

Sistema de càlcul d'hàbitats amb dispositius Android

Ferran Obiols i Colomina

Resum— En l'actualitat les empreses de telecomunicacions centren les estratègies comercials en les dades que els usuaris consumeixen connectar-se a Internet, en comptes dels antics mercats de les trucades o els SMS. Aquest moviment d'interessos fa que els preus de contractació d'aquests serveis augmentin de forma desproporcionada a la qualitat que es contracta, en relació a les diferents generacions de protocols e infraestructura que utilitza GSM. Les xarxes DTN poden ser una alternativa que permet utilitzar una infraestructura de baix cost comparat amb GSM, utilitzant els mateixos dispositius per rebre i transmetre dades. Aquestes comunicacions poden tenir una certa intel·ligència en l'encaminament de les dades utilitzant diferents criteris. En aquest projecte es vol crear un sistema que permetrà calcular l'hàbitat d'un dispositiu mòbil i mostrar aquesta zona a l'usuari. Utilitzant aquesta informació es podrà prendre decisions per un enrutament de les dades més eficient.

Paraules clau— Delay Tolerant Network (DTN), Hàbitat, Localització, Android

Abstract— Nowadays telecommunication companies focus his business strategies in the data that users consume when they connect to Internet, instead of the common markets like phone calls or SMS. This movement of interest make the price for contracting these services increase disproportionately to the quality that is contracted, in relation to the different generations of GSM protocols and the infrastructure that uses. Delay Tolerant Networks can be an alternative that allows using a low cost infrastructure compared with GSM, using smartphones as infrastructure to send or receive data. These communications can have a certain intelligence in routing data using different protocols. This project want to create a system to calculate the habitat of a mobile device and show this area to the user. Using this information a node can make more efficient routing decisions.

Index Terms— Delay Tolerant Network (DTN), Habitat, Location, Android

1 INTRODUCCIÓ

ACTUALMENT el mercat de smartphones no para de créixer i s'han consolidat com la tecnologia més utilitzada per sobre dels ordinadors portàtils i les tablets. Es calcula que si continua creixent aquest ritme fins al 2020, arribaran a la xifra de 3,4 bilions de dispositius en tot el món [1]. Les companyies telefòniques ja han vist que l'actual mercat es troba en garantir l'accés a Internet per a tots els dispositius per sobre de les trucades o els SMS, i per tant estan fent un canvi en les estratègies per poder treure més beneficis a partir del consum de dades.

Actualment Espanya ja forma part dels països amb les taxes més elevades per dades i amb un servei que no es pot comparar amb altres de preu similar [2]. El problema és que en un futur el preu de les connexions per GSM (Global System Mobile) augmentaran, però no ho farà de la mateixa forma el servei que ofereixen les empreses de telecomunicacions.

Aquest projecte vol buscar una possible solució a

aquest problema i oferir una alternativa a les actuals formes de comunicació pels dispositius mòbils. Una possible sol·lució pot ser el fet d'utilitzar xarxes ad hoc utilitzant els mateixos dispositius com a infraestructura inalàmbrica, però els actuals sistemes operatius de smartphones tenen limitades aquestes característiques i no garanteixen un correcte funcionament. Una segona sol·lució passa per utilitzar xarxes DTN (Delay Tolerant Network) en una infraestructura que cada dispositiu fos partícip de l'encaminament de les dades.

Aquest encaminament podria tenir una certa intel·ligència afegint la probabilitat de trobar diferents dispositius i ser capaç de triar el millor d'aquests per reenviar les dades depenent de la càrrega d'aquests o del que trigara en arribar a un altre node.

En aquest punt és on entra la funcionalitat del treball, l'objectiu és desenvolupar un sistema que permeti conèixer el moviment dels nodes a partir del desplaçament dels usuaris. Coneixent aquesta informació dels possibles nodes amb qui es pot comunicar, es podria prendre la decisió d'enrutament de la informació. Aquest sistema ha de calcular l'hàbitat del dispositiu mitjançant una aplicació i poder veure el seu comportament a partir dels desplaçaments de l'usuari en una plataforma web.

A la resta de l'article es presentarà: l'estat de l'art en

-
- E-mail de contacte: ferran.obiols@e-campus.uab.com
 - Menció realitzada: *Tecnologies de la Informació*.
 - Treball tutoritzat per: Ramon Martí Escalé (dEIC)
 - Curs 2014/15

què es s'explicaran els conceptes que es tractaran al llarg del treball, els diferents objectius del treball, la metodologia que s'ha fet servir, les tecnologies utilitzades en el projecte i l'estructura del sistema com a resultats de l'anàlisi, el disseny i la implementació de cada mòdul del projecte, els resultats obtinguts i les conclusions.

2 ESTAT DE L'ART

Aquest apartat està destinat a introduir els diferents conceptes que es veuran al llarg de l'article i a explicar d'on s'ha tret el concepte d'hàbitat.

L'ús dels hàbitats per la presa de decisions en l'enrutament de dades ja s'ha utilitzat anteriorment, de fet aquest treball s'ha originat a partir de la tesi doctoral de l'Adrià Sánchez Carmona, més concretament el paper "PrivHab: a Multiagent Secure Georouting Protocol for PodCast Distribution on Disconnected Areas" [3]. Aquest article tracta la problemàtica de com garantir l'accés a continguts en zones que no garanteixen d'infraestructura per les comunicacions. La solució que es proposa és utilitzar xarxes DTN amb el protocol que aporta intel·ligència en l'enrutament de dades a partir dels desplaçaments dels nodes de la xarxa.

2.1 Xarxes DTN

Les DTN o Delay Tolerant Network són un tipus de xarxes que es diferencien de totes les altres perquè no és necessari una connexió punt a punt per a rebre o transmetre informació, permeten les pèrdues de connexió o els retards sense generar problemes a la comunicació. És important destacar que cada element de la xarxa té la funcionalitat de node normal que permet enviar o rebre, i d'encaminador [4].

El secret del funcionament de les DTN és que cada node és capaç d'emmagatzemar els missatges que ha d'enviar, tant els que ha generat com els rebuts per altres nodes, i un cop es troba amb un altre dispositiu pot prendre les decisions per enviar els missatges o no depenent del protocol que implementin.

Aquest protocol poden ser més o menys complexos aportant una certa intel·ligència en l'encaminament de dades. Els protocols més comuns es basen en enviar el missatge i esperar que aquest arribi o en què la font envia diverses còpies a diferents nodes per augmentar la probabilitat que arribi. El protocol proposat per l'Adrià en el seu treball va més enllà i utilitza el comportament dels nodes de la xarxa per seleccionar quin és el millor camí perquè arribin les dades.

En les xarxes DTN normals, els nodes acostumen a tenir una posició fixe i s'utilitza un node especial que és l'encarregat de desplaçar-se i recollir les dades emmagatzemades en els altres nodes de la xarxa. En aquest cas es farà servir un tipus especial de xarxes DTN en què els nodes també tenen moviment, anomenades MDTN o Mobile Delay Tolerant Network. Les xarxes MDTN permetran millor adaptar-se a l'entorn que es proposa en aquest treball, utilitzant cada dispositiu com a node per enviar i rebre afegint el moviment continuu.

2.2 Els hàbitats

Com ja s'ha comentat, el protocol creat per l'Adrià utilitza el comportament dels nodes per prendre decisions en l'encaminament dels missatges. Per modelar aquest desplaçament es fa servir el concepte d'hàbitat, que es tracta de l'àrea on és més probable trobar el node.

En la tesi aquesta àrea es representa per una esfera, però es podria utilitzar diferents figures geomètriques en dues dimensions que es puguin adaptar als moviments dels usuaris. Es justifica l'ús d'un cercle perquè els dispositius que s'utilitzen tenen certes limitacions i és més ràpid calcular les dades necessàries per a una figura simple com un cercle, en aquest treball s'utilitzarà la mateixa figura per problemes trobats per mostrar en un mapa figures complexes com poden ser el lípse.

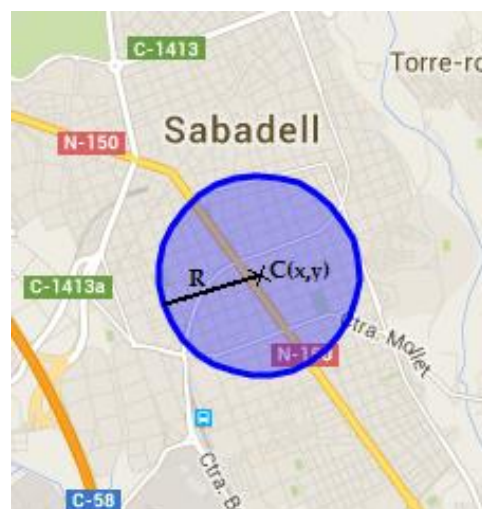


Figura 1. Exemple d'hàbitat del sistema.

Per definir els hàbitats com a cercles, com es pot veure en la figura 1, es fan servir dos elements diferents: el centre de l'hàbitat $C=(lat,lng)$, i el radi R . Aquesta serà la informació que necessitarà per poder definir l'hàbitat i prendre les decisions per fer l'encaminament.

3 OBJECTIUS

Per una correcta evolució del treball es divideix el sistema en dues parts diferenciades en què cada una tindrà diferents objectius.

3.1 Aplicació

L'objectiu principal del treball consisteix en desenvolupar una aplicació pel sistema operatiu Android que s'encarregi de:

- Localitzar el dispositiu: l'aplicació ha d'obtenir les coordenades actuals del dispositiu en què s'està executant, via GPS o Xarxa, en una freqüència determinada per l'usuari.
- Càlcul d'hàbitats: l'aplicació ha de fer els càlculs necessaris per mostrar la figura geomètrica que representa l'hàbitat cada cop que s'obtingui una nova localització del dispositiu.

- Enviar dades: l'aplicació ha de poder enviar les dades que s'han calculat al servidor central.
- Mostrar l'hàbitat: l'aplicació ha de poder mostrar l'hàbitat actual de l'usuari en un mapa juntament amb la localització de l'últim mostreig.

3.2 Servidor

L'objectiu secundari del treball tracta de crear un servidor central que permetrà les següents funcionalitats:

- Rebre dades: el servidor ha de poder rebre les dades enviades des dels dispositius actius.
- Emmagatzemar: el servidor ha de poder emmagatzemar les dades rebudes a partir dels dispositius.
- Mostrar l'hàbitat: el servidor ha de poder mostrar per a cada usuari del sistema el seu hàbitat actual tal com fa l'aplicació.
- Creuar dades: el servidor ha de poder generar coneixement a partir de l'encreuament de dades de diferents usuaris en un panell per administradors.

El primer objectiu és de vital importància, ja que només amb aquest es pot fer el càlcul dels hàbitats i veure com es comporta la figura geomètrica a partir del desplaçament del dispositiu. El segon objectiu ens permet visualitzar tota la informació emmagatzemada i poder veure com es comporten els diferents usuaris.

4 METODOLOGIA

La metodologia que s'ha utilitzat per desenvolupar el sistema ha estat una metodologia de tipus iteratiu incremental. Aquesta és molt utilitzada pel desenvolupament de software, primer de tot es fa una anàlisi de tot el sistema i posteriorment dissenyant, implementant i provant cada part del sistema de forma individual.

4.1 Anàlisi

Aquesta fase és l'única que només es realitza un cop, si no es torna a requerir, ja que no forma part del cicle iteratiu com les altres fases. En aquesta etapa es fa un primer disseny d'alt nivell per identificar els diferents mòduls del sistema que posteriorment s'hauran d'implementar de forma individual. També ajuda al desenvolupador entendre com funcionarà tot el projecte i en les possibles solucions de cada mòdul abans d'arribar a la fase de disseny. En els apartats 5 i 6 s'explicaran els resultats que s'han arribat al fer l'anàlisi general del sistema.

4.2 Disseny

En aquesta fase es pensa en la solució concreta per a cada mòdul. Es busquen quines tecnologies es faran servir pel desenvolupament i com s'han d'utilitzar, en el cas que no es conegui la tecnologia es realitzarà un període d'investigació per aprendre a utilitzar-la de forma bàsica. Al final d'aquesta etapa cal realitzar diagrames UML de Classes i de Seqüència per plasmar la solució i facilitar la programació.

4.3 Implementació

En la fase d'implementació es programen aquelles solucions generades en l'etapa de disseny utilitzant els diagrames de classes per crear els objectes que es faran servir i els diagrames de seqüència per implementar les relacions entre els objectes. Com a resultat d'aquesta fase cal tenir el codi que implementi la funcionalitat del mòdul.

4.4 Test

En aquesta última fase es fan les proves dels objectes que s'han creat en l'etapa d'implementació per verificar que els mètodes realitzin les funcionalitats adequades i no tinguin comportaments estranys. També s'aprofita per veure si tot el sistema implementat fins al moment funciona adequadament o el nou mòdul ha generat nous problemes. Un cop assegurat el funcionament es passa a la fase de disseny del següent mòdul.

Al final del desenvolupament del servidor i de l'aplicació es faran testos generals de tot el sistema per veure que el funcionament és fluid i no existeixen errors en el projecte.

5 TECNOLOGIES UTILITZADES

La primera part de l'anàlisi es va fer un estudi de quines tecnologies seria millor utilitzar en cada part del sistema. En aquest apartat s'explicarà els resultats d'aquest primer anàlisi dividint segons quines tecnologies s'han fet servir tant per l'aplicació Android, com pel servidor central.

5.1 Tecnologies de l'aplicació

El sistema operatiu triat per desenvolupar l'aplicació està relacionat amb el llenguatge de programació que cal utilitzar i amb les facilitats que permet tant la comunitat com els propietaris d'aquest. El llenguatge principal de programació per Android és Java, encara que es poden utilitzar d'altres com C#, .NET, C, ... Aquest és un llenguatge molt utilitzat a la carrera i per tant facilita el temps d'aprenentatge. També cal dir que la comunitat de desenvolupadors per Android sumat a les facilitats que Google dona, és molt més fàcil crear una aplicació sense tenir coneixements previs.

Concretament es fa servir el SDK de Java jdk1.7.0_51 i l'aplicació es compila utilitzant la API 22 per la versió 5.1 d'Android, però permet executar-se fins a la versió 2.3. Aquests paquets són els que venien instal·lats amb l'última versió de l'entorn de programació que s'ha fet servir, Android Studio.

5.2 Tecnologies del servidor

El servidor es troba en una màquina virtual EC2 d'Amazon Web Service que utilitza el sistema operatiu Amazon Linux AMI 2015.03, s'ha fet servir aquest servidor perquè es buscava una alternativa de baix cost que no es necessites gran potència per la pàgina. Es fa servir Apache com a servidor web per la familiaritat que es té al haver-lo utilitzat a la carrera, aquest és fàcil d'utilitzar i de configurar a nivell bàsic. Com a base de dades s'utilitza MySQL[5] juntament amb PhpMyAdmin per administrar-la de forma remota, aquesta ja ha sigut utilitzada en diferents

assignatures de la carrera i facilitarà la creació de la base de dades del sistema. Finalment com a llenguatge de programació per un entorn LAMP (Linux Apache MySQL PHP), s'ha fet servir PHP5 [6]. Per desenvolupar les pàgines web es fa servir HTML5, CSS3 per l'estil de la pàgina i JavaScript per les animacions en els mapes.

Utilitzant tecnologies Open Source permet que no tingui cap cost el desenvolupament del servidor, permetent un rendiment adequat per aquest projecte.

6 ESTRUCTURA DEL SISTEMA

En la segona part de l'anàlisi es va fer un estudi complet del sistema per poder dividir aquest en diferents parts i fer més fàcil el desenvolupament del conjunt. En aquesta secció s'explicarà aquesta divisió en diferents mòduls de les dues diferents parts del sistema, el servidor i l'aplicació Android.

La figura 2 és la representació final del sistema dividint entre l'aplicació en la part superior i el servidor a la part inferior. Cada una de les parts es diferencia entre el front end, els mòduls visibles pels usuaris, i el back end.

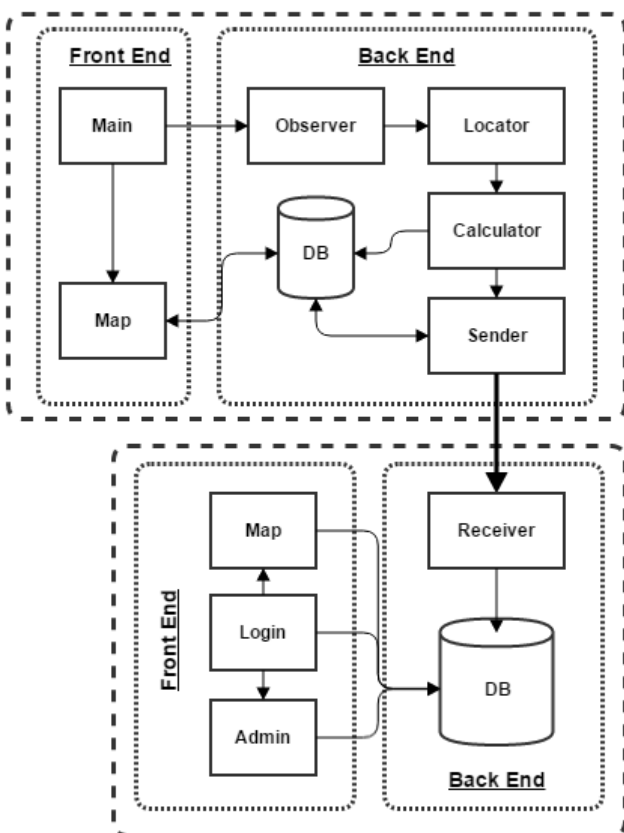


Figura 2. Estructura del sistema.

6.1 Estructura de l'aplicació

En aquest apartat s'explicaran els diferents mòduls que formen part de l'aplicació Android. Aquests mòduls estan dividits segons si el client pot interactuar amb ells, i per tant són activitats d'Android, o en les funcionalitats de background que seran diferents tipus de services.

6.1.1 Front end

En el front end de l'aplicació podem veure els dos mòduls amb què l'usuari pot interactuar. El Main és la pantalla que inicia tot el funcionament del programa i el que permet configurar la freqüència de mostreig. El Map és la pantalla que permet a l'usuari veure l'hàbitat i la posició actual en un mapa.

6.1.2 Back end

En el back end de l'aplicació existeixen més mòduls que s'encarreguen del funcionament de l'aplicació. L'Observer és el responsable de gestionar la freqüència de mostreig per encendre la resta de mòduls. El Locator és l'encarregat d'obtenir les coordenades del dispositiu depenent del desplaçament de l'usuari. El Calculator s'encarrega d'obtenir les dades necessàries per a representar un hàbitat i guardar-les a la base de dades. Per últim el Sender envia les dades emmagatzemades del Calculator i la localització al servidor.

Aquesta estructura jeràrquica permet a cada mòdul activar-se només quan és necessari, per exemple no s'executarà el Calculator si no s'han obtingut les coordenades del dispositiu. Aquest disseny permet reduir el consum de bateria de l'aplicació i reduir el nombre de connexions que es realitzen contra el servidor.

6.2 Estructura del servidor

En aquest apartat s'explicarà els diferents apartats que formen part del servidor. Aquest està format per la part que l'usuari pot interactuar directament, anomenat front end o vista de l'aplicació, i les funcions que l'usuari no pot veure, anomenat back end o model del servei web.

6.2.1 Front end

El front end del servidor és la part més important, ja que permet visualitzar els hàbitats pels usuaris. Està format per tres pantalles diferents: el login de la web que s'encarrega d'autenticar als usuaris, la pàgina de map permet veure segons l'identificador de cada usuari el seu hàbitat actual juntament amb la posició i per últim el panell d'administrador permet fer encreuament de dades entre els diferents hàbitats.

6.2.2 Back end

El back end del servidor només es dedica a guardar les dades rebudes i emmagatzemar-les en la base de dades pel futur ús de les diferents vistes.

7 DISSENY I IMPLEMENTACIÓ

Com que aquest treball té la finalitat de desenvolupar el sistema per calcular els hàbitats i mostrar-los als usuaris, en aquest apartat s'explicarà la funcionalitat dels diferents mòduls del sistema i aquells que siguin més interessants es farà una breu explicació del funcionament.

7.1 Mòduls de l'aplicació

Aquests són els mòduls resultants de la fase de l'anàlisi principal, en cada mòdul s'ha pensat en un disseny i després s'ha implementat, en comptes de fer un disseny grupal. Això ha permès dividir la feina i anar implementant mòduls a mesura que cada funcionés correctament.

7.1.1 Main

Aquest primer mòdul és l'entrada de l'aplicació i la primera vista, anomenada MainActivity. Des de aquesta pantalla l'usuari pot iniciar o parar el procés d'obtenir dades, canviar de activity per veure l'hàbitat actual en el mapa i modificar la freqüència de mostreig de dades. Aquesta classe només implementa els controls necessaris per navegar per l'aplicació.

7.1.2 Observer

Aquest mòdul és l'encarregat d'engegar els serveis en background cada cop que es compleixi la freqüència de mostreig. En aquest apartat han aparegut forces problemes, ja que depenent del tipus d'acció que realitzes el servei es podria utilitzar una o altres llibreries.

La primera idea a implementar era a partir de la MainActivity suspendre l'aplicació un temps determinat, però si aquest temps era massa gran, el sistema operatiu passava l'aplicació a un segon pla i es deixava d'executar.

La segona idea era utilitzar Threads per mantenir l'aplicació principal oberta, però poder deixar en un segon pla els serveis quan no s'havien de fer servir. Aquest mètode solucionava el problema anterior, però un cop es bloquejés el dispositiu l'aplicació deixaria d'executar-se.

La tercera opció implicava utilitzar una API anomenada AlarmManager [7] que permet configurar alarmes del sistema operatiu i deixar que aquest s'encarregui d'engegar els serveis. Aquesta opció permetia un baix consum de bateria i l'execució encara que el dispositiu estigues bloquejat, però només es podien utilitzar IntentService per a les notificacions i aquest tipus de serveis no permetien l'execució de serveis tan llargs com els de l'aplicació.

La quarta i última opció era seguir utilitzant AlarmManager, però en comptes de què el sistema operatiu engegues els serveis, les notificacions arriben a una classe BroadcastReceiver i aquest s'encarrega d'iniciar els serveis.

7.1.3 Locator

El mòdul Locator és l'encarregat d'obtenir les coordenades actuals del dispositiu. Aquestes dades es demanen directament al sistema operatiu i per tant cal demanar permisos a l'usuari per recollir-les [8]. L'aplicació prioritza les coordenades rebudes a partir de la xarxa, ja que són més ràpides, es consumeix menys bateria i es poden recollir dins d'edificis. En el cas que no es puguin recollir les coordenades de xarxa, es passa a demanar per GPS.

7.1.4 Calculator

Aquest és un dels mòduls més interessants, ja que és l'en-

carregat de calcular les dades necessàries per representar els hàbitats. Un cop es reben dades del mòdul de localització, es realitzen els següents passos:

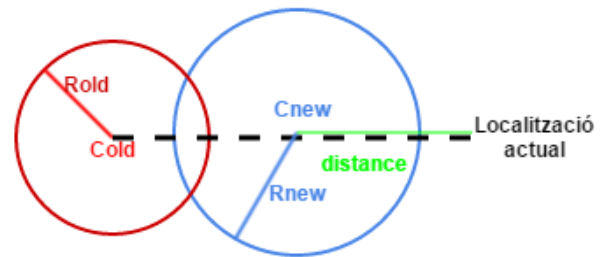


Figura 3. Representació del càlcul d'un nou hàbitat.

- 1 Si són les primeres coordenades, s'inicialitza l'hàbitat actual en què radi és 0 i el centre són les coordenades actuals.

$$H = (C(actLat, actLng), 0)$$

- 2 En el cas que ja existeixi un hàbitat el primer que es fa és buscar el nou centre, que serà un punt de la recta entre el centre anterior i la posició actual depenent del percentatge de modificació.

$$\begin{aligned} newLat &= actLat * \alpha + oldLat * (1 - \alpha) \\ newLng &= actLng * \alpha + oldLng * (1 - \alpha) \\ newC &= (newLat, newLng) \end{aligned}$$

- 3 El segon pas si existeix un hàbitat, és calcular el nou radi, que es farà de forma similar al nou centre. En aquest cas es farà servir el radi anterior, el percentatge de modificació i la distància entre el nou centre i la posició actual (es pot veure com el radi s'adapta al desplaçament de l'usuari).

$$newR = distance * \alpha + oldR * (1 - \alpha)$$

A la figura 3 es pot veure gràficament les diferents mesures del càlcul del nou hàbitat. Segons els interessos del sistema o depenent de l'entorn dels dispositius es pot modificar el percentatge de modificació i ajudarà a l'hàbitat a adaptar-se més als canvis o ser més estàtic i prioritzar la forma anterior. Per exemple en un entorn on els dispositius es mouen de forma constant pot interessar utilitzar un alt percentatge i permetrà que l'hàbitat s'adapti més a la posició actual.

7.1.5 Base de dades

Aquest mòdul s'utilitza per guardar les dades que cal enviar al servidor i també les necessàries per a fer els càlculs dels nous hàbitats. Com que l'aplicació només ha de mostrar l'hàbitat actual es va considerar que no calia guardar l'històric d'hàbitats d'aquest usuari, i aquesta funcionalitat implementar-la en el servidor on la base de dades si guarda tots els hàbitats.

Es pot dir que l'aplicació té una base de dades només quan s'està executant, utilitzant una classe que exten de Application [9] que emmagatzema les dades i ser utilitza-

des per tota l'aplicació. Per tant cada cop que s'engega l'aplicació és com si tornes a carregar per primer cop i cal tornar a inicialitzar l'hàbitat.

7.1.6 Map

El mòdul Map és l'encarregat de mostrar l'hàbitat actual per l'aplicació. Per poder utilitzar Google Maps en Android [10] cal crear una activity especial que extengui de FragmentActivity i importar les llibreries de Maps (com.google.android.gms.maps.*). Per poder utilitzar aquestes llibreries, cal demanar una clau e incloure-la en el projecte. Per dibuixar en el mapa es fa servir la mateixa llibreria de Google Maps, només cal crear l'objecte que es desitgi, en aquest cas un objecte de tipus Circle e inserir les propietats d'aquest objecte.

Un exemple de com es veu l'aplicació és la figura 4, podem veure l'esfera dibuixada a partir de les últimes dades calculades i la localització actual del dispositiu dibuixada utilitzant un marcador.

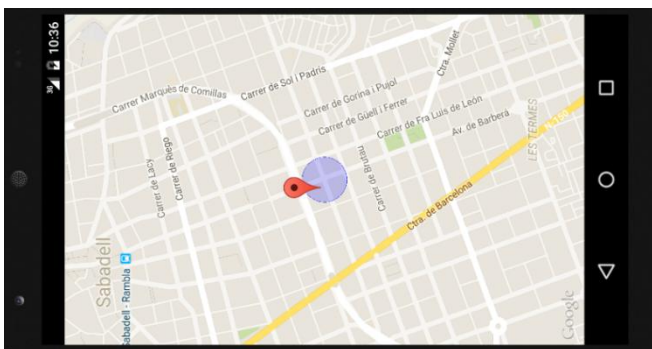


Figura 4. ViewActivity exemple del simulador.

7.1.7 Sender

Per últim, el sender és l'encarregat d'enviar les dades recollides del locator i les dades calculades de l'hàbitat actual al servidor. Per enviar-les es primer de tot cal guardar-les en BasicNameValuePair, que relaciona el nom de la dada amb el seu valor. Un cop guardades cal crear un objecte HttpPost per enviar la petició al servidor. En el cas que no funcionés la petició POST, està preparat un segon mètode molt similar per enviar peticions GET. Les dades que s'envien són: imei, les coordenades del centre de l'hàbitat, el radi i les coordenades actuals.

7.2 Mòduls del servidor

Com s'ha mencionat anteriorment, per a cada mòdul s'ha fet el disseny i la implementació tot seguit. Els mòduls més importants del servidor són les vistes amb les que els usuaris poden interactuar.

7.2.1 Login

Aquest primer mòdul del servidor és l'encarregat d'autenticar els usuaris per accedir al mapa on es pot veure l'hàbitat actual o al panell d'administració. Per a no obligar als usuaris a registrar-se per utilitzar l'aplicació web i poder identificar de forma única a cada dispositiu, s'utilitza el IMEI (International Mobile System Equipment Identity) que és un identificador únic per a cada dispositiu.

7.2.2 Map

En aquest mòdul es pot veure el mateix mapa que es mostra per l'aplicació. Aquesta pantalla es pot utilitzar únicament per fer el seguiment en temps real del dispositiu per altres persones que no puguin accedir a l'aplicació Android. Com ja s'ha comentat anteriorment les pàgines estan creades sense utilitzar una comunicació entre el client i el servidor per rebre informació, el que fa el mòdul per proporcionar la informació necessària és utilitzar camps amagats que tindran la informació guardada al moment de crear-se la pàgina.

Un cop en client, utilitzant JavaScript es recull aquesta informació i es passa a la API de Google Maps per dibuixar els hàbitats.

7.2.3 Receiver

El receiver és el punt de connexió entre el servidor i l'aplicació Android. Aquest és una pantalla en blanc que recull les dades enviades per POST o per GET. Un cop es connecten si l'usuari no està registrat, primer s'introdueix un nou usuari i després les dades. Les dades rebudes són totes aquelles

7.2.4 Base de dades

La base de dades que s'utilitza en el servidor és molt senzilla, només es fan servir dues taules:

- La taula d'usuaris que guarda l'identificador i un camp boolea per saber si aquell usuari és administrador.
- La taula que com es pot veure en la figura 5 conté tots aquells camps necessaris per dibuixar els hàbitats. Els camps centerLat, centerLng i radius són els que realment es fan servir, però per aportar més funcionalitats al servidor s'han afegit altres.

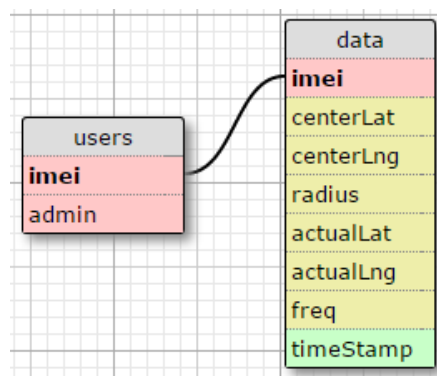


Figura 5. Diagrama de la base de dades.

7.2.5 Admin

El mòdul d'administració és el més interessant del servidor, aquest fins al moment permet dues funcionalitats diferents que creuen informació de diferents hàbitats:

- Veure l'últim hàbitat filtrant per l'identificador del dispositiu: aquesta funcionalitat permet veure l'últim hàbitat que s'ha registrat a la base de dades de tots els dispositius, o filtrar per IMEI. Aquesta funci-

onalitat ens permet relacionar l'estat actual de tots els hàbitats i quines possibles zones en comú tenen. Com es pot veure en la figura 6 es poden veure els quatre últims hàbitats dels dispositius que han fet servir el sistema.

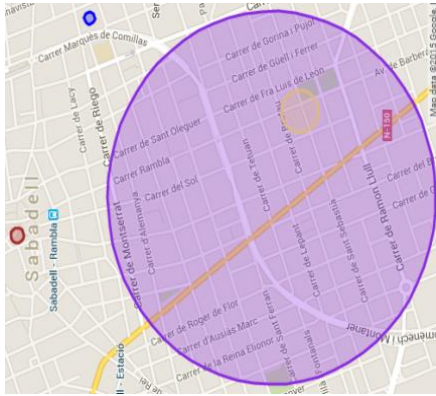


Figura 6. Captura de mapa amb diferents hàbitats.

- Veure l'historial d'un dispositiu en concret: aquesta funcionalitat és molt important i ens permet veure quins són tots els hàbitats d'un determinat dispositiu. Amb aquesta opció es pot veure com el radi s'adapta al moviment dels dispositius, com augmenta quan l'usuari s'allunya del centre i com disminueix quan l'usuari es mou per dintre de l'hàbitat com es pot veure en la figura 7.

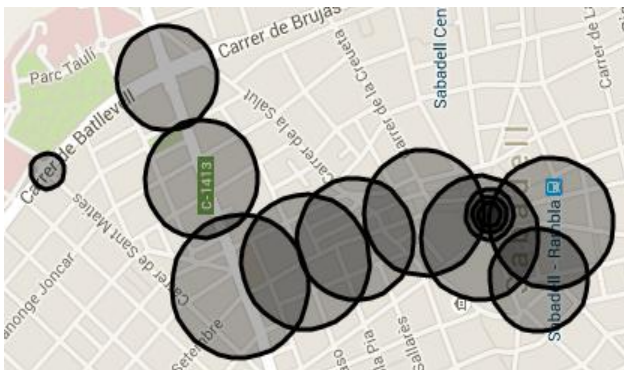


Figura 7. Captura de mapa amb l'historial.

8 TESTS

En aquest apartat es mostraran els resultats de diferents tests per veure si el comportament dels hàbitats que mostra el sistema és correcte. Es realitzaran dues proves per verificar si l'hàbitat reacciona amb el desplaçament del dispositiu, i també es faran proves amb diferents percentatges de modificació per veure si aquest s'adapta de forma correcta.

8.1 Desplaçament

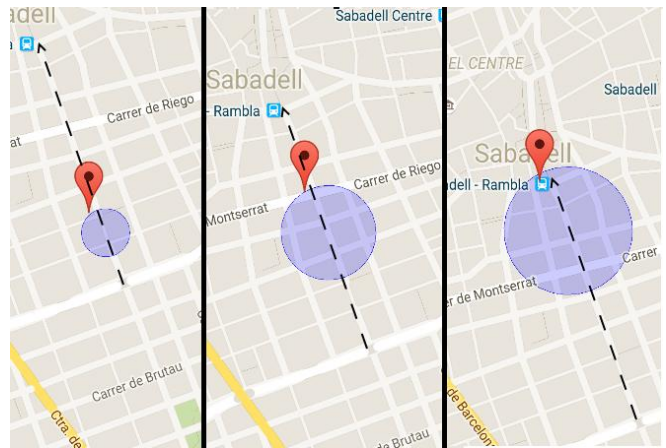
La primera prova es tracta de veure en diferents casos com reacciona l'hàbitat amb desplaçaments completament diferents amb el mateix percentatge de modificació, i s'avaluarà si la reacció de l'hàbitat és coherent amb la

teoria explicada en l'estat de l'art. Per a poder tractar les màximes situacions es plantejaran 3 escenaris diferents.

8.1.1 Augment del radi

El primer escenari traca amb un moviment rectilini uniforme amb un percentatge de modificació del 50%. El resultat esperat és un increment del radi de l'hàbitat proporcional amb la velocitat en què es mou l'usuari, en una velocitat constatat l'hàbitat hauria d'augmentar de forma constant també.

Figura 8. Desplaçament per augmentar l'hàbitat.



Com es pot veure en la figura 8 el recorregut que segeix el dispositiu és el marcat amb la línia discontinua. Es pot apreciar que la posició actual del dispositiu en cada captura va desplaçant-se a una velocitat constant i d'una forma similar augmenta el radi de l'hàbitat.

8.1.2 Disminució del radi

En el segon escenari es vol comprovar el concepte contrari a la prova anterior, es tracta de fer moviments dins de l'hàbitat per veure com ha de disminuir el radi. El resultat esperat és veure com el radi disminueix si el dispositiu es mou dins de l'hàbitat i veure que si es deixa de moure's completament, tendirà a 0 més ràpid.

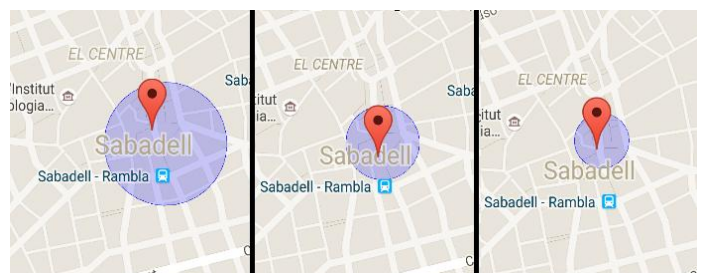


Figura 9. Desplaçament per disminuir l'hàbitat.

Com es pot veure a la figura 9, els resultats esperats s'han complert i quan l'usuari es desplaça per dintre de l'hàbitat o es queda quiet, disminueix per ajustar-se a la zona en què s'està movent.

8.1.3 Adaptació del radi

L'últim escenari es vol comprovar què fa l'hàbitat si el dispositiu es mou entre els mateixos dos punts de forma contínua. Els resultats esperats és que l'hàbitat s'adapti aquesta zona i el cercle inclogui aquests dos punts a prop del seu perímetre.

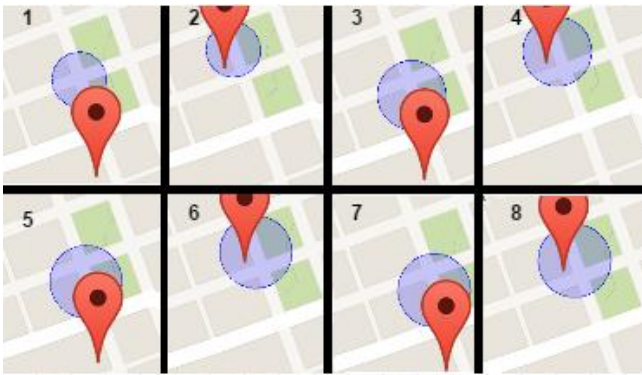


Figura 10. Desplaçament per adaptar l'hàbitat.

Com podem veure en les iteracions 1 i 2 que l'hàbitat un radi similar, en la iteració 3 l'hàbitat es fa més gran i es comença a centrar. En les altres iteracions es mantenen com la tercera augmentant de forma poc significativa el radi.

8.2 Modificació del percentatge

La segona prova es tracta de fer el mateix recorregut que en la figura 8, però modificant el percentatge de modificació per veure com afecta per sota o per sobre del 50%.

8.2.1 Molta adaptació al dispositiu

En el primer escenari es farà servir una probabilitat de modificació major de 50%, concretament 75%. Els resultats esperats és que l'hàbitat s'adapti més a les noves posicions i estigui més aprop del marcador de la posició actual.

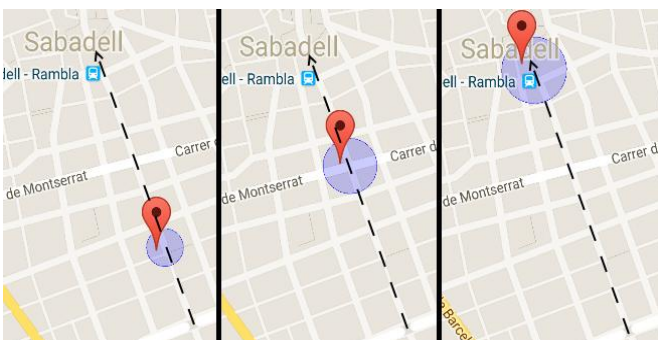


Figura 11. Desplaçament amb probabilitat del 75%.

Com podem veure a la figura 11, la localització actual del dispositiu sempre es troba dins del nou hàbitat, per tant es demostra que els resultats concorden amb la teoria.

8.2.2 Poca adaptació al dispositiu

En el segon escenari es farà la prova contrària a l'anterior fent servir una probabilitat de modificació del 25%. El resultat esperat és que la localització actual s'allunyi del centre de l'hàbitat i que aquest augmenti el radi, però que es desplaci poc.



Figura 12. Desplaçament amb probabilitat del 25%

Com es pot veure en la figura 12, el punt on es troba el dispositiu cada cop s'allunya més del centre de l'hàbitat i aquest només augmenta el radi amb un desplaçament molt petit. Comparant les figures 8, 11 i 12, es pot veure com reacciona l'hàbitat segons el percentatge d'adaptació amb que es calcula. A la figura 8 amb un 50% es pot veure que la posició actual del dispositiu es troba sempre al límit de l'hàbitat, a la figura 11 amb un 75% la posició actual es troba dins de l'hàbitat i amb un 25% sempre es troba a fora.

9 RESULTATS

Com s'ha pogut veure en l'apartat anterior, s'ha complert els objectius que es proposaven al principi. S'ha aconseguit crear un sistema que és capaç de fer els càlculs i les representacions dels hàbitats a partir de les dades extretes en una aplicació Android, i fer creuaments d'informació en un servei web en la següent ruta ec2-52-26-14-167.us-west-2.compute.amazonaws.com.

L'evolució del projecte ha estat complicada per endarreriments de tasques i la modificació de les diferents planificacions de seguiment. Les tasques que han sigut més problemàtiques són les relacionades amb el primer objectiu, ja que aquesta era la primera aplicació per Android que feia i per tant al principi calia buscar molta documentació per començar a fer el projecte. Les tasques de l'objectiu 1 que han fet endarrerir més el projecte són:

- Aprenentatge del funcionament del sistema operatiu: abans de començar a desenvolupar l'aplicació s'ha seguit diferents guies per aprendre com crear aplicacions bàsiques [11].
- Implementació del mòdul que controla la freqüència: com s'ha comentat a l'apartat de disseny i implementació, per a fer que cada cert temps s'agafés les coordenades a través del sistema operatiu s'han hagut d'intentar diverses opcions que poguessin executar-se de forma continuada i sense consumir recursos en el moment que no es volgués localitzar el dispositiu.

- Localització del dispositiu: per a fer la localització la primera idea va ser utilitzar un servei web que al connectar-se retornava les coordenades de l'usuari, però al cap d'unes certes peticions les coordenades rebudes eren incorrectes. Després d'una reunió de seguiment el tutor va proposar obtenir les dades directament del dispositiu utilitzant l'API de Android.
- Dibuixar l'hàbitat: en un principi es volia utilitzar una el·lipse per representar l'hàbitat, però més enllà dels càlculs l'única forma de poder mostrar una el·lipse per pantalla era calcular diversos punts del perímetre i traçar línies per crear un polígon similar.
- Enviar les dades al servidor: en un principi es volia enviar les dades utilitzant el format JSON per una més fàcil recollida e inserció a la base de dades, però les llibreries que té Android per utilitzar JSON són obsoletes i portava problemes en enviar. Per aquest problema es va passar a utilitzar peticions POST contra un servidor en PHP.

Les tasques del segon objectiu no han portat cap problema i s'han realitzat en el temps especificat en la planificació, només modificant la data de finalització pels endarreriments de la resta del projecte. L'únic inconvenient va ser que en un principi enviant les dades per GET o per POST no es rebien en el servidor, però fent proves va acabar funcionant sense modificar el codi.

10 CONCLUSIONS

Durant l'evolució del projecte s'han implementat les següents funcionalitats:

- Localització del dispositiu: l'aplicació és capaç d'obtenir les coordenades actuals del dispositiu a partir de GPS o xarxa.
- Càlculs dels elements necessaris per un cercle com a hàbitat: l'aplicació és capaç de calcular el centre del cercle i el radi a partir de les coordenades actuals del dispositiu i de l'informació de l'hàbitat anterior.
- Visualització de l'hàbitat actual a partir de l'aplicació Android: l'aplicació es capaç de mostrar l'hàbitat actual i la posició actual del dispositiu.
- Concentrar les dades en un servidor central: tots els dispositius que utilitzen l'aplicació envien les dades a un servidor central que les guarda.
- Visualitzar l'historial d'un hàbitat: l'aplicació web és capaç de mostrar tots els hàbitats guardats d'un dispositiu en el mateix mapa.
- Visualitzar l'estat actual de diferents hàbitats: l'aplicació web és capaç de mostrar l'estat actual de tots els dispositius guardats en la base de dades. També es capaç de filtrar i només mostrar aquells s'específicui.

S'han complert els dos objectius inicials satisfactoriament: creant una aplicació Android que permet obtenir les coordenades del dispositiu, calcular l'hàbitat i enviar-lo al servidor; i un servidor que reb les dades i permet diferents visualitzacions per els usuaris.

10.1 Línies futures

En un futur aquest projecte és pot ampliar en el que es refereix a l'extracció de coneixement a partir de les dades rebudes dels dispositius. Actualment només existeixen dues funcionalitats, però en un futur pot ser interessant afegir diversos apartats com podrien ser:

- Cercar hàbitats que estiguin en una certa zona, per poder estudiar aquells usuaris que tenen hàbitats similars.
- Estudiar els diferents hàbitats a partir d'informació de l'usuari. Es podria estudiar quins són els hàbitats entre diferents edats o professions, per poder extreure més coneixement i a partir d'aquesta informació de cada usuari poder fer millors encaminaments.
- Cercar possibles rutes d'informació utilitzant un històric dels moviments dels usuaris, podent especificar l'inici i el final de la comunicació i per quins dispositius passaria aquelles dades.
- Millorar les visualitzacions per ser més interactives, com per exemple poder mostrar en animacions el desplaçament per hores de cada usuari.
- Cerca per hores i per dies, per veure la localització dels dispositius en una franja horària en concret o per saber a quines hores es disposa de més dispositius.
- Fer prediccions de futurs hàbitats a partir de l'experiència apresada fins al moment.
- Poder utilitzar diferents formes d'hàbitats per veure quina s'ajusta més al comportament de les persones.

Com es pot veure les línies futures es centren en millorar les funcionalitats de l'aplicació web, per extreure més coneixement. L'aplicació no té molt més que millorar, ja que aquesta només serveix per poder extreure les dades.

11 AGRAÏMENTS

Vull donar les gràcies al meu tutor Ramon Martí Escalé per l'ajuda que m'ha donat al llarg del treball i a l'Adrià Sánchez Carmona per l'ajuda amb la seva tesi doctoral. També volia agrair a la meua família, parella i amics pel suport durant la realització del projecte.

12 BIBLIOGRAFIA

- [1] International Data Corporation . Smartphone OS Market Share. [Online] Disponible: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
- [2] Almudena Martín. España, cuarto país de Europa con la telefonía móvil más cara. [Online] Disponible: <https://www.kelisto.es/telefonía-móvil/actualidad/espana-cuarto-pais-de-europa-con-telefonía-móvil-mas-cara-2439>
- [3] Adrià Sánchez Carmona. PrivHab: a Multiagent Secure Georouting Protocol for Podcast Distribution on Disconnected Areas.
- [4] RFC 4838, Delay-Tolerant Networking Architecture. [Online] Disponible: <https://tools.ietf.org/html/rfc4838>.
- [5] LNH. Basic MySQL Configuration. [Online] Disponible: http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch34_:_Basic_MySQL_Configuration#.VYwIW_

mppBc.

- [6] The PHP Group: Php: Manual de PHP. [Online] Disponible: <http://php.net/manual/es/>
- [7] Android Developers. AlarmManager. [Online] Disponible: <http://developer.android.com/reference/android/app/AlarmManager.html>
- [8] Android Developers. 'Location and Sensors APIs' [Online] Disponible: <http://developer.android.com/guide/topics/sensors/index.html>
- [9] Android Developers. Application. [Online] Disponible: <http://developer.android.com/reference/android/app/Application.html>
- [10] Android Developers. Google Maps Android API. [Online] Disponible: <https://developers.google.com/maps/documentation/android/?hl=es>.
- [11] Android Developers. Getting Started. [Online] Disponible: <https://developer.android.com/training/index.html>