

EfficientRoutes



Noel Amaya Moreno

Albert Grau Perisé

Jordi Almirall López

Juny 2013

Universitat Oberta de Catalunya

ÍNDEX

PRESENTACIÓ	2
Introducció	2
Motivació i justificació	3
Descripció del projecte	4
Referències	6
CARACTERÍSTIQUES TÈCNiques	9
Requeriments	9
Eines utilitzades	9
PLANIFICACIÓ	10
Llistat de tasques	11
Diagrama de Gantt	12
ANÀLISI	13
Mètodes d'indagació	13
Perfils d'usuaris	15
DISSENY CONCEPTUAL	18
Escenaris d'ús	18
Fluxos d'interacció	21
DISSENY GRÀFIC	22
Interfície gràfica	22
Diferències amb els prototipus inicials	24
ARQUITECTURA DE L'APLICACIÓ	25
Classes	
ActualitzaMapa	25
BuscaDireccio	26
CarregarURL	27
CreadorGoogleJSON	28
InterpretadorGoogleJSON	29
DibuixaRuta	30
MainActivity	31
AVALUACIÓ	35
Prova d'errors	35
Proposta de modificacions futures	35
CONCLUSIONS	36
BIBLIOGRAFIA	37

PRESENTACIÓ

INTRODUCCIÓ

La irrupció dels *smartphones* a les nostres vides des de fa alguns anys ens està generant una veritable necessitat i dependència d'aquests terminals. Ja formen part de la imatge quotidiana de qualsevol taula on paren a dinar o a prendre una copa els amics o familiars, sobretot en les franges d'edat menys veteranes. Poc a poc ens estem acostumant als serveis que ofereixen les empreses mitjançant petites aplicacions per l'*smartphone* i que ens fan la vida més fàcil i còmode.

Aquest nou mercat que s'obre amb els telèfons d'última tecnologia ha provocat una ànsia als departaments de màrqueting per disposar una aplicació pròpia que doni valor afegit a l'empresa o pugui atraure a futurs clients. A més, amb la proliferació de les *tablets* el negoci es duplica i es torna més important encara. Conseqüentment, les consultories informàtiques estan al peu del canó per posar-se al dia tecnològicament i poder absorbir tota la demanda que és generada.

Trobar la idea original d'aplicació per a dispositius mòbils que sigui exclusiva i es desmarqui de la resta és la gran troballa que cerquen els caps de projecte, en una tecnologia que és pràcticament nova i que no para de revisar-se.

MOTIVACIÓ I JUSTIFICACIÓ

En multitud d'ocasions ens trobem amb diverses tasques que hem de realitzar en punts o llocs diferents: el repartidor en realitzar els lliuraments, al fer la compra de reis en diversos establiments, visitar museus en una ciutat, o fins i tot, qualsevol de nosaltres en fer els encàrrecs diaris (anar a comprar el pa, la fruita i el periòdic), i és aquí on pensem ...

“Per on aniria millor per acabar abans?”

En general, intentarem trobar la millor ruta, la que ens permeti passar per tots els punts en el menor temps possible o recorrent la menor distància possible, tenint en compte tots aquells factors importants que puguin intervenir en el càlcul d'aquesta ruta: distància a recórrer, sentit de les vies, senyalització, etc. Un altre factor important, podria ser el mitjà en el qual es realitzi la ruta (cotxe / a peu).

Una situació que en teoria pot resultar senzilla, de manera pràctica no ho és tant, sobretot quan augmenten el nombre de punts intermedis o parades i la distància entre els punts comença a créixer. El dilema esdevé un problema, i a més amb nom propi: El problema del viatjant, o TSP: *Travelling Salesman Problem*, encara que això ja pertany més al camp de les matemàtiques i l'optimització combinatòria computacional.

L'aplicació es desenvoluparà en Android de Google, atenent a diferents aspectes. El més important és que és el principal sistema operatiu en les vendes de *smartphones*, acaparant el 68.8% de la quota de mercat el 2012, i la seva progressió sembla imparable, segons Xakatamovil (<http://www.xatakamovil.com/mercado/android-y-ios-se-quedan-con-el-87-6-de-la-cuota-de-mercado-en-2012-segun-idc>). Altres aspectes menys importants però útils és que dispenso de diferents terminals de telèfons i tablets amb diferents versions d'Android per fer el testeig, i que Java és el llenguatge de programació que em resulta més conegut.

DESCRIPCIÓ DEL PROJECTE

EfficientRoutes vol ser una aplicació d'optimitzador de rutes, una eina per calcular la millor ruta: introduïm una sèrie d'adreces per les que volem passar indicant el punt d'inici i el punt final, i l'aplicació calcula la millor ruta passant per tots els punts indicats. Aquests punts són localitzacions que l'usuari pot entrar mitjançant una cerca escrita de les adreces o prement un punt sobre el mapa, i són representat per marcadors. Aquests marcadors s'hauran de confirmar, i són de color vermell en primer estat i passen a ser de color verd quan es confirmen.

En el cas que la localització es vulgui aconseguir prement un punt sobre el mapa es crea un marcador vermell que mostra la direcció d'aquella localització en una finestra adjunta, i en el cas que la localització es vulgui aconseguir amb la cerca per adreça l'aplicació mostrarà un màxim de quatre resultats probables (marcadors vermells), els quals es poden consultar les seves adreces picant sobre el marcador i obrint-se una finestra adjunta. En ambdós casos, si estem segurs de que és la localització que volíem, picant a sobre de la finestra generada amb l'adreça es confirmarà i quedarà guardada i el marcador es tornarà de color verd i fixat sobre el mapa. Repetim aquests passos tants cops com localitzacions vulguem visitar.

Un cop tinguem tots els marcadors desitjats confirmats i abans d'optimitzar la ruta, hem d'especificar el tipus de transport en que ens volem moure, a escollir entre cotxe, bici (per carril bici) o a peu. Per defecte es calcularà en cotxe si no es modifica. Al optimitzar ruta es mostren gràficament els resultats, si n'hi han, a un mapa de Google Maps dibuixant una línia de color blau que representa el recorregut. Aquesta línia va d'un blau molt fluix fins a un blau pur segons el sentit de la ruta, canviant la tonalitat cada cop que passa per una localització escollida.

Val a dir que hi ha punts que no estan permesos al no ser transitables, com per exemple el mar, llacs, rius... En aquest cas l'aplicació informa a l'usuari que no és un punt vàlid i no guarda la localització. També ens podem trobar rutes impossibles que creuen per aigua o que hi han camins no transitables o que, simplement, l'API de direccions de Google no és capaç de retornar un ruta. En aquests casos també

s'adverteix a l'usuari de que la ruta no és possible i no es dibuixa.

L'aplicació té un menú d'opcions en el qual es pot alternar la vista de mapa entre vista mapa, satèl·lit, híbrid o relleu i també es pot decidir si volem que la nostra ubicació actual sigui presa com a posició d'origen, de destinació de la ruta, ambdues o cap. En cas que no fos la posició d'origen, aquesta la prendríem en el primer marcador que confirméssim ja que es necessari un origen establert.

Quan estan els marcadors confirmats es guarden en memòria i queden fixes en el mapa per fer qualsevol modificació en el tipus de ruta o afegir més localitzacions. En qualsevol moment es pot esborrar la ruta actual i “netejar” el mapa amb el botó corresponent.

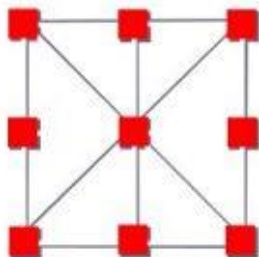
REFERÈNCIESApps Android:

BestRoute	
Plataforma	Android
Desenvolupador	Spiral Scratch Software
Web desenvolupador	www.facebook.com/BestRoute
Preu	Gratis / Pagament
Nota	3,9 estrelles
Data d'actualització	2 de febrer de 2013
Idioma	Anglès
Versió mínima	2.2
Tamany	1,3 Mb
Descàrregues	100.000 – 500.000
Punts a destacar: <ul style="list-style-type: none"> - Punts mostrats en mapa - Resultats mostrats numerats en mapa o les localitzacions en llista - Opció a peu/en vehicle - Permet marcar els punts visitats - Possibilitat de tornar a origen - Exporta ruta a Navigation - Permet intercanviar l'ordre de les localitzacions després d'afegir-les - Pot afegir localitzacions des d'un punt del mapa 	
Punts a millorar: <ul style="list-style-type: none"> - No mostra ruta pintada - Botons poc comprensibles - Botó d'optimitzar poc visible - Informació de ruta molt confusa (duració/distància) 	



ESPO – Routenplaner

Plataforma	Android
Desenvolupador	Duy Khuong Nguyen
Web desenvolupador	www.facebook.com/espo.route.planner
Preu	Gratis / Pagament
Nota	4,0 estrelles
Data d'actualització	2 de setembre de 2012
Idioma	Anglès
Versió mínima	2.2
Tamany	2,3 Mb
Descàrregues	5.000 – 10.000



Punts a destacar:

- Punts mostrats en mapa
- Resultats mostrats numerats en mapa o les localitzacions en llista
- Ruta pintada en mapa
- Opció a peu/en vehicle
- Possibilitat de tornar a origen
- Exporta ruta a Navigation
- Mostra punt en Street View
- Pot afegir localitzacions des d'un punt del mapa
- Informació de ruta (duració/distància)

Punts a millorar:

- La càrrega i maneig del mapa imprecisa.
- No pot intercanviar l'ordre de les localitzacions després d'afegir-les

Aplicacions Web:

Optimizador de Rutas
http://blogistica.es/recursos/optimizador-rutas/
<p>Punts a destacar:</p> <ul style="list-style-type: none">- Punts mostrats en mapa- Opció a peu/en vehicle- Possibilitat de tornar a origen- Pot afegir localitzacions des d'un punt del mapa- Exporta ruta a Google Maps- Permet intercanviar assignar localitzacions com a punt d'inici/final després d'afegir-les- Informació de ruta (duració/distància)
<p>Punts a millorar:</p> <ul style="list-style-type: none">- Algunes cerques per direcció dolentes- No mostra ruta pintada

CARACTERÍSTIQUES TÈCNIQUES

REQUERIMENTS

- El desenvolupament de l'aplicació s'ha realitzat amb tecnologia Android, compilada amb la última versió 4.1.2 (Jelly Bean) i està garantida la compatibilitat a partir de la versió 2.2 (Froyo) d'Android. A causa de la utilització de Google Play Services no es pot utilitzar l'emulador d'Android que pot utilitzar l'Eclipse.
- L'aplicació utilitza i necessita de la utilitat Google Play Services. En cas que el terminal en el que s'executa l'aplicació no ho tingui instal·lat, informará a l'usuari i oferirà de manera còmoda un enllaç de descàrrega del Google Play Services al Google Play, el qual és gratuït. En el cas de la compilació, s'ha d'importar la llibreria google-play-services_lib al nostre projecte.
- L'aplicació mòbil necessitarà permisos per accedir a dades d'Internet i obtenir la ubicació tant per xarxa com per GPS.

EINES UTILITZADES

- L'eina utilitzada pel desenvolupament és Eclipse amb l'ADT plug-in i l'SDK d'Android versió 21.1.
- Per al testeig real de l'aplicació disposem de 3 smartphones amb versions d'Android 4.0.4 i dos amb 4.1.2, i una tablet amb versió 3.2, de diferents fabricants i característiques.

PLANIFICACIÓ

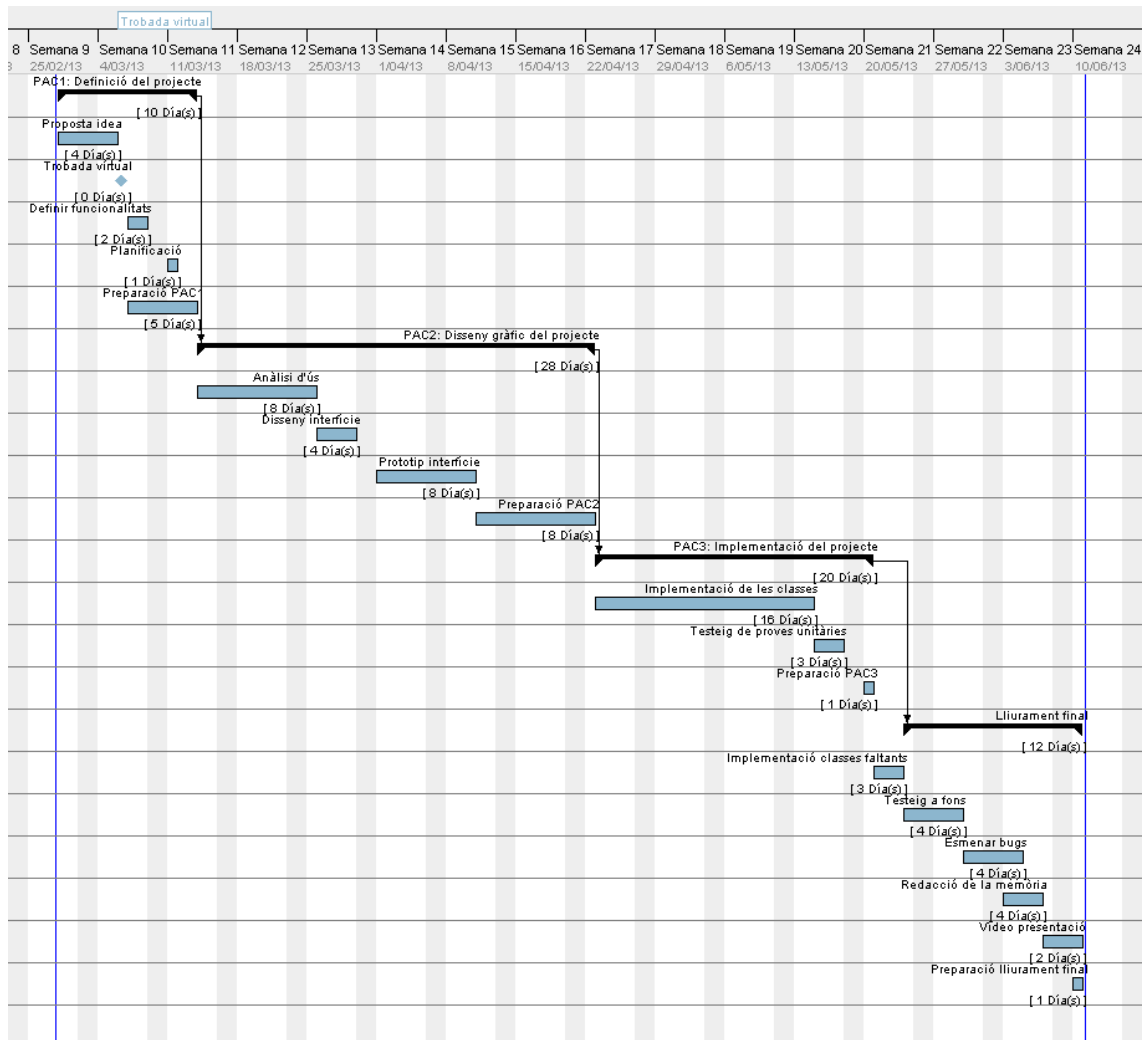
L'aplicació ha sigut desenvolupada utilitzant una metodologia de disseny centrat en l'usuari (DCU). Aquesta metodologia consisteix en un model de procés format per varies fases iteratives (investigació i anàlisi dels usuaris, el disseny i l'avaluació) que es retroalimenten de la resposta dels usuaris per a millorar i adaptar els elements dissenyats fins al moment. Això vol dir que la planificació del projecte es focalitza en l'usuari i contínuament s'està tornant a analitzar totes les fases i modificant-les. Aquest mètode implica una dificultat per establir una planificació seguida, ja que no ha sigut un procés lineal, però a grans trets s'han seguit unes fases implícites.

El projecte estava definit en quatre entregues (PACs) en dates determinades. Durant el projecte m'han sorgit alguns imprevistos i problemes familiars que m'han endarrerit algunes fases, sobretot les dues primeres PACs, lo qual ha derivat en una reducció de terminis per les dues últimes fites.

LLISTAT DE TASQUES

Nombre	Fecha de inicio	Fecha de fin
● PAC1: Definició del projecte	28/02/13	13/03/13
● Proposta idea	28/02/13	5/03/13
● Trobada virtual	6/03/13	6/03/13
● Definir funcionalitats	7/03/13	8/03/13
● Planificació	11/03/13	11/03/13
● Preparació PAC1	7/03/13	13/03/13
● PAC2: Disseny gràfic del projecte	14/03/13	22/04/13
● Anàlisi d'ús	14/03/13	25/03/13
● Disseny interfície	26/03/13	29/03/13
● Prototip interfície	1/04/13	10/04/13
● Preparació PAC2	11/04/13	22/04/13
● PAC3: Implementació del projecte	23/04/13	20/05/13
● Implementació de les classes	23/04/13	14/05/13
● Testeig de proves unitàries	15/05/13	17/05/13
● Preparació PAC3	20/05/13	20/05/13
● Lliurament final	24/05/13	10/06/13
● Implementació classes faltants	21/05/13	23/05/13
● Testeig a fons	24/05/13	29/05/13
● Esmenar bugs	30/05/13	4/06/13
● Redacció de la memòria	3/06/13	6/06/13
● Vídeo presentació	7/06/13	10/06/13
● Preparació lliurament final	10/06/13	10/06/13

DIAGRAMA DE GANTT



ANÀLISI

MÈTODES D'INDAGACIÓ

He utilitzat uns mètodes d'indagació que no requerissin de molts recursos i que fossin el menys intrusius possible. Així, he optat pels mètodes següents:

-Observació i investigació contextual

-Anàlisi competitiva (benchmarking)

Ara passarem a descriure l'experiència i els resultats amb cadascun d'aquests mètodes.

L'Observació i investigació contextual: He realitzat el seguiment de la prova mentre els usuaris complien les tasques sobre les quals l'aplicació podria treure més profit, fent una simulació in situ, observant i analitzant els comportaments del usuaris vers l'aplicació en el moment concret de la seva necessitat. Per tant, s'ha forçat en certa mesura l'entorn i el moment en el qual els usuaris fan l'ús de l'aplicació, encara que els usuaris tenen una certa rutina que permet validar de manera fiable els resultats de la prova.

Deduccions generals:

- Partim de que l'aplicació no és d'ús continu, és per una necessitat puntual que pot ser més o menys freqüent segons el perfil d'usuari. Al ser programadora de rutes, tenen un objectiu inicial i un final. Els usuaris han de dedicar temps al principi de cada ruta que volen programar, per tal de introduir els punts de passada desitjats i veure el resultat optimitzat.
- Els usuaris que no tinguin el GPS connectat ni cobertura de xarxa poden trobar problemes a la precisió de la seva posició coordinada.
- L'ús de l'aplicació mentre es fa la ruta és bastant freqüent, ja que indica en mapa la ruta a seguir d'una localització a una altra.
- Per lo general, els usuaris sempre tornen al final de la ruta al punt d'inici.

Aspectes a tenir en compte:

- El menú i la tasca d'inserció de localitzacions havia de ser hàbil, simple i deductiva, a fi d'estalviar el màxim de temps en la funció que més atenció activa dedica.
- A fi de fer una precisió acurada de la posició, s'haurà de tenir engegat el GPS per utilitzar l'aplicació, encara que primer troba la posició a través de la xarxa mòbil.
- S'ha d'oferir que la ruta acabi en la posició que comença l'usuari.

Anàlisi competitiva: He analitzat tres aplicacions de la mateixa plataforma (Android) per la raó de que només dispo de dispositius mòbils amb aquest SO, i no ens serveix SO d'escriptori com Windows, Linux o Mac perquè aquesta aplicació està molt enfocada a la mobilitat.

Després de fer ús de les aplicacions he observat alguns defectes importants a tenir en compte de cara al desenvolupament de la nostra aplicació, que poden resultar especialment molestos:

- Les icones han de ser molt clares i intuïtives, ja que si no rebaixa la usabilitat i crea desconfiança.
- Els punters en el mapa també han de ser senzills i definits per no confondre el punt en el mapa.

Hi ha punts en comú en les aplicacions:

- Al començar, mostren en la pantalla principal:
 - La vista de mapa.
 - Una icona per afegir una nou punt de pas.
- Totes les aplicacions tenen de manera bastant accessible:
 - Una icona per centrar en pantalla la nostra posició actual.
 - Una icona per mostrar el llistat o resum dels punts de pas.
 - Desar rutes planejades i recuperar-les.

- La majoria incorporen:
 - Un commutador de vista de mapa/satèl·lit
 - Opció de tornar al punt de sortida.

PERFILS D'USUARIS

Perfil	Shopper
Característiques	
Mobilitat geogràfica	Zona urbana a peu.
Interessos	Moda.
Motivacions	Recórrer diferents comerços coneguts i previstos.
Experiència amb tecnologia mòbil	Ús intensiu del mòbil en poques aplicacions.
Contexts d'ús	
· Abans de sortir de compres, a casa o al carrer.	
Anàlisi de tasques	
· Cercar el comerç pel nom o direcció a Google Maps.	
· Afegir el comerç a una llista.	
· Mostrar els marcadors de comerços introduïts.	
· Cercar la ruta més ràpida.	
· Mostrar el resultat de la ruta al mapa o en llista.	
Perfil	Turista
Característiques	

Mobilitat geogràfica	Zona urbana a peu i en transport públic.
Interessos	Cultura, restauració
Motivacions	Recórrer diferents punts turístics d'una la ciutat.
Experiència amb tecnologia mòbil	Ampli rang d'experiència d'usuaris
Contexts d'ús	
· De vacances o visita a una ciutat, al començar la jornada en el hotel o en el carrer.	
Anàlisi de tasques	
· Cercar el punt turístic pel nom o direcció a Google Maps.	
· Afegir el punt turístic a una llista.	
· Mostrar els marcadors de punt turístic introduïts.	
· Cercar la ruta més ràpida.	
· Mostrar el resultat de la ruta al mapa o en llista.	

Perfil	Comprador domèstic
Característiques	
Mobilitat geogràfica	Zona urbana a peu.
Interessos	Compres quotidianes
Motivacions	Fer les compres habituals de menjar y de més necessitat.
Experiència amb tecnologia mòbil	Baixa experiència.
Contexts d'ús	
· Abans de sortir a fer la compra, a casa o al carrer.	

Anàlisi de tasques
· Cercar el comerç pel nom o direcció a Google Maps.
· Afegir el comerç a una llista.
· Mostrar els marcadors de comerços introduïts.
· Cercar la ruta més ràpida.
· Mostrar el resultat de la ruta al mapa o en llista.

Perfil	Repartidor
Característiques	
Mobilitat geogràfica	Zona urbana i interurbana, en vehicle.
Interessos	Ofici
Motivacions	Fer entregues en diversos punts de la ciutat
Experiència amb tecnologia mòbil	Acostumat a fer servir algun dispositiu mòbil per la seva tasca.
Contexts d'ús	
· Abans de fer la ruta diària, a casa o al carrer, o en qualsevol moment de la ruta.	
Anàlisi de tasques	
· Cercar el comerç pel nom o direcció a Google Maps.	
· Afegir el comerç a una llista.	
· Mostrar els marcadors de comerços introduïts.	
· Cercar la ruta més ràpida.	
· Mostrar el resultat de la ruta al mapa o en llista.	

DISSENY CONCEPTUAL

ELS ESCENARIS D'ÚS

Els escenaris d'ús en el nostre cas són molt comuns en els diversos perfils d'usuaris, ja que tots els casos previstos segueixen la mateixa rutina d'ús. Les necessitats, al cap i a la fi, són les mateixes i són les que l'aplicació vol solucionar.

<p>Perfils d'usuaris: Shopper, comprador domèstic</p>
<p>Context d'ús: Uns moments abans d'emprendre la ruta, o al sortir de casa. Són més flexibles ja que solen ser distàncies curtes i no donen tanta importància al temps, tenen menys pressa.</p>
<p>Quin o quins objectius té: Programar la ruta per tenir una previsió.</p>
<p>Les tasques que duu a terme per l'assoliment dels seus objectius:</p> <ul style="list-style-type: none"> • Cercar el punt de pas pel nom o direcció. • Afegir el punt de pas a una llista. • Possibilitat de canviar el tipus de transport. • Possibilitat de canviar quines seran les posicions d'origen i destinació. • Cercar la ruta més ràpida. • Mostrar el resultat de la ruta al mapa o en llista.
<p>Les seves necessitats d'informació: Vol saber quin és el camí més ràpid que visiti tots els comerços afegits, per tal de saber a quin punt ha de dirigir-se primer i fer-se una previsió de la ruta. Normalment solen fer el camí a peu agafant el camí més curt.</p>
<p>Les funcionalitats que necessita: Ha de tenir el GPS engegat, ha de fer una cerca d'indrets, ha de fer la optimització de la ruta i s'ha de tenir una visió del mapa on senyali la posició actual, els punts de pas i la ruta calculada.</p>
<p>Com desenvolupa aquestes tasques: Mitjançant una interfície senzilla amb icones clares i deductives, amb un protagonisme especial en el mapa, que és la pantalla principal de l'aplicació.</p>

Perfils d'usuaris: Repartidors
Context d'ús: Abans d'emprendre la ruta, al començar la jornada. Són els que més volen optimitzar el temps.
Quin o quins objectius té: Programar la ruta per tenir una previsió.
Les tasques que duu a terme per l'assoliment dels seus objectius: <ul style="list-style-type: none"> • Cercar el punt de pas pel nom o direcció. • Afegir el punt de pas a una llista. • Possibilitat de canviar el tipus de transport. • Possibilitat de canviar quines seran les posicions d'origen i destinació. • Cercar la ruta més ràpida en vehicle. • Mostrar el resultat de la ruta al mapa o en llista.
Les seves necessitats d'informació: Vol saber quin és el camí més ràpid circulant en vehicle que visiti tots els punts de pas afegits, per tal de saber a quin punt ha de dirigir-se primer i fer-se una previsió de la ruta.
Les funcionalitats que necessita: Ha de tenir el GPS engegat, ha de fer una cerca d'indrets, ha de fer la optimització de la ruta i s'ha de tenir una visió del mapa on senyali la posició actual, els punts de pas i la ruta calculada.
Com desenvolupa aquestes tasques: Mitjançant una interfície senzilla amb icones clares i deductives, amb un protagonisme especial en el mapa, que és la pantalla principal de l'aplicació.

Perfils d'usuaris: Turistes
Context d'ús: Abans d'emprendre la ruta, unes hores/dies abans a casa o a l'hotel.
Quin o quins objectius té: Programar la ruta per tenir una previsió.

Les tasques que duu a terme per l'assoliment dels seus objectius:

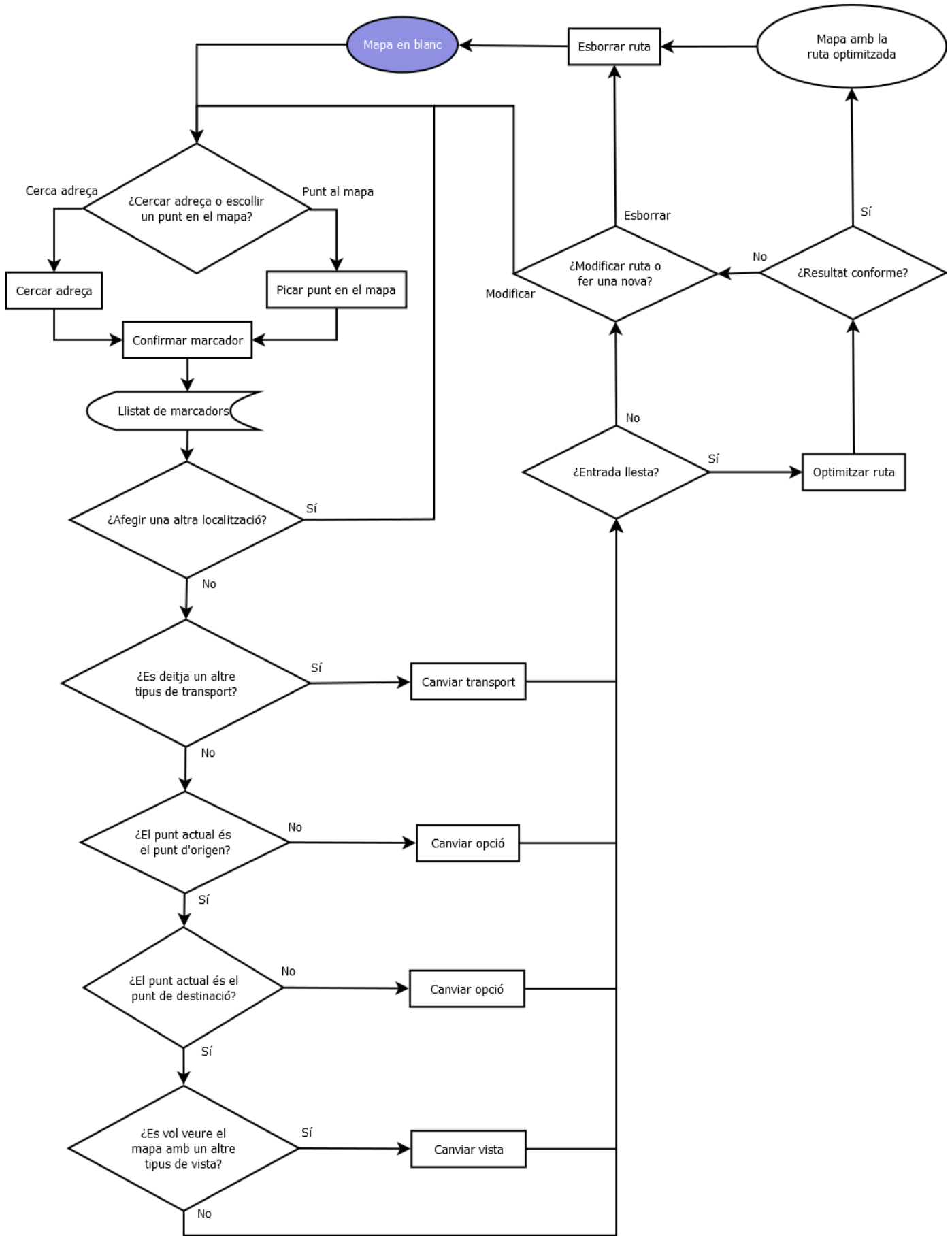
- Cercar el punt turístic pel nom o direcció.
- Afegir el punt turístic a una llista.
- Possibilitat de canviar el tipus de transport.
- Possibilitat de canviar quines seran les posicions d'origen i destinació.
- Cercar la ruta més ràpida.
- Mostrar el resultat de la ruta al mapa o en llista.

Les seves necessitats d'informació: Vol saber quin és el camí més ràpid que visiti tots els punts de pas afegits, per tal de saber a quin punt ha de dirigir-se primer i fer-se una previsió de la ruta. Moltes vegades van en bici si les distàncies són grans per fer-les a peu.

Les funcionalitats que necessita: Ha de tenir el GPS engegat, ha de fer una cerca d'indrets, ha de fer la optimització de la ruta i s'ha de tenir una visió del mapa on senyali la posició actual, els punts de pas i la ruta calculada.

Com desenvolupa aquestes tasques: Mitjançant una interfície senzilla amb icones clares i deductives, amb un protagonisme especial en el mapa, que és la pantalla principal de l'aplicació.

FLUXOS D'INTERACCIÓ

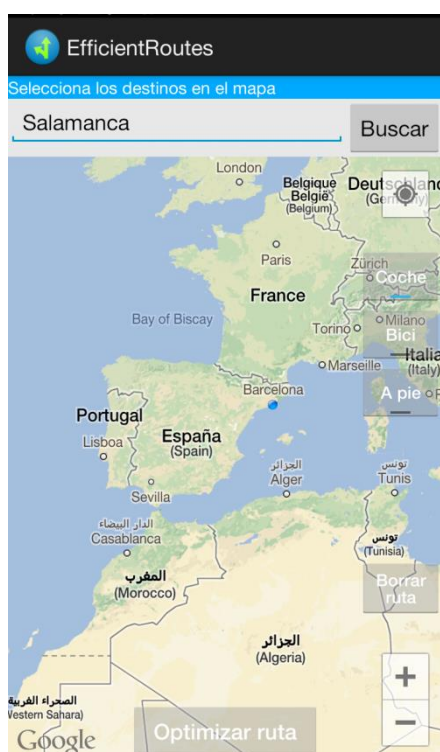


DISSENY GRÀFIC

INTERFÍCIE GRÀFICA



Pantalla inicial



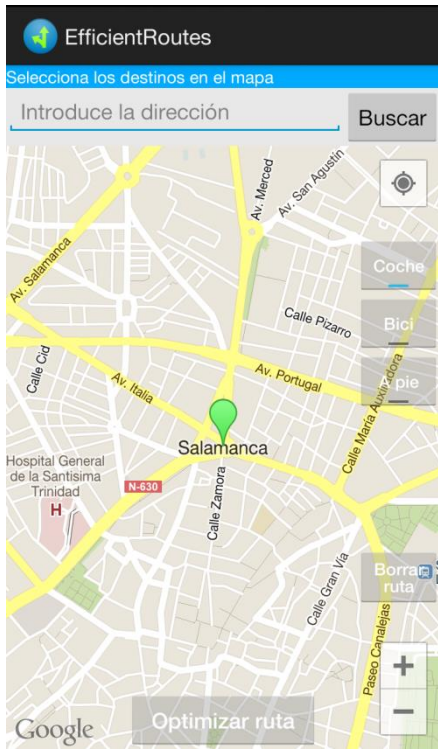
Cerca d'adreça per paraules



Resultats al mapa



Consultem adreces dels resultats



Localització confirmada



Localització mitjançant punt al mapa

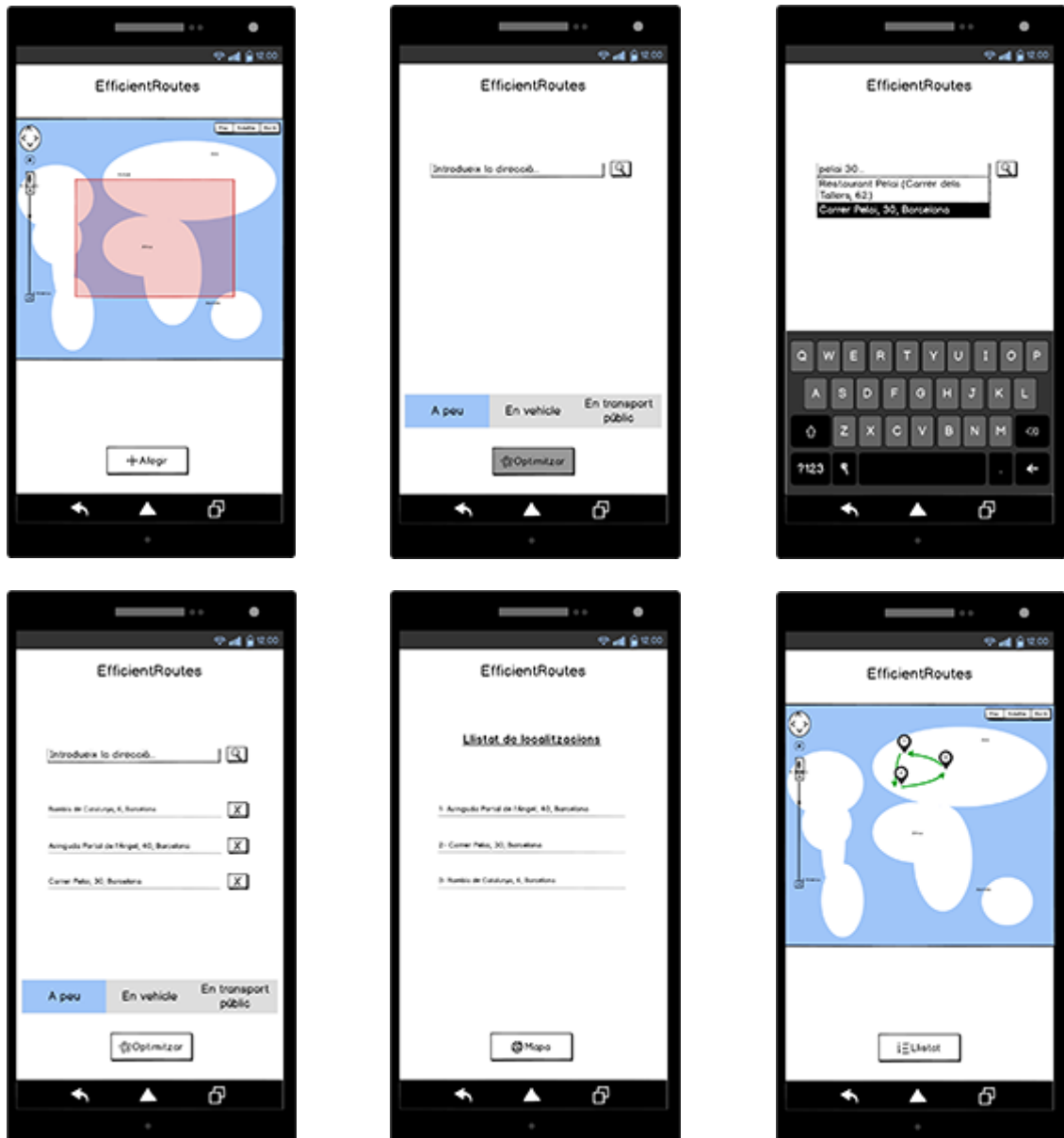


Ruta optimitzada



Opcions de menú

DIFERÈNCIES AMB ELS PROTOTIPUS INICIALS



S'han fet alguns canvis del prototipus inicial que es va entregar per suggeriments del consultor i per variacions de la funcionalitat de l'aplicació.

Aquests canvis en l'aplicació i que afecten gairebé a totes les pantalles previstes principalment són dos: per consell del consultor s'ha fet la cerca d'adreces en la mateixa pantalla del mapa sense canviar a una altra específica per això, ja que així li és més còmode a l'usuari, li redueix la càrrega cognitiva, simplificaria la navegació i és més còmoda d'utilitzar. Ara la pantalla del mapa cobra molt més importància i sempre és visible.

El llistat de localitzacions no s'ha implementat de forma de llistat per falta de temps, quedant només mostrades en el mapa en forma de marcadors confirmats.

ARQUITECTURA DE L'APLICACIÓ

CLASSES

ActualitzaMapa

És una classe simple amb un mètode estàtic que actualitza el mapa netejant-lo de marques i deixant visible només els marcadors que s'han seleccionat prèviament.

> **static void actualitzaMapa(GoogleMap, List<Marker>)**

El mètode únic que neteja el GoogleMap i reinserta al mapa els marcadors seleccionats i guardats en el seu moment. Serveix per eliminar entrades no confirmades, o sigui no volgudes.

BuscaDireccio

Classe heretada d'AsyncTask que s'encarrega de donar els resultats de la cerca d'una adreça mitjançant una crida al web service del Geocoder de l'API de localització de Google. Ho fa en un fil en segon pla, fora de la User Interface, i retorna els resultats més probables coincidents amb les paraules d'entrada amb un màxim de quatre resultats, per tal de no marejar a l'usuari omplint el mapa amb resultats.

Els resultats es mostren en el mapa en forma de marcador vermell. Pot trobar qualsevol adreça o nom d'equipament o establiment a nivell mundial per tal de no restringir l'ús en un altre país, tant per si es ús turístic (es un cas d'ús) com si és per necessitat (un repartidor internacional, per exemple, un altre cas d'ús) o com, simplement, si és per internacionalitzar l'aplicació i que tothom la pugui utilitzar en el seu país.

> **List<Address> doInBackground(String...)**

Funció sobreescrita heretada d'AsyncTask que s'executa en segon pla. Llegeix l'entrada que es vol cercar (adreça amb carrer i número, una localització, un nom d'un equipament, una carretera, etc.) i retorna una llista d'adreces coincidents amb un màxim de quatre resultats.

> **void onPostExecute(List<Address>)**

Mètode sobreescrit heretat d'AsyncTask. S'encarrega de mostrar en el mapa els marcadors vermells sobre les localitzacions abans trobades, si n'hi han hagut. En cas contrari mostra un missatge conforme la cerca no ha tornat resultats.

Els marcadors mostraran en primera instància en pantalla de forma que es vegin tots, i cadascun d'aquests tindrà la seva adreça emmagatzemada.

CarregarURL

Classe heretada d'AsyncTask que s'encarrega de fer la connexió HTTP d'una URL en forma de String que li passem. Aquesta URL farà una connexió a l'API de direccions de Google, amb els paràmetres corresponents a cada cerca, i ens retornarà una cadena de caràcters que s'haurà d'interpretar en una altra classe creada per a tal.

Aquesta classe genera missatge sobreposat en pantalla que informa a l'usuari de que hi ha un procés en segon pla pendent d'una resposta visible. Al ser la que provoca el principi de les peticions més costoses i nombroses i les quals s'executen en segon pla, és convenient avisar ara a l'usuari d'una possible espera per la seva pròxima interacció.

> **void onPreExecute()**

Mètode sobreescrit heretat d'AsyncTask. Mostra el missatge que informi a l'usuari de que l'aplicació està ocupada i ha d'esperar per tornar a interaccionar amb l'aplicació.

> **String doInBackground(String...)**

Funció sobreescrita heretada d'AsyncTask que s'executa en segon pla. Una funció que s'encarrega de la connexió en si. Mitjançant una URL en String, prova d'obrir la connexió amb protocol HTTP, guarda la resposta del servidor, també en format String, i tanca la connexió. Com que les connexions HTTP poden llençar excepcions, en cas que hi hagués alguna la recollirem i mostrarem el missatge d'error.

> **void onPostExecute(String)**

Mètode sobreescrit heretat d'AsyncTask. Després d'obtenir la resposta de l'API de direccions de Google en forma de String, haurem d'interpretar-la per tal de manejar i d'extreure la informació útil. Per això cridarem a la classe CreadorGoogleJSON passant-li aquesta resposta com a paràmetre.

CreadorGoogleJSON

Classe heretada d'AsyncTask que ens llegirà la resposta enviada per l'API d'adreces de Google i la transformarà al format JSON per després accedir a les seves claus i valors, d'on obtindrem la informació que voldrem.

> **List<List<HashMap<String, String>>>doInBackground(String...)**

Funció sobreescrita heretada d'AsyncTask que s'executa en segon pla. Li passem la resposta de Google a la nostra petició i, mitjançant un interpretador de rutes de Google que ens hem creat a la classe InterpretadorGoogleJSON, crea un llistat de llistats de parells clau-valor que nosaltres podrem llegir.

> **void onPostExecute(List<List<HashMap<String, String>>>)**

Mètode sobreescrit heretat d'AsyncTask. Un cop tenim els parells clau-valor de la ruta de Google en llistes els enviem a la classe DibuixaRuta, que farà la representació gràfica de la ruta.

InterpretadorGoogleJSON

Classe que s'encarrega d'analitzar i interpretar l'objecte JSON que envia CreadorGoogleJSON amb els parells clau-valor provinents de la crida a l'API de Google.

> **List<List<HashMap<String, String>>> interpretadorGoogleJSON(JSONObject)**

Funció que recorre l'objecte JSON corresponent a una ruta, l'analitza i el descompon en parells clau-valor. L'API de direccions de Google retorna la ruta total formatjada en forma de arbre, subordinant-se en rutes, branques i passos. En primer lloc revisa l'etiqueta de rutes, d'on agafa el nombre de branques i les revisa en bucle sumant les distàncies i temps que van acumulant aquestes. A la vegada que revisa les branques, per cada una fa un bucle per revisar cada punt de coordenada els quals formen la branca.

> **List<LatLng> decodificaPolilinia(String)**

Aquesta funció demana les coordenades en String que retorna l'interpretador d'objectes de ruta en format JSON i les converteix en coordenades en format LatLng, amb una latitud i una longitud.

DibuixaRuta

Classe amb una sol mètode que ens representa gràficament la ruta o ens diu si no es possible mostrar-la (ja sigui per la no introducció de destinacions intermitges o que la ruta no és possible calcular).

L'objectiu és que l'usuari vegi de la manera més intuïtiva possible la següent informació: primer, a grans trets, l'ordre de les localitzacions seleccionades per on ha de passar la ruta, i en segon terme, la ruta en si per on ha d'anar. Per esbrinar l'ordre, davant la impossibilitat que m'ha sorgit de numerar els marcadors (com a primera idea i més aclaridora) he optat per fer una degradació del color blau en la ruta segons el sentit de la marxa. D'aquesta manera, l'usuari haurà d'escollir primer el camí amb un blau més clar (teòricament quasi blanc), i cada cop que passem per una destinació inserida el camí es va tornant en un blau més pur en la mateixa proporció, essent l'últim tram de ruta d'un color blau real (0,0,255 en format RGB) passant pel cyan.

> void dibuixaRuta()

Mètode únic de la classe DibuixaRuta encarregat d'agafar coordenada a coordenada de la resposta donada per l'API de Google i muntar polilínies d'un color, amplada desitjats i traçat determinat per les coordenades. Per tal de fer més fàcil la visualització general s'enquadra tota la ruta dins de la pantalla. En cas que no hi hagués un ruta per mostrar, s'indicarà amb un missatge superposat.

Aquest mètode també s'encarrega d'esbrinar quan la ruta passa per una destinació desitjada per tal de donar-li un toc de blau més pur a la línia dibuixada, així com també gestiona les distàncies recorregudes i temps de recorregut estimats i els mostra a la part superior del mapa.

MainActivity

Classe principal del programa la qual es cridada al principi de l'execució de l'aplicació.

> **void onCreate(Bundle)**

El mètode onCreate(), sobreescrit de la classe heretada FragmentActivity, és invocat quan l'activitat s'inicia. Primer de tot, comprova que estigui Google Play Services instal·lat en el dispositiu ja que l'aplicació el fa utilitzar i es un component bàsic. En cas que no hi estigués, informa a l'usuari i li proposa la descàrrega proporcionant un enllaç directe a la descàrrega de la utilitat en el Google Play.

Un cop ha verificat que hi és instal·lat, inicia algunes variables, enllaça objectes amb vistes, fa les assignacions pertinents de escoltadors (listeners) i de funcionalitats a alguns components, com per exemple activar la localització del dispositiu o que la pantalla d'inici s'enquadri centrant-se (és purament pràctic i per una qüestió de comoditat per aquest cas, en cas general no es posarien delimitadors). En el següent apartat parlarem dels escoltadors que s'han assignat.

> **boolean onCreateOptionsMenu(Menu)**

La funció típica d'Android onCreateOptionsMenu() sobreesciu la mateixa funció de la classe heretada FragmentActivity i serveix per crear les diferents opcions que es poden accedir mitjançant el botó d'opcions del terminal.

> **void onMyLocationChange(Location)**

Es tracta d'un mètode que sobreesciu el mateix mètode de la classe escoltadora OnMyLocationChangeListener() i ens guarda la posició actual constantment en una variable global.

> **boolean onOptionsItemSelected(MenuItem)**

És un mètode molt comú, com tots els que sobreescriven la classe Activity. Assigna les accions a fer segons el botó d'opcions que sigui pulsat. En el nostre cas tenim tres opcions:

- Alternar el tipus de vista del mapa.
- Fixar la posició actual com a posició d'origen de la ruta.
- Fixar la posició actual com a posició final de la ruta.

> **void alternarVista()**

Aquest mètode és el que s'executa amb la primera opció del llistat del menú: alterna el tipus de vista del mapa. Tenim els tipus mapa de carrers, mapa de satèl·lit, mapa de relleu i mapa híbrid (de satèl·lit però amb carrers marcats).

> **String getDirectionsUrl(LatLng, LatLng, List<LatLng>)**

Aquesta funció retorna una cadena de caràcters muntat per enviar al servei web de l'API de direccions de Google. Hem d'entrar a la funció la posició d'origen i la de destinació de la ruta i les localitzacions intermitges, si ni hagueren. Tots tres en format de coordenada correcte LatLng. La funció envia com a paràmetres, entre d'altres, el tipus de transport que s'ha escollit per fer la ruta i demana que la resposta sigui interpretable en format JSON, sent XML una altra opció menys recomanada.

> **void optimizaRuta(List<Marker>)**

Aquesta es el mètode que comença la crida, des de la classe principal, al *core business* de l'aplicació. Assignant la posició d'origen i final i amb la llista de marcadors que l'usuari ha decidit, cridem a la funció que munta l'URL de consulta a Google i la passem a la classe que s'encarrega de fer la connexió HTTP. Cal puntualitzar que en cas que l'usuari desmarqui la opció de prendre la posició actual com a origen, la posició

d'origen serà la del primer marcador que estableixi, per lògica, ja que l'usuari haurà de saber forçosament on voldrà començar llavors.

> **OnClickListener clickEnBuscar**

Definim un escoltador pel botó 'Buscar', el qual manarà buscar la direcció que hi és escrita al camp de text contigu i associat, si no està buit.

> **OnClickListener clickEnOptimitzar**

Definim un escoltador pel botó 'Optimitzar', el qual manarà fer la optimització de la ruta en la situació que es trobi en aquell moment, actualitzant el mapa. Si no pot fer la ruta comunica a l'usuari el motiu.

> **OnClickListener clickEnEsborrar**

Definim un escoltador pel botó 'Borrar', el qual esborra del mapa les rutes i marcadors que hi hagin i neteja la memòria dels marcadors seleccionats fins aleshores.

> **OnMapClickListener clickEnMapa**

Es tracta d'un escoltador que, en seleccionar un punt en el mapa, es crea un marcador vermell en la mateixa coordenada, que recollirà i mostrarà la informació de l'adreça que hi hagi en aquell punt del mapa. En cas que la coordenada no fos un punt vàlid (per exemple, aigua) retorna un missatge d'informació a l'usuari sense mostrar el marcador.

> **OnMarkerClickListener clickEnMarcador**

Es tracta d'un escoltador que, en pitjar un marcador mostrat en pantalla a causa d'una

cerca per paraules, mostra un globus d'informació amb la adreça associada a la coordenada on està.

> **OnInfoWindowClickListener clickInfoMarcador**

Es tracta d'un escoltador que confirma un marcador generat prèviament i que ha mostrat la seva adreça. Això vol dir que l'usuari està conforme amb aquella adreça i vol guardar aquella coordenada perquè es la desitjada. Per fer-ho, l'usuari ha de pitjar el globus de l'adreça i automàticament el marcador es tornarà verd a mode de confirmació.

> **OnCheckedChangeListener canviToggleButton**

Aquest mètode sobre escriu el de l'escoltador `OnCheckedChangeListener()` i fa comportar-se als tres `ToggleButtons` com si foren un `RadioButton`, de manera que si hi ha un seleccionat no poden estar els altres, i al menys hi ha d'haver un sempre marcat. Això respon a una preferència visual i és una decisió estrictament estètica, ja que per la forma general de la interfície de l'aplicació veia més convenient que els botons fossin de l'estil d'un `ToggleButton`.

AVALUACIÓ

Per fer l'avaluació de l'aplicació, i tal i com estableix el mètode de Disseny Centrat a l'Usuari, he fet provar-la a mida que anava fent el desenvolupament per tal d'adaptar l'aplicació a la manera habitual de fer-la usar.

Després de l'explicació introductòria sobre que va l'aplicació, he observat com es desenvolupaven, que intuïcions tenien i quins errors havia de millorar o de solucionar en cas d'error greu.

PROVA D'ERRORS

He trobat un seguit d'errors, trobats forçadament o no, que han estat corregits:

- ✓ Seleccionar un punt sobre l'aigua.
- ✓ Fer la optimització de ruta sense cap localització introduïda.
- ✓ Fer la optimització de ruta sense ser la posició actual el punt d'origen o punt final i només una localització introduïda.
- ✓ Fer una ruta molt llarga o impossible.
- ✓ Fer una ruta amb bici sense camí bici.

PROPOSTES DE MODIFICACIONS FUTURES

- × Crear un llistat de text de les d'adreces guardades.
- × Gestió de les adreces més directa (canviar-les de posició o esborrar-ne algunes)
- × Numerar els marcadors per ordre de pas.
- × Fer la aplicacions en diferents idiomes.
- × Enllaçar la ruta amb Google Navigation.
- × Poder fer les rutes en transport públic, encara que en això influeix el dia i hora de la ruta, amb lo qual s'hauria d'entrar a l'aplicació.

CONCLUSIONS

L'aplicació és més senzilla del que m'hagués agradat a causa de problemes familiars que m'han sorgit al llarg del projecte i a imprevistos que m'han fet dedicar menys temps del que m'hagués agradat. Encara així, aquest projecte m'ha aportat dues experiències en una.

Per una banda, he fet una aplicació "sencera" des del principi en solitari, passant per totes els cicles de vida de l'aplicació, a falta del manteniment que pogués sorgir. Abans només havia fet alguna part o algun cicle perquè l'assignatura s'enfocava en aquell cicle en concret, o bé perquè formava part d'un equip i ens dividíem la feina.

Per una altra banda, és el primer apropament que tenia amb Android, i amb un llenguatge de programació per a mòbils en general. Amb la base de Java el lèxic d'Android no és difícil, la dificultat més gran que he trobat ha radicat en les capes de presentació (la interfície gràfica), en l'ús dels XML, l'estructura de l'aplicació i en el multi-thread, ja que, encara que en Java si es poden utilitzar XML i fer multi-threading, no ho havia fet abans.

Encara que ja em motivava la idea d'aprendre i conèixer Android perquè considero que és una tecnologia de futur (com tot el que va implícit a la mobilitat), aquest projecte m'ha suposat un repte constant i m'ha animat encara més seguir investigant i profunditzant d'aquí en davant, intentant desenvolupar les funcionalitats que falten i poden millorar EfficientRoutes i ficar-me, perquè no, en altres projectes interessants que hem sorgeixin.

BIBLIOGRAFIA

- **Head First Android Development** (PDF) *Jonathan Simon*. O'Reilly – Llibre per programadors que s'inicien en Android.
- **Android. Guía para desarrolladores** (Llibre) *Frank Ableson, Charlie Collins, Robi Sen*. Anaya – Llibre dirigit tant a programadors professionals com per aficionats amb base de Java.
- **Disseny centrat en l'usuari** (PDF) *Muriel Garreta Domingo i Enric Mor Pera*. UOC – Document acadèmic.
- **Disseny centrat en l'usuari per a dispositius mòbils** (PDF) *Jordi Almirall López*. UOC – Document acadèmic
- **Android Developers** (Web) Web oficial de Google per als desenvolupadors d'Android, on és tota la documentació del SO i hi són totes les classes que ho componen.
<http://developer.android.com/reference/packages.html>
- **Google Maps Android API v2** (Web) Web oficial de Google amb la documentació de la API que utilitza l'actual versió v2 dels mapes d'Android.
<https://developers.google.com/maps/documentation/android/>
- **Stackoverflow** (Web) Lloc web col·laboratiu on els programadors recorren per plantejar problemes de programació i trobar solucions als mateixos.
<http://stackoverflow.com/>
- **sgoliver.net blog** (Web) Blog amb petits tutorials temàtics d'Android on repasen les principals funcionalitats de cada tema.
<http://www.sgoliver.net/blog/>

- **Android: Programación de aplicaciones para móviles** (Web) Pàgina web pertanyent a la Universitat Politècnica de València amb tutorials molt complets sobre Android, amb una àmplia gama de temes.

<http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android>

- **Knowledge by Experience** (Web) Blog amb tutorials i exemples de d'aplicacions d'Android, centrant-se en funcionalitats molt concretes.

<http://wptrafficanalyzer.in/blog/>