

Llicències (memòria, treball pràctic i videopresentació)

(*Creative Commons*)

Aquest treball està subjecte —excepte que s'indiqui el contrari— a una llicència de Reconeixement - No Comercial – Sense Obra Derivada 2.5 Espanya de *Creative Commons*. Podeu copiar-lo, distribuir-lo i transmetre'l públicament sempre que en citeu l'autor i l'obra, no se'n faci un ús comercial i no se'n faci còpia derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es> .

UNIVERSITAT OBERTA DE CATALUNYA



Enginyeria Tècnica d'Informàtica de Gestió

MONITORITZACIÓ TRAILWALKER

Autor: Albert Jordà Vinardell

Direcció: Francesc Puigvert Pell

Professor Responsable: Antoni Pérez-Navarro

CURS 2012-13 (Setembre)

ÍNDEX

1.	INTRODUCCIÓ.....	5
2.	OBJECTIUS.....	6
3.	TASQUES, TEMPORITZACIÓ I FITES.....	7
3.1.	Tasques.....	7
3.2.	Fites.....	9
3.3.	Diagrama de Gantt.....	9
4.	ANÀLISI DE RISCOS.....	11
5.	MATERIAL.....	12
5.1.	Introducció.....	12
5.2.	Desenvolupament a mida.....	12
5.2.1.	ESRI ArcGIS.....	12
6.	ANÀLISI D'OPCIIONS TECNOLÒGIQUES.....	13
6.1.	Introducció.....	13
6.2.	Requisits.....	13
6.2.1.	Autonomia.....	13
6.2.2.	Marxadors.....	13
6.2.3.	Informació.....	13
6.2.4.	Despeses Associades.....	14
6.2.5.	Complexitat.....	14
6.2.6.	Resistència.....	14
6.2.7.	Cobertura de senyal.....	14
6.3.	Quadre Comparatiu.....	14
6.3.1.	Recerca de solucions.....	14
6.3.2.	No idoneïtat de les solucions trobades.....	15
6.3.3.	Comparativa de les solucions finalistes.....	15
6.4.	Solució Final.....	18
6.4.1.	Punts forts de la solució d' <i>Arduino</i>	18
6.4.2.	Punts dèbils de la solució d' <i>Arduino</i>	18
6.4.3.	Punts forts de la solució <i>Smartphone</i>	19
6.4.4.	Punts dèbils de la solució <i>Smartphone</i>	19
7.	SISTEMES <i>ANDROID</i> i ArcGIS.....	21
7.1.	Introducció.....	21
7.2.	<i>Android</i>	21
7.2.1.	Desenvolupament d'aplicacions per <i>Android</i> – <i>Android SDK</i>	22
7.2.2.	Posicionament i sistemes <i>Android</i>	23
7.2.3.	Posicionament i rendiment de bateria en sistemes <i>Android</i>	24
7.3.	ArcGIS for Server.....	25
7.3.1.	ArcGIS, REST, JSON i API de Javascript.....	25
8.	Descripció de la solució.....	27
8.1.	Introducció.....	27

8.2.	Arquitectura de la solució	27
8.3.	Vista física	27
8.4.	Vista de desplegament.....	28
8.5.	Tecnologies a utilitzar.....	30
8.5.1.	Aplicació feta a mida en Android, amb Android SDK	30
8.5.2.	Servidor ArcGIS de ESRI	31
8.5.3.	Servidor ArcGIS Trailwalker	31
8.5.4.	Servidor Web.....	33
8.6.	Casos d'ús.....	33
8.6.1.	Visualització de localització i rànquing d'equips.....	33
8.6.2.	Activació i desactivació de captura de posicions	33
8.6.3.	Modificació del nom de l'equip	34
8.6.4.	Modificació configuració aplicació.....	34
8.7.	Implementació Aplicació <i>Android</i> SDK	35
8.7.1.	Introducció.....	35
8.7.2.	Requeriments de l'aplicació per Android.....	35
8.7.3.	Disseny de la Interfície d'Usuari.....	36
8.7.4.	Estructura de classes i lògica de l'aplicació	36
9.	BANC DE PROVES	52
9.1.	Millors pràctiques en l'estalvi de consum de bateria en Android.....	52
9.2.	Descripció de banc de proves i resultats	52
10.	CONCLUSIONS	54
10.1.	Millores.....	55
11.	BIBLIOGRAFIA	56
12.	GLOSSARI	57
13.	ANNEX.....	59

ÍNDIX D'IL·LUSTRACIONS

Il·lustració 1: Diagrama de Gantt de les activitats del TFC	10
Il·lustració 2: Línia de temps en què una aplicació rep modificacions en la localització font: (Android Developer - SDK - Activity class).....	25
Il·lustració 3: Arquitectura global del sistema.....	28
Il·lustració 4: Diagrama de desplegament del sistema.....	29
Il·lustració 5: Diagrama de casos d'ús.....	35
Il·lustració 6: Diagrama de classes.....	37
Il·lustració 7: Diagrama de flux de dades en plantejament inicial	39
Il·lustració 8: Diagrama de flux de dades en plantejament final.....	40
Il·lustració 9: Visualització informació per pantalla	45
Il·lustració 10: Botó captura posicions - OFF.....	45
Il·lustració 11: Botó captura posicions - ON	45
Il·lustració 12: Captura de posicions - GPS no activat	46
Il·lustració 13: Modificació de nom de l'equip (menú)	46
Il·lustració 14: Modificació de nom de l'equip	46
Il·lustració 15: Tancament aplicació (menú).....	47
Il·lustració 16: Tancament aplicació.....	47
Il·lustració 17: Configuració aplicació (menú).....	48
Il·lustració 18: Configuració aplicació (avís).....	48
Il·lustració 19: Configuració aplicació.....	48
Il·lustració 20: Configuració d'ubicació (menú)	49
Il·lustració 21: Serveis d'ubicació.....	49
Il·lustració 22: Cicle de vida d'una <i>Activity</i> (Android Developer - SDK - Activity class).....	50
Il·lustració 23: Notificació de sistema de l'aplicació	51
Il·lustració 24: Notificació de Sistema i Barra d'estat.....	51
Il·lustració 25: Imatge posicions Banc de Proves	53

ÍNDIX DE TAULES

Taula 1: Detall de tasques del TFC.....	9
Taula 2: Comparativa de les solucions finalistes	17
Taula 3: Ús de versions d' <i>Android</i> a 3/12/2012 - (Wikipedia - Android)	30
°Taula 4: Millors pràctiques	52
Taula 5: Banc de Proves - Mesures realitzades	53

AGRAÏMENTS

Primer de tot, aquest treball no hagués vist la llum sense l'ajut, paciència i comprensió de la Mírian. També vull fer menció a l'ajuda i dedicació rebuda per part director del projecte, en Francesc.

1. INTRODUCCIÓ

Intermón Oxfam realitza anualment una cursa solidària mitjançant la qual és capaç de recaptar diners en forma de donacions. Aquesta cursa és anomenada *Trailwalker* i es realitza per equips. Hi ha la necessitat de poder tenir una localització de tots els equips a temps real, així com el rànquing, i que aquesta informació sigui pública (si pot ser, visible des de la mateixa plana web de la cursa: <http://trailwalker.intermonoxfam.org/>).

Aquest document defineix el pla de treball dissenyat per a la realització del treball de fi de carrera “Monitorització”, que intenta donar una solució viable a aquesta monitorització. En el pla s’hi inclouen els elements bàsics d’un projecte, i que han estat definits prèviament a l’execució del projecte:

- Objectius: allò que es persegueix en l’execució del projecte
- Estructura: aproximació de l’estructuració del document de memòria del projecte
- Tasques, Temporització i Fites: establiment de tasques a realitzar, amb fites marcades. Aquestes dades queden fixades en un calendari.
- Anàlisi de riscos: estudi de possibles incidències en l’execució del projecte, amb les corresponents accions correctives.
- Material: definició d’allò que s’emprarà en l’execució del projecte (programari, maquinari, ...)
- Bibliografia

2. OBJECTIUS

A banda de l'objectiu principal del projecte descrit en la introducció, el Treball de Fi de Carrera com a tal persegueix assolir una sèrie d'objectius Generals i d'Específics:

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

- Específics
 - o Analitzar de viabilitat de diverses solucions sota limitacions tant econòmiques com de d'autonomia.
 - o Justificar la solució proposada sota les condicions descrites.
 - o Aprendre a utilitzar el posicionament GPS d'un mòbil i saber localitzar l'usuari .
 - o Desenvolupar una aplicació o simulació que corrobore la viabilitat de la solució.

3. TASQUES, TEMPORITZACIÓ I FITES

Per dur a terme una correcta execució del projecte, es plantegen un seguit de tasques a realitzar. Aquestes s'han fixat en un calendari, indicant-ne la durada i les dependències entre elles (diagrama de *Gantt*). També es descriuen les fites d'execució del projecte, les dates clau respecte a entregues (PACs). Tota aquesta planificació s'ha realitzat en base a la disponibilitat i càrrega de feina setmanal màxima.

3.1. Tasques

El projecte s'agruparà en 5 tasques principals:

- Tasca 1: Estat de l'art de les tecnologies actuals
S'inicia el projecte amb una anàlisi de les diferents opcions tecnològiques disponibles en el mercat tenint en compte els requeriments del projecte. Com a resultat de l'anàlisi s'elaborarà un quadre comparatiu de les solucions considerades, i se'n deduirà una solució òptima pel projecte.
- Tasca 2: Familiarització i estudi de viabilitat
Es realitzarà un exercici de familiarització amb les eines relacionades amb la solució escollida (plataforma de desenvolupament, ...). També es definirà i es durà a terme una validació de la mateixa (proves de camp, definició de les restriccions d'ús, banc de proves real, ...)
- Tasca 3: Disseny i implementació de l'aplicació de monitorització en ruta
Aquesta tasca inclou la part de disseny i codificació de la solució escollida i validada en les tasques anteriors. Es refereix tant a la part client. En la fase de disseny, s'elaborarà la documentació corresponent (casos d'ús, ...)
- Tasca 4: Disseny i implementació del visor web
Aquesta tasca inclou la part de disseny i codificació de la solució escollida i validada en les tasques anteriors. Es refereix tant a la part servidor.
- Tasca 5: Finalització de la memòria i presentació.
Tot i que la memòria s'haurà anat generant en base a la informació sortint de cadascuna de les 4 primeres tasques, s'ha de contemplar una última tasca de revisió i finalització de la memòria, per a l'entrega final
- Tasca 6: Creació de la presentació virtual
Tal i com es demana en l'enunciat del Treball de Fi de carrera, com a punt final es realitzarà una presentació virtual (imatge i so), que expliqui els aspectes més rellevants del treball realitzat i quins han sigut els resultats obtinguts.

TASCA	Descripció
Descàrrega documentació TFC	Descàrrega del pla d'estudis i de l'enunciat del Treball de Fi de carrera.
Lectura documentació TFC	Lectura del pla d'estudis i de l'enunciat del Treball de Fi de carrera.
Trobada Virtual	
PAC1 Realització: Pla de treball (esborrany)	Creació del pla de treball (esborrany).
PAC1 Entrega: Pla de treball (esborrany)	
PAC1 aplicació correccions: Pla de treball (esborrany)	Aplicació de les correccions suggerides pel consultor. Creació de la versió definitiva del Pla de Treball
PAC1 Entrega: Pla de treball	
TASCA 1 - Estat de l'art de les tecnologies actuals	
Recerca informació	Recerca d'informació sobre opcions tecnològiques (dispositius receptors de GPS amb capacitat de transmetre la informació).

Quadre comparatiu	Creació de quadre comparatiu de les opcions tecnològiques trobades.
Selecció Opció tecnològica	Selecció d'opció tecnològica més adaptada a les necessitats definides en la descripció del projecte
Redacció capítol " Anàlisi d'opcions tecnològiques"	Redacció del capítol "Anàlisi d'opcions tecnològiques", definint els requisits del projecte, el quadre comparatiu resultant de la investigació i la solució final escollida.
Tasca 2 - Familiarització i estudi de viabilitat	
Instal·lació entorn de desenvolupament (IDE, SDK, ...)	Instal·lació i configuració de l'entorn de desenvolupament (IDE, SDK, ...) per a la codificació de codi font, així com qualsevol altre programari o maquinari necessari.
Aprenentatge entorn de desenvolupament	Presa de contacte amb l'entorn de desenvolupament, per adquirir els coneixements necessaris per a la codificació i/o configuració de la solució escollida.
Definició de banc de proves	Definir banc de proves reals per esbrinar l'adequació (o no) de la solució escollida.
Comprovació en condicions en cursa	Realització de proves definides en el punt anterior.
Definició de restriccions d'ús	Establir les normes i restriccions de funcionament de la solució escollida.
Redacció capítol "Estudi de viabilitat"	Redacció del capítol "Estudi de viabilitat" segons la informació trobada en les tasques anteriors.
Tasca 3 - Disseny i Implementació de l'aplicació de monitorització en ruta	
Casos d'ús	Definició dels casos d'ús de l'aplicació client.
Disseny interfície	Disseny de la interfície de l'aplicació client.
Desenvolupament	Codificació de l'aplicació client. S'utilitzarà un servidor web local per rebre les peticions HTTP del client i detectar que està enviant informació.
Creació de la documentació	Generació de la documentació relacionada amb l'aplicació client.
Redacció capítol "Aplicació", "Casos d'ús" i "Estructura client"	Redacció capítol "Aplicació", "Casos d'ús" (del client) i "Estructura client", on s'especificaran els casos d'ús així com quina és l'estructura i particularitats de l'aplicació client.
Tasca 4 - Disseny i Implementació del visor web	
(Si s'escau) Instal·lació i configuració del programari servidor	Instal·lació i configuració del programari servidor escollit, en l'entorn designat.
Instal·lació i configuració del programari generador de capes	Instal·lació i configuració del programari generador de capes, per a la construcció posterior de les capes necessàries.
Aprenentatge del programari generador de capes	Presa de contacte amb l'entorn de desenvolupament de generació de capes.
(Si s'escau) Aprenentatge del programari web	Presa de contacte amb l'entorn de desenvolupament del programari web (API Javascript, ...)
Casos d'ús	Definició dels casos d'ús de la part servidora.
Generació de capes de localització	Creació de les capes de localització necessàries pel projecte.
Publicació del servei de la capa especificada en la tasca anterior.	Publicació del servei de la capa especificada en la tasca anterior.
Definició de la interfície de l'aplicació Web	Disseny de la interfície web.

Desenvolupament de l'aplicació "Visor Web"	Desenvolupament de la interfície web.
Execució de part del banc de proves i testegi general de tot el sistema	Execució en condicions de camp de la solució final, i testegi de tot el sistema.
Correcció d'errors de codi i configuració	Correcció dels errors trobats en la subhasta anterior
Creació de la documentació	Generació de la documentació relacionada amb l'aplicació servidora.
Redacció capítol "Estructura servidor"	Redacció capítol "Estructura servidor" de la memòria, on s'especificarà l'estructura de la part servidora així com s'afegiran en el capítol "Casos d'ús", els casos d'ús de la part servidora.
Tasca 5 - Finalització de la memòria i presentació	
Recapitulació de millores a fer	Recapitulació de millores a fer, trobades durant tot el procés del projecte.
Redacció capítol "Millores"	Redacció capítol "Millores", que contindrà la informació obtinguda en la subhasta anterior.
Recapitulació "Conclusions"	Recapitulació de conclusions fruit de l'execució del projecte.
Redacció capítol "Conclusions"	Redacció capítol "Conclusions", que contindrà la informació obtinguda en la subhasta anterior.
Redacció capítol "Arquitectura"	Redacció capítol "Arquitectura, que contindrà informació sobre l'arquitectura de la solució global.
Redacció capítol "Introducció"	Redacció capítols "Introducció"
Redacció capítol "Glossari de termes"	Redacció capítol "Glossari de termes", on es construirà el glossari de tots els termes relacionats amb el projecte.
Redacció capítol "Bibliografia"	Construcció de la bibliografia.
Agraïments i maquetació	Redacció dels agraïments i maquetació de la memòria del projecte.
Creació de la presentació	Creació de la presentació del projecte.

Taula 1: Detall de tasques del TFC

3.2. Fites

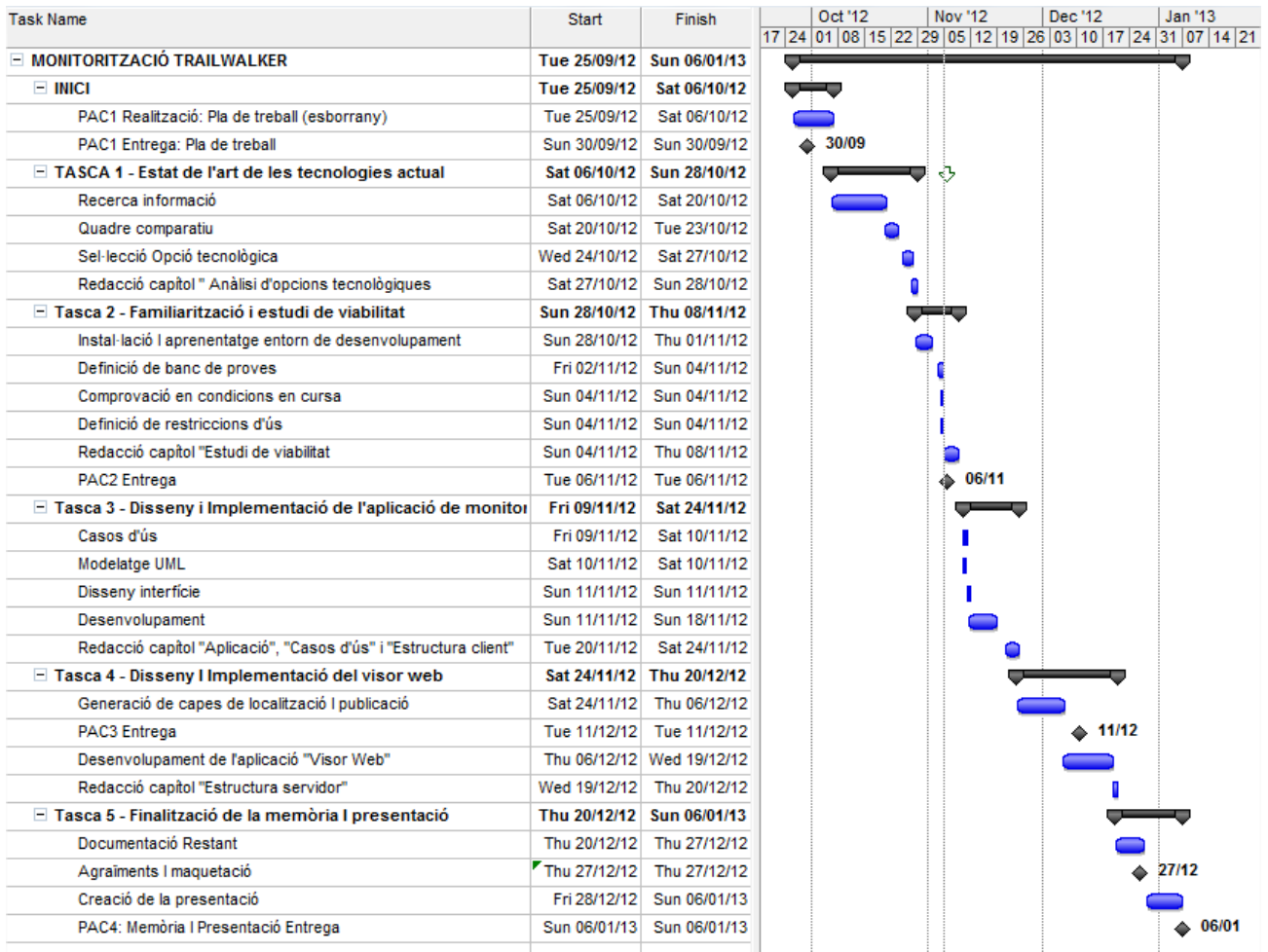
L'assignatura marca una sèrie de fites concretes que són els entrega-les: PAC1 (Pla de treball), PAC2, PAC3, PAC4 (Memòria i Presentació) i el debat virtual.

En la PAC1, es realitza l'entrega d'aquest document, el Pla de Treball. En la PAC2 i PAC3, es realitzaran entregues parcials de la feina documentada (memòria del projecte) feta fins al moment de l'entrega. La PAC4 és l'entrega definitiva del projecte.

En totes les entregues de PAC, s'ha acordat lliurar una versió esborrany uns dies abans, per tal de que el consultor faci comentaris, suggeriments i correccions, que es faran efectius de cara a l'entrega de cada PAC.

3.3. Diagrama de Gantt

La temporització de les tasques és la següent:



Il·lustració 1: Diagrama de Gantt de les activitats del TFC

4. ANÀLISI DE RISCOS

En aquest apartat es detallen els possibles esdeveniments que pugin afectar negativament a la correcta execució del projecte. Per cadascun dels esdeveniments, es proposa una solució que solucioni totalment la incidència o que mitigui l'impacte negatiu.

Desplaçament per motius laborals: És molt improbable que succeeixi aquest esdeveniment. En tot cas s'intentarà evitar aquesta situació. Si es donés, l'impacte no seria molt ja que els dies laborables són els que menys hores de feina hi ha estimades.

Avaria al punt de treball: En el cas de que hi hagués una avaria en el punt de treball que impossibilités el curs normal del projecte, es realitzarien diferents accions correctives en funció del tipus d'incidència. Si l'ordinador no funcionés, es passaria a treballar amb un segon ordinador (ja hi ha aquesta possibilitat). També es disposa de redundància en les possibilitats de connexió a internet. En el cas de que el mòbil (*Android*) quedés inutilitzable o no es pogués fer servir per qualsevol motiu, es parlaria amb el consultor per avaluar la millor solució possible.

Motius de salut: Immediatament s'intentarà analitzar l'impacte d'una incidència d'aquest tipus, comunicant-ho al consultor i avaluant les conseqüències. Si fos necessari, s'incrementarien les hores destinades a projecte per donar solució a la situació generada per una eventual malaltia.

5. MATERIAL

5.1. Introducció

Per realitzar la documentació del treball de fi de Carrera s'utilitzaran els següents materials:

- PROGRAMARI
 - o Microsoft Office (Word, Excel, Power Point, Project, Visio)
- MAQUINARI
 - o Ordinador amb S.O. Microsoft Windows 7 Home Premium 64 bits

5.2. Desenvolupament a mida

En cas de que el projecte impliqui desenvolupament a mida, s'utilitzaran els següents materials:

- PROGRAMARI
 - o Plataforma IDE Eclipse Juno (Eclipse SDK Versió: 4.2.0)
 - o Plugin ADT per Eclipse
 - o *Android* SDK
 - o Programari ESRSI ArcGIS:
 - ArcGIS for Desktop
 - ArcGIS Server
- MAQUINARI
 - o Ordinador amb S.O. Microsoft Windows 7 Home Premium 64 bits
 - o *Smartphone Samsung SIII* – S.O. *Android* 4.0

5.2.1. ESRI ArcGIS

ArcGIS és una suite de productes de ESRI, que permet crear, analitzar, emmagatzemar i difondre dades, models, mapes i globus en 3D, posant-los a disposició de tots els usuaris segons les necessitats de cada organització. En l'àmbit d'aquest projecte, ESRI ha posat a disposició de la UOC llicències dels productes ArcGIS for Desktop i ArcGIS for Server.

ArcGIS for Desktop: Producte d'escriptori que permet crear mapes, informes, edició i anàlisi de dades GIS o *Geographic Information System* (Sistemes d'informació geogràfica), entre altres funcionalitats relacionades amb GIS. En el nostre cas, utilitzaríem aquesta aplicació per a generar els objectes (mapes, ...) que posteriorment es publicarien en una instància de ArcGIS for Server, per a la seva explotació.

ArcGIS for Server: Producte que permet interactuar (transformar, visualitzar...) amb els objectes (mapes, imatges...) creats a través de ArcGIS for Desktop, a través de serveis web. Al tractar-se de serveis web, són accessibles des d'arreu de la xarxa (si es publiquen en un servidor visible a Internet), i poden donar suport a un ampli ventall d'aplicacions de creació cartogràfica. En el nostre cas, serà el suport a través del qual podríem gestionar els mapes i objectes relacionats, creats amb l'ArcGIS for Desktop, per a guardar la informació de localització dels equips i mostrar-la amb un format adequat.

6. ANÀLISI D'OPCIONS TECNOLÒGIQUES

6.1. Introducció

L'objectiu del projecte és el de proposar i justificar una solució tecnològica que s'ajusti als requisits descrits per Intermón Oxfam per la seva cursa solidària *Trailwalker*. Es tracta d'una cursa a peu de 100 quilòmetres de distància total, a superar en un màxim de 32 hores. Intermón Oxfam demana que es realitzi una monitorització de tots els equips i que aquesta informació sigui fàcilment accessible i pública des del punt de vista de l'organització, participants i espectadors.

6.2. Requisits

Els requisits venen marcats per la definició del problema a solucionar per Intermón Oxfam (fer seguiment dels equips) i també per les condicions pròpies de l'activitat que s'hi duu a terme (una cursa). Partint d'aquestes premisses, existeix un conjunt de requisits que són els indicats a continuació.

6.2.1. Autonomia

Les condicions de la cursa marquen un temps màxim de 32 hores per completar el circuit de 100 Km. Per tant la solució ha de contemplar una autonomia de fins a 32 hores. Malgrat que la solució ideal ha de ser capaç d'estar en funcionament durant les 32 hores de cursa, cal tenir present que cada equip, format per quatre marxadors, compta amb l'ajut d'un equip de suport de 2 persones que els segueixen durant la cursa. S'ha de considerar llavors que, en cas de que hi hagi una solució viable en els altres aspectes menys en aquest, es pot arribar a considerar l'ajut de l'equip de suport per la provisió de fonts alternatives d'energia, per exemple bateries, o fins i tot per realitzar canvis de dispositius.

6.2.2. Marxadors

Es tracta d'una cursa a peu, d'una distància considerable si es té en compte el temps màxim en què s'ha de realitzar. A partir d'això se n'extreu que la solució ha de ser respectuosa amb el marxador. Per tant, no es considera vàlida cap solució que:

- Dificulti la mobilitat del marxador.
- Afegeixi pes excessiu en l'equipament de cursa dels participants. S'ha de tenir en compte que els participants ja porten allò que necessiten per la cursa i allò que els suggereix i obliga l'organització¹.

6.2.3. Informació

Aquest punt és el motiu principal del projecte, ja que el que l'organització busca és tenir informació sobre la situació dels equips durant la cursa. Com que la informació haurà de plasmar la situació de la cursa, es té en compte que s'ha d'actualitzar. Es fa el supòsit que una freqüència d'actualització de 10 minuts és suficient ja que es tracta d'una cursa que té una durada amb un ordre de magnitud d'hores².

La informació a proporcionar tindrà les següents característiques:

- Facilitarà la localització dels equips. Es pren el supòsit que aquesta informació es presentarà en forma de punts en un mapa per facilitar la comprensió.
- Mostrarà la classificació dels equips. La solució haurà de mostrar de forma comprensible quina és la classificació del conjunt d'equips en tot moment.
- La localització d'equips i classificació serà accessible per l'organització, marxadors i públic en general. Es fa el supòsit que la informació es mostrarà incrustada en la web de *Trailwalker* amb el que automàticament estarà disponible per tot tipus de públic. Segons les normes de la cursa, i més concretament respecte al material obligatori pels marxadors, cal dur un mòbil. Si es té en compte que avui en dia és freqüent disposar de *Smartphone* amb connexió a internet, es pot arribar a

¹ <http://trailwalker.intermonoxfam.org/ca/equips/entrenament-preparacio#4>

² Es poden consultar els resultats per temps de les edicions anteriors a <http://trailwalker.intermonoxfam.org/ca/que-es-tw/edicions-anteriors>

considerar com a solució viable la de mostrar la informació en una pàgina web, i per tant accessible per tothom.

6.2.4. Despeses Associades

Un dels punts més importants de la solució és que sigui viable econòmicament. Tenint en compte que estem parlant d'una cursa solidària en la que el que s'intenta és recaptar diners, s'ha de prioritzar al màxim que la solució no representi una despesa afegida a l'organització ni als marxadors.

6.2.5. Complexitat

La solució tecnològica ha de ser de fàcil implantació de forma que tant l'organització com els equips no hagin de destinar gaire esforç en la implantació i ús durant la cursa de la mateixa. S'ha de prioritzar tota aquella solució tecnològica que sigui com més autònoma possible en l'aspecte de funcionament.

6.2.6. Resistència

Qualsevol solució que impliqui que els marxants incorporin en el seu equipament de ruta un element tecnològic, aquest ha de tenir certa resistència i preparat per resistir una cursa d'aquesta intensitat.

6.2.7. Cobertura de senyal

S'ha de tenir present que cap solució tecnològica que impliqui l'enviament o recepció de senyal del tipus que sigui (GPS, 2G/3G, ...) ha d'estar preparada per períodes en què el senyal no estigui disponible. Per exemple, pot ser que es passin per zones on la cobertura de telefonia mòbil no estigui disponible.

6.3. Quadre Comparatiu

6.3.1. Recerca de solucions

Per realitzar un quadre comparatiu de diferents solucions, prèviament s'ha realitzat una recerca de solucions viables. S'han fet els següents supòsits:

- La solució global haurà de constar a grans trets de 3 parts:
 - Client: utilitzarà la tecnologia GPS per capturar la localització i l'enviarà a la part servidora:
 - Servidor: rebrà les posicions GPS dels clients i en farà el processat adequat.
 - Capa d'Informació: mostrarà en forma de mapes la informació processada pel Servidor.
- Es parteix de la base que, en cas de necessitat, es disposa del producte ArcGIS Server per a la part servidora.

La recerca s'ha realitzat sota la premissa de trobar una solució tecnològica (solució disponible en el mercat), o un conjunt d'aquestes que donin resposta als requisits plantejats.

En resum, el resultat de la recerca ha trobat 4 grans grups de tipus de solucions possibles:

- Solucions Comercials de seguiment: No s'han trobat solucions comercials que estiguin pensades per a fer *tracking* dels participants de curses a peu. Malgrat això hi ha solucions que s'acosten molt a aquest propòsit com són les solucions de mercat destinades a seguiment de persones o a control de flotes de vehicles. Un parell de les més rellevants que s'ha trobat són:
 - “Zoomback”:
 - Solució de localitzadors GPS: <http://www.zoombak.com/products/universal/>
 - “Global AVL”:<http://www.globalavl.eu/>
 - Solució per *Smartphones*: <http://www.globalavl.eu/en/compatible-gps-trackers/compatible-gps-trackers/gps-Smartphone-tracking.html>
 - Solució amb *trackers* de GPS: <http://www.globalavl.eu/en/compatible-gps-trackers/compatible-gps-trackers/gps-personal-trackers.html>

- Solucions destinades a *tracking* d'activitats a l'aire lliure: Hi ha varies solucions disponibles a internet que permeten descarregar-te un programari que cal instal·lar en un *Smartphone*, i que fan *tracking* de recorreguts realitzats. Utilitzen el GPS del telèfon i envien les dades al *cloud* a través de la connexió de dades del telèfon. De les més destacables hi ha:
 - “My Tracks” de Google: <http://www.google.com/mobile/mytracks>
 - “Map My Tracks”: <http://www.mapmytracks.com>
 - “Runtastic”: <http://www.runtastic.com>
 - “Endomondo”: <http://www.endomondo.com/>
- Solucions a mida: Desenvolupament del client per diferents plataformes. Es llisten a continuació.
 - *Smartphone* (*Android*, iOS, RIM, Windows Mobile, ...)
 - Plataforma de Hardware (*Arduino*): <http://www.Arduino.cc/>

A priori es determina que a les solucions a mida cal realitzar un desenvolupament en totes 3 parts (Client, Servidor, Capa d'Informació). La resta de grups de solucions donen només solució a la part Client i Servidor. Per tant en aquests casos és necessari realitzar un desenvolupament a mida de la Capa d'Informació, i una integració entre aquesta i el Servidor.

Per totes les solucions trobades per *Smartphone*, es contempla la possibilitat d'utilitzar un receptor GPS extern amb connexió *Bluetooth* per aquells casos en què l' *Smartphone* no tingui sensor GPS integrat o per possibles millores en l'autonomia de la solució.

6.3.2. No idoneïtat de les solucions trobades

Abans de realitzar la comparativa entre solucions trobades, n'hi ha que queden excloses perquè no compleixen amb els requisits mencionats anteriorment.

- Solucions Comercials de seguiment: En aquest cas, la marca “Global AVL” té experiència en el *tracking* de flotes de vehicles i de persones. Ofereix un ampli ventall de serveis i ofereix diferents tipus de solucions tecnològiques que s'adaptarien en part als requisits. Tenen programari client a instal·lar en *Smartphones* o dispositius específics per a fer *tracking*. Aquesta informació és centralitzada en el *cloud* (els seus sistemes), i la informació es mostra en format *SaaS* (Software as a Service), amb la possibilitat de mostrar-ho en web. També ofereixen una API (Application Programming Interface o interfície de programació d'aplicacions) d'integració que permetria recuperar la informació generada de la part client a la servidora, i mostrar-la en una capa d'informació (ArcGIS for Server en aquest cas). Al tractar-se d'una solució comercial, es presumeix uns costos excessivament elevats que per tant fan que la solució no sigui viable.
- Solucions destinades a *tracking* d'activitats a l'aire lliure: Totes aquestes solucions tenen molt ben resolta la part Client – Servidor. Les dades es poden arribar a emmagatzemar en el *cloud*, però aquesta operació és manual en la majoria de les solucions llistades i en tot cas es realitza al final de l'operació de *tracking*, és a dir al final de la cursa. Només “My Tracks” i “Map My Tracks” ofereixen una API per realitzar una integració amb un tercer sistema (ArcGIS for Server en el nostre cas, per exemple). Tot i tenir disponible la API, la gestió és individualitzada per compta (són aplicacions pensades per gestionar les caminades, curses, passejades, etc... d'un mateix) que fa impossible una integració per un eventual grup de 400 equips. Totes elles tenen una versió gratuïta (aparentment suficient pel nostre fi). Degut a això, no tenim cap acord de nivell de servei amb el proveïdor, que, per tant no ens assegura la disponibilitat i el suport en cas d'incidència amb el sistema.

6.3.3. Comparativa de les solucions finalistes

La comparació de les finalistes queda entre els dos tipus de solucions a mida, ja que permeten una configuració com la definida (Client – Servidor – Capa d'Informació), i, a priori, sense restriccions importants. A més, es realitza exclusivament de la part client, que és la part variable entre solucions.

Com s'ha comentat anteriorment, s'ha afegit en la combinació de solucions el fet de disposar d'un receptor GPS extern amb connexió *Bluetooth*, pel cas de la solució d' *Smartphone*, ja que es presumeix que tingui un impacte en diferents aspectes relatius amb els requisits.

Quant a l'arquitectura de la part Servidor i la part de Capa d'Informació a grans trets constarà de:

- Servidor: ArcGIS for Server com a programari de gestió d'informació GIS (com s'ha comentat anteriorment, és compta amb aquest recurs en cas de no trobar una solució comercial adient).
- Capa d'Informació: ArcGIS for Server + Servidor pàgines web (Internet Information Services, Apache, ...)

S'ha realitzat un quadre comparatiu tenint en compte una sèrie de variables (requisits definits anteriorment). Concretament són:

- **Autonomia:** capacitat de la solució per funcionar de forma ininterrompuda. Valor mesurable.
- **Pes:** característica relacionada amb les condicions de cursa dels marxadors. Com menor sigui, millor és la solució. Valor mesurable.
- **Mobilitat:** característica relacionada amb les condicions de cursa dels marxadors. Indica el grau de molèstia en la mobilitat dels marxadors. Valor no mesurable.
- **Informació:** grau d'adequació amb els requisits informatius. Valor no mesurable.
- **Cost Desplegament:** aquells costos relacionats amb la posta amb marxa de la solució. Valor mesurable.
- **Cost Recurrent:** costos recurrents relacionats amb la solució. Valor mesurable.
- **Complexitat:** dificultat en l'adopció de la solució en termes generals. Valor no mesurable.
- **Resistència:** capacitat de la solució tecnològica per no malmetre's durant la cursa. Valor no mesurable.
- **Cobertura Senyal:** capacitat de la solució per no interrompre el funcionament durant els períodes de no cobertura de senyal (GPS, 2G, 3G, ...)

D'aquestes variables, se'n realitza una puntuació (1-5) per cada solució tecnològica. Els valors més petits indiquen "menys idoni" mentre que els valors més grans indiquen "més idoni", respecte a donar solució al problema plantejat. S'ha valorat el fet de donar diferents ponderacions a les variables, en el cas que es considerés que una variable és més determinant que altres. En aquest cas s'ha considerat que totes les variables són igualment determinants.

No s'han trobat referències bibliogràfiques formals a partir de les quals establir les puntuacions. En canvi sí que s'han trobat altre tipus de referències com són entrades de blogs i fòrums a internet que tracten els temes de dispositius GPS (dispositius dedicats, GPS en *Smartphones...*)³.

Les referències trobades fan esment sobretot a comparatives entre l'ús dels *Smartphones* com a GPS respecte a solucions comercials de navegació per cotxes i dispositius GPS d'ubicació. *Arduino*, al ser una plataforma de desenvolupament i prototipatge, no és present en les referències, ja que el seu ús no està centrat en el posicionament.

Per últim cal remarcar que aquesta avaluació global és teòrica i caldria una validació de camp.

³ veure Annex: Referències per la valoració de solucions

Solució	Autonomia	Pes	Mobilitat	Informació	Cost Desplegament	Cost Recurrent	Complexitat	Resistència	Cobertura de Senyal	TOTAL
<i>Arduino</i>	4	4	5	5	1	2	1	3	5	30
<i>Smartphone (GPS Integrat)</i>	2	4	5	5	5	4	5	4	5	39
<i>Smartphone+GPS Extern</i>	3	4	5	5	3	4	4	5	5	38

Taula 2: Comparativa de les solucions finalistes

6.4. Solució Final

Hi ha una sèrie de variables que tenen una valoració igual entre les 3 solucions: Pes, Mobilitat, Informació i Cobertura de Senyal. Respecte al pes i mobilitat, un dispositiu *Arduino* pesa molt poc i té unes dimensions reduïdes, pel que no afecta a les condicions de cursa del marxador. A banda de que els mòbils (o *Smartphones*) tenen dimensions reduïdes i, generalment, pesen relativament poc (100 – 150gr), cal recordar que els marxadors estan obligats a dur-ne, com a mínim un durant la cursa. Per tant, no es pot considerar que afectin al marxador ja que sí o sí serà un equipament de cursa. Respecte a l'addició d'un element com és el GPS extern (portant *Smartphone*), tampoc es considera un handicap a causa de que les dimensions i el pes són menyspreables (70gr i 7x4x2cm aproximadament)⁴.

La informació és exactament la mateixa per totes 3 solucions, ja que són capaces de transmetre el mateix tipus d'informació cap a la part servidora i de capa d'informació, que és comuna. Totes 3 solucions també estan preparades (o es poden preparar) per a suportar períodes de no cobertura.

Un cop aclarits els punts no vinculants de les solucions (perquè són iguals entre elles), es passa a descriure els punts forts i dèbils de les solucions

6.4.1. Punts forts de la solució d' *Arduino*

Bàsicament la solució d' *Arduino* suposadament té un gran punt fort que és l'autonomia. Cal tenir present que l' *Arduino* és una plataforma de maquinari que en aquest cas estarà pensada i dissenyada per realitzar la tasca de capturar les posicions i enviar-les per la xarxa (GPRS). Per tant el seu rendiment estarà molt optimitzat. A més, les bateries que pot utilitzar són piles convencionals del tipus AAA, que són de mida reduïda i per tant facilita el fet de que els marxadors en duguin de reserva.

6.4.2. Punts dèbils de la solució d' *Arduino*

El tret diferencial de la plataforma *Arduino* amb la solució a mida en un *Smartphone*, és que en l' *Arduino* s'ha d'adquirir un maquinari específic. Aquest maquinari només s'utilitzarà en aquesta cursa, que fa que l'adquisició tingui un retorn de la inversió molt menor (és més atractiu haver d'adquirir un maquinari que puc reaprofitar per altres activitats que no pas un que només té una funció específica com l' *Arduino*). Es calcula un cost d'adquisició de l' *Arduino* entre 150 i 200€ aproximadament⁵.

Cal tenir en compte la possibilitat d'haver de tenir costos recurrents de línies addicionals contractades, ja que el dispositiu necessitarà d'una xarxa 2G/3G per a realitzar la transmissió d'informació. Es descarta l'ús de targetes SIM (targetes de telefonia mòbil) personals en l' *Arduino*, ja que això implicaria manipulació del dispositiu per part dels marxadors, així com un servei de suport per part de l'organització, cosa que en faria més complexa la seva gestió, sobretot en un entorn de molts equips (400).

Quan a la complexitat, caldria d'un suport específic i especialitzat per part de l'organització. Malgrat és un dispositiu que no cal intervenció externa, la seva instal·lació i posta en marxa pot esdevenir complexa, ja que la manipulació només es podria realitzar per part de personal experimentat.

Respecte a la resistència, com ja hem comentat l' *Arduino* és una plataforma de desenvolupament i prototipatge, i no està pensada per aquest ús (curses), pel que se li suposa una certa resistència menys que als altres 2 dispositius. En canvi, sí que hi ha moltes referències bibliogràfiques⁶ respecte a la robustesa de dispositius GPS preparats comercialment per aquesta finalitat (activitats a l'aire lliure).

⁴ veure Annex: Referències de preus i models de GPS Externs per *Bluetooth*

⁵ Veure Annex. Referències de preus d' *Arduino*

⁶ veure Annex: Referències de preus i models de GPS Externs per *Bluetooth*

6.4.3. Punts forts de la solució *Smartphone*

Es considera al cost de desplegament d'aquesta solució com a gairebé nul. Cal pensar que el que es demanarà als marxadors és instal·lar-se una aplicació (desenvolupada a mida) i disponible des d'internet (probablement des dels catàlegs d'aplicacions de les diferents plataformes: *Android*, *iOS*, *RIM*, ...). L'aplicació no tindria cost i la instal·lació es podria dur a terme pels propis usuaris de l'*Smartphone*, amb el que l'organització no hauria de recórrer en despeses addicionals en aquest aspecte (potser només dotar d'un punt d'ajut en l'inici de la cursa, o quelcom semblant). Avui en dia és molt comú disposar d'un *Smartphone* amb GPS integrat per ús personal, amb el que tampoc s'estarien afegint despeses extres en aquest aspecte.

El GPS Extern implica que algú (marxadors o organització) l'adquirís i el configurés a l'*Smartphone*. A més, aquesta solució implica el mateix que la solució d'*Smartphone* (descàrrega de l'aplicació,...). Per tant el cost d'implantació és el mateix que el cost d'implantació que la solució *Smartphone* (GPS Integrat) més el cost d'adquisició del GPS Extern, que hauria de ser assumit per alguna de les parts (marxants o organització). Malgrat que es considera que degut al perfil d'alguns dels participants (marxadors habituals) és probable que aquests comptin amb un dispositiu GPS extern, aquest fet no serà molt extens. El cost d'adquisició d'un GPS extern està entre 20 i 150€ aproximadament⁷ en funció sobretot de les funcionalitats d'aquest.

Cal tenir en compte que els costos recurrents seran també menyspreables ja que habitualment els usuaris d'*Smartphone* disposen d'una línia de dades contractada.

La senzillesa (no complexitat) també és un punt fort de la solució *Smartphone*, ja que com hem comentat, només caldria instal·lar-se una aplicació. Seguidament vindria la solució del *Smartphone*+GPS Extern ja que caldria realitzar l'operació anterior més la configuració del dispositiu, que no es presumeix gaire complexa. Per totes dues solucions caldria tenir en compte un servei de suport per part de l'organització. La quantitat d'incidències i problemes en la configuració es pressuposa baixa ja que és un sistema força senzill.

6.4.4. Punts dèbils de la solució *Smartphone*

Un dels punts dèbils més importants de la solució *Smartphone* és l'autonomia. En condicions normals (és a dir, fent-ne un ús quotidià) aquests dispositius acostumen a tenir rendiments baixos en aquest aspecte. Es pressuposa que l'autonomia es reduiria considerablement si es fa un ús del GPS. Es considera que el fet de no utilitzar el GPS integrat sinó utilitzar un canal de dades *Bluetooth* pot no penalitzar tant l'autonomia. Segons les referències per la valoració de solucions⁸ es comprova que si s'utilitza un *Smartphone* com a *tracker* GPS, aquest té una autonomia al voltant de les 4 o 5 hores. Tot i que el valor és realment baix, cal tenir en compte que degut a les condicions de cursa, no cal realitzar un *tracking* tant exhaustiu, i per tant espaiar les captures de posició en el temps (fins a 10 minuts), amb el que l'autonomia no se'n veuria tant afectada.

A més, hem de tenir en compte que hi ha referències que indiquen que la captació de senyal de GPS penalitza més que la transmissió de dades per *Bluetooth*. En aquesta direcció, hem de suposar que l'ús d'un GPS extern amb connexió *Bluetooth* amb l'*Smartphone* perjudicarà menys a l'autonomia que no pas si s'utilitza l'*Smartphone* amb el seu receptor GPS integrat.

Un cop tinguts en compte tots els requisits amb les solucions tecnològiques triades, la solució *Smartphone* és la solució recomanada a priori per cobrir les necessitats exposades (veure puntuació final en la Taula 2). Cal fixar-se que *Arduino* té com a únic punt fort l'autonomia, que és justament el punt dèbil dels *Smartphones*. Malgrat que l'autonomia es podria considerar com a el gran handicap a superar i per tant posaria *Arduino* en una bona posició, no té sentit decantar-se per *Arduino* ja que com a solució té molts punts dèbils. En canvi es preveu que aquesta gran mancança dels *Smartphones* es pot suplir bàsicament amb les següents propostes:

⁷ veure Annex: Referències de preus i models de GPS Externs per *Bluetooth*

⁸ veure Annex: Referències per la valoració de solucions

- Ús d'un GPS extern amb connexió *Bluetooth*
- Desenvolupar un programari de localització per la cursa que optimitzi els recursos de l' *Smartphone*
- Ús de tecnologies alternatives al GPS per a realitzar posicionament amb menys penalització pel que fa al consum de bateria
- Recolzament per part de l'equip de suport, que els pot proveir serveis o recanvis durant la cursa, i que per tant, poden arribar a suplir certes mancances definides com a requisits (autonomia, per exemple).
- Ús de bateries de recanvi, o de bateries d'un sol ús⁹ que tenen un valor de compra relativament baix (5-10€) i unes dimensions i pes reduïts, amb el que poden arribar a ser un gran complement. Les bateries d'un sol ús permeten connectar-les als *Smartphones* habitualment a través d'un cable (no reemplaquen les bateries existents).

Malgrat que de forma teòrica s'ha determinat que la solució *Smartphone* és la solució més adient, cal indicar que això és una primera aproximació i que caldrà realitzar un estudi de viabilitat per comprovar-ne la validesa.

⁹ Veure Annex: Referències de bateries d'un sol ús

7. SISTEMES *ANDROID* i ArcGIS

7.1. Introducció

En aquest apartat es pretén donar una breu introducció als conceptes i sistemes involucrats en la solució final proposada.

Els *Smartphones* o telèfons intel·ligents, són telèfons mòbils amb més capacitat de processat i connectivitat que els telèfons mòbils convencionals. Estan construïts sobre sistemes operatius específics que els dona alta capacitat d'interacció amb l'usuari, i més facilitats en la creació d'aplicacions de cara als programadors informàtics.

Tornant al concepte de més capacitat de processat i connectivitat, cal destacar que aquests factors són els que han sigut determinants en l'èxit d'aquests dispositius. Disposen de connexió a Internet d'alta velocitat (3G), sovint de dispositiu GPS, dispositiu *Bluetooth*... És per això que no és d'estranyar que aquests dispositius hagin adquirit popularitat gràcies a aplicacions que han explotat aquestes capacitats com són el correu electrònic, xarxes socials...

De cara a aquest projecte, els *Smartphones* esdevenen un recurs molt valuós a tenir en compte ja que són dispositius molt comuns i que ofereixen un ventall molt ampli de possibilitats pel que fa a la creació d'aplicacions que formin part d'una solució tecnològica més o menys complexa des del punt de vista d'arquitectura.

Els GIS o *Geographic Information System* (Sistemes d'informació geogràfica) són sistemes dissenyats per gestionar (capturar, emmagatzemar, manipular, analitzar, gestionar i mostrar) tot tipus de dades geogràfiques. Els seus orígens no estan basats en tecnologia, pel que el seu concepte és més ampli que això. Posteriorment s'ha realitzat una integració natural dels GIS amb sistemes d'informació, que fa que de forma comuna es concebin els GIS amb sistemes tecnològics d'informació geogràfica.

La relació amb aquest projecte radica en la capacitat que ens donen aquests sistemes sobre la captura posicionament (localització en la superfície terrestre) i la possibilitat de mostrar aquesta informació sobre una interfície fàcilment comprensible per l'usuari final, com és la representació de punts sobre un mapa.

Quant a la representació gràfica dels GIS, actualment hi ha sistemes que són capaços de mostrar la informació a través de pàgines web i serveis web, que incrementa la seva accessibilitat i la capacitat d'integració amb sistemes tercers.

7.2. Android

Android és un sistema operatiu enfocat a ser utilitzat en dispositius mòbils com són *Smartphones* i tauletes (*tablets*). Està basat en el sistema operatiu Linux Està desenvolupat per la Open Handset Alliance¹⁰, que és una aliança comercial de diverses companyies que es dedica a desenvolupar estàndards oberts per a dispositius mòbils. Un dels membres més importants és Google.

Des del punt de vista tècnic i concretament respecte a aquest projecte cal destacar les següents característiques d' *Android*:

- Com s'ha comentat, és un sistema operatiu basat en Linux, que és un sistema operatiu caracteritzat per tenir un nucli lliure i robust (en les seves distribucions estables).
- Suporta moltes tecnologies de connectivitat: *Bluetooth*, I-FI (o xarxes sense fils), HSDPA, HSPA+, GSM/EDGE... (HSDPA)

¹⁰ <http://www.openhandsetalliance.com/>

- Suporta un ampli ventall de maquinari addicional: GPS, càmera de fotos, pantalla tàtil, acceleròmetres, giroscopis, magnetòmetres...
- És un sistema multitasca
- Permet desenvolupar aplicacions en codi Java i C/C++, i inclou un ampli ventall de biblioteques per accedir a recursos de sistema

7.2.1. Desenvolupament d'aplicacions per *Android* – *Android SDK*

La plataforma *Android* (<http://www.Android.com>) té una secció dedicada al desenvolupament d'aplicacions (<http://developer.Android.com>). Aquesta secció aporta des de principis i guies pel que fa al disseny d'interfícies d'usuari, passant per la documentació tècnica i finalment com publicar i promoure l'aplicació (a través de Google Play¹¹). A tall d'exemple en el següent enllaç: <http://developer.Android.com/about/start.html>, s'ofereix una guia completa per desenvolupar una aplicació.

Passada la fase de disseny, se'ns suggereix descarregar i instal·lar l' *Android SDK* (Software Development Kit). L' *Android SDK* ens proveeix d'una sèrie de llibreries API necessàries per generar, testejar i depurar aplicacions.

Quant a l'entorn de desenvolupament, a la secció de “Developer” d' *Android* se'ns recomana utilitzar l'ADT o *Android Developer Tools*. L'ADT és un paquet que ens podem descarregar des de la mateixa web i ens proveeix d'allò necessari per codificar i compilar una aplicació per *Android*, com un entorn de desenvolupament integrat (Eclipse), eines de l'SDK, l'última versió de plataforma d' *Android*, emulador de *Smartphone*... Per a més referències es pot consultar: <http://developer.Android.com/sdk/exploring.html>.

En el SDK, el llenguatge de programació per a noves aplicacions d' *Android* és el Java. És un llenguatge de programació d'alt nivell i orientat a objectes. Cal comentar que està molt estès i és multi plataforma. *Android* també permet programar amb codi natiu, mitjançant els llenguatges C/C++, però per això cal utilitzar un altre paquet anomenat *NDK*. Es recomana utilitzar C/C++ només en els casos en què sigui necessari (per temes de rendiment), i sobretot de cara a generació de llibreries a ser utilitzades des de Java.

Android defineix un esquema per les seves aplicacions:

- Components d'aplicació: Són les diferents parts que poden compondre una aplicació. Aquestes es separen en:
 - o *Activity*: és una pantalla o interfície d'usuari
 - o *Service*: component que s'executa en segon pla pensat per a operacions de llarga duració (de forma que no afectin a les *Activity*).
 - o *ContentProvider*: són gestors de contingut o dades
 - o *BroadCastReceiver*: components pensats per rebre o escoltar de notificacions de sistema. S'executen en un segon pla.
- Arxiu “Manifest”: en aquest arxiu s'hi descriuen els components que el sistema operatiu d' *Android* pot executar d'una aplicació en concret. S'hi ha de declarar doncs, tots els components que un programador vulgui executar en el cicle de vida d'una aplicació (les *Activity*, *Service*, ...). També s'hi defineixen permisos que necessita l'aplicació (accés a Internet, accés al servei de localització...), versions mínimes d'API que necessita l'aplicació, característiques de programari requerits o utilitzats (càmera, serveis de *Bluetooth*...) i llibreries API que l'aplicació necessita enllaçar.

¹¹ *Google Play* és un catàleg d'aplicacions disponibles per *Android* des d'on es poden descarregar i instal·lar.

La documentació de l'SDK és molt més extensa i concreta. Per a més informació es pot consultar la pàgina web: <http://developer.Android.com/sdk/index.html>

7.2.2. Posicionament i sistemes *Android*

A continuació es fa una breu introducció del posicionament a través de dispositius *Android*.

En la secció Developer *Android* hi ha una sèrie de referències a diferents usos finals que tindran les aplicacions. El cas del posicionament no és una excepció. En la direcció web <http://developer.Android.com/guide/topics/location/strategies.html>, es defineixen les diferents estratègies a seguir per determinar la localització del dispositiu des de les aplicacions desenvolupades amb l'SDK.

Per a la determinació de la posició del dispositiu o localització, *Android* ens ofereix els següents recursos o proveïdors d'ubicació:

- Posicionament per GPS
- Posicionament per “Servei d'ubicació de Google” (xarxes mòbils i Wi-Fi)

L'accés a aquests recursos es realitza a través d'una classe de sistema anomenada *LocationManager*¹². El *LocationManager* s'utilitza amb una *interface* anomenada *LocationListener*¹³. De forma resumida, a través del *LocationManager*, es realitza una petició d'activació de servei (per exemple rebre la localització), i hi subscriu el *LocationListener*. En la implementació del *LocationListener* es defineix què fer en el cas de l'aparició de diferents *events*. Per cadascun d'aquests *events*, es crida a una de les funcions implementades. Per exemple la funció *onLocationChanged* de *LocationListener*, és cridada cada cop que el sistema detecta que la posició ha canviat.

El *LocationManager* permet realitzar 3 tipus d'operacions de localització:

- Una petició activa: el sistema utilitza el recurs triat demanant-li que li proveeixi les dades de localització. Funció “requestSingleUpdate”¹⁴ del *LocationManager*.
- Una programació de peticions actives: el sistema de forma programada proveeix les dades de localització amb un temps mínim definit i/o distància mínima a partir de l'anterior localització. Funció “requestLocationUpdates” del *LocationManager*.
- Una programació d'escolta passiva: el sistema no fa peticions actives als recursos, sinó que resta a l'espera de localització demanada per altres aplicacions. Aquesta escolta passiva es realitza passant la constant “LocationManager.PASSIVE_PROVIDER”, en la funció “requestLocationUpdates” del *LocationManager*.

La informació de posicionament es rep de forma asíncrona a través de les funcions implementades del *LocationListener*. Com complement, la classe *LocationManager* ens ofereix la funció “getLastKnownLocation” que torna la última posició trobada pel proveïdor d'ubicació seleccionat. Mitjançant aquesta funció, podem saber de forma immediata aquesta informació, sempre tenint present que pot ser que estigui des actualitzada en el moment de la consulta.

A través del *LocationManager*, a banda de poder indicar a l'aplicació quin recurs utilitzar, dota també d'eines per triar el millor recurs a utilitzar a partir d'una sèrie de criteris definits, així com la de triar la

¹² <http://developer.Android.com/reference/Android/location/LocationManager.html>

¹³ <http://developer.Android.com/reference/Android/location/LocationListener.html>

¹⁴ Disponible a partir del nivell d'API 9 (versió Android 2.3)

posició més exacte (amb menys marge d'error) o vigent (que no sigui massa antiga) d'un grup de posicions trobades.

S'ha de tenir en compte que els dispositius són telèfons, i per tant, són aparells pensats per a interactuar amb l'usuari, i que aquest últim és el que té control sobre els recursos (quins estan activats i quins no). Per exemple, un usuari pot desactivar el GPS del telèfon que fa que no estigui disponible per una eventual aplicació que en faci ús. *Android* permet consultar l'estat dels serveis de localització en tot moment (activat/desactivat). Es pot consultar de les següents formes:

- A través del *LocationManager*: Realitzar una consulta activa de l'estat en un moment determinat. Aquesta consulta es realitza a través de la funció "isProviderEnabled" passant-li com a paràmetre el nom (tipus *String*) del proveïdor de servei. Els proveïdors estan representats com a constants en la classe *LocationManager* i poden ser:
 - o ***LocationManager.GPS_PROVIDER***: en el cas que es vulgui comprovar l'estat del proveïdor de servei de localització per GPS.
 - o ***LocationManager.NETWORK_PROVIDER***: en el cas que es vulgui comprovar l'estat del proveïdor de servei de localització per "Servei d'ubicació de Google" (xarxes mòbils i I-FI).
- A través del *LocationListener*: De forma passiva, i a través de la implementació de les funcions:
 - o *onProviderDisabled*: en el cas que es desactivi un proveïdor de serveis de localització
 - o *onProviderEnabled*: en el cas que s'activi un proveïdor de serveis de localització

Finalment, per poder fer servir el posicionament en *Android*, cal demanar el permís d'ús de serveis de localització al sistema. Com s'ha comentat en el punt anterior, això es realitza introduint una entrada en l'arxiu "Manifest" de l'aplicació. Si això no es fa, es rep una excepció en temps d'execució indicant-nos que no es té permís per a accedir als recursos de localització.

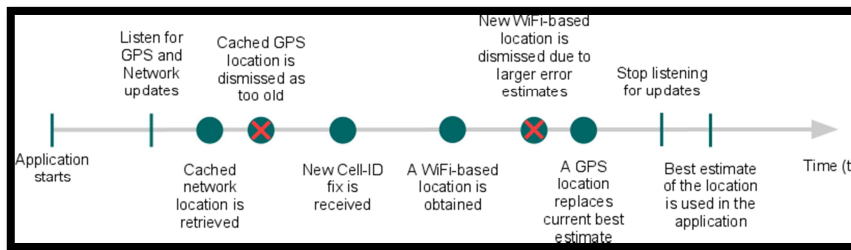
7.2.3. Posicionament i rendiment de bateria en sistemes *Android*

En la documentació de Developer *Android* s'hi indica una sèrie de recomanacions per obtenir el millor rendiment i per tant per minimitzar el consum de bateria, que com ja s'ha comentat, és un dels desavantatges dels *Smartphones*¹⁵.

(Android Developer - Location Strategies) Inicia explicant el flux típic per obtenir la localització del dispositiu, i a partir d'aquesta informació explica les recomanacions. El flux típic segueix les següents passes:

- 1- S'inicia l'aplicació
- 2- Posteriorment, s'inicia l'escolta de modificacions en la localització pels proveïdors d'ubicació seleccionats
- 3- Mantenir una "millor estimació actual" de la localització mitjançant el filtratge de les noves posicions capturades però de menys exactitud
- 4- Aturar l'escolta de modificacions en la localització
- 5- Utilitzar l'última millor estimació obtinguda

¹⁵ Apartat "Defining a Model for the Best Performance" a <http://developer.Android.com/guide/topics/location/strategies.html>



Il·lustració 2: Línia de temps en què una aplicació rep modificacions en la localització font: (Android Developer - SDK - Activity class)

A mode de resum, ens proposa les següents accions per minimitzar el consum de bateria:

- Reduir el temps de finestra de peticions actives
- Definir als proveïdors d'ubicació o recursos que tornin actualitzacions menys freqüentment
- Restringir el conjunt de proveïdors d'ubicació a utilitzar

7.3. ArcGIS for Server

Ja s'ha comentat en punts anteriors que ArcGIS for server, és una aplicació servidora que, entre d'altres característiques, permet exposar serveis REST (*Representational State Transfer*) per a funcionalitats GIS.

7.3.1. ArcGIS, REST, JSON i API de Javascript

Com a concepte, REST (Wikipedia REST (Representational State Transfer)) és una tècnica d'arquitectura de programari per a sistemes de text, imatge, vídeo, so, mapes i altres suports d'informació, distribuïts com la World Wide Web. Habitualment utilitza XML sobre HTTP, tot i que aquest esquema pot tenir variacions, com JSON (Introducing JSON) sobre HTTP. JSON és un format d'intercanvi de dades, caracteritzat per la seva simplicitat en la interpretació per part d'humans, i en la facilitat de interpretació i generació per part dels sistemes d'informació.

Es basa en els següents principis:

- És un protocol client/servidor sense estat, que equival a dir que cada missatge conté la informació necessària per comprendre la petició
- Un conjunt d'operacions ben definides que s'apliquen a tots els recursos d'informació
- Una sintaxi universal per identificar els recursos. Cada recurs és accessible únicament a través de la seva URI (*Uniform Resource Identifier*), o més habitualment a través de la seva URL (*Uniform Resource Locator*) o direcció web.
- Ús d'hipermèdia per identificar els recursos

Quant a ArcGIS, és possible consumir els serveis REST exposats a través de peticions JSON sobre HTTP/HTTPS. Es resumeix llavors afirmant que és possible utilitzar els serveis d' ArcGIS fent peticions HTTP enviant i rebent dades amb format JSON. Aquest fet fa que siguin fàcilment integrables en solucions globals, com és el cas d'aquest projecte. A tenir en compte que:

- Es facilita la integració multi plataforma: és viable la comunicació entre plataformes diferents (com per exemple una aplicació desenvolupada amb *Android* SDK i un servidor ArcGIS Server)
- Posa a l'abast tot el potencial d'una aplicació de tercers comés l'ArcGIS Server
- A l'utilitzar JSON, les estructures de dades són simples i fàcilment integrables en les aplicacions fetes a mida

Finalment cal mencionar que ArcGIS posa a disposició una API de Javascript pel consum dels seus serveis REST. Aquesta API permet crear de forma fàcil un *mashup*, o pàgina web que combina dades, presentacions i funcionalitat provinents de diverses fonts, amb la finalitat de crear nous serveis. És a dir, de forma fàcil permet generar una pàgina web que consumeixi els serveis REST exposats. El grans avantatges d'aquesta API són:

- La seva facilitat d'ús. Només cal fer una inclusió en el codi HTML per utilitzar-la:
`<script type="text/javascript" src="http://serverapi.arcgisonline.com/jsapi/arcgis/?v=3.2"></script>`
- La compatibilitat que ofereix respecte als navegadors: ArcGIS és qui s'encarrega que la API sigui compatible, mentre que el programador final n'és un usuari.
- La transparència: L'usuari final no ha d'instal·lar cap component addicional per a visualitzar la informació en el navegador.

8. Descripció de la solució

8.1. Introducció

Un cop realitzat l'anàlisi d'opcions tecnològiques, i una breu introducció a *Android* i ArcGIS for server, es passa a identificar l'arquitectura de la solució escollida, les tecnologies a utilitzar, així com les possibilitats que aquestes ofereixen. També s'hi inclou el desenvolupament d'una d'elles, com es veurà més endavant.

En el disseny general de la solució es fa una separació entre 2 subsistemes:

- Subsistema de captació d'informació
- Subsistema de consum d'informació

Tots 2 subsistemes s'han dissenyat amb una arquitectura per capes:

- **Capa de presentació:** encarregada d'interactuar amb l'usuari, ja sigui donant-li informació o recollint-ne. Aquesta capa es comunica amb la capa de negoci.
- **Capa de negoci:** encarregada d'executar els processos definits. Recullen les interaccions de l'usuari de la capa de presentació, les processen i tornen els resultats a través del mateix canal. També està en contacte amb la capa de dades.
- **Capa de dades:** encarregada d'emmagatzemar les dades i dels accessos d'aquestes segons les indicacions de la capa de negoci.

8.2. Arquitectura de la solució

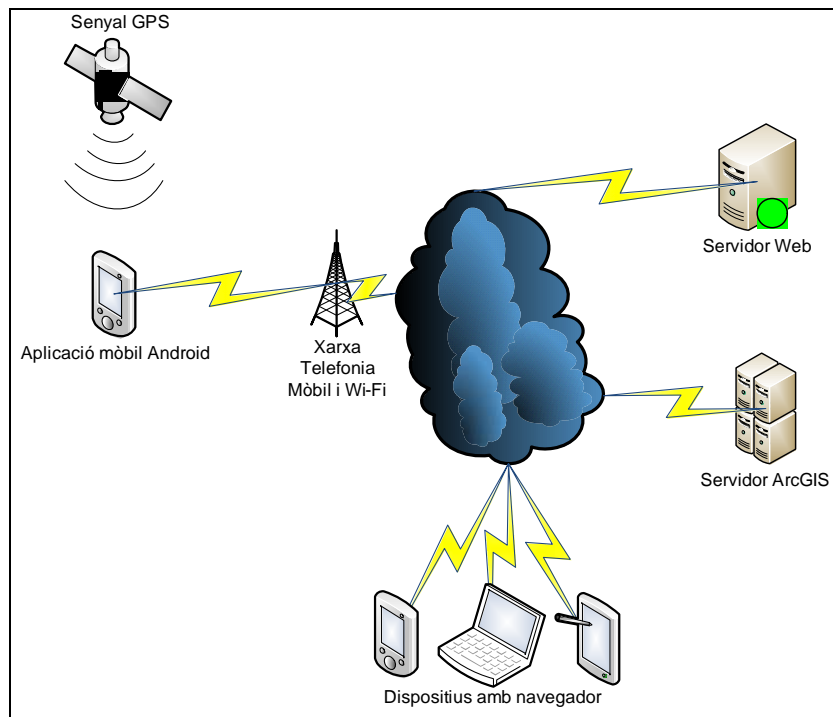
Tal i com s'ha comentat anteriorment, l'arquitectura de la solució estarà composta per 3 parts:

- Mòbil *Android* amb aplicació
- Servidor GIS – ArcGIS Server
- Servidor Web – Internet Information Services

El sistema funcionarà de forma que l'aplicació de l' *Android*, utilitzant un grup d'*API* (Application Programming Interface) de sistema capturarà les coordenades a través de GPS o Xarxa de Telefonía Mòbil, i les enviarà al servidor GIS. El servidor web proveirà un recurs HTML on es mostri un mapa amb les posicions de tots els equips. Aquesta informació l'extraurà fent una consulta al servidor GIS.

8.3. Vista física

A continuació en la Il·lustració 3: Arquitectura global del sistema, es mostra la vista física de l'arquitectura global del sistema.



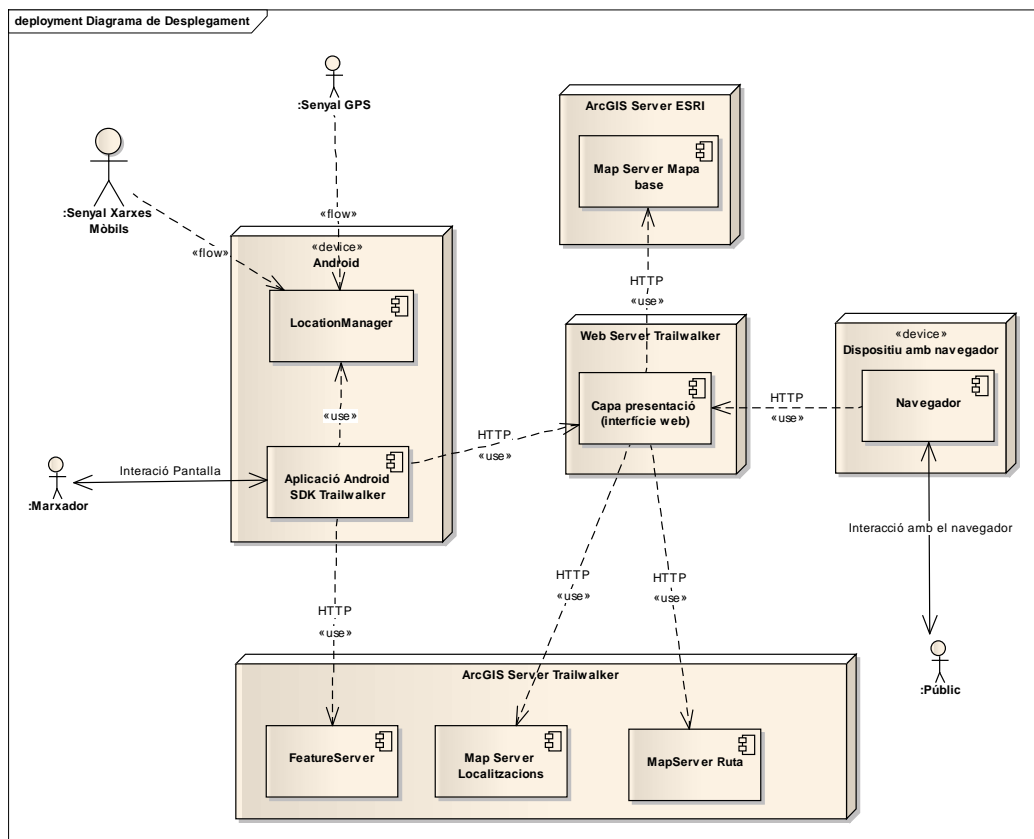
Il·lustració 3: Arquitectura global del sistema

La connectivitat entre els components serà de tipus WAN, utilitzant HTTP sobre el protocol TCP/IP.

En la figura s'afegeix un quart element, que és el grup de dispositius amb navegador. Aquest dispositius, malgrat no formar part directament de la solució, es representa en el diagrama ja que intervé en la solució, concretament en la de proveir informació a l'organització i públic en general.

8.4. Vista de desplegament

En la Il·lustració 4: Diagrama de desplegament del sistema, es descriu el diagrama de desplegament del sistema:



Il·lustració 4: Diagrama de desplegament del sistema

Com es pot observar en el diagrama, el sistema es divideix en 5 nodes diferents:

- Dispositiu *Android*: telèfon mòbil que portaran els marxadors i que intervé en tots 2 subsistemes:
 - o Subsistema de captació d'informació: de forma periòdica detectarà la seva posició, a través del component "LocationManager" de l'SDK d' *Android* que és capaç de determinar les coordenades a partir de GPS o Xarxa de Telefonia Mòbil. Aquesta localització va acompanyada de la data i hora, així com de l'estat de la bateria i l'identificador IMEI del telèfon. Aquesta informació serà enviada al node "ArcGIS Server Trailwalker".
 - o Subsistema de consum d'informació: Des del dispositiu *Android*, un marxador d'un equip podrà consultar la seva posició actual i la de la resta d'equips.
- ArcGIS Server Trailwalker: servidor d' ArcGIS for Server. Quant als 2 subsistemes, aquest node intervé de la següent manera:
 - o Subsistema de captació d'informació: rebirà la informació recollida dels del dispositiu *Android* i l'emmagatzemarà en la base de dades GIS. Aquesta operació es realitzarà a través del servei "Feature" publicat a tal efecte.
 - o Subsistema de consum d'informació: proveeix la informació de localització de tots els dispositius *Android* (els equips), així com d'altres dades com és la ruta de la cursa. Aquesta informació és consumida des de la capa de presentació, en el Web Server TrailWalker.
- Web Server Trailwalker: servidor web que proveirà la capa de presentació (localització equips) a els marxadors, a través de l'aplicació *Android* SDK Trailwalker, i al públic en general a través d'un navegador. Aquest node només intervé en el subsistema de consum d'informació, i utilitza recursos de l'ArcGIS Server Trailwalker i ArcGIS Server ESRI.

- ArcGIS Server ESRI: servidor de mapes. Aquest servei serà consumit per la capa de presentació per mostrar la informació continguda en el ArcGIS Server Trailwalker sobre una base de mapa. Només intervé en el subsistema de consum d'informació.
- Dispositiu amb navegador: equival a qualsevol element que pugui executar un navegador web. Serà el mitjà amb el qual qualsevol persona (organització i públic en general) pugui visualitzar la posició dels equips. Només intervé en el subsistema de consum d'informació, i per tant només té relació amb la capa de presentació.

8.5. Tecnologies a utilitzar

Com ja s'ha vist en la vista física i en la vista de desplegament, bàsicament es disposarà de:

- Aplicació feta a mida en *Android*, amb *Android SDK*
- Servidor ArcGIS de ESRI
- Servidor ArcGIS de la UOC (Windows Server 2008)
- Servidor Web

8.5.1. Aplicació feta a mida en Android, amb Android SDK

Es tractarà d'una aplicació funcionament en un dispositiu *Android*. L'aplicació estarà desenvolupada amb l'SDK d' *Android*. I es codificarà en Java. Aquesta aplicació serà l'encarregada de realitzar les tasques del dispositiu *Android* dins la solució global. Per a maximitzar l'ús de l'aplicació en els marxadors, s'ha definit que l'aplicació es pugui instal·lar i executar en sistemes *Android* amb versió mínima de 2.2. L'aplicació està pensada per ser utilitzada en sistemes *Android* amb versió 4.0.x.

En la Taula 3: Ús de versions d' *Android* a 3/12/2012 - , s'indica el percentatge d'ús de les diferents versions d' *Android*. A la taula de la font, s'hi ha afegit la columna "Coverage minimum version", a on s'hi detalla el percentatge de cobertura que es dona a una aplicació si es fixa la versió mínima en cada cas. S'ha decidit fixar la versió mínima "2.2 – Froyo", ja que s'hi aconsegueix un salt qualitatiu entre "2.3 – Gigerbread", d'un 10% aproximadament. Les versions més antigues a 2.2 representen menys de l'1% amb el que reforça la decisió que s'ha de mantenir la retro compatibilitat fins la 2.2 només.

Version	Code name	Release date	API level	Distribution (December 3, 2012)	Coverage minimum version
4.2	Jelly Bean	November 13, 2012	17	0,80%	0,80%
4.1.x	Jelly Bean	July 9, 2012	16	5,90%	6,70%
4.0.x	Ice Cream Sandwich	December 16, 2011	15	27,50%	34,20%
3.2	Honeycomb	July 15, 2011	13	1,20%	35,40%
3.1	Honeycomb	May 10, 2011	12	0,40%	35,80%
2.3.3–2.3.7	Gingerbread	February 9, 2011	10	50,16%	85,96%
2.3–2.3.2	Gingerbread	December 6, 2010	9	0,20%	86,16%
2.2	Froyo	May 20, 2010	8	10,30%	96,46%
2.0–2.1	Eclair	October 26, 2009	7	2,70%	99,16%
1.6	Donut	September 15, 2009	4	0,30%	99,46%
1.5	Cupcake	April 30, 2009	3	0,10%	99,56%

Taula 3: Ús de versions d' *Android* a 3/12/2012 - (Wikipedia - Android)

Cal destacar que el dispositiu *Android* haurà de tenir receptor GPS (l'aplicació l'utilitzarà) integrat. Tant la tecnologia GPS com el servei de "Servei d'ubicació de Google" seran els components de l' *Smartphone* utilitzats per la recepció de les posicions en l'aplicació.

Els dispositius *Android* hauran de comptar amb una línia de dades associada, que permetrà la comunicació entre ells i els servidors ArcGIS i Web. L'aplicació realitzarà peticions HTTP amb estructures de dades de tipus JSON. Aquest fet fa que l'aplicació sigui independent de la plataforma receptora de les dades.

En el punt 8.7. 8.7 Implementació Aplicació *Android* SDK es detalla la implementació d'aquesta aplicació.

8.5.1.1. Per què *Android*?

Cal especificar que s'ha escollit *Android* com a plataforma mòbil per davant d'altres plataformes (iOS, RIM, Windows...) per motius d'abast de públic (*Android* és un sistema molt comú) i de facilitat en la construcció d'aplicacions per *Smartphones*. Malgrat que el projecte està centrat en una única tecnologia mòbil, les altres plataformes són candidates totalment viables per a posteriors ampliacions de la solució.

8.5.2. Servidor ArcGIS de ESRI

ESRI proporciona una sèrie de serveis entre els quals hi ha el de visualització de capes de mapes. Per la solució d'aquest projecte, es farà un consum d'un d'aquests serveis a través de REST. A mode d'informació, es poden consultar els serveis REST disponibles en el següent enllaç: <http://server.arcgisonline.com/ArcGIS/rest/services/>. La versió d' ArcGIS Server instal·lat a data de redacció d'aquesta memòria és 10.1.

Dels serveis REST publicats en la URL <http://server.arcgisonline.com/ArcGIS/rest/services/>, en aquesta solució es suggereix utilitzar el servei utilitzat "World_Street_Map". Aquest servei ofereix una capa de mapa de fons de tot el món, amb la informació de carreteres i poblacions. Aquest servei REST pot ser consumit a través de la API Javascript d' ArcGIS. Es pot consultar el detall del servei a: http://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer

8.5.3. Servidor ArcGIS Trailwalker

Dins de l'àmbit d'aquest projecte, la UOC ha proporcionat un servidor amb una versió d' ArcGIS Server instal·lada i configurada. Els serveis de l'ArcGIS Server són accessibles des d'internet. La versió d' ArcGIS Server instal·lat és 10.1. Aquest servidor ArcGIS gestionarà la informació de posicionament dels equips. També contindrà informació addicional com és la ruta de la cursa.

El mateix servidor, també s'ha proporcionat una sèrie de serveis publicats per a ser utilitzats en aquest projecte. Són els següents:

- Servei de rutes
- Servei de visualització de l'últim punt (posició) de tots els equips
- Servei de visualització de tots els punts (posicions) de tots els equips
- Servei d'addició de punts nous (posicions) en el servidor

Els serveis estan publicats en la URL: <http://NomDelServidor/arcgis/rest/services>. No s'indica el nom actual del servidor ja que aquest és susceptible de que canviï.

8.5.3.1. Servei de rutes

El nom d'aquest servei en el servidor és "rutes(MapServer)". Aquest servei conté la informació de la ruta de la cursa amb les diferents etapes. A través de la API de Javascript és possible mostrar aquesta informació sobre una capa de mapa (veure punt ArcGIS Server de ESRI). La URL del servei és: <http://NomDelServidor/arcgis/rest/services/rutes/MapServer>.

8.5.3.2. Servei d'últim punt

El nom d'aquest servei en el servidor és "ÚltimaEntrada (MapServer)". Aquest servei mostra el punt (posició) més nou de cada equip, és a dir, mostra l'última posició capturada i enviada al servidor ArcGIS Trailwalker. La llegenda, està formada per punts de diferents colors. El punt indica la posició de l'equip, i el color indica el nivell de bateria que el dispositiu disposa en el moment de captura de la posició. La URL del servei és: <http://NomDelServidor/arcgis/rest/services/UltimaEntrada/MapServer> .

8.5.3.3. Servei visualització tots els punts

El nom d'aquest servei és "TrailWalker (MapServer)". La funcionalitat del servei és la mateixa que la del "Servei d'últim punt", amb la diferència que aquest mostra tots els punts. La URL del servei és: <http://NomDelServidor/arcgis/rest/services/TrailWalker/MapServer> .

8.5.3.4. Servei d'addició de punts nous

El nom d'aquest servei és "TrailWalker (FeatureServer)". Aquest servei, a diferència dels altres ofereix una funcionalitat d'escriptura. És a dir, no s'utilitza per extreure informació sinó per afegir informació en el servidor ArcGIS. En aquest cas, l'aplicació d' *Android* consumirà aquest servei per a afegir nous punts (localitzacions) a través de peticions HTTP. S'utilitza el servei mitjançant l'enviament de dades en format JSON. La URL del servei és: <http://NomDelServidor/arcgis/rest/services/TrailWalker/FeatureServer> .

Quant al tipus de petició HTTP, el servei espera peticions de tipus POST amb els següents paràmetres:

Nom de paràmetre	Valor
f	Json
adds	Cadena JSON amb la informació*

* A continuació es mostra el patró de cadena JSON a incloure en el paràmetre "adds:

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

Les cadenes que contindran valors variables són:

x: Valor de latitud de la posició del dispositiu emissor

y: Valor de longitud de la posició del dispositiu emissor

CodiEquip: IMSI del dispositiu emissor

Equip: Nom de l'equip especificat en l'aplicació d' Android.

Bateria: Nivell de bateria del dispositiu en el moment de realitzar la captura de posició.

Entrada: Data i hora de captura de posició del dispositiu

A continuació es mostra un exemple de cadena JSON per aquest servei:

```
{ "geometry": { "x": 317597.9340811574, "y": 5135996.4232025035, "spatialReference": { "wkid": 102100 } }, "attributes": { "CodiEquip": 7160652952503786, "Equip": "uoc", "Bateria": 78, "Entrada": 1319666400000, "Tipus": "Android" } }
```

8.5.4. Servidor Web

Serà l'encarregat de servir la pàgina web de consulta de l'estat de la cursa. La pàgina web pot ser fàcilment construïda amb codi HTML estàtic i ús de Javascript i API de Javascript proporcionats per ArcGIS (veure punt 7.3.1 ArcGIS, REST, JSON i API de Javascript o la documentació oficial d' ArcGIS a (ArcGIS API for JavaScript - ArcGIS Resource Center)).

En el marc del projecte se n'ha construït una per demostrar les capacitats de l'API de Javascript d'ArcGIS.

8.6. Casos d'ús

Seguint els requeriments descrits s'han definit els següents casos d'ús:

8.6.1. Visualització de localització i rànkung d'equips

DESCRIPCIÓ: Qualsevol persona (marxador, organització, públic...) ha de ser capaç de visualitzar la localització i classificació dels equips.

ACTORS: Qualsevol persona

PRECONDICIONS: Cap.

FLUXE NORMAL:

Qualsevol persona	Sistema
	1. De forma recurrent, captura la posició, dia, hora, IMEI i capacitat de la bateria de tots els dispositius mòbils (localitzacions dels equips)
	2. De forma recurrent, emmagatzema les localitzacions dels equips
3. Un usuari fa una petició de visualització de la informació de localització dels equips en un mapa	4. El sistema mostra la informació de localització dels equips emmagatzemada fins llavors, en un mapa

8.6.2. Activació i desactivació de captura de posicions

DESCRIPCIÓ: Els marxadors seran capaços d'activar o desactivar la captura de posicions de l'aplicació en *Android*.

ACTORS: Marxadors

PRECONDICIONS: L'aplicació en *Android* està engegada.

FLUXE NORMAL:

Marxador	Aplicació en <i>Android</i>
1. Escull activar o desactivar la captura de posicions per GPS o "Servei d'ubicació de Google".	2. L'aplicació inicia o atura la captura de posicions a través de GPS o "Servei d'ubicació de Google".

EXTENSIONS:

- 2.1. Si l'aplicació atura la captura de posicions per GPS o de "Servei d'ubicació de Google", deixarà de realitzar aquesta acció, i d'emmagatzemar-les en el servidor, pel que no es generarà informació pel cas d'ús "Visualització de localització i rànquing d'equips".
- 2.2. Si l'aplicació inicia la captura de posicions per GPS o "Servei d'ubicació de Google", iniciarà l'acció i començarà a emmagatzemar-les en el servidor, pel que la informació tornarà a estar disponible pel cas d'ús "Visualització de localització i rànquing d'equips".

8.6.3. Modificació del nom de l'equip

DESCRIPCIÓ: Els marxadors seran capaços de modificar el nom de l'equip que identificarà les posicions capturades pel seu dispositiu.

ACTORS: Marxadors

PRECONDICIONS: L'aplicació en *Android* està engegada. L'aplicació proposa un nom d'equip per defecte a l'iniciar l'aplicació.

FLUXE NORMAL:

Marxador	Aplicació en <i>Android</i>
1. Escull l'opció de menú per modificar el nom de l'equip	2. L'aplicació mostra un formulari amb un camp de text editable, que inclou el nom antic
3. L'usuari modifica el nom de l'equip i prem "D'acord".	4. L'aplicació guarda el nou nom d'equip. Es tornarà a mostrar la pantalla principal de l'aplicació.

EXTENSIONS:

3.1. Si l'usuari prem "Cancel·la", no es realitza cap canvi i es torna a mostrar la pantalla principal de l'aplicació.

POSTCONDICIONS: Si l'usuari modifica el nom de l'equip, l'aplicació l'utilitzarà per identificar les posicions capturades a partir d'aquell moment

8.6.4. Modificació configuració aplicació

DESCRIPCIÓ: Els marxadors seran capaços de modificar la configuració de l'aplicació, previ avís dels riscos que això comporta.

ACTORS: Marxadors

PRECONDICIONS: L'aplicació en *Android* està engegada. L'aplicació proposa una configuració per defecte.

FLUXE NORMAL:

Marxador	Aplicació en <i>Android</i>
1. Escull l'opció de menú per modificar la configuració de l'aplicació	2. L'aplicació mostra un formulari amb les dades de configuració de l'aplicació
3. L'usuari fa les modificacions pertinents i prem "D'acord".	4. L'aplicació guarda la nova configuració i aplica els canvis al seu funcionament. Es tornarà a mostrar la pantalla principal de l'aplicació.

EXTENSIONS:

3.1. Si l'usuari prem "Cancel·la", no es realitza cap canvi i es torna a mostrar la pantalla principal de l'aplicació.

POSTCONDICIONS: Si l'usuari modifica la configuració de l'aplicació, a partir d'aquell moment l'aplicació utilitzarà la nova configuració per funcionar.

NOM: Modificació de l'idioma de l'aplicació

DESCRIPCIÓ: Els marxadors seran capaços de modificar l'idioma dels textos de l'aplicació. Els idiomes disponibles seran: català, castellà i anglès.

ACTORS: Marxadors

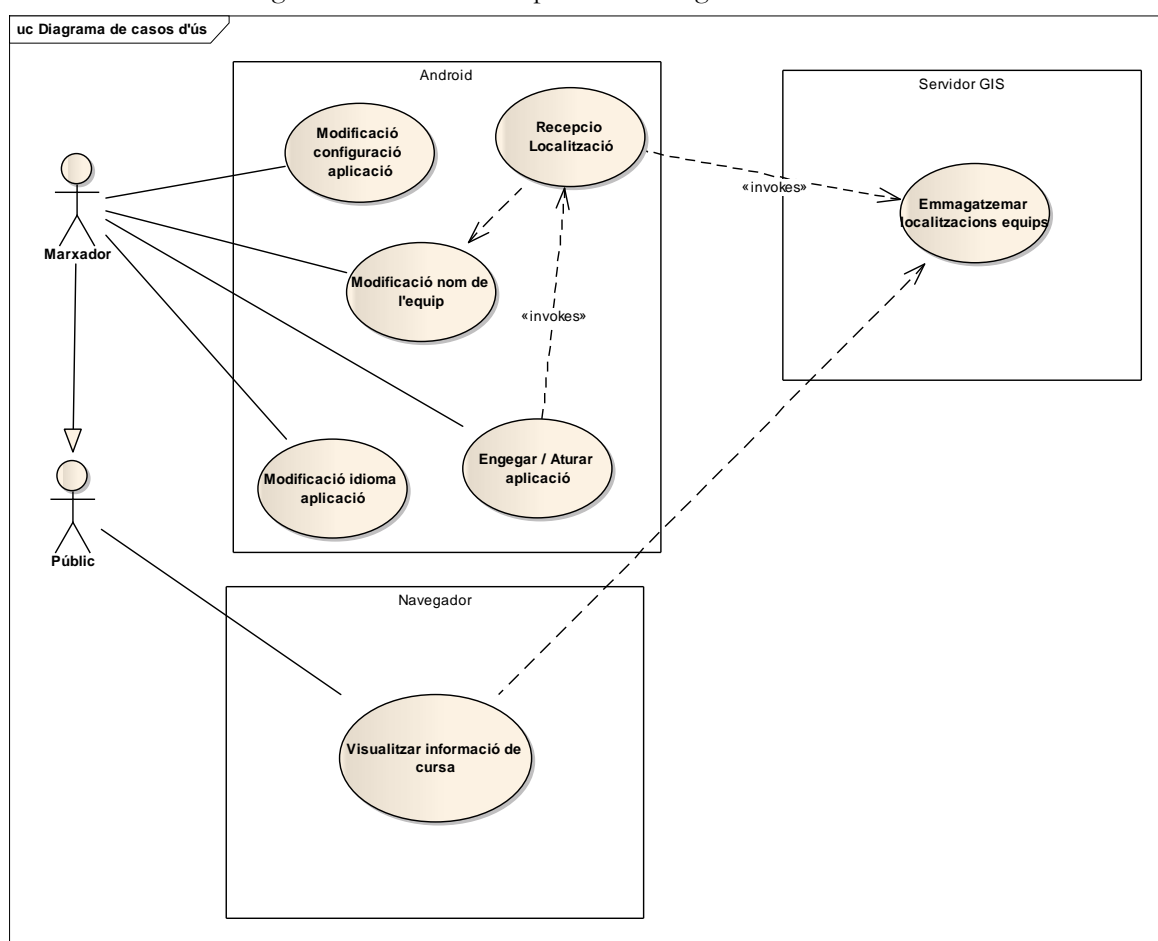
PRECONDICIONS: L'aplicació en *Android* està engegada.

FLUXE NORMAL:

Marxador	Aplicació en <i>Android</i>
1. Modifica l'idioma del sistema operatiu del dispositiu.	2. L'aplicació es mostra en el mateix idioma que el sistema operatiu.

POSTCONDICIONS: Si l'usuari modifica l'idioma del sistema operatiu del dispositiu, i aquest no és català, castellà o anglès, per defecte l'aplicació mostrarà l'anglès.

En la Il·lustració 5: Diagrama de casos d'ús es presenta el diagrama dels casos d'ús descrits.



Il·lustració 5: Diagrama de casos d'ús

8.7. Implementació Aplicació *Android* SDK

8.7.1. Introducció

Dins de l'estudi de la solució, s'hi inclou la implementació de l'aplicació per *Android*.

8.7.2. Requeriments de l'aplicació per *Android*

A continuació es detallen els requeriments específics de l'aplicació per *Android*. Aquests venen derivats de la solució global definida, dels requeriments propis de l'enunciat del projecte, i s'han plasmat en els

casos d'ús. També inclouen certs requeriments no funcionals. A partir d'aquest punt es definirà l'arquitectura de l'aplicació i el disseny de la interfície d'usuari.

- L'aplicació haurà de ser accessible per l'usuari
- L'aplicació haurà de permetre a l'usuari especificar el nom d'equip. Aquesta serà la informació que identificarà l'autor dels punts de ruta en el mapa (en la visualització d'informació en la web).
- Haurà de permetre modificar-ne la configuració de funcionament, malgrat que caldrà avisar a l'usuari dels perills que això comporta (opció destinada a usuaris avançats)
- Haurà de capturar la informació necessària i en els moments necessaris per a ser enviada posteriorment al servidor ArcGIS Trailwalker.
- La informació a capturar serà:
 - o Posició
 - o Dia i Hora en què s'ha capturat la posició
 - o IMSI (identificador de la targeta SIM) del telèfon
 - o Nivell de bateria en el moment de captura de posició
- La informació serà capturada cada 10 minuts (temps considerat com a suficient per a portar un seguiment suficient, dintre del marc d'una cursa que dura hores).
- La informació serà enviada immediatament al servidor ArcGIS Trailwalker després de la captura.
- L'aplicació ha d'enviar la informació al servidor ArcGIS amb peticions HTTP. Enviarà estructures JSON, definides en els serveis publicats del servidor ArcGIS.
- El sistema serà tolerant a situacions sense cobertura de senyal GPS i/o de dades. En casos en què no sigui possible enviar dades, la informació haurà de ser transmesa posteriorment.
- L'usuari ha de ser capaç d'iniciar o aturar la captura de posicions de l'aplicació
- L'aplicació ha de ser compatible amb la majoria de versions d' *Android* que hi ha al mercat. S'aconsegueix un 99,16% de cobertura amb la versió 2.2, com s'ha indicat en el punt Tecnologies a utilitzar.
- L'aplicació serà multi Idioma
- L'usuari haurà de poder veure un resum de posicions capturades, enviades i pendents d'enviar, així com el dia i hora del refresc de la informació
- L'aplicació ha de promoure un consum mínim de bateria
- L'aplicació continuï funcionant en segon pla, quan es prem el botó "enrere" del dispositiu. Habitualment aquest botó tanca les aplicacions.

8.7.3. Disseny de la Interfície d'Usuari

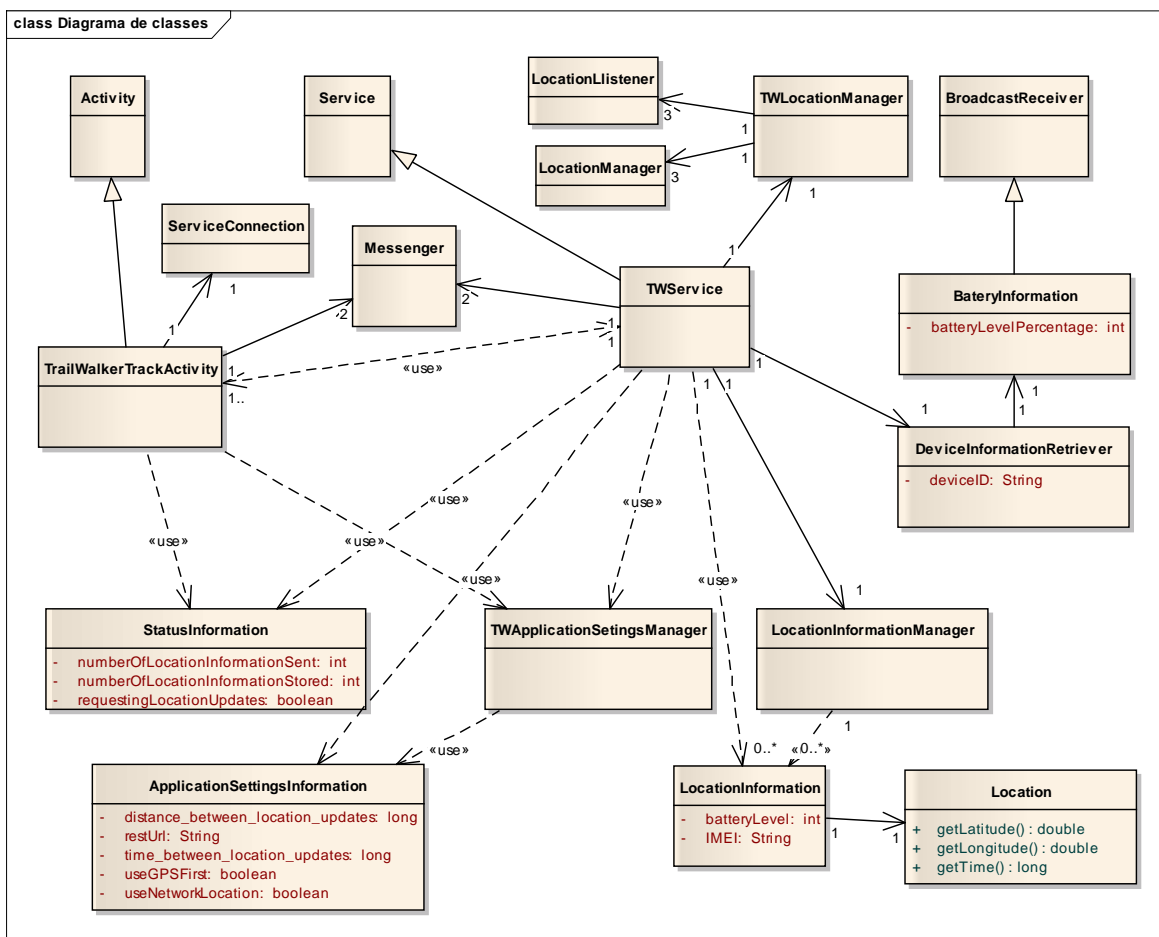
Per motius de manca de temps, no s'han pogut seguir res recomanacions d'usabilitat i disseny d'aplicacions per *Android* segons la documentació d' *Android Developer*.

8.7.4. Estructura de classes i lògica de l'aplicació

A continuació es descriurà el procés de definició de l'arquitectura de l'aplicació *Android SDK*. Com s'ha comentat en punts anteriors, l'aplicació a desenvolupar estarà realitzada amb *Android SDK* i el llenguatge de programació serà Java i es desenvoluparà seguint el paradigma de Programació Orientada a Objectes (POO).

8.7.4.1. Estructura de classes

En aquest apartat es descriu l'estructura de classes de l'aplicació *Android SDK*. En la Il·lustració 6: Diagrama de classes, es presenta el diagrama del disseny de classes.



Il·lustració 6: Diagrama de classes

A continuació s’ofereix una breu descripció de les classes que intervenen en l’aplicació:

TrailWalkerTrackActivity: Classe que hereta de la classe d’ *Android SDK Activity*, que implementa la interfície d’usuari. Utilitza classes d’ *Android SDK ServiceConnection* i *Messenger* per realitzar la comunicació amb “TWService”.

TWService: Classe que hereta de la classe d’ *Android SDK Service*, que funciona com el nucli de l’aplicació. Utilitza una instància de la classe “DeviceInformationRetriever” per obtenir les dades referents al dispositiu (estat bateria i IMSI), una instància de la classe “TWLocationManager” per gestionar el posicionament i obtenir les posicions, i una instància de la classe “LocationInformationManager” per gestionar l’enviament de posicions al servidor ArcGIS.

TWLocationManager: Gestiona el posicionament (quan realitzar les captures, quin servei emprar per les captures...), obté les posicions i transmet la informació al “TWService”.

ApplicationSettingsInformation: Classe contenidora d’informació referent a la configuració de l’aplicació.

TWApplicationSettingsManager: Classe que gestiona la persistència de dades de l’aplicació, com són la configuració de l’aplicació i el nom de l’equip. Utilitza la classe “ApplicationSettingsInformation” per intercanviar la informació amb altres classes.

StatusInformation: Classe contenidora d’informació per a la comunicació entre “TWservice” i “TrailWalkerTrackActivity”.

LocationInformation: Classe contenidora de la informació que serà enviada al servidor ArcGIS. Té capacitat per generar codi JSON. Utilitza una instància de la classe d’ *Android SDK Location*.

BatteryInformation: Classe utilitzada per obtenir les dades del nivell de bateria. Hereta de la classe d' *Android* SDK "BroadcastReceiver" per a poder subscriure's a *events* de sistema.

DeviceInformationRetriever: Classe utilitzada per obtenir les dades referents al dispositiu (IMEI/IMSI) i estat de la bateria, utilitzant una instància de "BatteryInformation".

LocationInformationManager: Classe delegada de "TWSERVICE". S'encarrega de gestionar la informació un cop capturada. Realitza l'enviament al servei REST del servidor ArcGIS. Utilitza instàncies del tipus "LocationInformation". Utilitza una instància de "DeviceInformationRetriever".

8.7.4.2. Estructura del codi i justificació

Seguint les recomanacions de la documentació de *Android* SDK, l'esquelet de l'aplicació estarà format bàsicament per dues classes:

- Classe per representar la interfície d'usuari, i executar certa lògica de negoci. Aquesta classe serà una especialització¹⁶ de la classe *Activity*¹⁷. El seu nom és "TrailWalkerTrackActivity"
- Classe per orquestrar les operacions de captura de posicions i enviament de dades al servidor. Aquesta classe serà una especialització de la classe *Service*¹⁸. El seu nom és "TWSERVICE".

S'ha escollit aquest esquelet bàsic perquè a nivell d'arquitectura, l'aplicació no té més complexitat. A través de la "TrailWalkerTrackActivity" l'usuari interactua amb l'aplicació. Aquesta classe serà la responsable d'arrencar la classe "TWSERVICE". Els *Services* d' *Android* SDK són classes pensades per a executar processos en segon pla. Per això se li ha assignat la tasca d'orquestració de captura de posicions i enviament de dades. Cal tenir present, i així ho indica la documentació pròpia de la classe, que un *Service* no arrenca un nou fil, i que en cas d'haver d'executar tasques amb gran despesa de CPU, s'hauria d'abordar el problema des d'una altra perspectiva: obrir un nou fil a dins del "TWSERVICE" (classe *Thread*), ús de la classe *AsyncTask*... Per més referència es pot consultar la part de la documentació dedicada a aquest punt: <http://developer.Android.com/guide/components/processes-and-threads.html>.

Si no es fes així, la *Activity* es veuria afectada, i l'usuari percebria un rendiment pobre en la interfície usuari. Per sort, no és el cas de la nostra situació. Hem de tenir present que les operacions que el *Service* té assignades són capturar posició i enviar les dades al servidor.

S'ha aplicat el principi de modularitat, amb el que s'han delegat rols a altres classes. Com s'ha indicat anteriorment La classe "TWSERVICE" fa, en certa manera, de controlador i delega certes funcions a altres classes es veu més endavant.

8.7.4.3. Captura de posicions

La captura de posicions ha esdevingut un dels punts clau en aquest projecte. Com ja s'ha indicat anteriorment, hi ha documentació i referències a internet, així com la documentació de *Android* Developer, que indiquen que l'ús del receptor GPS de forma massiva, augmenta considerablement el consum de bateria del dispositiu. Per tant es pot afirmar que la captura de posicions és el que decantarà cap a un costat o l'altre de la balança, si l'aplicació és o no respectuosa amb el consum de bateria.

¹⁶ Especialització, fa referència a un dels conceptes de la programació orientada a objectes, com és l'herència. La classe heretarà d'una classe mare.

¹⁷ La informació de referència d'aquesta classe és: <http://developer.Android.com/reference/Android/app/Activity.html>

¹⁸ <http://developer.Android.com/reference/Android/app/Service.html>

Per tant el problema plantejat, no era tant de resolució tècnica, sinó de definició de solució òptima, entre els resultats esperats (capturar posicions) i el consum de bateria. Com s'ha comentat en el punt Posicionament i rendiment de bateria en sistemes *Android*, la documentació d' *Android Developer* fa referència a trobar un equilibri entre l'exactitud del posicionament i el consum de bateria. Sugereix fer això en base a:

- Reduir el temps de finestra de peticions actives
- Definir als proveïdors d'ubicació o recursos que tornin actualitzacions menys freqüentment
- Restringir el conjunt de proveïdors d'ubicació a utilitzar

Si comencem pel primer punt, pel nostre cas no és aplicable, ja que constantment (cada 10 minuts) hem d'estar posicionant el dispositiu. Aquest punt és aplicable per a aplicacions que realitzen posicionament en un moment determinat, que no és el cas.

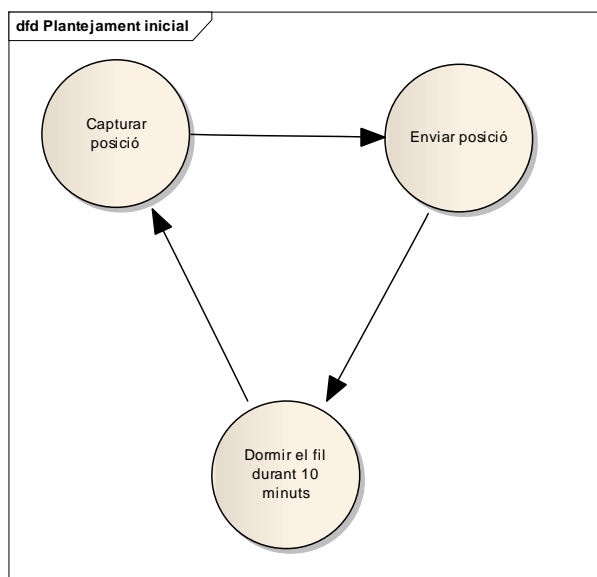
Quant al punt de restringir el conjunt de proveïdors d'ubicació a utilitzar, ha sigut un punt a tenir en compte. Inicialment, la solució s'ha enfocat a utilitzar únicament el proveïdor de GPS, de forma que el proveïdor de posicionament per "Servei d'ubicació de Google" no s'utilitza. Cal indicar que, a banda de no ser utilitzat, és recomanable desactivar-lo a nivell de sistema operatiu, per evitar el consum de bateria que pugui ocasionar.

Respecte al punt de reduir el temps entre actualitzacions d'ubicació, per sort estem en el cas en què els requeriments ens demanen una actualització que es podria considerar poc freqüent (10 minuts). Igual que amb el primer punt, aquest fa referència a aplicacions que fan un ús massiu del servei.

Per últim, cal destacar que l'escolta passiva és l'opció menys agressiva en el consum de bateria, ja que l'aplicació queda a l'escolta de posicions sol·licitades per altres aplicacions. Tot i així, el seu punt fort també és el punt que fa descartar aquesta opció pel cas que ens toca. Una escolta passiva no permet assegurar la cadència de localitzacions. De cara a oferir una informació homogènia, es necessita que les posicions dels equips s'actualitzin amb temps més similars possibles.

Plantejament Inicial

El plantejament inicial va ser el de programar un cicle iteratiu del tipus:



Il·lustració 7: Diagrama de flux de dades en plantejament inicial

La idea era tenir control exacte de quan s'executaven les peticions de captura de posició (*Android SDK* no t'assegura que amb la funció "requestLocationUpdates" les captures de posició siguin amb el temps mínim indicat de forma exacta), i addicionalment la possibilitat d'aturar el servei de GPS

i/o de dades durant el període d'espera per minimitzar al màxim el consum de bateria. Aquest últim punt no s'ha testejat ni comprovat la seva viabilitat.

Per aconseguir que el cicle s'executés de forma indefinida, es cridava a una funció a través d'un parell de classes que permeten fer programacions d'execucions de funcions en un fil separat:

- Classe *Timer*
- Classe *Timertask*

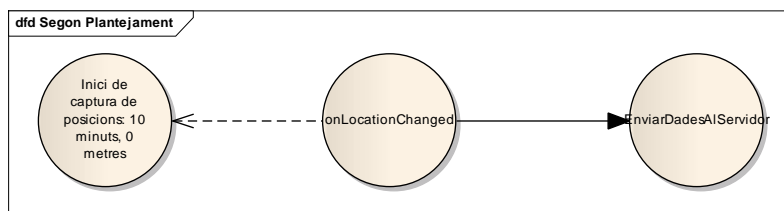
Això implicava cridar el mètode “requestSingleUpdate” del *LocationManager* en primera opció, capturar la posició tornada i enviar-la al servidor ArcGIS.

Aquesta opció es va desestimar per diferents motius:

- 1- Hi havia problemes en la sincronització del fil creat pel *Timer* amb el fil principal, a causa de la crida “requestSingleUpdate” del *LocationManager*.
- 2- Al dormir el fil, l'event “onLocationChanged” de la classe *LocationListener* era cridat quan el fil tornava a aixecar-se, amb el que aconseguíem que, quan s'estava capturant una posició, el sistema se n'adonava que havia canviat la localització (corresponent a la iteració anterior), amb el que s'enviava la posició amb un cicle de retràs. Això no és viable si tenim en compte que el temps entre captures ha de ser de 10 minuts.
- 3- La petició de localització és un procés asíncron, pel que no es pot assegurar que el resultat sigui immediat posterior a la crida.
- 4- En resum, s'estava intentant fer el que ja es pot fer amb la funció “requestLocationUpdates” de la classe *LocationManager*.
- 5- Finalment, el mètode “requestSingleUpdate” està disponible a partir de l'API SDK 9, mentre que el projecte està enfocat a proporcionar compatibilitat des de la versió ADPI SDK 8, que fa inviable l'ús d'aquest mètode.

Segon Plantejament

Es descarta la idea d'implementar el cicle iteratiu. En comptes d'això, s'utilitza la funció “requestLocationUpdates” de la classe *LocationManager*.



Il·lustració 8: Diagrama de flux de dades en plantejament final

En la crida de la funció se li especifica el temps mínim i distància mínima entre peticions de localització. Per tant, un cop especificats els paràmetres en la crida de la funció, és el sistema qui se n'encarrega de dur el control de quan fer la localització del dispositiu.

L'estructura del codi és molt més senzilla, i es delega al sistema operatiu que realitzi el control. En la implementació de “onLocationChanged” de *LocationListener*, s'hi inclou l'enviament de les dades al servidor ArcGIS.

Com a contrapartida, cal destacar que el paràmetre de temps mínim de la crida “requestLocationUpdates”, no implica que el sistema respecti les peticions de localització a aquest temps, sinó que el que ens certifica és que no farà localitzacions abans d'haver consumit aquest temps des de l'última localització. Això pot arribar a succeir si el sistema està ocupat utilitzant el servei GPS per una altra aplicació, amb el que hi hauria un retràs.

Com a resultat, la funció per iniciar la captura de posicions és:

```
private void startGPSLocationUpdates(){
    checkIsGPSLocationProviderEnabled();
```

```

    if(locatingGPS)return;
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
    minTimeBetweenGPSLocUpdates, minDistBetweenGPSLocUpdates, locationListenerGPS);
    locatingGPS = true;
}

```

Els events s'han capturat en la implementació de la classe *LocationListener*.

```

private void initializeLocationManagerAndListener(){
    locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);
    locationListener = new LocationListener() {
        public void onLocationChanged(Location location) {
            storeLocationInformation(location);
        }
        public void onStatusChanged(String provider, int status, Bundle extras) {}
        public void onProviderEnabled(String provider) {
            startRequestLocationUpdates();
        }
        public void onProviderDisabled(String provider) {
            stopRequestLocationUpdates();
        }
    };
}

```

En aquest fragment de codi, es pot comprovar que la funció “storeLocationInformation(location)” és cridada quan s’ha realitzat una captura de posició. Aquesta funció emmagatzema la localització per al seu enviament immediat.

Per aturar la captura de posicions, s’ha fet realitzant una crida a la funció “removeUpdates” de la classe *LocationManager*.

```

private void stopGPSLocationUpdates(){
    locationManager.removeUpdates(locationListenerGPS);
    locatingGPS = false;
}

```

Control estat GPS

L’aplicació és tolerant a eventuais apagades del GPS per part de l’usuari. Per tant, quan l’aplicació està en mode “trobar posicions” activat, verifica que el GPS estigui encès. Per fer això s’han fet 2 accions:

- Controlar que quan l’usuari premi el botó de començar a buscar posicions, el GPS estigui controlat. Això es realitza fent una crida a la funció “isProviderEnabled(LocationManager.GPS_PROVIDER)” d’una instància de *LocationManager*. Torna cert o fals en funció si el GPS està o no activat. La crida a aquesta funció es realitza abans de la crida de “requestLocationUpdates”. Si no està activat, torna el valor “false”, i s’envia la informació a l’*Activity* per a que l’usuari faci el pertinent (activar el servei GPS).
- Implementar les funcions de captura d’events de desactivació del GPS (**public void onProviderDisabled(String provider)**), en la classe *LocationListener*. De forma que quan el sistema se n’adona que el GPS s’ha desactivat, immediatament dona l’ordre de deixar de capturar posicions.

Plantejament Final

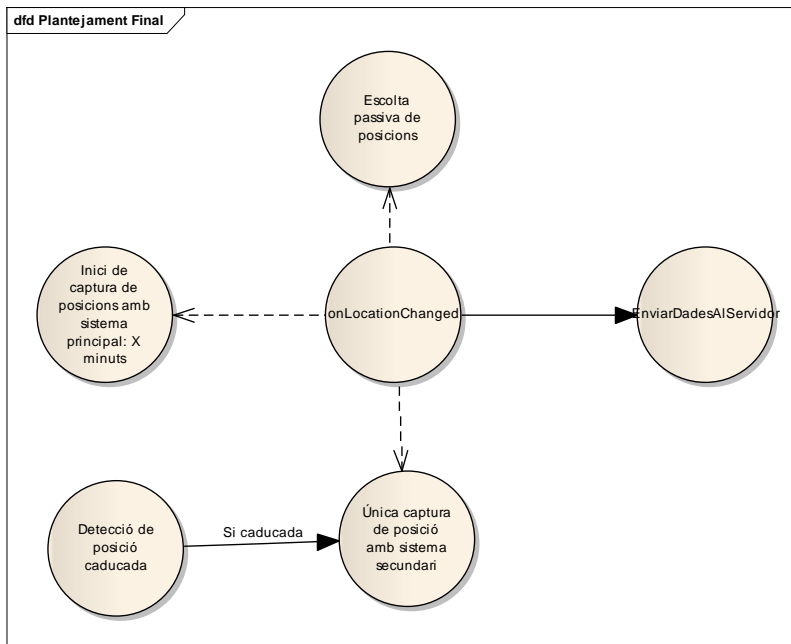
Es replanteja el fet d’utilitzar només posicionament per GPS. S’incorporen els següents canvis:

- Ús del posicionament passiu de forma permanent (no afecta al rendiment de la bateria). De forma que si alguna altra aplicació sol·licita posicionament, aquesta informació es capturarà i s’utilitzarà.
- S’incorpora una opció que permet a l’usuari escollir entre posicionament per GPS, “Servei d’ubicació de Google” (xarxes mòbils i I-FI) o mixt. En el sistema mixt, hi ha un sistema de captura principal i un secundari i es permet escollir entre GPS(principal) + “Servei d’ubicació de Google” (xarxes mòbils i I-FI) (secundari) o “Servei d’ubicació de Google” (xarxes mòbils i I-FI) (principal) + GPS(secundari).

S’utilitzen les mateixes classes que en segon plantejament, però amb implementacions per GPS, “Servei d’ubicació de Google” i Escolta passiva.

La principal millora respecte al segon plantejament (a banda de deixar que l'usuari pugui decidir el sistema de posicionament), és el sistema mixt. Bàsicament, s'afegeix una funcionalitat addicional de verificació de caducitat de l'última posició adquirida. Si aquesta té una hora de captura de més de 150% del temps entre captures, es força un únic posicionament amb el sistema alternatiu. És a dir, el sistema posiciona sempre amb GPS, i si detecta que l'última posició està "caducada", força un posicionament per Xarxes Mòbils, o viceversa (a discreció de l'usuari). El sistema de verificació de caducitat de l'última posició es realitza amb una implementació de les classes *Timer* i *Timertask*, ja explicades en el primer plantejament.

L'esquema queda així:



El motiu d'aquesta lògica de funcionament són:

- Afegir l'escolta passiva no afectarà al rendiment i pot proporcionar informació de posicionament addicional
- Poder escollir entre GPS, "Servei d'ubicació de Google" o sistema mixt, permetrà establir un banc de proves, recollir-ne els resultats, extreure'n conclusions i recomanar un ús concret a l'usuari. Cal mencionar que, segons la documentació d' *Android* SDK, el posicionament per GPS consumeix més bateria i és més precís (error de pocs metres), que el posicionament per "Servei d'ubicació de Google", que és molt menys precís (error de centenars de metres) i consumeix menys bateria.
- Tot i que la solució del segon plantejament era tolerant a fallades de recepció del GPS, el sistema mixt incorpora una millora en aquest aspecte, ja que si el sistema no és capaç de posicionar amb el sistema principal (per problemes de cobertura, per exemple), realitza un posicionament amb el sistema secundari.

Finalment cal indicar que no s'està contradient el principi de limitació de proveïdors de localització, ja que en un mateix moment, només n'hi ha un de funcionant.

8.7.4.4. Captura d'informació de dispositiu

Abans d'abordar l'enviament de dades al servidor ArcGIS, cal especificar el mètode seguit per incorporar la resta de dades definides en els requeriments.

IMSI

Per aconseguir el número d'identificació de la targeta SIM del *Smartphone* (IMSI), s'ha realitzat una implementació d'una funció en una nova classe anomenada "DeviceInformationRetriever".

S’ha utilitzat la funció “getSystemService” de la classe *Context*. A través d’aquesta funció, es té accés a un servei a nivell de sistema. La funció

```
public String getIMSI(){
    if(iMSI==null){
        String serviceName = Context.TELEPHONY_SERVICE;
        TelephonyManager m_telephonyManager =
            (TelephonyManager)mContext.getSystemService(serviceName);
        iMSI = m_telephonyManager.getSubscriberId();
    }
    return iMSI;
}
```

Pel cas de IMEI:

```
public String getIMEI() {
    if(iMEI==null){
        String serviceName = Context.TELEPHONY_SERVICE;
        TelephonyManager m_telephonyManager =
            (TelephonyManager)mContext.getSystemService(serviceName);
        iMEI = m_telephonyManager.getDeviceId();
    }
    return iMEI;
}
```

Estat de la Bateria

Per conèixer el nivell de bateria disponible en cada moment, l’*Android* SDK no deixa realitzar peticions actives. En comptes d’això, en la documentació s’indica que cal subscriure’s a l’escolta d’uns events determinats de sistema, que en aquest cas són els d’estat de bateria. La implementació d’això s’ha realitzat sobre una nova classe anomenada “BatteryInformation”, que és instanciada des de la classe “DeviceInformationRetriever”. La “BatteryInformation” hereta de la classe *BroadcastReceiver*, que és la que ens permet sobreescrivre la funció “onReceive”, per a realitzar l’acció que ens convingui.

Per realitzar la subscripció, en el constructor es crida al mètode “registerReceiver” de la següent forma:

```
public BatteryInformation (Context context){
    IntentFilter batteryLevelFilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
    cw = new ContextWrapper(context);
    cw.registerReceiver(this, batteryLevelFilter);
}
```

La sobre escriptura de la funció “onReceive” és la següent:

```
@Override
public void onReceive(Context context, Intent intent) {
    int rawlevel = intent.getIntExtra("level", -1);
    int scale = intent.getIntExtra("scale", -1);
    BatteryInformation.batteryLevelPercentage = -1;
    if (rawlevel >= 0 && scale > 0) {
        batteryLevelPercentage = (rawlevel * 100) / scale;
    }
}
```

On “batteryLevelPercentage” és una variable global de la classe de tipus enter. Per tant, des de la classe “DeviceInformationRetriever” només cal instanciar el “getter” de “batteryLevelPercentage” pes esbrinar el nivell de bateria en cada moment (recordem que el sistema avisa de tot canvi en el nivell de bateria i aquests són capturats per la funció “onReceive”).

8.7.4.5. Enviament de dades al servidor ArcGIS

L'aplicació envia les dades al servidor ArcGIS. Aquest enviament el realitza al servei REST del servidor mencionat anteriorment en el punt 8.5.3.4 Servei d'addició de punts nous.

En l'aplicació, de l'event "onLocationChanged", en surt un objecte de tipus *Location*. Aquests tipus d'objectes, a banda de la informació de posicionament, contenen també la informació referent al dia i hora en què s'ha realitzat la captura de la posició.

El nom de l'equip es guarda emmagatzemat en una variable de "TWService", mentre que el nivell de bateria i IMSI es recuperen a través de la classe "DeviceInformationRetriever" (com ja hem vist).

S'ha creat un tipus de classe anomenat "LocationInformation", que és una classe contenidora de tota aquesta informació. A més, hi ha implementat un mètode anomenat "toJSONObject" que torna un tipus "JSONObject". Els tipus "JSONObject" són classes les quals el seu mètode "toString" torna una cadena amb format JSON, que és el format que el servei d' ArcGIS té definit per rebre. La definició de la cadena JSON s'especifica més endavant, en l'apartat de Serveis ArcGIS.

En la implementació "onLocationChanged", es feia una crida a la funció "storeLocationInformation". A continuació s'observa la implementació d'aquesta funció:

```
private void storeLocationInformation(Location loc){
    LocationInformation locI = new LocationInformation(loc, "102100",
        deviceInformationRetriever.getIMSI(), team,
        deviceInformationRetriever.getBatteryLevelPercentage());
    locationInformationManager.storeAndSendLocationToServer(locI);
}
```

En la primera línia es crea una nova instància de "LocationInformation", que contindrà les dades definides anteriorment. Posteriorment, es passa aquesta nova instància a una funció "storeAndSendLocationToServer", d'una classe anomenada "LocationInformationManager". Aquesta classe és l'encarregada de gestionar la informació capturada.

La funció "storeAndSendLocationToServer", realitza les següents passes:

- Primer de tot l'emmagatzema en una cua prioritària: `PriorityQueue<LocationInformation>`, de forma que la primera posició sempre l'ocuparà la `LocationInformation` més antiga.
- Inicia un bucle per totes les "LocationInformation" contingudes en la cua prioritària. Si la situació és normal, només n'hi hauria d'haver una (la afegida en el punt anterior). A cada passada de bucle, es treu la primera posició i es fa una petició HTTP contra el servidor, passant-li el JSON sortint de la `LocationInformation`. Si la petició és exitosa, vol dir que les dades han arribat al servidor ArcGIS. Si no és així, es captura una excepció i es torna a afegir el `LocationInformation` que no s'ha pogut enviar a la cua prioritària. Tornarà a ocupar el primer lloc ja que és una cua ordenada per temps de captura de localització.
- Si la petició HTTP contra el servidor ArcGIS no és exitosa, es surt del bucle, perquè el sistema interpretarà que no hi ha connectivitat amb el servidor. El proper cop que la funció sigui cridada, es tornarà a intentar buidar tota la cua prioritària. Això succeirà amb la propera localització nova. La petició HTTP es realitza mitjançant una instància de la classe `HTTPClient` que utilitza una instància de la classe `HttpPost`.

Per tal de que l'usuari se n'assabenti de les accions descrites en aquest punt (posicions enviades, pendents d'enviar i capturades), des de "LocationInformationManager" es llança un event contra tots els subscriptors d'aquest event (la classe "TWService"), de forma que se li fa saber que hi ha hagut canvis. Quan el `TWService` rep l'event, utilitza els "getters" pertinents del "LocationInformationManager" per obtenir la informació. Posteriorment el "TWService" li fa saber a la interfície d'usuari per a que actualitzi els valors de pantalla.

La informació es visualitza per pantalla de la següent forma (veure Il·lustració 9: Visualització informació per pantalla):



Il·lustració 9: Visualització informació per pantalla

8.7.4.6. Altres característiques

L'aplicació també té altres característiques com són les que es detallen a continuació.

Iniciar / Aturar captura de posicions

L'aplicació posa a disposició de l'usuari un botó mitjançant el qual es pot aturar i engegar la captura de posicions. Tal i com s'ha comentat anteriorment, el sistema és capaç de detectar si el servei configurat per realitzar captures està actiu o no. A continuació es detalla aquest procés:

L'aplicació no està capturant posicions. El botó mostra el missatge "OFF".



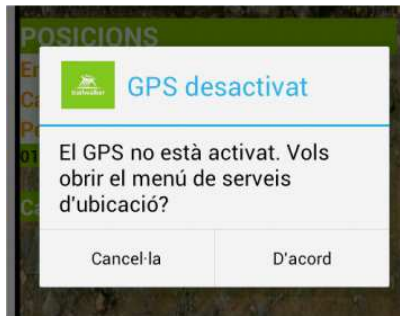
Il·lustració 10: Botó captura posicions - OFF

Si l'usuari el prem, el sistema inicia la captura de posicions i passa a mostrar el missatge "ON".



Il·lustració 11: Botó captura posicions - ON

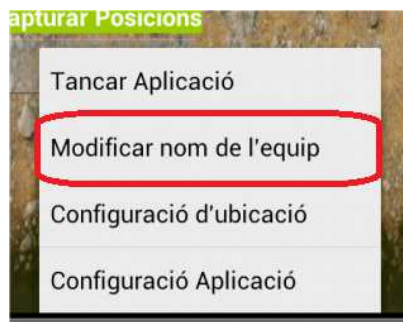
Si el servei de posicionament utilitzat està desactivat o es desactiva durant la captura de posicions, es mostra el següent missatge (exemple pel cas d'ús de GPS):



Il·lustració 12: Captura de posicions - GPS no activat

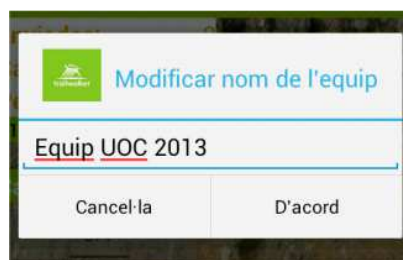
Canvi de nom

L'aplicació permet a l'usuari especificar el nom d'equip, de forma que els punts enviats al servidor ArcGIS estiguin degudament identificats. Prement el botó "menú" del dispositiu s'accedeix a un llistat d'opcions. Una d'elles és "Modificar nom de l'equip":



Il·lustració 13: Modificació de nom de l'equip (menú)

Un cop es prem el menú corresponent, apareix la següent caixa de diàleg:



Il·lustració 14: Modificació de nom de l'equip

On es pot especificar el nou nom.

La interfície d'usuari d'aquesta funcionalitat està gestionada per la classe *TeamSettingsDialog*.

Cal destacar que aquesta informació es persisteix en el sistema, amb el que és tolerant a apagades de l'aplicació o el dispositiu. La persistència la gestiona la classe *TWApplicationSetingsManager*.

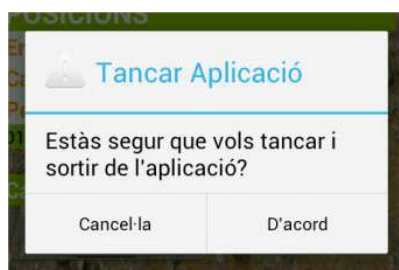
Tancament de l'aplicació

Per tancar l'aplicació, s'ha de realitzar a través de l'opció del menú "Tancar Aplicació", tal i com es mostra a la següent figura:



Il·lustració 15: Tancament aplicació (menú)

Posteriorment es mostra una caixa de diàleg en que es demana la confirmació a l'usuari, tal i com es mostra a continuació:



Il·lustració 16: Tancament aplicació

Si la resposta és afirmativa, es tanca l'aplicació.

La interfície d'usuari d'aquesta funcionalitat està gestionada per la classe *ShutdownApplicationDialog*.

Canvi de configuració de l'aplicació

L'aplicació permet a l'usuari modificar certs paràmetres de funcionament de l'aplicació. Aquesta funcionalitat està destinada sobretot a realització de proves per usuaris experts, ja que una configuració incorrecta pot provocar efectes adversos com consum excessiu de bateria o mal funcionament de l'aplicació.

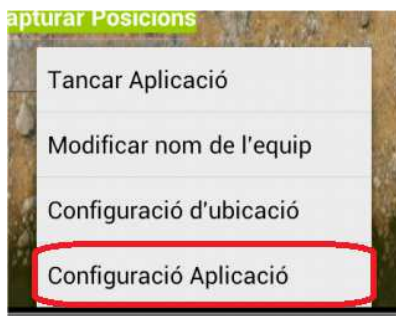
Cal destacar que aquesta informació es persisteix en el sistema, amb el que és tolerant a apagades de l'aplicació o el dispositiu. La persistència la gestiona la classe *TWApplicationSettingsManager*.

Els paràmetres que permet modificar són:

- Temps entre captures: Temps entre captures de la posició (600 minuts per defecte).
- Distància entre captures: Distància mínima que el sistema té en consideració per definir que dues posicions són diferents (5 metres per defecte).
- URL API REST: Direcció URL de l'API REST del server ArcGIS on s'envien els nous punts.
La URL per defecte és:
<http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer/0/addFeatures>
- Sistema de posicionament: Sistema que el dispositiu utilitzarà per realitzar el posicionament
 - o GPS: Posicionament exclusiu per GPS.
 - o Xarxes mòbils: Posicionament exclusiu per "Servei d'ubicació de Google".
 - o GPS + Xarxes mòbils: Valor per defecte. Sistema mixt amb GPS com a primari i "Servei d'ubicació de Google" com a secundari
 - o Xarxes mòbils + GPS: Sistema mixta amb "Servei d'ubicació de Google" com a primari i GPS com a secundari.

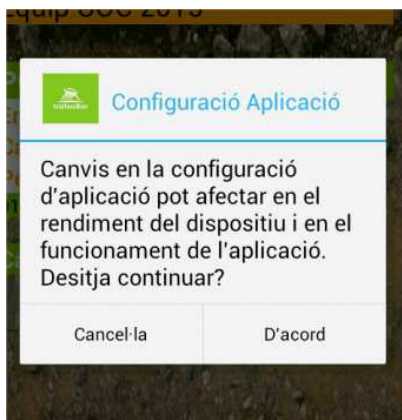
També permet establir la configuració per defecte.

Per accedir-hi cal prémer l'opció "Configuració Aplicació" del menú.



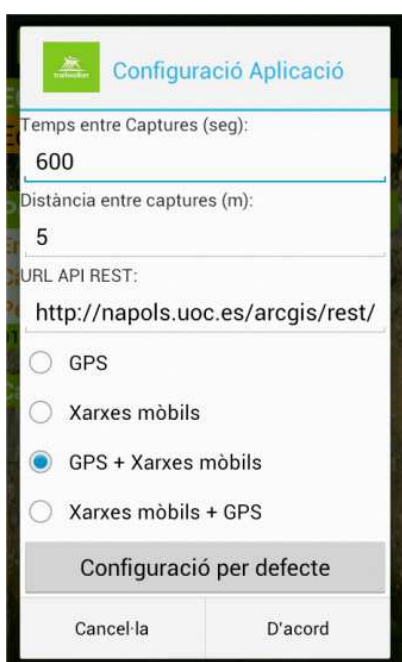
Il·lustració 17: Configuració aplicació (menú)

Posteriorment s'avis a l'usuari del perill de manipular la configuració de l'aplicació:



Il·lustració 18: Configuració aplicació (avís)

Si l'usuari prem "D'acord", s'accedeix a la pantalla per gestionar la configuració:



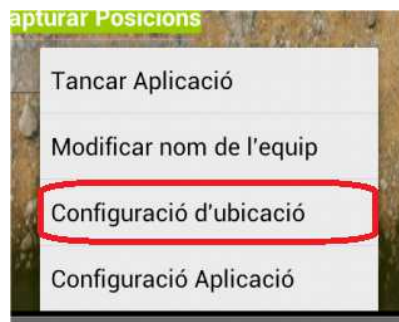
Il·lustració 19: Configuració aplicació

D'es d'on es pot especificar la configuració de l'aplicació.

La interfície d'usuari d'aquesta funcionalitat està gestionada per les classes *ApplicationSettingsDialogIni* i *ApplicationSettingsDialog*. Un cop acceptada la caixa de diàleg, es passa a persistir la nova informació, i se li passa a la instància de la classe *TWService* per a que tingui els nous valors pel funcionament de l'aplicació.

Accés als sistemes de posicionament

Per comoditat, s'ha afegit un menú per accedir a la pantalla d' *Android* de "Serveis d'ubicació" des de l'aplicació. Per accedir-hi, cal prémer el menú "Configuració d'ubicació":



Il·lustració 20: Configuració d'ubicació (menú)

Apareix la pantalla de configuració esmentada:



Il·lustració 21: Serveis d'ubicació

Aplicació Multi Idioma

Android SDK agrupa les cadenes a mostrar per la interfície gràfica en un arxiu anomenat "strings.xml", dins de la carpeta /res/values/ de l'estructura de carpetes de l'aplicació en l'entorn de desenvolupament. *Android* SDK ja està preparat pel multi Idioma de forma que l'aplicació per defecte buscarà la carpeta /res/values-xx on "xx" és el codi ISO de l'idioma en què està configurat el sistema operatiu. Per tant, si tenim el sistema operatiu en anglès, i existeix una carpeta anomenada /res/values-en, buscarà l'arxiu "strings.xml" dins d'aquesta carpeta preferentment. Si no existeix, busca la carpeta per defecte, que és /res/values/. Per tant, des del punt de vista de programació, només hem de traduir les cadenes contingudes en "strings.xml" per cadascuna de les carpetes d'idioma que es vulgui tenir per l'aplicació. Cal destacar que el canvi d'idioma el fa en temps

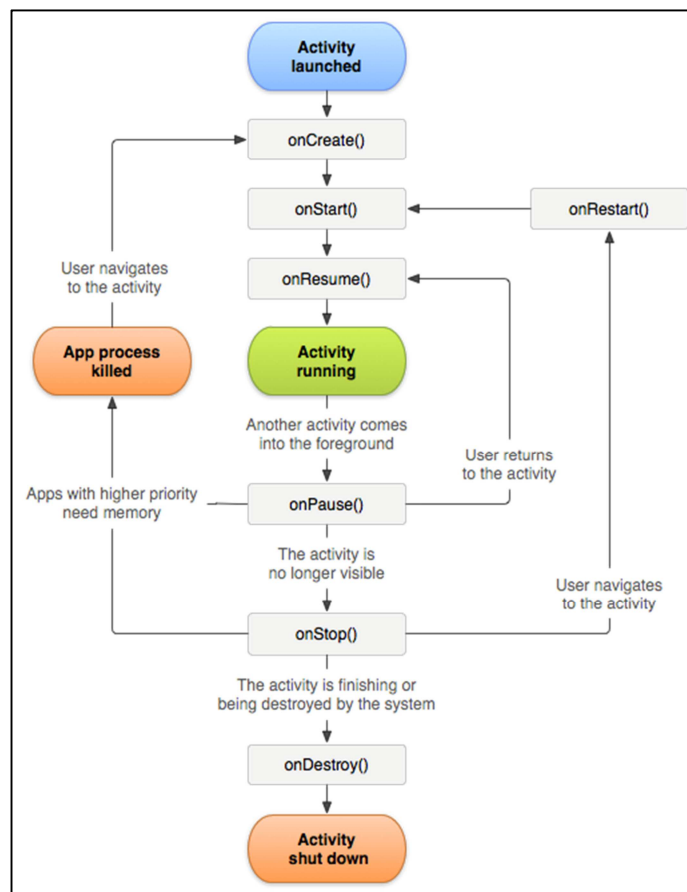
d'execució, pel que per canviar l'idioma amb l'aplicació executant-se, només cal canviar l'idioma del sistema operatiu *Android*.

A tall d'exemple es mostra un fragment de l'arxiu "strings.xml" del nostre projecte:

```
<resources xmlns:Android="http://schemas.Android.com/apk/res/Android">
  <string name="app_name">TrailWalker Track</string>
  <string name="menu_configuracio_ubicacio">Configuració d\'ubicació</string>
  <string name="title_activity_trail_walker_track">TrailWalkerTrack</string>
  <string name="toggleButtonGPSServiceON">ON</string>
  <string name="toggleButtonGPSServiceOFF">OFF</string>
  <string name="posicions_capturades">Capturades: </string>
  <string name="posicions_enviades">Enviades: </string>
  <string name="posicions_pendents_enviar">Pendents Enviar: </string>
  <string name="numero_zero">0</string>
  <string name="posicions">POSICIONS</string>
  <string name="tancar_aplicacio">Tancar Aplicació</string>
  ...
</resources>
```

Aplicació Funcionant en Segon Pla

Per explicar aquest punt, es farà referència al cicle de vida de les classes *Activity*. En la Il·lustració 22 es mostra el cicle de vida mencionat:



Il·lustració 22: Cicle de vida d'una *Activity* (Android Developer - SDK - Activity class)

Quan l'aplicació s'està executant en primer pla, l'*Activity* està en l'estat "Activity Running". Si l'usuari prem el botó "Back" del dispositiu, el que provoca és que l'aplicació faci el següent:

`onPause() → onStop() → onDestroy()`

Aquest procés és irreversible, i l'aplicació queda destruïda. En el nostre cas, voldria dir que l'aplicació deixaria de capturar i enviar posicions en quant l'usuari fes aquesta acció. El *Service*, també s'elimina, ja que l'*Activity* que l'ha invocat es destrueix.

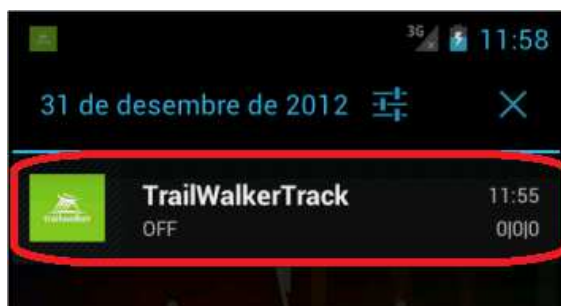
Per evitar això només cal sobreescriure el mètode “onKeyDown” (al prémer una tecla), de forma que controlem què ha de fer l'aplicació quan es prem el botó “Back”. La millor opció és portar l'aplicació a un segon pla, però que continuï executant-se. Per realitzar aquesta acció, es fa una crida a la funció de sistema “moveTaskToBack”. El codi queda de la següent manera:

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    boolean b = false;
    //Si la tecla premuda és "Enrere"...
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        moveTaskToBack(true);
        b = true;
    }
    else b = super.onKeyDown(keyCode, event);
    return b;
}
```

Al passar l'aplicació en segon pla, l'*Activity* passa a estat onPause() i s'hi queda. En punt, a través del sistema de notificació d' *Android* s'informa a l'usuari que l'aplicació continua funcionant, i la pot restaurar fent un clic en la notificació (veure Il·lustració 23: Notificació de sistema de l'aplicació i Il·lustració 24: Notificació de Sistema i Barra d'estat), on torna a l'estat onResume():



Il·lustració 23: Notificació de sistema de l'aplicació



Il·lustració 24: Notificació de Sistema i Barra d'estat

En la barra de notificació s'hi pot observar un resum de l'estat de l'aplicació: Mostra ON/OFF en funció de si l'aplicació està capturant posicions o no, respectivament. També s'hi indica una notació del tipus “X|Y|Z”, on X són el nombre de posicions enviades al servidor GIS, Y són el nombre de posicions capturades i Z són el nombre de posicions capturades i no enviades. Finalment també s'hi indica l'hora en què s'ha actualitzat la informació.

9. BANC DE PROVES

Un cop implementada l'aplicació, s'ha dissenyat i executat un banc de proves per comprovar la viabilitat de l'aplicació *Android* com a dispositiu emissor de posicions pels marxadors, així com per la determinació de recomanacions d'ús entre els diferents tipus de serveis a utilitzar (GPS i "Servei d'ubicació de Google"). El banc de proves va dirigit concretament al punt feble més important d'aquesta solució que és l'autonomia del dispositiu.

En l'execució de les proves s'ha utilitzat un *Smartphone* Samsung SIII (model GT-I9300) amb *Android* versió 4.1.1. En totes les proves s'han seguit les millors pràctiques viables definides en el següent punt (9.1).

9.1. Millors pràctiques en l'estalvi de consum de bateria en Android

El consum de bateria en els *Smartphones* és totalment variable, i ni tant sols els fabricants en donen dades concretes. Això és sobretot degut a que són dispositius oberts i l'usuari en pot fer diferents usos, que provocaran diferents consums. Malgrat això, sí que existeixen una sèrie de recomanacions per evitar al màxim el consum de bateria. A continuació es detalla una llista d'accions a realitzar per minimitzar el consum de bateria, i addicionalment es descriu si són viables pel banc de proves d'aquest projecte i si s'han utilitzat en el banc de proves:

Recomanació	Viable	Utilitzada
Desactivar Sincronització	x	x
Desactivar la pantalla (no manipular el telèfon)	x	x
Disminuir l'activitat d'usuari	x	x
Desactivar el I-Fi	x	x
Desactivar el Bluetooth	x	x
Desactivar el NFC (Near Field Communication)	x	x
Eliminar widgets innecessaris	x	x
Disminuir la brillantor de la pantalla	x	
Desactivar la rotació automàtica	x	x
Desactivar la vibració	x	
Desactivar GPS		
Desactivar 2G/3G		
Disminuir al mínim el temps d'apagat automàtic de pantalla	x	x
Activar "Estalvi d'Energia"	x	x
Desactivar "Hàptic Automàtic"	x	

^oTaula 4: Millors pràctiques

9.2. Descripció de banc de proves i resultats

S'han definit 2 tipus de proves, una prova d'estrès per accentuar diferències de consum de bateria entre l'ús de GPS i l'ús de "Servei d'ubicació de Google" en el posicionament, i una prova de condicions de cursa, per veure l'impacte real de l'ús de l'aplicació. En tots dos casos s'ha realitzat una mesura del consum de bateria del dispositiu sense ús per poder realitzar la comparació. A continuació es mostra una taula amb les mesures realitzades i les condicions en què s'han realitzat:

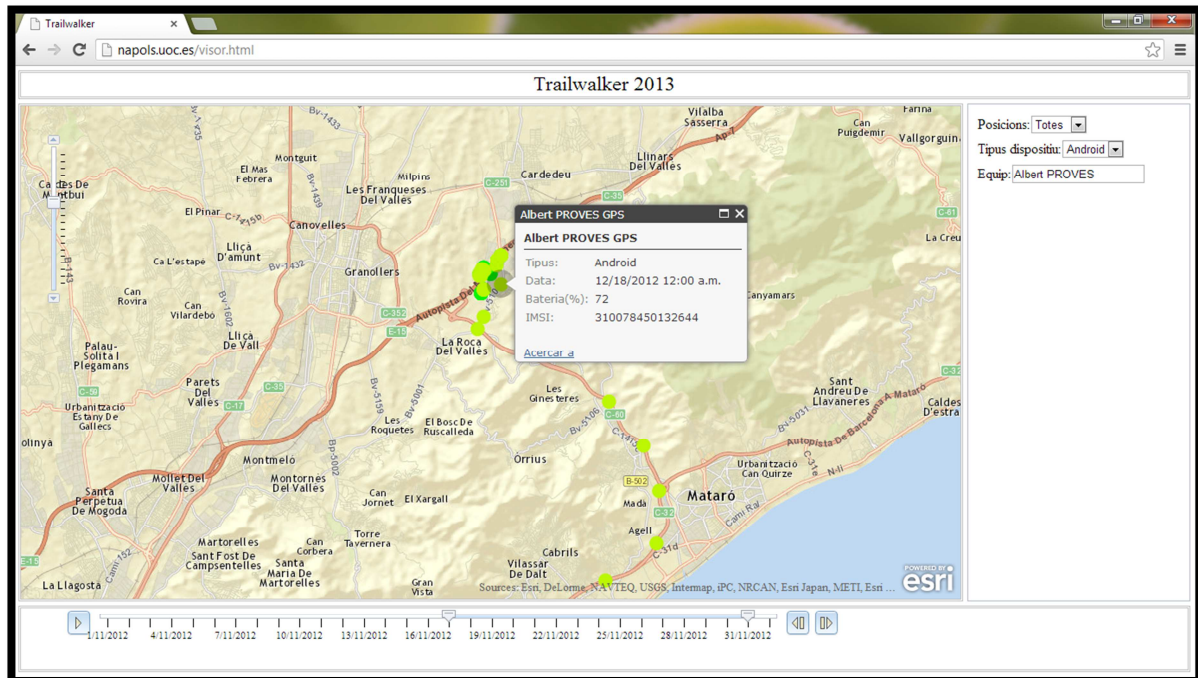
Prova	Sistema de Captura	Temps total de prova (min)	Temps entre captures	Nombre de posicions capturades	Consum de Bateria
Estrès	Sense captures	30	n/a	n/a	1%
Estrès	GPS	30	1 minut	30	3%
Estrès	“Servei d’ubicació de Google”	30	1 minut	31	3%
Condicions cursa	Sense captures	60	n/a	n/a	1%
Condicions cursa	GPS	60	10 minuts	7	1%
Condicions cursa	“Servei d’ubicació de Google”	60	10 minuts	10	1%

Taula 5: Banc de Proves - Mesures realitzades

Com es pot observar, en algunes de les proves s’han realitzat més captures de les programades. Cal recordar que en tot moment està funcionant l’escolta passiva. Aquestes captures de més són degudes a aquest fet.

Per últim cal indicar que en el transcurs del banc de proves s’han executat una sèrie de verificacions per confirmar que l’aplicació funciona com ha estat pensada. De forma general s’ha constatat que l’aplicació funciona tal i com s’ha dissenyat, i que els diferents sistemes de posicionament funcionen de la forma esperada.

S’ha pogut confirmar el funcionament de tot el sistema verificant que les posicions capturades finalment són mostrades a través del servidor ArcGIS. En el banc de proves s’ha accedit a la pàgina de mostra construïda: Aquesta pàgina servida utilitza les dades del servidor ArcGIS Trailwalker especificat en aquest projecte. A continuació es mostra una imatge d’exemple:



Il·lustració 25: Imatge posicions Banc de Proves

10. CONCLUSIONS

En aquest apartat es descriuen les conclusions que s'extreuen de l'execució del projecte així com de les millores observades durant el procés.

Respecte al banc de proves, i més concretament a les proves d'estrès, es verifica que l'ús de l'aplicació, és a dir, ús del posicionament tant per GPS com per "Servei d'ubicació de Google" té un efecte negatiu sobre el consum de bateria. En canvi, no s'observen diferències entre l'ús del GPS o de "Servei d'ubicació de Google", ja que tots dos provoquen un mateix consum de bateria en 30 minuts de funcionament (un 3%, respecte a l'1% en l'ús normal del mòbil).

Quan a les proves en condicions de cursa, se n'extreu que l'ús de posicionament per GPS i "Servei d'ubicació de Google" no ha afectat al rendiment normal del dispositiu, ja que en totes 3 mesures s'ha consumit un 1% de bateria durant l'hora en què s'han realitzat les proves.

Tot i que hi ha moltes referències (per exemple la documentació d' *Android* SDK) a un gran consum de bateria en l'ús de GPS integrat respecte a posicionament per "Servei d'ubicació de Google", en el nostre marc de proves això no es dona. L'explicació ve donada per un matís en la documentació i en les referències. Quan indiquen que l'ús del GPS integrat en els *Smartphone* augmenta considerablement el consum de bateria, es fa referència a un ús permanent o molt freqüent del mateix, a on sí que es fa molt evident aquest fet. En el nostre cas, en canvi, l'ús és molt reduït (6 posicions per hora), amb el que els efectes negatius són difícilment observables.

La decisió de realitzar la implementació d'ús de "Servei d'ubicació de Google" pel posicionament per l'aplicació per *Android* va venir impulsada pel fet que s'ha comentat en el paràgraf anterior, amb el que s'intentava donar una alternativa al posicionament per GPS, per minimitzar el consum de bateria. Per contrapartida, el posicionament per "Servei d'ubicació de Google" tenia una precisió molt baixa (centenars de metres) respecte al GPS (pocs metres). Finalment, com que realment no hi ha una diferència de consum de bateria entre ús del GPS o "Servei d'ubicació de Google", no es veu justificat l'ús de "Servei d'ubicació de Google" a causa de la reducció de precisió.

Respecte a l'ús del sistema de posicionament mixt, queda descartat el de tipus "Servei d'ubicació de Google" (primari) + GPS(secundari), pels motius que acabem de descriure, i en canvi, queda reforçat l'ús del sistema mixt GPS (primari) + Xarxes mòbils (secundari), ja que el posicionament es farà sempre per GPS i només en aquells casos en què hi hagi una pèrdua de cobertura de senyal GPS, s'utilitzarà puntualment el posicionament per "Servei d'ubicació de Google".

Tot i que la implementació i proves d'un receptor GPS extern per *Bluetooth* no ha estat inclòs en el projecte, amb els resultats obtinguts es pot afirmar que el seu ús en la solució final no és una bona opció. L'explicació ve donada sobretot pel fet que el posicionament experimentat en la prova de condicions de cursa ha provocat un consum de bateria nul o menyspreable respecte al consum normal del *Smartphone*. Per tant no es veu justificat l'ús del GPS per *Bluetooth*, ja que aquest implicaria un sobre cost i una càrrega innecessaris als marxadors.

Respecte a l'aplicació *Android* es pot concloure que, tot i que en actualment cap *Smartphone* no és capaç d'estar en funcionament 32 hores (durada màxima de la cursa) de forma ininterrompuda per motius de capacitat de bateria, la solució és viable. La viabilitat ve donada pel fet que els equips de marxadors compten amb un equip de suport. És factible que els equips de suport tinguin *Smartphones Android* addicionals de recanvi pels marxadors, quan el seu esgoti la bateria (si és que els propis marxadors no els porten a sobre).

Com ja s'ha comentat, no és possible determinar la durada de la bateria d'un *Smartphone*, malgrat això, hi ha iniciatives que intenten fer una recopilació d'aquestes dades. Una d'elles és a : <http://www.rootinstaller.com/2012/12/battery-statistic-on-android-powered.html> . Com s'hi indica, a Desembre del 2012 la mitjana de durada de la bateria és de 20,4h pels dispositius mesurats (474 dispositius *Android* diferents, amb 5585 mesures). Es podria considerar un valor més conservador de 11h pel nostre cas, per tal de tenir un major ventall de dispositius en el rang. Si es té en compte aquest valor, i el fet que l'aplicació

desenvolupada en el marc d'aquest projecte no afecta al consum de bateria, voldria dir que molts equips només necessitarien 1 o 2 dispositius de recanvi, fet totalment acceptable.

Respecte al projecte cal mencionar que la proposta inicial era força ambiciosa. Sí que s'ha aconseguit l'objectiu principal, quant a donar solució als requeriments de la cursa Trailwalker d'Intermon Oxfam. Tot i això, no s'han pogut assolir tots els objectius i tasques plantejats en el marc del Treball de Fi de Carrera. S'ha realitzat una anàlisi d'opcions tecnològiques i se n'ha triat la més idònia i definit l'arquitectura, de la qual també se n'ha desenvolupat la part client (*Android*). Han quedat exclosos del projecte la programació i configuració del servidor GIS, i la implementació de la part web ha quedat reduïda a una demostració. Centrant-nos en els objectius definits a l'inici del projecte s'ha aconseguit:

- Plantejar i raonar els problemes de mobilitat, posicionament y autonomia dels dispositius mòbils i les seves capacitats de GEO posicionament.
- Conèixer les característiques i conceptes de la tecnologia dels sistemes de GEO posicionament i d'informació geogràfica.
- Familiaritzar-se amb els objectes específics de la programació per a dispositius mòbils.
- Analitzar de viabilitat de diverses solucions sota limitacions tant econòmiques com de d'autonomia.
- Justificar la solució proposada sota les condicions descrites.
- Aprendre a utilitzar el posicionament GPS d'un mòbil i saber localitzar l'usuari .
- Desenvolupar una aplicació o simulació que corrobore la viabilitat de la solució.
- Familiaritzar-se amb els objectes específics de la programació web.

I no s'han assolit:

- Saber gestionar dades geogràfiques, com pot ser la gestió de mapes o publicació de serveis de mapes.

Queda pendent per a propers projectes la definició de la part web i de la part del servidor GIS (ArcGIS).

10.1. Millores

Tal i com s'ha comentat, l'aplicació *Android* que s'ha desenvolupat en el marc d'aquest projecte, té el paper principal de capturar informació i enviar-la a al servidor ArcGIS. Respecte al punt de vista dels marxadors, una millora substancial seria oferir la informació de posicionament de tots els equips de la cursa en la mateixa aplicació, a través d'un visualitzador de mapes i punts integrat. Una eina adient per dur a terme aquesta tasca és l'ArcGis per *Android* (<http://resources.arcgis.com/es/content/arcgis-android/about>), malgrat que cal destacar que l'API encara està en una versió beta.

Les bones pràctiques per un menor consum de la bateria del telèfon, no deixen de ser recomanacions que estan a criteri de l'usuari. Moltes d'elles són accions que es poden encapsular en una aplicació, de forma que es puguin activar a l'engegar l'aplicació (previ avís a l'usuari). Per tant, aquest seria un punt a investigar de cara a futures versions de l'aplicació *Android*.

Cal destacar que la seguretat no s'ha tingut en la definició i execució del projecte. Com a millora es proposa desenvolupar un sistema per evitar frau en la identificació d'equips (que cap equip es pugui fer passar per cap altre). També caldria desenvolupar un sistema per evitar l'enviament de dades falses, ja que tal i com està plantejat el sistema actualment, el servei REST està totalment obert, i per enginyeria inversa és relativament fàcil utilitzar-lo fora de l'àmbit de l'aplicació *Android*.

Per últim, es recorda que *Android* és la plataforma mòbil utilitzada en aquest projecte. Per fer que la solució global estigui a l'abast de més públic, caldria fer l'aplicació disponible per altres plataformes mòbils.

11. BIBLIOGRAFIA

- Android*. (sense data). Consultat el Desembre / 2012, a <http://www.android.com>
- Android Developer - Location Strategies*. (sense data). Recollit de Android Developer - Location Strategies: <http://developer.android.com/guide/topics/location/strategies.html>
- Android Developer - SDK*. (sense data). Consultat el Novembre i Desembre / 2012, a <http://developer.android.com/sdk/index.html>
- Android Developer - SDK - Activity class*. (sense data). Consultat el 11 / 2012, a <http://developer.android.com/reference/android/app/Activity.html>
- ArcGIS - ESRI*. (sense data). Consultat el Novembre / 2012, a <http://www.esri.com/software/arcgis>
- ArcGIS API for JavaScript - ArcGIS Resource Center*. (sense data). Consultat el Desembre / 2012, a <http://help.arcgis.com/en/webapi/javascript/arcgis/>
- Arduino*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.Arduino.cc/>
- Campderrich Falgueras, B. (2004). *Enginyeria del programari - UOC*. Barcelona.
- CODE PROJECT*. (sense data). Consultat el Novembre i Desembre / 2012, a <http://www.codeproject.com/Articles/430014/N-Tier-Architecture-and-Tips>
- developer.android.com*. (sense data). Consultat el 12 / 2012, a <http://developer.android.com/reference/android/app/Activity.html>
- Endomondo*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.endomondo.com/>
- Global AVL*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.globalavl.eu/>
- Intermón Oxfam trailwalker*. (sense data). Consultat el Octubre, Novembre i Desembre / 2012, a <http://trailwalker.intermonoxfam.org>
- Introducing JSON*. (sense data). Consultat el 11 / 2012, a <http://www.json.org/>
- Map My Tracks*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.mapmytracks.com>
- My Tracks - Google Mobile*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.google.com/mobile/mytracks>
- OMG*. (sense data). Consultat el Desembre / 2012, a <http://www.omg.org/spec/UML/2.1.2/>
- Open Handset Alliance*. (sense data). Consultat el Novembre i Desembre / 2012, a <http://www.openhandsetalliance.com>
- Pérez Navarro, A., Bataller Díaz, A., Beneito Montagut, R., Sáenz Higuera, N., & Vidal Oltra, R. (2008). *Treball final de carrera - UOC*. Barcelona: Eureka Media, SL.
- Pérez Navarro, A., Botella Plana, A., Muñoz Bolas, A., Olivella, R., Olmedillas Hernández, J., & Rodríguez Lloret, J. (2009). *Sistems d'informació geogràfica i geotelemàtica - UOC*. Barcelona: Eureka Media, SL.
- Root Uninstaller Team*. (sense data). Consultat el Desembre / 2012, a <http://www.rootuninstaller.com>
- Runtastic*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.runtastic.com>
- Wikipedia - Android*. (sense data). Consultat el 06 / 12 / 2012, a [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- Wikipedia - Bluetooth*. (sense data). Consultat el 11 / 2012, a <http://ca.wikipedia.org/wiki/Bluetooth>
- Wikipedia REST (Representational State Transfer)*. (sense data). Consultat el 11 / 2012, a http://es.wikipedia.org/wiki/Representational_State_Transfer
- Zoombak - Personal GPS Locators*. (sense data). Consultat el Octubre i Novembre / 2012, a <http://www.zoombak.com/products/universal/>

12. GLOSSARI

2G: El concepte 2G fa referència a la segona generació de la tecnologia de telefonia mòbil.

3G: Fa referència a la tercera generació de transmissió de dades i veu a través de telefonia mòbil.

API: (Application Programming Interface, en Anglès) Són aquells recursos que ofereix un programari concret per a que altres programaris externs hi interactuïn.

Arduino: És una plataforma electrònica de prototipatge i de programari lliure.

Bluetooth: Especificació industrial per xarxes sense fils d'àrea personal per a la transmissió de veu i dades entre dispositius.

C: Llenguatge de programació creat el 1972 per Dennis M. Ritchie, orientat inicialment a Sistemes Operatius, concretament Unix.

C++: Llenguatge de programació dissenyat a mitjans dels 80 per Bjarne Stroustrup. És un llenguatge fortament lligat al paradigma de programació orientada a objectes o POO (Programació Orientada a Objectes).

Cloud: Fa referència a "Cloud computing", que és una forma de computació a on els recursos de tecnologies de la informació es proporcionen en forma de serveis.

CSS: Ve de l'anglès *Cascading Style Sheets*, És un llenguatge utilitzat per definir la presentació d'un document estructurat en HTML.

Eclipse: És un entorn de desenvolupament integrat de codi obert multi plataforma.

Event: En entorns de computació, és una acció iniciada en un àmbit concret del programari i gestionat per un àmbit diferent.

Getter: Forma com es coneixen a les funcions d'accés al valor de les variables privades de les classes.

GIS: Respon a les sigles *Geographic Information System*. Són sistemes dissenyat per capturar, emmagatzemar, manipular, analitzar, gestionar i mostrar tot tipus de dades geogràfiques.

Google Play: Botiga de programari en línia desenvolupada per Google, pels dispositius *Android*.

GPS: Respon a les sigles *Global Positioning System*. És un sistema de navegació per satèl·lit que permet determinar en tot el món la posició d'un objecte. Va ser desenvolupat, instal·lat i actualment operat pel Departament de Defensa dels Estats Units d'Amèrica.

GSM: És l'acrònim de *Global System for Mobile communications*. És un estàndard lliure de telefonia mòbil digital.

EDGE: És l'acrònim per *Enhanced Data Rates for GSM*. És una tecnologia per telefonia mòbil.

HSDPA: És l'acrònim per *High Speed Downlink Packet Access*. És una tecnologia per internet mòbil, també coneguda 3.5G.

HSPA+: És un estàndard per internet mòbil.

HTML: És l'acrònim de *HiperText Markup Language*. Fa referència al llenguatge predominant utilitzat en la construcció de pàgines web.

HTTP: És l'acrònim de *HiperText Transfer Protocol*. És un protocol utilitzat en les transaccions de la World Wide Web (sistema de distribució d'informació basat en hipertext i accessible a través d'Internet).

HTTPS: És l'acrònim de *HiperText Transfer Protocol Secure*. És la versió segura del protocol HTTP.

Java: Llenguatge de programació inicialment desenvolupat per James Gosling, i publicat el 1995. És un llenguatge lligat al paradigma de programació orientada a objectes.

JavaScript: És un llenguatge de programació interpretat, fortament utilitzat en la construcció de pàgines web.

JSON: És l'acrònim de *JavaScript Object Notation*. És un format lleuger per a l'intercanvi de dades, originàriament pensat per a ser utilitzat en entorns de Javascript.

Mashup: En desenvolupament web, és un pàgina web que utilitza i combina dades, presentacions i funcionalitat procedents d'una o més fonts de serveis.

NDK: És un conjunt d'eines per *Android* que permeten implementar aplicacions utilitzar C i C++.

POO: És l'acrònim per Programació Orientada a Objectes. És un paradigma de programació.

POST: És un dels mètodes de peticions HTTP.

REST: És l'acrònim de *Representational State Transfer*. És una tècnica d'arquitectura de programari per a sistemes de text, imatge, vídeo, so, mapes i altres suports d'informació, distribuïts com la World Wide Web.

SaaS: És l'acrònim de *Software as a Service*. Model de distribució de programari on el programari i les dades estan allotjats en servidors d'una companyia, i els clients hi accedeixen a través d'Internet (habitualment). Lligat al concepte de *Cloud*.

SDK: Respon a l'acrònim *Software Development Kit*. Conjunt d'eines de desenvolupament de programari.

Servei d'ubicació de Google: Servei de posicionament dels dispositius *Android* caracteritzat per utilitzar la triangulació d'antenes de telefonia i xarxes I-Fi.

SIM: Targeta utilitzada en telèfons mòbils que emmagatzema la clau de servei del subscriptor per identificar-se a la xarxa de telefonia.

Tablet: De l'Anglès *tablet computer*, és un tipus d'ordinador portàtil caracteritzat per ser utilitzat a través del contacte amb la pantalla.

Tracking: Paraula anglesa per referir-se al rastreig o posicionament d'una persona o objecte en el temps.

URI: Respon a l'acrònim *Uniform Resource Identifier*. És una cadena de caràcters curta que identifica de forma inequívoca un recurs (servei, pàgina...)

URL: De l'Anglès *Uniform Resource Locator*. Tipus d'URI per identificar recursos d'internet.

Wi-Fi: Mecanisme de connexió sense fils de dispositius electrònics.

13. ANNEX

Referències per la valoració de solucions

<http://forums.groundspeak.com/GC/index.php?showtopic=272320>
<http://angelcarron80.wordpress.com/2012/09/12/Smartphone-vs-cuentakm-gps/>
<http://adventurecolorad.hubpages.com/hub/How-to-Choose-the-Right-GPS-for-Geocaching>
<http://www.pocketables.com/2011/06/can-a-Smartphone-replace-a-dedicated-gps-receiver.html>
<http://www.canadagps.com/Bluetoothgps.html>
<http://www.htcmania.com/showthread.php?t=236713>
<http://forum.compegps.com/archive/index.php/t-9547.html>
<http://dealnews.com/features/Can-You-Trust-a-Smartphone-to-Guide-You-in-the-Wilderness-/526969.html>
<http://www.retrevo.com/content/blog/2010/05/can-your-Smartphone-replace-gps>
<http://singletrackworld.com/forum/topic/cheap-Android-Smartphone-as-a-dedicated-gps-for-the-bike>
<http://alatest.com/blog/finding-your-way-dedicated-gps-vs-Smartphones/>
<http://bestgpsnavigatorreviews.com/Smartphone-gps-vs-dedicated-gps/>
http://www.backpacker.com/gear/ask_kristin/319
<http://gps.about.com/od/gpsproductoverview/a/new-gps-setup.htm>
<http://boards.straightdope.com/sdmb/showthread.php?t=634942>
<http://adroitaiien.hubpages.com/hub/The-Benefits-Of-A-Dedicated-GPS-vs-A-Smart-Phone-GPS>
<http://www.rei.com/learn/expert-advice/gps-receiver.html>
http://www.ehow.com/info_12223759_Smartphone-vs-handheld-gps.html
<http://forums.groundspeak.com/GC/index.php?showtopic=266585>
<http://www.makeuseof.com/tag/gps-enabled-Smartphone-tracking-device/>

Referències de bateries d'un sol ús

http://www.ciao.es/Staycon_Bateria_Desechable_Opinion_1148894

Referències de preus i models de GPS Externs per *Bluetooth*

<http://tienda.zonagps.com/comprar/150-gps-Bluetooth>
http://www.blauden.com/catalogo/telefon%25C3%25ADa/pdas_y_Smartphones/terminales_y_dispositivos_m%25C3%25B3viles/sistemas_gps/accesorios_telefon%25C3%25ADa_m%25C3%25B3vil/pda_industria_l/receptores_gps_Bluetooth
<http://www.bestbuy-int.com/es/productos/consumer-products/gps/browse/15-gps.html?sef=hc>
<http://www.semsons.com/bluetgpsrec.html>
http://www.amazon.com/s/ref=sr_st?keywords=gps+Bluetooth&qid=1352238208&rh=n%3A172282%2Cn%3A1493964%2Cn%3A172526%2Cn%3A559942%2Ck%3Agps+Bluetooth&sort=relevancerank

Referències de preus d' *Arduino*

<http://www.Arduino.cc/en/Main/Buy> (llistat de proveïdors en la pàgina oficial d' *Arduino*)