

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria Tècnica d'Informàtica de Gestió

MONITORITZACIÓ TRAILWALKER

Alumne/a: Roger Riera Díaz

Dirigit per: Francesc Puigvert Pell

Co-dirigit per: Antoni Pérez Navarro

CURS 2012-13 (Febrer)

Llicència (Creative Commons)

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-los i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.

Agraïments

Voldria agrair enormement el suport rebut pel director del projecte, en Francesc Puigvert, ja que sense la seva ajuda i les seves indicacions no m'hagués estat possible tirar endavant aquest treball. També agraeixo el suport del responsable d'àrea SIG de la UOC, en Toni Pérez, per ajudar-me a lidiar amb la burocràcia d'Apple. Però sobretot no puc deixar de mencionar la paciència, la comprensió i el sacrifici de la meva dona Graciela i dels meus pares Albert i Pepa durant el transcurs d'aquest projecte, gràcies a ells, m'ha resultat una tasca molt més fàcil de dur a terme.

Taula de continguts

1. INTRODUCCIÓ	1
1.1. El projecte i el client	1
1.2. Els sistemes SIG	2
1.3. L'entorn tecnològic iOS	2
2. OBJECTIUS	3
2.1. Objectius Generals	3
2.2. Objectius Específics	3
3. FITES I TEMPORITZACIÓ	4
4. DIAGRAMA DE GANTT	7
5. INCIDÈNCIES I RISCOS	9
5.1. Problemes amb l'equip de treball	9
5.2. Problemes amb la connexió de xarxa	9
5.3. Desplaçament per motius laborals	10
5.4. Baixa per malaltia	10
6. MATERIAL REQUERIT	11
6.1. Programari documentació	11
6.2. Programari desenvolupament	11
6.3. Maquinari	11
7. ANÀLISIS SOLUCIONS PREVIES	12
7.1. Estudi i conclusions TFC Albert Jordà	12
7.2. Estudi i conclusions TFC Montse Baldomà	15
8. DESCRIPCIÓ DE LA SOLUCIÓ	18
8.1. Arquitectura	18
8.1.1. Sistema iOS	19
8.2. Casos d'ús	20
8.2.1. Procés de captura de la posició	20

8.2.2. Consulta d'informació	22
8.2.3. Configuració de l'aplicació	23
8.3. Diagrames col·laboració i jerarquia	25
8.4. Prototipus de pantalles	26
9. PROGRAMACIÓ EN iOS	29
9.1. Entorn de desenvolupament	29
9.2. Utilització Frameworks	30
9.3. Llicència de desenvolupament	31
9.4. Patrons de disseny	32
9.4.1. Delegació entre classes	32
9.4.2. Classes Singleton	33
9.4.3. Categories	34
9.4.4. Execució en segon pla (Background)	34
9.4.1. Localització i Internacionalització	36
10. BANC DE PROVES	39
10.1. Proves unitàries	39
10.2. Anàlisi de rendiment	42
10.3. Proves de camp	44
10.3.1. Canvi significat d'ubicació	45
10.3.2. Sistema d'ubicació estàndard	46
10.4. Proves d'estrès	47
11. CONCLUSIONS	48
11.1. Sobre la implementació	48
11.1.1. Identificador únic de l'equip	48
11.1.2. Selecció del mètode de captura	49
11.1.3. Selecció de l'idioma de l'aplicació	49
11.2. Sobre els resultats	49
11.3. Millores futures	50

12. BIBLIOGRAFIA	52
13. GLOSSARI	54

Taula d'il·lustracions

Figura 1: GANTT del plà de treball des de l'inici fins el lliurament de la PAC2	7
Figura 2: GANTT del plà de treball des del lliurament de la PAC2 fins la finalització	8
Figura 3: Diagrama d'arquitectura Trailwalker	18
Figura 4: Esquema de capes d'iOS	19
Figura 5: Diagrama casos d'ús de l'aplicació iOS	24
Figura 6: Diagrama patró disseny MVC	25
Figura 7: Diagrama de classes aplicació iOS Trailwalker	26
Figura 8: Exemple de la utilització de frameworks.	30
Figura 9: Exemple de delegació per la classe UIWindow de l'AppKit.	32
Figura 10: Esquema instància de classes Singleton.	34
Figura 11: Transició d'estats d'una app d'iOS.	35
Figura 12: Selecció de nova localització dins Xcode.	37
Figura 13: Opció de configuració de l'idioma en el dispositiu.	38
Figura 14: Cicle de vida típic d'una aplicació informàtica.	39
Figura 15: Resultat execució de test OCUit a Xcode.	42
Figura 16: Gràfica de rendiment d'Instruments per a l'app TrailWalker.	43
Figura 17: Valor de l'anàlisi de rendiment d'Instruments per a les diferents ...	43
Figura 18: Visor web TrailWalker. Mesura d'una ruta pedestre.	45
Figura 19: Visor web TrailWalker. Mesura d'una ruta automobilística.	46

1. INTRODUCCIÓ

1.1. El projecte i el client

Intermón Oxfam és una organització no governamental de cooperació, que treballa en programes de desenvolupament per a un total de 41 països. La seva tasca es centra en els següents principis:

- Defensar la justícia econòmica.
- Treballar en acció humanitària.
- Impulsar ciutadania i governabilitat.
- Promoure serveis socials bàsics.
- Lluitar pel dret de les dones.

La seva principal font de finançament es basa en l'aportació privada dels seus socis, ja sigui en forma de donació puntual o com a quota periòdica. També aconseguen una recaptació considerable, gràcies a les iniciatives solidàries que organitzen.

Una d'aquestes iniciatives és la Trailwalker, que consisteix en una cursa de 100 km a peu per equips, que enguany es celebrarà la seva tercera edició dins del territori català.

Arrel de les experiències prèvies, ha sorgit la necessitat de implementar una solució tecnològica que permeti tenir localitzats els diferents equips participants en tot moment.

A nivell arquitectònic, el projecte requereix implementar una aplicació per a telèfon mòbil iPhone, que s'encarregui d'enviar, durant tota la cursa, la senyal de la posició de l'equip i el nivell de bateria del dispositiu a un servidor habilitat per la UOC.

Mitjançant aquesta informació, es podrà ubicar cada equip participant dins d'un mapa, a temps real.

1.2. Els sistemes SIG

Per dur a terme el projecte descrit, s'haurà de desenvolupar un Sistema d'Informació Geogràfic (S.I.G.).

Els SIG, són qualsevol combinació de programari, maquinari i dades que es relacionen, de tal forma, que permetin la interpretació i l'anàlisi d'una informació de caràcter geogràfic representativa d'una circumstància o situació.

En el nostre cas concret, el programari contarà de l'aplicació nativa iOS d'enviament de la ubicació que haurem de desenvolupar, així com el visor de mapa web que ens facilitarà la UOC per tal de veure representat els resultats.

En quant al maquinari del SIG, considerarem el dispositiu mòbil iPhone on s'executarà l'aplicació client i també el servidor web proporcionat per la UOC per la recepció de les dades.

Pel que fa a les dades, considerarem les coordenades enviades que ens serviran per localitzar en tot moment a cada equip participant, veure el ritme al que avancen i comprovar en quina posició del rànquing, es troben cada un d'ells. També es consideren dades del sistema SIG els diferents atributs no espaials que complementen les coordenades entre les que estan el nom de l'equip al que refereixen i el nivell de bateria del dispositiu.

1.3. L'entorn tecnològic iOS

iOS és el sistema operatiu per a aparells mòbils que utilitza en exclusiva la empresa Apple i que fa funcionar la seva família de dispositius iPod, iPhone i iPad.

Segons informació de finals de 2012¹, Apple tenia el 53% de la quota de mercat dels smartphone als EEUU, sent el terminal més venut en aquell país. El sistema operatiu iOS era el segon més extès amb un 35% del total, per darrere d'Android que engloba multitud de fabricants. A mes a mes, disposa d'una tenda virtual amb més de 500.000 aplicacions de gran qualitat.

Per tot això, s'ha de considerar iOS com una de les plataformes mòbils més importants i exteses del mon.

¹ Informació extreta de: http://www.comscore.com/Insights/Press_Releases/2013/1/comScore_Reports_November_2012_U.S._Mobile_Subscriber_Market_Share

2. OBJECTIUS

Els objectius que s'han plantejat com a resultat de la realització d'aquest treball de final de carrera, són els següents:

2.1. Objectius Generals

1. Plantejar i raonar els problemes de mobilitat, posicionament i autonomia dels dispositius mòbils i les seves capacitats de geoposicionament.
2. Conèixer les característiques i conceptes de la tecnologia dels sistemes de geoposicionament i d'informació geogràfica.
3. Familiaritzar-se amb els objectes específics de la programació per a dispositius mòbils iOS.
4. Saber gestionar dades geogràfiques, com pot ser la gestió de mapes o publicació de serveis de mapes.

2.2. Objectius Específics

1. Analitzar la viabilitat de la solució sota la limitació d'autonomia.
2. Justificar la solució proposada sota les condicions descrites.
3. Aprendre a utilitzar el posicionament GPS d'un mòbil Apple i saber localitzar l'usuari.

3. FITES I TEMPORITZACIÓ

Per dur a terme el treball de la forma més adequada possible, es proposa la distribució de la feina en una sèrie de tasques genèriques, que posteriorment seran detallades i ubicades dins d'una planificació temporal que permeti assolir els compromisos adquirits dins de l'àmbit del projecte.

Tasca 1: Anàlisis de les solucions prèvies

Degut a que algun dels requeriments de la solució proposada, parteixen dels projectes desenvolupats en el semestre anterior, és considerada primordial realitzar un acurat anàlisis tant de la memòria, com de les presentació i fins i tot, del codi presentat en els dos treballs precedents.

D'aquest anàlisis es pretén obtenir, entre d'altres, la informació concreta relativa a la connexió amb els serveis web ArcGIS.

Tasca 2: Familiarització de la tecnologia a utilitzar

Com a resposta a la inexperiència en el desenvolupament de sistemes d'informació geogràfica sota la plataforma iOS, s'ha previst reservar un temps d'aprenentatge en les diferents tecnologies involucrades en l'arquitectura escollida per a la solució d'aquest treball.

Tasca 3: Disseny i implementació de l'aplicació client

Aquesta tasca inclou el disseny, la implementació i les proves unitàries de l'aplicació client mòbil nativa iOS encarregada de monitoritzar els participants i enviament de la localització utilitzant un servei Web.

El disseny contindrà l'anàlisi dels casos d'ús, col·laboració i jerarquia, prototipatge de l'aplicació i si s'escau, el detall de la persistència.

Tasca 4: Banc de proves

S'ha considerat que les proves necessàries per garantir la qualitat de la solució desenvolupada, tenen la suficient importància com per formar part d'una tasca pròpia. Aquesta tasca, inclou la realització de les proves en camp que simulin les condicions reals en les que s'utilitzarà el programari, una bateria de proves d'estrès per comprovar la robustesa i les correccions en el codi que es derivin dels resultats obtinguts.

Tasca 5: Finalització de la memòria i presentació

Es finalitzarà el document de la memòria del projecte amb els apartats que no s'hagin realitzat a les PACs anteriors. També es crearà un vídeo de presentació del treball realitzat, que servirà al tribunal de l'assignatura per avaluar tota la feina feta.

A continuació és defineix el detall de les tasques enumerades anteriorment:

#	FITA	SETMANA	TASCA
Inici projecte			
1		9 - 25/2	Lectura dels diferents anuncisats
2		9 - 25/2	Selecció del projecte a realitzar
	F0	9/3/2013	Trobada virtual
3		10 - 4/3	Redacció PAC1
4		11 - 11/3	Entrega esborrany PAC1
5		11 - 11/3	Aplicació correccions PAC1
	F1	12/3/2013	Lliurament PAC1
Tasca 1: Anàlisi de les solucions prèvies			
6		11 - 11/3	Descàrrega memòria i presentació solucions anteriors
7		11 - 11/3	Lectura acurada documentació solucions anteriors
8		11 - 11/3	Redacció apartat "anàlisi solucions anteriors"
Tasca 2: Familiarització de la tecnologia a utilitzar			
9		12 - 18/3	Lectura temari SIG
10		12 - 18/3	Instal·lació entorn desenvolupament iOS/SIG
11		12 - 18/3	Comprovació funcionament del programari servidor
12		12 - 18/3	Formació desenvolupament XCode (iOS SDK)
13		13 - 25/3	Formació desenvolupament ArcGIS iOS SDK
Tasca 3: Disseny i implementació de l'aplicació client			
14		13 - 25/3	Diagrama d'arquitectura
15		13 - 25/3	Definició textual de casos d'ús
16		13 - 25/3	Diagrama col·laboració i jerarquia
17		14 - 1/4	En cas de ser necessari, disseny de persistència
18		14 - 1/4	Disseny de prototipus d'interfase d'entrada/sortida
19		14 - 1/4	Redacció apartat "descripció de la solució"
20		14 - 1/4	Composició esborrany PAC2
21		14 - 1/4	Entrega esborrany PAC2
22		15 - 8/4	Aplicació correccions PAC2

MONITORITZACIÓ TRAILWALKER

	F2	9/4/2013	Lliurament PAC2
23		15 - 8/4	Implementació de l'aplicació client
24		19 - 06/5	Realització proves unitàries
25		19 - 06/5	Redacció apartat "Banc de proves: Unitàries"
26		19 - 06/5	Redacció apartat "Programació en iOS"
27		19 - 06/5	Redacció apartat "Patrons de disseny"
28		19 - 06/5	Composició esborrany PAC3
29		20 - 13/5	Entrega esborrany PAC3
30		20 - 13/5	Aplicació correccions PAC3
	F3	14/5/2013	Lliurament PAC3
Tasca 4: Banc de proves			
31		20 - 13/5	1ª etapa proves de camp per l'aplicació iOS
32		20 - 13/5	Conclusions parcials del resultats
33		21 - 20/5	Correcció dels errors trobats en la 1ª etapa
34		21 - 20/5	2ª etapa proves de camp per l'aplicació iOS
35		21 - 20/5	Última fase de proves d'estrès per l'aplicació iOS
36		21 - 20/5	Redacció resta apartat "Banc de proves"
Tasca 5: Finalització de la memòria i presentació			
37		22 - 27/5	Redacció apartat "Conclusions"
38		22 - 27/5	Redacció apartat "Agraïments"
39		22 - 27/5	Redacció apartat "Millores futures"
40		22 - 27/5	Redacció apartat "Bibliografia"
41		22 - 27/5	Redacció apartat "Glossari"
42		22 - 27/5	Composició esborrany complert de la memòria
43		22 - 27/5	Entrega esborrany memòria
44		23 - 3/6	Aplicació correccions memòria
45		23 - 3/6	Gravació de la presentació
	F4	10/6/2013	Entrega memòria i presentació
46		26 - 24/6	Debat virtual

Algunes de les tasques enumerades, com la Bibliografia i el Glossari, estan ubicades en un moment concret de la planificació, però tenen un caràcter transversal i per tant, s'hauran de realitzar al llarg del transcurs del projecte.

4. DIAGRAMA DE GANTT

En aquest capítol, es presenta el diagrama de GANTT de la planificació temporal de totes les tasques definides a l'apartat anterior.

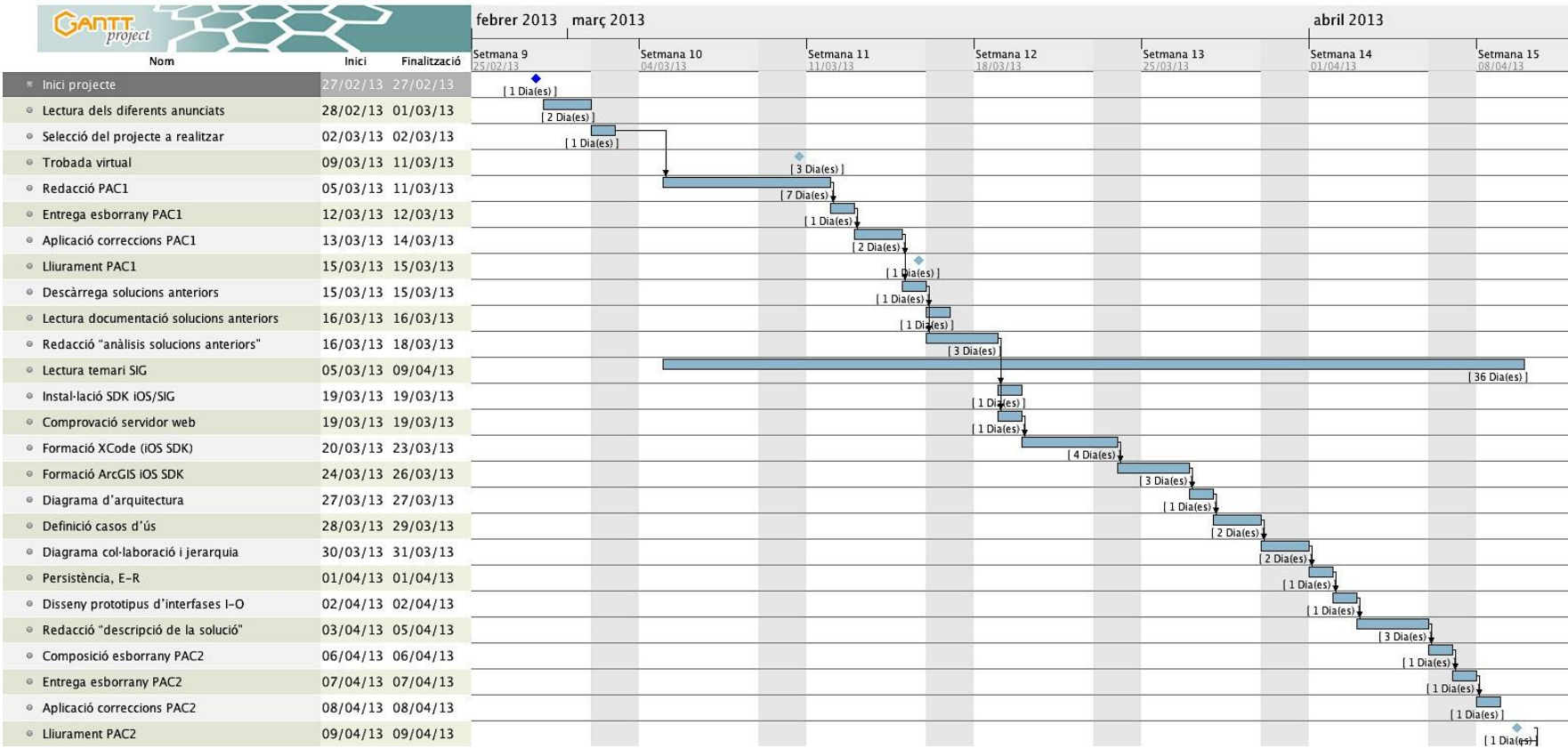


Figura 1: GANTT del pla de treball des del començament fins el lliurament de la PAC2

MONITORITZACIÓ TRAILWALKER

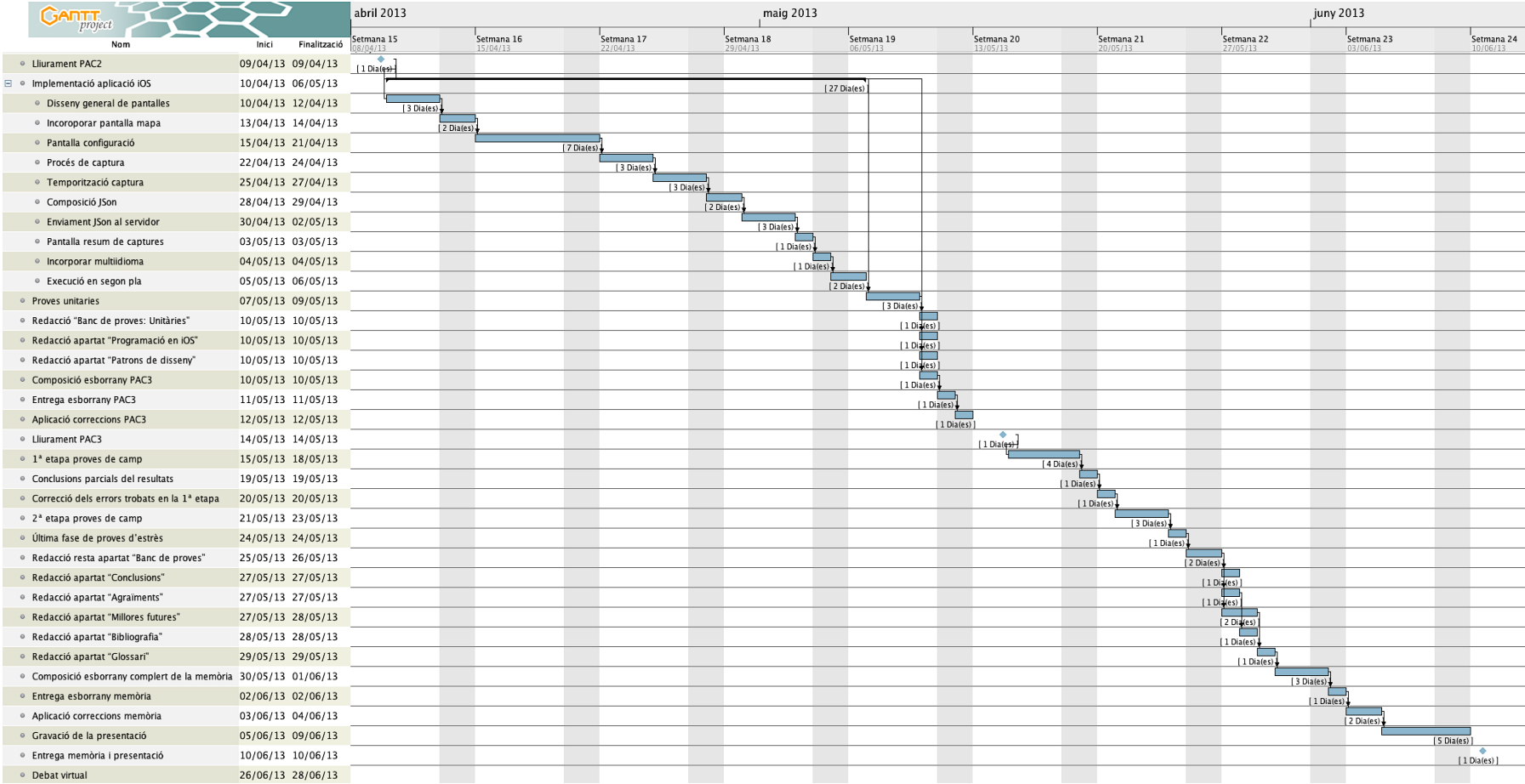


Figura 2: GANTT del pla de treball des del lliurament de la PAC2 fins la finalització

5. INCIDÈNCIES I RISCOS

A continuació, es realitza un llistat dels diferents successos que poden esdevenir durant el transcurs del projecte i que afectarien negativament a la planificació definida anteriorment.

5.1. Problemes amb l'equip de treball

Està previst realitzar el TFC amb un ordinador iMac 21" de sobretaula que aconsegueix amb les necessitats de maquinari requerides pels diferents programaris que s'utilitzaran en el projecte.

Com a mesura de prevenció, es realitzaran còpies de seguretat privades setmanalment a Google Drive, tant de la documentació generada, com del codi font.

En el cas que succeís algun inconvenient que no permetés continuar utilitzant l'equip descrit, passaria a utilitzar-se un portàtil marca Toshiba amb el sistema operatiu Windows 7.

Degut a que el desenvolupament de l'aplicació mòbil en iOS requereix, un entorn de desenvolupament en un S.O. Mac OS X, s'utilitzaria una màquina virtual com VMware o similar, per simular l'entorn necessari.

5.2. Problemes amb la connexió de xarxa

En cas de talls prolongats en el servei d'ADSL de l'entorn de treball habitual, es continuaria utilitzant el mateix equip sense connexió i es passaria a realitzar les còpies de seguretat de la feina generada en un llaç de memòria per transmetre tota la informació necessària, mitjançant la xarxa de l'entorn laboral.

Com a últim recurs, en cas d'emergència i sempre i quan no sigui necessari enviar gran quantitat d'informació, es podria utilitzar la connexió 3G contractada del telèfon mòbil.

5.3. Desplaçament per motius laborals

Es considera que aquesta situació és molt poc probable tenint en compte els antecedents. De totes formes, si es produís aquest fet, s'actuarià de la mateixa manera que s'ha definit en cas de tenir problemes amb l'equip de treball.

5.4. Baixa per malaltia

En cas d'una baixa per malaltia de curta durada (menys d'una setmana), no es consideraria crític de cara al seguiment de la planificació prevista i la dedicació hauria de poder ser compensada una vegada recuperat del problema.

En cas de baixa prolongada (més d'una setmana) i en funció de la gravetat de la malaltia, es plantejaria al consultor del projecte una rebaixa acordada sobre els requeriments assumits inicialment segons el pla de treball.

6. MATERIAL REQUERIT

En aquest apartat es detallen els materials necessaris per a dur a terme el projecte.

6.1. Programari documentació

El programari necessari per a la generació de la documentació del projecte (PAC's, memòria, etc...) i per a la presentació final en vídeo, és el següent:

- *Microsoft Word 2011* → Redacció de texts.
- *GanttProject v2.5* → Disseny de diagrames Gantt.
- *Balsamiq Mockups v2.2* → Disseny de prototips.
- *MagicDraw v17* → Disseny UML i casos d'usos.
- *Microsoft PowerPoint 2011* → Gravació i muntatge de la presentació en vídeo.

6.2. Programari desenvolupament

El programari requerit per al desenvolupament del producte final tecnològic, és el següent:

- *ArcGIS Server* → Programari que proporciona serveis GIS per mapes i demés serveis GIS.
- *iOS SDK Xcode v4.6* → Kit de desenvolupament per a la plataforma iOS.
- *ArcGIS Runtime SDK for iOS* → Conjunt de llibreries ArcGIS per a la plataforma iOS.

6.3. Maquinari

El maquinari requerit serà el següent:

- *Equip amb MAC OS X* → Desenvolupament del programari.
- *Dispositiu IOS iPhone* → Proves del programari creat.
- *Servidor ArcGIS* → Proporcionat per la UOC.

7. ANÀLISIS SOLUCIONS PREVIES

En el plec de requeriments, es fa referència a dos treballs previs que donaven una solució vàlida per a les funcionalitats proposades. Per aquest motiu, s'ha considerat necessari realitzar un estudi d'aquests dos projectes, per tal d'establir un bon punt de partida.

7.1. Estudi i conclusions TFC Albert Jordà

El primer dels treballs, és el que va publicar l'Albert Jordà Vinardell el 7 de Gener de 2013 i que pot ser consultat íntegrament en aquesta direcció web: <http://hdl.handle.net/10609/19762>.

La memòria d'aquest projecte es divideix en dos grans blocs.

Per un costat, es realitza un anàlisi en profunditat de les diferents opcions tecnològiques que hi ha en el mercat i que permetrien desenvolupar el sistema requerit. Es considera primordial que la solució escollida compleixi els principis d'autonomia, mobilitat, fiabilitat, economicitat i robustesa.

D'entre els diferents sistemes analitzats, se n'acaba considerant viables dos de diferents:

- Un sistema basat en la plataforma de maquinari Arduino.
Aquesta solució té com a principal punt fort, una gran autonomia d'ús en les circumstàncies previstes per a la cursa Trailwalker, a més d'una gran facilitat de reposició de les bateries en cas d'esgotar la seva capacitat (és tracta de les habituals piles de tipus AAA de 1,5V).
En la seva contra, juga el fet de que se suposa que els equips participants no disposaran de la placa Arduino i la seva adquisició està valorada entorn els 150€.
- L'altre opció analitzada, és el desenvolupament de programari a mida per a un dispositiu smartphone amb GPS integrat. Les forteses i febleses d'aquest sistema són precisament les contraries de l'Arduino.
En aquest cas si que és suposa que els participants disposen d'un telèfon smartphone (o com a mínim un participant per cada equip) i per aquest motiu, no es planteja incórrer en cap cost econòmic addicional, ni per part del maquinari, ni per part del programari que suposarà la instal·lació d'una app gratuïta. El principal motiu en contra, és l'autonomia.

MONITORITZACIÓ TRAILWALKER

La majoria d'smartphones actualment en el mercat, incorporen bateries amb capacitats que van dels 1200 als 1800 mAh, que suposa una autonomia d'unes 8 hores en conversa telefònica i sobre unes 6 de navegació per Internet.

Finalment, s'acaba determinant que la solució basada en l'smartphone és la més adequada per complir amb les necessitats previstes per a la Intermón Oxfam Trailwalker.

La segona part de la memòria es focalitza en descriure la solució escollida en el punt anterior. L'arquitectura bàsica de la solució es basa en 3 parts diferenciades:

- Un mòbil smartphone amb el sistema Android i una aplicació feta a mida que envia les posicions.
- Un servidor ArcGIS que s'encarrega de rebre i persistir la informació.
- Un servidor web que disposa la capa de presentació, per tal de mostrar les posicions dels equips a través d'un navegador web.

Primerament, es defineix el sistema Android de forma general i s'especifica la forma en que es componen les aplicacions construïdes sota aquest entorn. Addicionalment, es donen explicacions de com es realitza la captura de la posició i les conseqüències en quant al consum de la bateria.

També es defineix la tecnologia implicada amb el servidor ArcGIS. S'explica que el format d'intercanvi de dades és el format JSON i que els serveis REST són els responsables d'exposar la funcionalitat GIS, per a que la utilitzi la capa de presentació.

Finalment, es realitza el disseny detallat de l'aplicació Android:

- Es defineixen els diferents casos d'ús que la formen.
- Es detalla la estructura de classes utilitzada, la interrelació entre elles i la jerarquia d'herències.

Un dels apartats més transcendents, és el que es refereix a la estratègia escollida en relació a la captura de la posició.

Es comenta que inicialment es va realitzar un primer plantejament que tractava de tenir el control de l'execució sobre la obtenció de la posició, adormint el procés

entre captura i captura, però va ser descartada degut als problemes de sincronia entre fils, a compatibilitat en la versió SDK utilitzada i a certs problemes.

Es va buscar una alternativa, en la que cedia el control de la cadència temporal i de la tolerància en la distància al propi sistema, mitjançant les opcions de les classes utilitzades.

El plantejament final escollit, incorpora una sèrie de millores a la estratègia anterior al permetre utilitzar la escolta passiva de posicionament i sobretot a l'incorporar el Servei d'ubicació de Google com a mètode de localització.

Una vegada finalitzada la implementació del producte, s'executa un banc de proves per tal d'obtenir resultats reals de rendiment, robustesa i precisió del programa. En general, es determina que l'aplicació es comporta correctament, tal com s'esperava. Una de les conclusions que es desprèn de les proves realitzades, és que no s'aprecia un diferència significativa entre el consum de bateria utilitzant el GPS o el servei d'ubicació de Google. També s'aconsegueix obtenir una estimació de la durada total de la bateria de l'smartphone, que s'aproximaria a les 11 hores de vida, aspecte que fa bastant viable la solució adoptada dins de les condicions descrites per la prova.

El treball realitzat per l'Albert Jordà planteja el desenvolupament tecnològic, com a una forma de demostrar la viabilitat de la solució escollida a l'apartat inicial de l'estudi.

En relació al disseny de l'aplicació Android transmissora, es defineixen les funcionalitats bàsiques que s'hauran de desenvolupar per a l'aplicació iPhone necessària en aquest projecte:

- Iniciar / Aturar captura de posicions
- Captura de posicions
- Captura d'informació de dispositiu
- Enviament de dades al servidor ArcGIS
- Canvi de nom de l'equip
- Tancament de l'aplicació
- Canvi de configuració de l'aplicació
- Multi Idioma
- Funcionament en segon pla

En quant als problemes i dificultats experimentats en el transcurs del seu treball, de ben segur, anticipen les mateixes dificultats que sorgiran al llarg del projecte actual, pel fet de compartir arquitectures tan semblants.

7.2. Estudi i conclusions TFC Montse Baldomà

La segona solució a analitzar és la de Montse Baldomà Cases publicada el dia 7 de Gener de 2013 i que podem trobar a <http://hdl.handle.net/10609/19261>.

En aquest segon projecte, s'ha optat per una solució basada en el maquinari específic Arduino. El sistema consisteix els següents elements:

- Placa Arduino Uno rev3
- Placa 3G/GPRS amb GPS intern
- Antena GPS interna
- Antena GPRS-GSM-UMTS externa

La placa Arduino conté un microprocessador ATmega328 que permet utilitzar les funcionalitats de localització y navegació web de la placa 3G/GPRS/GPS, una vegada estan connectades mitjançant un parell de pins digitals.

L'Arduino es pot programar amb un software propi anomenat Sketch, que és un llenguatge amb una sintaxi similar a la del C derivat del Wiring, però que aporta les comandes necessàries per a gestionar el maquinari descrit. L'executable generat s'emmagatzema dins de la memòria flash de 32 Kb que pot incorporar la placa Arduino.

La principal virtut del sistema escollit, és la de proporcionar una autonomia suficient com per a donar servei a la cursa plantejada per Intermón Oxfam. És realitza un ampli estudi sobre el consum teòric i sobre la tensió necessària a subministrar a l'equip i es conclou que utilitzant 6 piles alcalines AA de 2900 mA · h en sèrie i transmetent la informació cada 10 minuts, es podria arribar a mantenir en funcionament el sistema més de 27 hores seguides. Espaiant la freqüència de la connexió o ampliant el nombre de bateries utilitzades a 8 en total, es preveu acomplir sobradament l'objectiu.

MONITORITZACIÓ TRAILWALKER

De la mateixa manera que en l'anterior projecte, s'inclou un apartat on es defineix la tecnologia implicada amb el servidor ArcGIS a més d'identificar el directori de serveis utilitzat.

Finalment, s'introdueix el llenguatge JavaScript que serà necessari per la realització del visor web que explotará la informació recollida.

Una vegada analitzat aquest treball, es conclou que la solució tecnològica adoptada per la Montse Baldomà s'allunya considerablement de l'arquitectura marcada en el projecte actual. Per aquest motiu, es descarta traslladar cap aspecte relatiu al disseny de la interfície de l'aplicació client. En contrapartida, localitzem una descripció completa de les capes exposades en els serveis REST, que haurem d'utilitzar dins per a la nostra solució:

- El primer dels serveis, és l'únic que no està ubicat en el servidor ArcGIS de la UOC, sinó en un servidor ArcGIS propi de la companyia ESRI. La capa que proporciona és de tipus raster i servirà com a mapa de base de fons: http://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer
- Un altre servei és el que aporta informació sobre les rutes de la Trailwalker: <http://napols.uoc.es/arcgis/rest/services/rutes/MapServer>
Aquest servei conté dues capes diferents. La identificada com a 0, proporciona informació dels punts de control de la cursa. La identificada com a 1, proporciona informació de les diferents etapes.
- La següent capa vectorial, retorna la totalitat dels punts de seguiment capturats i enviats pels diferents equips participants a la cursa: <http://napols.uoc.es/arcgis/rest/services/TrailWalker/MapServer>
- Aquesta és la capa de tipus FeatureServer que s'utilitzarà, mitjançant la funció Add Feature, per afegir les captures realitzades per cada equip: <http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer>
- L'última capa disponible és aquella que retorna un únic punt amb la última posició rebuda pels diferents equips participants: <http://napols.uoc.es/arcgis/rest/services/UltimaEntrada/MapServer>

MONITORITZACIÓ TRAILWALKER

Finalment es defineix l'estructura elemental dels missatges JSON que s'utilitzarà en la comunicació amb els diferents serveis REST.

En el cas del servei responsable de l'addició dels punts capturats pel dispositiu iOS, el format serà el següent:

```
[{"geometry":{"x":xxxxxx.xxxxxxxxxx,"y":yyyyyyyy.yyyyyyyyyy,"spatialReference":{"wkid":102100}},{"attributes":{"CodiEquip":codiEquip,"Equip":equip,"Bateria":nivellBateria,"Entrada":nnnnnnnnnnnnnnnn,"Tipus":tipusSistema}}]
```

On podem veure que les diferents dades transmises, estan dividides en parelles indicades pel nom de l'atribut i el seu valor: *"nom":valor*.

Veiem que identifica la geometria amb:

- X → Correspon a la longitud de la captura realitzada pel GPS del dispositiu.
- Y → Correspon a la latitud de la captura realitzada pel GPS del dispositiu.
- SpatialReference → Amb el valor *"wkid":102100*, que correspon² al datum WGS84 utilitzat habitualment en els sistemes GPS.

Els atributs alfanumèrics de la geometria, són els següents:

- CodiEquip → Codi IMSI del dispositiu des d'on s'ha realitzat la captura.
- Equip → Nom que identifiqui l'equip participant.
- Bateria → % del nivell de la bateria en el moment de la captura.
- Entrada → Data i hora de la captura.
- Tipus → En el nostre cas serà el literal iOS.

² Llistat complet de la codificació wkid per ESRI: http://resources.esri.com/help/9.3/arcgisserver/apis/javascript/arcgis/help/jsapi_start.htm#jsapi/spatialreference.htm

8. DESCRIPCIÓ DE LA SOLUCIÓ

8.1. Arquitectura

L'arquitectura del sistema queda representada en el següent diagrama:

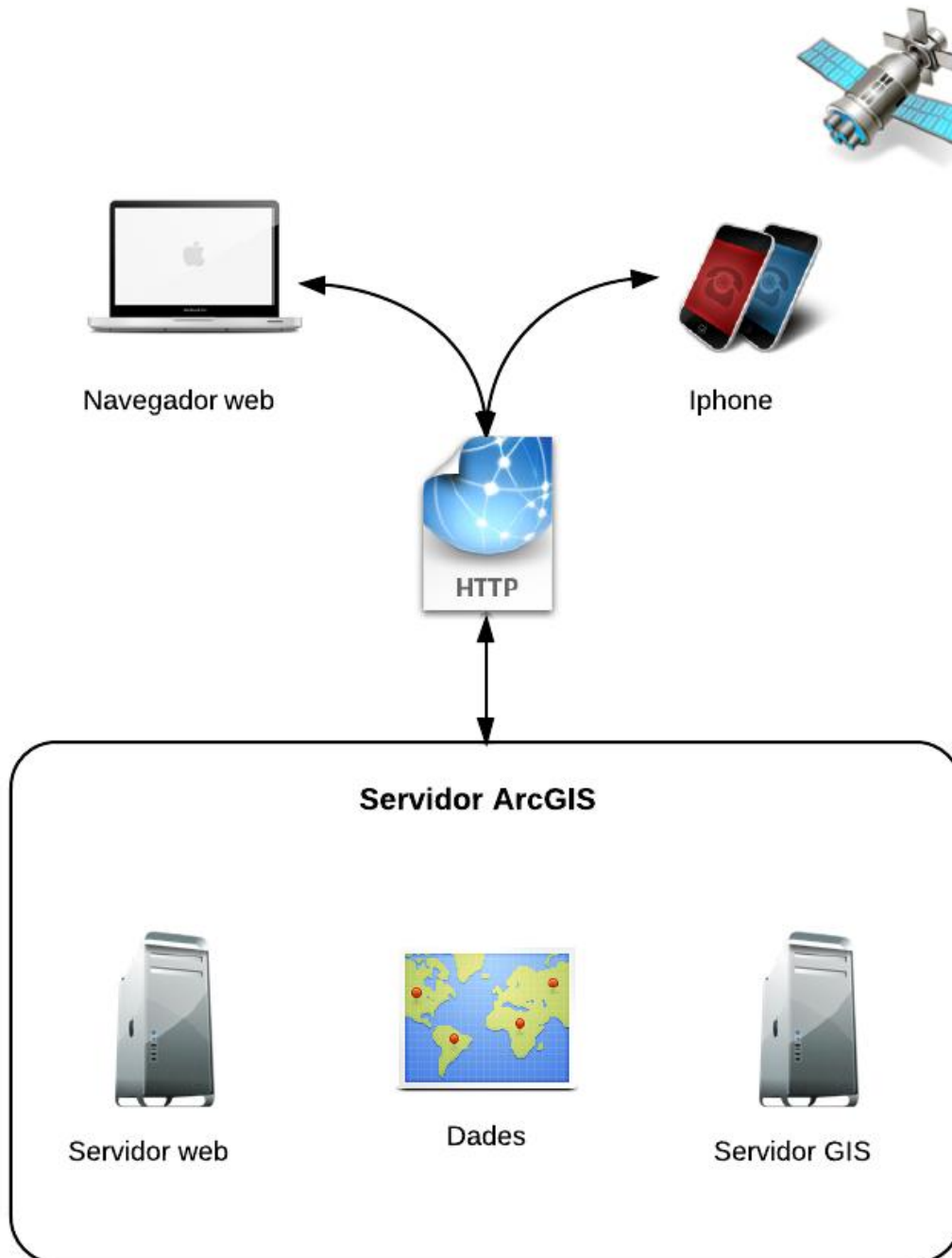


Figura 3: Diagrama d'arquitectura Trailwalker

En el bloc central tenim el servidor ArcGIS proporcionat per la UOC i que està format per:

- L'ArcGIS for Server que disposa els serveis REST per la actualització i el consum de les capes necessàries.
- Les pròpies dades de caràcter espacial que s'utilitzaran des dels diferents clients i englobarà tant les capes vectorials amb el detall de les posicions, com les capes raster amb els mapes de base.
- Un servidor web que donarà servei a les consultes relatives a la posició dels equips.

Per altra banda tenim els diferents clients:

- En el cas de la consulta de la informació en temps real, serà un dispositiu amb connexió a Internet que realitzarà les peticions al servidor web mitjançant crides HTTP.
- En el cas de l'aplicació de transmissió de la posició, es tractarà dels diferents dispositius iPhone dels equips participants.

8.1.1. Sistema iOS

El sistema iOS està basat en una tecnologia de capes ordenades jeràrquicament. Les capes inferiors són les que realitzen el control dels serveis fonamentals del sistema, mentre que les capes superiors són les que tracten les característiques més sofisticades.

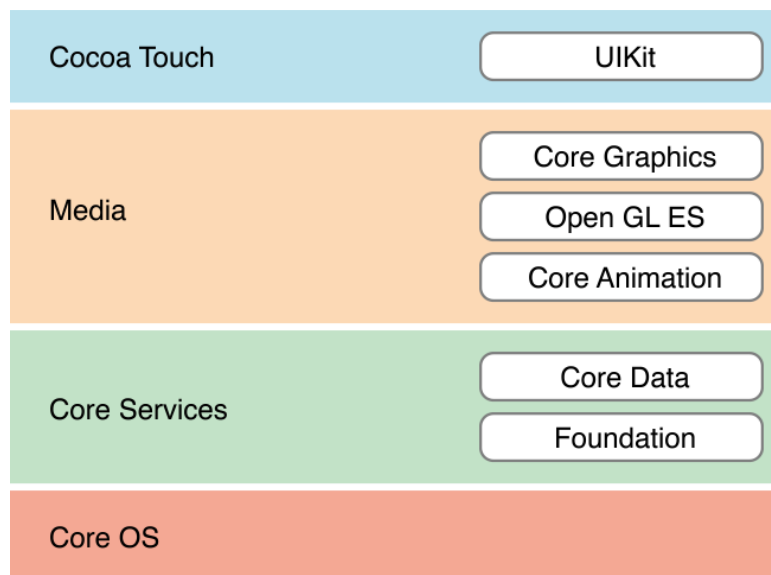


Figura 4: Esquema de capes d'iOS

<http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/RoadMapiOS/chapters/SurveytheMajorFrameworks/SurveytheMajorFrameworks/SurveytheMajorFrameworks.html>

En el moment de crear una aplicació iOS, l'accés a aquestes capes es realitza mitjançant la utilització de frameworks que, en aquest context, refereix a les llibreries de mètodes proporcionades principalment per la pròpia companyia Apple.

Els diferents estàndards de programació d'iOS recomanen utilitzar les capes superiors del sistema, gracies a les quals s'aconsegueix simplificar el codi a generar.

8.1.1. Sistema d'ubicació d'iOS

iOS ens proporciona dos sistemes diferents de localització.

Per un costat està el servei estàndard de localització (*Standard Location Service*). És el sistema més habitual per obtenir la ubicació actual del dispositiu en el que es combinen el GPS integrat, la detecció de punts d'accés WIFI i d'antenes de telefonia, per triangular una ubicació amb la major precisió possible. Aquest mètode permet definir els paràmetres de precisió mínima desitjada i filtre de distància en metres a la hora de considerar una nova localització a demanda.

Des de la versió 4.0 d'iOS també tenim disponible el servei de canvi significatiu d'ubicació (*Significant-Change Location Service*). Les diferències amb l'altre servei són notables; és menys agressiu en el consum energètic ja que sols utilitza la captura de senyal d'antena per realitzar la triangulació, per tant, ofereix una menor precisió però suficient per facilitar el geolocalització necessària. La parametrització del servei, ve predefinida a 500 metres de filtre de distància amb actualitzacions mínimes de 5 minuts entre captura (mai inferior). Aquest mètode està especialment dissenyat per ser utilitzat en tasques de llarga durada en segon pla.

8.2. Casos d'ús

8.2.1. Procés de captura de la posició

A continuació es defineixen, textualment, els diferents casos d'ús relacionats amb el procés principal de captura i enviament de la posició:

Cas d'ús	Activar procés de captura de posició
Descripció	Inicialitza el procés de captura de dades, establint la cadena de generació i els paràmetres relatius a la precisió de la localització.
Casos d'ús	Captura de posició.

MONITORITZACIÓ TRAILWALKER

relacionats	Enviament de dades al servidor.
Pre condició	Aplicació engegada.
Post condició	És finalitza el procés de captura de posició o es tanca l'aplicació.
Disparador	L'usuari prem el botó corresponent.
Flux bàsic	L'aplicació inicia el procés de localització i enviament de dades al servidor ArcGIS.
Flux alternatiu	El servei de localització GPS no està habilitat, mostra avís a l'usuari per tal d'activar-ho.

Cas d'ús	Finalitzar captura de posició
Descripció	Permet desactivar la captura de posicions. Aquesta acció no esborra de l'aplicació les captures realitzades.
Casos d'ús relacionats	Activar procés de captura de posició.
Pre condició	S'ha activat prèviament el procés de captura.
Post condició	
Disparador	L'usuari prem el botó corresponent.
Flux bàsic	L'aplicació deixa de capturar posicions i dades.
Flux alternatiu	

Cas d'ús	Captura de posició
Descripció	L'aplicació obté el nom de l'equip, la posició, el nivell de la bateria, la data i hora del sistema i la identificació del dispositiu.
Casos d'ús relacionats	Activar procés de captura de posició. Enviament de dades al servidor. Modificar els paràmetres de l'aplicació. Llistar l'estat de les posicions capturades.
Pre condició	S'ha activat prèviament el procés de captura.
Post condició	Enviament de la captura al servidor ArcGIS.
Disparador	Procés periòdic disparat automàticament.
Flux bàsic	La informació es captura correctament i queda pendent de l'enviar al servidor.
Flux alternatiu	

MONITORITZACIÓ TRAILWALKER

Cas d'ús	Enviament de dades al servidor
Descripció	L'aplicació construeix una estructura JSON amb la informació capturada i és enviada al servidor ArcGIS pel seu registre i tractament.
Casos d'ús relacionats	Activar procés de captura de posició. Captura de posició. Llistar l'estat de les posicions capturades.
Pre condició	S'ha activat prèviament el procés de captura. Existeix una captura pendent d'enviar al servidor.
Post condició	
Disparador	L'aplicació disposa d'una nova captura d'informació o d'una captura anterior que no havia estat possible enviar.
Flux bàsic	Enviament d'informació al servidor ArcGIS realitzat correctament.
Flux alternatiu	No s'aconsegueix realitzar l'enviament d'informació per no tenir connexió HTTP. Marca la captura com a no enviada i la deixa pendent de processar de nou posteriorment.

8.2.2. Consulta d'informació

Aquestes serien les consultes d'informació que podria realitzar l'usuari mitjançant l'aplicació iOS.

Cas d'ús	Llistar l'estat de les posicions capturades
Descripció	Es mostra un llistat amb totes les captures realitzades durant una sessió.
Casos d'ús relacionats	Activar procés de captura de posició. Captura de posició. Enviament de dades al servidor.
Pre condició	S'ha activat prèviament el procés de captura. S'ha generat com a mínim una captura d'informació.
Post condició	
Disparador	L'usuari es desplaça fins la pantalla on es mostra l'estat de les captures.
Flux bàsic	L'usuari aconsegueix consultar les diferents captures realitzades per corroborar l'estat de les mateixes.
Flux alternatiu	

Cas d'ús	Consultar la posició actual en el mapa
-----------------	-----------------------------------------------

Descripció	La pantalla principal de l'aplicació mostra un mapa de la regió, centra la vista en la posició actual del dispositiu i utilitza un marcador per identificar la ubicació actual.
Casos d'ús relacionats	
Pre condició	El servei GPS ha d'estar activat.
Post condició	
Disparador	L'usuari es desplaça fins la pantalla on es mostra el mapa.
Flux bàsic	Es mostra el mapa i la posició del dispositiu.
Flux alternatiu	Si el GPS no està activat, el marcador d'ubicació actual no apareixerà en el mapa.

8.2.3. Configuració de l'aplicació

Aquests són els casos d'ús que permeten a l'usuari canviar el comportament de l'aplicació:

Cas d'ús	Canviar el nom de l'equip
Descripció	Es permet a l'usuari la modificació del nom de l'equip al que pertanyen les captures realitzades pel dispositiu.
Casos d'ús relacionats	Captura de posició.
Pre condició	Aplicació engegada.
Post condició	Continuar capturant posicions.
Disparador	L'usuari introdueix amb el teclat virtual el nom de l'equip dins d'un camp de text.
Flux bàsic	L'usuari confirma la modificació del nom de l'equip.
Flux alternatiu	L'usuari no confirma la modificació del nom de l'equip.

Cas d'ús	Modificar els paràmetres de l'aplicació
Descripció	Es permet a l'usuari la modificació d'alguns paràmetres propis de l'aplicació, com poden ser l'idioma en el que es mostren els textos, el temps de cadència entre captures o la precisió de la localització.
Casos d'ús relacionats	Captura de posició.
Pre condició	Aplicació engegada.

Post condició	Continuar capturant posicions.
Disparador	L'usuari accedeix a una pantalla que permet, mitjançant diversos components d'entrada, la modificació del paràmetres.
Flux bàsic	L'usuari confirma la modificació dels paràmetres de l'aplicació.
Flux alternatiu	L'usuari no confirma la modificació dels paràmetres de l'aplicació.

A continuació es mostra el diagrama, on es relacionen els diferents casos d'ús que formen l'aplicació.

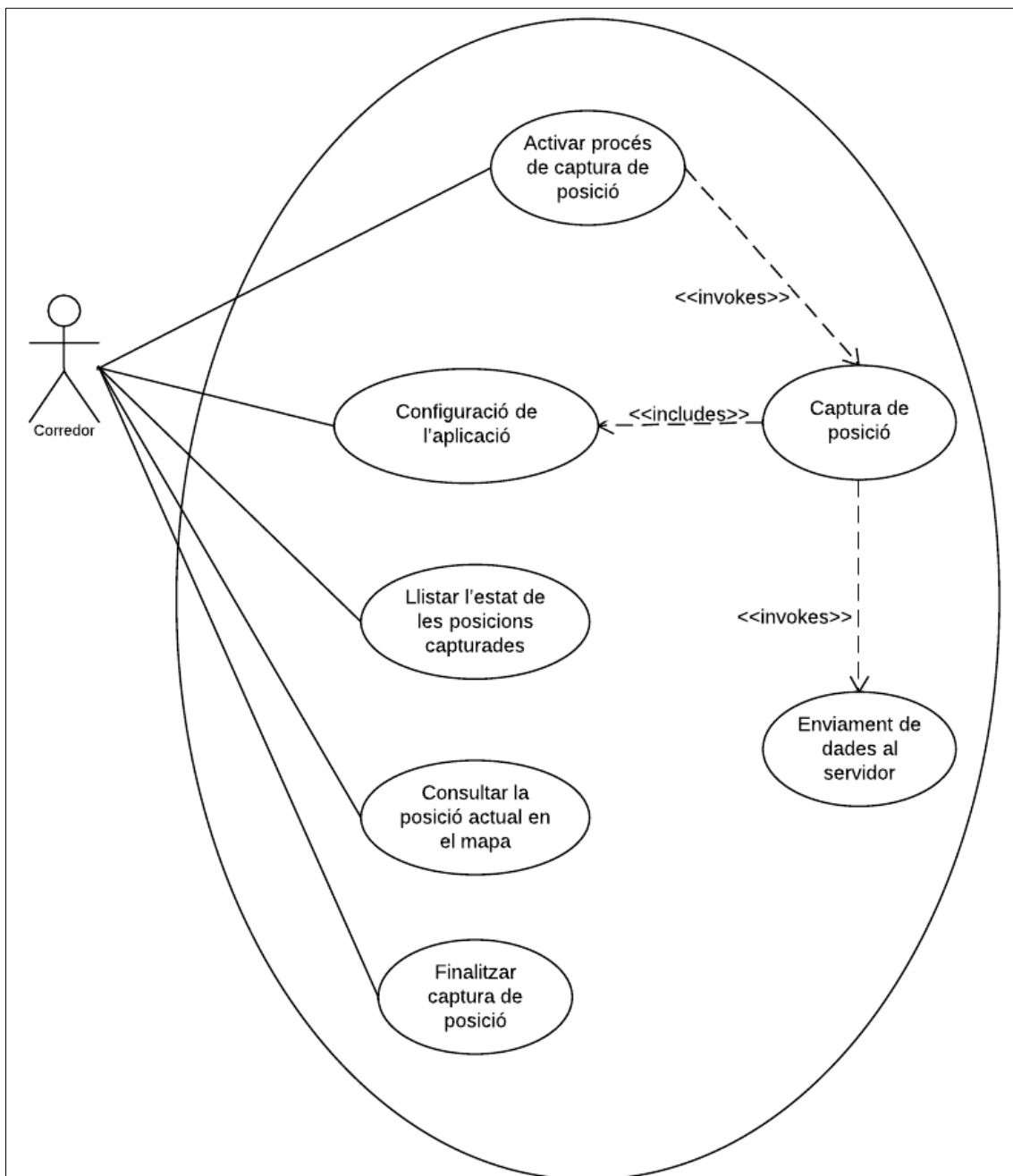


Figura 5: Diagrama casos d'ús de l'aplicació iOS

8.3. Diagrames col·laboració i jerarquia

Dins de les referències d'Apple, trobem diversos apartats que destaquen el patró de disseny Model-Vista-Controlador (MVC) com a un model vàlid per a la construcció d'aplicacions iOS. Aquest patró determina que els diferents objectes que es poden construir, estan englobats en una de les tres categories que li donen el nom.

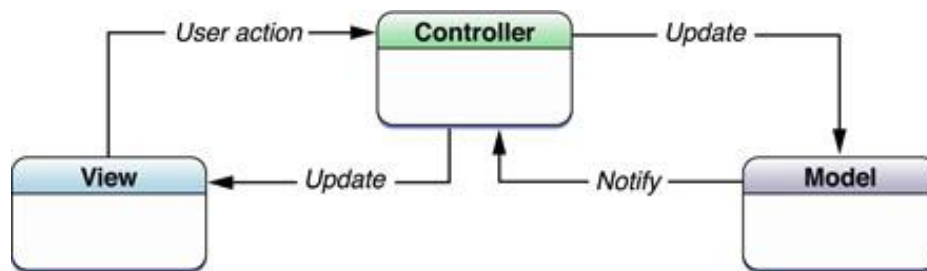


Figura 6: Diagrama patró disseny MVC

<https://developer.apple.com/library/mac/#documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>

De forma molt esquemàtica, podríem definir:

- Els objectes de tipus Model són aquells que encapsulen les dades utilitzades dins l'aplicació.
- Els de tipus Vista, són els objectes encarregats d'interactuar amb l'usuari, entre els que s'inclouen els que s'utilitzen per mostrar o modificar el valor d'un objecte de tipus Model.
- Els de tipus Controlador, són els que actuen com a intermediari dels dos rols anteriors.

D'aquesta manera, podem dividir les classes del sistema segons el model MVC:

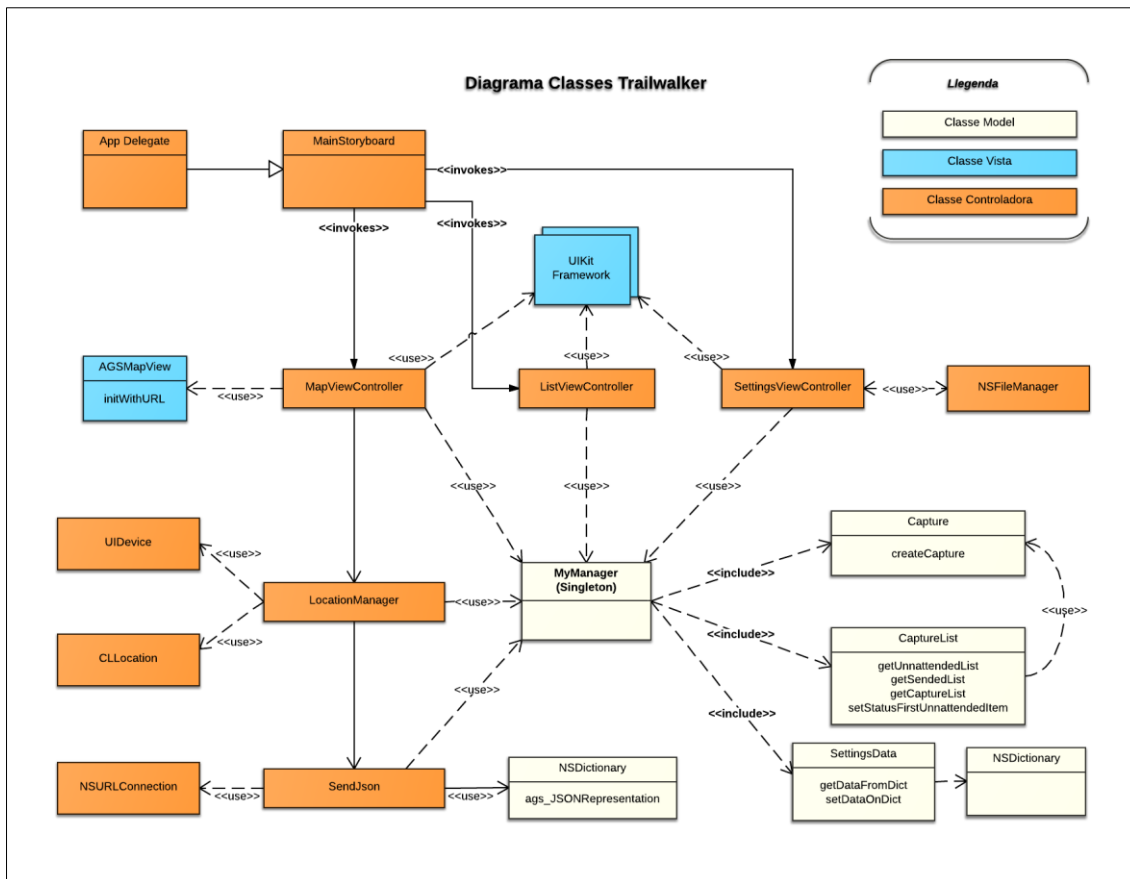


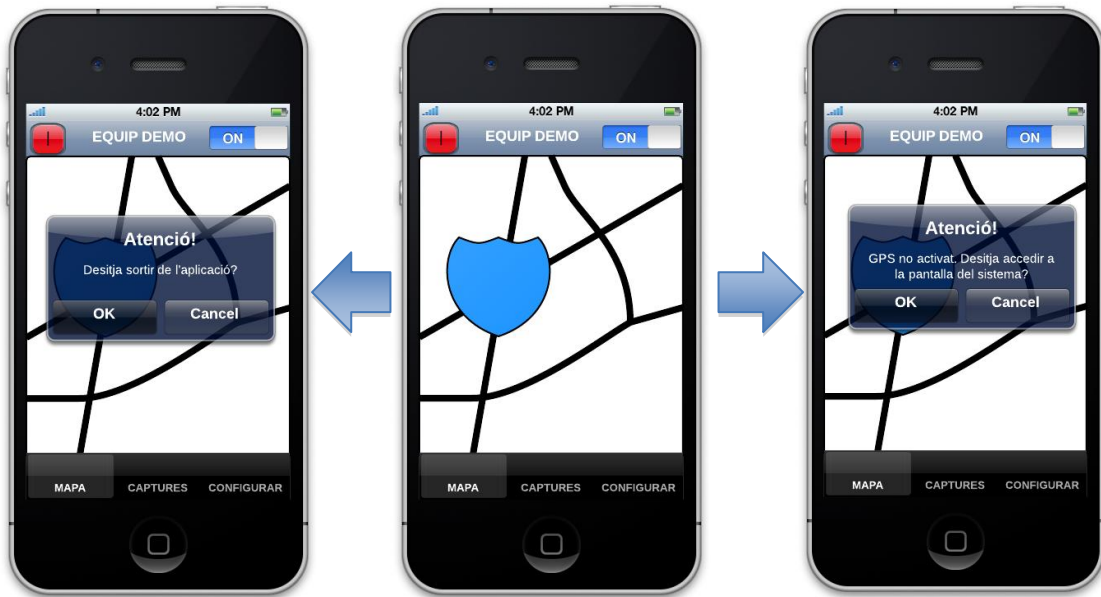
Figura 7: Diagrama de classes aplicació iOS Trailwalker

8.4. Prototipus de pantalles

La pantalla principal de l'aplicació, mostrarà el mapa de la zona centrant la vista en la ubicació actual, dos botons i un menú horitzontal inferior.

Dels botons, un d'ells serà de tipus interruptor i servirà per activar i desactivar el procés de captura d'informació. En cas d'activar la captura i no tenir el servei GPS de l'smartphone habilitat, es mostrarà un missatge per pantalla avisant a l'usuari del fet.

L'altre botó serveix per apagar definitivament l'aplicació, previ avís de confirmació a l'usuari.



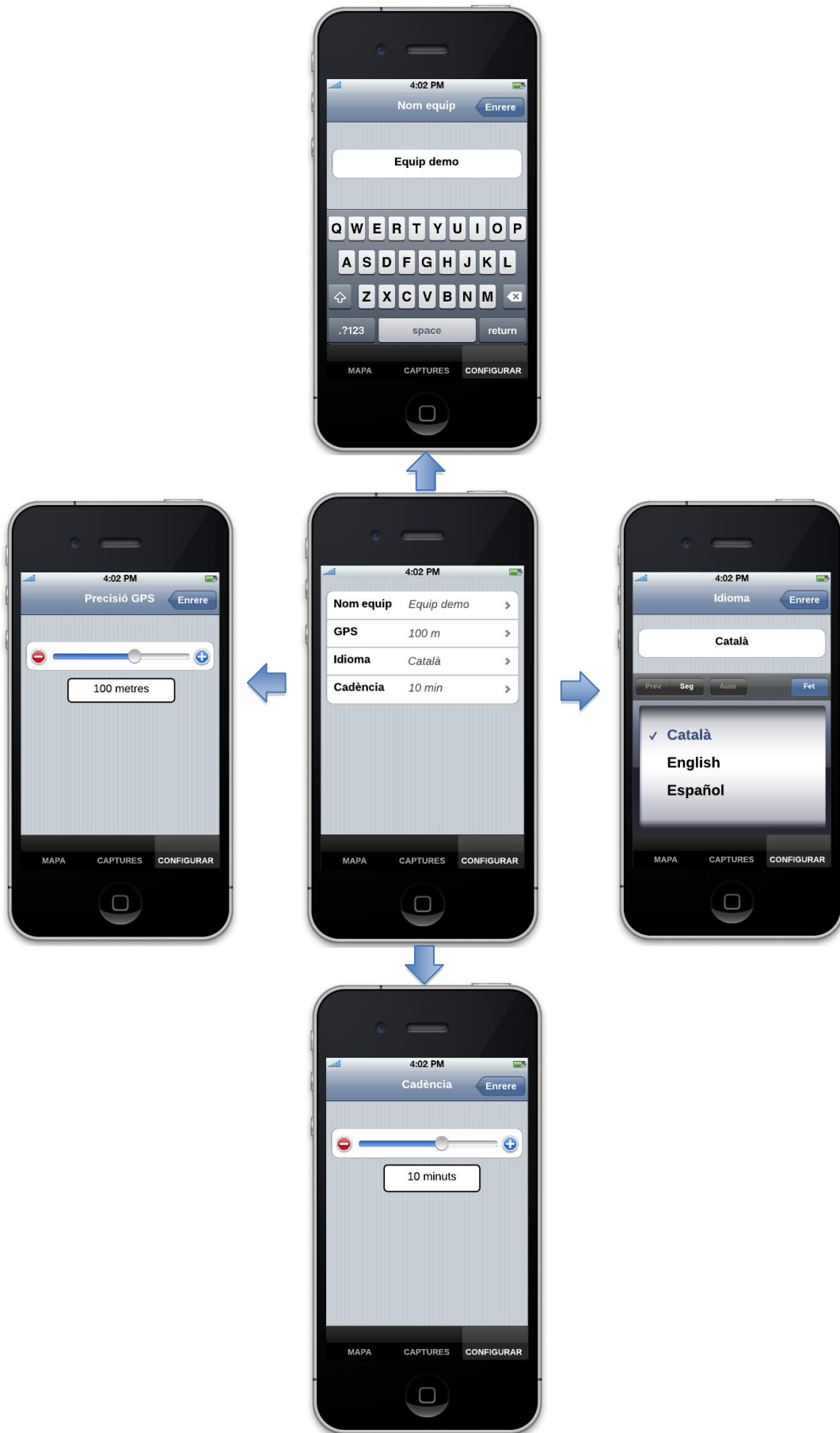
Amb el menú inferior es pot navegar per les altres opcions. La segona pantalla de l'aplicació, mostra un llistat de les diferents captures realitzades identificant l'estat de les mateixes mitjançant un codi de colors; verd per les captures enviades i groc per les captures pendents. La informació mostrada serà la data i hora de la captura, el nivell de la bateria i la ubicació capturada. A la part inferior, es mostrarà un resum del nombre de captures en funció de l'estat.



La última pantalla accessible mitjançant el menú inferior, serà la de configuració dels paràmetres de l'aplicació. Els elements configurables són els següents:

- Nom de l'equip.
- Nivell de precisió del GPS.
- Idioma.
- Cadència temporal de captures de posició.

MONITORITZACIÓ TRAILWALKER



9. PROGRAMACIÓ EN iOS

En aquest apartat es pretén donar una visió detallada de com es prepara un entorn adequat per dur a terme el desenvolupament d'aplicacions iOS. Addicionalment s'expliquen els diferents patrons de disseny i bones pràctiques que recomana Apple i s'identifica on han estat utilitzats dins l'àmbit de l'aplicació TrailWalker.

9.1. Entorn de desenvolupament

Per tal d'emprendre el desenvolupament d'una aplicació sota la plataforma iOS o OSX, és necessari instal·lar l'Entorn de Desenvolupament Integrat (IDE) propi d'Apple anomenat Xcode que pot ser descarregat gratuïtament en el següent enllaç de la Mac App Store: itunes.apple.com/us/app/xcode/id497799835?mt=12

Aquesta eina sols pot ser instal·lada en màquines governades pels sistemes operatius OS X Mountain Lion i OS X Lion.

Xcode incorpora nativament un compilador del llenguatge de programació Objective C, que és el que s'utilitza principalment per desenvolupar aplicacions iOS i OS X, tot i que també permet compilar codi escrit en C i en C++, que s'utilitza habitualment per a la creació de llibreries i frameworks.

Apart de l'habitual ajuda sintàctica que incorporen tots els IDE's, aporta una sèrie d'eines addicionals per a millorar i facilitar l'escriptura de codi font, entre les que es troben:

- Els últims Kits de Desenvolupament de Software (*SDK*) d'iOS i Mac OS X.
- Una interfase d'usuari per el disseny de les aplicacions.
- Eines de debugació i testing.
- Un simulador iOS de dispositiu mòbil per a la depuració del codi.
- Eines d'anàlisis de la gestió dels recursos utilitzats.

9.2. Utilització Frameworks

Els frameworks són biblioteques que poden ser incorporades als nostres projectes de desenvolupament i que aporten una varietat de recursos empaquetats, tal com poden ser classes, imatges i pantalles.



Figura 8: Exemple de la utilització de frameworks.

<http://developer.apple.com/library/ios/#documentation/General/Conceptual/DevPedia-CocoaCore/Framework.html>

Existeixen una sèrie de frameworks que són afegits per defecte i que controlen els recursos bàsics del dispositiu utilitzat:

- **Foundation** → Accés a les diferents estructures de dades (cadena, data i hora, tuples) i serveis (manipulació d'URL's, configuració de ports i sockets de comunicació).
- **UIKit** → S'encarrega principalment de gestionar els recursos propis de la interfície d'usuari de l'aplicació desenvolupada (p.ex. Finestres, vistes, botons, caixes de text, etiquetes, etc...).
- **CoreGraphics** → És un framework de baix nivell encarregat de les accions gràfiques del sistema com són la renderització d'imatges, la gestió de colors, així com altres efectes visuals.

La llista completa de frameworks propis d'iOS pot ser consultada a la documentació d'Apple³.

³ Llista de frameworks: <https://developer.apple.com/library/IOS/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSFrameworks/iPhoneOSFrameworks.html>

En el nostre cas, apart dels frameworks per defecte, pel fet d'haver utilitzat un SDK propi d'ArcGIS, hem hagut d'afegir les referències a les següents llibreries:

- MobileCoreServices.framework
- MediaPlayer.framework
- QuartzCore.framework
- libz.dylib
- libc++.dylib
- CoreText.framework
- CoreLocation.framework
- Security.framework
- OpenGLES.framework

9.3. Llicència de desenvolupament

Una vegada descarregat i configurat apropiadament l'IDE d'Apple, disposem d'un entorn adequat per a la programació d'aplicacions iOS (tant per a dispositius mòbils com per a Mac OSX), així com eines de test per simulació, eines per facilitar la implementació de tests unitaris i eines d'anàlisi del consum de recursos.

En el cas de voler provar la aplicació construïda en un dispositiu final o en el cas de plantejar la publicació a la tenda d'aplicacions d'Apple (AppStore), es requereix la obtenció prèvia d'una llicència de desenvolupament.

Actualment existeixen 3 tipus diferents de programes que ho permeten⁴:

- **L'iOS Developer Program** és la llicència bàsica individual i la única que permet les 2 accions comentades anteriorment. És de pagament i té un cost de 99\$ anuals.
- **L'iOS Developer Enterprise Program** és una llicència enfocada a l'ús intern dins de companyies, que pretenen distribuir aplicacions dins l'àmbit de la seva pròpia empresa. També és de pagament i el seu preu és de 299\$ anuals.

⁴ Més informació: <https://developer.apple.com/programs/start/ios/>

- La última de les llicències disponibles és la **iOS Developer University Program**, destinada a la comunitat educativa i tot i ser gratuïta (prèvia aprovació d'Apple) només permet el testeig en dispositius finals; no permet la distribució de les apps en cap de les seves variants.

9.4. Patrons de disseny

Al llarg de la documentació aportada per la companyia Apple relativa al desenvolupament sota la seva plataforma, se'ns presenta un conjunt de bones pràctiques recomanades. Aquests patrons de disseny pretenen, per un costat, aportar les eines adequades per facilitar i optimitzar les aplicacions construïdes, i per altre banda, intenten que el programari resultant tingui una arquitectura sòlida i robusta.

9.4.1. Delegació entre classes

La delegació és una metodologia de programació que defineix una relació de coordinació entre 2 objectes. L'objecte que delega l'acció envia un missatge a la classe delegada abans de resoldre l'esdeveniment en qüestió i d'aquesta manera dona la oportunitat d'interceptar-lo i fer canviar en l'estat de l'aplicació en general.

L'objectiu principal d'aquest patró és el de centralitzar el tractament de diverses classes en un punt central.

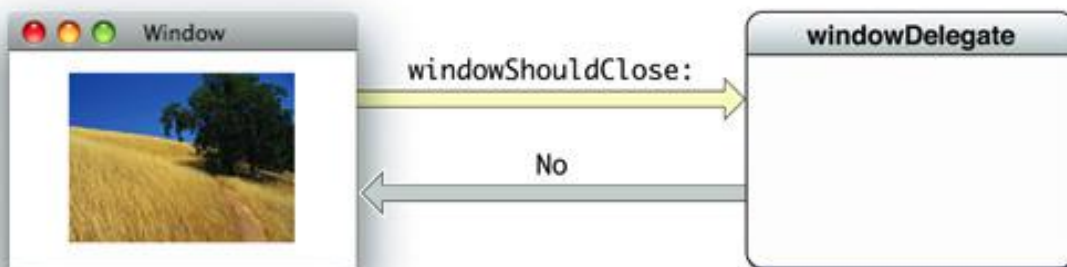


Figura 9: Exemple de delegació per la classe `NSWindow` de l'`AppKit`.

<http://developer.apple.com/library/ios/#documentation/General/Conceptual/DevPedia-CocoaCore/Delegation.html>

Aquesta metodologia pot posar-se en pràctica de dues formes diferents:

- Per un costat, podem implementar tant la classe que delega com la o les classes delegades. En el projecte que ens ocupa, hem implementat aquest cas per comunicar la pantalla de configuració amb la resta de les sub-pantalles. Aquestes últimes s'encarreguen d'obtenir les dades mitjançant els diferents elements d'entrada sortida de la llibreria UIKit, mentre que la primera s'encarrega de persistir les diferents dades en diccionari habilitat.
- Molt sovint ens trobem amb la imposició d'utilitzar el patró. Per a resoldre una funcionalitat concreta, Apple pot oferir-nos una llibreria oficial (un framework) que implementa la classe necessària i aquesta pot requerir la utilització del seu objecte delegat equivalent. En aquests casos, nosaltres com a desenvolupadors només ens hem de preocupar d'utilitzar la classe concreta i referenciar el delegat.

Aquesta estratègia ha estat utilitzada en diversos punts de l'app TrailWalker, com per exemple amb la classe de l'ArcGIS SDK que realitza la càrrega del servei de mapa (AGSMapViewLayer i AGSMapViewLayerDelegate) o amb la encarregada d'obtenir la posició de l'aparell (CLLocationManager i CLLocationManagerDelegate), entre d'altres.

9.4.2. Classes Singleton

Les Singleton són unes classes especials que tenen com a característica principal el fet que sempre retornen la mateixa instància de si mateixes, amb indiferència de quantes vegades són invocades.

Aquest fet és de vital importància en aquelles situacions que requereixen compartir els serveis o les dades d'una classe, des de diferents punts de l'aplicació.

Per garantir aquest punt centralitzat a l'objecte en qüestió, la classe té restriccions a la còpia i a la multi-instanciació, per tal d'assegurar la seva coherència.

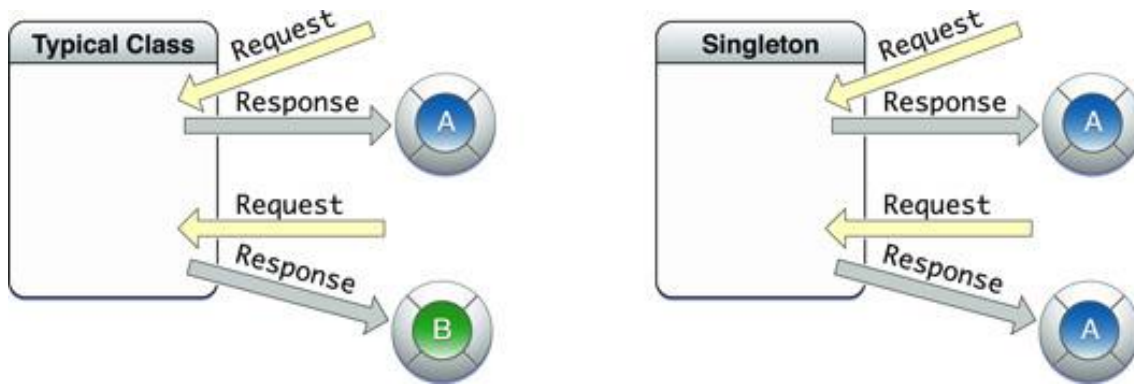


Figura 10: Esquema instància de classes Singleton.

<http://developer.apple.com/library/ios/#documentation/General/Conceptual/DevPedia-CocoaCore/Singleton.html>

En el cas que ens pertoca, ha estat adequada la utilització d'aquesta tipologia de classe per gestionar la llista de captures d'informació realitzada que utilitza els diferents controladors d'obtenció, enviament i consulta de les captures.

També s'ha utilitzat la tècnica per tal de compartir informació relativa al diccionari d'opcions de l'app.

9.4.3. Categories

El llenguatge Objective C també ens permet crear nous mètodes sobre classes ja existeixen en el sistema. Aquesta implementació es coneix com a categoria i ens permet ampliar la funcionalitat habitual.

Les classes ampliades per una categoria tant poden ser alienes a Apple (llibreries privades d'usuari), com sobretot, les estàndards proporcionades per Cocoa.

Hem pogut incorporar una categoria dins l'àmbit de la nostra aplicació per ampliar la funcionalitat de la classe que representa la cadena de text *NSString*. D'aquesta forma hem incorporat el mètode *stringByURLEncoding*, que s'encarrega de realitzar la transformació dels caràcters especials de la cadena de connexió Json amb el format que requereix el servei REST d'ArcGIS que ens ha proporcionat la UOC.

9.4.4. Execució en segon pla (Background)

Les aplicacions iOS segueixen un cicle de vida que va transcorrent al llarg de diferents estats. Aquests flux d'estats, ajuda al sistema a gestionar els mitjans necessaris en cada moment amb l'objectiu d'evitar un malbaratament dels

recursos, tal com són la quantitat de memòria disponible en el dispositiu o el nivell de bateria restant.

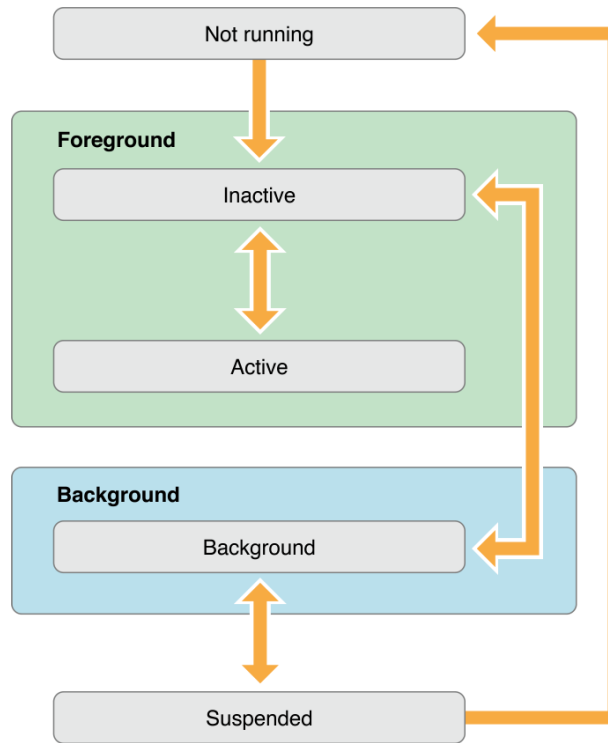


Figura 11: Transició d'estats d'una app d'iOS.

<http://developer.apple.com/library/ios/ - documentation/iPhone/Conceptual/iPhoneOS ProgrammingGuide/ManagingYourApplicationsFlow/ManagingYourApplicationsFlow.html>

En termes generals una aplicació pot estar executant-se en primer o en segon pla.

En el primer pla, l'aplicació és visible a la pantalla del dispositiu i pot ser interactuada mitjançant diferents esdeveniments tàctils.

En canvi, quan una aplicació no es interactuada durant un període de temps o és forçada a 'dormir' mitjançant el botó corresponent, passa a un estat de segon pla durant un període limitat de temps (que acostuma a ser de 10 minuts) i sempre i quan, es continui executant codi.

En el cas que finalitzin totes les tasques pendents o es sobrepassi el període de temps comentat i no existeix alguna característica especial relacionada amb l'app que ho impedeixi, aquesta passarà a un estat terminal de suspensió.

En aquest últim estat, no es garanteix la vida de l'aplicació i es delega en el gestor del sistema operatiu la seva finalització amb l'objectiu d'alliberar memòria.

En el cas del TrailWalker, degut a que tenim un cas d'ús que justifica la utilització del segon pla de forma permanent, com és la monitorització de la posició del dispositiu, definirem el paràmetre adequat per tal de mantenir viu el procés de captura i enviament de posició, evitant els límits imposats per Apple en quant a la utilització del mode en segon pla.

Per poder controlar els estats des del codi font de l'aplicació, l'Objective C ens proporciona una sèrie de mètodes dins de la classe App Delegate (la classe inicial de tota app):

- `application:willFinishLaunchingWithOptions:` .
- `application:didFinishLaunchingWithOptions:` .
- `applicationDidBecomeActive:` .
- `applicationWillResignActive:` .
- `applicationDidEnterBackground:` .
- `applicationWillEnterForeground:` .
- `applicationWillTerminate:` .

9.4.1. Localització i Internacionalització

La localització i la internacionalització són dos conceptes que en conjunt permeten que la experiència d'un usuari no pateixi cap davallada pel fet pertànyer a un país o cultura diferent per la qual estava inicialment dissenyada l'aplicació. En concret, diríem que internacionalització refereix al disseny i a la construcció necessària per a dur a terme el procés de localització.

La localització abraça diferents aspectes com poden ser la traducció de text en un idioma concret, la utilització de simbologia pròpia, la utilització d'imatges o icones més apropiats culturalment o fins i tot, la modificació del nom de l'aplicació per un de més adequat.

En el nostre cas, ens hem centrat en la tasca de localització més habitual que refereix a la traducció del text estàtic a les diferents llengües per les que està preparada l'aplicació. En concret l'aplicació TrailWalker està disponible en els idiomes següents:

- Català
- Castellà
- Anglès
- Francès

La forma d'incorporar aquest tipus de localització a l'aplicació és molt senzilla. El disseny de la interfície d'usuari s'ha realitzar sobre una entorn de treball anomenat Storyboard que utilitza una localització base (per defecte l'idioma anglès d'Estats Units). En el moment en que es decideix incorporar una nova localització, s'escull

un locale que representa una llengua dins d'un país mitjançant una codificació ISO estàndard (en el cas del català, l'identificador és ca-ES).

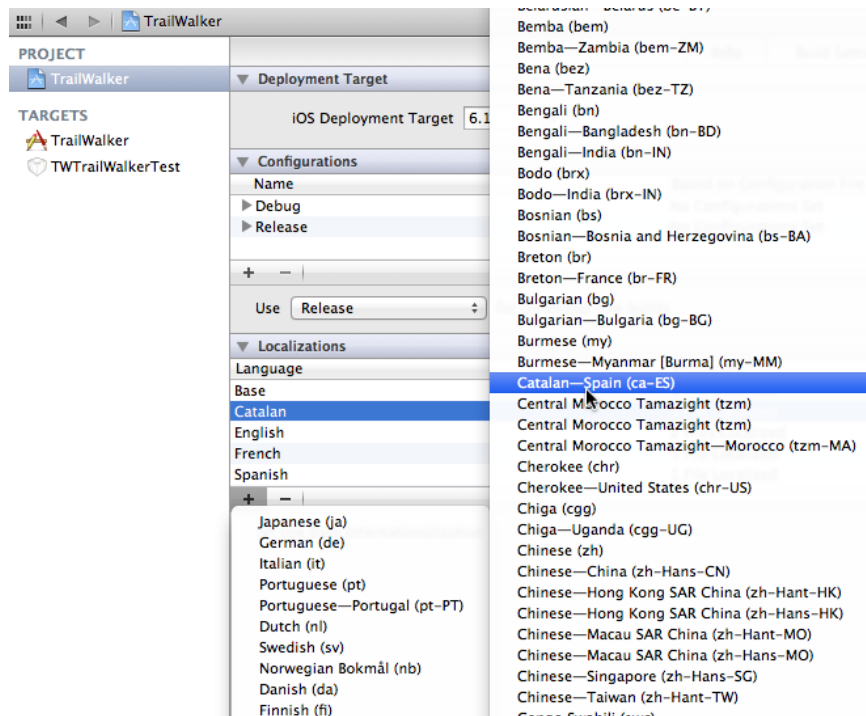


Figura 12: Selecció de nova localització dins Xcode.

Una vegada seleccionat el locale, s'ha d'indicar si la localització escollida es definirà a nivell de traducció de llengua, a nivell de característiques de l'aplicació o d'ambdues opcions.

En el nostre cas, havent seleccionat sols traducció, aconseguim que es generi automàticament un arxiu anomenat MainStoryboard.strings (Catalan) que conté un a un, tots els textos estàtics de l'aplicació per a cada component de la interfície gràfica amb els mateixos valors de la base. Sols resta la feina a traduir cada un dels valors dels components a la llengua desitjada. A mode d'exemple es mostra un fragment de l'arxiu strings de l'aplicació TrailWalker:

```
/* Class = "IBUITabBarItem"; title = "CAPTURES"; ObjectID = "6"; */
"6.title" = "CAPTURES";

/* Class = "IBUITabBarItem"; title = "MAP"; ObjectID = "7"; */
"7.title" = "MAPA";

/* Class = "IBUINavigationControllerItem"; title = "Frequency"; ObjectID = "6oM-hJ-KEW"; */
"6oM-hJ-KEW.title" = "Freqüència";

/* Class = "IBUILabel"; text = "Frequency"; ObjectID = "CHy-m9-FQb"; */
"CHy-m9-FQb.text" = "Freqüència";

/* Class = "IBUILabel"; text = "Team name"; ObjectID = "Id3-Nm-N2K"; */
"Id3-Nm-N2K.text" = "Nom equip";
```

```

/* Class = "IBUITabBarItem"; title = "CONFIGURATION"; ObjectID = "M5t-ZM-HJ6"; */
"M5t-ZM-HJ6.title" = "CONFIGURACIÓ";

/* Class = "IBUINavigationItem"; title = "Language"; ObjectID = "OWJ-G5-0AK"; */
"OWJ-G5-0AK.title" = "Idioma";

/* Class = "IBUINavigationItem"; title = "Captures"; ObjectID = "PcP-iM-hJg"; */
"PcP-iM-hJg.title" = "Captures";

/* Class = "IBUITableViewSection"; headerTitle = "Capture frequency"; ObjectID = "RwQ-Ec-bGd"; */
"RwQ-Ec-bGd.headerTitle" = "Freqüència captures";

/* Class = "IBUINavigationItem"; title = "Map"; ObjectID = "SqO-wB-T1d"; */
"SqO-wB-T1d.title" = "Mapa";

```

La llengua en la que es mostra l'aplicació TrailWalker és la mateixa en la que està el funcionant el dispositiu. En el cas que s'utilitzi la opció de sistema Configuració > General > Internacional > Idioma (en iOS 6) per canviar l'idioma, s'actualitzarà el de l'aplicació en el mateix moment, sempre que sigui un dels descrites anteriorment, en cas contrari, es mostrarà l'idioma del locale base (anglès - EU).



Figura 13: Opció de configuració de l'idioma en el dispositiu.

10. BANC DE PROVES

Cap projecte de desenvolupament de programari, amb independència del llenguatge de codificació utilitzat i de la grandària de l'aplicatiu construït, pot considerar-se finalitzat i preparat per la seva posada en marxa, si no s'ha definit, implementat i executat amb èxit un cicle complet de proves.

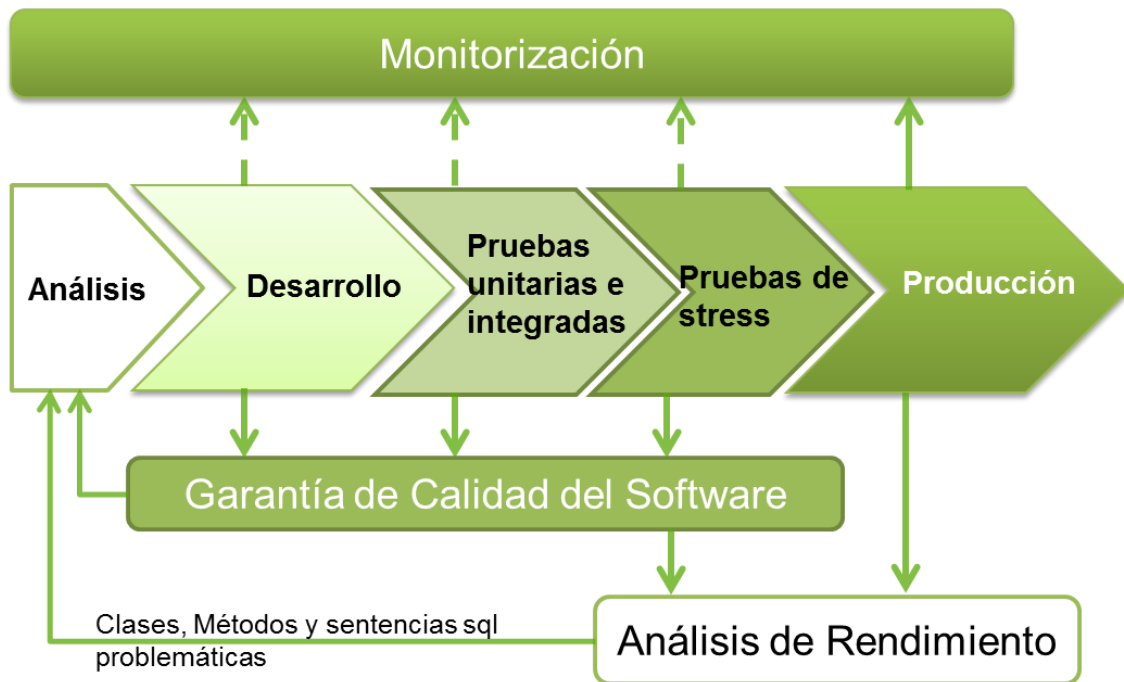


Figura 14: *Cicle de vida típic d'una aplicació informàtica.*

<http://www.b-clvr.com/html/>

Les diferents fases de proves plantejades per tal de garantir el correcte funcionament de l'app TrailWalker, han estat les següents:

- Proves unitàries
- Anàlisi de rendiment
- Proves en camp (integrades)
- Proves d'estrès

10.1. Proves unitàries

La primera fase de proves que ha de superar qualsevol programari realitzat, és la de proves unitàries. Aquestes proves tenen com a objectiu comprovar el correcte

funcionament per separat de cada una de les unitats mínimes que formen un programari.

En el cas de l'aplicació TrailWalker i degut a la metodologia de programació implementada (Orientació a Objectes), s'ha escollit com a unitat mínima representativa la entitat classe.

En el cas de l'Xcode, el propi IDE incorpora un framework anomenat OCUit que facilita la implementació d'aquest tipus de proves i les integra fàcilment dins del projecte realitzat.

La forma més senzilla d'habilitar l'OCUnit a un projecte de desenvolupament iOS, és indicar-ho en el moment inicial de la creació, d'aquesta forma, el propi Xcode s'encarrega de realitzar totes les accions necessàries. En cas contrari, l'usuari haurà d'afegir manualment les llibreries requerides, incorporar el paquet Cocoa Touch Unit Testing Bundle i especificar manualment la relació entre els dos projectes.

Una vegada que hem pogut adequar l'entorn de programació per a la fase de test, se'ns habilita una grup de macros per a la construcció dels tests unitaris. Tenen una similitud als mètodes de classes utilitzats fins ara, però es diferencien en que cada un d'ells implementa internament una condició lògica, amb o sense paràmetres d'entrada, que en el cas de no complir-se, envia un error a la consola d'incidències de l'editor.

D'entre tot el ventall de macros⁵ que disposem, hem considerat necessari utilitzar les següents:

- STAssertNotNil → Retorna un error en el cas de que el paràmetre avaluat tingui el valor nul, que no té cap valor definit.
- STAssertEquals → Retorna un error en el cas de que les dos variables avaluades tinguin valors diferents.
- STAssertEqualObjects → Retorna un error en el cas de que els dos objectes avaluats siguin diferents.

⁵ Llistat complet: https://developer.apple.com/library/mac/#documentation/DeveloperTools/Conceptual/UnitTesting/AB-Unit-Test_Result_Macro_Reference/result_macro_reference.html

- STAssertTrue → És la macro més versàtil que tenim a la nostre disposició, ja que retorna un error sempre que no es compleixi una condició lògica que li haguem definit com a paràmetre.

Tal com s'havia comentat en un apartat anterior, hem seguit el patró de disseny M-V-C, que divideix les classes entre objectes de tipus Model, Vista i Controlador.

Les classes de tipus Vista no són gaire bones candidates a la hora d'implementar els tests unitaris, ja que la seva funció és realitzar la definició de la interfície gràfica i relacionar-hi els events, cosa que automatitza el component StoryBoard de l'IDE.

Hem preparat tests unitaris per a la resta de tipologies de classes:

- TWLocationManager
- TWSendJson
- NSString+UrlEncoding
- TWCaptura
- TWCaptureList
- TWSettingsData

A mode d'exemple llistem a continuació la implementació del test realitzat per a la classe TWCaptura.h:

```
static int const prevLat = 5;
static int const prevLong = 15;

#import "TWCapturaTest.h"

@implementation TWCapturaTest {
    TWCaptura *capture;
    CLLocation *location;
}

- (void)setUp
{
    [super setUp];
    capture = [[TWCaptura alloc] init];
    location = [[CLLocation alloc] initWithLatitude:prevLat longitude:prevLong];
}

- (void)tearDown
{
    capture = nil;
    location = nil;
    [super tearDown];
}

- (void)testCaptureConstructorReturnsNotNil
{
    STAssertNotNil(capture, @"Instancia d'objecte nul.");
}

- (void)testCreateCaptureSetAllData
{
    [capture createCapture:location];
    STAssertTrue(
        ([[capture.latitudeCapt intValue] == prevLat],
        @"Latitud de la captura mal grabada.");
    );
}
```

```

    STAssertTrue(((capture.longitudeCapt intValue) == prevLong),
        @"Longitud de la captura mal grabada.");
}
@end

```

Una vegada construïts els diferents tests unitaris i havent-los executat tots amb un resultat satisfactori obtenim, per part d'Xcode, un resum com el següent:

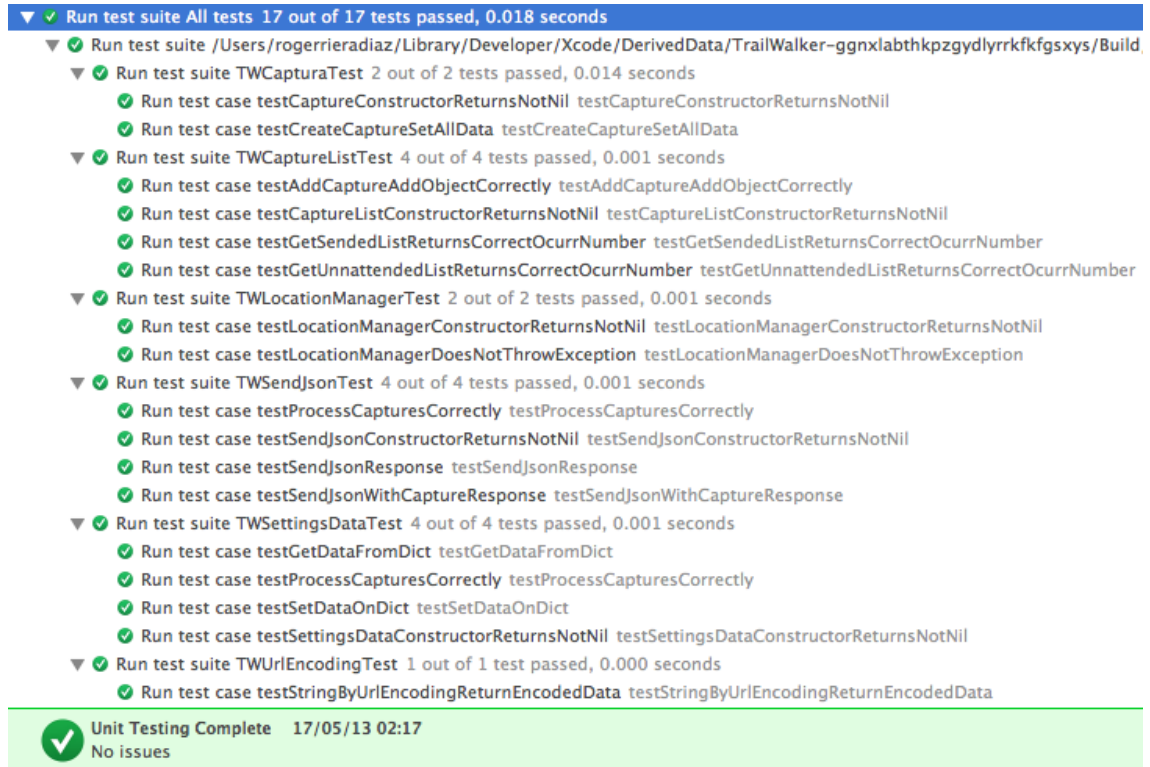


Figura 15: Resultat execució de test OUnit a Xcode.

10.2. Anàlisi de rendiment

Un cop superades les proves unitàries podem determinar, fins a cert punt, que el codi implementat està lliure d'errors de programació. El següent pas és analitzar fins a quin punt l'aplicació creada realitza una gestió adequada del recursos del dispositiu.

Novament, és el propi IDE d'Apple el qui incorpora una eina específica anomenada *Instruments*, per poder realitzar aquest feina.

L'*Instruments* ens permet seleccionar una sèrie de plantilles amb les que rastrejar diferents aspectes del dispositiu durant la execució de la nostra app. En el nostre cas concret, utilitzarem la Energy Diagnostics que tal com indica el nom, ens permet veure el consum de bateria en viu.

Per tal de realitzar un estudi prou significatiu realitzarem la següent seqüència d'accions sobre un mòbil iPhone 5 amb pantalla retina i el sistema iOS versió 6.1:

1. Arranquem l'aplicació TrailWalker.
2. La configurem per tal de realitzar una captura de posició pel mètode estàndard amb una cadència de 10 segons.
3. Arranquem el procés de captura.
4. Naveguem pel mapa i ampliem i reduïm la nostra posició actual.
5. Esperem un minut per donar temps a realitzar unes quantes captures.
6. Consultem la llista de captures.
7. Deixem la aplicació en segon terme.
8. La tornem a encendre i consultem de nou les captures realitzades.
9. Parem la captura de posicions i apaguem TrailWalker.

Una vegada realitzat aquest procés obtenim una gràfica com la següent:

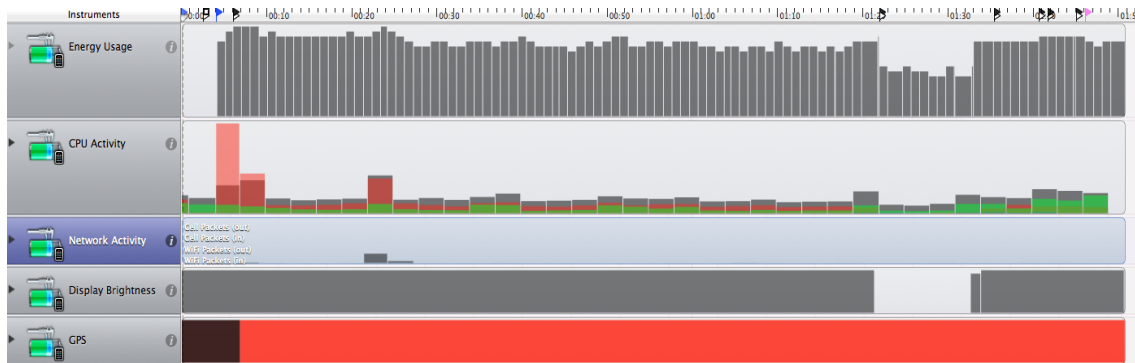


Figura 16: Gràfica de rendiment d'Instruments per a l'app TrailWalker.

Analitzant un per un, els diferents moments claus establerts, Instruments ens dona el següent detall:

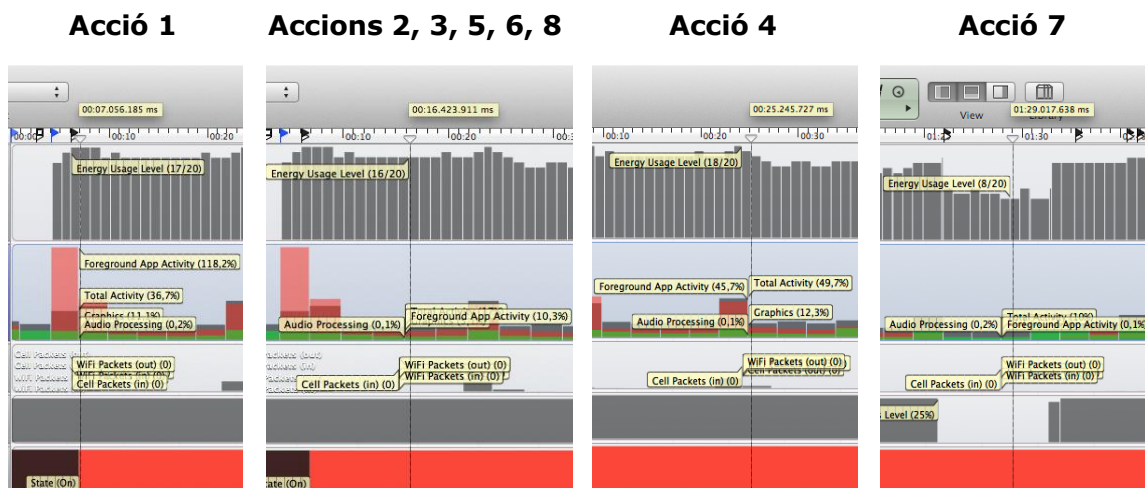


Figura 17: Valor de l'anàlisi de rendiment d'Instruments per a les diferents funcionalitats de l'app TrailWalker.

La primera de les accions és la que requereix un major consum de computació i d'energia, però no pot ser considerada crítica al realitzar-se una sola vegada.

Una cosa similar ocorre amb l'acció 4. L'aplicació permet fer ús del mapa, però no és la funcionalitat principal.

Per les accions 2, 3, 5, 6 i 8, veiem que el consum de CPU baixa dràsticament, però no així el consum d'energia, que queda establert en un nivell mitjà de 16 sobre 20.

L'acció 7 és la que considerariem com l'estat habitual de l'app. Està en segon terme capturant i enviant la posició. En aquest cas veiem que el consum de CPU es gairebé inapreciable i que el consum d'energia està per sota de la meitat (8/20).

Podem determinar que el major consum d'energia correspon a la visualització de la pantalla. En el moment en que es queda en segon terme, obtenim unes estadístiques assumibles.

10.3. Proves de camp

Una vegada finalitzats els anàlisis tècnics del programari, necessitem realitzar una bateria de proves sobre el terreny, per tal de comprovar la eficàcia dels 2 mètodes de captura implementats i verificar el seu funcionament en el dispositiu final. L'objectiu d'aquest test és el següent:

- Confirmar que els ambdós sistemes es comporten segons el disseny realitzat i segons les especificacions que els defineixen.
- Analitzar i comparar el consum de bateria durant un procés significatiu de captura.

Per a la seva realització, s'han tingut en compte els següents condicionants:

Model de dispositiu:	iPhone 5 amb pantalla retina.
Versió iOS instal·lada	6.1.4 (10B350)
Entorn de la prova	Urbà (àrea metropolitana)
Temps total de la prova	280 minuts
Mode execució aplicació	En segon pla.
Variabilitat de condicions	Pèrdua de senyal 3G ocasional (metro). Saturació de xarxa (aglomeració). Alta demanda de servei GPS (aglomeració). Diferent moments horaris (matí/nit).

10.3.1. Canvi significant d'ubicació

Recordem que aquest sistema de localització estableix uns paràmetres de freqüència temporal i distància de captura prefixats en 5 minuts i 500 metres. El resultat de la prova és el següent:

- **Temps total de la prova:** 280 minuts
- **Nombre de captures:** 36
- **Consum de bateria:** 40%

Una vegada finalitzat el procés de seguiment, es verifica que cada una de les captures realitzades pel dispositiu, han estat rebudes pel servidor ArcGIS i que la informació coincideix en els dos sistemes, per tant, descartem pèrdua o corrupció de la informació enviada.

A continuació, s'analitza la freqüència temporal i espacial entre captures. Per realitzar aquesta prova, ens servim d'un visor web proporcionat per la UOC, que mostra la capa vectorial dels punts enviats per la nostra aplicació client:

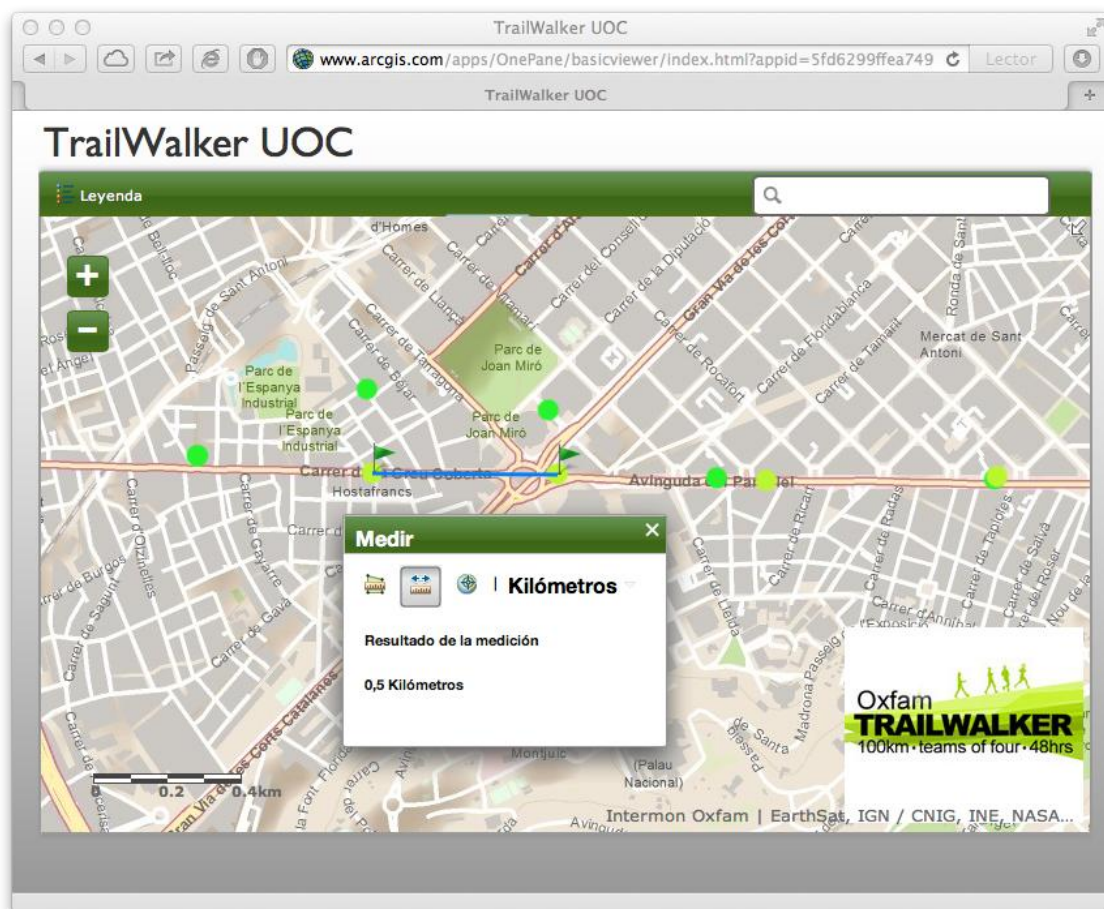


Figura 18: Visor web TrailWalker. Mesura d'una ruta pedestre.

Pels punts seleccionats, podem constatar que la distància entre ells és de 500 metres que s'han cobert en un període de 7 minuts, per tant, confirmem que aquest cas compleix amb les característiques publicades per Apple del servei de localització per canvi significatiu.

En una prova posterior d'aquest mateix mètode, però realitzant la ruta en cotxe, hem pogut comprovar que l'esdeveniment disparador de la captura era el mínim temporal de 5 minuts, degut a que en aquest període s'havien recorregut distàncies majors a 7 km per autopista:

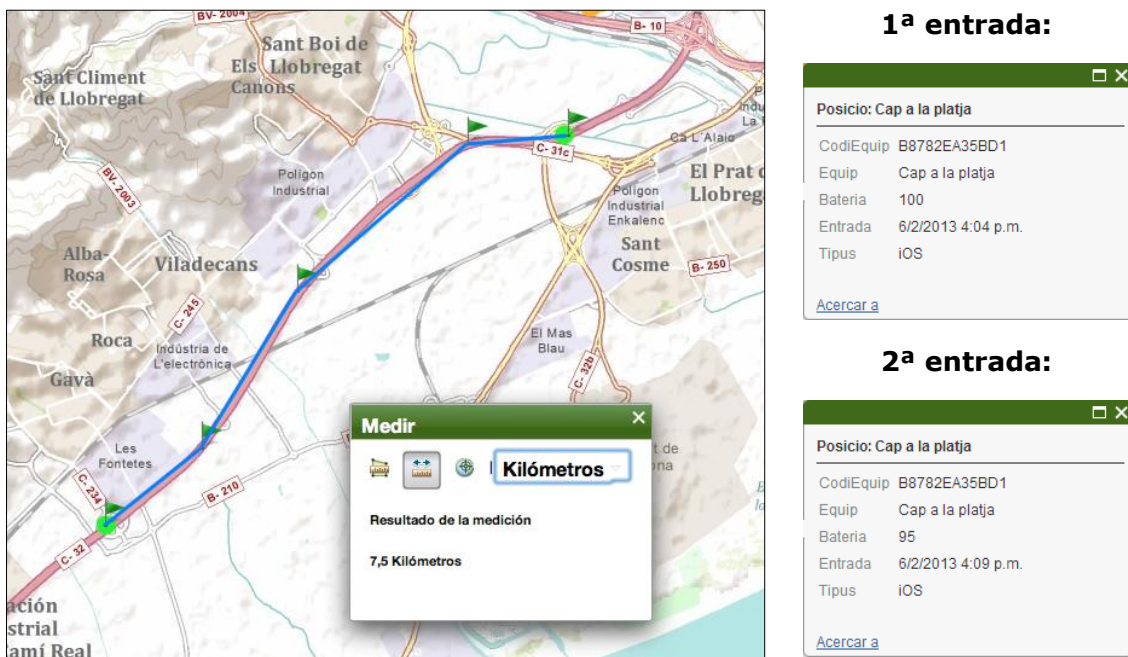


Figura 19: Visor web TrailWalker. Mesura d'una ruta automobilística.

10.3.2. Sistema d'ubicació estàndard

Per les proves del sistema d'ubicació estàndard hem parametrizat una cadència de captura de 10 minuts i un filtre de distància de 200 metres. El resultat de la prova és el següent:

- **Temps total de la prova:** 280 minuts
- **Nombre de captures:** 28
- **Consum de bateria:** 50%

Primerament, constatem que el consum de bateria és lleugerament superior al del sistema de canvi significatiu, tot i haver realitzat un nombre de captures menor durant aquest període. Això es deu a que el sistema estàndard realitza un major ús

del GPS de l'iPhone per obtenir la localització actual, component que té un major cost energètic que els utilitzats en la localització mitjançant punts d'accés Wi-Fi i torres de telefonia mòbil.

Hem realitzat la mateixa comprovació que en el sistema de canvi significatiu per descartar pèrdua o corrupció de la informació enviada.

En el moment d'analitzar la freqüència temporal, hem detectat alguns casos en els que la hora exacta de la captura era lleugerament anterior a la prevista pel procés cíclic. Hem detectat 2 casos de 28 amb una desviació aproximada d'1 minut. Aquest fet es produeix per que en el moment de recuperar la ubicació, el sistema detecta una localització recent generada per un altre procés del dispositiu i convé en aprofitar-la. Considerem acceptable aquesta desviació.

10.4. Proves d'estrès

Les proves d'estrès s'han centrat en el mètode estàndard de localització degut a que és l'únic que permet parametritzar la cadència de captura. L'objectiu d'aquest tipus de prova era doble. Per un costat, es volia comprovar el comportament de tot el sistema integrat davant d'unes condicions fictícies d'alt rendiment, accelerant el procés de captura a 10 segons. Es volia comprovar si el dispositiu era capaç d'obtenir ubicacions diferenciades en un període tan curt de temps, i per altra banda, es volia verificar la resposta del servidor ArcGIS davant d'un elevat nombre de peticions consecutives. El segon objectiu tornava a ser la mesura del consum de bateria del dispositiu, per contrastar amb les condicions de proves anteriors.

Els resultats són els següents:

- **Temps total de la prova:** 12 minuts
- **Nombre de captures:** 68
- **Consum de bateria:** 10%

Els resultats no han detectat respostes negatives del servidor GIS i les captures són pròximes entre si, però diferenciades. Determinem una bona tolerància a les condicions definides i podem extrapolar una autonomia final aproximada de 2 hores, dràsticament inferior a les obtingudes amb una freqüència de 10 minuts.

11. CONCLUSIONS

11.1. Sobre la implementació

Durant la fase de disseny s'han pres una sèrie de decisions relatives a la implementació, que posteriorment han estat variades per diversos motius:

11.1.1. Identificador únic de l'equip

En un primer moment s'havia escollit la mateixa estratègia que en els treballs anterior, pel que fa a la identificació dels equips participants mitjançant el codi IMSI particular per a cada targeta SIM.

El principal argument per a la utilització d'aquest codi, era el de continuar identificant unívocament un equip una vegada realitzat el canvi de dispositiu en el transcurs de la cursa. Aquest argument perd valor si considerem un aspecte purament pràctic; la substitució de la targeta SIM en la totalitat de models d'iPhone no és senzilla, sobretot per haver-se de realitzar a l'aire lliure i considerant condicions de poca llum. De totes formes, en el moment que s'ha intentat obtenir aquesta informació programàticament, ens hem trobat amb la negativa per part d'Apple al vulnerar les normes sobre la publicació d'aplicacions a la AppStore⁶:

"For security reasons, iPhone OS restricts an application (including its preferences and data) to a unique location in the file system. This restriction is part of the security feature known as the application's "sandbox." The sandbox is a set of fine-grained controls limiting an application's access to files, preferences, network resources, hardware, and so on."

Posteriorment, s'ha intentat utilitzar el codi UDID com a alternativa, assumint que només identificaria el dispositiu, però tot i així, ha estat descartat per que el mètode per recuperar-lo està marcat com a deprecat (a rebutjar) per Apple des de la versió 5.0 d'iOS, possiblement per la mateixa raó que amb l'IMSI.

Finalment, s'ha pres la decisió d'utilitzar un codi únic UUID generat per nosaltres mateixos (amb l'ajuda d'una classe pròpia d'iOS) en el moment d'iniciar l'aplicació per primer cop en un dispositiu. Aquest codi està compost per 36 caràcters alfanumèrics i té la propietat de ser únic per l'equip en el que s'ha generat.

⁶ Guia completa de revisió d'apps: <https://developer.apple.com/appstore/guidelines.html>

Degut a que l'equip de la UOC finalment ens ha definit un camp propi per emmagatzemar l'UUID en el servei REST de captura de posició, hem decidit informar l'atribut codiEquip amb el valor de la direcció MAC de l'iPhone, que a dia d'avui, no vulnera la política de privacitat d'Apple.

11.1.2. Selecció del mètode de captura

Inicialment estava previst utilitzar únicament el mètode de captura estàndard d'iOS disparat per un temporitzador. Després d'aprofundir en la documentació d'Apple, hem considerat primordial incorporar el mètode de captura de canvi significatiu d'ubicació del que em parlat en un apartat anterior, i que està disponible en el sistema operatiu a partir de la versió 4.0. S'ha cregut innecessari realitzar cap mena de verificació relativa a la compatibilitat amb el sistema, degut a que les versions anteriors ocupen un nombre residual del parc total de dispositius Apple, segons les estadístiques recents⁷.

Per tal de poder realitzar proves sobre els 2 mètodes, hem incorporat una pantalla de selecció dins del menú de configuració de l'app TrailWalker. Cal destacar, que les opcions de freqüència temporal i precisió de la captura són paràmetres propis del mètode estàndard i per tant, quedaran des-habilitats al seleccionar l'altre sistema.

11.1.3. Selecció de l'idioma de l'aplicació

S'havia plantejat una pantalla de configuració pròpia de la TrailWalker per poder canviar l'idioma de l'aplicació, però s'ha constatat que aquesta acció ha de ser realitzada necessàriament per l'usuari mitjançant la pantalla de configuració Internacional del propi dispositiu. Aquesta opció, modifica l'idioma en el que es visualitzen totes les aplicacions del sistema. Per aquest motiu, hem eliminat aquesta opció dins de l'aplicació desenvolupada.

11.2. Sobre els resultats

Podem concloure, que en línies generals, l'aplicació construïda funciona adequadament des de diferents punts de vista.

⁷ Informe sobre utilització de versions d'iOS: <http://david-smith.org/iosversionstats/>

Realitza amb èxit la seva funció principal que és la de capturar i enviar la ubicació del dispositiu, a més, demostra ser robusta en situacions de poca o nul·la cobertura de dades, ja que genera una llista de posicions pendents d'enviament que és atesa en el moment en que torna a haver disponibilitat de línia.

Dels dos mètodes analitzats, el que obté uns millors resultats en quant a consum d'energia, és el de canvi significatiu d'ubicació, tot i que com a inconvenient ens trobem que la precisió és un xic menor i que no permet realitzar cap tipus de parametrització. En el cas de les proves de camp, ha estat realitzada caminant a un ritme constant. Gràcies a aquest fet, considerem que la freqüència de captura és bastant adequada per realitzar el seguiment d'una cursa de resistència. A més, el fet que el sistema limiti el temps d'actualització entre captures a un mínim de 5 minuts, ajudarà a que el consum de bateria no varii gaire negativament.

Després d'haver realitzat les proves de camp i fent una extrapolació dels resultats obtinguts, considerariem que en el cas d'utilitzar la mateixa configuració de l'equip de proves i de mantenir les condicions constants durant la totalitat de la cursa, es podria aconseguir una autonomia superior a les 11 hores, que podria ser millorada en cas de seguir totes les recomanacions de la guia d'Apple sobre optimització de la bateria⁸.

Òbviament no s'aconsegueix cobrir la totalitat de la prova, però tenint en compte la durabilitat habitual dels telèfons de tipus smartphone, podríem considerar uns resultats acceptables de cara a una implementació productiva real, sempre que es complementés amb el relleu de diversos dispositius dins de l'equip participant o amb el suport d'un equip de la organització que realitzi tasques de manteniment.

11.3. Millores futures

Tot i que el projecte ha complert amb gran part dels objectius plantejats inicialment, alguns altres aspectes han quedat pendents d'una major profunditat.

La gestió dels diferents serveis REST publicats, de les capes vectorials d'informació i la persistència en la BD espacial, ha vingut realitzada pel professorat de la UOC.

⁸ Consultar optimització d'ajustaments a <http://www.apple.com/es/batteries/iphone.html>

Es podria haver afegit una major funcionalitat GIS a l'aplicació per tal de recuperar informació d'altres serveis REST que permetessin la monitorització de les diferents captures prèvies de l'equip participant. Aquesta funcionalitat ha estat delegada al visor web.

També es podria investigar els avantatges del nou servei de localització d'Apple inclòs en la versió 6.0 del sistema, anomenat monitorització de regió basat en formes. Aquest sistema envia una alerta en el moment en el que l'iPhone traspasa el límit d'una o varies regions geogràfiques definides prèviament. En aquest cas, podríem definir diverses regions, com poden ser els actuals punts de control de les etapes de la cursa o la sortida i la meta de la cursa.

Finalment, podria haver estat interessant incloure algun mètode per tal de persistir localment les captures no transmeses en cas d'alguna fallada en el servidor ArcGIS i poder-les enviar amb posterioritat mitjançant un altre sistema de transmissió (p.ex. per correu electrònic, cloud storage).

12. BIBLIOGRAFIA

Documentació oficial d'Apple

- (1) *About iOS App Programming*. Consultat el Maig 2013, a <http://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/>
- (2) *Apple – iOS*. Consultat el Març 2013, a <http://www.apple.com/es/iphone/ios/>
- (3) *Choosing an iOS Developer Program*. Consultat el Maig 2013, a <https://developer.apple.com/programs/start/ios/>
- (4) *Cocoa Core Competencies*. Consultat Abril 2013, a <http://developer.apple.com/library/ios/#documentation/General/Conceptual/DevPedia-CocoaCore/>
- (5) *Location Awareness Programming Guide*, Consultat l'Abril de 2013, a <http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/LocationAwarenessPG>
- (6) *Start Developing iOS Apps Today*. Consultat Abril 2013, a <https://developer.apple.com/library/ios/#referencelibrary/GettingStarted/RoadMapiOS/index.html>
- (7) *Xcode Unit Testing Guide*. Consultat Abril 2013, a <https://developer.apple.com/library/mac/#documentation/DeveloperTools/Conceptual/UnitTesting/>

Documentació àmbit de la UOC

- (8) Baldomà Cases, Montse (2012). *Monitorització Trailwalker* - UOC
- (9) Beneito Montagut, Roser *Presentació de documents i elaboració de presentacions UOC* - UOC.
- (10) Jordà Vinardell, Albert (2012). *Monitorització Trailwalker* - UOC
- (11) Pérez Navarro, A., Botella Plana, A., Muñoz Bolas, A., Olivella, R., Olmedillas Hernández, J., & Rodríguez Lloret, J. (2009). *Sistemes d'informació geogràfica i geotelemàtica* - UOC. Barcelona: Eureka Media, SL.
- (12) Sáenz Higuera, Nita, Vidal Oltra, Rut. *Redacció de textos científicotècnics* - UOC.

Altra documentació

- (13) *Batalla de baterías*. Consultat Abril 2013, a <http://www.xatakamovil.com/mercado/batalla-de-baterias-que-telefono-tiene-mejor-autonomia>
- (14) *Catàleg Phone House Març de 2013* The Phone House Spain S.L.U.
- (15) *ComScore - anàlisi de mercat smartphone*. Consultat el Març 2013, a http://www.comscore.com/Insights/Press_Releases/2013/1/comScore_Reports_November_2012_U.S._Mobile_Subscriber_Market_Share
- (16) *Glyphish*. Consultat el Març 2013, a <http://www.glyphish.com/>
- (17) *ESRI - Productes*. Consultat el Març 2013, a <http://www.esri.com/products>
- (18) *Intermon Oxfam*. Consultat el Març 2013, a <http://www.intermonoxfam.org>
- (19) *iOS Multitasking: Background Location*. Consultat el Maig 2013, a <http://mobile.tutsplus.com/tutorials/iphone/ios-multitasking-background-location/>
- (20) *Kantarworldpanel - anàlisi de mercat plataforma*. Consultat el Març 2013, a <http://www.kantarworldpanel.com/global/News/news-articles/US-iOS-Maintains-Lead-Among-US-Smartphone-OS-Sales>
- (21) *Stack Overflow*. Consultat l'Abril i Maig 2013, a <http://stackoverflow.com>
- (22) *Wikipedia - iOS*. Consultat el Març 2013, a [http://es.wikipedia.org/wiki/IOS_\(sistema_operativo\)](http://es.wikipedia.org/wiki/IOS_(sistema_operativo))

13. GLOSSARI

2G(GSM)/GPRS/UMTS(3G): Diferents serveis i xarxes de transmissió de dades per telefonia mòbil. Es diferencien en la velocitat màxima teòrica que ofereixen.

Android: Sistema operatiu de Google per dispositius mòbils.

App: Abreviatura d'aplicació, defineix el programari creat per Apple.

Apple Inc: Empresa americana que produeix equips electrònics i programari.

AppStore: És la botiga virtual d'aplicacions d'Apple pels sistemes iOS i OSX.

ArcGIS: Oferta de programari SIG de la empresa ESRI.

Arduino: Plataforma de maquinari lliure basat amb processador ATmel.

CPU: Unitat Central de Procés contingut en tot dispositiu programable.

Google Drive: Servei d'emmagatzematge gratuït en línia de la empresa Google.

GPS: Sistema de Posicionament Global de navegació per satèl·lit.

HTTP: Protocol sense estat de transferència de dades utilitzat a Internet (www).

IDE: Entorn Desenvolupament Integrat format per editor, intèrpret i compilador.

iMac/iPad/iPhone/iPod: Família de productes de maquinari d'Apple.

iOS: Sistema operatiu d'Apple Inc per dispositius mòbils.

JSON: Format lleuger d'intercanvi de dades.

Locale: Conjunt de paràmetres que defineixen un idioma i un país.

MAC: Identificador de 48 bits que identifiquen un dispositiu de xarxa.

Objective-C: Llenguatge P.O.O. derivat del C per aplicacions iOS i OSX.

OSX: Sistema operatiu d'Apple Inc basat en Unix per a computadores Mac.

POST: Un dels 8 mètodes de petició utilitzats en el protocol HTTP.

REST: Interfase web simple d'intercanvi de dades mitjançant HTTP.

S.O.: Sistema Operatiu. Programari de control dels recursos d'un maquinari.

SDK: Acrònim de Software Development Kit. Conjunt d'eines de programació.

SIG o GIS: Sistema d'Informació Geogràfica.

SIM: Acrònim de Subscriber Identity Module. Targeta utilitzada en telefonia.

Smartphone: Nova generació de telèfons mòbils.

UDID: Identificador únic utilitzat en els dispositius d'Apple.

UUID: Identificador únic universal de software de 32 dígit hexadecimals.

WI-FI: Acrònim de Wireless Fidelity. Dispositiu de connexió sense fils.

WKID: Codificació de Sistema de Referència espacial utilitzat per ESRI.