



App QualiMove

Xavier Massotti Godino

Enginyeria Tècnica d'Informàtica de Gestió

Joan Vicent Orega Serisuelo, Jordi Almirall Lopez

Data Lliurament
08/01/2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Copyright © Xavier Massotti Godino

FITXA DEL TREBALL FINAL

Títol del treball:	<i>App QualiMove</i>
Nom de l'autor:	<i>Xavier Massotti Godino</i>
Nom del consultor:	<i>Joan Vicent Orenge Serisuelo, Jordi Almirall Lopez</i>
Data de lliurament (mm/aaaa):	<i>01/2016</i>
Àrea del Treball Final:	<i>Desenvolupament aplicacions dispositius mòbils (Android)</i>
Titulació:	<i>Enginyeria Tècnica d'Informàtica de Gestió</i>
Resum del Treball (màxim 250 paraules):	
<p>Aplicació Android de presentació de rutes o itineraris en transport públic d'una ciutat amb informació de la congestió o quantitat de gent a les estacions o parades trobats a la ruta demanada.</p> <p>El sistema es compon de un servei WEB que processa imatges de càmeres a les estacions i realitza un càlcul de la seva saturació per, posteriorment, donar una ruta a l'aplicació client Android amb aquesta informació de les estacions.</p>	

Abstract (in English, 250 words or less):

Android Application for create routes of public transport with congestion level information of people in the stations or stops found on the route requested.

The system consists of a web service that processes images from cameras at stations and performs a calculation of its saturation to create a route for the client application Android with this information.

Paraules clau (entre 4 i 8):

Imatges, mapa, ruta, imagen, transport, images, maps, route

Índex

<i>Desenvolupament aplicacions dispositius mòbils (Android)</i>	ii
1. Introducció	3
1.1 Context i justificació del Treball	3
1.2 Objectius del Treball	4
1.3 Enfocament i mètode seguit	4
1.4 Planificació del Treball	5
1.5 Productes obtinguts	6
1.6 Descripció memòria	6
2. Servei WEB	7
2.1 Funcionament i flux del sistema	8
2.2 Classes Java de servei	8
2.3 Servei de test	8
2.4 Servei de foto	9
2.5 Servei de ruta	9
2.6 Diagrames de classes	10
3. App Android	11
4. Càlcul de saturació	17
4.1 Preprocessat	18
4.2 Procés "fast"	18
4.3 Procés "partícules"	19
5. Càlcul de ruta	21
6. Conclusions	22
6.1 Assoliment d'objectius	23
6.2 Desviació de planificació i metodologia	23
6.3 Aplicacions futures	23
4. Glossari	24
5. Bibliografia	24
6. Annexos	25
Fitxers i desplegament d'App	25
Estructura fitxer xml de ruta	26
Estructura fitxer xml d'enllaç entre nodes	26
Paràmetres preprocessat d'imatges (<photoProcess>)	27
Paràmetres procés càlcul saturació (<parameterPhotoProcess>)	27
Exemple de fitxer de rutes xml	27

Llista de taules

Taula 1: Tasques per hores.....	6
Taula 2: Terminis projecte	6
Taula 3: Proves tipus procés “fast”	19
Taula 4: Proves tipus procés “particles”	20

Llista de figures

II·lustració 1: Flux general del sistema	5
II·lustració 2: Diagrama de classes de servei WEB.....	10
II·lustració 3: Pantalla inicial	11
II·lustració 4: Menú funcions.....	12
II·lustració 5: Accés a configuració	12
II·lustració 6: Opcions configuració.....	13
II·lustració 7: Valors configuració	13
II·lustració 8: Valors configuració	13
II·lustració 9: Menú funcions.....	14
II·lustració 10: marcadors inicials	14
II·lustració 11: Cercar per text 1	15
II·lustració 12: Cercar per text 2	15
II·lustració 13: Cercar per text 3	15
II·lustració 14: Ruta 1	16
II·lustració 15: Ruta 2	16
II·lustració 16: Ruta 3	17

1. Introducció

1.1 Context i justificació del Treball

Actualment la mobilitat amb transport públic a les grans ciutats és un problema cada vegada major i en augment, encara havent diversos mitjans de transport, a certes hores hi ha massa confluència de gent per poder donar el servei adequat o per poder donar el servei a tothom i moltes vegades s'arriba a col·lapsar durant cert temps.

Aquest problema que es dona a totes les grans ciutats té solucions que solen ser molt costoses, reduir freqüència dels medis, posar més unitats de transport, afegir una nova variant de cert transport, etc.

Per altre banda les apps que poden trobar [\[1\]](#), ens donen rutes per arribar a destí o informació dels llocs de transport, també hi ha que donen la saturació del transit rodat o com arribar des de diferents mitjans de transport, però no hi ha cap que doni com de ple de gent es troba una estació o un autobús, és a dir, la saturació d'un transport públic.

Per el dia a dia, evidentment no servirà aquesta App si la trobes, però si m'he de moure puntualment, una gestió, anar al cinema, etc. llavors si que m'agradaria disposar d'aquesta informació. Si anés de turista a una gran ciutat, en aquest cas, crec que la trobaria molt necessària, no vull trobar-me amb un metro ple de gom a gom per anar a veure un lloc i esperar-me a veure passar trens fins que sigui "acceptable" la quantitat de gent que puja, si ho hagués sabut, canviaria el lloc a visitar o l'hora o el mitjan de transport.

Per aquest motiu la intenció d'aquest projecte és proveir una App que informi de la saturació del transport per arribar a destí i que puguis triar la forma més convenient per realitzar el trajecte que vols fer.

1.2 Objectius del Treball

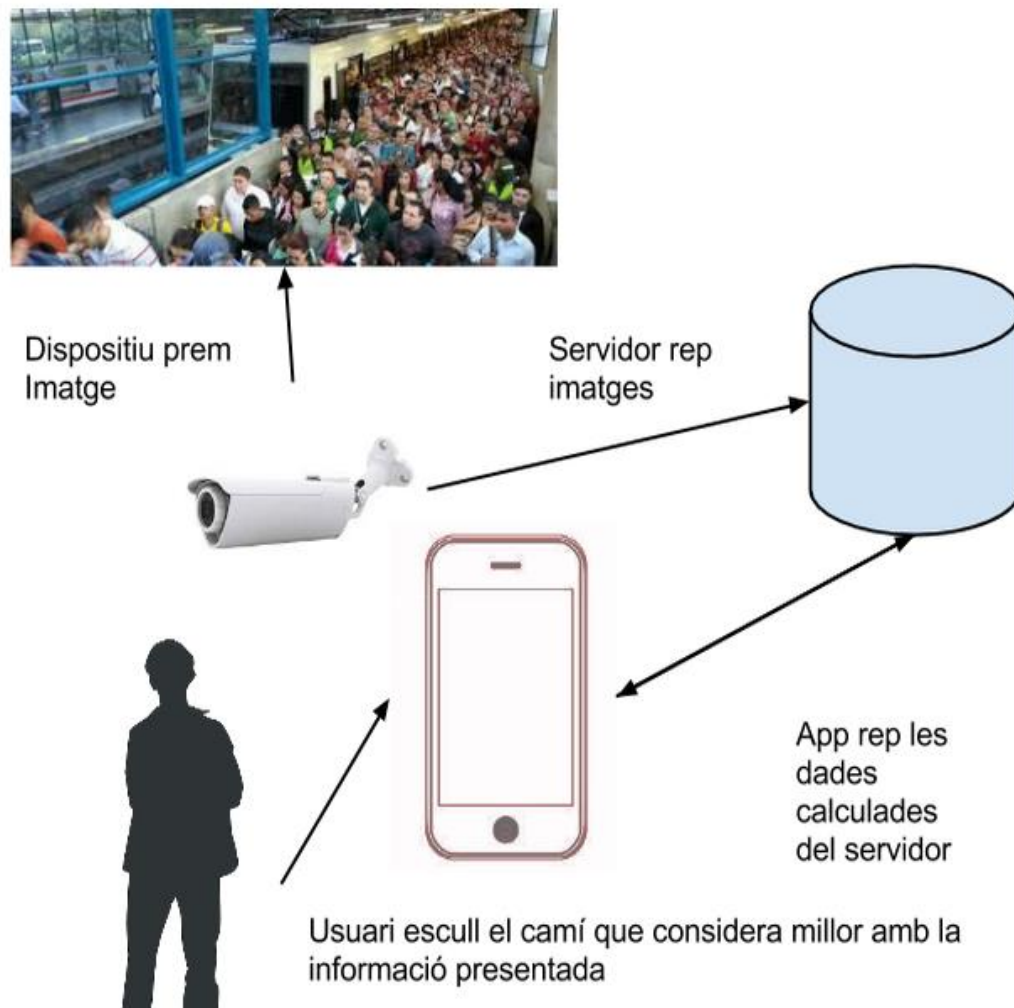
- Servei per calcular la saturació d'un transport per tal de donar dades del mateix a un terminal client.
- App de client que amb les dades del servei poguí donar una informació de saturació del transport a l'itinerari a mostrar.

1.3 Enfocament i mètode seguit

El principal problema és com obtenir les dades de saturació, el control per entrada en molts llocs dóna més d'un lloc final (estació) possible i no serveix, els control per "comptadors" en certs llocs també porta problemes en el moviment i cost elevat. La solució que es planteja és mitjançant càmeres de fotos a estacions de tren i autobusos o el transport a controlar, és una solució amb un cost i manteniment molt econòmic i prou fàcil d'implementar amb les càmeres ip actuals.

Per donar una visió general del sistema, primer la càmera enviarà una foto al servidor o el servidor s'encarrega de realitzar la petició a la càmera per recollir la foto. El servidor s'encarrega d'actualitzar les dades utilitzant un algorisme aplicat a la foto per saber el nivell de saturació, actualitzant un vector amb la informació per enviar al client. El client (App) recollirà la informació i realitzarà l'itinerari en el terminal per tal de fer la presentació amb la informació afegida del transport.

L'idea del projecte és equilibrar la feina entre el servidor i el terminal del client, aprofitant que avui dia els terminals són prou potents i amb serveis adaptats (Google Maps) per tal d'obtenir una aplicació de qualitat.



Il·lustració 1: Flux general del sistema ¹

Tant per el sistema del servei com per l'App s'ha seleccionat la tecnologia Java, per la qualitat de documentació a l'hora de poder desenvolupar, com també, el fet que les eines són gratuïtes.

El sistema del servei es desenvoluparà sobre l'eina Eclipse per desplegar en un servidor d'aplicacions Tomcat i per l'App s'utilitzarà l'eina Android Studio.

1.4 Planificació del Treball

¹ Imatges d'editor o tret de free images [\[2\]](#)

La instal·lació de les eines i organització inicial no es compta, tal com s'indica a la taula de terminis es compta a partir del disseny.

Previsió d'hores per les principals tasques, les F representa una hora en festiu.

Tasca / Hora laboral – (F)estiu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Servei servidor														F	F	F	F								
Algorisme tractament foto								F	F	F															
App interface																			F	F	F	F	F	F	F
App serveis Google																		F	F	F	F	F			
App connexió i càlcul dades												F	F	F	F										
Memòria														F	F	F	F	F							
Vídeo presentació								F	F	F															

Taula 1: Tasques per hores

Previsió de terminis del projecte.

	01/10/2015	15/10/2015	28/10/2015	12/11/2015	26/11/2015	10/12/2015	16/12/2015	24/12/2015	07/01/2016
Disseny									
Impl. servei servidor									
Impl. app Android									
Memòria									
Vídeo presentació									

Taula 2: Terminis projecte

1.5 Productes obtinguts

- Memòria
- Fitxer WAR per desplegar sobre un servidor d'aplicacions de Servlets (Tomcat)
- Projecte Eclipse de servei WEB
- APK d'App Android per instal·lar manualment.
- Projecte Android Studio d'App.
- Vídeo presentació de projecte

1.6 Descripció memòria

Per tal de donar una descripció detallada, començarem amb les limitacions que s'han trobat per realitzar el projecte com tal, a continuació una descripció detallada del servei de Servlets, amb una visió general de com funciona. El següent és la descripció de l'App Android, per donar la visió del que trobarà el client.

Els capítols següents trobaren quins mètodes s'han incorporat per realitzar el càlcul de saturació a partir de les fotos i el sistema de crear la ruta a partir de les coordenades donades pel client.

Com últim a les conclusions, trobarem la reflexió i anàlisi d'aquest projecte així com les limitacions trobades i millores a realitzar en el futur.

2. Servei WEB

El sistema del servei WEB funciona amb tecnologia Java, en un servidor d'aplicacions amb suport de Servlets. Implementat sobre Amazon AWS en un servidor Tomcat versió 8 amb Java versió 8.

Aquest servei utilitza la llibreria JSON <http://www.json.org/java/index.html> per tractar cadenes en format JSON d'ús lliure i codi disponible a <https://github.com/douglascrockford/JSON-java> i la llibreria ImageJ <http://imagej.net/Welcome> plataforma de processament científic d'imatge open source amb llicència BSD-2.

Aquest servidor realitza tres serveis bàsics, un servei de test, un servei de registre i càlcul de saturació a partir d'una imatge i un servei de peticions de ruta per l'app Android.

Per aquest prototip s'han definit tres trajectes de transports públics per realitzar les proves, totes de la ciutat de Barcelona, dos de metro, L1 (Hospital de Bellvitge - Fondo) i L2 (Paral·lel - Badalona Pompeu Fabra) i una d'autobús V17 (Carmell – Port Vell)

2.1 Funcionament i flux del sistema

- En el mètode d'inicialització, es carreguen les dades a partir de fitxers xml, creant els objectes de cerca i gestió dels nodes a memòria.
- En el mètode de servei que rep les peticions get i post, crida al mètode indicat per donar resposta segons el paràmetre "service", aquest mètode és multipeticició i s'ha tingut en compta a l'hora de fer el serveis.
- El servei de test, retorna una text amb les rutes carregades.
- El servei de foto, es connecta a una url per demanar una imatge, registra i realitza el càlcul de saturació del node.
- El servei de petició de client, calcula una ruta en base a les coordenades demanades i retorna una llista dels nodes amb format JSON.

2.2 Classes Java de servei

ServeiQM: Servlet Java que gestiona la carrega inicial i les peticions i el control bàsic de les mateixes.

NodeQM: Dades de node.

RouteQM: Dades de ruta.

ManageRQM: Classe principal de dades a memòria de les rutes.

UtilRQM: Utilitats de gestió i càlcul de les rutes, creada com una extensió separada de ManagerQM per evitar conflictes de variables locals entre fils d'execució.

CrtImagesQM: Classe encarregada del procés de registre i càlcul de les fotos.

2.3 Servei de test

Url: <http://default-environment-nza5kmp3ms.elasticbeanstalk.com/serviceqm?service=test>

Aquest servei bàsicament es per comprovar que el sistema està funcionant i que s'han carregat les rutes i nodes enregistrats en els fitxers xml, donant una sortida bàsica de totes les rutes i nodes associats en una pàgina de text.

2.4 Servei de foto

Url:<http://default-environment-nza5kmp3ms.elasticbeanstalk.com/serviceqgm?service=photo&route=routeName&node=nodeName>

Aquest servei rep el nom de ruta i node com petició, agafa els paràmetres adjacents del node per realitzar una petició a l'URL de la càmera IP o servei per recollir la imatge, després segons els paràmetres de procés indicats al node, enregistra si cal, realitza modificacions si cal i envia al procés indicat per determinar la saturació del node.

2.5 Servei de ruta

Url:<http://default-environment-nza5kmp3ms.elasticbeanstalk.com/serviceqgm?service=route&latori=41.12212&lngori=21.2122&latdest=41.56524&lngdest=21.562354&dist=1000>

El servei rep les coordenades d'inici i final seleccionades pel client i la distància màxima per donar avis si cal. Es crea una ruta en base a aquestes dades, si els nodes més propers són a més d'1,7 kilòmetres el sistema retorna un avis de que no s'ha trobat cap ruta, si troba ruta comprova segons els límits indicats en els paràmetres per donar l'avis de distància al node d'inici o final.

2.6 Diagrames de classes

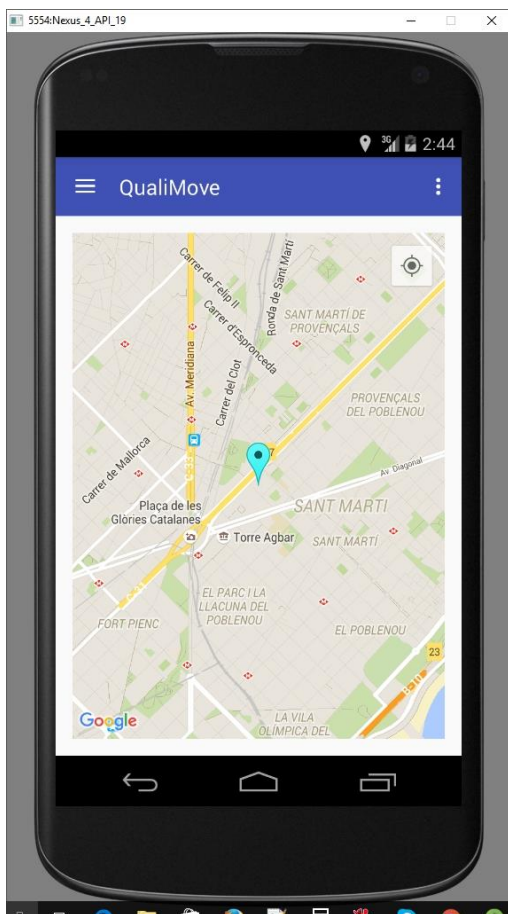


II-lustració 2: Diagrama de classes de servei WEB

3. App Android

Aquesta App està provada amb dispositiu físic amb Android 4.2.2 (API 17) i amb dispositius virtuals (emuladors) fins a la versió d'Android 6.0 (API 23)

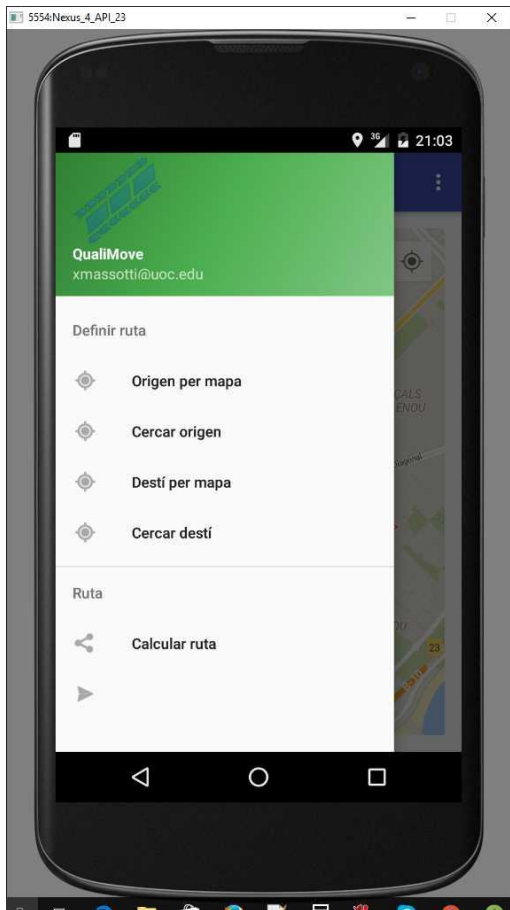
L'app està internacionalitzada amb tres idiomes, català, espanyol i anglès, utilitzant el sistema estàndard d'Android i per tant, facilitant l'ampliació d'idiomes de forma ràpida i senzilla.



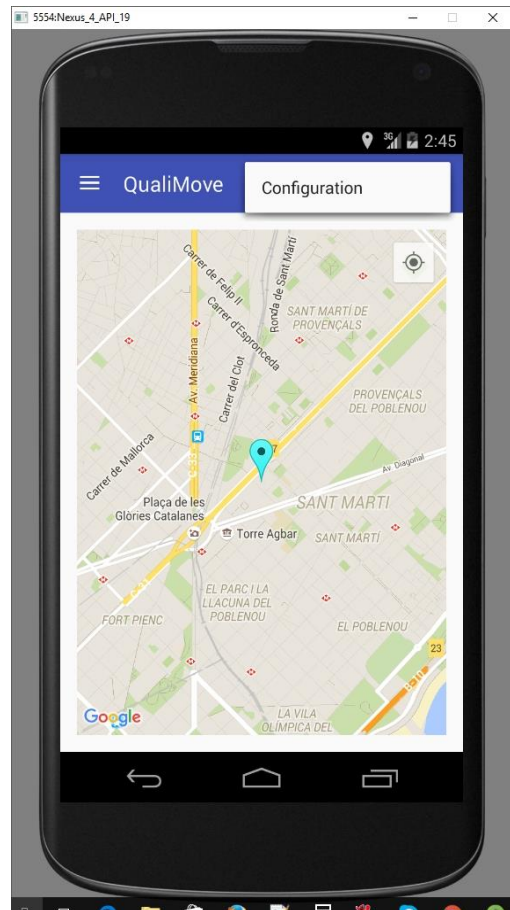
Il·lustració 3: Pantalla inicial

L'inici de l'App és la presentació directa d'un mapa, si permet la localització del dispositiu, automàticament es situarà la localització del mateix al centre de la pantalla, la icona a la part superior dreta del mapa, porta el centre del mapa a la localització actual del dispositiu.

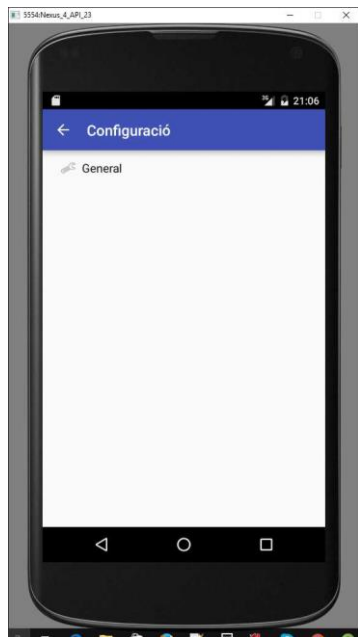
Hi ha dos opcions de menú, l'esquerra pel funcionament i el dret per la configuració.



Il·lustració 4: Menú funcions



Il·lustració 5: Accés a configuració



Il·lustració 6: Opcions configuració

A la configuració solament tenim un apartat i dintre tres opcions.

Un indicador per presentar un avis per distancia major que el nivell indicat.

El nom d'usuari per mostrar, si cal.

El nivell de la distancia màxima al node (inici o final)



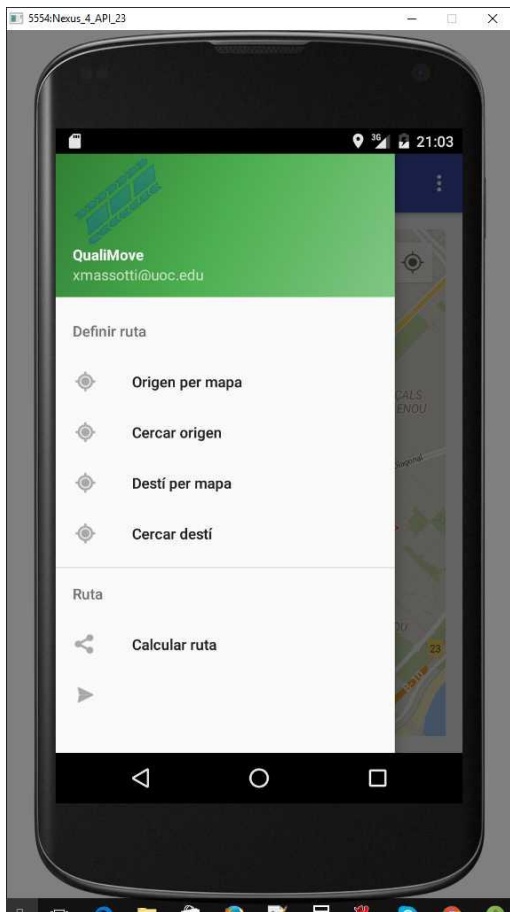
Il·lustració 7: Valors configuració



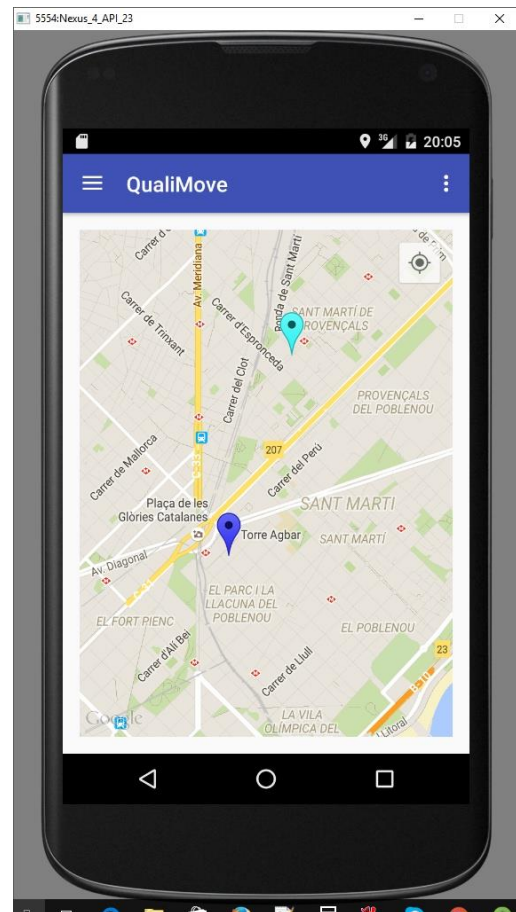
Il·lustració 8: Valors configuració

En el menú de funcionamiento, disponem de la situació d'origen i final per mapa, si seleccionen aquestes opcions mostra el mapa i al clicar sobre ell fixa el marcador, origen o final.

El marcador d'origen és blau clar i el de destí blau fosc.



Il·lustració 9: Menú funcions

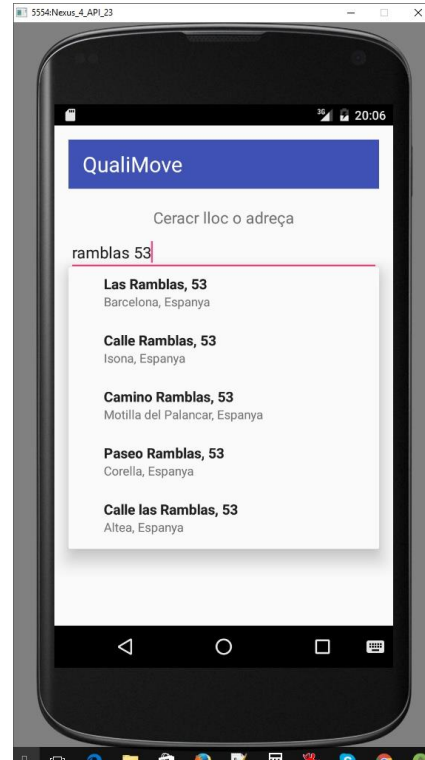


Il·lustració 10: marcadors inicials

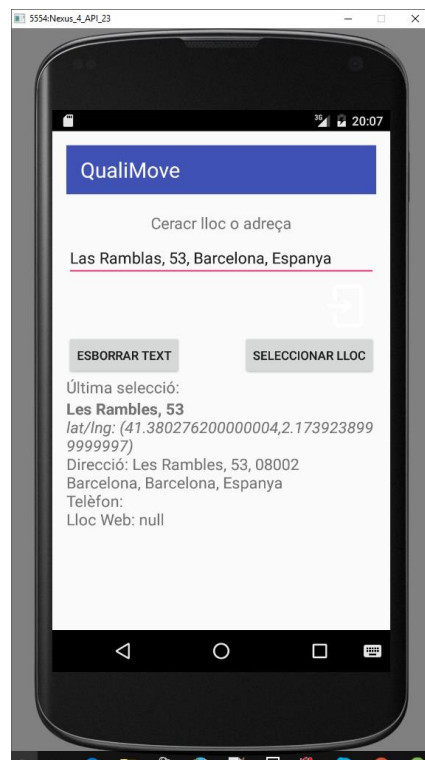
Seleccionant la cerca per text. És pot veure les diferents pantalles de selecció, la cerca és dinàmica, a mida que s'escriu presenta una llista de possibles llocs.



II-lustració 11: Cercar per text 1

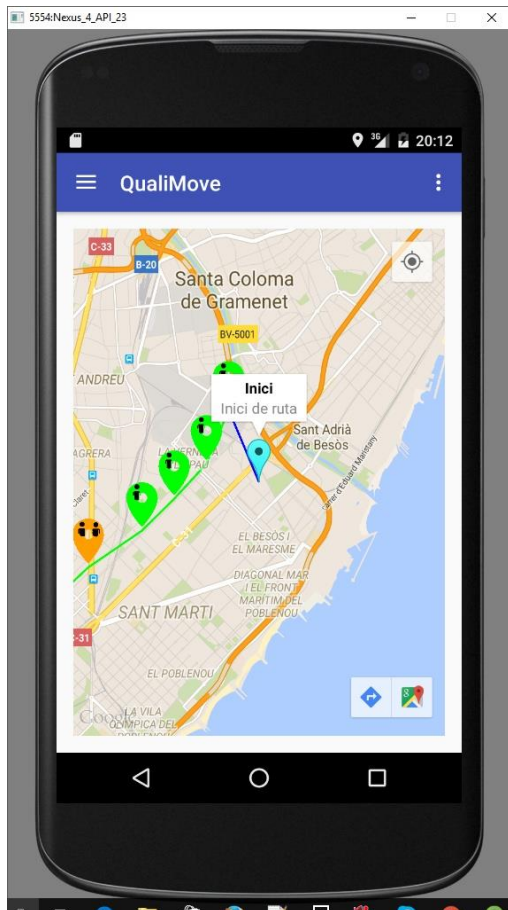


II-lustració 12: Cercar per text 2

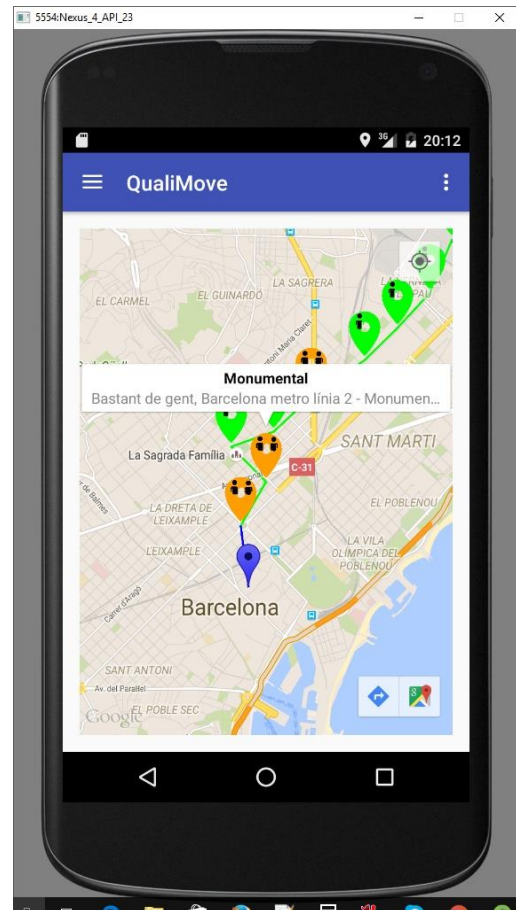


II-lustració 13: Cercar per text 3

Una vegada definits els punts d'origen i final de la ruta, l'opció "calcular ruta" presentarà la ruta en el mapa.

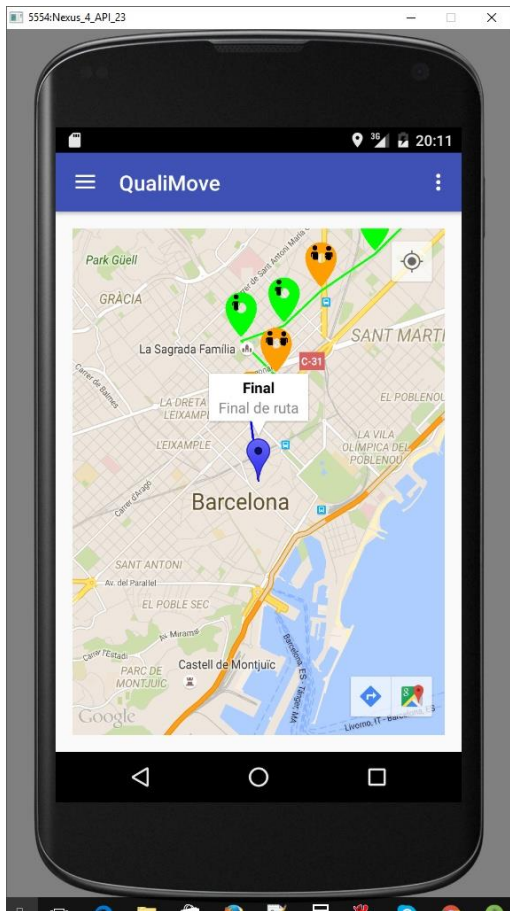


II-Il·lustració 14: Ruta 1



II-Il·lustració 15: Ruta 2

Com es pot veure, els marcadors donen informació visual amb pictogrames de persones en el marcadors (1,2,3) segons el nivell de saturació del node presentat, i també en text si es selecciona el marcadors a la pantalla, es mostra un quadre de text amb el nom del node (estació o parada) i una descripció més completa a sota, amb el nivell de saturació i el nom de la línia del transport.



Il·lustració 16: Ruta 3

El sistema permet moure per el mapa i anant mirant els marcadors per comprovar la ruta en si.

Cada marcador mostra el nom de l'estació i la línia de transport amb la direcció, estació inicial - estació final, a la segona línia del marcador.

4. Càlcul de saturació

El sistema incorporat per realitzar el càlcul intenta ser el menys rígid possible ja que s'entén que el procés pot variar molt d'un node a un altre o poden sorgir mètodes més adients per segons quins nodes i evolucionar o parametritzar aquests segons canvis o condicions de les càmeres. Un canvi de càmera, una remodelació d'una estació poden comportar un procés amb una parametrització oposada a la que hi havia fins el moment.

Per aquest motiu el node conté dos paràmetres que són cadenes de text en format JSON, es a dir, poden definir una estructura adient completa dintre del paràmetre.

Disposen del paràmetre “photoProcess” per realitzar un preprocessat, registrar la imatge, carregar patró, retallar, pintar els punts negres del patró sobre la imatge, convertir a grisos, convertir amb mètode Solbe i per últim tipus de procés a realitzar, ràpid o per partícules.

Per el procés com tal, tenim el paràmetre “parameterPhotoProcess” on les dades són per el tipus de procés que realitzarà el servei.

Com tenim les dades de parametrització amb JSON no cal modificar tota l’estructura xml per afegir nous preprocessats o per afegir nous mètodes de procés d’imatge, això també és important de cara a fer servir una base de dades relacional en el cas de moltes definicions de rutes d’una ciutat gran.

4.1 Preprocessat

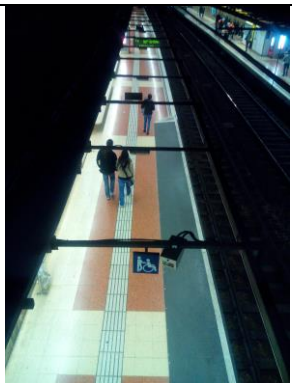


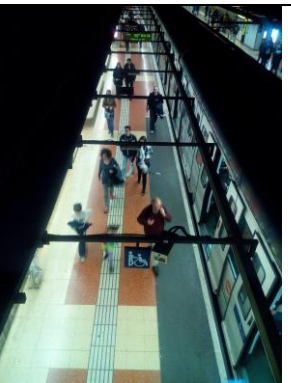
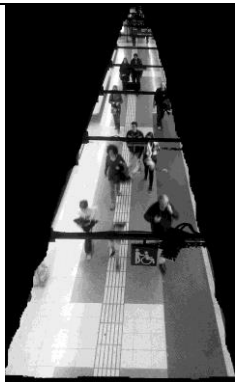
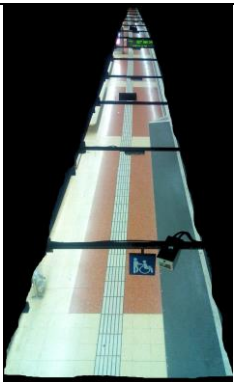
Dintre del preprocessat està inclòs la carrega de patró d’imatge, donat que segons el procés pot ser no és necessari, i el registre a disc de la imatge, aquest registre es determina pels paràmetres en el node i la ruta dels camis i tipus d’extensió que es tracta, així com el nom inicial al que s’afegirà el temps en format ansi “aaaammddhmmss”

4.2 Procés “fast”

Aquest procés dóna bons resultats amb imatges “ideals” imatges que no tenen moviment de càmera i en el que la llum sempre és la mateixa, la majoria de les estacions de metro compleixen aquestes normes i pot ser un bon sistema de càlcul. Per les proves realitzades, aquest mètode no ha donat bons resultats, les imatges són preses des de un mòbil i per tant hi ha moviment de la càmera.

El sistema es basa amb comparar la imatge actual amb un patró i donant un valor segons la diferència de colors dels píxels que hi ha, aquesta diferència es calcula amb una tolerància de tons donada com paràmetre.

Per les proves realitzades s’ha fet un preprocessat amb retallada d’imatge, modificar punts amb negre segons patró i canvi a nivell de grisos, hi ha una imatge que mostra aquest tractament.

			
Saturació: 0.7435248	Saturació: 0.393202	Saturació: 0.676432	Saturació: 0.242631
			
Exemple de resultat final d'imatge tractada.	Patró utilitzat per la prova.		

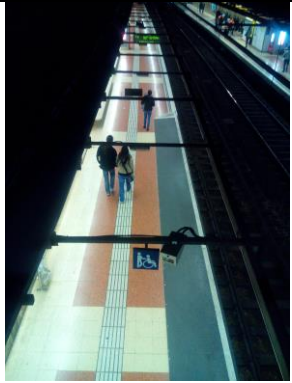



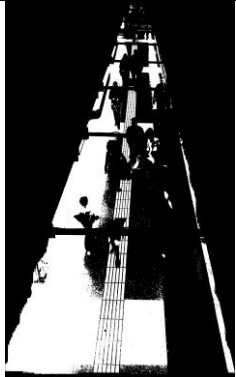
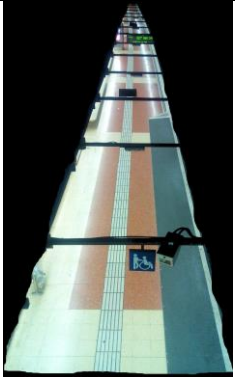
Taula 3: Proves tipus procés "fast"

Com es pot veure pel resultats, no són molt coherents amb la imatge, això es degut al moviment de la càmera al realitzar les fotos provocant un desplaçament de tots els punts a tractar i per tant de moltes diferències que desvirtuen el valor de saturació.

4.3 Procés "particles"

Aquest procés es realitza amb la llibreria ImageJ, agafant la funció d'analitzar partícules es pot arribar a un nivell bastant acceptable per comptar les persones. El mètode cerca i compta partícules a la imatge des de unes mides de cercles donades. Per aquesta prova s'han fet

parametritzacions fins trobar la més adient. S'utilitza el preprocessat per retallar i pintar el punts de negre segons patró per minimitzar els erros i optimitzar el procés a realitzar, el mateix procés de ImageJ realitza més modificacions a la imatge.

			
Saturació: 0.108	Saturació: 0.232	Saturació: 0.114	Saturació: 0.15
			
Exemple de resultat final d'imatge tractada	Patró utilitzat per la prova.		

Taula 4: Proves tipus procés "particles"

Es pot comprovar que aquest mètode encara no és fiable del tot, és molt més coherent encara que hi hagi moviment de càmera però no té en compte la geometria de la imatge i la seva perspectiva i per tant les imatges de persones properes a la càmera poden desvirtuar el valor de saturació.

5. Càlcul de ruta

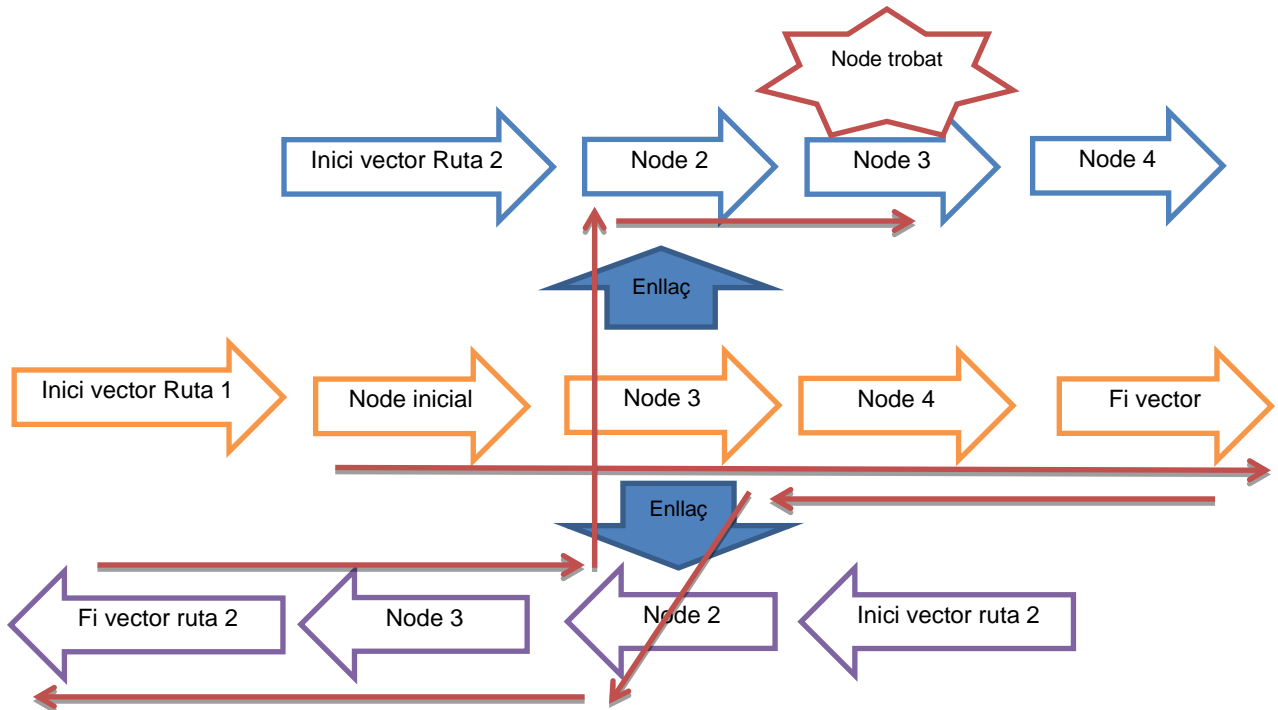
La realització de cerca i càlcul de la ruta és tot a memòria mitjançant objectes NodeQM, primer el servei carrega els objectes de NodeQM amb una llista dintre d'un objecte RouteQM que finalment el objecte ManagerQM els fica a una llista, d'aquesta manera tenim una llista de rutes en la que cada ruta conté una llista de nodes. La definició d'objecte únic per node és la suma del nom de ruta més el nom de node.

El sistema de cerca i càlcul és molt bàsic, primer cerca els nodes de totes les rutes més propers a les coordenades d'inici i final, si troba nodes dintre del perímetre d'1,7 km, llavors fica tots els nodes trobats a una llista, hi ha una per inici i un altre pel final.

El pas següent cerca una ruta entre node inicial i node final, aquesta cerca es realitza per combinació de tots els nodes trobats.

La cerca de ruta tracta els nodes com un vector amb direcció única, cada vector té inici i final i cada node pot tindre enllaços a altres vectors (rutes), per exemple, la línia 2 de metro són dues rutes, una en cada direcció i cada ruta conté enllaços en els nodes corresponents amb la línia 1 i la línia V17 de bus.

Al mètode se li dona un node inicial i un node final, cerca pel vector del node inicial fins a trobar el node final, comprovant que no ha passat per el mateix node per evitar cerques circulars, si el troba retorna un valor de veritat i una llista de nodes (passada com paràmetre) i en cas contrari un valor fals. Aquest mètode és recursiu i primer analitza el vector i si no troba res al final del mateix, va comprovant els enllaços mentre va "retrocedint", si troba un enllaç realitza el mateix en el nou vector, fins que troba el node final o retrocedeix al node inicial sense res.



Aquest procés solament agafa les dues rutes amb menys distancia entre les coordenades inicial i final i els nodes i amb menys distancia total comptant tots els nodes, encara que posteriorment solament s'envia una ruta al client.

6. Conclusions

Amb aquest treball es posa de manifest que encara queden molts projectes per innovar, amb idees senzilles que s'han de desenvolupar, cercant les noves eines que surten, cada vegada més potents i especialitzades. Això no vol dir que sigui fàcil el procés però sí que dóna la certesa de que hi ha un mètode o forma de trobar una solució per donar el servei de qualitat que un esperava.

6.1 Assoliment d'objectius

La idea inicial es basava a donar una ruta alternativa sense saturació, però per qüestions de temps no s'ha pogut assolir, la feina que tinc actualment ha fet reduir considerablement el temps per dedicar al projecte i de totes maneres no sé si hagués tingut prou temps disponible per donar el producte que volia. Encara i així, el principal motiu, donar una informació addicional d'utilitat al trajecte pel usuari, s'ha assolit.

L'altra qüestió que per temps no ha quedat molt clara és el tractament de la saturació, hi ha eines i mètodes per realitzar-ho correctament però un altre vegada, el temps ha tallat la qualitat esperada.

6.2 Desviació de planificació i metodologia

A l'inici del projecte, per raons de la feina actual que porto, no s'ha dedicat el temps necessari per portar el projecte, a la part de la pac 2 va ser inclòs dramàtic, sense poder realitzar metodologia d'aquesta. Això a comportat que el producte final a pareixin mancances de cara a l'interface de l'usuari.

6.3 Aplicacions futures

Queden pendents molts punts, la cerca de ruta hauria de trobar més rutes amb els mateix node d'inici i final i quedar-se amb la més curta, actualment pot donar-te un enllaç i no ser necessari.

Els mètodes de tractament d'imatges s'haurien de afinar més, tractar geometria en el de partícules, moure imatge per enquadrar en el ràpid.

Encara que visualment mostra la ruta, hauria d'haver una opció per mostrar l'itinerari en format llista i destacar els enllaços si hi ha.

Queda pendent també trobar una ruta alternativa a la donada en cas de que estigui saturada.

Encara que s'utilitza Google Maps a la part d'App, podria utilitzar-se la part de cercar trajecte, pot donar moltes possibilitats a l'aplicació.

A la part del tractament d'imatges, la plataforma ImageJ té molts plugins que poden oferir un tractament de més qualitat per tractar les imatges i donar un valor molt acurat.

En aquest projecte solament s'ha tractat la part d'informació immediata per l'usuari, però si es tracten les dades històriques de saturació, pot oferir:

- Optimització necessària de les línies de transport per les empreses
- Prediccions per l'usuari, pot demanar com ho trobarà el dia x.
- Previsió a nivell de gestió de ciutat per els grans esdeveniments, un partit de futbol molt esperat, un concert multitudinari, una manifestació, etc.

4. Glossari

APK: Aplicació Android.

ImageJ: Plataforma de tractament científic d'imatge.

JSON: Format per intercanvi de dades.

Plugin: Tros de programa que es pot afegir com component.

Servlet: Programa en Java per realitzar serveis WEB.

Vector: Definició abstracte d'un element amb direcció.

WAR: Fitxer per desplegar un servei WEB amb Servlets.

5. Bibliografia

[1] Apps de transport públic.

- 27/09/2015 <http://www.transportnsw.info/en/travelling-with-us/keep-updated/apps.page>
- 28/09/2015 <http://www.telegraph.co.uk/travel/travel-advice/9280598/The-best-apps-for-public-transport.html>

- 28/09/2015 <http://www.imore.com/best-us-transit-apps-iphone-google-maps-moovit-citymapper-more>
- 28/09/2015 <http://www.appolicious.com/articles/9581-top-39-local-public-transportation-iphone-apps-recommended-by-city-residents-visitors>

[2] Imatges gratuïtes

free images <http://es.freeimages.com>

6. Annexos

Fitxers i desplegament d'App

Junt amb aquesta memòria en el fitxer zip corresponent, es trobarà:

Fitxer eclipse.zip amb el projecte Eclipse de servei WEB

Fitxer android.zip amb el projecte Android Studio d'App.

Fitxer app-debug.apk amb l'App per instal·lar en un dispositiu Android.

Fitxer QualiMove.war de desplegament de servei WEB.

I els documents "QualiMove Guia d'us" i "Importació projectes Eclipse i Android"

Per desplegar el fitxer war en un servidor d'aplicacions Java, desar el fitxer en el directori root o deploy del servidor (Tomcat o semblant)

Per executar el servei WEB des de l'entorn Eclipse, comprovar que hi ha instal·lat i configurat un servidor Tomcat a Eclipse, importar el projecte, comprovar la JRE i llibreries externes i executar com servidor.

Per instal·lar l'App a un dispositiu Android, accedir al fitxer des de el dispositiu i instal·lar, s'ha de tindre configurat la instal·lació d'aplicacions no segures.

Per executar des de Android Studio amb emulador, importar projecte i comprovar que hi ha les ultimes versions de google play service, executar amb emulador.

Estructura fitxer xml de ruta

```

<route>
  <properties>
    <name>Nom de ruta</name>
    <displayName>Nom de ruta per mostrar</displayName>
    <pathPatern>Camí per fotos de patrons dels
nodes</pathPatern>
    <pathState>Camí per les fotos preses dels
nodes</pathState>
  </properties>
  <node>
    <name>Nom del node</name>
    <displayName> Nom del node per mostrar </displayName>
    <coordinates latitude="0.0" longitude="0.0" />
    <limitSaturation value="0.0" />
    <namePhotoPatern>Nom de foto amb
patró</namePhotoPatern>
    <preNamePhotoState>part inicial per identificar la foto de
node

    </preNamePhotoState>
    <urlCameraIP>url per recollir foto camera</urlCameraIP>
    <userCameraIP>usuari accés camera</userCameraIP>
    <pwdCameraIP>Contrasenya accés
camera</pwdCameraIP>
    <photoProcess>Paràmetres de preprocessat
d'imatge</photoProcess>
    <parameterPhotoProcess>Paràmetres procés càlcul de
saturació </parameterPhotoProcess>
  </node>
  <node> ... </node>

</route>

```

Estructura fitxer xml d'enllaç entre nodes

```

<relations>
  <nodeRelation route="name" node="name"
relRoute="name" relNode="name" />
  ...

```

```
<nodeRelation route="name" node="name"
  relRoute="name" relNode="name" />
</relations>
```

Paràmetres preprocessat d'imatges (<photoProcess>)

```
save_original_image: boolean
load_pattern: boolean
pattern_to_gray: boolean
image_crop: boolean crop_x: int
crop_y: int
crop_w: int
crop_h: int
apply_black_pattern_to_image: boolean
image_to_gray: boolean
apply_sobel_to_image: boolean
type_process: String ["fast" | "particles"]
```

Paràmetres procés càlcul saturació (<parameterPhotoProcess>)

Tipus procés "fast"

```
tolerance: int
multiplier: double
```

Tipus procés "particles"

```
minSize: double
maxSize: double
minCirc: double
maxCirc: double
subtract_particles: double
maximun_number_particles: double
```

Exemple de fitxer de rutes xml

```
<xml...>
<properties>
  <name>BCNLinia2</name>
```

```
<displayName>Metro L2 Paral·lel - Badalona Pompeu Fabra</displayName>
<pathPattern>paterns</pathPattern>
<pathState>historic</pathState>
<photoExtension>jpg</photoExtension>
</properties>

<node>
  <name>parallel</name>
  <displayName>Paral·lel</displayName>
  <coordinates latitude="41.3749047" longitude="2.1652453" />
  <limitSaturation value="0.5" />
  <namePhotoPattern>l2parallel</namePhotoPattern>
  <preNamePhotoState>l2parallel</preNamePhotoState>
  <urlCameraIP><![CDATA[http://localhost:8080/image?size=640x480]]></urlCameraIP>
  <userCameraIP>user</userCameraIP>
  <pwdCameraIP>pwd</pwdCameraIP>

  <photoProcess>{"save_original_image":true,"load_pattern":true,"pattern_to_gray":true,"i
mage_crop":true,"crop_x":32,"crop_y":48,"crop_w":2487,"crop_h":4112,"apply_black_pattern_to
_image":true,"image_to_gray":true,"apply_sobel_to_image":false,"type_process":"particles"}</p
hotoProcess>

  <parameterPhotoProcess>{"minSize":2.0,"maxSize":7.0,"minCirc":0.0,"maxCirc":0.64,"s
ubtract_particles":10,"maximun_number_particles":500}</parameterPhotoProcess>

</node>

<node>
...
```