



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE CARRERA

TÍTOL DEL TFG: Creació d'una aplicació de Geocatching per l'ajuntament de Badalona

TITULACIÓ: Enginyeria Tècnica en Telecomunicacions especialitat en Telemàtica

AUTOR: Carles Ventura Duran

DIRECTOR: Antoni Oller Arcas

DATA: 30 de Septiembre de 2015

Títol: Creació d'una aplicació de Geocatching per l'ajuntament de Badalona

Autor: Carles Ventura Duran

Director: Antoni Oller Arcas

Data: 30 de Septiembre de 2015

Resum

Les tecnologies de la informació i la comunicació han generat una gran revolució en la manera d'accedir i transmetre informació, generant nous paradigmes de comunicació. En concret, en el camp de la cultura, aquesta evolució tecnològica permet que es millori la interactivitat entre l'usuari i l'entorn digital i que els continguts culturals siguin emmagatzemats i entregats a qualsevol lloc fent servir la connectivitat que ens proporcionen les noves xarxes de comunicacions i en especial les xarxes inalàmbriques mòbils.

La gamificació és una tècnica que utilitza un medi de transmissió de la informació lúdica per oferir continguts que podrien ser més pesats de manera més amena. En aquest context, un joc com el Geocatching pot ajudar a oferir informació cultural sobre un event molt popular a la ciutat de Badalona com és, la "cremada del dimoni". D'altra banda, existeix un gran desconeixement a la població sobre l'origen d'aquesta festivitat.

El departament de cultura de l'ajuntament de Badalona ha definit un itinerari per la ciutat de Badalona (la ruta del dimoni) que explica amb detall la història d'aquesta festa que es remunta al 1940 y la seva evolució en el temps. No obstant, la ruta actual de Geocatching, es realitza mitjançant dispositius GPS i papers i la seva gestió és bastant complexa.

Aquest projecte ha analitzat la situació actual del sistema i s'ha desenvolupat una aplicació mòbil que simplifica el funcionament actual i ofereix millores significatives que seran argumentades en aquesta memòria

Title: Creating a Geocaching Application for city of Badalona

Author: Carles Ventura Duran

Director: Antoni Oller Arcas

Date: September 30 of 2015

Overview

Information and communications technology (ICT) has generated a great revolution in the way to access and transfer information. ICT has also created new communication paradigms. In particular, in the field of culture, this technological evolution allows for improving the interactivity between the user and the digital environment. It also permits to store and deliver cultural contents anywhere using the connectivity provided by communication networks and, especially, by wireless networks.

Gamification is a technique that uses playful data feeds to offer contents that might be tedious in a more enjoyable manner. In this context, a game such as Geocaching helps offer cultural information about a very popular event in the city of Badalona; 'La cremada del dimoni'. On the other hand, there is a lack of knowledge about this festivity among the population.

The Department of Culture of Badalona City Council has established an itinerary through the city -La Ruta del Dimoni- that explains in detail the history of the festivity previously mentioned - which dates back to 1940- and its evolution over time. However, the current Geocaching itinerary is done with GPS devices and papers and its management is quite complex.

This project has analysed the current situation of the system and has developed a mobile phone application to simplify the ongoing operation of this system and offers significant improvements. That will be explained in this document

ÍNDIX

INTRODUCCIÓ	1
CAPÍTOL 1. GEOCATCHING	3
1.1. GEOCATCHING	3
1.2. GAMIFICACIÓ	4
1.3. RUTA DEL DIMONI	5
1.4. OBJECTIUS	5
CAPÍTOL 2. ARQUITECTURA	7
2.1. DIAGRAMA DE L'ARQUITECTURA	7
2.2. FACEBOOKSDK	8
CAPITOL 3. IMPLEMENTACIÓ	9
3.1. ENTORN DE DESENVOLUPAMENT	9
3.2. ELECCIÓ DEL SISTEMA OPERATIU	10
3.2.1. ANDROID	11
3.3. IMPLEMENTACIÓ DE L'APLICACIÓ	11
3.3.1. LOGIN AMB FACEBOOK	11
3.3.2. MAPA DE LA RUTA	12
3.3.3. PUBLICACIÓ DE MISSATGES I FOTOS	13
3.3.4. PUNTS D'INTERÈS (POI)	15
3.3.5. NAVEGACIÓ	16
3.3.6. PROVA SUPERADA	17
3.4. INTEGRACIÓ AMB <i>FACEBOOK SDK</i>	18
3.4.1. LOGIN	18
3.4.2. PUBLICAR COMENTARIS	19
3.5. DESENVOLUPAMENT DE LA BRÚIXOLA	21
3.6. DIAGRAMA DE FLUX DE L'APLICACIÓ	24
3.7. DISTRIBUCIÓ DE CLASSES	25
CAPITOL 4. PLANIFICACIÓ	27
CAPITOL 5. CONCLUSIONS	30
5.1. CONCLUSIONS	30
5.2. IMPACTE MEDIAMBIENTAL	30
5.3. COST DEL PROJECTE	31
5.4. TREBALLS FUTURS	31
5.5. CONCLUSIONS PERSONALS	31
BIBLIOGRAFIA	33

FIGURES

Fig. 1.0.1 Logotip de Geocatching.....	3
Fig. 0.2 Paper que proporciona l'ajuntament de Badalona	4
Fig. 2.1 Diagrama de l'arquitectura	7
Fig. 3.1 Login Satisfactori	12
Fig. 3.2 Splash Screen	12
Fig. 3.3 Mapa de la ruta.....	13
Fig. 3.4 Ruta del dimoni finalitzada	13
Fig. 3.5 Reiniciar ruta	13
Fig. 3.6 Missatge de confirmació.....	14
Fig. 3.7 Publicació d'un comentari.....	14
Fig. 3.8 Selecció de lloc de publicació.....	14
Fig. 3.9 Publicació d'una foto a Facebook.....	14
Fig. 3.10 Punt A.....	15
Fig. 3.11 Punt D.....	15
Fig. 3.12 Configuració localització.....	15
Fig. 3.13 Alerta de GPS.....	15
Fig. 3.14 Pantalla navegació	16
Fig. 3.15 resposta incorrecta	17
Fig. 3.16 Pregunta sobre el POI	17
Fig. 3.17 Resposta correcta	17
Fig. 3.18 Pantalla prova superada.....	17
Fig. 3.19 Codi inicialització	18
Fig. 3.20 Negociació Login	19
Fig. 3.21 Quadres de diàleg natiu.....	19
Fig. 3.22 Publicació de comentaris.....	20
Fig. 3.23 Publicació de fotos	21
Fig. 3.24 Inicialització de sensors i GPS	22
Fig. 3.25 Registre de sensors i GPS	22
Fig. 3.26 Funcionament Sensors.....	23
Fig. 3.27 Diagrama de flux	24
Fig. 3.28 Distribució de classes.....	25
Fig. 4.1 Percentatge de temps destinat a tasques	28

INTRODUCCIÓ

Aquest projecte consisteix en el desenvolupament d'una aplicació Android en l'àmbit cultural i d'entreteniment. Es tracta d'una aplicació que utilitza el sistema de *Geocatching* per gaudir d'una passejada històrica per la Cremada del Dimoni, des dels seus inicis fins els nostres dies. L'origen, els protagonistes, els llocs més emblemàtics i La Cremada, l'acte central que els badalonins reproduïen any rere any amb motiu de la celebració de la festa en honor al patró Sant Anastasi.

El *Geocatching* és una activitat lúdica o joc de pistes que es pot jugar tant a la ciutat com a la natura i que et porten a la descoberta d'un tresor o 'caché' amagat per mitjà de pistes i utilitzant un aparell amb dispositiu de GPS (com un mòbil).

Per tal de facilitar la realització d'aquesta activitat s'ha optat per dissenyar una aplicació específica per a tal fi gràcies a l'ús de telèfons intel·ligents i aprofitar les xarxes socials per interconnectar les persones que fan el *Geocatching*. D'aquesta manera tots els participants de l'activitat podran aportar comentaris sobre la ruta i aprendre dels altres mentre gaudeixen d'una activitat cultural.

Aquest treball s'organitza de la següent manera. En el primer capítol es fa una breu introducció en que consisteix el *Geocatching*, quina relació té amb la ruta del dimoni, quina es la problemàtica que hem de solucionar i els objectius proposats.

En el segon capítol s'explica l'arquitectura que fa servir l'aplicació per al seu funcionament així com les tecnologies escollides per tal d'interconnectar els usuaris a través de la xarxa social més popular.

El tercer capítol és el que més volum de treball té i s'aprofundeix en les eines i com s'ha dut a terme la implementació sobre el disseny planificat per poder assolir els objectius de l'aplicació. Aquí es pot trobar el funcionament pas a pas de l'aplicació així com quins mètodes s'han fet servir per poder desenvolupar les funcions de l'aplicació.

Al quart capítol es detalla la planificació del projecte així com els grups en els que s'han dividit les diferents tasques i un petit resum de les hores que ha consumit cada tasca.

El cinquè capítol conté les conclusions finals sobre el projecte (si s'han assolit o no els objectius proposats inicialment), impacte mediambiental, cost del projecte, futures línies de treball i unes conclusions personals.

CAPÍTOL 1. GEOCATCHING

En aquest capítol s'explica en que consisteix el *Geocaching* o geocerca i què és la ruta del dimoni, així com la problemàtica a solucionar i els objectius proposats.

1.1. Geocaching

El *Geocaching* és una activitat que consisteix en buscar tresors o també anomenats "caches" amagats per altres persones que realitzen la mateixa activitat mitjançant un dispositiu amb GPS o un telèfon mòbil.

Aquesta activitat és pot realitzar per persones de qualsevol edat i pot fer-se tant en mig de la natura com a la ciutat. Només requereix uns quants coneixements d'orientació.

El joc tradicional, com s'ha comentat al paràgraf anterior, consisteix en amagar objectes perquè d'altres persones el trobin i el substitueixin per un altre deixant algun comentari en el llibre de registre que hi ha dins la caixa.

En aquest cas el que s'ha fet és, aprofitar les característiques del *Geocaching* per fer una ruta cultural per Badalona de manera didàctica i divertida sense la necessitat de buscar tresors.

A la web de *Geocaching* [10] es pot veure una aplicació comercial que ofereix serveis de *Geocaching* i en particular es troba una ruta anomenada "Ruta del dimoni" [12]. Durant aquest projecte, s'ha avaluat aquesta aplicació i s'ha descartat per dos motius: el cost de l'aplicació és massa elevat (9€) i no és viable per a tothom, com per exemple el col·lectiu d'estudiants, als que es pretén oferir continguts culturals sobre Badalona. Així que s'ha decidit oferir una aplicació gratuïta per la promoció de la ciutat i de la ruta del dimoni, basada en l'ús de les xarxes socials (*Facebook*)



Fig. 1.0.1 Logotip de Geocaching

[Escriba texto]

[Escriba texto]

[Escriba texto]

El joc tradicional, com s'ha comentat al paràgraf anterior, consisteix en amagar objectes perquè d'altres persones el trobin i el substitueixin per un altre deixant algun comentari en el llibre de registre que hi ha dins la caixa.

A la *Fig. 1.2* es pot veure l'equació necessària per calcular la posició final del tresor en base a unes preguntes parcials.

Coordenadas del cache

N	41°	E	002°	
N	41°	A.C	.	C-D	(A-E)-1	E-C	E	002°	D.E	.	B-38	A-9	C+D

Resultados pruebas:

A =
B =
C =
D =
E =

Fig. 0.2 Paper que proporciona l'ajuntament de Badalona

1.2. Gamificació

El terme gamificació significa emprar mecàniques de joc en entorns i aplicacions no lúdiques amb la finalitat de potenciar la motivació, la concentració, l'esforç, la fidelització i altres valors positius comuns a tots els jocs. Es tracta d'una nova i potent estratègia per influir i motivar a grups de persones.

La correcta implementació d'estratègies de gamificació permet passar de la mera connectivitat al compromís, aconseguint que els membres d'una comunitat, els treballadors d'una empresa, els estudiants, els habitants d'una ciutat o qualsevol col·lectiu o individu participin de manera dinàmica i proactiva en accions que generalment requereixen un esforç de la voluntat.

La integració de dinàmiques de joc en entorns no lúdics no es algo que esdevé com a nou, però el creixement exponencial de l'ús de videojocs en els últims anys ha despertat l'interès de experts en comunicació, psicologia, educació, salut per desxifrar les claus que fan que d'un joc un medi tan eficaç. En aquests darrers anys també ha començat l'expansió per l'estudi de l'aplicació en altres

[Escriba texto]

[Escriba texto]

[Escriba texto]

àmbits no necessàriament lúdics. Així doncs, es podria dir que aquesta tendència és en essència la gamificació.

1.3. Ruta del dimoni

Com s'ha dit a la introducció, aquest projecte respon a la necessitat de facilitar als badalonins i turistes la possibilitat de conèixer la ciutat de manera divertida fent servir el sistema de *Geocatching*.

Aquí, és on sorgeix aquesta passejada anomenada "La Ruta del dimoni". Aquesta ruta té com a eix central donar a conèixer la "Cremada del dimoni" que és la festa tradicional més important de Badalona, i és l'acte central de les "Festes de Maig" que se celebren les primeres setmanes del mes de maig. El 10 de maig a partir de les deu de la nit té lloc a la Rambla de Badalona, davant del mar, el piromusical i la Cremada del Dimoni, que és la culminació d'un seguit d'actes que conformen la Nit de Sant Anastasi. La Generalitat va declarar la Cremada festa d'interès turístic el 1991.

Així doncs, la ruta del dimoni consisteix en una distribució de cinc punts d'interès pels indrets més històrics de la ciutat. Per exemple, el punt A ens porta directament al lloc que va ser l'escenari de les primeres cremades, remuntant-nos al 1940.

1.4. Objectius

Per a poder seguir uns passos correctes s'han establert uns objectius o fites a complir:

- Dissenyar una aplicació mòbil per la realització de la Ruta del dimoni.
- Facilitar eines socials per als usuaris de l'aplicació (enviament fotos, missatgeria...) aprofitant les eines que ens ofereixen les xarxes socials.
- Simplificar processos per a la realització de la ruta (càlcul de coordenades i brúixola integrada al dispositiu mòbil)
- Aprofitar les tècniques de gamificació per oferir continguts culturals de manera amena i al mateix temps captar usuaris als que podem oferir en un futur altres tipus de continguts

CAPÍTOL 2. ARQUITECTURA

En aquest capítol es fa una introducció a les tecnologies escollides per a la realització del projecte.

Per la creació d'aquesta aplicació no ha sigut necessari un estudi profund sobre quines arquitectures utilitzar, ja que s'ha intentat fer servir el que els organitzadors de l'activitat tenien pensat. Per tant, consta d'un diagrama sobre els elements del sistema i una introducció al *SDK* de la xarxa social.

2.1. Diagrama de l'Arquitectura

La següent figura (*Fig. 2.1*) il·lustra a mode de diagrama de blocs, els principals components del sistema.

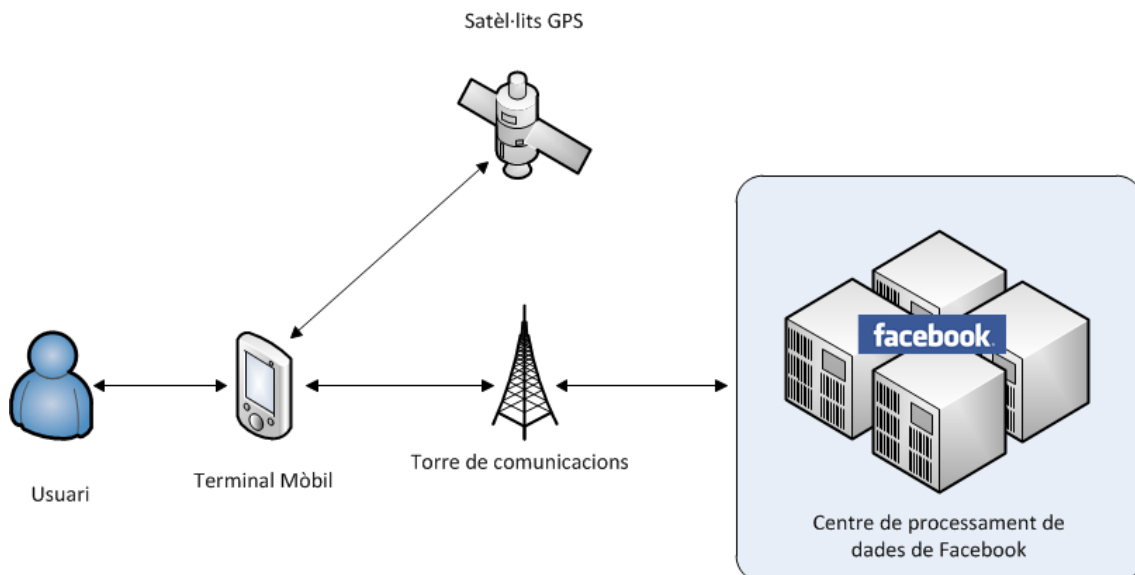


Fig. 2.1 Diagrama de l'arquitectura

- **Usuari:** Representa l'usuari que vol realitzar l'activitat lúdica
- **Terminal mòbil Android:** dispositiu per executar l'aplicació dissenyada en aquest projecte que s'encarrega de realitzar la ruta del dimoni a través de mètodes com el GPS i els sensors incorporats.
- **Satèl·lits GPS:** es un sistema de posicionament que permet determinar la posició exacte del dispositiu per tot el món a través d'una xarxa de satèl·lits.
- **Xarxa de comunicacions:** la xarxa que proporciona als dispositius mòbils la capacitat per connectar-se a internet i fer servir les xarxes socials.

- **Xarxa social Facebook:** xarxa social que proporciona als usuaris la possibilitat de compartir històries fotos, creació d'esdeveniments, pàgines i una infinitat de possibilitats que disposa d'una quantitat enorme d'usuaris.

2.2. FacebookSDK

Per a l'interconnexió amb Facebook de l'aplicació s'ha fet servir el *FacebookSDK*. Que consta d'un conjunt de classes i objectes que permeten realitzar gairebé tot el que fa l'aplicació nativa de Facebook que s'instal·la en els dispositius sense haver de recórrer a ella.

Les seves característiques son:

- Login multiplataforma
- Permet integrar Facebook en aplicacions
- Suporta més de 30 permisos per controlar la privacitat

CAPITOL 3. IMPLEMENTACIÓ

En aquest apartat s'expliquen els passos que s'han seguit per realitzar la implementació de l'arquitectura en el projecte, així com les vistes de l'aplicació i el seu funcionament.. Primer de tot cal definir l'entorn de desenvolupament amb les seves eines i tecnologies.

3.1. Entorn de desenvolupament

Durant el desenvolupament d'aquest projecte s'ha fet servir:

- Un ordinador portàtil amb el sistema operatiu *OSX* per desenvolupar l'aplicació Android amb els entorns de desenvolupament *Eclipse* i *Android Studio*.
- Un telèfon mòbil *Samsung Galaxy S3* per fer les proves.

Per a poder programar i configurar aquest projecte es necessiten una sèrie de software i hardware que es descriuen a continuació.

- **SVN:** És un sistema de revisions de software, de manera que ens permet tenir sempre una còpia anterior del nostre projecte si cal recuperar-lo i permetre que més d'una persona pugui treballar en el mateix desenvolupament. Hi ha altres alternatives com *GIT*, que és descentralitzat. S'ha escollit aquest, per estar familiaritzat amb ell.
- **Adobe Photoshop CS 5.5.** És l'editor d'imatges més reconegut a nivell mundial. S'ha fet servir per la creació de les imatges dels punts, la brúixola, i per redimensionar algunes imatges que eren massa grans per la pantalla d'un telèfon mòbil.
- **Eclipse.** És un entorn de desenvolupament impulsat sobretot per la creació de projectes basats en *JAVA*, encara que suporta altres tipus de llenguatges de programació. Es va començar el desenvolupament amb aquest *IDE* i després es va migrar a *Android Studio* perquè *Google* en va retirar el suport.
- **Android Studio.** És l'entorn de desenvolupament de facto per a la creació d'aplicacions Android. Abans d'aquest es feia servir l'*Eclipse*, però *google* va decidir crear un *IDE* dedicat a partir de la versió community (gratuïta) del *IntelliJ Idea*
- **Office y Visio 2013.** S'han fet servir a la redacció d'aquest document i la realització de diagrames
- **Samsung Galaxy S3.** S'ha fet servir aquest terminal mòbil per a la realització i tests de l'aplicació.

- **Java JRE 1.6:** la màquina virtual de Java es necessària per poder executar l'aplicació sobre *Android* y l'*SDK* (Software Development Kit) d'*Android*.
- **Android SDK 18 :** l'*SDK* ofereix les eines necessàries per poder desenvolupar aplicacions en *Android*. La màquina virtual d'*Android*, el *DDMS* (*Dalvik Debug Monitor Server*) i una interfície gràfica sobre la que dissenyar l'aplicació, és el que Google ens proporciona. Corres pon a la versió 4.3 del sistema operatiu
- **Facebook SDK:** és un conjunt de llibreries que ens donen accés a mètodes predefinitos per relacionar-nos amb els endpoints de *Facebook*, despreocupant-nos de fer nosaltres mateixos les crides als seus serveis web. Per contrapartida, es necessari anar actualitzant l'aplicació amb la última versió d'aquest, ja que contínuament van millorant i canviant els processos de Login entre d'altres.
- **BugSense:** és una eina que serveix per un seguiment dels errors que es produeixen a l'aplicació en temps d'execució. És bastant útil perquè es tracta d'un sistema que permet fer un debugging de l'aplicació una vegada està instal·lada a diferents dispositius mòbils i així poder observar errors atribuïts a un terminal específic.

3.2. Elecció del Sistema Operatiu

Avui dia hi ha plataformes mòbils diferents en la indústria mòbil. Tots són orientats a internet i fets per facilitar als usuaris l'ús d'internet. No hi ha una plataforma correcte i d'altres incorrectes a l'hora de fer-les servir per desenvolupar sobre elles, ja que depèn en l'aproximació que se li vol donar a l'aplicació o si es prefereix arribar a més gent o tenir més beneficis.

Les següents, son les plataformes més emprades avui dia:

- **iOS:** Desenvolupat per *Apple*, només el pots trobar en els seus propis dispositius mòbils com l'*iPhone*, *iPad* i *iPod Touch*. És la segona plataforma per darrere d'*Android* amb més usuaris.
- **Android:** Desenvolupat inicialment per *Android Inc.* i millorat més endavant per *Google* quan va absorbir a la companyia creadora. És la que més quota d'usuaris disposa.
- **Windows Phone:** És la plataforma més jove de totes i per la qual Microsoft vol recuperar el terreny perdut per la anterior plataforma.

La elecció d'*Android* ve per la idea de la facilitat per desenvolupar sobre aquest *SDK*, gràcies a que fa ús del llenguatge de programació d'alt nivell com *JAVA* que redueix la complexitat de la programació alhora que dificulta la seva optimització. A més no es necessària cap llicència per desenvolupar i testejar

les aplicacions, només un terminal mòbil per poder fer ús dels sensors si els necessitem, ja que l'emulador només serveix per a un ús bàsic.

Val a dir que l'aplicació es podria portar a qualsevol de les altres plataformes sense gaire tipus de problemes, ja que la seva separació per capes (Model Vista Controlador) que implementen els tres sistemes operatius facilitaria la feina.

A continuació es fa una petita descripció sobre com funciona el sistema Android.

3.2.1. Android

Android és un sistema operatiu basat en el kernel de Linux, per aquesta raó ja es pot deduir que es tracta de un entorn lliure, gratuït i multi-plataforma. Així doncs, la eina de desenvolupament (*SDK*) és gratuïta. El seu *SDK* fa servir popular llenguatge de programació *JAVA* però utilitza una màquina virtual adaptada anomenada *Dalvik* responsable de interpretar el nostre codi amb una agilitat i rapidesa considerable. Actualment, les darreres versions d'*Android* inclouen una nova màquina virtual més eficient i ràpida que l'anterior anomenada *ART*. Així doncs, mitjançant Java podem accedir a totes les funcionalitats del dispositiu amb simplicitat.

3.3. Implementació de l'aplicació

En aquest apartat es mostra i es detalla el resultat de la implementació. Les explicacions dels resultats estan combinats amb captures de pantalla del terminal de manera que sigui més fàcil entendre les seves funcionalitats.

3.3.1. Login amb Facebook

Aquesta és la pantalla d'Inici o també anomenada "Splash Screen" Aquest és el punt de partida de l'aplicació i com es pot observar a la *Fig. 3.1* conté tots els organismes promotors de la "Ruta del dimoni", un botó de *Login a Facebook* i una petita imatge a la part superior esquerre que serveix per indicar quin és l'usuari logejat en el moment.



Fig. 3.2 Splash Screen



Fig. 3.1 Login Satisfactori

Val a dir que per a la correcta iniciació del procés de Login, el terminal ha de disposar de l'aplicació oficial de Facebook instal·lada (Es pot descarregar des del Google Play). Així doncs, una vegada es clica sobre el botó d'iniciar sessió, s'inicia el procés d'intercanvi d'informació entre la app i l'*SDK* de Facebook on s'aconsegueix el token autoritatiu que permetrà la correcta validació de l'usuari (Fig. 3.2).

En l'apartat posterior "Integració amb Facebook" s'explicarà més detalladament com s'ha implementat el sistema de Login.

3.3.2. Mapa de la ruta

Tot seguit, si no s'ha produït cap error durant el Login, la següent pantalla que es veurà es la de la Fig.3.3. És la pantalla principal de l'aplicació. Aquí es veuen tots els punts de la ruta al complet i el seu estat (finalitzats o no), a més es poden observar la integració de la part social a la part inferior de la pantalla amb una barreta per escriure missatges semblant a la que disposa la popular aplicació de missatgeria *WhatsApp*.



Fig. 3.3 Mapa de la ruta



Fig. 3.4 Ruta del dimoni finalitzada

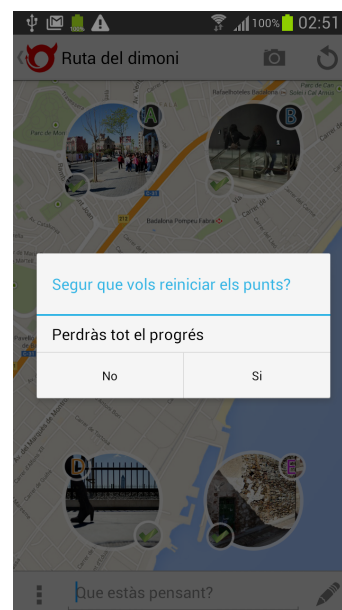


Fig. 3.5 Reiniciar ruta

Com es pot observar a la mateixa figura, les imatges dels punts no trobats estan en blanc i negre, i a mesura que es van trobant recuperen el seu color original i apareix un “checked” de color verd confirmant la superació de la prova. La Fig. 3.4 correspon a la pantalla amb totes les rutes completades. També apareix una icona nova a la *ActionBar* (barra superior d'accions) que serveix per mostrar un diàleg per començar la ruta fent un “reset” de tots els punts (Fig.3.5)

Les icones i la barra inferior s'explicaran amb més detall a l'apartat següent.

3.3.3. Publicació de missatges i fotos

La publicació de missatges i fotos a *Facebook* es pot realitzar des de la vista de l'apartat anterior (Fig. 3.3) i de la vista dels diferents Punts d'Interès (Fig. 3.10). En aquest cas hem aprofitat la mateixa vista per posar l'exemple.

Per realitzar la publicació de missatges a *Facebook* es tant senzill com introduir el comentari a la barra inferior on posa “Que estàs pensant” i a continuació apretar el botó del bolígraf. Una vegada realitzat apareixerà un missatge de confirmació a la pantalla (Fig. 3.8). Per defecte el missatge es publicarà al mur de l'usuari logat, però si es pitja el botó de l'esquerra de la barra de missatges (tres punts), apareixerà un popup amb la opció per canviar el lloc de publicació a la pagina oficial de Facebook de la ruta del dimoni (Fig.3.6)

Pels dos tipus de publicacions, al final del missatge s'afegeix de manera automàtica el hashtag *#rutadeldimoni*, perquè tothom que vulgui llegir els missatges relacionats amb l'activitat ho pugui fer de manera senzilla.



Fig. 3.8 Selecció de lloc de publicació



Fig. 3.7 Publicació d'un comentari



Fig. 3.6 Missatge de confirmació

Per publicar fotos sobre l'activitat, només caldrà pitjar el botó dret de la *ActionBar*. S'ha escollit aprofitar la càmera que porta el sistema *Android* d'stock, ja que integrar una aplicació de càmera portaria molta més carrega de treball alhora que no milloraria la qualitat de la càmera original.



Fig. 3.9 Publicació d'una foto a Facebook

Una vegada rebut el missatge de confirmació. Es pot anar a Facebook i comprovar que la foto s'ha pujat correctament a la xarxa(*Fig.3.9*),

3.3.4. Punts d'Interès (POI)

Aquesta vista correspon a la fase prèvia al recorregut de cada punt, on explica una petita història sobre el punt d'interès específic a mode de guia turística. A les figures (Fig.3.10, Fig.3.11) següents, es veu un exemple dels diferents punts, cada un d'ells amb la seva història particular.

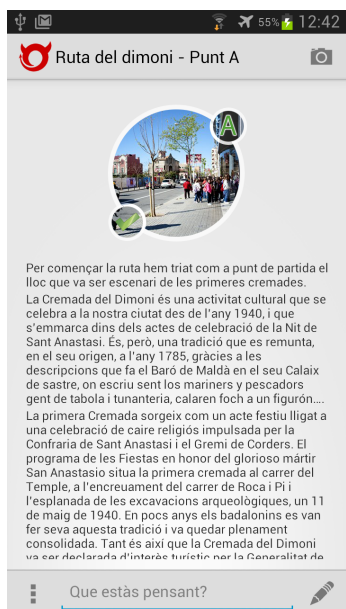


Fig. 3.10 Punt A



Fig. 3.11 Punt D

Si es vol iniciar la ruta pel punt, només caldrà pitjar la imatge del punt. Llavors la aplicació inicia el procés per comprovar si el sistema de localització està activat, sinó es així, sortirà un popup avisant-nos que no està activat i oferint-nos la possibilitat d'activar-ho (Fig. 3.12 i 3.13)



Fig. 3.13 Alerta de GPS

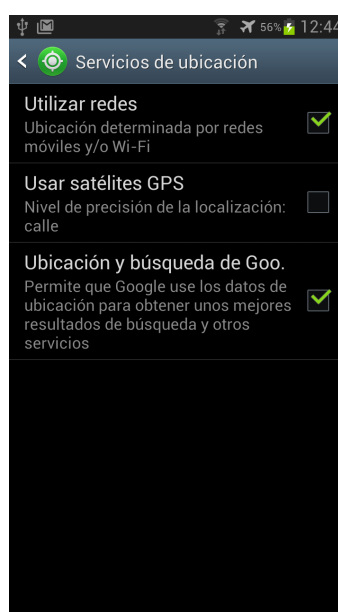


Fig. 3.12 Configuració localització

3.3.5. Navegació

Aquesta vista és la més important de l'activitat. Ens va ajudant a orientar-nos cap al punt correcte a través d'una fletxa que apunta a les coordenades exactes del Punt en qüestió. Cada un dels punts d'interès té unes coordenades (latitud i longitud) proporcionades per els organitzadors de l'activitat. Aquesta pantalla ens dona informació relativa al procés de cerca del punt.

A la *Fig. 3.14* es veu com aquesta vista, a la part superior es proporcionen les coordenades de la posició actual de l'usuari i les coordenades del punt a trobar. A la part inferior hi ha un comptador per saber el temps emprat en finalitzar la prova i la distància en metres fins al punt. Al centre la fletxa a mode de "brúixola" que ens marcarà la direcció a seguir per tal de cercar la ubicació correcte. Per tant, amb ajuda de la "brúixola" i la distancia fins el punt aconseguirem arribar al destí.

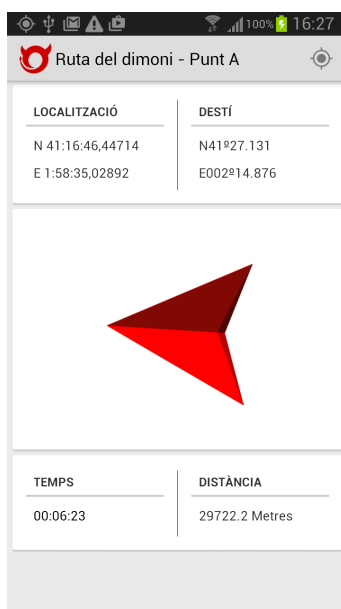


Fig. 3.14 Pantalla navegació

Una vegada l'usuari estigui a la ubicació (aproximadament entre 1 i 2 metres), podrà pitjar el botó dret de la *ActionBar*. A continuació apareixerà un diàleg amb una pregunta relativa al punt es trobi l'usuari (*Fig.3.15*). L'usuari haurà de contestar correctament a la qüestió per superar la prova.

Val a dir que s'ha implementat un sistema de validació de resposta a temps real, per evitar l'obertura i tancament de diàlegs si l'usuari s'equivoca.

A les *Fig. 3.16 i 3.17* es pot observar el sistema de validació de resposta a temps real.

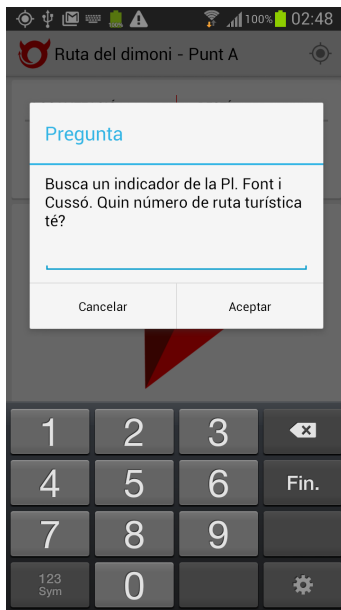


Fig. 3.16 Pregunta sobre el POI

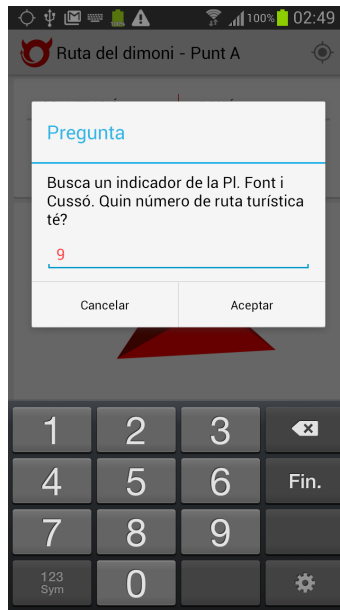


Fig. 3.17 Resposta correcta

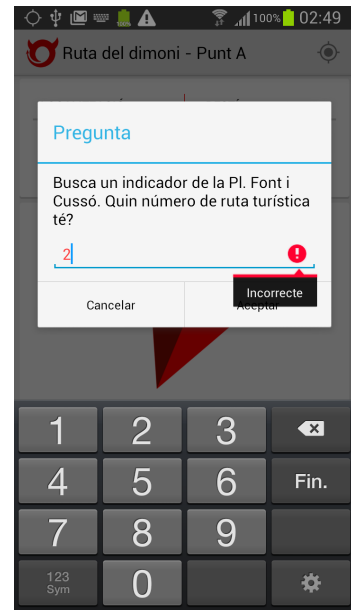


Fig. 3.15 resposta incorrecta

3.3.6. Prova superada

Finalment, si s’han aconseguit superar els 5 punts i per tant, l’usuari ha trobat la “cache” li apareixerà una ultima vista donant-li la enhorabona i un “badge” o medalla per haver superat totes les proves (Fig. 3.18)

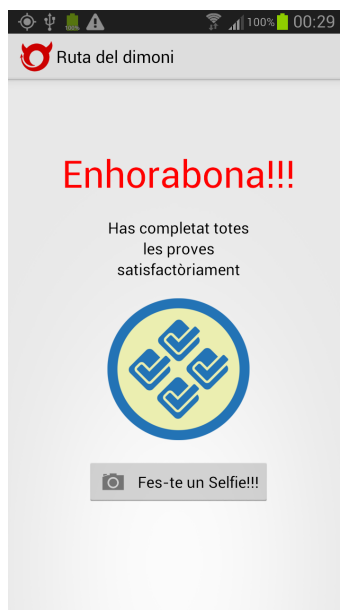


Fig. 3.18 Pantalla prova superada

També li oferirà la possibilitat d'immortalitzar el moment de la trobada del punt amb un "selfie" que serà pujat a la pagina de Facebook de la ruta del dimoni.

3.4. Integració amb *Facebook SDK*

En aquesta secció s'argumenta la implementació de dues de les diferents característiques de les que disposa la llibreria *Facebook SDK*. El procediment per a la configuració del projecte per al correcte funcionament de la llibreria, esta documentat pas a pas a la web de desenvolupadors de *Facebook* [7].

3.4.1. Login

El procés de Login de del Facebook SDK esta dissenyat per ser ràpid i lleuger. Facebook proporciona tres tipus de permisos basics sense haver de omplir cap formulari. Gràcies a aquests permisos l'aplicació podrà efectuar accions o consultar informació en nom de l'usuari.

- **public profile:** permet fer consultes sobre el perfil públic l'usuari logat
- **email:** permet veure el seu correu electrònic
- **user_friends:** permet veure quins amics té l'usuari

Hi ha dues maneres d'Implementar un *Facebook Login*. Fent servir una d'aquestes dues classes:

- **LoginButton:** aquesta classe proporciona el famós botó blau de *Facebook* que es pot afegir a l'interfície gràfica. Fa servir el token autoritatiu de l'aplicació oficial de *Facebook* instal·lada al dispositiu i permet el *login* i *logout* d'usuaris a l'aplicació.
- **LoginManager:** classe encarregada d'iniciar el *login* sense cap element gràfic.

S'ha escollit la primera opció perquè els usuari puguin escollir qual volen i quan no estar logejats.

```
FacebookSdk.sdkInitialize(getApplicationContext());  
callbackManager = CallbackManager.Factory.create();
```

Fig. 3.19 Codi inicialització

A la *Fig. 3.19* es pot observar com s'inicia l'SDK a l'aplicació. Fa una crida al mètode estàtic "sdkInitialize" on se li passa per paràmetre el *Context* de tota la aplicació, perquè pugui fer-se servir durant totes les *Activities*


```
LoginButton authButton = (LoginButton) findViewById(R.id.authButton);
authButton.setReadPermissions(Arrays.asList("user_likes", "user_status"));
authButton.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) { fbSessionState=1; }

    @Override
    public void onCancel() {
    }

    @Override
    public void onError(FacebookException e) {
    }
});
```

Fig. 3.20 Negociació Login

A la Fig.3.20 es pot observar la instanciació de l'objecte *LoginButton*, la definició dels permisos bàsics per logejar (use_likes, user_status) i el registre d'un callback perquè quan acabi el procés de Login, ja sigui correcte o incorrecte, executi el codi dels mètodes *onSuccess*, *onCancel* i *onError*, segons el cas. Concretament aquí posa un 1 a la variable "fbSessionState" perquè s'iniciï la següent *Activity*.

3.4.2. Publicar Comentaris

Per pujar comentaris a Facebook fent servir el Facebook SDK es pot realitzar de dues maneres. La primera és aprofitar els quadres de diàleg nadius de Facebook (Fig.3.21)

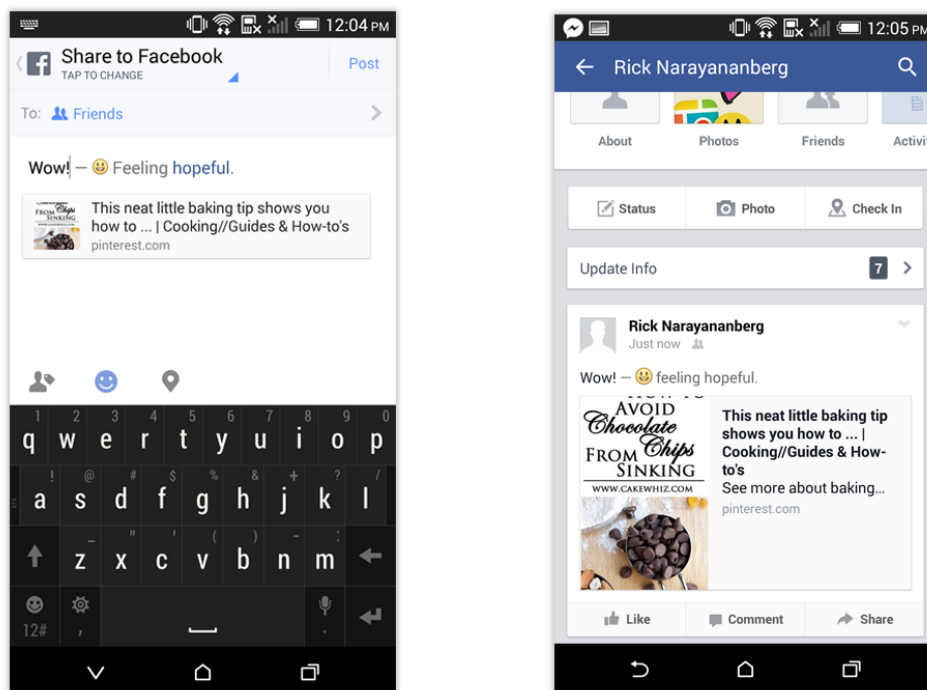


Fig. 3.21 Quadres de diàleg nadius

La segona és desenvolupar una interfície pròpia per tal de que la integració sigui més transparent i amigable a l'usuari. Aquesta opció ha sigut la escollida.

Com s'explica l'apartat d'implementació de l'aplicació, secció de publicació de comentaris i fotos (Fig. 3.7), per enviar missatges es fa servir un *EditText* on que agafa el comentari, el guarda a la variable "comment" l'introdueix a un *Bundle* anomenat "parameters" i l'injecta a una crida a la *GraphApi* de Facebook amb l'objecte *GraphRequest*. Aquest procés es realitza en un fil de execució en segon pla (*AsyncTask*) i que proporciona un missatge de resposta quan s'ha completat l'operació (*onCompleted*) (Fig.3.22).

```
private void postStatus(boolean feed) {
    Bundle parameters = new Bundle();
    parameters.putString("message", comment);

    String path;

    if (feed) {
        path = "me/feed";
    } else {
        path = "badamoni/feed";
    }

    new GraphRequest(AccessToken.getCurrentAccessToken(), path, parameters, HttpMethod.POST, new GraphRequest.Callback() {
        @Override
        public void onCompleted(GraphResponse graphResponse) {
            if (graphResponse.getError() == null) {
                Utils.showToast("Comentari pujat correctament!!!");
            } else {
                Utils.showToast("S'ha produït un error al pujar el comentari. Intenta-ho més tard");
                Utils.showToast(graphResponse.getError().getErrorMessage());
            }
        }
    }).executeAsync();
}
```

Fig. 3.22 Publicació de comentaris

La publicació de fotos és totalment diferent al mètode esmentat abans. Com es pot veure a la Fig. 3.22, es crida al *ContentResolver* per a que obri el fitxer de la foto que se li passa (*fileUri*) i creï un *InputStream* amb nom "input". Després mitjançant el mètode estàtic de la classe *BitmapFactory*, es crea un objecte *Bitmap* a partir de l'stream "input", es processa, reduint-ne les dimensions perquè no consumeixi tant de tràfic la pujada i se li passa a un objecte del tipus *SharePhoto*. A continuació es crea un objecte *SharePhotoContent* que fa servir el patró de disseny *Builder* al que se li passa al constructor l'objecte *SharePhoto* anterior. Finalment es fa una crida al mètode estàtic *share* de la classe *ShareApi* i se li passa com a paràmetre l'objecte del tipus *SharePhotoContent* que s'ha creat en el pas anterior. Quan el procés hagi acabat, llançarà el *callback* amb el mètode corresponent (*onSuccess*, *onCancel*, *onError*) en funció de la resposta. Tant si el resultat és correcte com incorrecte, es mostrarà un *Toast* amb informació relativa.

```

private void postPhoto() throws FileNotFoundException, IOException{
    //Bitmap bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(), fileUri);
    System.gc();
    InputStream input = this.getContentResolver().openInputStream(fileUri);
    //Bitmap bitmap = decodeInputStream(input);

    Bitmap bitmap = BitmapFactory.decodeStream(input);
    Bitmap scaledBitmap = Bitmap.createScaledBitmap(bitmap, bitmap.getWidth() / 2, bitmap.getHeight() / 2, true);
    bitmap.recycle();

    SharePhoto photo = new SharePhoto.Builder().setBitmap(scaledBitmap).setCaption("#rutadelimoni").build();
    SharePhotoContent share = new SharePhotoContent.Builder().addPhoto(photo).build();

    ShareApi.share(share, new FacebookCallback<Sharer.Result>() {
        @Override
        public void onSuccess(Sharer.Result result) { utils.showToast("Foto pujada correctament"); }
        @Override
        public void onCancel() {
        }
    });
    @Override
    public void onError(FacebookException e) {
        utils.showToast("S'ha produït un error al pujar la foto. Intenta-ho més tard");
    }
}
}
}

```

Fig. 3.23 Publicació de fotos

3.5. Desenvolupament de la Brúixola

L'accés a la posició GPS a Android es fa mitjançant les llibreries *Android.Location* i fent servir les classes *LocationManager* i *LocationListener*. L'accés als sensors es fa mitjançant la llibreria *Android.Hardware* i les classes *SensorManager* i *Sensor*.

- **LocationManager:** aquesta classe dona accés al servei de localització del dispositiu mòbil. Aquest servei permet a les aplicacions obtenir actualitzacions periòdiques de la situació geogràfica del dispositiu o llançar esdeveniments quan el terminal s'aproxima a una localització concreta.
- **LocationListener:** aquesta classe es fa servir per rebre les notificacions des de la classe *LocationManager* a mesura que la localització del dispositiu canvia. Per a que funcioni, el *LocationListener* s'ha de registrar amb el servei location manager a través del mètode *requestLocationUpdates*.
- **SensorManager:** aquesta classe permet l'accés als sensors del dispositiu

- **Sensor:** es una classe on es defineixen mitjançant constants, tots els tipus de sensors suportats per Android.

```
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
mAccelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
mMagneticField = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
```

Fig. 3.24 Inicialització de sensors i GPS

A la Fig. 3.24 es veu com s’inicialitzen els objectes “locationManager” i “sensorManager”. També s’observa com s’inicialitzen dos objectes del tipus *Sensor*, per obtenir valors de l’acceleròmetre i el camp magnètic. El acceleròmetre ens servirà per saber la posició del mòbil relativa a la terra i el magnètic per saber on esta el nord.

```
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    sensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    sensorManager.registerListener(this, mMagneticField, SensorManager.SENSOR_DELAY_NORMAL);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);
}

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    sensorManager.unregisterListener(this);
    locationManager.removeUpdates(this);
}
```

Fig. 3.25 Registre de sensors i GPS

Els mètodes *onResume* i *onPause* fan referència moments de represa i parada de l’*Activity*. Per tant, és important que el registre i desregistre dels sensors i del proveïdor d’ubicació es realitzin en aquest apartat ja que si es tanca l’*Activity* i no es paren els sensors, ells seguiran treballant en segon pla i consumiran la bateria del terminal. Tot això es pot observar a la Fig.3.25

Finalment la part del codi més interessant es la de la *Fig. 3.26* i *Fig. 3.27*. Aquí, cada vegada que l'usuari mou el terminal, l'Activity crida al mètode *onSensorChanged*. En aquest mètode es capturen els valors dels sensors segons el tipus i es busca el vector de rotació amb el mètode *getRotationMatrix* i el resultat se li passa al mètode *getOrientation* que proporciona la orientació del terminal envers la terra. Després es fa el càlcul del rumb entre la coordenada actual i la coordenada destí del punt d'interès i se li passa l'objecte al *ImageView* "arrow" que correspon a la fletxa per a que marqui la direcció a seguir.

Mentre es realitzen aquests càlculs cada vegada que l'usuari canvia de posició, es crida a la funció *onLocationChanged* que s'encarrega d'actualitzar la latitud i longituds actuals a la pantalla i de calcular la distància en línia recta amb el mètode *distanceTo*. Aquest mètode és una implementació pròpia de Google de la l'algorisme de *Havershine*.

```
@Override
public void onSensorChanged(SensorEvent event) {
    // TODO Auto-generated method stub
    switch (event.sensor.getType()) {
        case Sensor.TYPE_ACCELEROMETER:
            accelValues = event.values;
            //AccelInfo.setText("" + accelValues[0] + " " + accelValues[1] + " " + accelValues[2]);
            break;
        case Sensor.TYPE_MAGNETIC_FIELD:
            geoValues = event.values;
            //GeoInfo.setText("" + geoValues[0] + " " + geoValues[1] + " " + geoValues[2]);
            break;
    }
    SensorManager.getRotationMatrix(mR, null, accelValues, geoValues);
    SensorManager.getOrientation(mR, orientation);
    float azimuth = (float) ((Math.toDegrees(orientation[0])+360)%360);
    azimuth = azimuth + currLoc.bearingTo(GeoCatchingApp.destLocation[getIntent().getExtras().getInt(RutaActivity.POINTEXTRA)]);
    RotateAnimation rotateArrow = new RotateAnimation(currentDegrees, -azimuth, RotateAnimation.RELATIVE_TO_SELF, 0.5f, RotateAnimation.RELATIVE_TO_SELF, 0.5f);
    rotateArrow.setDuration(250);
    rotateArrow.setFillAfter(true);
    arrow.startAnimation(rotateArrow);
    currentDegrees = -azimuth;
}
```

Fig. 3.26 Funcionament Sensors

```
@Override
public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub
    Log.d("LaLaL", "" + location.getLatitude() + " " + location.getLongitude());
    currentLat.setText("N " + Location.convert(location.getLatitude(), Location.FORMAT_SECONDS));
    currentLong.setText("E " + Location.convert(location.getLongitude(), Location.FORMAT_SECONDS));
    currentDist.setText("" + location.distanceTo(GeoCatchingApp.destLocation[getIntent().getExtras().getInt(RutaActivity.POINTEXTRA)]));
}
```

Fig. 3.27 Funcionament GPS

3.6. Diagrama de flux de l'aplicació

A continuació es mostra el diagrama de flux del funcionament de l'aplicació. Aquest diagrama resumeix el funcionament dels estats de l'aplicació així com la interacció entre les vistes de l'aplicació.

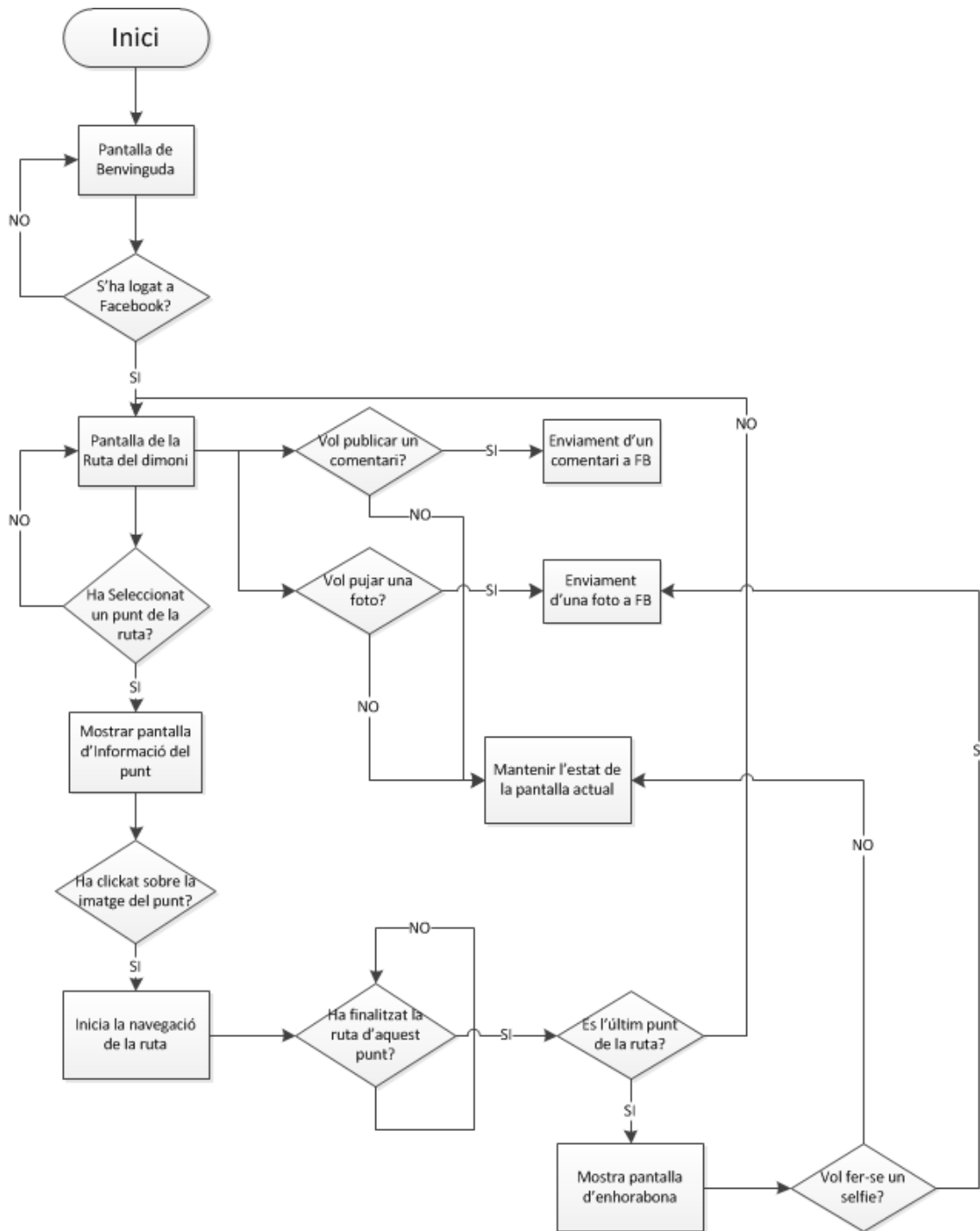


Fig. 3.28 Diagrama de flux

3.7. Distribució de Classes

En aquets apartat es farà una descripció de les classes que s'han desenvolupat i quina es la funció de cada una d'elles.

L'Aplicació conta de quatre *packages* i deu *classes* distribuïts de la següent manera:

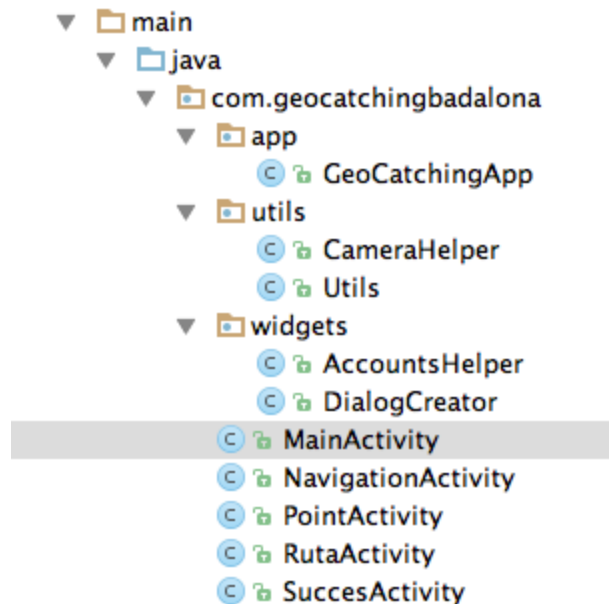


Fig. 3.29 Distribució de classes

El *package* “com.geocatchingbadalona”, és el principal, i esta compostat per les cinc Classes principals que componen les vistes de l'aplicació. A cada una d'aquestes classes esta la relació entre la lògica de l'aplicació i els components de l'interfície gràfica.

El *package* (“com.geocatchingbadalona.app”), només conté una classe, que servirà per guardar dades importants per al correcte funcionament de l'aplicació (token de *Facebook*, resultats de proves, etc).

El *package* (“com.geocatchingbadalona.utils”), conté dues classes on hi ha codi que es reutilitzarà a diferents *Activities*, per tal de no haver d'escriure el mateix codi més d'un cop i fer l'aplicació més eficient, per exemple, els missatges de confirmació i error.

L'últim *package* (“com.geocatchingbadalona.widgets”), conté unes altres dues classes relatives als diferents *Widgets* dels que es fan ús a l'aplicació (quadres de diàleg)

Així doncs les classes de l'aplicació s'encarreguen de:

- **MainActivity:** aquesta classe és la que correspon a la primera pantalla de l'aplicació o “Splash Screen”. S'encarrega de tota la lògica de negociació d'inici de sessió amb crides a mètodes del *Facebook SDK*.

També conté la instanciació i inici de *BugSense* (la aplicació de control d'errors)

- **RutaActivity:** s'encarrega de mostrar la vista de la ruta a partir del fitxer XML "activity_ruta.xml" que conté tota la definició de tota la inicialització de la interfície gràfica estàtica (*ImageViews*, *ImageButtons*, *EditText*, etc). A més inclou tota la lògica per publicar missatges a Facebook i arrencar la càmera per realitzar fotos.
- **PointActivity:** aquesta classe s'encarrega de la lògica de la pantalla de cada Punt i s'encarrega de mostrar la informació relativa a cada punt carregant-la des d'un XML anomenat "strings.xml". També incorpora crides als mètodes encarregats de la publicació de missatges i fotos de la *RutaActivity*.
- **NavigationActivity:** aquesta classe és la que més pes computacional té, ja que incorpora la definició i crida de diferents sensors. S'encarrega de **rebre el senyal GP**, llegir les coordenades i calcular la distància al punt d'Interès mitjançant la fórmula de *Havershine* [11], que és una fórmula per la navegació astronòmica que permet el càlcul de la distància en línia recta entre dos punts de coordenades tenint en compte la curvatura de la terra.
- **SuccesActivity:** és la classe de la última vista i correspon a la pantalla d'Enhorabona. És la que conté menys codi i implementa crides al mètode de publicació de fotos de *RutaActivity*.
- **GeocatchingApp:** és una classe diferent a la resta, ja que no hereta d'*Activity*, sinó que hereta de la classe *Application*. Aquesta classe és la primera que s'arrenca durant la inicialització de l'aplicació i està present al llarg de totes les *Activities*. Normalment s'empra per definir objectes que només s'han d'instanciar un cop i s'han de poder utilitzar a les altres *Activities*. Conté variables estàtiques com el resultat de les preguntes, la localització dels punts d'interès i si ha estat finalitzat o no.
- **CameraHelper:** aquesta classe conté els mètodes que s'encarreguen de crear i donar-li nom un fitxer a la memòria externa del telèfon (*SD*), o a la memòria interna si no en té. Aquest fitxer creat és al que accedirà més endavant la càmera per guardar la informació de la foto.
- **Utils:** com indica el seu nom, és una classe d'utilitats, conté un mètode que s'encarrega de donar feedback a l'usuari sobre els errors i events que es produeixen per pantalla.

CAPITOL 4. PLANIFICACIÓ

En aquest capítol es mostren les diferents tasques realitzades amb la fita d'aconseguir els objectius que es van fixar a l'inici i el temps dedicat a cada una d'aquestes tasques. Les tasques s'han distribuït en 5 grups:

- **Estudi previ:** Aquesta tasca inclou l'estudi de tota la documentació i les tecnologies necessàries per dur a terme la programació de l'aplicació
- **Disseny:** Aquest grup inclou les tasques referents al disseny dels components de l'aplicació, arquitectura i diagrames *UML*.
- **Implementació:** Aquestes tasques estan relacionades amb tot el desenvolupament de l'aplicació a partir de les la informació generada en les tasques anteriors.
- **Tests:** Grup de tasques relacionades en fer proves a l'aplicació per tal d'avaluar possibles errors de funcionament i d'usabilitat.
- **Documentació:** Recopilació d'informació i redacció de la memòria del projecte.

Tasca realitzada	Descripció	Hores invertides
Estudi previ	Estudi funcionament d'altres aplicacions de Geocatching.	20
Estudi previ	Estudi funcionament GPS i	15
Estudi previ	Configuració de les eines i l'entorn de treball	5
Estudi previ	Estudi d'aplicacions d'exemple per agafar idees sobre disseny	10
	TOTAL HORES DE ESTUDI PREVI	50
Disseny	Disseny de la interfície gràfica de l'aplicació	20
Disseny	Disseny de les icones i imatges utilitzades a l'aplicació	15
Disseny	Disseny del comportament de l'aplicació	18
Disseny	Disseny de la lògica de l'aplicació (classes i mètodes)	20
Disseny	Tria de mètodes i característiques que es faran servir de la GraphApi de Facebook	20
	TOTAL HORES DE DISSENY	93
Implementació	Desenvolupament de la interfície gràfica de l'aplicació	50
Implementació	Desenvolupament de les diferents Activities	40
Implementació	Programació de la Brúixola	20
Implementació	Integració de Login de Facebook a l'aplicació	20
Implementació	Integració de funcions de GraphApi de Facebook	20

	TOTAL HORES EN IMPLEMENTACIÓ	150
Tests	Testing de l'aplicació mentre es desenvolupava	20
	TOTAL HORES TESTS	20
Documentació	Recopilació de la documentació utilitzada	10
Documentació	Redacció de la memòria	50
	TOTAL HORES EN DOCUMENTACIÓ	60
	TOTAL HORES DEL PROJECTE	373

Taula. 4.1 Resum d'hores destinades a cada tasca

El següent gràfic mostra el percentatge de temps emprat per cada tasca.

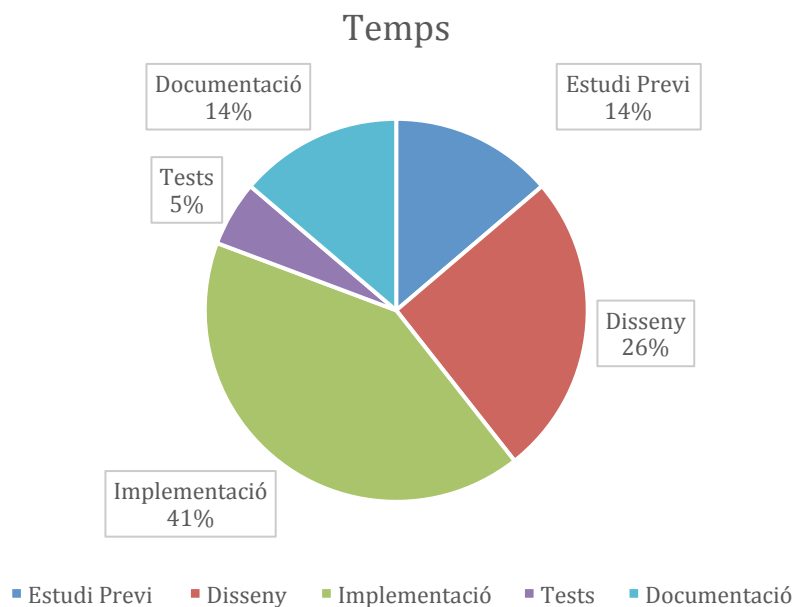


Fig. 4.1 Percentatge de temps destinat a tasques

Com es pot observar en la *Fig. 4.1* les dues tasques que més temps han consumit son el disseny i la implementació respectivament.

La part de disseny potser és la més complicada creativament parlant, ja que és el moment en el que has de concebre quina forma tindrà l'aplicació, quins seran els botons, imatges i resta de components que formen la interfície gràfica tenint en compte que ha de ser un disseny usable i fàcil de fer servir de cara a un ampli col·lectiu d'usuaris. També cal estudiar com relacionar les tecnologies emprades per tal de que la integració sigui correcte.

A més, tot el que comporta el disseny de la interfície d'usuari a Android, és bastant pesat. L'editor visual que inclouen les eines de *Google* no , i convé

familiaritzar-se amb la creació del “layout” a partir del de les corresponents etiquetes *XML*. Fins a un cert punt, per a interfícies estàtiques és suficient. Potser, al que s’ha dedicat més temps és a la generació d’activitats que contenen elements dinàmics. Com és el cas del comportament de la brúixola que ha estat dissenyat programàticament.

Tot i així la part que més temps ha portat ha esta la implementació, bàsicament això es deu a que mentre es programa l’aplicació van sorgint constants canvis provocats perquè algun test que es fan per comprovar el funcionament de l’aplicació no ha tingut el resultat esperat. A més a més d’algun canvi que s’ha hagut de fer sobre el *SDK* de *Facebook* per adaptar-lo a les ultimes versions que anaven sorgint mentre es feia aquest treball.

CAPITOL 5. CONCLUSIONS

En aquest apartat es resumeixen les conclusions, s'exposa l'impacte mediambiental i es plantegen futures línies de treball

5.1. Conclusions

Una vegada finalitzat el treball podem concloure fent una revisió dels objectius establerts prèviament.

Es pot considerar que s'han assolit gairebé la totalitat dels objectius. Els proposats a l'inici de la memòria, tots han estat assolits, però algunes millores sobre la precisió la brúixola s'han quedat pel camí, però que poden servir com a futura línia de treball.

Així doncs podem dir que un usuari que s'instal·la l'aplicació podrà logarse automàticament a Facebook per realitzar la "Ruta del dimoni" mentre gaudeix de la història de l'activitat cultural alhora que fa servir les xarxes socials per relacionar-se amb gent que està fent la mateixa activitat. Tot això estalviant-se haver d'anar a l'ajuntament a demanar uns dispositius específics *GPS* i ajudant als organitzadors a veure el feedback de l'activitat perquè en un futur s'ofereixin altres tipus de continguts.

Per acabar, es pot dir que el comportament de l'aplicació es robust i respon positivament al feedback de l'usuari. S'ha tractat de minimitzar la possibilitat que l'usuari trobi defectes de disseny així com la possibilitat d'exploitar-los.

5.2. Impacte mediambiental

Avui dia l'impacte mediambiental és molt important de cara a la iniciació de projectes. La viabilitat mediambiental és algo que es té en compte a l'hora d'aprovar o rebutjar nous projectes.

Sobre aquests tipus de projectes és difícil estudiar i extreure l'impacte sobre el medi ambient, ja que pot semblar que no pel mig maquinària industrial que generi grans quantitats de contaminació, però la realitat és diferent. Per exemple, que la gent pugui estar interconnectada les 24h del dia mitjançant les xarxes socials, implica tenir en funcionament grans centres de processament de dades que consumeixen altes quantitats d'energia per al seu correcte funcionament.

Una altra cosa que gairebé no es valora, és el fet d'haver de carregar el mòbil. Avui dia la duració de la bateria dels telèfons intel·ligents és molt limitada, principalment per la quantitat de serveis en segon pla que s'executen en el terminal per tal de satisfer els requeriments de l'usuari.

Si es veu des del punt de vista d'un sol usuari el consum es molt baix, però si sumem totes les persones que carreguen a diari el seus terminals sense

comptar que hi ha persones que tenen més d'un dispositiu llavors la xifra d'impacte energètic comença a assolir nivells d'alerta.

5.3. Cost del projecte

Basant-nos en la taula Taula. 1, on es veuen desglossades les hores i la dedicació que s'ha invertit en el projecte i assolint com a cost per hora d'un Enginyer en pràctiques es de 8€, tenim un total de 2984€ al que se li hauria de sumar el cost d'un terminal mòbil amb Android, que pot oscil·lar sobre els 300 €

En aquest tipus de projectes, on es desenvolupen aplicacions mòbils no es requereix d'una gran inversió inicial, ja que avui dia tothom disposa d'un ordinador i un telèfon mòbil intel·ligent (per fer les proves) en els quals es podria començar a desenvolupar sense cap tipus de problema. En aquest sentit es bastant accessible.

5.4. Treballs Futurs

Com gairebé tots els projectes, sempre hi ha marge de millora. En aquest cas les futures línies de treball a tenir en compte serien:

- Millorar la interfície gràfica per donar-li un toc més professional
- Migrar l'aplicació a les últimes versions d'*Android* i així poder implementar-ne les noves funcionalitats i millores de cara a localització i tractament de sensors (reducció del consum de bateria).
- Dissenyar un sistema que permeti la auto integració per part dels usuaris de noves rutes culturals per Badalona
- Integrar noves xarxes socials en el marc de la temàtica de l'aplicació. És a dir tindria sentit integrar *Twitter*, però no *LinkedIn*.
- Modificar la brúixola perquè ofereixi un posicionament en tres dimensions i així facilitar la cerca dels punts.
- Crear una versió de l'aplicació per altres plataformes mòbils (*iOS*, *Windows Phone*)

5.5. Conclusions Personals

Considero que la realització d'aquest projecte ha sigut molt interessant i enriquidora com a finalització dels meus estudis d'Enginyeria telemàtica. Si més no, ja havia fet alguna aplicació Android, però m'he adonat que el que sabia fins ara era bastant poc en comparació amb el que he après fent aquest treball.

La utilització d'*Android* como a plataforma per a desenvolupar m'ha sigut molt útil a l'hora de conèixer com es desenvolupen aplicacions a través dels mòbils actuals basats en aquests sistema operatiu i les experiències que poden proporcionar als usuaris mitjançant el conjunt de sensors que disposen. A més,

em pot ser molt útil de cara a la meva carrera professional en curt termini. A més l'estudi de noves tecnologies ajuda a incrementar els coneixements i la facilitat de comprensió de cara a futurs estudis.

També cal dir que la creació d'aquets projectes ajuda a cercar la informació tècnica necessària de manera correcta quan sorgeixen problemes o quan no se sap com continuar, alhora que ajuda a aprendre dels errors comesos mentre es desenvolupa el projecte. En aquets tipus de situacions és important treballar en un equip. Tant amics com el director del projecte son un bon suport alhora de contestar a preguntes, donar idees o veure els problemes des d'un altre punt de vista.

BIBLIOGRAFIA

- [1] Facebook, URL, En Línia <http://www.facebook.com/> [Últim accés: septembre 2015].
- [2] Android Developers, URL: <http://developer.android.com/> [Últim accés: septembre 2015].
- [3] Universitat Politècnica de Catalunya, URL: <http://www.upc.edu/> [Últim accés: septembre 2015].
- [4] Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, URL: <http://eetac.upc.edu/> [Últim accés: septembre 2015].
- [5] Stackoverflow, URL: <http://stackoverflow.com/> [Últim accés: septembre 2015].
- [6] Geocaching, URL: <http://www.geocaching.com/> [Últim accés: septembre 2015].
- [7] Facebook Developers, URL: <https://developers.facebook.com/> [Últim accés: septembre 2015].
- [8] Movable Type, URL: <http://www.movable-type.co.uk/> [Últim accés: septembre 2015].
- [9] Eclipse, URL: <http://www.eclipse.org/> [Últim accés: septembre 2015].
- [10] Geocaching, URL: <http://www.movable-type.co.uk/> [Últim accés: septembre 2015].
- [11] Fórmula de Haversine. URL: https://es.wikipedia.org/wiki/F%C3%B3rmula_del_Haversine [Últim accés: septembre 2015].
- [8] Movable Type, URL: <http://www.movable-type.co.uk>
- [8] Movable Type, URL: <http://www.movable-type.co.uk>