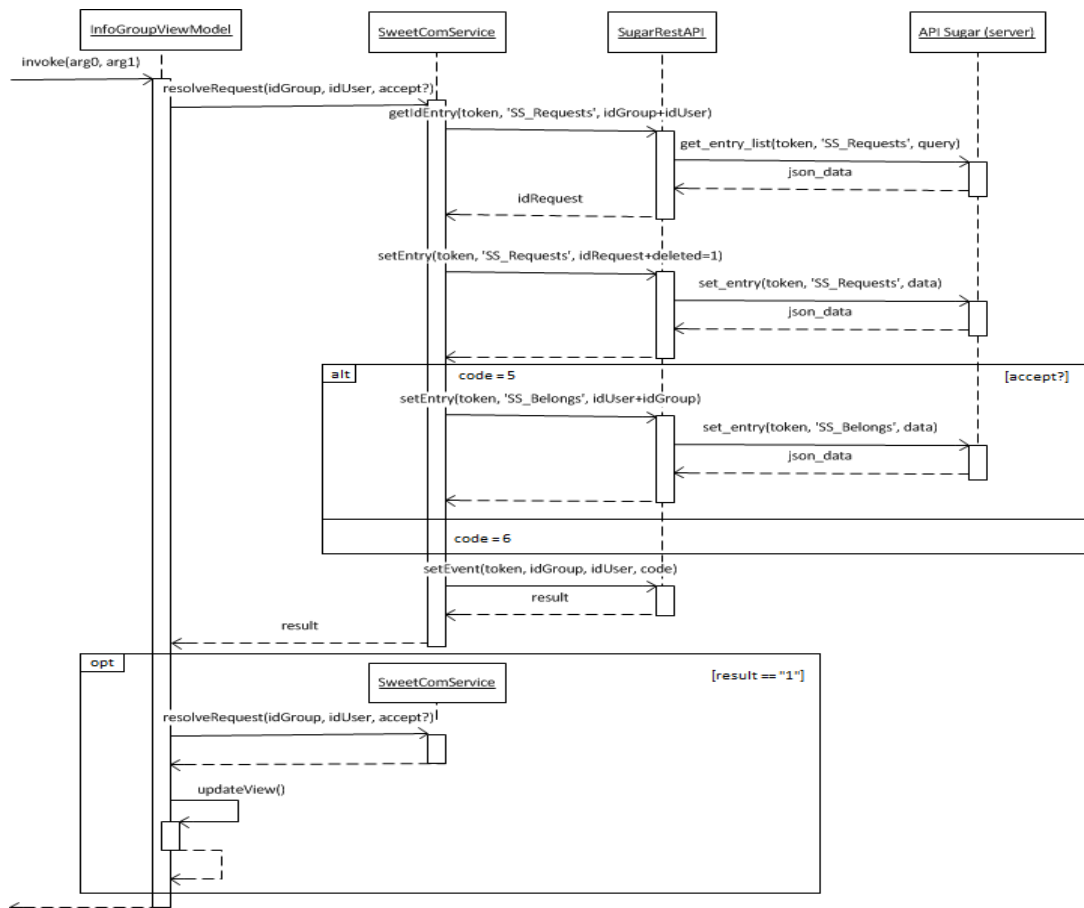


Figura 138: Diagrama seqüència disseny respondreSolicitud



### 3. Disseny de navegació

Per finalitzar l'etapa de disseny a continuació es mostra un diagrama on es poden veure les interaccions entre les vistes.

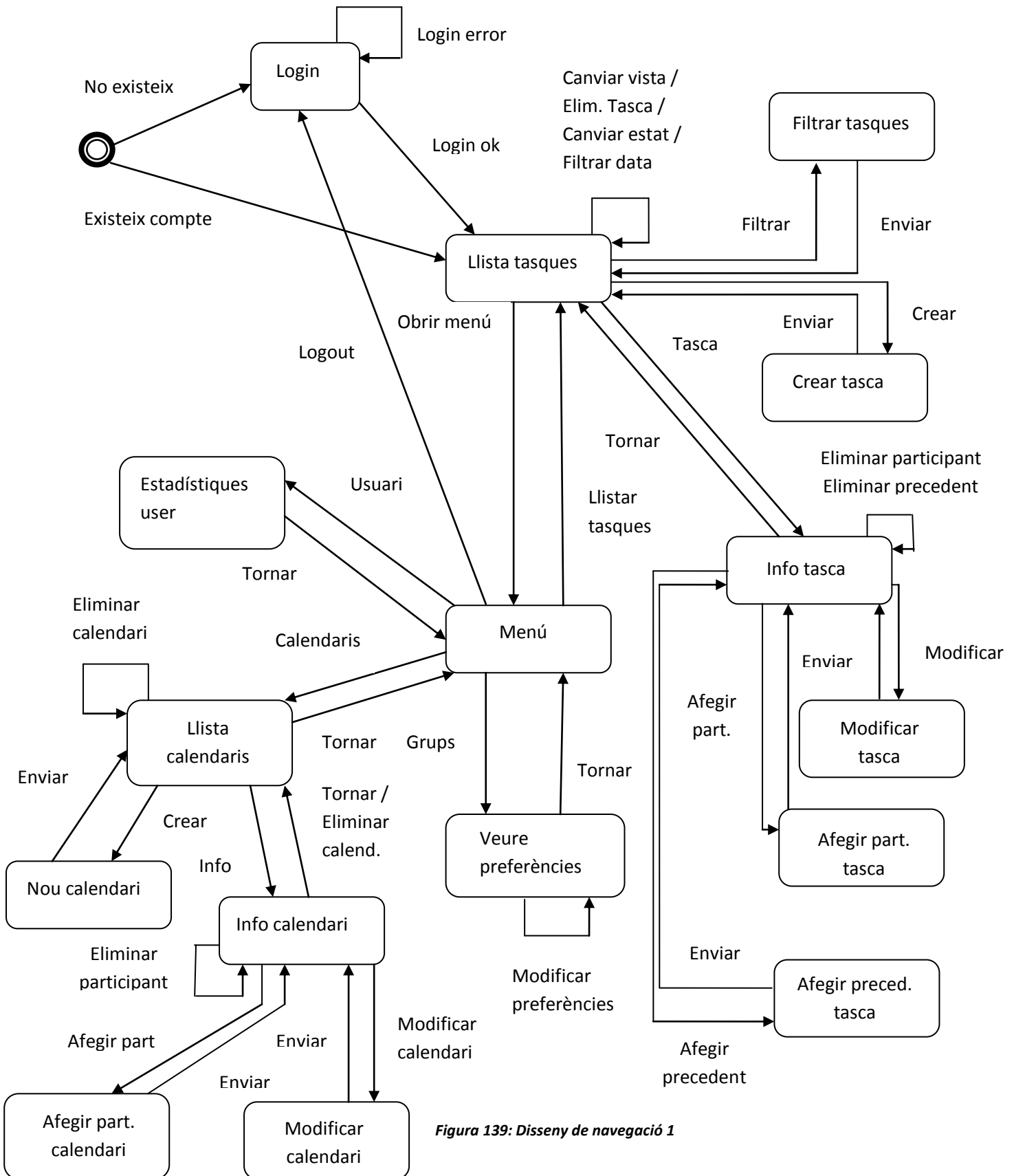


Figura 139: Disseny de navegació 1



# Implementació

A l'etapa d'implementació es codifica el sistema en un o diversos llenguatges de programació, obtenint així un programa o aplicació utilitzable per a l'usuari. Aquest apartat no està dedicat a explicar detalladament la implementació del sistema, sinó que únicament s'explicaran els aspectes més rellevants.

# 1. Aspectes importants

## Importància de la claredat en el codi

Per començar, independentment de com s'implementi el sistema i del llenguatge de programació utilitzat, s'ha de destacar la importància de la claredat i netedat del codi de programació, especialment si aquest codi pot ser ampliat en un futur o alguna altra persona se n'ha d'encarregar del manteniment. Per tant, a l'hora de programar s'ha tingut en compte que el codi del sistema estigui ordenat, net i correctament comentat de manera breu i clara.

## Prioritat: funcionament vs estètica

Quan s'implementa un sistema cal tenir en compte que hi ha molts aspectes a tenir en compte: correcte funcionament, estètica, rendiment, memòria utilitzada, etc. Tots ells són importants per tal d'obtenir una bona aplicació que satisfaci el client, però a vegades cal fixar prioritats ja que no és possible dedicar-los-hi el mateix temps a tots. En el meu cas, vaig decidir centrar-me més en el funcionament en sí de l'aplicació (sense errors, buscar les opcions més eficients, etc.) abans que crear les vistes que més puguin atreure a l'usuari. Es va decidir fer-ho així per una banda perquè els *smartphones* tenen recursos limitats i és important minimitzar els recursos que utilitzi l'aplicació, i per l'altra perquè, personalment, la part estètica és la que pitjor se'm dóna. D'aquesta manera les vistes obtingudes són bastant senzilles i amb poca varietat cromàtica, però sense traspasar un mínim nivell que pugui causar animadversió a l'usuari.

# 2. Desplegament a la infraestructura RDLab

Tot el software corresponent a la part del servidor s'ha instal·lat en un servidor del RDLab (Laboratori de Recerca i Desenvolupament), del Departament del LSI. Aquest software consisteix bàsicament en SugarCRM, memcached i les dependències que aquests dos softwares necessiten.

La instal·lació del software ha estat realitzada pel personal del RDLab ja que no disposava de prou permisos per fer-ho. Un cop instal·lat, se'm va proporcionar un usuari amb els permisos suficients per poder configurar el software com fos necessari.

Gran part de la configuració de SugarCRM es pot fer mitjançant el seu panell d'administració, accessible via navegador web, i així s'ha fet. No obstant, també ha calgut realitzar canvis en el codi de SugarCRM; concretament, ha calgut ampliar la seva API en PHP per tal que oferís totes les funcionalitats necessàries i també ha calgut afegir una tasca periòdica per finalitzar els pending approvals no resposts al cap d'un cert temps. La configuració de SugarCRM va donar alguns problemes de permisos inicialment, ja que l'usuari que executava SugarCRM no era el mateix que l'usuari utilitzat per modificar el codi.

No va ser necessari configurar memcached, és suficient utilitzar la configuració per defecte. SugarCRM ja disposa dels mecanismes per incorporar aquesta caché a les seves operacions, no cal afegir-hi res més.

Per l'apartat d'experimentació es va haver de convertir el servidor en un *rendezvous peer* i un *relay peer*. La llibreria *PeerDroid* utilitzada ofereix els mecanismes per fer-ho, però cal executar els programes pertinents a l'entorn del servidor. Com que la manera més senzilla d'executar aquests programes era utilitzar Apache Ant, es va instal·lar una distribució d'aquest software dins de l'entorn de treball de l'usuari proporcionat i es va utilitzar per executar el codi quan fos necessari. Si es volgués que els programes encarregats d'actuar com a *rendezvous peer* i com a *relay peer* fossin permanents caldria programar-los perquè s'executessin automàticament a l'iniciar al sistema.

Finalment, també es va haver de modificar el firewall per a permetre les comunicacions necessàries. Les modificacions van ser fetes pel personal del RDLab en funció de les necessitats que els hi comunicava. Concretament, va ser necessari permetre tràfic de sortida cap als servidors de Google encarregats del servei GCM i cap als servidors d'Apache Ant, i també tràfic d'entrada cap als programes encarregats de comportar-se com a *rendezvous peer* i *relay peer* ja que tots dos utilitzaven un port diferent als que solen ser habituals.

### 3. Principals problemes trobats

Durant la implementació del sistema han anat apareixent alguns problemes que no s'havien previst inicialment i que han obligat a realitzar alguns canvis en la implementació. A continuació es llisten els més importants.

#### **Android-Binding encara s'està desenvolupant**

La llibreria Android-Binding que, com s'esmenta anteriorment, es va decidir utilitzar per tal d'aplicar el patró MVVM, encara està en fase de desenvolupament, cosa que provoca que algunes funcionalitats i elements no estiguin implementats. Alguns dels que s'han trobat a faltar són:

- AutoCompleteTextView
- DatePicker
- TimePicker
- ViewFlipper
- Les pestanyes (tabs) es troben en una fase experimental
- onFocusChangeListener
- AuthenticatorActivity
- No es poden crear components d'un layout dinàmicament

En els cinc primers casos es va optar per assignar els valors a aquests components utilitzant les funcions per defecte d'Android, obligant a que els ViewModels corresponents siguin menys independents de les vistes.

En el cas del listener, es va optar per no utilitzar aquest event, es va substituir per un altre modificant lleument la funcionalitat.

Per tal d'implementar el mecanisme de login i d'Accounts cal utilitzar un tipus especial d'Activity anomenat AuthenticatorActivity que és subclasse de la classe Activity. Per altra banda, per utilitzar Android-Binding cal utilitzar la classe BindingActivity, també subclasse d'Activity. Com que en java no es pot utilitzar herència múltiple, tots dos mecanismes són incompatibles. Una possible solució seria implementar una tercera classe que contingui les funcionalitats de tots dos tipus, però com que el projecte s'estava endarrerint es va descartar l'opció i es va optar per no utilitzar Android-Binding en el mecanisme de log in.

Per últim, per tal de visualitzar el calendari es van considerar dues opcions. La primera d'elles, crear tots els TextViews necessaris cada cop que es canviï el mes; la segona, tenir tots els TextViews creats inicialment i modificar les valors, drawables i visibilitat en funció del mes visualitzat.

Android-Binding bàsicament afegeix algunes propietats i funcionalitats als components d'Android. La manera d'aplicar aquests components és a través dels fitxers XML ja que la seva política és separar la vista del ViewModel, no tindria sentit haver de recuperar la vista per tal d'aplicar les propietats que eviten haver de recuperar la vista. És a dir, des del ViewModel no es poden aplicar les propietats de binding, de manera que no podem utilitzar Android-Binding sobre objectes creats dinàmicament, que és el que fem amb la primera opció esmentada al paràgraf anterior.

Per tant, si volem aplicar Android-Binding hem d'escollir la segona opció, però aquesta també té els seus inconvenients. Per començar, tindrem una determinada quantitat de TextViews que seran visibles en casos molt determinats i que tot i així seguiran ocupant memòria. A més, el rendiment serà similar en tots dos casos, ja que sempre haurem de recórrer tots els dies, ja sigui per crear el TextView com per modificar-lo. El principal inconvenient, però, ve derivat de Android-Binding. Cada text diferent que hagi de ser modificat des del ViewModel necessita tenir un StringObservable associat, és a dir, necessitaríem un total de 42 StringObservables pels dies, un per cada TextView. A més, cada cop que es clica en un dia s'ha d'identificar quin dia s'ha clicat, i amb els Commands això no és possible, l'única manera seria tenir 42 Commands, un per cada TextView, que fessin tots el mateix però amb un valor de dia diferent. Tenint en compte la limitació de recursos d'Android, aquesta opció és molt poc recomanable, així que es va optar per crear les vistes dinàmicament sense utilitzar Android-Binding.

En resum, les limitacions d'Android-Binding han obligat a no complir el patró MVVM en alguns casos determinats.

## SugarCRM: mòduls vs relationships

SugarCRM utilitza mòduls i relacions per a emmagatzemar la informació. Els mòduls són un equivalent a les classes d'un model UML, mentre que les *relationships* serien l'equivalent a les associacions. Inicialment s'havia decidit utilitzar *relationships* per representar les associacions, però mentre s'implementava aquesta solució va aparèixer un problema: tot i que les *relationships* poden tenir més informació guardada a part dels identificadors de les instàncies dels mòduls que uneixen (no obstant, afegir informació extra no és fàcil), aquesta informació és bastant difícil de recuperar. La API de *SugarCRM* ofereix un conjunt de funcions per recuperar i modificar instàncies de mòduls i de relacions, però cap d'elles permet recuperar aquesta informació extra de les *relationships*, l'única manera d'aconseguir-ho és atacar la base de dades directament i, amb paraules textuais d'un dels administradors de *SugarCRM*, "it can get quite hairy". El fet d'atacar-la no és excessivament complicat (sempre que es coneguin els noms de les taules i de les seves columnes), però els mecanismes de consulta i modificació que ofereix *SugarCRM* permeten controlar l'accés i la concurrència a les dades, cosa que no es garanteix si s'ataca la base de dades directament, així que es poc recomanable utilitzar aquesta opció.

La principal informació de les *relationships* que ens interessa és la data de modificació per tal de poder recuperar la informació modificada a partir de certa data (necessari per a una correcta sincronització). La API ofereix una funció anomenada *get\_modified\_relationships* que, a simple vista, sembla que permeti recuperar les relacions modificades dins del període de temps indicat per paràmetre. No obstant, quan es va observar en detall la implementació d'aquesta funció es va descobrir que aquesta funció només serveix per a les relacions *User-Meeting*, *User-Call* i *User-Contact*, i la data de modificació que consulta és la del segon mòdul (*Meeting*, *Call* o *Contact*), no la de la *relationship*. És a dir, la API *SugarCRM* no ofereix cap funció que permeti consultar la data de modificació d'una *relationship* o filtrar una consulta en base a aquesta data, de manera que no és possible implementar la sincronització d'aquest projecte amb *relationships* a no ser que s'implementi també una funció que compleixi aquest objectiu.

Això no és un inconvenient massa important, ja que es pot modificar fàcilment la funció *get\_modified\_relationships* per tal d'ampliar-la a altres relacions i que s'adapti a les nostres necessitats (i, de fet, es farà). Tot i així, arribat a aquest punt es va decidir realitzar un anàlisi d'avantatges i inconvenients d'utilitzar *relationships* o, en canvi, utilitzar mòduls per representar aquestes relacions.

	Relationships	Mòduls
Avantatges	<ul style="list-style-type: none"><li>-Més coherent amb el model.</li><li>-Evita associacions repetides automàticament.</li><li>-Ocupa menys espai a BD.</li></ul>	<ul style="list-style-type: none"><li>-Recuperar informació extra és molt fàcil.</li><li>-Consultes de relacions són molt semblants siguin 1-1, 1-* o *-*.</li></ul>

<b>Inconvenients</b>	<ul style="list-style-type: none"> <li>-Recuperar informació extra és complicat.</li> <li>-Funcions de Sugar pensades per recuperar relacions 1-*. Per recuperar *-* cal cridar funció 1-* diverses vegades.</li> <li>-En relacions recursives no permet diferenciar extrems.</li> </ul>	<ul style="list-style-type: none"> <li>-Menys coherent amb el model i menys abstracte de la BD.</li> <li>-Cal comprovar manualment que no existeixi ja una instància per representar la relació.</li> <li>-Ocupa més espai a BD.</li> <li>-No es pot definir foreign key.</li> <li>-En certes operacions calen més consultes.</li> </ul>
----------------------	--	--

*Taula 155: Avantatges i inconvenients mòduls i relationships*

El fet de no poder recuperar informació extra ens obliga a utilitzar mòduls en el cas dels *Vots* i *Respostes*, ja que aquestes dues classes tenen més informació a part dels identificadors d'enllaç. A part, com que no es poden diferenciar els extrems en funcions recursives, la relació *Precedeix* ha de ser un mòdul obligatòriament. Així doncs, calia decidir què fer en la resta d'associacions.

Els principals inconvenients d'utilitzar mòduls són l'espai que necessiten, que en alguns casos requereixen més consultes a la base de dades i que és més fàcil equivocar-se (ja que no hi ha restricció de clau forana ni de clau primària, poden haver-hi associacions repetides o amb identificadors que no existeixen). Com que totes les insercions a la base de dades les realitza el sistema, l'usuari mai atacarà directament la BD, el problema d'equivocar-se no implica cap problema a l'usuari, simplement s'ha de tenir en compte a l'implementar les funcions. Pel què fa al nombre de consultes (i, per tant, al temps necessari per a obtenir la informació), utilitzar mòduls és bastant pitjor en el cas de voler recuperar relacions 1-\*, però l'operació *get\_info* (utilitzada durant la sincronització) fa moltes operacions \*-\*, i aquesta és l'operació de consulta més utilitzada amb diferència, així que el fet de calcular quin dels dos mètodes és més eficient no és trivial.

Per altra banda, el principal inconvenient de les *relationships* és la dificultat de recuperar informació extra. Cal tenir en compte que un dels objectius del projecte és facilitar possibles ampliacions futures. Ara mateix es poden implementar la majoria d'associacions utilitzant *relationships* perquè no necessitem recuperar-ne informació extra, però potser en una futura ampliació s'hi inclou algun atribut important o simplement es vol recuperar la data de modificació per tal de desar-la a nivell local. En aquest cas, s'haurien de convertir les *relationships* en mòduls i modificar totes les funcions afectades, una feina que comportaria bastant temps perdut. Si afegim això al fet que els inconvenients d'utilitzar mòduls per representar les associacions són bastant acceptables, la decisió presa és **utilitzar mòduls per representar les relacions**.

Cal afegir, però, que a l'apartat d'experimentació es farà una comparació entre l'eficiència i l'espai necessari de totes dues versions per tal d'analitzar si la decisió presa ha estat correcte.



## **Relotges dels mòbils i del servidor no sincronitzats**

Pot succeir (i és probable que passi) que els rellotges dels telèfons mòbils no estiguin sincronitzats amb el del servidor, provocant una diferència horària que s'ha de tenir en compte a l'hora de sincronitzar. Per exemple, si es fa un canvi al servidor a les 00:00:00 i hi ha telèfons mòbils que en aquest moment tenen com a hora les 00:00:30, aquests canvis no serien sincronitzats al telèfon ja que es detectarien com a canvis anteriors a la darrera sincronització.

Per a solucionar-ho, quan l'usuari faci log in a l'aplicació es calcularà aquesta diferència horària. Per tal de calcular-ho, el telèfon enviarà al servidor la seva data actual, el servidor la compararà amb la seva data actual i retornarà la diferència. No obstant, aquesta diferència no serà el nombre exacte, ja que cal tenir en compte el temps que tarda el telèfon a preparar i enviar el missatge i també el temps que tarda el missatge a viatjar del telèfon al servidor, quan el servidor rep l'hora del telèfon ja han passat unes centèsimes de segon des de que el telèfon l'ha calculat. La diferència de temps serà positiva si el telèfon va avançat respecte el servidor i negativa altrament.

`data_consulta = data_telèfon - time_difference`

En cas que el rellotge del telèfon estigui avançat el fet que la diferència no sigui el nombre exacte no implicarà cap problema, ja que la diferència de temps serà superior al que hauria de ser i agafarem informació d'una mica abans del temps que hauríem de fer-ho. És a dir, no perdrem informació, en tot cas podríem rebre algun canvi dues vegades, i això no suposa cap problema.

En cas que el rellotge del telèfon estigui endarrerit respecte el del servidor sí que podem tenir problemes, ja que la data de consulta seria superior a la que hauria de ser, i si es fes un canvi just en aquesta diferència entre la data de consulta real i la que fem no el recuperariem. És per aquest motiu que, en aquest cas, s'augmentarà la diferència en un segon per tal de compensar l'error de càlcul de manera que, com en el cas anterior, ens podríem trobar que rebem alguns canvis més d'una vegada, però mai perdrem informació.

## **SugarCRM no disposa de transactions ni semàfors**

La major part de funcions implementades per al servidor realitzen més d'un canvi a la base de dades. En aquests casos és molt útil disposar de transactions, ja que si qualsevol dels canvis falla el sistema fa un rollback automàtic i deixa la base de dades tal com estava inicialment. SugarCRM, però, no implementa un sistema de transaccions, de manera que els rollbacks s'han hagut de fer manualment, un fet bastant molest.

Per altra banda, al servidor s'està implementant un sistema multi-usuari, i en aquests casos cal tenir en compte que els usuaris poden voler modificar la mateixa informació al mateix temps. Per tal d'evitar errors s'utilitzen els coneguts semàfors, però SugarCRM tampoc en disposa, i com que cada crida al servidor utilitza una instància diferent del webservice utilitzat per l'API no es poden utilitzar variables globals per simular-lo. Només hi ha dues maneres de simular els semàfors: desar

aquestes variables globals a la base de dades o instal·lar una caché externa i utilitzar-la per tal de compartir variables entre totes les crides. Com que les consultes a bases de dades solen ser bastant lentes s'ha optat per la segona opció, utilitzar una caché externa.

### **Bug a la llibreria Android-Binding**

Durant la implementació es va descobrir un bug a la llibreria d'Android-Binding que provoca que l'opció de carregar tasques del mes següent no funcioni, "s'encalla". Es tracta d'un error molt estrany i curiós que només succeeix en unes condicions molt determinades.

Es van descobrir dos dels elements que contribuïen a provocar l'error: les pestanyes (tabs) i la funció setCursor de la llibreria Android-Binding. Amb Android-Binding, per tal de mostrar les tasques en una llista s'utilitza un CursorCollection al qual li proporciona el Cursor obtingut de la consulta a la base de dades i t'ho assigna a la vista determinada. Per assignar el cursor al CursorCollection s'utilitza la funció setCursor esmentada anteriorment.

Es va intentar substituir les pestanyes d'Android (TabHost) per un mecanisme similar que ho simulés, però l'error seguia succeint. Es va estar treballant amb un dels creadors de la llibreria, i davant l'incapacitat de solucionar l'error es va optar per implementar aquesta part de l'aplicació sense utilitzar Android-Binding, de manera que en aquesta part es deixa d'aplicar el patró MVVM.

## **4. Proves i correcció d'errors**

Per tal de garantir que el sistema funciona correctament s'han utilitzat dues metodologies. Primer de tot, com és habitual en tota implementació, s'han anat provant tots els escenaris possibles d'una funcionalitat cada cop que s'acabava d'implementar per tal de corregir-ne els errors, ja que sol ser freqüent que n'apareguin i no és recomanable seguir implementant sense saber si el que s'ha fet fins ara està bé o no. Un cop finalitzada la implementació de tot el sistema s'ha tornat a provar l'aplicació sencera simulant diversos escenaris per tal de corregir errors que s'hagin pogut passar per alt o que vinguin derivats de funcionalitats implementades posteriorment a la tractada.

No obstant, és difícil tractar tots els casos possibles, de manera que es pot haver passat per alt algun error. És per això que es va decidir instal·lar l'aplicació en diversos dispositius mòbils per tal que provessin l'aplicació durant un període determinat de temps, obtenint així feedback per part dels usuaris, dades estadístiques del sistema i un conjunt d'escenaris més real i ampli que els que es puguin simular.

# Experimentació

A l'etapa d'experimentació s'implementen algunes alternatives en el procés de sincronització dels calendaris i es realitzen un seguit de proves amb diferents escenaris per tal de comparar els avantatges i inconvenients de cada cas.

# 1. Client-Servidor vs Peer-to-peer

Una de les arquitectures més utilitzades en la informàtica és la de client-servidor, però això no vol dir que sigui l'única que existeix. Alguns programes o serveis molt coneguts utilitzen P2P per transferir informació o fitxers entre els seus usuaris. En aquest experiment observarem la diferència en temps, espai i tràfic d'aquestes dues arquitectures aplicades al cas pràctic que s'ha implementat en aquest projecte.

Per tal de fer aquesta comparació s'ha implementat un sistema P2P molt senzill i limitat. Pera més informació sobre les xarxes P2P, llegir l'apartat 7.3 de la Introducció del projecte,.

Una xarxa P2P pura no utilitza cap servidor, sinó que tota la comunicació es fa entre clients o *peers* i tots ells tenen la mateixa responsabilitat. Aquest tipus de xarxa no és gens pràctica, ja que si no s'organitza el tràfic entre *peers* utilitzant mètodes eficients el resultat és un sistema amb un rendiment molt baix. Per tal de solucionar-ho cal que alguns *peers* tinguin més responsabilitats que altres i els ajudin a trobar altres *peers*. Aquest tipus de "*super peers*" s'anomenen *rendezvous peers*. Una xarxa P2P sol estar formada per un conjunt de *rendezvous peers*, i cadascun d'ells s'encarrega d'organitzar un grup de *peers*.

Hi ha un altre tipus especial de *peers* anomenats *relay peers*. Aquests s'utilitzen quan un *peer* no pot ser accedit directament (per exemple, si es troba darrere d'un firewall o NAT) i s'utilitza com una espècie de *proxy*: la resta de *peers* envien els missatges al *relay peer* i aquest els redirigeix cap al *peer* que representa.

Com que la nostra xarxa és petita tindrem únicament un *relay peer* i un *rendezvous peer*, tots dos ubicats al mateix servidor. La responsabilitat del *rendezvous peer* s'ha limitat al màxim per tal d'acostar-se tant com es pugui a una xarxa P2P pura: el *rendezvous peer* es limitarà a mantenir la llista de *peers* amb les seves adreces i a transmetre-la als nous *peers* que s'afegeixin a la xarxa.

Com ja s'ha comentat anteriorment, el sistema implementat és molt senzill i limitat. Mitjançant P2P únicament sincronitzarem relacions de precedència entre tasques i suposarem que no hi ha conflictes entre els usuaris (escenari optimista). En cas que es volguessin tractar els conflictes seria necessari que els *rendezvous peers* emmagatzemessin també la informació i fossin ells qui tractessin els conflictes, ja que tots els *peers* els enviarien els canvis a ells enlloc de als *peers* participants i serien els *rendezvous* els que els propagarien. Com es pot observar, aquesta arquitectura és molt semblant a la client-servidor amb estratègia *push*, amb la diferència que hi hauria més d'un servidor i aquests es comunicarien entre ells per propagar la informació. Per simplificar i com que ens interessa obtenir una xarxa P2P el més pura possible per poder observar bé les diferències entre les arquitectures s'ha preferit generar situacions sense conflictes.

En aquest experiment utilitzarem 4 dispositius o *peers*: dos d'ells seran emuladors connectats a la mateixa xarxa *wi-fi*, mentre que els altres dos seran dispositius mòbils reals connectats a la xarxa 3G. En cada execució s'enviarà un conjunt diferent de relacions *Precedes* i s'observarà el temps que es tarda a sincronitzar, el volum dels missatges enviats i el tràfic generat. Cal tenir en compte, a més, que en

la xarxa P2P un *peer* pot rebre la informació de dues maneres: o bé la rep directament quan un altre *peer* la modifica i l'envia a tots els participants, o bé li demana a un altre *peer* que li envii els canvis fets a partir de certa data (cas que succeirà si el *peer* ha estat desconnectat de la xarxa mentre es feien els canvis). Per tal de tractar els dos casos es simularan desconnexions per tal d'observar quan temps tarden els *peers* a sincronitzar-se un cop es reconnecten a la xarxa.

Podem resumir les característiques de l'experiment amb la taula següent:

<b>Observació</b>	Hi ha arquitectures millors i pitjors pel cas concret en què s'apliquen.
<b>Plantejament</b>	Escollim dues arquitectures i observem els seus avantatges i inconvenients.
<b>Hipòtesis</b>	Totes les arquitectures són iguals en temps i tràfic (H0) o hi ha arquitectures millors que altres en alguns casos.
<b>Mètode</b>	<p>-Executarem un total de 7 rèpliques. Cada rèplica l'executarem deu vegades i anotarem el temps mitjà (espai i n<sup>o</sup> missatges és constant).</p> <p>-La xarxa estarà composta de 4 <i>peers</i>: dos reals i dos emuladors.</p> <p>-Les rèpliques es poden agrupar en dues categories: analitzar temps i tràfic amb tots els dispositius connectats i analitzar temps i tràfic quan un dispositiu es reconnecta a la xarxa.</p> <p>-En els experiments en què s'analitzi quan tarda un dispositiu a sincronitzar-se quan es reconnecta a la xarxa només s'analitzarà el tràfic del dispositiu analitzat.</p> <p>-En les rèpliques en que s'enviïn missatges consecutius, el temps es calcularà com la suma dels temps de cadascun dels missatges. És a dir, si enviem 3 missatge consecutius i anomenem II a l'instant en què es realitza el canvi i IF a l'instant en què el canvi s'ha propagat a tots els dispositius, el temps total seria:</p> $T = (IF1 - II1) + (IF2 - II2) + (IF3 - II3).$

Taula 156: Resum experiment 1

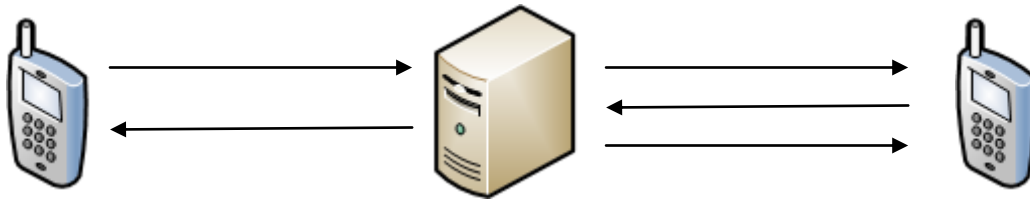
## 1.1 Estudi previ

En aquest experiment s'analitzarà el temps que es tarda a sincronitzar els dispositius i el tràfic generat (quantitat de missatges i mida d'aquests). Comencem per analitzar el tràfic que es generarà.

### Quantitat de missatges

Com ja hem comentat anteriorment, el nostre sistema GCM no és perfecte, es podrien realitzar algunes millores per optimitzar-lo i reduir la quantitat de

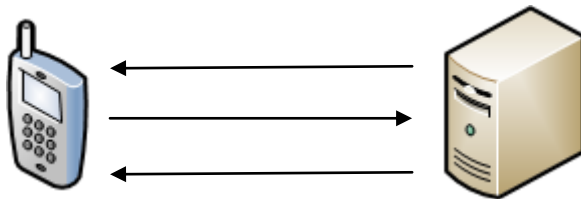
missatges. A la imatge següent es poden apreciar els missatges enviats quan un dispositiu fa un canvi que ha de rebre un altre dispositiu.



*Figura 141: Esquema enviar dades client-servidor*

És a dir, la quantitat de missatges enviats és de  $2 + 3 \cdot \text{nombre\_receptors}$ .

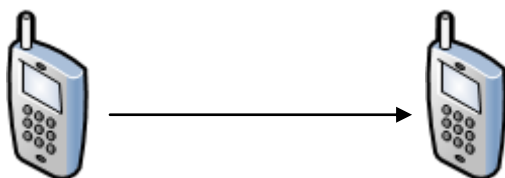
A les imatges següents es poden apreciar els missatges enviats quan un dispositiu es reconnecta a la xarxa i ha de rebre dades. Si un dispositiu es reconnecta a la xarxa i ha d'enviar dades el procediment és el de la imatge anterior. Si el dispositiu ha de rebre i enviar dades s'executen tots dos diagrames, de manera que la quantitat de missatges és la suma dels dos.



*Figura 142: Esquema demanar canvis client-servidor*

La quantitat de missatges rebuts del servidor depèn de la quantitat de canvis no simultanis fets. És a dir, si 3 usuaris han fet un canvi cadascun el dispositiu rebrà 3 missatges del servidor, però farà una única crida per demanar dades. És a dir, la quantitat de missatges per rebre dades és de  $2 + \text{nombre\_canvis\_no\_simultanis}$ .

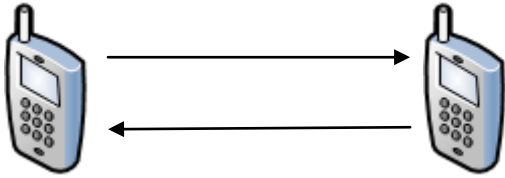
Anem ara amb la xarxa P2P. Quan un dispositiu fa un canvi que ha de rebre un únic dispositiu el procediment seguit és el següent:



*Figura 143: Esquema enviar dades P2P*

És a dir, la quantitat de missatges enviats és igual al nombre de dispositius que han de rebre els canvis.

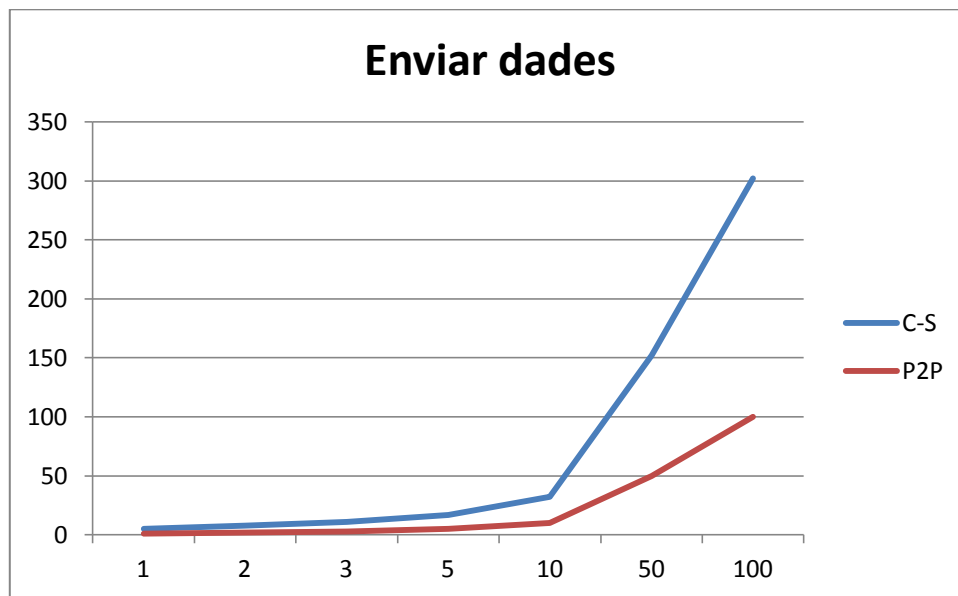
Si un dispositiu es reconnecta a la xarxa, però, haurà de demanar canvis a **tots** els dispositius que podrien haver fet algun canvi, és a dir, tots els dispositius dels usuaris que participen en algun dels seus calendaris. Si només hi hagués un possible dispositiu, el comportament seria el següent:



**Figura 144: Esquema demanar canvis P2P**

És a dir, la quantitat de missatges és de  $2 \cdot \text{dispositius\_participen\_algun\_calendari}$ . Com es pot preveure, en casos reals aquest nombre de missatges es dispararà.

Observem ara de manera gràfica com s'incrementa la quantitat de missatges en funció de certs paràmetres. Comencem per veure la quantitat de missatges en el cas d'enviar un sol canvi en funció del nombre de dispositius que l'han de rebre:



**Figura 145: Gràfica enviar dades**

Com es pot veure, el cas client-servidor sempre està per sobre del cas P2P. Anem ara a veure què passa quan un dispositiu es reconnecta a la xarxa. Comencem primer pel cas en què només s'hagin d'enviar dades. En el cas P2P, encara que no hi hagi hagut canvis el dispositiu seguirà demanant canvis a tots els dispositius que en puguin tenir algun. Per tal de simplificar la gràfica, la quantitat de dispositius que han de rebre els missatges és igual a la quantitat de dispositius als quals es demanaran canvis en P2P.

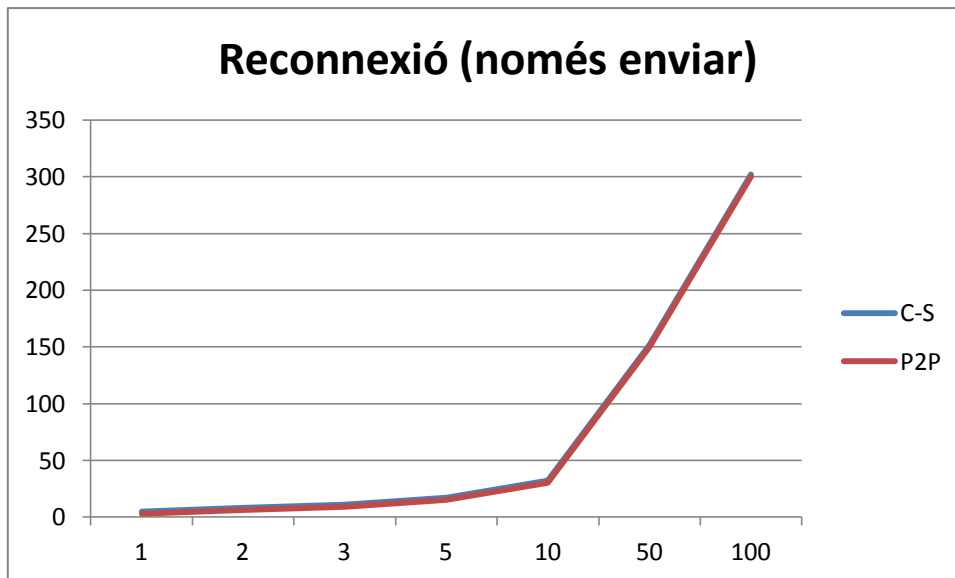


Figura 146: Gràfica reconexió només enviar

Tot i que P2P segueix estant sempre per sota de C-S, aquesta diferència és mínima (2 missatges). Això és així perquè hem igualat el nombre de destinataris amb el nombre de dispositius als que demanar canvis. El nombre de dispositius als que demanar canvis sempre serà **igual o més gran** que el nombre de destinataris, així que la gràfica anterior és el cas que afavoreix més a P2P. Si augmentéssim el nombre de dispositius als que demanar canvis la quantitat de missatges en P2P superaria a client-servidor.

Quan el dispositiu es reconnecta i només ha de rebre canvis la cosa es complica, ja que el nombre de missatges en client-servidor i en P2P depenen de variables diferents i no relacionades. És per aquest motiu que el fet de realitzar una gràfica perd sentit. Si el dispositiu ha d'enviar i rebre passa el mateix amb la diferència que llavors sí que hi ha una certa relació entre les fórmules, tot i que segueixen contenint variables diferents.

Finalment, cal parar especial atenció al cas concret en què un dispositiu es reconnecta a la xarxa i no hagi d'enviar ni rebre res. En el cas de client-servidor el dispositiu no farà res, mentre que en el cas de P2P seguirà preguntant a tots els dispositius si hi ha canvis nous, i aquests ho hauran de comprovar. És a dir, si el dispositiu s'està desconnectant i connectant constantment, la xarxa P2P s'omplirà de missatges demanant canvis que no són necessaris, i els dispositius invertiran temps i recursos en comprovar si hi ha canvis. Al gràfic següent es pot observar la quantitat de missatges enviats en funció del nombre de dispositius que cal consultar.



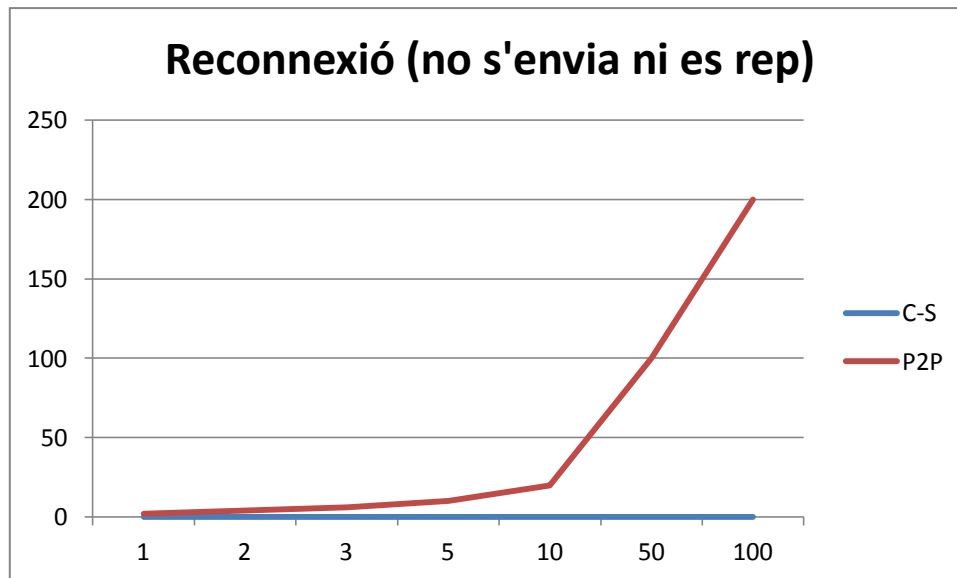


Figura 147: Gràfica reconeixió no s'envia ni es rep

Com es pot observar, a mesura que augmenten els dispositius a consultar anem saturant la xarxa inútilment.

### Mida dels missatges

En quant a la mida dels missatges, els missatges rebuts per part del servidor únicament contenen el temps en Unix de la realització del canvi, de manera que es poden ignorar al fer l'estimació de la mida total. Els missatges per demanar canvis de totes dues xarxes tindran una mida semblant i bastant petita: en el cas client-servidor contenen el token i la data, mentre que en el cas P2P contenen la data i l'adreça de l'emissor. Tot i que aquests dos missatges no seran exactament iguals, per fer l'estimació suposarem que sí.

Això ens deixa amb els missatges que sí que porten dades: els que envien canvis, les respostes de demanar canvis i la resposta d'enviar canvis (en el cas client-servidor). Òbviament, el volum d'aquests missatges depèn completament dels canvis fets, però sí que es pot establir una relació entre el volum dels missatges en client-servidor i el volum en P2P.

El volum de la resposta d'enviar canvis serà similar al missatge d'enviar canvis que provoca la resposta, ja que ens retornarà els canvis que s'han desat al servidor. En el cas d'enviar canvis (imatges 1 i 3), tot i que els destinataris rebran els mateixos canvis, els missatges de client-servidor seran majors ja que retornaran tot un conjunt de camps buits que ocupen un cert espai (els destinataris demanen **tots** els canvis ja que no saben què s'ha modificat). En els casos que tractarem en aquest experiment **tots els dispositius rebran tots els canvis**, de manera que podem assumir que el missatge enviat per l'emissor en l'arquitectura client-servidor tindrà un volum inferior que cadascun dels missatges rebuts pels destinataris, ja que com ja hem comentat aquests últims tindran uns certs camps afegits a més dels canvis enviats.

Pel què fa a la resposta de demanar canvis de P2P, en els casos tractats en aquest experiment tots els dispositius demanats disposaran dels canvis realitzats, de manera que tots els dispositius consultats li retornaran el mateix conjunt de canvis.

## **Temps**

Veient la quantitat de missatges enviats i rebuts es pot fer una estimació de la diferència de temps entre totes dues xarxes en els diferents casos. És difícil fer una estimació precisa, ja que s'haurien de realitzar moltes suposicions i el temps necessari per enviar un missatge d'un dispositiu a un altre o al servidor depèn de moltes variables incertes. El que sí que podem analitzar són les tasques que es realitzen en els dispositius mòbils i en el servidor.

En el cas de client-servidor, quan es realitza un canvi en un dispositiu (imatge 1) aquest és enviat ràpidament al servidor, que el desa i notifica els destinataris. Les parts que requereixen més temps en aquest procés són l'enviament dels missatges i l'operació de recuperar les dades del servidor, que ha d'analitzar totes les dades referents a l'usuari que fa la petició per comprovar si hi ha hagut canvis. En el cas d'enviar canvis en P2P (imatge 3), el dispositiu emissor ha de buscar les adreces dels destinataris en el seu llistat de *peers* i, un cop trobades, enviar-los-hi els canvis necessaris. Tot i que aquesta cerca d'adreces té un cert cost temporal, podem suposar que el cost en client-servidor serà major, ja que s'envien més missatges i s'ha de realitzar una cerca de canvis fets en tota la informació dels usuaris. Es podrien donar casos extrems en què el P2P arribés a ser més lent: el cas client-servidor és el màxim de ràpid (l'usuari receptor gairebé no té dades per comprovar) i el cas P2P té una quantitat molt elevada d'adreces per comprovar (tot i que només s'hagués d'enviar el missatge a un dispositiu). No obstant, aquests casos són molt poc probables i no els tractarem en aquest experiment.

Passem ara al cas en què un dispositiu es reconnecta a la xarxa. Abans de res, cal tenir en compte que en aquest experiment ens interessa saber el temps que tarda el dispositiu a estar sincronitzat, no el temps que es tarda a acabar l'operació. Per exemple, si un dispositiu demana canvis a un conjunt de dispositius i tots li han de retornar el mateix, un cop hagi tractat la primera resposta el dispositiu es considerarà sincronitzat, encara que després segueixi rebent respostes que no provocaran cap canvi.

Observem primer què passa si el dispositiu només ha de rebre dades, no ha d'enviar res. En el cas client-servidor el dispositiu farà una petició al servidor, que comprovarà totes les dades de l'usuari cercant si hi ha hagut algun canvi i els hi retornarà. Com ja hem comentat abans, aquesta operació de cercar canvis és bastant costosa. En el cas P2P el dispositiu primer de tot cerca el llistat d'usuaris als quals ha de demanar canvis, a continuació tradueix aquests usuaris a adreces i els envia una petició de canvis. Aquests dispositius revisaran les seves dades i retornaran els canvis que hi hagi hagut. Com ja hem dit, ens interessa saber quan tarda el dispositiu a estar sincronitzat, que serà quan es rebi la primera resposta i es tracti. En aquest cas, per fer una estimació del cas P2P hi intervenen diverses variables, com la quantitat d'adreces del dispositiu, la quantitat de dispositius als quals s'han de demanar canvis, la quantitat d'informació emmagatzemada en els dispositius, si tots els dispositius retornen els mateixos canvis, etc. El nombre d'adreces i el nombre d'usuaris als que demanar canvis influeixen en el temps del

cas P2P, però no en el cas de client-servidor. En casos en què hi hagi poques adreces i pocs usuaris, com passarà en aquest experiment, podem suposar que el cas P2P serà més ràpid que el cas client-servidor, ja que les consultes als dispositius són més ràpides que al servidor. A mesura que el nombre d'usuaris i adreces vagi augmentant, però, aquesta diferència anirà disminuint fins a arribar a un punt en què el cas client-servidor serà més ràpid. Per poder trobar més o menys en quin punt el temps és igual seria recomanable experimentar augmentant el nombre de dispositius reals (no emuladors), fet que no és senzill i que no s'ha pogut tractar.

Tractem ara el cas contrari a l'anterior, el dispositiu es reconnecta i ha d'enviar dades, però no rebre'n. Inicialment pot semblar que aquest cas és exactament igual al d'enviar canvis quan tots els dispositius estan connectats, però hi ha una petita diferència: en P2P, quan el dispositiu es reconnecta **sempre** demana canvis, ja que no sap si n'hi ha hagut o no. En client-servidor, en canvi, només es demanen si es rep un missatge del servidor. Per tant, aquest cas serà igual que el primer cas tractat en aquest apartat però afegint un cost temporal al cas de P2P, que en el cas esmentat era més ràpid. És difícil fer una suposició de quin dels dos serà més ràpid en aquest cas ja que hi intervenen massa variables. Sí que podem suposar que en casos en què hi hagi bastants usuaris el cas client-servidor serà més ràpid, ja que aquest increment afegit al cas P2P serà bastant notable, però no podem saber amb exactitud com afectarà als casos amb pocs usuaris com els del nostre experiment. Fins que no es realitzi l'experiment no es podrà sortir de dubtes, però és probable que aquest *handicap* hagi invertit la tendència que havia predominat fins ara.

Òbviament, el següent cas a tractar és quan el dispositiu es reconnecta i ha d'enviar i rebre dades. Com ja hem dit, la nostra previsió és que rebre dades sigui més ràpid en P2P i que enviar-ne, tot i no ser massa clar, suposem que serà més ràpid en client-servidor. Així doncs, si combinem les dues operacions quina arquitectura serà més ràpida? L'operació d'enviar dades era més ràpida en client-servidor perquè en P2P tenim el *handicap* de demanar dades, però en aquest cas l'arquitectura client-servidor també n'ha de demanar, i com s'ha comentat anteriorment aquestes dues operacions per separat són més ràpides en P2P. Així doncs, suposem que en aquest cas l'arquitectura més ràpida serà P2P.

### Previsió final de l'estudi

Un cop reunida tota la informació presentada en aquest estudi previ, podem resumir els resultats esperats de l'experiment en la taula següent:

Rèplica	Experiment	Client-servidor	P2P
1	Enviar una relació amb tots els dispositius connectats (tots els dispositius l'han de rebre).	Més lent 11 missatges Mida $\approx 2X$	Més ràpid 3 missatges Mida = X
2	Enviar 4 relacions en un mateix missatge amb tots els dispositius connectats.	Més lent 11 missatges $X < \text{Mida} < 2X$	Més ràpid 3 missatges Mida = X

3	Enviar 3 relacions en 3 missatges consecutius amb tots els dispositius connectats.	Més lent 33 missatges Mida $\approx 2X$	Més ràpid 9 missatges Mida = X
4	Dispositiu es reconnecta a la xarxa i rep 4 relacions modificades simultàniament, no envia res.	Més lent 3 missatges Mida = X	Més ràpid 6 missatges $2X < \text{Mida} < 3X$
5	Dispositiu es reconnecta a la xarxa i rep 4 relacions modificades una a una, no envia res.	Més lent 6 missatges Mida = X	Més ràpid 6 missatges $2X < \text{Mida} < 3X$
6	Dispositiu es reconnecta a la xarxa i envia 4 relacions, no rep res.	Més ràpid 2 missatges Mida = X	Més lent 6 missatges Mida $\approx 2x$
7	Dispositiu es reconnecta a la xarxa, envia 4 relacions i en rep 4 també.	Més lent 5 missatges Mida = X	Més ràpid 9 missatges $2X < \text{Mida} < 3X$

Taula 157: Estudi previ experiment 1

## 1.2 Experimentació

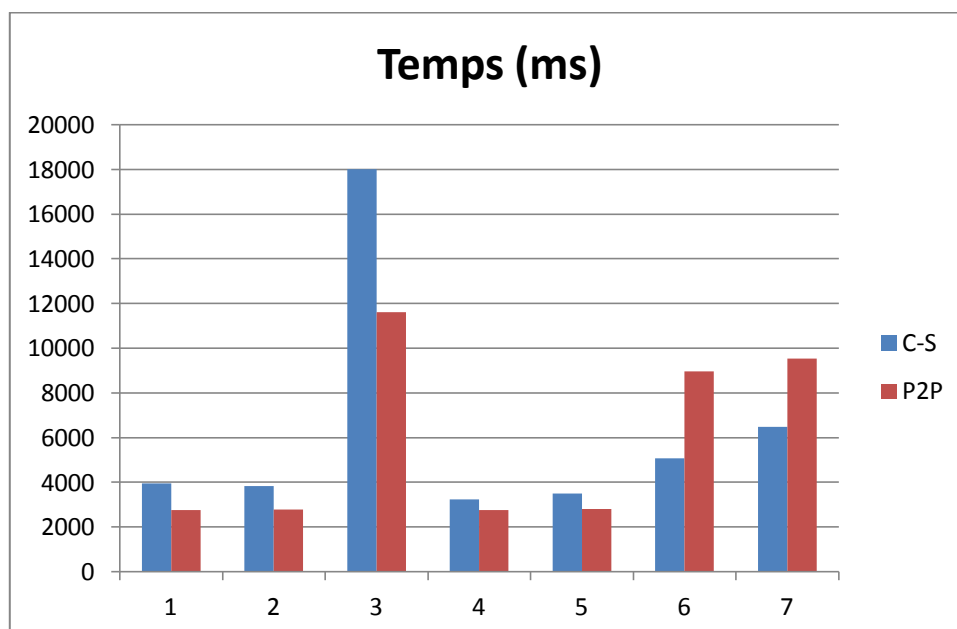
Un cop realitzat l'experiment, recol·lectem la informació que volem utilitzar per provar la nostra hipòtesi. A la taula següent es poden observar els resultats d'un experiment com el que hem definit.

Rèplica	Experiment	Client-servidor	P2P
1	Enviar una relació amb tots els dispositius connectats (tots els dispositius l'han de rebre).	3.938,5 ms 11 missatges 2.185 bytes	2.750,8 ms 3 missatges 768 bytes
2	Enviar 4 relacions en un mateix missatge amb tots els dispositius connectats.	3.830,3 ms 11 missatges 4.192 bytes	2.777,7 ms 3 missatges 2.316 bytes
3	Enviar 3 relacions en 3 missatges consecutius amb tots els dispositius connectats.	17.993,5 ms 33 missatges 7.893 bytes	11.609,2 ms 9 missatges 2.304 bytes

4	Dispositiu es reconnecta a la xarxa i rep 4 relacions modificades simultàniament, no envia res.	3.242,3 ms 3 missatges 1.265 bytes	2.766,1 ms 6 missatges 2.478 bytes
5	Dispositiu es reconnecta a la xarxa i rep 4 relacions modificades una a una, no envia res.	3.497,4 ms 6 missatges 1.295 bytes	2.811,2 ms 6 missatges 2.478 bytes
6	Dispositiu es reconnecta a la xarxa i envia 4 relacions, no rep res.	5.059,3 ms 2 missatges 787 bytes	8.952,9 ms 6 missatges 2.502 bytes
7	Dispositiu es reconnecta a la xarxa, envia 4 relacions i en rep 4 també.	6.487,2 ms 5 missatges 2.725 bytes	9.538,6 bytes 9 missatges 7.074 bytes

**Taula 158: Resultats experiment 1**

A la següent gràfica podem observar de manera més visual la diferència en temps:



**Figura 148: Gràfic comparació temps experiment 1**

Procedim ara a comparar els resultats obtinguts amb la previsió feta a l'estudi previ. La quantitat de missatges, com era d'esperar, no varia, ja que els càlculs eren exactes. Sí que trobem algunes variacions en la mida dels missatges. No són variacions massa importants respecte a la previsió, i són degudes a que al fer l'estimació s'havien de fer algunes suposicions respecte la relació entre les mides d'aquests missatges, de manera que els càlculs no eren tot el precisos que podien ser.

En quant al temps, la previsió s'ha encertat excepte en un cas: l'última rèplica, quan el dispositiu ha d'enviar i rebre relacions. A la previsió s'ha comparat les dues operacions (enviar i rebre) per separat i s'ha arribat a la conclusió que P2P seria més ràpid ja que les dues operacions per separat eren més ràpides. No s'ha tingut en compte, però, la limitació de recursos dels dispositius mòbils. En l'arquitectura client-servidor una bona part de les operacions les realitza el servidor, que tot i realitzar tasques més lentes és més ràpid i disposa de més recursos que no pas els *smartphones*. En el cas P2P, però, totes les operacions les realitzen els dispositius, i això és el que provoca que aquesta arquitectura sigui més lenta. Fixem-nos en els dispositius que estan sempre connectats a la xarxa: per una banda reben un seguit de canvis que han de desar, i per l'altra han de consultar la seva base de dades cercant altres canvis fets i els han de retornar a l'emissor. Els dispositius no tenen massa problemes realitzant aquestes operacions per separat, però com s'ha demostrat en aquest experiment quan han de realitzar totes dues coses el seu rendiment baixa, provocant que la xarxa tardi més a estar sincronitzada. Aquest fet no s'havia contemplat en l'estudi previ.

La hipòtesi H0 és que totes les arquitectures són iguals. Amb aquest experiment s'ha comprovat que no és així, sinó que hi ha arquitectures millors o pitjors en certs casos.

Podem concloure que l'arquitectura P2P és més ràpida i utilitza menys tràfic quan tots els dispositius estan sempre connectats, però quan els dispositius comencen a desconnectar-se i reconnectar-se l'arquitectura client-servidor és més eficient en temps i, especialment, en tràfic. Com que en la utilització de dispositius mòbils és habitual que els dispositius es desconnectin freqüentment degut al seu moviment i al fet que alguns usuaris poden no tenir connexió 3G, l'arquitectura client-servidor és més adequada pel cas que estem tractant que no pas l'arquitectura P2P definida en aquest experiment.

## 2. Relationships vs Modules

Com ja s'ha comentat a l'apartat d'implementació, mentre es desenvolupava el projecte va aparèixer la necessitat d'escollir entre *relationships* o mòduls per representar les associacions a *SugarCRM*. La decisió presa va ser mòduls, però en aquest experiment es compararan en detall els avantatges i inconvenients mesurables (eficiència i espai) de totes dues versions.

Per tal de comparar-los es generaran els mateixos models utilitzant mòduls i *relationships* i es realitzaran diverses consultes i insercions al servidor. Es realitzaran consultes de dos tipus: sense tenir en compte data de modificació (sincronització completa) i tenint-la en compte (sincronització parcial). Com ja s'ha comentat també a l'apartat d'implementació, *SugarCRM* no ofereix cap funcionalitat que permeti filtrar les *relationships* obtingudes en funció de la seva data de modificació, de manera que per aquest experiment s'ha modificat la funció *get\_modified\_relationships* per tal que compleixi aquest objectiu. També s'ha implementat la funció *get\_info* (obté tota la informació relacionada amb l'usuari, utilitzada durant la sincronització) utilitzant *relationships* enlloc de mòduls per representar les associacions.

Podem resumir les característiques de l'experiment amb la taula següent:

<b>Observació</b>	Utilitzar mòduls o <i>relationships</i> per representar associacions a <i>SugarCRM</i> no implica el mateix cost de temps ni d'espai.
<b>Plantejament</b>	Generem els mateixos models utilitzant <i>relationships</i> i mòduls i comparem els resultats obtinguts.
<b>Hipòtesis</b>	Utilitzar mòduls o <i>relationships</i> té el mateix cost en temps i espai (H0) o un dels dos és millor que l'altre.
<b>Mètode</b>	<ul style="list-style-type: none"><li>-Executarem un total de 11 rèpliques.</li><li>-Cada rèplica l'executarem deu vegades i anotarem el temps mitjà (l'espai és constant).</li><li>-Dins de cada rèplica utilitzarem les dues metodologies sobre el mateix model.</li><li>-Per tal que els resultats siguin més correctes, dins de cada rèplica s'executaran alternativament les execucions de les dues metodologies una a una. És a dir, s'executarà la repetició 1 amb mòduls, després la repetició 1 amb <i>relationships</i>, tot seguit la repetició 2 amb mòduls, etc. Això evita els casos en què coincidís que la càrrega de treball del servidor augmentés o disminuís tot just acabar una de les metodologies, de manera que el resultat de la segona metodologia es veuria influït per aquest augment/disminució de la càrrega de treball.</li></ul>

	<p>-La comparació de resultats s'ha de fer entre metodologies dins d'una mateixa rèplica, no es poden comparar resultats de rèpliques diferents ja que algunes rèpliques es poden realitzar en dies diferents i la càrrega del servidor podria variar. L'únic cas en què es poden comparar els resultats de dues rèpliques diferents és en el cas de les rèpliques 5 i 6, que s'executaran alternativament.</p> <p>-En les rèpliques d'introduir informació (8, 9, 10 i 11) es comptabilitzaran dos temps diferents en el cas de <i>relationships</i>: comprovant si la relació existeix abans d'introduir-la i introduint la relació sense comprovacions prèvies. SugarCRM gestiona les <i>relationships</i> i evita repeticions si s'envia una relació que ja existeix, de manera que les comprovacions no són necessàries, però es realitzaran tots dos casos per poder comparar la diferència temporal. El temps <b>amb</b> comprovació prèvia serà sempre el de <b>l'esquerra</b>.</p>
--	---

Taula 159: Resum experiment 2

## 2.1 Estudi previ

En aquest experiment s'analitzarà el temps que es tarda a obtenir o inserir dades i l'espai que cada model ocupa en cadascun dels casos. Comencem pel més senzill: l'espai.

### Espai

Calcular l'espai és bastant senzill, ja que totes les *relationships* i els mòduls que representin relacions ocuparan el mateix espai. Les *relationships* ocupen 128 bytes, mentre que els mòduls dedicats a relacions n'ocupen 219. Així doncs, calcular l'espai necessari per emmagatzemar les dades específiques de cada model de l'experiment serà senzill. Òbviament, el cas amb *relationships* sempre ocuparà menys espai que el cas amb mòduls.

### Temps

Analitzar el temps no és tan senzill, però podem realitzar una estimació dels diferents casos. Quan volem obtenir una sola relació l'operació és més ràpida amb *relationships*, ja que contenen un índex a la base de dades per tal que sigui senzill de buscar-les a partir d'un dels identificadors dels mòduls que relaciona. Quan utilitzem mòduls, en canvi, els identificadors es desen en atributs afegits per l'administrador, de manera que aquests atributs no contenen cap índex i la seva cerca a la base de dades té un cost temporal major.

No obstant, a mesura que les consultes es van complicant ens trobem amb certes limitacions a les *relationships*, i és que estan pensades per realitzar consultes u-a-molts. Per exemple, obtenir totes les tasques en les que participa un usuari. Si en realitat el que volem és obtenir totes les tasques en què participen un conjunt d'usuaris, haurem de fer la crida 1-a-molts n vegades, on n és el nombre d'usuaris. Aquesta iteració d'usuaris suposa un increment de cost que, tot i que en consultes



senzilles no és massa elevat, a mesura que el volum d'informació demanat vagi creixent suposarà un augment de temps considerable. Amb mòduls, en canvi, podem utilitzar una sentència *where* per tal d'aconseguir totes les relacions demanades amb una sola consulta.

Pel què fa a les insercions ens torbem que en el cas de mòduls hem d'evitar nosaltres les entrades repetides, mentre que amb *relationships* el propi SugarCRM s'encarrega d'evitar que inserim la mateixa relació dues vegades. Arribats a aquest punt haurem de realitzar algunes suposicions, ja que desconeixem exactament com funciona aquesta comprovació de Sugar. El més probable és que el mecanisme utilitzat sigui un dels dos següents: abans d'inserir es fa una cerca per comprovar si existeix (com fem nosaltres amb els mòduls) o bé s'intenta fer una inserció i si es viola la restricció de clau primària es genera una excepció i es cancel·la la inserció.

Suposem primer que ens trobem en el primer cas, Sugar realitza una comprovació prèvia. Com ja hem comentat, l'obtenció d'una única entrada és més ràpida amb *relationships* que amb mòduls, de manera que la comprovació serà més ràpida amb *relationships*. Desconeixem si la inserció en sí té alguna diferència de temps, així que suposarem que és igual en tots dos casos. Per tant, basant-nos en aquesta suposició podem assumir que el cas de *relationships* sempre serà més ràpid que el cas amb mòduls pel què fa a les insercions.

Suposem ara que Sugar utilitza la restricció de clau primària per evitar les duplicitats. Com he comprovat personalment amb projectes anteriors, utilitzar aquesta opció ofereix una major eficiència si l'entrada no està repetida ja que ens estalviem la comprovació, però si l'entrada està repetida el cost temporal afegit que suposa la violació de la restricció és superior al cost de si tinguéssim una comprovació prèvia. Així doncs, suposant que la inserció en sí (sense comprovacions ni violacions de restriccions) té el mateix cost temporal en tots dos casos, el fet d'inserir una entrada que no existeix seria més ràpid amb *relationships*, però si l'entrada existeix seria més ràpid amb mòduls.

### Previsió final de l'estudi

Un cop reunida tota la informació presentada en aquest estudi previ, podem resumir els resultats esperats de l'experiment en la taula següent.

L'espai inclòs a la taula és el necessari per a emmagatzemar les associacions utilitzades en cada rèplica que s'estan observant, s'ha ignorat el de la resta de mòduls ja que és el mateix en tots dos casos.

A les rèpliques d'inserció d'entrades repetides hi apareixen dues opcions al temps. Una d'elles és suposat que el mecanisme de *relationships* tingui una comprovació prèvia (COMPR), l'altra és suposant que s'utilitzi la clau primària per evitar repetits (EXC).

Rèplica	Operació	Mòduls	<i>Relationships</i>
1	Obtenir participants 1 tasca (5 participants)	Més lent 1.095 bytes	Més ràpid 640 bytes

2	Obtenir <u>nous</u> participants 1 tasca (5 participants, 3 de nous)	Més lent 1.095 bytes	Més ràpid 640 bytes
3	Obtenir participants tasques que l'usuari hi participa (7 tasques, 15 parts)	Més lent 3.285 bytes	Més ràpid 1.920 bytes
4	Obtenir <u>nous</u> participants i tots part. <u>noves</u> tasques que l'usuari hi participa (2 noves tasques amb 3 participants, 3 nous parts tasques no noves)	Més lent 3.285 bytes	Més ràpid 1.920 bytes
5	Obtenir tota la informació (usuari dpalou)	Més ràpid 46.428 bytes	Més lent 27.136 bytes
6	Obtenir tota la <u>nova</u> informació (usuari dpalou)	Més ràpid 46.428 bytes	Més lent 27.136 bytes
7	Obtenir tota la informació (usuari admin)	Més ràpid o igual 46.428 bytes	Més lent o igual 27.136 bytes
8	Introduir 1 nou <i>Participa</i>	Més lent 219 bytes	Més ràpid 128 bytes
9	Introduir 1 nou <i>Rep</i>	Més lent 219 bytes	Més ràpid 128 bytes
10	Introduir 1 <i>Rep</i> repetit	Si COMPR: Més lent Si EXC: Més ràpid 219 bytes	Si COMPR: Més ràpid Si EXC: Més lent 128 bytes
11	Introduir 6 <i>Participa</i> a 3 tasques diferents (3 repetits, 3 nous)	Si COMPR: Més lent Si EXC: Igual? 1314 bytes	Si COMPR: Més ràpid Si EXC: Igual? 768 bytes

**Taula 160: Estudi previ experiment 2**

En l'última rèplica si el mecanisme de *relationships* utilitza la clau primària per evitar repetits és difícil fer una previsió de quin dels dos mètodes tardarà més, ja que caldria establir una relació entre la diferència de temps de les insercions que no existeixen amb la diferència de temps de les insercions que ja existeixen. És a dir, sabem que inserir una relació que no existeix és més ràpid amb *relationships*, però no sabem en quina quantitat, i el mateix passa amb les relacions que existeixen però a la inversa. Com que no sabem la quantitat d'aquestes diferències no podem saber si es compensen entre elles o si la balança es decanta cap a alguna de les dues opcions. És per aquest motiu que s'ha assumit que les diferències es compensen, però és purament una especulació, no té cap base lògica, per això s'ha introduït un interrogant en la previsió.

## 2.2 Experimentació

Un cop realitzat l'experiment, recol·lectem la informació que volem utilitzar per provar la nostra hipòtesi i comparar les dues metodologies. A la taula següent es poden observar els resultats d'un experiment com el que hem definit.

L'espai inclòs a la taula és el necessari per a emmagatzemar les associacions utilitzades en cada rèplica que s'estan observant, s'ha ignorat el de la resta de mòduls ja que és el mateix en tots dos casos.

Rèplica	Operació	Mòduls	Relationships
1	Obtenir participants 1 tasca (5 participants)	159,224 ms 1.095 bytes	93,656 ms 640 bytes
2	Obtenir <u>nous</u> participants 1 tasca (5 participants, 3 de nous)	108,1 ms 1.095 bytes	32,535 ms 640 bytes
3	Obtenir participants tasques que l'usuari hi participa (7 tasques, 15 parts)	565,342 ms 3.285 bytes	308,228 ms 1.920 bytes
4	Obtenir <u>nous</u> participants i tots part. <u>noves</u> tasques que l'usuari hi participa (2 noves tasques amb 3 participants, 3 nous parts tasques no noves)	270,25 ms 3.285 bytes	209,993 ms 1.920 bytes
5	Obtenir tota la informació (usuari dpalou)	1,526 segons 46.428 bytes	2,213 segons 27.136 bytes
6	Obtenir tota la <u>nova</u> informació (usuari dpalou)	1,578 segons 46.428 bytes	1,922 segons 27.136 bytes
7	Obtenir tota la informació (usuari admin)	140,537 ms 46.428 bytes	142,236 ms 27.136 bytes
8	Introduir 1 nou <i>Participa</i>	84,47 ms 219 bytes	107,423   80,826 ms 128 bytes
9	Introduir 1 nou <i>Rep</i>	94,307 ms 219 bytes	116,855   84,052 ms 128 bytes
10	Introduir 1 <i>Rep</i> repetit	36,351 ms 219 bytes	60,075   79,674 ms 128 bytes
11	Introduir 6 <i>Participa</i> a 3 tasques diferents (3 repetits, 3 nous)	222,762 ms 1314 bytes	284,585   231,746 ms 768 bytes

Taula 161: Resultats experiment 2

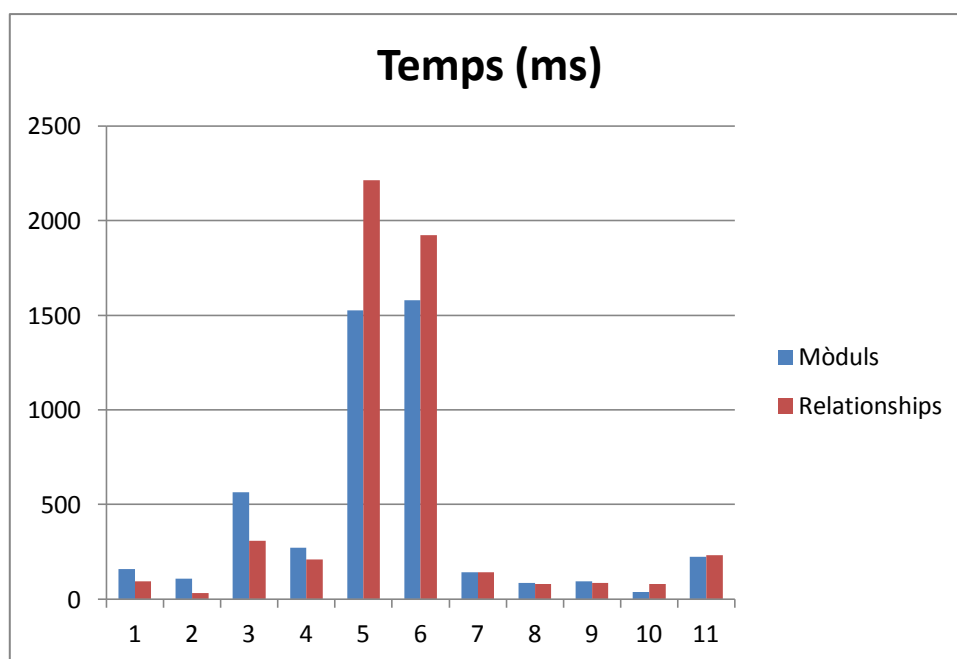
Les rèpliques 5, 6 i 7 s'han aplicat sobre el model següent: 5 usuaris, 5 grups, 11 *Pertany*, 1 sol·licitud, 5 *Comparteix*, 7 calendaris, 4 Té, 8 tasques, 0 *Precedeix*, 3 *Assisteix*, 15 *Participa*, 143 esdeveniments, 177 *Rep*, 0 resolucions pendents, 0 vots i 0 respostes.

A la **rèplica 5** es recupera la següent informació: 5 grups, 11 *Pertany*, 1 sol·licitud, 5 *Comparteix*, 4 usuaris, 4 calendaris, 8 tasques, 0 *Precedeix*, 3 *Assisteix*, 15 *Participa*, 51 esdeveniments, 0 resolucions pendents, 0 vots i 0 respostes.

A la **rèplica 6** es recupera la següent informació: 1 grup, 4 *Pertany*, 0 sol·licituds, 1 *Comparteix*, 3 usuaris, 2 calendaris, 3 tasques, 0 *Precedeix*, 2 *Assisteix*, 7 *Participa*, 2 esdeveniments, 0 resolucions pendents, 0 vots i 0 respostes.

A la **rèplica 7** es recupera la següent informació: 0 grups, 0 *Pertany*, 0 sol·licituds, 0 *Comparteix*, 1 usuari, 0 calendaris, 0 tasques, 0 *Precedeix*, 0 *Assisteix*, 0 *Participa*, 0 esdeveniments, 0 resolucions pendents, 0 vots i 0 respostes.

A la següent gràfica podem observar de manera més visual la diferència en temps. En el cas de *relationships* de les rèpliques d'insercions s'ha utilitzat el valor de la dreta, sense comprovacions prèvies manuals, ja que és el quin ha donat millor resultat en general.



**Figura 149: Gràfic comparació temps experiment 2**

A l'annex 1 d'aquesta memòria es poden observar amb detall els temps de les rèpliques 5, 6 i 7, de manera que es poden comparar els temps d'execució de les diferents parts de la funció per obtenir tota la informació, permetent-nos veure quines són les parts en què tarda més una metodologia o una altra.

Un fet que pot semblar xocant a simple vista és que la rèplica 6 (obtenir nova informació) amb mòduls tarda més que la rèplica 5 (obtenir tota la informació). Si observem la taula de l'annex que ofereix els resultats d'aquestes rèpliques en detall, veiem que en algunes parts la rèplica 5 és més lenta i en d'altres ho és la rèplica 6, però si es sumen totes les parts el resultat és que la rèplica 6 és més

ràpida que la rèplica 5. Així doncs, per què el resultat final és al contrari? Això es deu principalment a que, en cas de voler obtenir la nova informació, en cert punt hem d'agafar els identificadors de les tasques en què l'usuari és un nou participant per tal d'agafar-ne totes les resolucions pendents, mentre que quan volem obtenir tota la informació no fem distinció entre tasques noves o no, així que ens podem estalviar aquesta consulta. Aquesta operació es troba a la funció principal i no està comptabilitzada en les diferents parts de la taula de l'annex.

Com es pot veure, la previsió feta a l'estudi previ ha estat bastant encertada. En els casos senzills (rèplica 1 a 4) les *relationships* són més eficients, però quan les consultes es compliquen (rèpliques 5 a 7) els mòduls ens ofereixen un millor rendiment. Pel què fa a les insercions, observem que el resultat obtingut encaixa amb la suposició que el mecanisme de *relationships* utilitza la restricció de clau primària per evitar duplicitats. Això no vol dir que puguem assegurar que aquest és el mecanisme que utilitza, podria ser que utilitzés un tercer mecanisme no contemplat a l'estudi els resultats del qual coincidissin amb el cas presentat.

La hipòtesi H0 és que totes dues implementacions tenen el mateix cost en temps i espai. Amb aquest experiment s'ha comprovat que no és així, sinó que en alguns casos és millor utilitzar *relationships* i en altres casos és millor utilitzar mòduls.

Podem concloure que utilitzar *relationships* és millor en casos senzills com els que solen ser habituals en SugarCRM, mentre que en casos més complicats com és la principal consulta de SweetSyncal és més eficient temporalment utilitzar mòduls. És cert que en tots dos casos el cost espacial és major amb mòduls, però com que no tenim problemes d'espai al servidor amb les mides amb les que estem treballant això no suposa un inconvenient.

SweetSyncal combina operacions senzilles amb d'altres de més complicades, de manera que per saber exactament quina de les dues opcions és més favorable temporalment caldria fer un estudi d'una implementació real per tal d'analitzar la quantitat d'operacions de cada tipus que es realitzen. No obstant, cal tenir en compte que, per cada inserció, tots els usuaris involucrats faran una consulta de tota la informació modificada, de manera que podem intuir que probablement la quantitat de consultes de tota la informació o d'informació modificada serà major o igual a la quantitat d'operacions senzilles. Si sumem això als avantatges d'utilitzar mòduls plantejats a l'apartat d'implementació la conclusió és que **ens convé utilitzar mòduls.**

### 3. Bateria: SyncAdapter vs GCM

A l'aplicació Sweet d'en Vicent Seguí, projecte al qual estem donant continuïtat, s'utilitza un SyncAdapter per tal de sincronitzar els contactes amb el servidor. Aquest SyncAdapter s'executa cada 5 minuts. No obstant, a la nostra aplicació ens interessava disposar d'un temps de sincronització menor, ja que la sincronització de tasques hauria de ser bastant ràpida (en aquest experiment el SyncAdapter s'executarà cada 30 segons). La solució de consultar cada x segons si hi ha canvis al servidor pot ser acceptable en un ordinador, però en un telèfon mòbil suposa un consum de recursos, especialment de bateria, excessiu.

És per casos així que Google va implementar el seu servei *Google Cloud Messaging* que ens permet enviar missatges als dispositius mòbils des del nostre servidor. Utilitzant aquest servei només enviarem missatges al servidor des dels telèfons quan sigui absolutament necessari, és a dir, quan l'usuari ho indiqui o quan el servidor ens avisi que hi ha noves dades. Això implica un estalvi de recursos important en la majoria dels casos, i seria interessant poder mesurar aquest estalvi expressat com un estalvi de bateria.

En aquest experiment es compararan les dues alternatives per mesurar l'estalvi de bateria en diferents situacions. Per tant, també implementarem un *SyncAdapter* que demanarà i enviarà canvis al servidor cada 30 segons.

Podem resumir les característiques de l'experiment amb la taula següent:

<b>Observació</b>	Hi ha mètodes de sincronització que consumeixen més bateria que d'altres en certs casos.
<b>Plantejament</b>	Implementem la sincronització de dues maneres diferents i observem el consum de cadascun.
<b>Hipòtesis</b>	Tots dos mètodes de sincronització consumeixen la mateixa bateria (H0) o un és millors que l'altre en funció del cas.
<b>Mètode</b>	<ul style="list-style-type: none"><li>-Executarem un total de 12 rèpliques.</li><li>-El SyncAdapter s'executarà cada 30 segons.</li><li>-S'utilitzarà l'aplicació GSam Battery Monitor per mesurar el consum de bateria.</li><li>-Cada rèplica s'executarà durant una hora, durant la qual s'analitzarà el consum de bateria total del telèfon i el consum de bateria específic de la nostra aplicació. D'aquesta manera podem calcular la quantitat de bateria total que ha consumit la nostra aplicació durant aquesta hora.</li></ul>

Taula 162: Resum experiment 3

### 3.1 Estudi previ

Primerament s'analitzarà cadascun dels casos per tal de fer una estimació del consum de bateria en diferents situacions. Podríem dividir les situacions a què es veurà sotmesa l'aplicació Android en quatre tipus diferents: no hi ha cap canvi, només es reben canvis, només s'envien canvis o s'envien i es reben canvis. Els dos mètodes que seran comparats s'anomenaran a partir d'ara GCM (Google Cloud Messaging) i SA (SyncAdapter).

Cal tenir en compte que el "manteniment" de cadascuna d'aquestes opcions té un cost base del qual s'encarrega el sistema Android. És a dir, el sistema operatiu s'encarregarà automàticament d'executar el SyncAdapter cada cert temps, fet que durà un consum de recursos associat. El mateix passarà amb l'opció GCM, on s'utilitzaran *Receivers*. Aquest consum serà constant en el cas SA, ja que el nombre d'execucions al llarg del temps és constant, però variarà en el cas GCM on a priori no podem saber el nombre de vegades que s'executaran els procediments. Aquests costos bases s'anomenaran  $CB_{SA}$  i  $CB_{GCM}$ .

L'operació de sincronització es pot dividir en dues parts: agafar els canvis del servidor i enviar canvis al servidor. En GCM s'executen per separat i només si és necessari, mentre que en SA sempre s'executaran totes dues. Per fer una estimació del consum de bateria, però, cal entrar més en detall en aquestes parts.

Per agafar els canvis del servidor es fa una petició al servidor i aquest retornarà els canvis corresponents. Si es reben canvis, es desen a la base de dades local. El cost necessari de la petició al servidor s'anomenarà CPS, mentre que el cost de desar les dades a nivell local s'anomenarà CDDS. El cost CPS depèn del temps que tarda el servidor a retornar els canvis, que dependrà de diversos factors: nombre de canvis fets, càrrega del servidor, estat de la xarxa, etc. El cost CDDS dependrà de la quantitat de canvis rebuts. No obstant, per tal de simplificar els càlculs i tenint en compte que estem realitzant una estimació, es suposaran que tots dos són constants.

La part d'enviar els canvis al servidor es pot dividir en tres subparts: agafar els canvis fets a nivell local, enviar-los al servidor i fer els canvis necessaris en funció de la resposta. El consum de bateria de cadascuna d'aquestes subparts s'anomenaran CRC, CEC i CTRC respectivament. Com passa amb la part de demanar canvis al servidor, cadascun d'aquests consums depenen de diversos factors, però es consideraran constants per tal de simplificar els càlculs.

A continuació es mostren les fórmules per tal de calcular el consum de bateria estimat dels dos mètodes en les situacions definides anteriorment. El consum dependrà del nombre d'execucions de cada funcionalitat. Com ja hem comentat, en el mètode SA el nombre d'execucions ( $NE_{SA}$ ) de totes dues parts (demanar canvis i enviar canvis) és el mateix i depèn únicament del temps. En el mètode GCM, el nombre d'execucions de les parts poden ser diferents i s'anomenaran  $NED_{GCM}$  i  $NEE_{GCM}$  respectivament.

### **No hi ha canvis**

Consum GCM: 0

Consum SA:  $(CB_{SA} + CPS + CRC) * NE_{SA}$

### **Només es reben canvis**

Consum GCM:  $(CB_{GCM} + CPS + CDDS) * NED_{GCM}$

Consum SA:  $(CB_{SA} + CPS + CDDS + CRC) * NE_{SA}$

### **Només s'envien canvis**

Consum GCM:  $(CB_{GCM} + CRC + CEC + CTRC) * NEE_{GCM}$

Consum SA:  $(CB_{SA} + CPS + CRC + CEC + CTRC) * NE_{SA}$

### **S'envien i es reben canvis**

Consum GCM:  $(CB_{GCM} + CPS + CDDS) * NED_{GCM} + (CB_{GCM} + CRC + CEC + CTRC) * NEE_{GCM}$

Consum SA:  $(CB_{SA} + CPS + CDDS + CRC + CEC + CTRC) * NE_{SA}$

En aquestes fórmules hi ha massa variables diferents per tal de poder realitzar una comparació de tots dos mètodes. Per una banda, necessitem una estimació de la diferència que hi pot haver entre el consum base de GCM i el consum base de SA. Com que aquests consums depenen del funcionament intern del sistema operatiu no sabem la diferència que hi haurà entre ells, així que s'haurà de fer una suposició. **Suposarem que  $CB_{GCM} = CB_{SA}$ .**

Per altra banda, necessitem donar valors als diferents nombres d'execucions. Com ja s'ha comentat, el nombre d'execucions en SA depèn del temps (recordem que el SA s'executa cada 30 segons), mentre que el nombre d'execucions en GCM depèn de la quantitat canvis fets i de si aquests s'han realitzat simultàniament o no. Concretament,

$$NE_{SA} = \text{Num segons} / 30$$

$$NEE_{GCM} = \text{Num canvis nivell local}$$

$$NED_{GCM} = \text{Num canvis altres usuaris}$$

A continuació es mostra una taula on es compararan les estimacions de consum de tots dos mètodes en diferents situacions utilitzant els càlculs definits anteriorment. Es calcularà el consum durant una hora. Com que **hem suposat  $CB_{GCM} = CB_{SA}$** , anomenarem a tots dos CB. Per poder fer comparacions exactes hauríem de saber la relació entre cadascun dels diferents. Com que a priori no la podem saber haurem de fer algunes suposicions a l'hora de decidir quin dels dos mètodes consumirà més bateria. Per exemple, quan rebem 4 canvis no simultanis cada minut, si  $CRC = CB + CPS + CDDS$  obtenim que el consum en tots dos casos és igual. No obstant, és molt poc probable que recuperar els canvis de la base de dades locals consumeixi la mateixa bateria que el cost base, demanar canvis al servidor i desar els canvis a la base de dades local. Així doncs, tot i no saber la



relació assumirem que  $CRC < CB + CPS + CDDS$  i que, per tant, SA consumeix menys bateria. El mateix passa amb la resta de situacions.

Situació	Consum
No hi ha cap canvi.	GCM: 0 SA: $120*(CB + CPS + CRC)$ <b>SA &gt; GCM</b>
Rebem un canvi cada 4 minuts.	GCM: $15*(CB + CPS + CDDS)$ SA: $120*(CB + CPS + CDDS + CRC)$ <b>SA &gt; 8*GCM</b>
Rebem un canvi cada minut.	GCM: $60*(CB + CPS + CDDS)$ SA: $120*(CB + CPS + CDDS + CRC)$ <b>SA &gt; 2*GCM</b>
Rebem 4 canvis no simultanis cada minut.	GCM: $240*(CB + CPS + CDDS)$ SA: $120*(CB + CPS + CDDS + CRC)$ <b>GCM/2 &lt; SA &lt; GCM</b>
Fem un canvi cada 4 minuts.	GCM: $15*(CB + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CRC + CEC + CTRC)$ <b>SA &gt; 8*GCM</b>
Fem un canvi cada minut.	GCM: $60*(CB + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CRC + CEC + CTRC)$ <b>SA &gt; 2*GCM</b>
Fem 4 canvis no simultanis cada minut.	GCM: $240*(CB + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CRC + CEC + CTRC)$ <b>GCM/2 &lt; SA &lt; GCM</b>
Fem un canvi i la sincronització falla constantment (hi ha connexió).	GCM: $120*(CB + CRC + CEC)$ SA: $120*(CB + CPS)$ <b>SA ≈ GCM</b>
Fem un canvi i la sincronització falla constantment (no hi ha connexió).	GCM: $CB + CRC + CEC$ SA: $120*(CB + CPS)$ <b>SA &gt; GCM</b> Com que no sabem relació entre CRC+CEC i CPS no entrem en més detall.
Rebem i fem un canvi cada 4 minuts.	GCM: $15*(CB + CPS + CDDS + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CDDS + CRC + CEC + CTRC)$ <b>SA ≈ 8*GCM</b>
Rebem i fem un canvi cada minut.	GCM: $60*(CB + CPS + CDDS + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CDDS + CRC + CEC + CTRC)$ <b>SA ≈ 2*GCM</b>
Rebem i fem 4 canvis no simultanis cada minut.	GCM: $240*(CB + CPS + CDDS + CRC + CEC + CTRC)$ SA: $120*(CB + CPS + CDDS + CRC + CEC + CTRC)$ <b>2*SA ≈ GCM</b>

Taula 163: Estudi previ experiment 3

## 3.2 Experimentació

Un cop realitzat l'experiment, recol·lectem la informació que volem utilitzar per provar la nostra hipòtesi i comparar els dos tipus de consultes. A la taula següent es poden observar els resultats d'un experiment com el que hem definit.

Rèplica	Operació	GCM	SyncAdapter
1	No hi ha cap canvi.	0%	1,16%
2	Rebem un canvi cada 4 minuts.	0,072%	1,235%
3	Rebem un canvi cada minut.	0,21%	1,28%
4	Rebem 4 canvis no simultanis cada minut.	0,48%	1,3%
5	Fem un canvi cada 4 minuts.	0,126%	1,2544%
6	Fem un canvi cada minut.	0,345%	1,2838%
7	Fem 4 canvis no simultanis cada minut.	0,768%	1,2978%
8	Fem un canvi i la sincronització falla constantment (hi ha connexió).	0,06%	2,875%
9	Fem un canvi i la sincronització falla constantment (no hi ha connexió).	0,009%	0,024%
10	Rebem i fem un canvi cada 4 minuts.	0,136%	1,728%
11	Rebem i fem un canvi cada minut.	0,414%	1,863%
12	Rebem i fem 4 canvis no simultanis cada minut.	1,784%	1,8974%

*Taula 164: Resultats experiment 3*

A la següent gràfica podem observar de manera més visual la diferència de consum en cada rèplica.

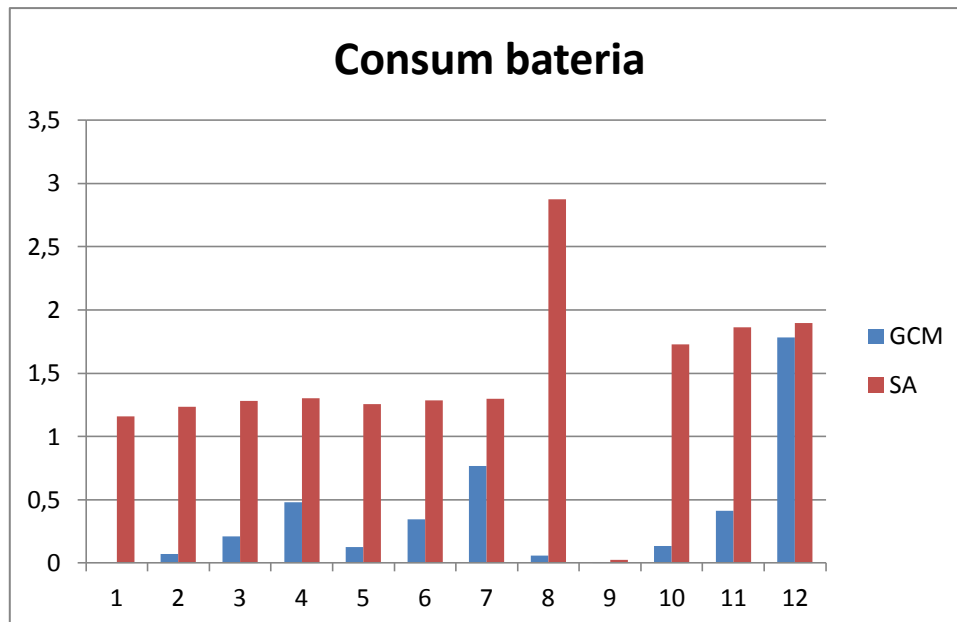


Figura 150: Gràfic comparació consum experiment 3

Com es pot veure clarament a la imatge, el consum de bateria en GCM en tots els casos tractats és sempre inferior al consum en SA. Això contradiu les previsions que havíem fet a l'estudi previ. A què és degut?

A l'estudi previ hem suposat que els consums base de tots dos mecanismes era el mateix. Observant els resultats podem deduir fàcilment que aquesta suposició és errònia, i que en realitat el cost base de SA és bastant superior al cost base de GCM. Per entendre per què això és així caldria un estudi exhaustiu del funcionament intern del sistema operatiu Android que escapa de l'àmbit d'aquest projecte.

A part d'això criden l'atenció un parell de dades. Per una banda, quan la sincronització falla perquè no hi ha connexió havíem previst que el SyncAdapter es seguiria executant, però això no ha sigut així. Els SyncAdapters, com el seu nom permet deduir, s'utilitzen per sincronitzar dades amb altres dispositius, fet que requereix disposar de connexió a Internet. Així doncs, sembla que el propi sistema operatiu inhabilita l'execució del SyncAdapter mentre no té connexió a Internet, de manera que el consum de bateria en aquest cas és molt baix.

L'altre cas que crida l'atenció és quan la sincronització falla i hi ha connexió, on havíem previst que en GCM seguiríem fent consultes al servidor ja que no podem saber quan aquest tornarà a estar *online* i que això tindria un cost de bateria elevat. Com es pot observar, tot i que el cost de bateria és superior a quan no hi ha connexió, aquest segueix sent molt baix. No hem aconseguit trobar indicis definitius per tal d'identificar el motiu amb seguretat, però suposem que el propi sistema operatiu deu tenir un mecanisme per tractar crides freqüents a la mateixa adreça, i aquest mecanisme deu estalviar gran part del cost de bateria de l'operació. En cas que sigui així, però, és estrany que aquest mecanisme no s'apliqui també en el cas de SA, ja que les crides són exactament iguals. Així doncs, no hem aconseguit trobar el motiu pel qual el consum de GCM quan l'operació falla per culpa del servidor és tan baix.

La hipòtesi H0 és que totes tots dos mètodes de sincronització consumeixen la mateixa bateria. Amb aquest experiment s'ha comprovat que no és així, sinó que en la majoria de casos és millor utilitzar GCM (concretament, en tots els casos provats).

Podem concloure que ens interessa enormement utilitzar GCM enlloc de SyncAdapter, ja que la millora de consum és molt significativa, especialment quan no es realitza cap canvi. Si bé és cert que en l'última rèplica tots dos mecanismes tenen un consum bastant similar i que en cas de seguir augmentant el nombre de canvis el consum de GCM superaria el de SA, cal dir que els canvis els realitzen persones manualment i que no solen ser canvis ràpids de fer, és poc probable que en un cas real un usuari faci més de 4 canvis per minut mentre un altre usuari en fa més de 4 també. Sí que és més fàcil que succeeixi que hi hagi diversos usuaris fent modificacions a la vegada i l'usuari rebi una quantitat de canvis superior als de l'experiment mentre treballa amb l'aplicació, però aquests casos no seran freqüents i, encara que succeeixin, el consum en GCM no serà gaire superior al consum en SA. Així doncs, **pel què fa al consum de bateria GCM és molt millor que SA.**

## 4. Desar només tasques vs desar-ho tot

Durant el disseny i implementació del projecte ens vam trobar amb un dubte que ja s'ha explicat anteriorment: quanta informació cal guardar als dispositius mòbils?

Com ja s'ha explicat, la decisió que es va prendre va ser desar tota la informació relacionada amb l'usuari: grups, calendaris, altres usuaris, etc. Això augmenta la quantitat d'informació emmagatzemada a nivell local, però també augmenta enormement la quantitat d'accions que l'usuari pot realitzar sense connexió a Internet.

En aquest experiment es compararan les dues alternatives per mesurar com afecta en temps i espai aquesta decisió. És a dir, coneixem l'avantatge de desar tota la informació a nivell local, que és l'increment de funcionalitats *offline*, però ens cal conèixer exactament els desavantatges que comporta aquesta decisió, i això és el que pretenem descobrir amb aquest experiment: si aquest augment de temps i espai és acceptable.

Cal aclarir, però, que el fet de guardar només les tasques també obliga a emmagatzemar certa informació de la resta de classes. Més concretament, en cada tasca s'hauria de desar els noms de tots els grups i usuaris participants, els identificadors i noms de les tasques precedents i el nom del calendari al qual pertany. També cal desar la informació de l'usuari que està utilitzant l'aplicació.

El cost de modificar informació es pot considerar el mateix en tots dos casos, ja que l'única diferència que hi ha és que, en cas de desar-ho tot a nivell local, els canvis fets a grups o calendaris també es modifiquen a nivell local, i fer un únic canvi a la base de dades local té un cost temporal insignificant comparat amb el cost temporal d'una petició al servidor. Per tant, només estudiarem les peticions de dades al servidor, concretament ens situarem en el cas pitjor i analitzarem el cost temporal i espacial de realitzar una sincronització completa en cadascun dels models.

Podem resumir les característiques de l'experiment amb la taula següent:

<b>Observació</b>	Emmagatzemar tota la informació a nivell local té un cost temporal i espacial superior a emmagatzemar només les tasques.
<b>Plantejament</b>	Generem un conjunt de models, realitzem les consultes pròpies de cada alternativa en cadascun d'ells i comparem els resultats obtinguts.
<b>Hipòtesis</b>	La quantitat d'informació que s'ha de desar a nivell local no influeix en el temps de la consulta ( $H_0$ ) o el temps de consulta depèn de la informació que s'ha de desar a nivell local.
<b>Mètode</b>	-Executarem un total de 8 rèpliques. -Cada rèplica l'executarem deu vegades i anotarem el temps mitjà

	<p>(l'espai és constant).</p> <ul style="list-style-type: none"> <li>-Dins de cada rèplica utilitzarem les dues alternatives sobre el mateix model. Sempre es realitzarà una sincronització completa sense enviar res, només rebent la informació necessària.</li> <li>-Per calcular l'espai no es tindrà en compte l'espai que ocupa tota la base de dades local, sinó que es mesuraran únicament les dades emmagatzemades dins de la base de dades.</li> <li>-En el càlcul de l'espai es tindrà en compte el cas pitjor, és a dir, que tots els camps modificables per l'usuari estiguin ocupats amb el màxim de caràcters permesos excepte els camps amb un cert conjunt de valors possibles, que es tractaran com si sempre hi hagués el valor més gran. En el cas dels identificadors locals de les tasques, es considerarà que ocupen sempre 8 bytes.</li> <li>-Degut a que la longitud d'alguns camps de les <i>Pending Resolution</i> és molt variable, s'ha fixat com a mida estàndard la mitjana de les seves mides abans de començar l'experiment: 348 bytes.</li> </ul>
--	---

*Taula 165: Resum experiment 4*

## 4.1 Estudi previ

En aquest experiment s'analitzarà el temps que es tarda a obtenir les dades i l'espai que cada model ocupa en cadascun dels casos. Comencem pel més senzill: l'espai.

### Espai

Com que coneixem els atributs de cadascuna de les classes i assumim que sempre ens trobem en el cas pitjor, calcular la mida que les dades ocuparan a la base de dades en cada model és bastant senzill. Els bytes que ocupa cada taula de la base de dades **si dessem tota la informació** són:

Grup: 69 bytes

Calendari: 199 bytes

Tasca: 306 bytes

Usuari: 61 bytes

Precedeix: 18 bytes

Participa/Assisteix: 46 bytes

Altres relacions: 72 bytes

Esdeveniment: 212 bytes

Meeting/Pending approval: 348 bytes

Si només dessem les tasques, la mida de les tasques i relacions a la base de dades local variarà ja que hi haurem d'incloure informació addicional. Per cada usuari o grup participant haurem d'incloure el seu nom, ja que a l'usuari vol veure el nom, no l'identificador, i com que no tenim els usuaris i els grups desats a la base de dades local i podem estar treballant *offline* aquesta informació s'haurà de desar en algun altre lloc. El mateix passa amb les tasques precedents o següents, ja que com que només desarem les tasques en què participa l'usuari és possible que la tasca precedent o següent no es trobi a nivell local. En aquest cas, però, cal desar també la data inicial i final de la tasca per poder evitar conflictes quan es modifiqui la data d'inici o la data final d'alguna tasca, ja que aquesta operació s'ha de poder fer *offline*.

Per cada user o grup part, +25 bytes.

Per cada tasca precedent, +61 bytes.

Si només dessem les tasques també desarem un usuari a nivell local: el que està utilitzant l'aplicació.

## **Temps**

Pel què fa al temps, la primera impressió és que el fet de desar només tasques sempre serà més ràpid, ja que ens interessa recuperar menys informació, però cal analitzar el cas més en detall abans de prendre conclusions precipitades.

Com es pot veure a l'anàlisi de l'espai, quan dessem només tasques hi ha una quantitat d'informació repetida, que és la informació addicional que desarem a les tasques. Per exemple, quan ho dessem tot tenim la informació d'un usuari un sol cop, i a partir d'aquí tenim referències a aquest usuari (per les referències utilitzem només l'identificador). En l'altre cas, quan dessem només les tasques, tindrem el nom de l'usuari desat tants cops com faci falta, repetint també aquesta informació a part de l'identificador. Aquesta recuperació d'informació repetida farà augmentar el temps de la consulta en casos en què hi hagi molts *Participa*, *Assisteix* o *Precedes*.

L'afirmació següent es pot simplificar de la següent manera:

-Com més *Participa*, *Assisteix* o *Precedes*, el temps de només tasques augmenta una mica més que el temps de desar tota la informació.

-Com més grups, calendaris, usuaris, esdeveniments, pending approvals i relacions entre ells, el temps de desar tota la informació augmenta significativament mentre que el temps de desar només tasques es manté igual.

Per tal que aquests dos increments s'igualin i el temps de tots dos mètodes sigui igual és obvi que la quantitat de *Precedes*, *Assisteix* o *Precedes* ha de ser bastant més elevada que la quantitat d'altres classes, però és possible que succeeixi. Així doncs, podem afirmar que el fet de desar només tasques serà més ràpid sempre i quan el nombre de relacions de les tasques mencionades anteriorment no sigui molt elevat comparat amb la resta de classes. El terme "molt elevat" no es pot quantificar en detall, caldrà realitzar l'experimentació per comprovar com es comporta. A més, també cal tenir en compte que no totes les classes impliquen el mateix temps, és possible que recuperar 10 calendaris necessiti més temps que

recuperar 10 grups o viceversa. El que sí que podem afirmar és que recuperar relacions és més ràpid que recuperar mòduls que no representen relacions.

### Previsió final de l'estudi

Un cop reunida tota la informació presentada en aquest estudi previ, podem resumir els resultats esperats de l'experiment en la taula següent.

Rèplica	Model	Tot	Només tasques
1	1 grup, 1 usuari, 1 calendari, 0 tasques	Més lent 473 bytes	Més ràpid 61 bytes (user)
2	1 grup, 1 usuari, 1 calendari, 10 tasques, 10 Participa	Més lent 3.993 bytes	Més ràpid 3.831 bytes
3	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 0 tasques	Més lent 3.805 bytes	Més ràpid 61 bytes (user)
4	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 20 tasques, 20 Participa, 15 esdeveniments	Més lent 14.025 bytes	Més ràpid 10.781 bytes
5	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 40 tasques, 40 Participa, 20 Assisteix, 30 esdeveniments	Més ràpid? 25.165 bytes	Més lent? 22.860 bytes
6	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 25 Comparteix, 0 tasques	Més lent 18.105 bytes	Més ràpid 61 bytes (user)
7	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 25 Comparteix, 40 tasques, 40 Participa, 20 Assisteix, 30 esdeveniments	Més lent 39.465 bytes	Més lent 22.860 bytes
8	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 20 Comparteix, 200 tasques, 200 Participa, 50 Assisteix, 50 Precedeix, 60 esdeveniments	Més ràpid? 136.825 bytes	Més lent? 130.520 bytes

Taula 166: Estudi previ experiment 4

Al no poder quantificar la diferència que hi ha d'haver entre relacions de les tasques i altres classes per tal que s'igualin els temps hem hagut de realitzar algunes suposicions. En aquests casos la previsió s'ha marcat amb un interrogant. És el cas de les rèpliques 5 i 8, on la quantitat d'altres classes és bastant major que la quantitat de relacions, però no sabem si aquesta diferència és prou important.



## 4.2 Experimentació

Un cop realitzat l'experiment, recol·lectem la informació que volem utilitzar per provar la nostra hipòtesi i comparar les dues alternatives. A les taules següents es poden observar els resultats d'un experiment com el que hem definit.

Rèplica	Model	Tot	Només tasques
1	1 grup, 1 usuari, 1 calendari, 0 tasques	1.457,2 ms 473 bytes	802,8 ms 61 bytes (user)
2	1 grup, 1 usuari, 1 calendari, 10 tasques, 10 Participa	2.500,5 ms 3.993 bytes	1.973,3 ms 3.831 bytes
3	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 0 tasques	2.284,1 ms 3.805 bytes	941,1 ms 61 bytes (user)
4	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 20 tasques, 20 Participa, 15 esdeveniments	4.003,5 ms 14.025 bytes	3195,6 ms 10.781 bytes
5	5 grups, 5 usuaris, 5 calendaris, 20 Pertany, 10 Comparteix, 40 tasques, 40 Participa, 20 Assisteix, 30 esdeveniments	6.356,1 ms 25.165 bytes	5.859,8 ms 22.860 bytes
6	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 25 Comparteix, 0 tasques	4.290,2 ms 18.105 bytes	1064,3 ms 61 bytes (user)
7	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 25 Comparteix, 40 tasques, 40 Participa, 20 Assisteix, 30 esdeveniments	7.284,8 ms 39.465 bytes	5.668,6 ms 22.860 bytes
8	10 grups, 30 usuaris, 15 calendaris, 150 Pertany, 20 Comparteix, 200 tasques, 200 Participa, 50 Assisteix, 50 Precedeix, 60 esdeveniments	24.371,2 ms 136.825 bytes	25.590,4 ms 130.520 bytes

*Taula 167: Resultats experiment 4*

A la següent gràfica podem observar de manera més visual la diferència de temps en cada rèplica.

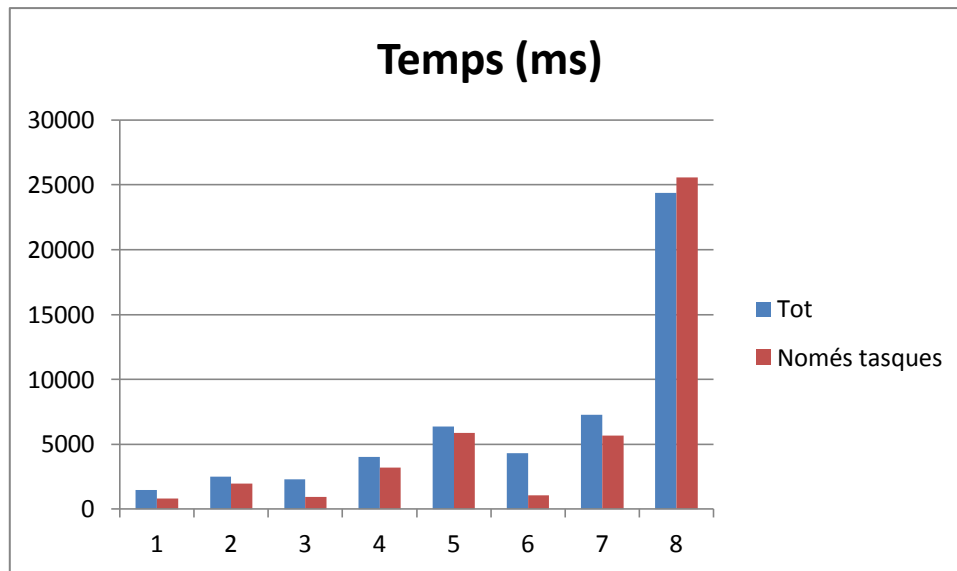


Figura 151: Gràfic comparació temps experiment 4

Òbviament, l'espai de la previsió s'ha encertat perquè els càlculs s'han realitzat amb el màxim detall, no hi havia marge d'error.

En quant al temps, en els casos amb poques relacions de tasques la previsió s'ha encertat, ja que era obvi que quan ho desem tot el cost temporal sigui major. El problema que teníem a la previsió era quantificar quina havia de ser la diferència entre la quantitat de relacions de tasques i la quantitat d'altres classes perquè el cost temporal de desar només les tasques superés el cost temporal de desar-ho tot. A les rèpliques 5 i 8, on aquesta diferència és elevada, s'havia fet la suposició que seria suficient, però com es pot veure en el gràfic anterior a la rèplica 5 ens vam equivocar. A la rèplica 5 la diferència és elevada, i això fa que els costos siguin molt semblants, però no és suficient. En el cas de la rèplica 8, en canvi, sí que ho és, i això fa que el cost de desar només tasques sigui superior al cost de desar-ho tot.

La hipòtesi  $H_0$  és que la quantitat d'informació que s'ha de desar a nivell local no influeix en el temps de consulta. Amb aquest experiment s'ha comprovat que no és així, sinó que com era d'esperar en la majoria de casos el fet de desar-ho tot té un cost temporal major, però no sempre.

En un cas real és probable que la quantitat de grups i calendaris en què participa un usuari no sigui massa elevada, però sí que ho seran les quantitats de tasques, relacions i esdeveniments. És clar que el cas de la rèplica 6 és molt poc probable que succeeixi, el més probable és que ens trobem amb casos com els de les rèpliques 2, 4, 5, 7 i 8. Així doncs, com es pot observar al gràfic, encara que el fet de desar-ho tot segurament tindrà un cost temporal superior, aquest no serà massa elevat comparat amb el cost de desar només les tasques, és un increment totalment assumible. Pel que fa a l'espai, podem veure que en un cas bastant extrem com és el de la rèplica 8 la quantitat d'informació emmagatzemada no és massa elevada per un telèfon mòbil.

Podem concloure, doncs, que l'increment de temps i d'espai provocat pel fet de desar-ho tot és assumible i que, per tant, podem utilitzar aquesta opció per aprofitar els avantatges que ens ofereix.

# **Planificació i costos**

En aquest apartat es presenta la planificació temporal inicial, la planificació temporal final i l'anàlisi econòmic del projecte tenint en compte la majoria de factors que l'afecten.

# 1. Planificació temporal

## 1.1 Planificació inicial

A continuació es pot observar un diagrama de Gantt amb la planificació inicial del projecte. S'ha dividit en dues parts: el llistat de tasques i el diagrama en sí. S'ha fixat una càrrega de treball de 6 hores diàries.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>Estudi documentació</b>	<b>2 días</b>	<b>lun 05/03/12</b>	<b>mar 06/03/12</b>	
Lectura i anàlisi memòria	1 día	lun 05/03/12	lun 05/03/12	
Anàlisi codi	1 día	mar 06/03/12	mar 06/03/12	2
<b>Investigació tecnologies</b>	<b>7 días</b>	<b>mié 07/03/12</b>	<b>jue 15/03/12</b>	<b>1</b>
Investigació i experimentació SugarCRM	3 días	mié 07/03/12	vie 09/03/12	
Investigació plataforma Android	3 días	vie 09/03/12	mar 13/03/12	
Investigació sincronització	1 día	mié 14/03/12	mié 14/03/12	
Anàlisi alternatives	1 día	jue 15/03/12	jue 15/03/12	5;6;7
<b>Planificació temporal</b>	<b>1 día</b>	<b>vie 16/03/12</b>	<b>vie 16/03/12</b>	<b>4</b>
Elaboració Gantt	1 día	vie 16/03/12	vie 16/03/12	
<b>Definició del projecte</b>	<b>3 días</b>	<b>lun 19/03/12</b>	<b>mié 21/03/12</b>	<b>9</b>
Definició objectius	1 día	lun 19/03/12	lun 19/03/12	
Planificació de costos	1 día	lun 19/03/12	lun 19/03/12	
Anàlisi de requisits	1 día	mar 20/03/12	mar 20/03/12	
Anàlisi de riscos	1 día	mié 21/03/12	mié 21/03/12	
<b>Especificació</b>	<b>7 días</b>	<b>lun 26/03/12</b>	<b>mar 03/04/12</b>	<b>11</b>
Model conceptual	1 día	lun 26/03/12	lun 26/03/12	
Definició actors	1 día	mar 27/03/12	mar 27/03/12	
Casos d'ús	4 días	mar 27/03/12	vie 30/03/12	
Diagrames seqüència	2 días	vie 30/03/12	lun 02/04/12	
Contractes operacions	2 días	lun 02/04/12	mar 03/04/12	
<b>Disseny</b>	<b>9 días</b>	<b>jue 05/04/12</b>	<b>mar 17/04/12</b>	<b>16</b>
Definició arquitectura	2 días	jue 05/04/12	vie 06/04/12	
Diagrames de seqüència	8 días	vie 06/04/12	mar 17/04/12	
Disseny navegació	1 día	mar 17/04/12	mar 17/04/12	
<b>Implementació</b>	<b>41 días</b>	<b>jue 19/04/12</b>	<b>jue 14/06/12</b>	<b>22</b>
Aplicació local	20 días	jue 19/04/12	mié 16/05/12	
Sincronització amb servidor	6 días	mié 16/05/12	mié 23/05/12	
Sincronització amb altres dispositius	14 días	jue 24/05/12	mar 12/06/12	
Revisió i correcció d'errors	2 días	mié 13/06/12	jue 14/06/12	
<b>Experimentació</b>	<b>12 días</b>	<b>lun 18/06/12</b>	<b>mar 03/07/12</b>	<b>26</b>
Proves sincronització	12 días	lun 18/06/12	mar 03/07/12	
<b>Versió beta</b>	<b>26 días</b>	<b>lun 09/07/12</b>	<b>lun 13/08/12</b>	
Aplicació versió beta	25 días	lun 09/07/12	vie 10/08/12	
Recollida dades	1 día	lun 13/08/12	lun 13/08/12	
<b>Documentació</b>	<b>106 días</b>	<b>jue 22/03/12</b>	<b>jue 16/08/12</b>	
Definició projecte	2 días	jue 22/03/12	vie 23/03/12	
Especificació	1 día	mié 04/04/12	mié 04/04/12	
Disseny	1 día	mié 18/04/12	mié 18/04/12	22
Implementació	1 día	vie 15/06/12	vie 15/06/12	26
Experimentació	3 días	mié 04/07/12	vie 06/07/12	31
Manual d'usuari	2 días	mar 10/07/12	mié 11/07/12	
Revisió i finalització	3 días	mar 14/08/12	jue 16/08/12	

Figura 152: Planificació inicial en llista de tasques

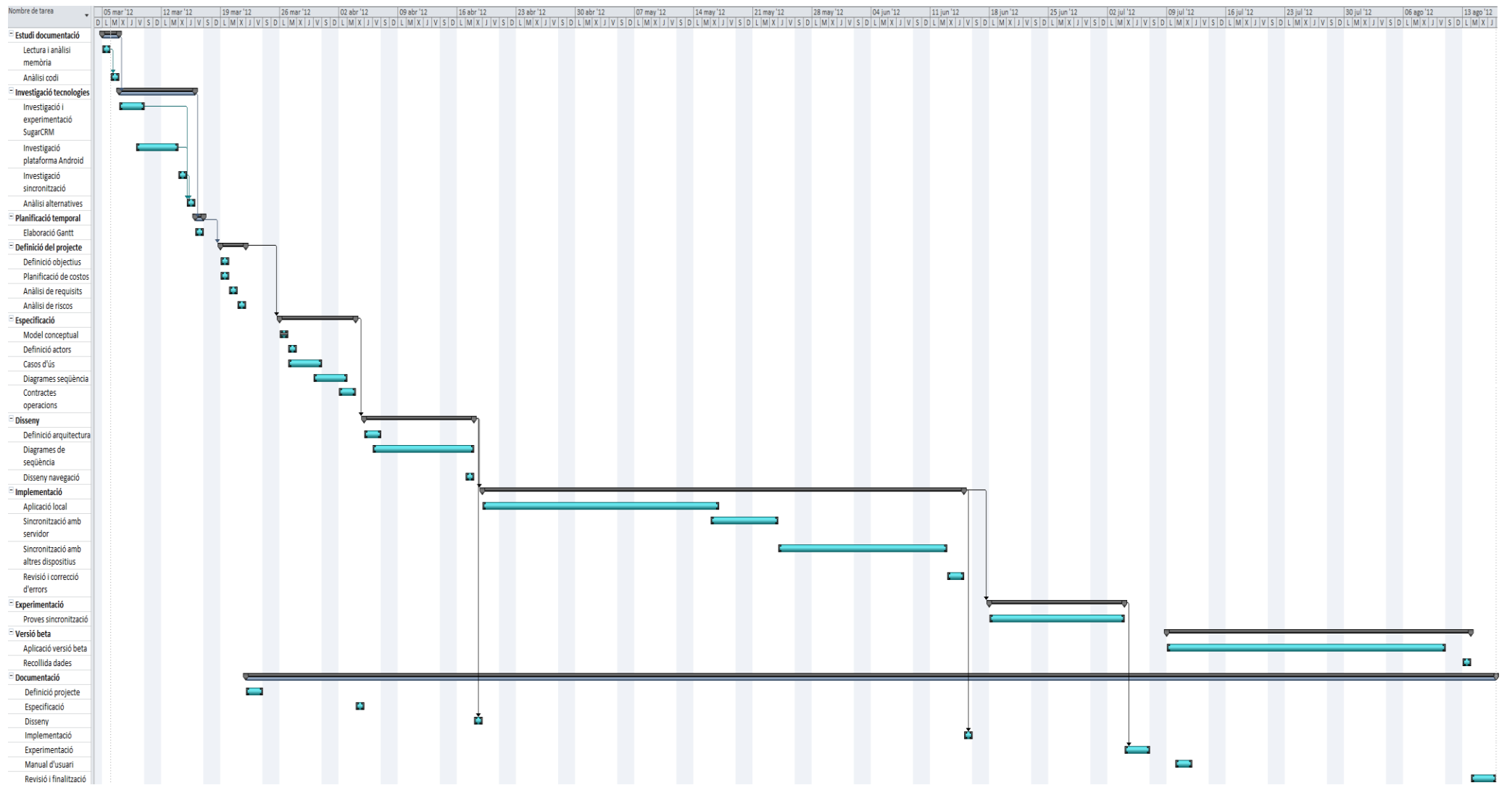


Figura 153: Planificació inicial en diagrama de Gantt

## 1.2 Planificació final

La planificació final és la que realment s'ha dut a terme i conté diferències amb la planificació inicial, especialment en les dates de finalització de les etapes. Per una banda, algunes de les etapes han necessitat més hores de les previstes, especialment la part d'implementació i experimentació. Això és degut a que mai havia treballat amb algunes de les tecnologies utilitzades (Android, SugarCRM, P2P, etc.), de manera que a la planificació inicial es va fer una estimació poc precisa, i a que han anat sorgint imprevistos que ha calgut corregir. Per altra banda, el fet de realitzar el projecte en paral·lel amb altres obligacions i imprevistos (una altra assignatura, feina, operació quirúrgica, etc.) ha provocat que no es poguessin dedicar les hores per setmana que s'havien fixat inicialment, cosa que ha provocat un endarreriment important de les dates de finalització de les etapes.

A la taula següent es mostren la quantitat d'hores de cada etapa de la planificació inicial i de la final per tal que es puguin comparar entre elles.

Fase	Inicial	Final	Desviació
Estudi documentació	12	12	0
Investigació tecnologies	42	42	0
Planificació i definició	36	36	0
Especificació	48	54	+6
Investigació android-binding	No previst	12	+12
Disseny	60	60	0
Implementació	252	408	+166
Experimentació	90	150	+60
Aplicació beta	6	6	0
Correccions i finalització	24	42	+18
Dies perduts per causes alienes al projecte	No previst	350	+350
<b>Total projecte</b>	<b>570</b>	<b>822</b>	<b>+252 (+602)</b>

*Taula 168: Planificació final*

## 2. Anàlisi econòmic

Per a realitzar el projecte no es requerirà cap despesa afegida, ja que dispenso de smartphones per tal de realitzar les proves, el departament de LSI dispenso d'un servidor per a la realització de projectes i el software utilitzat o és open source, o dispenso de versions gratuïtes o ja dispenso de la llicència pertinent. No obstant això, crec que el fet de calcular el cost del projecte en cas de no dispenso de cap dels recursos necessaris és interessant per una banda per conèixer el cost del projecte

en un entorn d'empresa i, per l'altra, com a referència per a possibles ampliacions futures on no es disposi d'alguns dels recursos esmentats.

Per tant, a continuació es presenta l'anàlisi econòmic del projecte en cas de no disposar de cap dels elements utilitzats durant el seu desenvolupament.

## 2.1 Costos de hardware

A la taula següent es poden observar els costos del hardware necessari:

Producte	Preu
1 PC IntelCore i5	600 €
1 servidor HP PROLIANT DL380 G7	1.945,95€
2 smartphones diferents	380 €
<b>Total</b>	<b>2.925,95 €</b>

*Taula 169: Costos de hardware*

## 2.2 Costos de software

A la taula següent es poden observar els costos del software necessari per a elaborar el projecte i la memòria:

Producte	Preu
Microsoft Office 2010	139 €
Windows 7	169,99 €
Gantt Project	0 €
ArgoUML	0 €
Microsoft Visio 2010	330€
Eclipse	0 €
SugarCRM Community Edition	0 €
<b>Total</b>	<b>638,99 €</b>

*Taula 170: Costos de software*

## 2.3 Costos fixes

A la taula següent es poden observar els costos fixos del projecte. Aquests costos depenen de la duració del projecte en mesos, no en hores. Com que estem tractant un entorn empresarial, suposarem una càrrega de treball de 40 hores setmanals i que les hores del projecte reflectides a la planificació final es realitzen en sèrie, suposarem que no hi ha cap tasca que es realitzi en paral·lel. És a dir,

822 hores / 40 hores/setmana = 20.55 setmanes ≈ 5 mesos

<b>Producte</b>	<b>Preu mensual</b>	<b>Total</b>
Internet	35 €/mes	175 €
<b>Total</b>	<b>75 €/mes</b>	<b>175 €</b>

*Taula 171: Costos fixes*

## 2.4 Costos de personal

Durant l'elaboració del projecte s'han dut a terme tasques corresponents a diferents perfils professionals. En la taula següent es representen les hores corresponents a cada perfil.

<b>Perfil</b>	<b>Hores</b>
Cap de projecte	60 hores
Analista funcional	68 hores
Analista tècnic	92 hores
Programador	624 hores

*Taula 172: Hores personal*

Com es pot observar, el total d'hores sobrepassa el total d'hores del projecte. Això és així perquè algunes tasques les realitzen més d'un rol en paral·lel, com per exemple l'estudi de les tecnologies.

A la taula següent es presenta el cost del personal a partir de la taula anterior.

<b>Perfil</b>	<b>Hores</b>	<b>Preu per hora</b>	<b>Preu</b>
Cap de projecte	60 h	40 €/h	2.400 €
Analista funcional	68 h	25 €/h	1.700 €
Analista tècnic	92 h	25 €/h	2.300 €
Programador	624 h	15 €/h	9.360 €
<b>Total</b>			<b>15.750 €</b>

*Taula 173: Costos de personal*



## 2.5 Cost total del projecte

<b>Concepte</b>	<b>Preu</b>
Costos de hardware	2.925,95 €
Costos de software	638,99 €
Costos fixes	175 €
Costos de personal	15.750 €
<b>Total</b>	<b>19.489,94 €</b>

*Taula 174: Cost total del projecte*

# **Manual d'instal·lació i d'usuari**

El manual d'instal·lació i d'usuari mostra com instal·lar els diversos components del sistema per tal de fer-lo operatiu i el funcionament de l'aplicació mòbil per part dels usuaris per tal que aquests puguin entendre i utilitzar totes les funcionalitats que l'aplicació mòbil ofereix.

# 1. Introducció

Com ja s'ha comentat anteriorment en aquest document, l'aplicació *SweetSynca* permet compartir calendaris i tasques amb altres usuaris. Els usuaris s'agrupen en grups, que també són creats i administrats pels propis usuaris. Els calendaris poden ser compartits amb els membres d'un o més grups. Tots els membres dels grups participants podran veure les tasques del calendari, però només podran modificar-les si participen a la tasca en qüestió.

Tota la interfície de l'aplicació mòbil està en anglès per tal que l'aplicació sigui accessible per un conjunt d'usuaris més ampli.

## 2. Instal·lació

### 2.1 Instal·lació de l'aplicació mòbil

La instal·lació de l'aplicació mòbil és molt senzilla, es realitza exactament igual com es faria amb qualsevol altra aplicació Android. Si l'aplicació es troba al *market* només cal descarregar-la i s'instal·larà automàticament. Si no hi és, cal passar el fitxer en format *.apk* al telèfon mòbil i instal·lar-la utilitzant un programa explorador d'arxius o simplement clicant en el fitxer en qüestió.

### 2.2 Instal·lació del servidor

Per tal que el sistema funcioni correctament i en la seva totalitat cal seguir una sèrie de passos en el nostre servidor. Els fitxers necessaris per a la instal·lació són:

- Instal·lador SugarCRM
- SweetSyncaWebServiceImpl.php
- \_AddJobsHere.php
- Instal·lador memcached i llibreria PHP memcache

Per començar, cal instal·lar SugarCRM al servidor. En el desenvolupament del projecte s'ha utilitzat la versió Community Edition 6.4.0, que es pot descarregar gratuïtament des de la pròpia web de SugarCRM.

Un cop instal·lat SugarCRM cal afegir-hi els mòduls necessaris. És important que els noms dels mòduls i els seus atributs coincideixin exactament amb els presentats a l'apartat 1.3.2 de la fase de disseny, ja que en cas contrari l'aplicació no funcionaria correctament. En cas que es vulgui canviar el nom d'algun mòdul o utilitzar un mòdul existent enlloc de crear-ne un de nou caldrà canviar totes les referències presents a l'API (fitxer *SweetSyncaWebServiceImpl.php*) i a l'aplicació mòbil. Els mòduls es poden crear fàcilment utilitzant l'eina *Module Builder*, que es troba a l'apartat *Admin* de SugarCRM (cal accedir-hi amb el compte d'administrador).

A continuació cal aplicar la nostra API enlloc de l'API per defecte de Sugar. Per fer-ho, cal introduir el fitxer *SweetSyncaWebServiceImpl.php* a la carpeta *service/v4\_1* i modificar el fitxer *rest.php*. Cal substituir les línies:

```
require_once('SugarWebServiceImplv4_1.php');
$webservice_impl_class = 'SugarWebServiceImplv4_1';
```

per:

```
require_once('SweetSyncalWebServiceImpl.php');
$webservice_impl_class = 'SweetSyncalWebServiceImpl';
```

Amb aquests dos passos la nostra aplicació mòbil ja podrà utilitzar la nostra API de Sugar i accedir a totes les funcionalitats que el nostre sistema ofereix. Però això no és suficient per tal que l'aplicació estigui completa.

Cada minut el sistema ha de comprovar si hi ha pending approvals que han de ser finalitzats perquè ha passat massa estona. Per tal de fer-ho haurem de modificar el fitxer `_AddJobsHere.php` present a la carpeta `modules/Schedulers`. Si aquest fitxer no ha estat modificat el podem substituir directament amb el fitxer de mateix nom proporcionat amb l'aplicació. En cas contrari, caldrà afegir una nova tasca en aquest fitxer que serà la que es dedicarà a revisar els pending approvals. El codi d'aquesta tasca es pot trobar al fitxer `_AddJobsHere.php` proporcionat amb l'aplicació i s'ha d'introduir al fitxer de mateix nom present a SugarCRM. Un cop fet això, cal programar la tasca per tal que s'executi. Per fer-ho, s'ha d'accedir a SugarCRM com a administrador i utilitzar l'eina *Scheduler* de l'apartat *Admin*. Amb aquesta eina cal crear un nou Scheduler que executi la tasca creada (en el fitxer proporcionat amb l'aplicació s'anomena "Check Pending Approvals") cada minut.

Per altra banda, l'API proporcionada utilitza memcached per tal de comprovar si dos usuaris estan modificant el mateix recurs al mateix temps. És per aquest motiu que cal instal·lar memcached al servidor i les seves dependències. Els fitxers necessaris per a la instal·lació es poden descarregar gratuïtament des de la pàgina oficial de memcached. També caldrà afegir la llibreria PHP de memcache [a la distribució de PHP que utilitzi SugarCRM](#). Si hi ha més d'una versió de PHP instal·lada al servidor cal assegurar-se que la llibreria s'ha afegit a la distribució que utilitza SugarCRM, altrament no funcionaria i diversos usuaris podrien modificar el mateix recurs al mateix temps.

Arribats a aquest punt el sistema del servidor ja està complet. Només resta afegir els usuaris que es desitja que utilitzin l'aplicació per tal que puguin accedir a l'aplicació mòbil. Això es pot fer amb l'eina *User Management*, present també a l'apartat *Admin*.

### 3. Accés a l'aplicació

Quan s'inicia l'aplicació per primer cop o després de fer logout amb èxit es mostrarà la interfície de login.

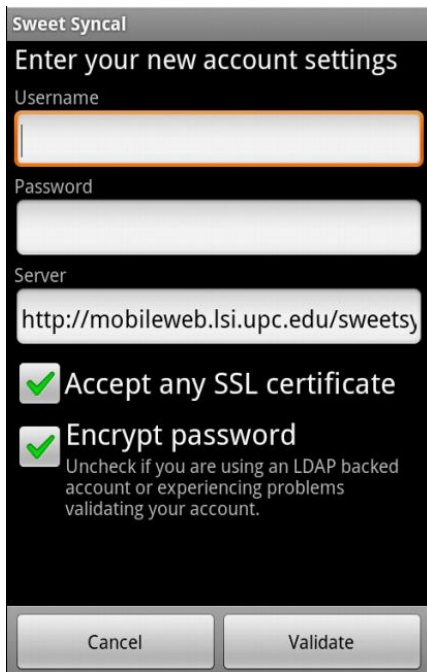


Figura 154: Vista login

El nom d'usuari i contrasenya venen definits per l'usuari corresponent al servidor SugarCRM.

En el camp servidor hi ha d'aparèixer la URL de l'API de Sugar instal·lada al servidor. Per defecte ja contindrà aquesta URL.

Els checkboxes estan marcats per defecte i no s'haurien de modificar a no ser que es produeixi algun error al realitzar el login.

Després de prémer el botó validar, l'aplicació validarà les dades amb el servidor i mostrarà una vista de confirmació. Si l'usuari confirma el login, l'aplicació agafarà les dades necessàries del servidor i mostrarà la interfície de llistat de tasques.

## 4. Gestió de tasques

### 4.1 Llistat de tasques

La interfície de llistat de tasques es mostra quan es completa un login amb èxit, quan s'inicia l'aplicació amb una sessió activa o quan s'escullen les opcions pertinents al menú principal.

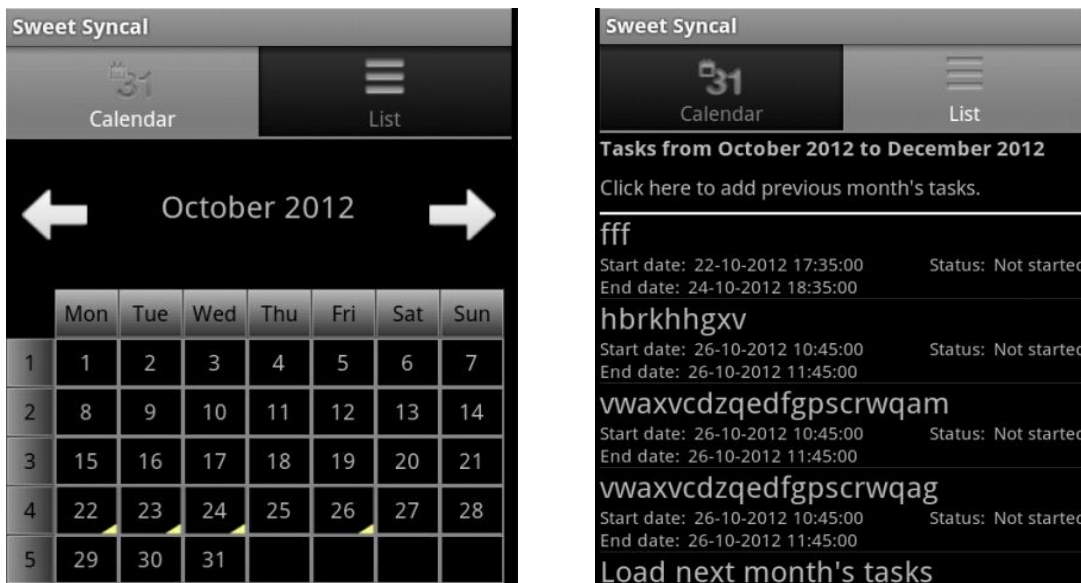


Figura 155: Vista llistar tasques

A la vista en mode calendari (imatge esquerra) es pot clicar:

- List:** Mostra la vista en mode llista (imatge dreta).
- Fletxes:** Avança/retrocedeix el mes mostrat.

- Nom del mes:** Mostra la vista en mode llista amb les tasques del mes clicat.
- Nom de dia de la setmana:** Mostra la vista en mode llista amb les tasques del dia de la setmana clicat (per exemple, tots els dilluns d'octubre de 2012).
- Número de la setmana:** Mostra la vista en mode llista amb les tasques de la setmana clicada.
- Dia:** Mostra la vista en mode llista amb les tasques del dia clicat.

A la vista en mode llista (imatge dreta) es pot clicar:

- Calendar:** Mostra la vista en mode calendari (imatge esquerra).
- Subtítol ("Click here..."):** Si la llista mostra les tasques d'un o més mesos, carrega les tasques del mes anterior al primer mostrat. En cas contrari (s'estan mostrant tasques d'un o més dies), es tornen a mostrar tasques d'un o més mesos.
- Tasca:** Mostra la vista amb la informació de la tasca (apartat 4.3).
- Peu de la llista (Load next...):** Carrega les tasques del mes següent a l'últim mostrat (només si es mostren tasques d'un o més mesos).

Si es manté premuda una tasca en mode llista s'obre el menú contextual amb les opcions:

- Modificar estat:** Hi ha un total de 4 opcions per modificar l'estat de la tasca en funció de l'estat que es vulgui assignar. L'opció de l'estat actual de la tasca no es mostrarà.
- Eliminar tasca:** Elimina la tasca clicada.

En qualsevol dels dos modes el menú mostra les opcions següents:

- Nova tasca:** Mostra la interfície per crear una tasca (apartat 4.4).
- Resetejar mesos llista:** Si la vista en mode llista mostra les tasques d'un o més mesos passarà a mostrar les tasques del mes actual.
- Filtrar tasques:** Mostra la interfície de filtrat de tasques.

## 4.2 Filtrar tasques

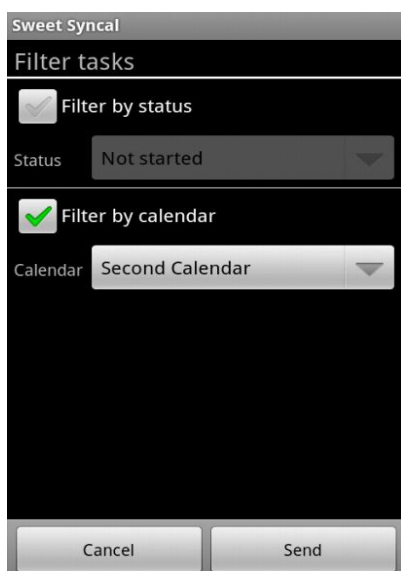


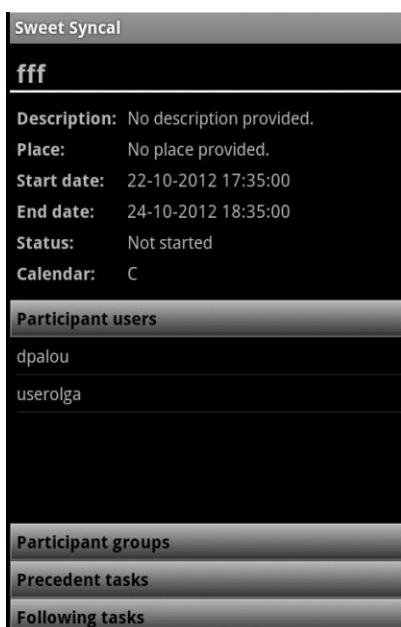
Figura 156: Vista filtrar tasques

La interfície de filtrar tasques permet definir els paràmetres per filtrar les tasques en la interfície de llistar tasques. Es pot filtrar per estat i/o per calendari.

El filtre per data es fa a la mateixa interfície de llistar tasques.

Quan es prem el botó d'enviar es mostrarà la interfície de llistar tasques amb el filtre actiu.

### 4.3 Informació tasca



Sweet Syncal

fff

Description: No description provided.

Place: No place provided.

Start date: 22-10-2012 17:35:00

End date: 24-10-2012 18:35:00

Status: Not started

Calendar: C

Participant users

dpalou

userolga

Participant groups

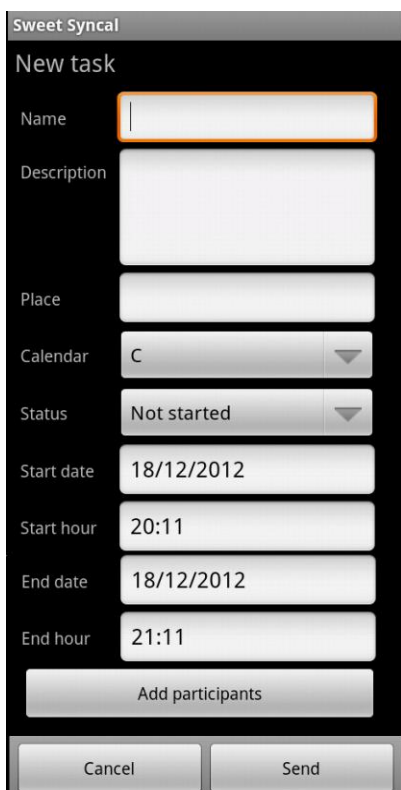
Precedent tasks

Following tasks

Figura 157: Vista informació tasca

Si es manté pressionat un participant o tasca precedent apareixerà un menú contextual per eliminar el participant o la tasca.

### 4.4 Crear/modificar tasca



Sweet Syncal

New task

Name

Description

Place

Calendar: C

Status: Not started

Start date: 18/12/2012

Start hour: 20:11

End date: 18/12/2012

End hour: 21:11

Add participants

Cancel Send

Figura 158: Vista crear tasca

En aquesta interfície es mostra la informació de la tasca juntament amb els seus participants, tasques precedents i tasques següents.

Si es fa clic en algun dels títols de participants, precedents o següents es mostrarà el llistat del títol clicat i s'amagaran la resta de llistes.

El menú conté les opcions següents:

**-Modificar tasca:** Mostra la interfície de modificació (apartat 4.4).

**-Eliminar tasca:** Elimina la tasca.

**-Afegir participants:** Mostra la interfície d'addició de participants (apartat 4.5).

**-Afegir precedents:** Mostra la interfície d'addició de precedents (apartat 4.5).

La interfície mostrada a l'esquerra correspon a la interfície de creació de tasques. La interfície de modificació de tasques és exactament igual però sense el camp calendari ni el botó d'afegir participants. Això és així perquè no es pot canviar el calendari d'una tasca un cop creada i perquè els participants es modifiquen des d'altres interfícies.

Quan es prem el botó d'afegir participants es mostrarà la interfície d'addició de participants (apartat 4.5).

Al prémer els camps de data o hora es mostrarà un *dialog* on es podran escollir els valors desitjats.

## 4.5 Afegir participants i Afegir precedents



*Figura 159: Vistes afegir participants i Afegir precedents*

A l'esquerra, interfície per afegir participants a una tasca. A la dreta, interfície per afegir precedents.

Quan s'escriu text al camp del nom d'usuari el sistema mostrarà les possibles opcions amb el text introduït. Al prémer el botó + s'afegirà l'usuari escrit al llistat de futurs participants (si el valor és vàlid). Al prémer el botó *Clear List* es buida el llistat de futurs participants. La pestanya d'afegir grups és exactament igual a la d'afegir usuaris. Quan es prem enviar s'afegeixen tots els usuaris i grups seleccionats.

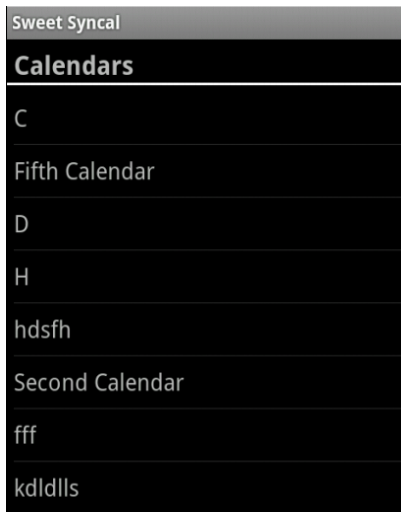
El funcionament de la interfície d'afegir precedents és molt semblant, però enlloc d'haver d'escriure les tasques s'han de seleccionar. Igualment, al prémer el botó + s'afegeixen al llistat de futures precedents, i si es prem el botó *Clear List* es buida el llistat (el botó només és visible si s'ha seleccionat alguna tasca). Quan es prem enviar s'afegeixen totes les tasques seleccionades.



## 5. Gestió de calendaris

### 5.1 Llistat de calendaris

La interfície de llistat de calendaris es mostra quan l'usuari escull l'opció pertinent al menú principal.

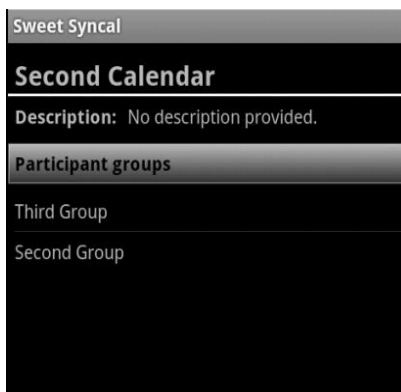


*Figura 160: Vista llistar calendaris*

En aquesta interfície el menú únicament conté l'opció de crear un nou calendari, que mostra la interfície de creació de calendaris (apartat 5.3).

Quan es clica el nom d'un calendari es mostra la interfície que mostra la informació del calendari clicat (apartat 5.2).

### 5.2 Informació calendari



*Figura 161: Vista informació calendari*

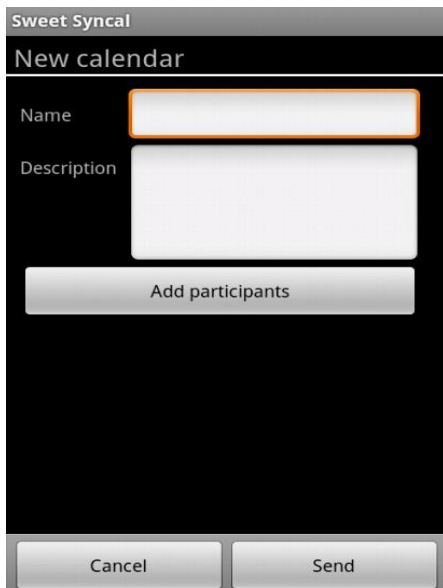
En aquesta interfície es mostra la informació del calendari (nom i descripció) juntament amb el llistat de grups participants.

Si es fa clic en algun dels grups es mostrarà la interfície d'informació del grup clicat (apartat 6.3). Si es manté pressionat un grup apareixerà un menú contextual per eliminar el participant.

El menú conté les opcions següents:

- Modificar calendari:** Mostra la interfície de modificació (apartat 5.3).
- Eliminar calendari:** Elimina el calendari.
- Afegir participants:** Mostra la interfície d'addició de participants (apartat 5.4).

## 5.3 Crear/modificar calendari



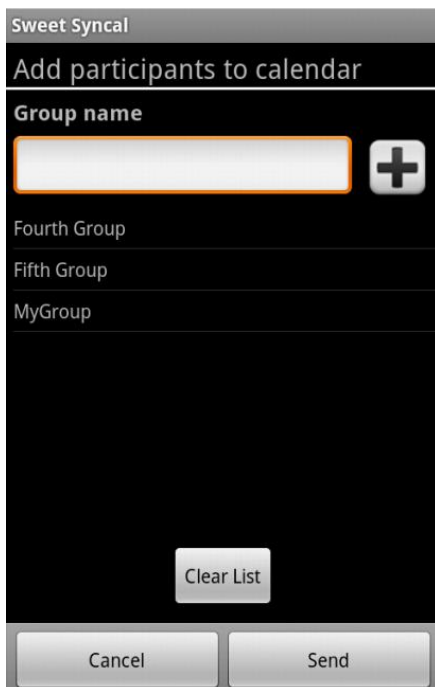
The screenshot shows the 'New calendar' interface. At the top, it says 'Sweet Syncal' and 'New calendar'. Below this, there are two input fields: 'Name' and 'Description'. The 'Name' field is highlighted with an orange border. Below the input fields is a button labeled 'Add participants'. At the bottom of the screen, there are two buttons: 'Cancel' and 'Send'.

Figura 162: Vista crear calendari

La interfície mostrada a l'esquerra correspon a la interfície de creació de calendaris. La interfície de modificació de calendaris és exactament igual però sense el botó d'afegir participants. Això és així perquè els participants es modifiquen des d'altres interfícies.

Quan es prem el botó d'afegir participants es mostrarà la interfície d'addició de participants (apartat 5.4).

## 5.4 Afegir participants



The screenshot shows the 'Add participants to calendar' interface. At the top, it says 'Sweet Syncal' and 'Add participants to calendar'. Below this, there is a 'Group name' label and an input field. To the right of the input field is a plus sign (+) button. Below the input field, there is a list of group names: 'Fourth Group', 'Fifth Group', and 'MyGroup'. At the bottom of the screen, there is a button labeled 'Clear List'. At the very bottom, there are two buttons: 'Cancel' and 'Send'.

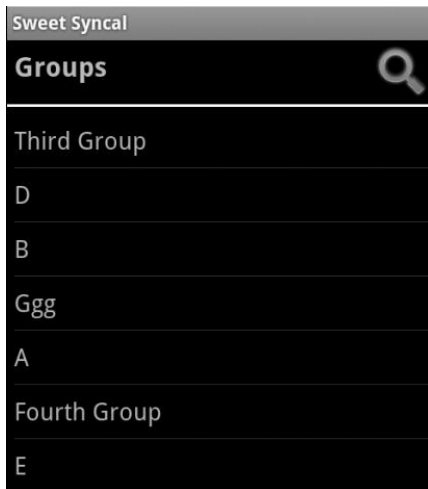
Figura 163: Vista afegir participants

Quan s'escriu text al camp del nom de grup el sistema mostrarà les possibles opcions amb el text introduït. Al prémer el botó + s'afegirà el grup escrit al llistat de futurs participants (si el valor és vàlid). Al prémer el botó *Clear List* es buida el llistat de futurs participants. Quan es prem enviar s'afegeixen tots els grups seleccionats.

## 6. Gestió de grups

### 6.1 Llistat de grups

La interfície de llistat de grups es mostra quan l'usuari escull l'opció pertinent al menú principal.



*Figura 164: Vista llistar grups*

En aquesta interfície el menú conté les opcions:

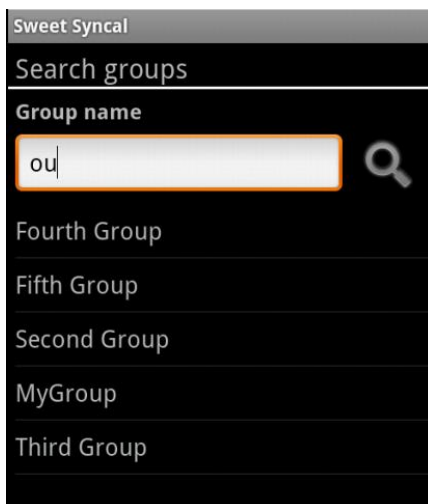
**-Nou grup:** Mostra la interfície per crear un grup (apartat 6.4).

**-Cercar grup:** Mostra la interfície de cercar un grup (apartat 6.2).

Quan es fa clic el nom d'un grup es mostra la interfície d'informació del grup clicat (apartat 6.3).

Quan es prem la icona en forma de lupa es mostra la interfície de cercar un grup (apartat 6.2).

### 6.2 Cercar grup



*Figura 165: Vista cercar grup*

Com el seu nom indica, aquesta interfície serveix per cercar grups.

Quan es prem la icona en forma de lupa es mostrarà un llistat amb tots els grups que continguin en el seu nom els caràcters introduïts en el camp de text.

Si es fa clic en algun dels grups es mostrarà la interfície d'informació del grup clicat (apartat 6.3).

## 6.3 Informació grup

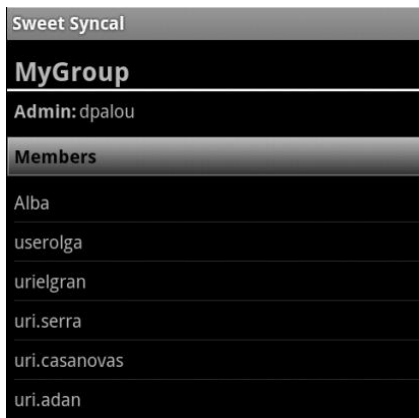


Figura 166: Vista informació grup

En aquesta interfície es mostra la informació del grup juntament amb el llistat de membres.

Si es manté pressionat algun dels membres apareix un menú contextual amb les opcions:

**-Eliminar membre:** Elimina el membre del grup.  
**-Fer administrador:** El membre passa a ser administrador del grup.

Si un usuari que no és l'administrador escull alguna d'aquestes opcions se li notificarà que no n'és l'administrador.

El menú conté les opcions:

- Canviar nom:** Mostra un *dialog* amb un camp de text per canviar el nom del grup.
- Sortir del grup:** L'usuari deixa de participar al grup i als seus calendaris i tasques.
- Afegir membres:** Mostra la interfície d'addició de membres (apartat 6.5).
- Veure sol·licituds:** Mostra la interfície de llistat de sol·licituds (apartat 6.6).

Les opcions canviar nom, afegir membres i veure sol·licituds només es poden utilitzar si l'usuari és l'administrador del grup. Addicionalment, si no hi ha sol·licituds pendents per aquest grup l'opció veure sol·licituds ho notificarà a l'usuari i no mostrarà cap nova interfície.

Si l'usuari no pertany al grup mostrat apareixerà un botó a sota del nom de l'administrador del grup. Aquest botó s'utilitzarà per enviar una sol·licitud per entrar al grup o, en cas que l'usuari ja hagi enviat una sol·licitud, per cancel·lar la sol·licitud existent.

## 6.4 Crear grup

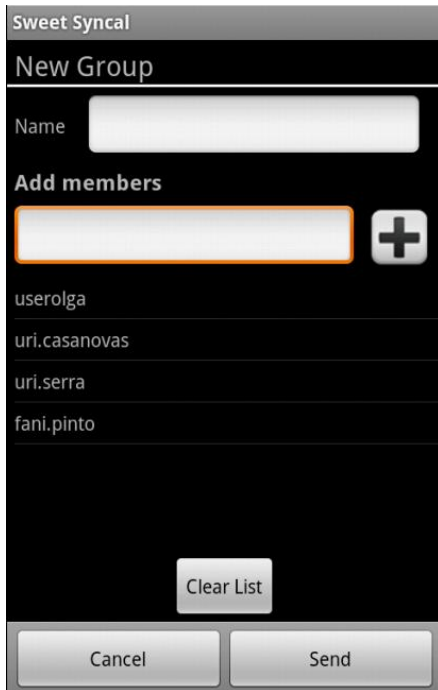


Figura 167: Vista crear grup

La interfície mostrada a l'esquerra correspon a la interfície de creació de grups.

Quan s'escriu text al camp d'afegir membres el sistema mostrarà les possibles opcions amb el text introduït. Al prémer el botó + s'afegirà l'usuari escrit al llistat de futurs membres (si el valor és vàlid). Al prémer el botó *Clear List* es buida el llistat de futurs participants. Quan es prem enviar es crea el grup i s'hi afegeixen els membres seleccionats. L'usuari que crea el grup sempre és membre del grup i n'és l'administrador.

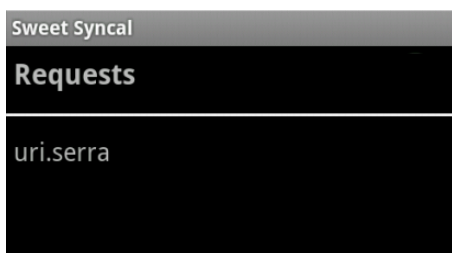
## 6.5 Afegir membres



Figura 168: Vista afegir membres

Quan s'escriu text al camp del nom d'usuari el sistema mostrarà les possibles opcions amb el text introduït. Al prémer el botó + s'afegirà el grup escrit al llistat de futurs membres (si el valor és vàlid). Al prémer el botó *Clear List* es buida el llistat de futurs membres. Quan es prem enviar s'afegeixen tots els usuaris seleccionats.

## 6.6 Llistat de sol·licituds



La interfície mostra el llistat d'usuaris que han sol·licitud entrar al grup. Quan es prem en un usuari apareix un *dialog* que permet acceptar o rebutjar la sol·licitud seleccionada. Si s'accepta una sol·licitud, l'usuari corresponent passa a ser membre del grup.

Figura 169: Vista llistat de sol·licituds

## 7. Gestió d'esdeveniments, votacions i preferències

### 7.1 Llistat d'esdeveniments

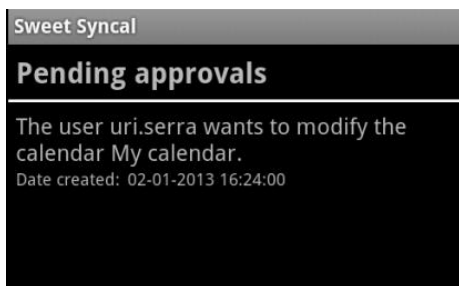


La interfície de llistat de grups es mostra quan l'usuari escull l'opció pertinent al menú principal.

Quan es prem un esdeveniment es mostra la interfície d'informació de l'element afectat, és a dir, informació d'una tasca (apartat 4.3), d'un calendari (apartat 5.2) o d'un grup (apartat 6.3).

Figura 170: Vista llistat d'esdeveniments

### 7.2 Llistat de pending approvals



La interfície de llistat de pending approvals es mostra quan l'usuari escull l'opció pertinent al menú principal.

Quan es prem un pending approval es mostra la interfície d'informació del pending approval premut (apartat 7.3).

Figura 171: Vista llistat de pending approvals

## 7.3 Informació pending approval

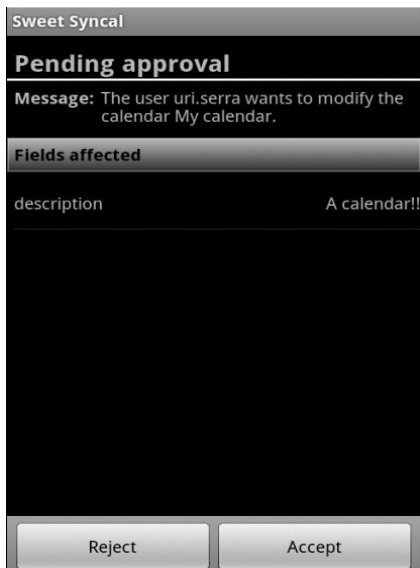


Figura 172: Vista informació pending approval

En aquesta interfície es mostra la informació d'un pending approval. Concretament, es pot veure el missatge del pending approval, els camps afectats i el valor que es vol que prengui els camps en qüestió. En el cas de l'exemple que es pot observar a l'esquerra, l'usuari *uri.serra* vol que la descripció del calendari *My calendar* passi a ser "A calendar!!".

Amb els botons de la part inferior de la interfície es pot acceptar o rebutjar el pending approval. Si ja s'ha acceptat o rebutjat, aquests dos botons són reemplaçats amb un text que indica la decisió presa.

## 7.4 Llistat de meetings

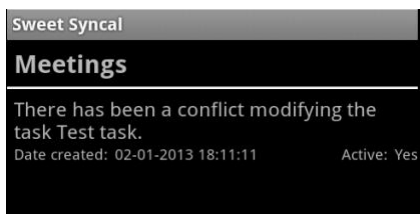


Figura 173: Vista llistar meetings

La interfície de llistat de meetings es mostra quan l'usuari escull l'opció pertinent al menú principal.

Quan es prem un meeting es mostra la interfície d'informació del meeting premut (apartat 7.5).

## 7.5 Informació meeting

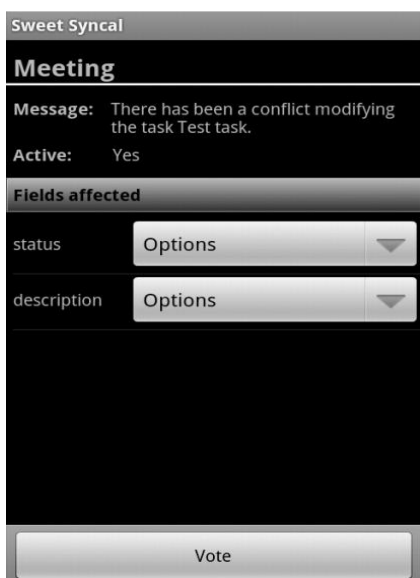


Figura 174: Vista informació meeting

En aquesta interfície es mostra la informació d'un meeting i permet votar les opcions que es prefereixen per cada camp afectat. En el cas de l'exemple que es pot observar a l'esquerra, hi ha hagut un conflicte amb l'estat i la descripció de la tasca *Test task*. Les llistes desplegable contenen els possibles valors que es poden votar.

Amb el botó de la part inferior es pot desar el vot amb les opcions seleccionades. És imprescindible que s'hagi seleccionat un valor per cada camp afectat. En cas que l'usuari ja hagi votat en aquest meeting, aquest botó es substituirà per un botó per eliminar el vot.

## 7.6 Estadístiques d'usuari

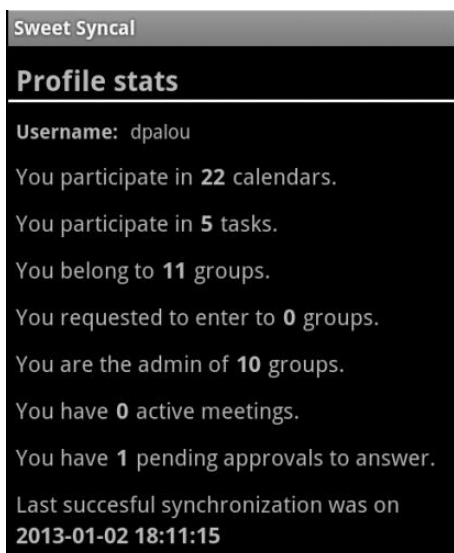


Figura 175: Vista estadístiques d'usuari

La interfície d'estadístiques d'usuari es mostra quan l'usuari escull l'opció pertinent al menú principal.

Aquesta interfície mostra algunes dades referents a l'usuari que utilitza l'aplicació.

## 7.7 Preferències

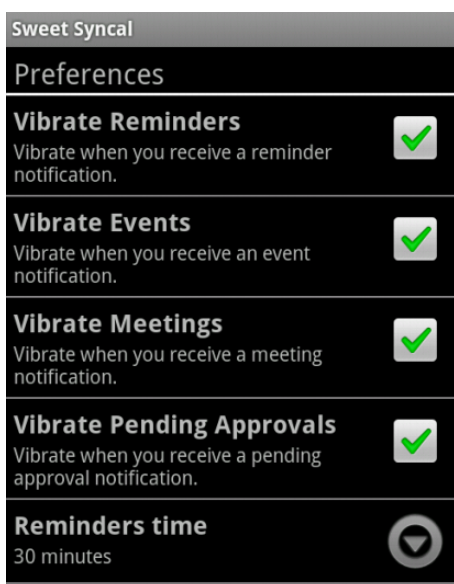


Figura 176: Vista preferències

La interfície de preferències es mostra quan l'usuari escull l'opció pertinent al menú principal.

Les modificacions en les preferències es desen automàticament cada cop que es realitza un canvi, no és necessari indicar-ho explícitament.

Quan es prem l'apartat de "Reminders time" s'obre un *dialog* que permet escollir el nombre i unitat del temps dels recordatoris o, en cas que l'usuari ho vulgui així, també permet desactivar els recordatoris.

## 7.8 Altres opcions del menú principal

**Full synchronization:** Realitza una sincronització completa amb el servidor.

**Logout:** Es tanca la sessió de l'usuari a l'aplicació, de manera que si vol tornar a utilitzar-la haurà d'accedir-ho un altre cop (apartat 3).



# Conclusions

En aquest apartat es recullen algunes reflexions finals sobre el projecte: objectius aconseguits, problemes trobats, possibles millores, opinió personal, etc.

## 1. Objectius aconseguits

Un cop finalitzades totes les fases de desenvolupament del projecte podem afirmar que s'han complert els objectius marcats a l'inici del projecte.

S'ha donat continuïtat al projecte d'en Vicent Seguí, reutilitzant tantes classes com s'ha pogut (autenticació i comunicació amb l'API de Sugar) i mantenint la tecnologia (sistema operatiu Android, SugarCRM al servidor).

S'ha desenvolupat una aplicació que permet als usuaris crear i compartir calendaris i tasques, orientada a afavorir el treball en grup i tractant els possibles conflictes que puguin sorgir.

Utilitzant l'experimentació s'han estudiat i comprovat els avantatges i inconvenients de diferents variacions i mètodes en el procés de sincronització. Tot i que els experiments realitzats només són un subconjunt de tots els possibles, creiem que són dels més importants i que els seus resultats són rellevants per a obtenir una aplicació més eficient i beneficiosa per l'usuari.

L'últim objectiu és més difícil de valorar, ja que és el més subjectiu. No és fàcil decidir si una aplicació està programada de manera que facilita possibles ampliacions. No obstant, durant la implementació de l'aplicació s'ha intentat complir aquest objectiu fent les funcionalitats i interfícies el màxim d'independents i comentant en detall el codi per tal de facilitar la seva comprensió. És per aquest motiu que considero que aquest objectiu s'ha complert.

## 2. Futures ampliacions

Malgrat que s'han complert els objectius fixats, sempre es poden seguir fent millores al sistema.

### Interfície

Com ja s'ha comentat a l'apartat d'implementació, la interfície escollida és senzilla i funcional, com la utilitzada en el projecte de Vicent Seguí al qual s'ha donat continuïtat, potser fins i tot massa senzilla. Els colors no criden massa l'atenció i l'estètica correspon a versions d'Android que ja es comencen a considerar antigues (2.x). Es podria millorar bastant l'estètica de les vistes, obtenint una aplicació molt més atractiva per a l'usuari sense perdre funcionalitats ni fer-la menys intuïtiva.

### Idiomes

Actualment l'aplicació està en anglès ja que aquest és l'idioma més estès, però seria interessant que pogués estar disponible en altres idiomes per tal d'ampliar el ventall d'usuaris que puguin utilitzar l'aplicació. A l'implementar l'aplicació s'ha seguit la mecànica d'Android de definir totes les frases de la interfície al fitxer *strings.xml*, fet que facilita la traducció d'aquesta.

## **Optimització de la sincronització amb GCM**

La sincronització utilitzant el servei GCM de Google podria ser bastant més eficient del que és actualment. Com ja s'ha dit en més d'una ocasió, quan es fa un canvi al servidor aquest es limita a notificar els telèfons corresponents, i són aquestes els que a continuació fan una petició de canvis. Aquest sistema es podria millorar fent que, quan hi ha un canvi al servidor, aquest automàticament l'envii als telèfons enlloc de simplement notificar-los. Llavors, per què no s'ha fet així directament?

El motiu és que el servei GCM va ser l'últim element que es va afegir a l'aplicació, no estava previst utilitzar-lo inicialment, ni tan sols coneixia la seva existència. Però quan es va instal·lar l'aplicació en un dispositiu i es va veure que el consum de bateria utilitzant un SyncAdapter era inacceptable es va haver de buscar una alternativa, i la solució va ser aquest servei de Google. Com que la quantitat d'hores dedicades a la implementació ja superava en molt les previstes inicialment es va preferir implementar aquest mecanisme més senzill però més lent i deixar l'optimització com a ampliació futura.

## **Més compatibilitat amb l'accés web**

El nostre sistema està pensat de manera que tots els canvis es facin a través del dispositiu mòbil, no a través de SugarCRM directament, ja que la majoria de canvis fets han de realitzar un seguit d'operacions i/o comprovacions que la interfície gràfica de Sugar, com és lògic, no ofereix. Es podria mirar a veure si es pot modificar la interfície gràfica de Sugar per tal que alguns canvis utilitzin les funcionalitats de l'API en PHP programada per a aquest projecte enlloc de realitzar les operacions habituals i, en cas que no es pugui, es podria implementar un accés web amb una interfície com la de Sugar que utilitzés l'API mencionada. Òbviament, aquesta segona opció du associada una càrrega de treball important.

## **3. Valoració personal**

Considero que l'elaboració d'aquest projecte ha sigut molt profitós per mi.

Per una banda, durant la carrera mai s'havia dut a terme un projecte complet com en aquest cas, amb totes les seves fases. En algunes assignatures es realitzava únicament la part d'especificació i disseny, mentre que en d'altres aquesta part s'ignorava o es realitzava de manera molt breu i superficial. Per primera vegada he desenvolupat un projecte amb totes les seves fases, de principi a fi, i això ha permès consolidar i repassar molts dels conceptes apresos durant la carrera.

Per altra banda, per fer aquest projecte he hagut d'investigar i aprendre alguns sistemes, tecnologies o llibreries que fins ara no havia tocat mai i que poden resultar molt útils en el meu futur laboral: Android, SugarCRM, JXTA, etc. A més, tota aquesta investigació i aprenentatge s'ha hagut de fer pel meu compte, sense apunts a seguir com fins ara, que és el que ens trobarem un cop entrem al món laboral. Valoro molt positivament aquesta experiència, i considero que és una de les lliçons més importants que he après al desenvolupar el projecte.

En conclusió, el projecte m'ha suposat una experiència molt positiva de cara al meu futur i m'ha permès aprendre i aplicar noves tecnologies que em poden ser molt útils.

# **Annex**

Aquest apartat conté informació complementària d'ajut en el projecte juntament amb la bibliografia.

# 1. Resultats detallats de l'experimentació

A l'apartat d'experimentació únicament s'han recollit els resultats finals de les proves realitzades. En aquest apartat es presenten totes les dades obtingudes durant l'experimentació i que s'han utilitzat per a obtenir els resultats presentats.

## 1.1 Client-servidor vs P2P

Com s'ha comentat en el propi experiment, el temps de cada rèplica és la mitjana de 10 execucions consecutives. A continuació es pot observar el temps de cadascuna d'aquestes repeticions.

		1	2	3	4	5	6	7	8	9	10	Mitjana
REP1	C-S	4458	4617	4190	4448	3238	3878	4154	3546	3318	3538	3938,5
	P2P	1997	2266	2726	3511	3207	2635	2385	2245	3795	2741	2750,8
REP2	C-S	4447	4074	6144	4123	2391	3514	2966	2743	2884	5017	3830,3
	P2P	2911	2429	2861	2581	2941	2799	3052	2963	2883	2357	2777,7
REP3	C-S	17175	26376	11593	12797	12693	29689	15122	21210	11461	21819	17993,5
	P2P	9901	10774	14132	10221	14541	11616	11635	10764	12855	9653	11609,2
REP4	C-S	2678	2430	2541	4246	2526	2757	6639	3353	2361	2892	3242,3
	P2P	2722	2864	3668	2324	2713	2902	2493	2506	2591	2878	2766,1
REP5	C-S	3799	4920	2273	2657	2345	2502	3217	5821	5116	2324	3497,4
	P2P	2554	2843	2523	2963	3002	2763	2607	2771	3257	2829	2811,2
REP6	C-S	6020	4427	5170	4340	4743	2829	6810	4728	7332	4194	5059,3
	P2P	8296	7658	9905	8967	8420	7713	9484	10096	9612	9378	8952,9
REP7	C-S	6580	5132	11164	3317	7931	7734	13121	2852	3076	3965	6487,2
	P2P	9901	8813	9121	9875	9270	10700	9664	10384	8603	9055	9538,6

Taula 175: Resultats detallats experiment 1

## 1.2 Relationships vs Modules

Com en l'experiment anterior, el temps de cada rèplica és la mitjana de 10 execucions consecutives. A continuació es poden observar en detall.

		1	2	3	4	5	6	7	8	9	10	Mitjana
REP1	Moduls	0,16013813	0,16154218	0,15524197	0,16071987	0,15856886	0,16248798	0,15882087	0,15678906	0,15967298	0,15826106	159,224296
	Rels	0,09409809	0,0946331	0,09267211	0,09151411	0,09513497	0,09238911	0,09210682	0,09434581	0,09456801	0,09509802	93,6560154
REP2	Moduls	0,10727286	0,10987687	0,10850382	0,10645485	0,10690522	0,11011386	0,10827518	0,11071111	0,10609293	0,10679412	108,10008
	Rels	0,03183198	0,03165603	0,03182697	0,0333209	0,03309107	0,03372598	0,03330112	0,03315187	0,03108788	0,03236008	32,5353861
REP3	Moduls	0,56149197	0,58601284	0,56531096	0,56455898	0,56782413	0,5554049	0,57356191	0,56170607	0,56249094	0,555058	565,342069
	Rels	0,30339503	0,30218816	0,29918003	0,29984188	0,30546904	0,3083818	0,30846906	0,34225512	0,31163788	0,30145907	308,227706
REP4	Moduls	0,26867509	0,27012587	0,27188897	0,27382708	0,27398896	0,27176404	0,26504397	0,26449203	0,27341795	0,269274	270,249796
	Rels	0,21119785	0,2085278	0,21043153	0,21267057	0,20977964	0,20674524	0,21343517	0,21204486	0,20668745	0,20841365	209,993377
REP5	Mod	1,51639104	1,529603	1,52995086	1,53077006	1,50083995	1,4969418	1,55559516	1,56275392	1,50268102	1,53749704	1526,30239
	Rel	2,22699404	2,21693993	2,19511199	2,21742702	2,19084287	2,20444393	2,26964617	2,23719692	2,23451805	2,13840199	2213,15229
REP6	Mod	1,53976297	1,59544301	1,55075192	1,62155199	1,58739996	1,58530998	1,63125396	1,57734203	1,53564787	1,55741215	1578,18758
	Rel	1,9288981	1,88831401	1,89032412	1,92610002	1,92937303	1,93028307	1,92718887	1,96518016	1,87258387	1,965832	1922,40772
REP7	Mod	0,13603401	0,14137793	0,13816094	0,14126706	0,14089704	0,14277101	0,13989305	0,14136386	0,14108586	0,142519	140,536976
	Rel	0,14320612	0,14619303	0,14331985	0,1434989	0,13961697	0,147928	0,14110398	0,13693094	0,13875389	0,14180803	142,23597
REP8	Mod	0,07891202	0,09665608	0,100842	0,07546186	0,07389712	0,07422495	0,09043193	0,08348489	0,07395196	0,0968349	84,4697714
	Rel 1	0,12991691	0,14849401	0,13908792	0,14291191	0,12460399	0,130548	0,12719584	0,13950396	0,13269591	0,12251306	133,747148
	Rel 2	0,0730269	0,1010139	0,07188487	0,07615685	0,06306791	0,08957505	0,07838798	0,07591891	0,1014719	0,07775784	80,826211
REP9	Mod	0,10352683	0,10084796	0,08117509	0,12104917	0,07295895	0,07429385	0,09918118	0,07906294	0,09615803	0,1148119	94,3065882
	Rel 1	0,12564802	0,09847999	0,1257	0,12127519	0,1280551	0,09449387	0,11630583	0,10476279	0,14910388	0,10473013	116,855478
	Rel 2	0,07005715	0,06810093	0,09598589	0,09340811	0,10435605	0,096591	0,09579182	0,08173513	0,07174301	0,06275105	84,0520144
REP10	Mod	0,03721881	0,03717899	0,03649187	0,03804588	0,03657579	0,03503609	0,03552103	0,03573203	0,03628206	0,03542399	36,3506556
	Rel 1	0,06021309	0,06030011	0,06121111	0,06069684	0,0602212	0,05989194	0,05813694	0,05985093	0,06026888	0,05996299	60,0754023
	Rel 2	0,08081388	0,07176805	0,07692599	0,07584906	0,08445716	0,06506896	0,071141	0,09327507	0,07839108	0,0990448	79,6735048
REP11	Mod	0,23779488	0,2379241	0,23477817	0,18118405	0,22668099	0,24509692	0,21525979	0,21961403	0,19458699	0,23469901	222,761893
	Rel 1	0,30108595	0,3028791	0,24635696	0,30174804	0,26151395	0,28110504	0,29945803	0,27786899	0,27526498	0,29857183	284,585285
	Rel 2	0,2295239	0,21983388	0,26846	0,2265389	0,22947502	0,25347996	0,2089541	0,22760606	0,22645211	0,22713518	231,74591

Taula 176: Resultats detallats temps total experiment 2

A continuació es pot apreciar el temps de les rèpliques 5, 6 i 7 en detall, desglossat en cadascuna de les fases en què es divideix l'operació. Cal aclarir, però, que en aquesta taula no hi són presents totes les fases de l'operació, només aquelles més importants. L'ordre de les taules és el següent: mòduls rèplica 5, *relationships* rèplica 5, mòduls rèplica 6, *relationships* rèplica 6, mòduls rèplica 7 i *relationships* rèplica 7.

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,489872	0,4828651	0,48487496	0,48992896	0,47656012	0,47961688	0,49340701	0,50065899	0,4767139	0,49496698	0,48694649
My Requests	0,0183172	0,01833105	0,01890898	0,01843619	0,01840901	0,01801205	0,01861906	0,01852703	0,01790094	0,01873398	0,01841955
Calendars	0,2269771	0,2284348	0,22749209	0,23067904	0,22316098	0,228652	0,23400187	0,22783184	0,22579408	0,23032093	0,22833447
Tasks	0,4618971	0,47410297	0,47534919	0,469414	0,46303797	0,45861411	0,48235512	0,48586607	0,46154904	0,47291493	0,47051005
Ids tasks	0,1131091	0,11467218	0,11291218	0,1136291	0,112993	0,10761905	0,11506319	0,11530113	0,112746	0,11364508	0,113169
Events	0,067641	0,06909609	0,07089901	0,06927299	0,06917191	0,0675292	0,06984496	0,07054996	0,06873608	0,06743407	0,06901753
PR	0,035584	0,03491592	0,03502893	0,03502297	0,03445983	0,03351116	0,03482413	0,03608704	0,03582597	0,03584981	0,03511097

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,5932281	0,59981704	0,59418297	0,59381485	0,60232711	0,59678221	0,59040403	0,6098721	0,59712601	0,60609198	0,59836464
My Requests	0,0269248	0,02642894	0,02793384	0,02810502	0,02649093	0,02713799	0,02675414	0,02793002	0,028157	0,02625513	0,02721179
Calendars	0,2450719	0,24731207	0,256675	0,25336194	0,24187398	0,2485218	0,25338292	0,25268483	0,25443506	0,23765397	0,24909735
Tasks	0,650502	0,66917491	0,65701008	0,66328621	0,65188813	0,65913892	0,64744806	0,66939902	0,66870308	0,6293931	0,65659435
Ids tasks	0,2093141	0,208359	0,20390892	0,21347904	0,21052289	0,21025991	0,20248103	0,20709991	0,21335316	0,19877696	0,20775549
Events	0,0999548	0,1014092	0,09814	0,10092211	0,0986979	0,10006189	0,09581399	0,10374594	0,10176706	0,09711099	0,09976239
PR	0,2591228	0,25883198	0,25093102	0,26011491	0,257061	0,25802183	0,25805402	0,25801206	0,26212597	0,24108815	0,25633638

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,464361	0,49316692	0,47947097	0,49626112	0,48771191	0,48673511	0,5086	0,49576497	0,46399808	0,47652388	0,48525939
My Requests	0,0174289	0,01893115	0,01810098	0,01828218	0,01856303	0,01734591	0,01895094	0,01823592	0,01768899	0,01914597	0,01826739
Calendars	0,198983	0,21172619	0,20570612	0,20829916	0,21059203	0,20762491	0,20901179	0,21058893	0,20629597	0,21068287	0,20795109
Tasks	0,4688659	0,47418404	0,45811486	0,49817204	0,47908211	0,47738791	0,48168206	0,46280503	0,46351004	0,46581817	0,47296221
Ids tasks	0,111923	0,11453915	0,1119082	0,11611104	0,11231804	0,11639786	0,12045193	0,11082101	0,11048603	0,11318588	0,11381421
Events	0,0414481	0,03996897	0,03886914	0,03874207	0,03918195	0,03941917	0,04369998	0,03875995	0,03820491	0,03921199	0,03975062
PR	0,036674	0,037009	0,03474402	0,035537	0,03631496	0,03596187	0,03823185	0,03584909	0,03455591	0,0349741	0,03598518

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,5061631	0,49278998	0,48804688	0,52149415	0,50559402	0,49597597	0,51150298	0,52015018	0,49495912	0,50957489	0,50462513
My Requests	0,0159638	0,015481	0,01596403	0,01563501	0,01503515	0,01537299	0,01602197	0,0169518	0,01542902	0,01636314	0,01582179
Calendars	0,186085	0,17803192	0,18102407	0,17490292	0,17857003	0,18572307	0,18078613	0,1900878	0,17798901	0,18786192	0,18210618
Tasks	0,5260298	0,51413989	0,51093483	0,5168798	0,52664495	0,53593707	0,52829599	0,53335094	0,52258492	0,53224993	0,52470481
Ids tasks	0,1995358	0,19969082	0,197294	0,20195508	0,20341897	0,20187712	0,20230508	0,20019698	0,19428587	0,2046051	0,20051649
Events	0,017112	0,01757002	0,01669598	0,01642799	0,01717401	0,01589608	0,016294	0,01747298	0,01694393	0,01664591	0,01682329
PR	0,2275221	0,2238059	0,23518801	0,23351598	0,23411894	0,22870708	0,22670412	0,23254395	0,20956898	0,235502	0,22871771

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,013572	0,01255894	0,01254702	0,01227093	0,01231003	0,01243591	0,01206183	0,01230407	0,01250911	0,01224899	0,01248188
My Requests	0,0105848	0,01154017	0,01027584	0,01089716	0,01134515	0,01133204	0,01108909	0,01104903	0,01136899	0,01085281	0,01103351
Calendars	0,0314739	0,03260994	0,03213501	0,03324699	0,03358793	0,0329299	0,03238201	0,03217387	0,03240895	0,03302288	0,03259714
Tasks	0,0116091	0,01157022	0,01119399	0,01108789	0,01164412	0,01123095	0,01107693	0,01115894	0,01155186	0,01162601	0,011375
Ids tasks	0,0119469	0,01302695	0,01186705	0,01185513	0,01205587	0,01322412	0,01289916	0,01232386	0,01303101	0,01224899	0,01244791
Events	0,010469	0,01081204	0,01131797	0,01135778	0,01085615	0,01124978	0,01070595	0,01158404	0,01105809	0,01118493	0,01105957
PR	0,0164001	0,01739788	0,01690507	0,01776886	0,0171299	0,01818895	0,01746893	0,017133	0,01684093	0,01796818	0,01732018

	1	2	3	4	5	6	7	8	9	10	Mitjana
Groups	0,013931	0,01379704	0,01382089	0,01426816	0,013448	0,01285696	0,013726	0,01272798	0,01354909	0,01406789	0,0136193
My Requests	0,011996	0,01204491	0,01167893	0,01200914	0,01113892	0,01086402	0,01152515	0,01114702	0,01198387	0,01112103	0,0115509
Calendars	0,0194781	0,01953316	0,01948714	0,01947117	0,01907396	0,01784205	0,01890683	0,0190661	0,01826286	0,01890016	0,01900215
Tasks	0,007349	0,008636	0,00862503	0,00778198	0,00746393	0,00807285	0,008178	0,00757003	0,00806594	0,00823808	0,00799809
Ids tasks	0,020591	0,02037811	0,0193119	0,01942682	0,01899815	0,01809001	0,01937485	0,01926804	0,019063	0,01988196	0,01943839
Events	0,01215	0,01167011	0,01161098	0,01178193	0,0122509	0,01031899	0,01207304	0,01171017	0,01088881	0,01124501	0,011157
PR	0,025053	0,02747488	0,02511811	0,02603579	0,02481508	0,02315688	0,02418303	0,02399993	0,02454114	0,02517486	0,02495527

Taula 177: Resultats detallats temps rèpliques 5, 6 i 7 experiment 2

### 1.3 Desar només tasques vs desar-ho tot

Un cop més, el temps de cada rèplica és la mitjana de 10 execucions consecutives. A continuació es pot observar en detall el temps de cada repetició.

		1	2	3	4	5	6	7	8	9	10	Mitjana
REP1	All	1381	1106	1647	1377	1490	1617	1407	1344	1741	1462	1457,2
	Tasks	734	691	909	663	695	664	1100	725	1129	718	802,8
REP2	All	2305	2562	2376	2574	2361	2656	2760	2270	2477	2664	2500,5
	Tasks	2032	1917	1868	1927	2149	2053	1925	2084	1963	1815	1973,3
REP3	All	2380	2378	2146	2646	1937	1905	2451	2298	2161	2539	2284,1
	Tasks	1217	1064	853	807	833	1039	801	886	1066	845	941,1
REP4	All	3660	4108	4062	4035	3584	4107	4244	4216	4290	3729	4003,5
	Tasks	3019	3003	3465	3032	3200	3743	3053	3377	2880	3184	3195,6
REP5	All	6278	6212	6063	6455	5918	6501	6576	6402	6519	6637	6356,1
	Tasks	5576	5178	5352	5643	5578	5811	6220	6448	6326	6466	5859,8
REP6	All	5780	3853	4790	3860	3973	4188	4123	4017	4044	4274	4290,2
	Tasks	1167	1008	995	1072	1031	1294	1067	1087	919	1003	1064,3
REP7	All	6924	7954	6802	7191	7716	7196	7372	7193	7282	7218	7284,8
	Tasks	5607	5436	5584	6162	5457	6142	5626	5844	5386	5442	5668,6
REP8	All	29935	22857	21874	22937	22603	21521	25610	28912	22371	25092	24371,2
	Tasks	24381	24536	24844	26098	25668	23910	27366	25981	25054	28066	25590,4

Taula 178: Resultats detallats temps experiment 4

## 2. Bibliografia

A continuació es presenten alguns llibres, webs i documentació utilitzats per a desenvolupar el projecte. Òbviament no hi són presents tots els webs consultats ja que el llistat és molt llarg, però sí aquells més importants i que poden resultar més útils per a entendre i/o ampliar el projecte.

- Projecte "Sistemes Android i treball en grup" de Vicent Seguí.
- Jérôme Verstrynge, *Practical JXTA II: Cracking the P2P puzzle*. Editor: lulu.com. 2010.
- Apunts assignatura "Taller de programació d'aplicacions Android per a Google Phones (TPAAGP)".
- Apunts assignatura "Base de dades (BD)".
- Android: <http://developer.android.com/>
- Resolució de dubtes: <http://stackoverflow.com/>
- Introducció Android Binding:  
<http://www.codeproject.com/Articles/145203/Android-Binding-Introduction>
- Codi Android Binding: <http://code.google.com/p/android-binding/>
- SugarCRM: <http://www.sugarcrm.com/>
- Memcached: <http://memcached.org/>
- PeerDroid: <http://code.google.com/p/peerdroid/>
- SyncAdapter: <https://sites.google.com/site/andsamples/concept-of-syncadapter-androidcontentabstractthreadedsyncadapter>
- SyncAdapter: <http://www.c99.org/2010/01/23/writing-an-android-sync-provider-part-1/>
- Informació general: <http://es.wikipedia.org>