

Universidade do Minho
Escola de Engenharia

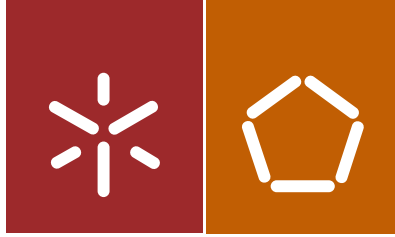
Diogo André da Silva Matos

where@UM - Aplicação móvel de
posicionamento
where@UM - Mobile application
for positioning

Diogo André da Silva Matos
where@UM - Aplicação móvel de posicionamento
where@UM - Mobile application for positioning

UMinho | 2014

Outubro de 2014



Universidade do Minho
Escola de Engenharia

Diogo André da Silva Matos

where@UM - Aplicação móvel de
posicionamento
where@UM - Mobile application
for positioning

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Filipe Meneses
Professor Doutor Adriano Moreira

Outubro de 2014

DECLARAÇÃO

Nome

Diogo André da Silva Matos

Endereço electrónico: a58666@alunos.uminho.pt Telefone: 914701721

Número do Bilhete de Identidade: 13502492

Título dissertação /tese

where@UM – Aplicação móvel de posicionamento

where@UM – Mobile application for positioning

Orientador(es):

Professor Doutor Filipe Meneses

Professor Doutor Adriano Moreira

Ano de conclusão: 2014

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia de Comunicações

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respectiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;
2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;
3. DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA TESE/TRABALHO

Universidade do Minho, ___/___/_____

Assinatura: _____

Agradecimentos

Ao terminar mais uma etapa, a última e a mais exigente do meu percurso académico, é altura de agradecer a todas as pessoas que, de uma forma ou de outra, presenciaram e contribuíram para o meu crescimento pessoal e profissional ao longo destes anos.

Primeiramente gostaria de agradecer a orientação e o apoio dos meus orientadores, Professor Doutor Filipe Meneses e Professor Doutor Adriano Moreira. Agradeço-lhes a disponibilidade, a dedicação e a ajuda prestada durante a realização deste projeto. Sem a sua contribuição este momento não seria possível, muito obrigado.

Gostaria de agradecer aos meus colegas e amigos que me acompanharam ao longo do curso, nos bons e nos maus momentos. Aos meus amigos de longa data, apesar de distantes, não foram esquecidos. A todos, o meu sincero obrigado.

À Sofia, obrigado pelo amor, amizade e principalmente paciência demonstrada ao longo destes anos.

Ao meu pai Emanuel, à minha mãe Maria e ao meu irmão Vitor, obrigado pela educação e valores que me transmitiram. Obrigado por me tornarem na pessoa que sou hoje.

Resumo

O contexto em que uma aplicação é executada e/ou onde o respetivo utilizador se encontra é, cada vez mais, considerado fundamental para algumas áreas aplicacionais. A localização é uma das dimensões mais exploradas, existindo atualmente variadas tecnologias para determinar a posição dos dispositivos móveis pessoais, o que frequentemente corresponde à posição do seu utilizador, em ambientes interiores, onde o popular sistema de localização GPS não funciona.

Nos últimos anos tem-se assistido ao surgimento de várias tecnologias que exploram as infraestruturas habitualmente instaladas em edifícios, oferecendo novas funcionalidades. Neste contexto destacam-se os sistemas de posicionamento baseados na técnica Wi-Fi *fingerprinting*. Estes sistemas têm vindo a ser alvo da atenção de muitos investigadores e empresas porque proporcionam uma solução de baixo custo para o posicionamento de pessoas e objetos em tempo real. No entanto, a construção dos mapas de rádio que suportam estas soluções continuam a constituir um desafio, principalmente para edifícios de grande escala.

O propósito desta dissertação centra-se no desenvolvimento de um sistema de posicionamento baseado em Wi-Fi *fingerprinting* que, através de uma solução colaborativa, minimize ou suprima o problema da construção dos mapas de rádio para edifícios de grandes dimensões.

A solução apresentada e descrita nesta dissertação baseia-se no desenvolvimento de uma aplicação para dispositivos móveis que combina os conceitos associados às redes sociais com a utilização de um motor de posicionamento já existente. O principal objetivo da solução passa por envolver os utilizadores do sistema, inconscientemente, na construção do mapa de rádio. A aplicação desenvolvida destina-se essencialmente à população da Universidade do Minho, mas não está limitada exclusivamente aos campi da mesma.

Abstract

The context which the application is executed and/or where the user finds itself is increasingly considered essential for some application areas. The location is one of the most explored dimensions, currently exists several technologies that estimate the position of personal mobile devices in indoor environments, which often corresponds to the user's location, where the popular location system GPS does not work.

In recent year many technologies that exploit the usually network deployed in buildings have emerged, offering new features. In this context positioning systems based on Wi-Fi fingerprinting have become very popular in recent years. These systems have been attracting the attention of many researchers and companies because they provide a low cost solution for real time positioning of both people and objects. However, the construction of the radio maps that support these solutions remain a challenge in indoor environments.

This thesis focuses on the development of a positioning system based on Wi-Fi fingerprinting using a collaborative solution that reduces or eliminates the radio map construction problem for large-scale buildings.

The solution presented and described in this thesis is based on developing an application for mobile devices which combines the concepts of social network with the use of an existing positioning engine. The main goal of this solution is to engage the system's users on radio map construction. The developed application is mainly intended to the population of University of Minho, but not limited to the campuses limits.

Índice de conteúdos

Agradecimentos	iii
Resumo	v
Abstract	vii
Índice de conteúdos	ix
Lista de figuras	xiii
Lista de tabelas	xvii
Lista de abreviaturas.....	19
1. Introdução	21
1.1 Enquadramento e motivação	21
1.2 Objetivos	23
1.3 Abordagem.....	23
1.4 Estrutura da dissertação	24
2. Sistemas de posicionamento e aplicações	25
2.1 Técnicas de posicionamento para ambientes interiores	26
2.2 Sistemas de posicionamento	27
2.2.1 Herecast	27
2.2.2 Molé.....	29
2.3 Produtos comerciais	34
2.4 Aplicações móveis	35
2.4.1 Foursquare	35
2.4.2 Find My Friends!	36
2.4.3 Crux Indoor Location	37

2.4.4	Vodafone Radar	38
2.4.5	Find My Friends	39
2.4.6	Facebook Phone Tracker	39
2.4.7	Friend Mapper	40
2.4.8	Friend Pilot	40
2.4.9	App2Find	41
2.4.10	People Tracker	41
2.4.11	Resumo	42
3.	O problema da calibração	43
3.1	Descrição do problema	43
3.2	Soluções para calibração automática	44
4.	Desenho do sistema	45
4.1	Funcionamento geral do sistema.....	46
4.2	Requisitos.....	46
4.2.1	Requisitos do sistema central	47
4.2.2	Requisitos da aplicação móvel	47
4.3	Processo de construção do mapa de rádio.....	49
4.4	Arquitetura da solução	49
4.4.1	Aplicação móvel	50
4.4.2	Servidor	52
4.4.3	Protocolos de comunicação	53
4.4.3.1	Interface A: módulo de suporte à aplicação	55
4.4.3.2	Interface B: módulo de construção do mapa de rádio	55
4.4.4	Modelo de dados.....	56
4.5	Estratégia de disseminação da aplicação	59

5.	Implementação do sistema	61
5.1	Aplicação móvel	61
5.1.1	Plataforma de desenvolvimento.....	61
5.1.2	Componentes da aplicação móvel where@UM	62
5.1.3	Recolha de dados do ambiente rádio Wi-Fi.....	64
5.1.4	Ecrãs e funcionalidades	68
5.1.4.1	Registo	68
5.1.4.2	Login.....	70
5.1.4.3	Ecrã principal.....	72
5.1.4.4	Ecrã com informação sobre os amigos	77
5.1.4.5	Ecrã de definições.....	78
5.2	Servidor.....	78
5.2.1	Base de dados	79
5.2.2	Integração com o Foursquare	83
5.2.3	Obtenção da localização dos utilizadores	86
5.2.4	Evolução do módulo de construção do mapa de rádio	89
5.2.5	Comparação de fingerprints.....	93
5.2.5.1	Distância entre fingerprints.....	94
5.2.5.2	Nível de similaridade entre fingerprints	94
5.2.5.3	Definição dos parâmetros defRSSI e threshold	95
5.2.5.4	Processo de comparação de fingerprints.....	95
5.3	Website	97
5.4	Testes	98
5.4.1	Semelhança entre fingerprints	98
5.4.2	Análise do processo de comparação de fingerprints	102

5.4.3	Consumo de energia da aplicação móvel	106
6.	Avaliação.....	107
6.1	Experiência no mundo real	107
6.1.1	Adesão e motivação dos utilizadores.....	108
6.1.2	Experiência dos utilizadores	108
6.2	Análise dos resultados.....	109
7.	Conclusões	111
7.1	Trabalho futuro	112
7.1.1	Integração	112
7.1.2	Aplicação móvel	113
	Referências	115
	Anexo A	123
	Anexo B	127
	Anexo C	133
	Anexo D	135

Lista de figuras

Figura 2.1 - Formulário de registo de um AP, retirado de [25]	28
Figura 2.2 - Interface do utilizador da aplicação Molé, retirado de [26]	30
Figura 2.3 - Exemplo do MAO, adaptado de [26]	31
Figura 2.4 - Interação entre componentes do Molé, retirado de [26]	32
Figura 2.5 - Comparação de leituras realizadas com e sem o detetor de movimento. A linha a tracejado representa o período em que o utilizador se encontra em movimento. O símbolo + mostra quando o localizador recebe uma nova <i>fingerprint</i> e produz uma nova estimativa. Retirado de [26]	33
Figura 2.6 - Ecrã de perfil e histórico da aplicação Foursquare	36
Figura 2.7 - Ecrã principal da aplicação “Find My Friends!”	37
Figura 2.8 - Ecrãs da aplicação Crux Indoor Location, editado de [40]	38
Figura 2.9 - Ecrã inicial da aplicação Vodafone Radar, retirado de [41]	39
Figura 2.10 - Ecrã da aplicação Friend Mapper, retirado de [45]	40
Figura 2.11 - Exemplo de realidade aumentada da aplicação App2Find, retirado de [47]	41
Figura 2.12 - Ecrã da aplicação Friend Mapper, retirado de [49]	42
Figura 4.1 - Desenho global do sistema de posicionamento para ambientes interiores baseado em Wi-Fi <i>fingerprinting</i>	45
Figura 4.2 – Arquitetura geral do sistema.....	50
Figura 5.1 - Desenho dos módulos e respetivos componentes	62
Figura 5.2 - Entidades da base de dados do módulo de recolha de dados	65
Figura 5.3 - Recolha de dados dos pontos de acesso Wi-Fi.....	66
Figura 5.4 - Exemplo de uma fingerprint em formato JSON	67

Figura 5.5 - Ecrã de registo.....	68
Figura 5.6 - Registo de um utilizador	69
Figura 5.7 - Arquitetura geral do serviço Google Cloud Messaging	70
Figura 5.8 - Ecrã de login	70
Figura 5.9 - Login de um utilizador	71
Figura 5.10 - Ecrã principal	72
Figura 5.11 - Informação detalhada sobre a localização de um utilizador	73
Figura 5.12 - Localização incorreta ou desconhecida. Ecrã que permite consultar a informação sobre o local em que o utilizador se encontra e ainda corrigir a sua localização.....	74
Figura 5.13 - Formulários apresentados para corrigir a localização	75
Figura 5.14 - Obtenção das coordenadas GPS.....	76
Figura 5.15 - Ecrã com informação sobre o estado das relações do utilizador.....	77
Figura 5.16 - Notificação de um pedido de amizade	77
Figura 5.17 - Ecrã de definições	78
Figura 5.18 - Tecnologias do servidor	79
Figura 5.19 - Entidades com informações relativas aos utilizadores.....	80
Figura 5.20 - Entidades com informações relativas ao sistema de posicionamento	81
Figura 5.21 - Exemplo de locais, dentro e fora da Universidade.....	82
Figura 5.22 - Dados extraídos de um local proveniente do Foursquare	84
Figura 5.23 - Integração com o Foursquare	85
Figura 5.24 - Obtenção da própria localização	86
Figura 5.25 - Localização do utilizador desconhecida.....	87
Figura 5.26 - Obtenção da localização dos amigos.....	88
Figura 5.27 - Processo de localização dos utilizadores (versão 2)	90
Figura 5.28 - Processo de localização dos utilizadores adotado (versão 3).....	92

Figura 5.29 - Processo de comparação de fingerprints	96
Figura 5.30 - Website where@UM, retirado de [62]	97
Figura 5.31 – Exemplo de 4 das salas e locais escolhidos para a recolha de fingerprints	99
Figura 5.32 - Gráficos estatísticos dos valores de falsos positivos e falsos negativos em função do <i>threshold</i> : defRSSI = -110 (esquerda) e defRSSI = -90 (direita).....	99
Figura 5.33 - Impacto do defRSSI no valor ótimo do <i>threshold</i>	100
Figura 5.34 - Desempenho geral da experiência, retirado de [63].....	101
Figura 5.35 - Gráfico comparativo entre as duas versões do processo de comparação	105
Figura 5.36 - Informação detalhada sobre a utilização da aplicação e a percentagem do consumo de bateria, captura de ecrã recolhida a partir da aplicação GSam Battery Monitor	106
Figura 6.1 - Gráfico com a preferência dos utilizadores relativamente a funcionalidades futuras	109
Figura 6.2 - Estatísticas de utilização da aplicação where@UM.....	110
Figura D.1 - Arquitetura da plataforma Android, retirado de [57]	135
Figura D.2 - Ciclo de vida de uma <i>activity</i> , retirado de [57].....	137
Figura D.3 - Ciclo de vida de um <i>service</i> , retirado de [57]	138

Lista de tabelas

Tabela 5.1 - Lista de permissões de utilização	63
Tabela 5.2 - Método de procura de locais próximos ao utilizador.....	83
Tabela 5.3 - Comparação entre <i>fingerprints</i> recolhidas no mesmo local por dispositivos diferentes.....	93
Tabela 5.4 - Resultados referentes ao dispositivo Asus Memo Pad HD7	103
Tabela 5.5 - Resultados referentes ao dispositivo LG 4X HD.....	104
Tabela 5.6 - Resultados referentes ao dispositivo Acer Liquid E2.....	104
Tabela A.1 - Resumo das aplicações estudadas.....	123
Tabela D.1 - Componentes adicionais de uma aplicação Android	139

Lista de abreviaturas

AP	<i>Access Point</i>
API	<i>Application Programming Interface</i>
BSSID	<i>Basic Service Set Identification</i>
CoO	<i>Cell of Origin</i>
DES	<i>Data Encryption Standard</i>
GCM	<i>Google Cloud Messaging</i>
GPS	<i>Global Positioning System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IPS	<i>Indoor Positioning System</i>
JSON	<i>JavaScript Object Notation</i>
LIFO	<i>Last In First Out</i>
MAC	<i>Media Access Control</i>
PDO	<i>PHP Data Objects</i>
PHP	<i>Personal Home Page</i>
PoA	<i>Phase of Arrival</i>
RFID	<i>Radio-Frequency Identification</i>
RSSI	<i>Received Signal Strength Indicator</i>
RTLS	<i>Real Time Location System</i>
RTT	<i>Round-Trip Time</i>
SLAM	<i>Simultaneous Localization And Mapping</i>
SNR	<i>Signal to Noise Ratio</i>

SSL	<i>Secure Sockets Layer</i>
TDoA	<i>Time Difference of Arrival</i>
ToA	<i>Time of Arrival</i>
UI	<i>User Interface</i>
URI	<i>Uniform Resource Identifier</i>
UWB	<i>Ultra Wide Band</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
XML	<i>Extensible Markup Language</i>

1. Introdução

1.1 Enquadramento e motivação

Atualmente, existem inúmeras aplicações móveis que requerem a identificação do local onde a aplicação é executada e/ou onde o utilizador se encontra. Algumas destas utilizam o sistema de localização GPS [1] que, apesar de ser o sistema de localização mais comum, apenas funciona bem em espaços abertos. Houve então necessidade de explorar técnicas que permitissem obter funcionalidades semelhantes às do sistema GPS mas para ambientes interiores. Nos últimos anos tem-se assistido ao surgimento de várias tecnologias que exploram as infraestruturas habitualmente instaladas em edifícios, oferecendo novas funcionalidades. Neste contexto, destacam-se os sistemas, designados por sistemas de localização em tempo real (RTLS), os quais permitem estimar e acompanhar em tempo real a posição de dispositivos a partir da informação proveniente dos pontos de acesso (APs) Wi-Fi. Hoje em dia já existem vários sistemas para determinar a posição de dispositivos móveis em ambientes interiores [2][3][4][5].

Com o elevado número de redes Wi-Fi instaladas em áreas urbanas, tanto em ruas como edifícios, é possível usufruir de uma grande área de cobertura. A disseminação da tecnologia Wi-Fi para dispositivos móveis (*smartphones* e *tablets*) contribuiu em muito para a evolução e generalização desta tecnologia, sendo fácil encontrar pontos de acesso Wi-Fi nos mais variados locais. Perante este cenário, houve um aumento de investigação desta tecnologia para os sistemas de posicionamento em ambientes interiores (IPS), que através da recolha de informação dos pontos de acesso Wi-Fi permitem estimar a posição dos dispositivos. O projeto RADAR foi um dos primeiros a explorar esta técnica. O RADAR regista e processa a informação da potência do sinal recebida proveniente dos vários pontos de acesso detetáveis ao seu redor de modo a estimar a posição dos dispositivos [6][7].

Com a integração destes sistemas de posicionamento em aplicações móveis é possível localizar e monitorizar os utilizadores, uma vez que a localização dos

dispositivos habitualmente corresponde à do seu utilizador. Existem várias áreas aplicacionais em que este sistema pode ser útil, desde a saúde (auxílio a invisuais, monitorização de utentes), as redes sociais (localizar e acompanhar amigos), o turismo (hotéis, museus), a indústria e a segurança, entre muitas outras. Recentemente, alguns centros comerciais têm explorado este tipo de aplicações de forma a auxiliar os utilizadores com um mapa do edifício e o acompanhamento real da posição do utilizador através da rede Wi-Fi instalada no edifício [8].

A maioria dos sistemas de posicionamento que exploram as redes Wi-Fi funcionam em duas fases: fase de calibração (*offline*) e a fase de determinar a localização do dispositivo móvel (*online*). Na primeira fase, é feita a recolha das potências dos sinais recebidos (RSSI) provenientes dos pontos de acesso Wi-Fi nos diversos locais do espaço em que o sistema dá cobertura. Ao conjunto dos endereços MAC dos pontos de acesso (BSSID) detetados juntamente com o respetivo RSSI recolhido, dá-se o nome de Wi-Fi *fingerprint*. Uma das técnicas mais comuns, designada por Wi-Fi *fingerprinting*, baseia-se na ideia de que uma localização pode ser representada por um único identificador. Esta técnica consiste na comparação das *fingerprints* recolhidas por dispositivos móveis (ou por outros tipos de terminais) com *fingerprints* dos locais conhecidos pelo sistema, de forma a tentar estimar a localização do dispositivo. Para este efeito, estes sistemas necessitam de um conhecimento prévio do ambiente rádio dos locais em que o sistema pretende ser utilizado. O ambiente rádio dos locais, ou mapa de rádio, é uma base de dados com *fingerprints* anotadas, isto é, *fingerprints* com um local associado. O seu processo de construção, ou fase de calibração, continua a ser um dos maiores desafios para estes sistemas de posicionamento, principalmente em edifícios de larga escala. O problema da calibração é discutido no capítulo 3.

1.2 Objetivos

O principal objetivo deste projeto é elaborar um sistema de posicionamento baseado na técnica Wi-Fi *fingerprinting* que, através da utilização de uma aplicação para dispositivos móveis, permita construir, de forma colaborativa e autónoma, mapas de rádio para edifícios de grande escala.

Durante o projeto foi necessário identificar e dar resposta a um conjunto de desafios, entre os quais se inserem:

- Como automatizar o processo de construção dos mapas de rádio;
- Como avaliar a viabilidade de uma solução colaborativa para a construção dos mapas de rádio;
- Como envolver as pessoas no processo colaborativo;
- Como avaliar a qualidade dos dados recolhidos e fornecidos pelos utilizadores.

1.3 Abordagem

No sentido de concretizar o objetivo proposto foi desenvolvida uma aplicação móvel de posicionamento destinada, principalmente, aos utilizadores da Universidade do Minho (estudantes, docentes e funcionários). Estes podem partilhar a sua localização com amigos para que, e se estes permitirem, possam encontrá-los com rapidez, facilidade e precisão. A ideia subjacente a esta aplicação é combinar os conceitos de uma rede social com um motor de posicionamento já existente, envolvendo os utilizadores, inconscientemente, na construção do mapa de rádio. O motor de posicionamento existente baseia-se num algoritmo de posicionamento e no conhecimento prévio que o motor possui do ambiente rádio de cada edifício, restringindo a sua utilização aos edifícios que foram previamente calibrados e para os quais foram elaborados mapas de rádio. Para os casos em que a localização não se encontra nos mapas rádio, a aplicação solicita ao utilizador que identifique o local em que se encontra, associando-o à leitura rádio. É desta forma que o sistema aprende novos locais, permitindo-lhe expandir a área de cobertura dos mapas de rádio.

1.4 Estrutura da dissertação

A presente dissertação está dividida em 7 capítulos. O primeiro capítulo, em que esta secção se insere, apresenta uma breve introdução sobre o tema desta dissertação, assim como os objetivos propostos e a abordagem adotada.

No segundo capítulo é apresentado o estado da arte alusivo às técnicas de posicionamento para ambientes interiores, e são descritos dois sistemas de posicionamento com fundamentos semelhantes ao sistema pretendido. São ainda apresentados alguns produtos comerciais e aplicações móveis das quais se extraíram ideias e funcionalidades para integrar na aplicação desenvolvida.

No terceiro capítulo o problema dos sistemas de posicionamento baseados na técnica *fingerprinting* é abordado e é apresentada uma solução para o problema referido.

No quarto capítulo é apresentado o desenho geral do sistema, bem como os requisitos, funcionais e não funcionais, dos componentes do sistema. O processo de construção do mapa de rádio e os elementos do sistema são especificados. Ainda neste capítulo é exposta a estratégia de disseminação da aplicação.

O quinto capítulo é iniciado por um estudo sobre as plataformas de desenvolvimento móvel mais comuns, seguido pela explicação da escolha da plataforma escolhida (Android) para o desenvolvimento da aplicação. Neste capítulo é descrita toda a implementação dos elementos constituintes do sistema, bem como testes realizados durante a fase de desenvolvimento.

O sexto capítulo é relativo à avaliação do sistema implementado, onde é explicado o processo de integração no mundo real e a recetividade por parte das pessoas à aplicação. Por fim são apresentados os resultados obtidos alusivos ao sistema.

O sétimo e último capítulo expõe uma análise geral do trabalho desenvolvido, enumerando os aspetos positivos retirados durante o desenvolvimento deste trabalho, assim como os aspetos menos positivos que foram identificados para posterior correção.

2. Sistemas de posicionamento e aplicações

Neste capítulo são identificados diferentes tipos de sistemas de posicionamento e diferentes técnicas de posicionamento para ambientes fechados. São ainda identificados produtos comerciais que utilizam técnicas de posicionamento baseados em redes Wi-Fi, e são descritas aplicações móveis semelhantes à idealizada para este projeto.

Como já referido, o GPS é o sistema de posicionamento mais comum, é o que oferece maior cobertura e está integrado na maioria dos dispositivos móveis. Contudo, é ineficiente em locais fechados, dado que não é possível a transmissão entre recetores e satélites nestes ambientes. A sua ineficiência conduziu ao estudo de diferentes técnicas de posicionamento para este tipo de ambientes. O posicionamento neste tipo de locais, como edifícios, é mais complexo que em espaços abertos. Em edifícios existem vários obstáculos, como paredes e equipamentos, que influenciam a propagação de ondas eletromagnéticas e dificultam o processo de posicionamento. Tendo em conta estes problemas, foram consideradas e estudadas diferentes tecnologias por empresas e universidades, incluindo-se: infravermelhos (IR) [9][10], ultrassom [11][12], identificação por radiofrequência (RFID) [13][14], redes de área local sem fios (WLAN / Wi-Fi) [2][3][4][5], Bluetooth [15][16], redes de sensores [17], redes com elevada largura de banda (UWB) [18], sinais magnéticos [19], análise de imagens [20][21] e som. Comparações e informações mais detalhadas relativas às tecnologias enumeradas podem ser consultadas em [22][23][24].

Este trabalho incide no estudo de sistemas de posicionamento baseados em redes Wi-Fi. A principal vantagem deste sistema, comparativamente aos restantes, é que não necessita de instalar novos equipamentos, reutiliza a infraestrutura de rede Wi-Fi instalada em edifícios e áreas urbanas. Uma das dificuldades presentes nesta tecnologia consiste na inconsistência da potência dos sinais Wi-Fi que diminui a precisão e dificulta o processo de localização. Os sinais são facilmente afetados por um conjunto de elementos como o movimento e a orientação do corpo humano, paredes, portas, janelas e a orientação do recetor (dispositivo móvel), entre outros. Este problema é

abordado com maior pormenor ao longo do trabalho e é apresentada uma solução que pretende minimizar o problema.

2.1 Técnicas de posicionamento para ambientes interiores

Constituídas por uma ou mais tecnologias de localização, os sistemas de posicionamento para ambientes interiores usam técnicas de posicionamento para localizar objetos/pessoas e para oferecer informações de localização absolutas, relativas ou de proximidade. De seguida são apresentados, de forma superficial, algumas técnicas existentes para determinar a posição nestes ambientes através de várias métricas, tais como proximidade, distância e observações angulares.

- ***Cell of Origin (CoO)***

CoO é uma técnica para determinar a posição de dispositivos móveis que usufrui da infraestrutura da rede móvel. Determina a posição do dispositivo móvel descobrindo em que célula (área de cobertura das antenas de rádio) se encontra e assume a sua posição como sendo a posição da antena de rádio mais próxima. O nível de precisão varia consoante a densidade de células existente num determinado local. Pode ser usada em aplicações com baixos requisitos de precisão.

- ***Lateration / Trilateration / Multilateration***

Estas técnicas determinam a posição através de medidas de distância. O conhecido sistema GPS utiliza a técnica de trilateração (semelhante à triangulação, mas não recorre aos ângulos entre os nós) para determinar a posição. É calculada a distância entre os nós envolventes (os mais próximos) e é assim estimada a posição de um dispositivo. Algumas das medidas de distância utilizadas são: *Time of Arrival (ToA)*, *Time Difference of Arrival (TDoA)*, *Round-Trip Time (RTT)*, *Phase of Arrival (PoA)* e *Signal to Noise Ratio (RSSI/SNR)*.

- ***Dead Reckoning (DR)***

A técnica *dead reckoning*, ou navegação estimada, estima a posição recorrendo a informações sobre posições anteriores, direção do movimento, velocidade e diferença de tempo. Uma das principais desvantagens apontadas a esta técnica é o facto de ser um processo cumulativo, o que provoca uma degradação da precisão ao longo do tempo.

- ***Map Matching*** (MM)

Os algoritmos de *map matching* consistem na comparação de dados recolhidos no momento sobre a posição dos dispositivos com informações dos mapas, de forma a identificar a localização dos dispositivos. Esta técnica é mais utilizada para acompanhar o movimento de veículos ou pessoas em áreas urbanas, é uma alternativa económica por não ser necessário instalar *hardware* adicional.

- ***Fingerprinting***

Fingerprinting, também conhecida por *Scene analysis*, pode ser realizada através de diferentes tipos de grandeza como som e imagem. Mas a mais comum e a que vai ser utilizada neste trabalho é a de radiofrequência RSSI. Esta técnica, como já referida anteriormente, consiste em duas fases distintas. A primeira, fase de calibração (ou *offline*) consiste na recolha do ambiente rádio (*Wi-Fi fingerprint*) nos diversos locais dentro do edifício pretendido que, de seguida são adicionados a uma base de dados (mapa de rádio). A segunda fase (*online*) consiste na deteção da posição, em tempo real, através da comparação do ambiente de rádio recolhido no momento com a informação armazenada no mapa de rádio do edifício, de forma a obter a localização do dispositivo.

2.2 Sistemas de posicionamento

Nesta secção são descritos dois sistemas de posicionamento com fundamentos semelhantes aos idealizados para este trabalho. Ambos baseiam-se na técnica *Wi-Fi fingerprinting* e utilizam um modelo de dados hierárquico. Estes sistemas distinguem-se de outros por não possuírem uma fase de calibração definida, isto é, a calibração é realizada à medida que os utilizadores utilizam a aplicação, corrigindo a posição onde se encontram. Este trabalho pretende implementar e avaliar esta solução num sistema de grande escala.

2.2.1 Herecast

Herecast [25] é sistema de posicionamento baseado na tecnologia *Wi-Fi*, cuja descrição foi publicada na conferência internacional “Wireless And Mobile Computing, Networking And Communications”, 2005 (*WiMob’2005*). Na altura em que foi

desenvolvido, abordava os problemas comuns, para a data, de determinação da posição para ambientes interiores como o elevado custo de *hardware* especializado para o uso, privacidade e dificuldades em configurar os sistemas de posicionamento e personalizar o conteúdo para um local específico.

O sistema Herecast apoia-se em um funcionamento simples: através da recolha do endereço MAC do AP, identificador único dos APs, com maior RSSI, o sistema retorna a informação referente ao mesmo. Esta informação é previamente introduzida por um utilizador na base de dados do sistema que se baseia num modelo hierárquico, com dados mais genéricos como o país, cidade e área até dados mais específicos como sala e andar. O projeto Herecast assume que são os utilizadores do sistema que mantêm estas informações atualizadas. Quando um utilizador deteta um AP desconhecido, a aplicação pede a este que adicione informação sobre o mesmo, preenchendo um formulário visível na Figura 2.1.

The figure shows two screenshots of a web application titled "Where Am I?".

The left screenshot shows the initial form with the following fields:

- Country: Canada
- Province: Ontario
- City: London (with a dropdown menu open showing London, Ottawa, and Toronto)

The right screenshot shows the form with the following fields:

- City: London, Ontario, Canada
- Area: LWC

Below the forms are two identical keyboard layouts with a Scanner icon.

Figura 2.1 - Formulário de registo de um AP, retirado de [25]

Um dos problemas encontrados durante o desenvolvimento do Herecast foi a dificuldade de manter o sistema atualizado por haver constantes alterações de APs em edifícios, podendo mesmo serem removidos ou adicionados novos.

Os dispositivos dos utilizadores do sistema Herecast necessitam de possuir interface 802.11 para poderem recolher os endereços MAC dos APs, e de conexão ao servidor do sistema. A comunicação ao servidor é efetuada por HTTP e o pedido de informação de um AP é feito enviando um pedido GET com o endereço MAC e a inserção de informação relativa a um AP é feita enviando um pedido POST.

Para este sistema de posicionamento foram desenvolvidas algumas aplicações:

- **Area Maps**

A aplicação Area Maps permite consultar informação detalhada sobre um determinado edifício ou espaço através de um mapa personalizado. O serviço Area Map permite a produção de mapas detalhados para uma área específica, por exemplo: campus de uma universidade ou a planta de um centro comercial.

- **Friend Finder**

A aplicação Friend Finder permite saber a última localização conhecida de um utilizador. O dispositivo do utilizador publica automaticamente a localização do utilizador numa página *web* à medida que este se desloca de local para local.

- **Heresay**

O serviço Heresay permite aos utilizadores deixar mensagens na sua localização e ver mensagens que outros utilizadores deixaram no mesmo edifício.

- **Bandwidth Advisor**

A aplicação Bandwidth Advisor avalia a ligação do dispositivo do utilizador à rede Wi-Fi e indica a carga associada a cada ponto de acesso. Se o dispositivo notar que a rede está lenta e congestionada, a aplicação avisa o utilizador para se descolar para uma zona em que a rede tem menor utilização. Além de ajudar os utilizadores ao sugerir áreas onde podem obter uma melhor ligação à Internet, também ajuda a balancear o tráfego de rede.

2.2.2 Molé

Molé [26] é um motor de localização móvel com recurso à infraestrutura Wi-Fi, desenhado especialmente para posicionamento em larga escala. A sua descrição foi

publicada na conferência internacional “Indoor Positioning and Indoor Navigation (IPIN), 2011”, em Guimarães. Molé distinguiu-se por introduzir novas técnicas, entre elas, a forma de associar um nome a uma *fingerprint*, um novo algoritmo estatístico de posicionamento para diferenciar locais vizinhos, redução do atraso de atualização utilizando um detetor de movimento (baseado num acelerómetro) e um sistema de distribuição de *fingerprints* escalável baseado em *cloud*.

Enquanto outros sistemas de posicionamento apresentam um mapa ao utilizador para associar um local clicando num ponto do mapa, como é o caso do OIL [27], ou como o sistema Redpin [28] que permite associar um nome a uma *fingerprint*, sem qualquer padronização, o sistema Molé organiza os nomes de forma hierárquica com cinco níveis: país, região, cidade, área e local específico. A Figura 2.2 mostra a interface de utilizador do Molé com a informação do local atual segundo esta hierarquia. Esta abordagem permite que dados sobre um edifício possam ser previamente carregados para o dispositivo móvel através duma estrutura de dados compacta, possibilitando que este diferencie potenciais locais com *fingerprints* semelhantes.

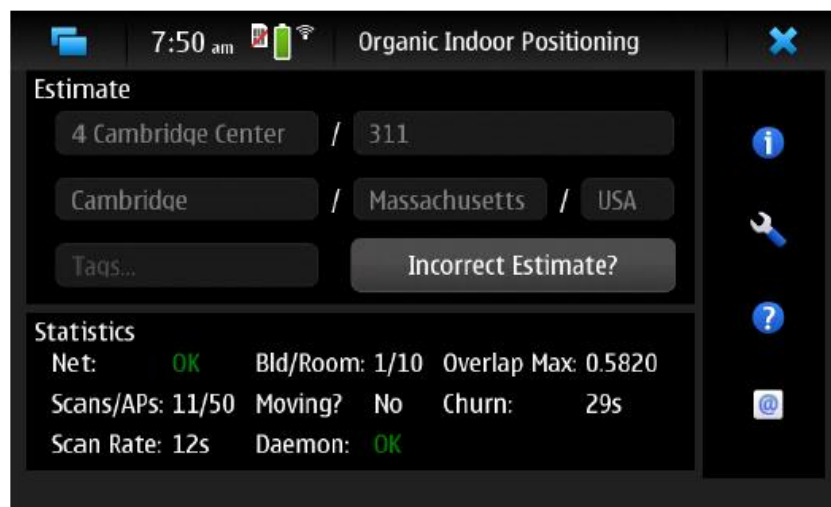


Figura 2.2 - Interface do utilizador da aplicação Molé, retirado de [26]

O algoritmo de localização introduzido pelo Molé, *Maximum Overlap* (MAO), estima o local através da seleção da *fingerprint* que se assemelha mais à *fingerprint* do utilizador recorrendo a uma função de similaridade. Além de ser um algoritmo eficiente em termos computacionais, disponibiliza também uma função de leituras de distâncias que pode ser usada para estimar distâncias entre objetos lidos.

As *fingerprints* MAO são criadas através de leituras RSSI. A cada local é atribuído uma *fingerprint*, para tal, são efetuadas um conjunto de leituras de APs detetáveis e, posteriormente, são apresentados em histogramas com recurso à função de Gauss, como ilustra a Figura 2.3. A *fingerprint* é composta por um conjunto de três parâmetros para cada AP detetado:

$$AP_i \Rightarrow \langle w_i, \mu_i, \sigma_i \rangle \quad (1)$$

Onde w é o peso do AP, μ o valor médio função de Gauss e σ o desvio padrão. O peso é obtido pela divisão do número de vezes que o AP é detetado durante as leituras pelo número total de deteções realizadas. Observando a mesma figura, é possível identificar que das 20 leituras realizadas para o *Place 1* foram detetados três APs, o primeiro foi detetado 20 vezes, o segundo 15 vezes e o terceiro 5 vezes. O peso resultante de cada AP é 20/40, 15/40 e 5/40, respetivamente.

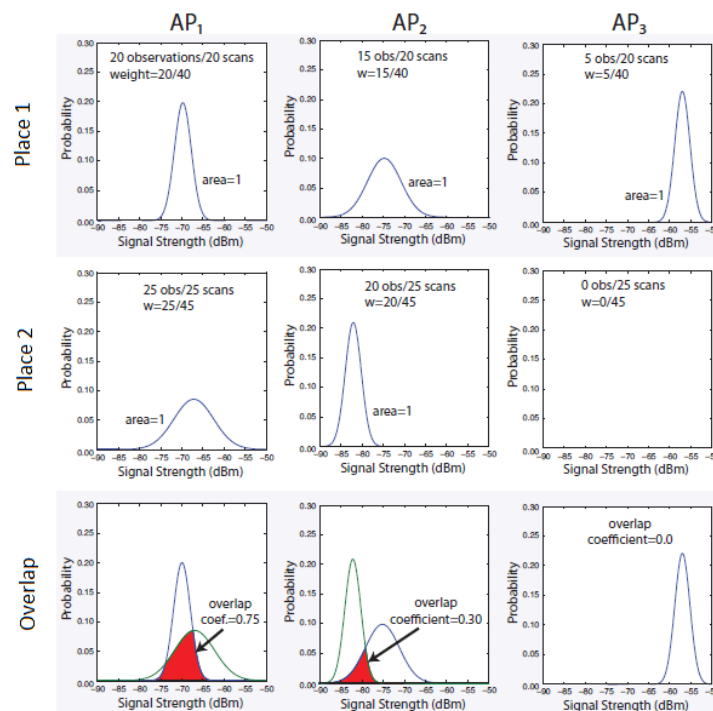


Figura 2.3 - Exemplo do MAO, adaptado de [26]

A semelhança entre *fingerprints* é determinada pela potência do sinal dos APs que existem em ambas as *fingerprints*, e penalizada por APs não encontrados em alguma das *fingerprints*. A última linha da Figura 2.3 (*Overlap*) apresenta um exemplo do cálculo da semelhança entre *fingerprints* do *Place 1* e 2.

O sistema Molé é dividido nos componentes cliente, que periodicamente faz leituras dos sinais Wi-Fi e disponibiliza a posição atual às restantes aplicações do mesmo dispositivo, e servidor, composto por quatro componentes visíveis na Figura 2.4.

O cliente é composto por duas partes, *daemon*, que corre em background e é responsável pelas leituras periódicas já referidas e por disponibilizar a posição atual a todas as aplicações do dispositivo caso o utilizador assim o deseje, e a interface do utilizador (Figura 2.2). Esta mostra ao utilizador a posição atual e alguns dados estatísticos relativos às leituras, como o número de leituras usado para estimar o local e o número de APs detetados. Caso o dispositivo do utilizador detenha um acelerómetro, o Molé usa-o para reduzir o consumo de energia. Isto é, se o dispositivo se encontrar parado, a frequência de leitura e algumas funcionalidades são colocadas em modo de poupança de energia, o que se traduz numa redução da frequência de leitura e o cancelamento de algumas funções. Se o detetor voltar a registar movimento a frequência de leitura e as funcionalidades voltam ao seu funcionamento normal.

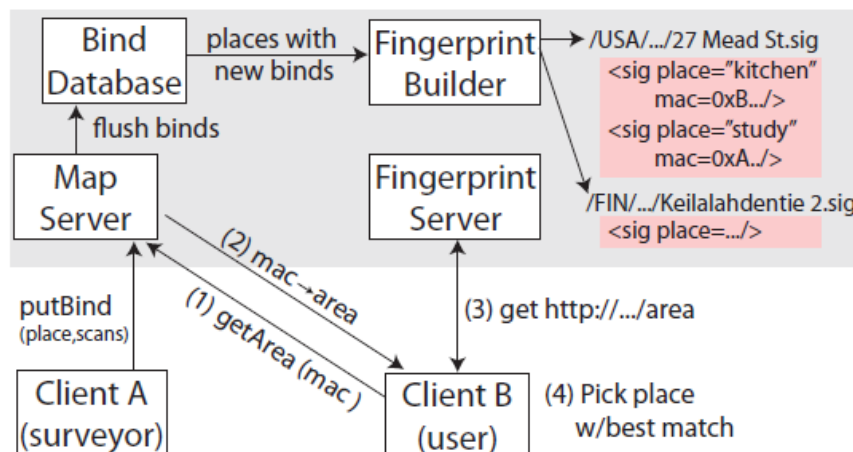


Figura 2.4 - Interação entre componentes do Molé, retirado de [26]

Como já referido, o servidor é composto por quatro componentes principais e por métodos essenciais para que os utilizadores possam introduzir e aceder às *fingerprints* guardadas no servidor (*Fingerprint Server*). Os componentes do servidor que interagem diretamente com os clientes são o *Map Server* e o *Fingerprint Server*. O *Map Server* recebe e reencaminha as *binds* adicionadas pelos clientes para a *Bind Database*. É ainda responsável por disponibilizar aos clientes as potenciais áreas em que estes se encontram, através do método “getArea” que recebe como parâmetro um endereço

MAC de um ponto de acesso aleatório recolhido pelo cliente. O *Fingerprint Builder*, periodicamente, consulta a *Bind Database* para atualizar os locais com novos dados ou para introduzir novos locais. A Figura 2.4 apresenta dois exemplos de interação entre as componentes do Molé, com dois clientes (A e B) a fazerem pedidos ao servidor. O cliente A introduz uma *bind* (leituras de um local) no servidor, este processa as *fingerprints* recebidas armazena no servidor respetivo. O cliente B atualiza a *cache* de *fingerprints* das áreas onde possivelmente se encontra e quando a *cache* estiver atualizada estima a sua posição atual.

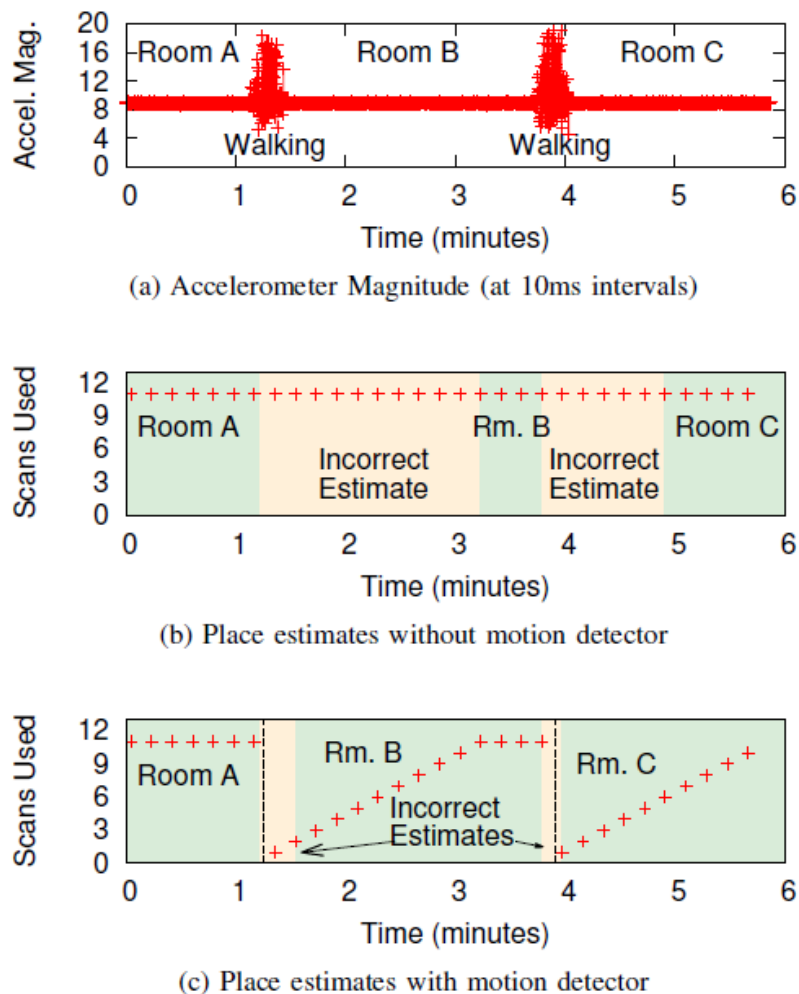


Figura 2.5 - Comparação de leituras realizadas com e sem o detetor de movimento, retirado de [26]

Em [26] foram apresentados alguns testes, e comparações com outras alternativas, relativos à utilização da frequência de resposta, algoritmos de posicionamento, deteção de movimento, utilização da semelhança de *fingerprints* e o comportamento de

crowdsourcing. Os resultados obtidos indicam um aumento de precisão de 10% com o algoritmo de localização MAO, comparativamente ao estado da arte da altura. Mostram como o uso do detetor de movimento pode reduzir significativamente o atraso de atualização e eliminando “falsas leituras”, resultando numa melhor estimativa da posição, como mostra a Figura 2.5. Para avaliar o comportamento de *crowdsourcing*, os autores do Molé disponibilizaram a aplicação a quatro voluntários inexperientes. Após a experiência, os voluntários acharam o Molé bastante preciso, contudo apontaram alguns problemas ao detetor de movimento, em alguns casos este não era suficientemente sensível para detetar movimento. Ainda assim, num curto espaço de tempo preencheram uma base de dados com informação precisa relativa a um edifício com tamanho médio.

2.3 Produtos comerciais

A extraordinária evolução dos dispositivos móveis ao longo dos últimos anos, cada vez com maior poder computacional, possibilitou a exploração e o desenvolvimento de aplicações móveis mais complexas. Empresas e programadores independentes apostam, cada vez mais, nesta área e no desenvolvimento de aplicações móveis por forma a tentarem acompanhar o avanço tecnológico.

Existem atualmente várias empresas que oferecem serviços de posicionamento para ambientes fechados. Entre elas incluem-se a infsoft, a Navizon e a Meridian.

A infsoft [29] oferece produtos de posicionamento, acompanhamento e navegação em edifícios de pequenas e grandes dimensões sem instalar *hardwares* adicionais. Para este efeito, são utilizados vários sensores e tecnologias integrados nos dispositivos móveis atuais (como *smartphones* e *tablets*): GSM, 3G/4G, Wi-Fi, campos magnéticos, compasso, pressão de ar, barómetro, acelerómetro, giroscópio, Bluetooth e GPS. A infsoft integrou, recentemente, a tecnologia realidade aumentada (AR), em que à imagem capturada pela câmara do dispositivo móvel é sobreposta informação de navegação.

Para além do sistema de posicionamento global, em que utiliza as tecnologias GPS, Wi-Fi e GSM, a Navizon [30] oferece também um sistema de triangulação, navegação e acompanhamento de dispositivos móveis e objetos (*tags* Wi-Fi) em edifícios. À

semelhança da infsoft, recorre também ao ambiente rádio dos edifícios e aos sensores instalados nos dispositivos móveis.

A empresa Meridian [31] disponibiliza de um serviço que integra um sistema de localização e navegação, e permite aos clientes criar e desenhar a sua própria aplicação móvel com informações sobre os edifícios (como museus, centros comerciais, entre outros), como páginas, eventos, e pontos de interesse. O motor de posicionamento Meridian utiliza as tecnologias Wi-Fi e GSM.

Existem outras empresas, semelhantes às anteriores, que disponibilizam serviços de localização, e que não foram descritas, são a Insiteo [32], a AlterGeo [33], e a Combain Mobile [34].

2.4 Aplicações móveis

De forma a poder desenvolver uma aplicação móvel que integre elementos de rede social e posicionamento, surgiu a necessidade de conhecer e compreender as funcionalidades de aplicações atuais que se enquadram neste contexto. De seguida são apresentadas as aplicações estudadas, identificando as suas funcionalidades e tecnologias.

2.4.1 Foursquare

O Foursquare [35] é uma rede social baseada em locais (*places*) direcionada aos utilizadores de dispositivos móveis. Permite aos utilizadores partilhar a sua posição geográfica (*check-in*) e procurar pontos de interesse de uma determinada zona, tais como museus e monumentos, restaurantes, centros comerciais e lojas tradicionais. Permite ainda deixar recomendações dos locais que visitou, integrar contactos e redes sociais, como Facebook e Twitter, podendo interagir com estes através de troca de mensagens. De forma a cativar os utilizadores, a aplicação incluiu um extra, a estilo de um jogo, que consiste em atingir *checkpoints* e desta forma, ganhar *badges* (crachás). A aplicação possui ainda um ecrã de perfil e histórico, Figura 2.6, onde se pode encontrar o histórico de *check-ins* e informações relativas ao utilizador (estatísticas,

fotos, *badges*, amigos). A comunidade do Foursquare já ultrapassa os 40 milhões de utilizadores em todo o mundo.

Para determinar a localização, a aplicação recorre ao fornecedor de rede (*network location provider*) e/ou GPS, dependendo da situação e do local em que se encontra. Os mapas que a aplicação disponibiliza são do MapBox [36], criados com dados fornecidos pelo OpenStreetMap (OSM) [36].



Figura 2.6 - Ecrã de perfil e histórico da aplicação Foursquare

2.4.2 Find My Friends!

“Find My Friends!” é uma aplicação de localização para dispositivos móveis Android [38] e iOS [39]. Permite aos utilizadores partilhar a própria posição, procurar e acompanhar amigos/familiares num mapa privado (Figura 2.7), isto é, apenas os utilizadores adicionados, que tenham a partilha de posição ativa, aparecem no mapa. A aplicação possui uma funcionalidade de *geo-fencing*, que possibilita aos utilizadores criar alertas com a entrada e/ou saída de amigos/familiares de uma determinada área. A partilha da localização é controlada pelos utilizadores, podendo-a configurar por grupo ou especificamente por utilizador. Os grupos, referidos atrás são outra funcionalidade da aplicação que permite aos utilizadores organizarem os contactos adicionados em grupos, ou seja, os utilizadores podem agrupar os contactos da forma que lhes seja mais conveniente, agrupando por exemplo em grupos de amigos, familiares e colegas de trabalho. Estes grupos podem ser criados e posteriormente alterados pelo(s)

administrador(es) dos mesmos. Os membros de cada grupo podem trocar mensagens entre si através do serviço de conversação que a aplicação disponibiliza. Cada grupo contém uma sala de conversação associada, na qual todos os elementos do mesmo têm acesso. Caso os utilizadores pretendam trocar mensagens de forma privada, podem-no fazer diretamente com outros utilizadores. A localização é determinada com recurso ao fornecedor de rede e/ou GPS, sendo que com a tecnologia GPS é possível acompanhar o movimento dos utilizadores com maior precisão.



Figura 2.7 - Ecrã principal da aplicação “Find My Friends!”

2.4.3 Crux Indoor Location

A aplicação Android Crux Indoor Location [40] permite monitorizar, em tempo real, a posição de dispositivos móveis em ambientes interiores. Foi desenvolvida com o intuito de auxiliar os clientes de grandes estabelecimentos como por exemplo centros comerciais, exposições e museus. Através da aplicação os utilizadores podem observar o mapa do edifício em questão, acompanharem a sua posição em tempo real e localizar os pontos de interesse do edifício, como é mostrado na Figura 2.8. O Crux oferece aos utilizadores a possibilidade de receber mensagens com promoções, anúncios e avisos de forma gratuita.

A aplicação utiliza a rede Wi-Fi do edifício para localizar a posição do dispositivo móvel. Os mapas dos edifícios necessitam de ser previamente construídos, assim como

os mapas de rádio com a informação correta dos pontos de acesso, de modo a disponibilizar a posição dos utilizadores com uma precisão próxima da realidade.



Figura 2.8 - Ecrãs da aplicação Crux Indoor Location, editado de [40]

2.4.4 Vodafone Radar

Vodafone Radar é um aplicativo para Android [41] e iOS [42] desenvolvido pela Vodafone, a estilo de rede social, e possui funcionalidades semelhantes à aplicação Foursquare. O Vodafone Radar integra as redes sociais Facebook e Foursquare, permite visualizar os *check-ins* feitos pelos seus amigos nas redes sociais referidas e partilhar o seu estado, com referência à sua localização. Através da aplicação, os utilizadores podem consultar que amigos estão perto de si e há quanto tempo efetuaram *check-in* (Figura 2.9), podem consultar pontos de interesse próximos, ler informações extra deixadas por utilizadores que já por lá passaram e descobrir os amigos que já fizeram *check-in* nesses locais. A aplicação disponibiliza também um serviço de alertas (Radar SMS), com uma tarifa associada, no qual os utilizadores podem receber mensagens no dispositivo móvel quando um amigo efetua *check-in* num determinado local.

Para determinar a localização, a aplicação recorre ao fornecedor de rede e à tecnologia GPS para determinar uma posição mais precisa.



Figura 2.9 - Ecrã inicial da aplicação Vodafone Radar, retirado de [41]

2.4.5 Find My Friends

Find My Friends [43] é uma aplicação Android, diferente da mencionada anteriormente, que permite acompanhar amigos em tempo real. Estes podem pertencer às redes sociais Facebook, Google+ e LinkedIn ou a contactos do dispositivo móvel, não sendo necessário efetuar registo no sistema da aplicação nem enviar convites para amigos. A aplicação integra o serviço Google Maps para Android, permitindo acompanhar os amigos através do mapa. Os utilizadores podem escolher quando e com quem querem partilhar a sua posição. Os utilizadores da aplicação podem partilhar mensagens de estado com outros utilizadores, e estas são mostradas no mapa por cima do ícone do utilizador que a partilhou.

A aplicação recorre às API de localização e ao GPS para determinar a posição dos utilizadores e para disponibilizar um acompanhamento em tempo real.

2.4.6 Facebook Phone Tracker

O Facebook Phone Tracker [44] é uma aplicação Android que permite acompanhar e ainda alargar os contactos da rede social Facebook. Através da aplicação é possível encontrar e conhecer pessoas que estão na mesma área geográfica. Para utilizar a aplicação é necessário efetuar *login* com a conta do Facebook, podendo depois seguir e localizar, com recurso à interface do Google Maps, os seus amigos utilizando o GPS e Wi-Fi. Como forma de segurança e controlo de consumo de energia, o utilizador pode

escolher a frequência de atualização da sua posição. A aplicação não possui a opção de esconder a partilha de localização, sendo inevitável mostrar a sua posição durante a utilização da aplicação.

2.4.7 Friend Mapper

A aplicação Friend Mapper [45] foi desenvolvida para dispositivos móveis iOS. Permite partilhar a posição com amigos e seguir os seus movimentos através de um mapa (Figura 2.10), com um limite de 24 utilizadores em simultâneo. A aplicação possui a opção de esconder a posição em qualquer altura, disponibiliza a última atualização efetuada de cada utilizador e possibilita ao utilizador, enviar mensagens e ligar, selecionando, no mapa, o ícone do amigo em questão.



Figura 2.10 - Ecrã da aplicação Friend Mapper, retirado de [45]

2.4.8 Friend Pilot

A aplicação móvel, Friend Pilot [46], para dispositivos Windows Phone, funciona em ambientes exteriores através das tecnologias GPS e Wi-Fi. Permite partilhar a localização entre amigos e obter direções até aos mesmos, de modo a se poderem encontrar com maior facilidade. Os utilizadores podem efetuar *login* pela rede social Facebook ou pela conta da Microsoft, possibilitando a partilha da sua posição com os contactos destas plataformas.

2.4.9 App2Find

A aplicação App2Find está disponível para dispositivos Android [47] e iOS [48], permite encontrar amigos, colegas e familiares. Para além das funcionalidades básicas deste género de aplicações, App2Find inclui a tecnologia de realidade aumentada (AR), que permite ao utilizador procurar amigos ao seu redor através da câmara do dispositivo móvel, como se pode observar através da Figura 2.11. De forma a auxiliar a imagem captada pela câmara, a aplicação apresenta um radar em que mostra os amigos num raio entre 100 m a 30 km. App2Find não prescinde das funcionalidades básicas como escolher os utilizadores que o seguem e quem pretende seguir, visualizar os amigos através de um mapa (Google Maps), configurar a frequência de atualização da posição GPS (entre 1 segundo a 6 horas), como também possibilita personalizar individualmente os perfis dos amigos. Os utilizadores podem filtrar os amigos para evitar uma visualização sobrecarregada ou alterar o alcance máximo para mostrar apenas os amigos que estão a uma determinada distância.



Figura 2.11 - Exemplo de realidade aumentada da aplicação App2Find, retirado de [47]

2.4.10 People Tracker

A aplicação móvel People Tracker [49], para dispositivos iOS, funciona em ambientes exteriores através das tecnologias GPS e Wi-Fi. Permite que os utilizadores se mantenham conectados com os seus empregados, amigos, família, colegas de trabalho ou qualquer outra pessoa que pretenda localizar e os acompanhe, em tempo real, através do mapa da Google (Figura 2.12). Com auxílio do mapa, é possível obter as direções para os amigos e criar áreas de dimensões variadas em que o utilizador é

notificado se algum amigo entra ou sai da respetiva área. A aplicação possibilita ainda visualizar a atividade recente de amigos, criar grupos de amigos e enviar a informação completa sobre a sua posição através do serviço de correio eletrónico.

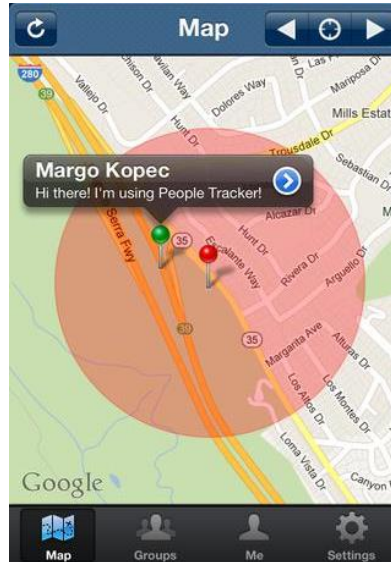


Figura 2.12 - Ecrã da aplicação Friend Mapper, retirado de [49]

2.4.11 Resumo

O estudo de algumas aplicações móveis com finalidades semelhantes às pretendidas para este trabalho permitiu evidenciar algumas funcionalidades básicas para este contexto aplicacional, como a inclusão de amigos, opção de partilha da localização e configuração de conta do utilizador. Foram ainda anotadas outras funcionalidades interessantes que podem enriquecer a aplicação, como a introdução de grupos com partilha da localização configurável para cada um, integração com redes sociais como o Facebook e Twitter, e a criação de alertas por áreas (*geofencing*).

É importante salientar que nenhuma destas aplicações, à exceção da aplicação Crux Indoor Location, integra um mecanismo de posicionamento próprio, nem contribuem para a construção de mapas de rádio. As aplicações estudadas utilizam maioritariamente o serviço de localização da Google, que envolve as tecnologias GPS e Wi-Fi / GSM.

A tabela presente no Anexo A resume as principais funcionalidades das aplicações acima descritas e as técnicas de localização envolvidas.

3.O problema da calibração

O processo de calibração, também conhecido por fase de treino ou fase *offline*, consiste na primeira das duas fases (*offline* e *online*) normalmente adotadas pelos métodos mais comuns de *fingerprinting* para os sistemas de posicionamento. Compete à fase de calibração fazer o reconhecimento do ambiente rádio dos diversos locais do edifício. Nesta fase colecionam-se as forças dos sinais detetados nos vários pontos, criteriosamente, escolhidos da planta do edifício. Em cada ponto é recolhida uma ou mais leituras do ambiente rádio (*fingerprint*) do local e é armazenada juntamente com o identificador do respetivo local (nome do local e/ou coordenadas). Quando este processo termina, considera-se que o sistema está calibrado e pode passar à fase de determinação da posição dos dispositivos em tempo real (fase *online*).

3.1 Descrição do problema

O processo de calibração, ou construção dos mapas de rádio, pode não ser um processo complexo e demorado para edifícios de pequenas dimensões, em que facilmente se recolhem manualmente *fingerprints* dos diversos locais do edifício e se guardam no mapa de rádio as *fingerprints* recolhidas com o local associado, também designadas por *fingerprints* anotadas. O mesmo não acontece para grandes edifícios, uma vez que possuem uma vasta área de cobertura e o processo de recolha manual de *fingerprints* torna-se muito demorado e em alguns casos impraticável [50].

A este problema acresce outro, a alteração da disposição dos pontos de acesso das redes Wi-Fi. Estes podem ser alterados de sítio, em alguns casos são mesmo removidos e adicionados novos, sendo necessário proceder, novamente, à calibração nos locais afetados por estas alterações, o que significa uma atualização dos mapas de rádio com as novas *fingerprints* recolhidas de modo a manter os mapas de rádio coerentes e fiáveis.

3.2 Soluções para calibração automática

Existem atualmente técnicas de calibração automática para ambientes fechados, a mais comum, SLAM (*Simultaneous Localization And Mapping*), é habitualmente utilizada em robôs e permite desenhar um mapa geográfico de ambientes desconhecidos através da observação de diversos fatores ao longo da navegação pelos espaços do edifício. A criação de mapas geográficos através de dispositivos móveis, como *smartphones* e *tablets*, é um processo complexo e requer o uso de uma infraestrutura específica. Os sistemas convencionais de posicionamento de robôs utilizam diversos sensores (como medidor de distâncias laser, sensor sonar multidimensional, sensor de imagem) que não estão, habitualmente, integrados em dispositivos móveis. Em [51][52] descreve-se um sistema de localização e acompanhamento pedestre em ambientes fechados para dispositivos móveis. Equiparam os utilizadores com um dispositivo adicional para o pé, que continha sensores (magnéticos e acelerómetros) que permitiam estimar a orientação dos utilizadores. Apesar deste sistema ter apresentado resultados positivos, a adoção de equipamentos adicionais para auxiliar os dispositivos móveis é dificilmente aceite nos dias de hoje.

A localização de objetos é normalmente representada por coordenadas: longitude, latitude e altitude. Ao contrário das máquinas, os humanos estão mais familiarizados com a representação de locais por nomes em vez da representação por coordenadas usada em robôs. Uma técnica estudada por desenvolvedores, semelhante à técnica referida em cima mas com um nível de abstração superior, consiste na criação de mapas topológicos. Esta técnica baseia-se na construção de mapas através de um método de representação que captura o ambiente do local em oposição à representação geométrica dos espaços [53]. Esta técnica reduz não só a complexidade da criação de um mapa e as exigências computacionais necessárias, como simplifica a perceção dos utilizadores, uma vez, que aos locais são associados nomes, por exemplo: “cafeteria”, “cantina”, “sala do diretor geral”.

Esta solução resolve o problema dos dispositivos móveis acima referido. Permite também que seja adotada a solução utilizada pelos sistemas de posicionamento descritos na seção 2.2, removendo assim a fase de calibração. Passando esta a ser realizada durante a utilização do sistema.

4. Desenho do sistema

Neste capítulo é realizada uma descrição da arquitetura do sistema de posicionamento proposto para ambientes interiores baseado em *Wi-Fi fingerprinting*. A Figura 4.1 apresenta o desenho global do sistema proposto, ilustrando os elementos chave para o funcionamento do sistema, dispositivo móvel e o servidor. O capítulo começa por explicar a ideia geral do sistema. De seguida são especificados os requisitos e a arquitetura do sistema, onde são descritos todos os componentes inerentes ao mesmo. São ainda especificados os protocolos de comunicação utilizados. Por fim, é descrito o processo de construção de mapas de rádio e a estratégia de disseminação da aplicação para dispositivos móveis.

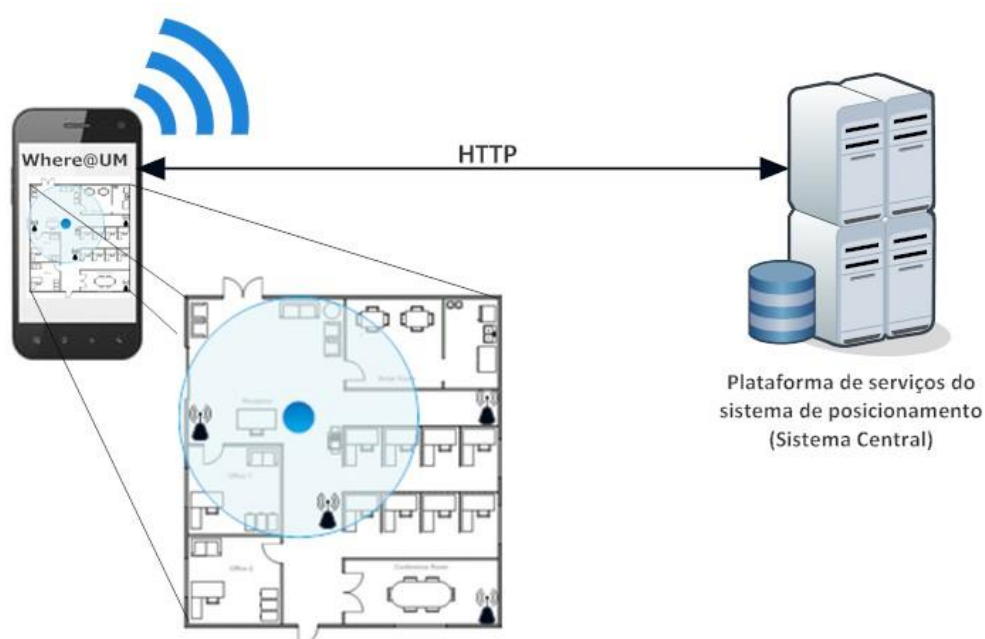


Figura 4.1 - Desenho global do sistema de posicionamento para ambientes interiores baseado em *Wi-Fi fingerprinting*

4.1 Funcionamento geral do sistema

Como inicialmente referido, pretende-se com este sistema solucionar os problemas mais comuns (descritos no capítulo 3) dos sistemas de posicionamento que utilizam a técnica Wi-Fi *fingerprinting* para localizar pessoas e/ou objetos em ambientes interiores.

A solução para o sistema proposto baseia-se numa abordagem colaborativa, onde o objetivo passa por envolver os utilizadores do sistema de posicionamento no processo de construção do mapa de rádio.

A ideia é oferecer aos utilizadores uma aplicação móvel, que combina os conceitos básicos de uma rede social com os de uma aplicação de localização, que lhes permita partilhar a sua localização com os amigos e, ao mesmo tempo, saber onde estes se encontram. Em contrapartida, os utilizadores contribuem com a introdução e anotação de novos locais quando se deparam com locais que ainda não são suportados pelo sistema. O servidor recebe e coleciona as *fingerprints* anotadas pelos utilizadores do sistema que, posteriormente, as utiliza para construir, progressivamente, um mapa de rádio alargado.

4.2 Requisitos

A análise de requisitos engloba as tarefas para determinar as necessidades, funcionalidades e características de um sistema ou de um *software*. Os requisitos podem ser relativos ao utilizador ou ao sistema em questão. Ambos subdividem-se em requisitos funcionais e não funcionais. Os requisitos funcionais descrevem explicitamente as funcionalidades e serviços que o sistema/utilizador deve desempenhar. Os serviços não funcionais definem propriedades e restrições do sistema (desempenho, segurança).

Nesta secção são apresentados os requisitos funcionais e não funcionais alusivos aos principais módulos do sistema.

4.2.1 Requisitos do sistema central

O sistema central engloba todos os componentes do lado do servidor, como tal, deve ser capaz e estar sempre disponível a interagir com os utilizadores e processar todos os pedidos provenientes dos mesmos. O sistema central é responsável por receber, processar e armazenar as *fingerprints* recebidas e conseqüentemente, construir os mapas de rádio.

Requisitos funcionais

- Armazenar e disponibilizar funções de gestão de utilizadores
- Receber e processar informações de posicionamento (*fingerprints*)
- Receber, processar e responder os pedidos provenientes da aplicação
- Criação dos mapas de rádio
- Manter *logs* das operações

Requisitos não funcionais

- Garantir a confidencialidade, integridade, disponibilidade e autenticidade das comunicações e dos dados armazenados
- Garantir a capacidade e disponibilidade dos serviços

4.2.2 Requisitos da aplicação móvel

A aplicação móvel acarreta uma responsabilidade acrescida no funcionamento geral do sistema, é responsável pela recolha periódica de *fingerprints* e envio das mesmas ao servidor. Este processo periódico, ou serviço, é independente da interação do utilizador com a aplicação.

Além do processo de recolha do ambiente rádio em volta do dispositivo, a aplicação permite ao utilizador encontrar os seus amigos e partilhar-lhes a sua localização. As funcionalidades habituais de gestão e configuração da aplicação também estão presentes.

A aplicação móvel deve estar em Inglês por forma a abranger um maior número de utilizadores. Este requisito é importante uma vez que existem muitos alunos, de outras

nacionalidades, a frequentar a Universidade do Minho, por exemplo através do programa Erasmus.

Requisitos funcionais

- Registo e autenticação
- Permitir a gestão de contactos (adicionar/remover amigos)
- Permitir que o utilizador fique *online/offline*
- Permitir que o utilizador faça *check-in* manual
- Permitir que o utilizador configure a partilha da localização
- Permitir que o utilizador configure a frequência de leitura
- Permitir a correção do nome da posição atual
- Apresentar uma lista com os locais atuais dos amigos (ordenado por proximidade)
- Recolher *fingerprints* através da interface Wi-Fi
- Enviar *fingerprints* periodicamente ao servidor
- Caso o sistema de posicionamento não conheça a posição atual, pede ao utilizador que faça o registo da mesma

Requisitos não funcionais

- Conexão com o servidor através da Internet
- Garantia de desempenho e fluidez
- Garantir a confidencialidade, integridade, disponibilidade e autenticidade dos dados do utilizador
- Impedir o acesso à aplicação sem autenticação
- Interface simples e visualmente atrativa
- Facilidade na visualização da localização dos amigos
- Permitir funcionamento em background
- Garantir que a aplicação não consome em demasia a bateria do dispositivo móvel

4.3 Processo de construção do mapa de rádio

Como referido na secção 3.2, a solução adotada para a construção do mapa de rádio, ou calibração, consiste na construção do mesmo durante o funcionamento do sistema. Isto é, à medida que os utilizadores utilizam a aplicação, encontram zonas não mapeadas (ou zonas desconhecidas para o sistema), nessa altura os utilizadores assumem um papel preponderante no funcionamento do sistema e colaboram para o crescimento do mesmo, introduzindo informação sobre o local onde se encontram através do preenchimento de um formulário apresentado pela aplicação. À informação do local é associado uma *fingerprint* recolhida no mesmo instante. Estes dados são enviados para o servidor do sistema, que processa e armazena o novo local na base de dados de suporte ao mapa de rádio do motor de posicionamento.

4.4 Arquitetura da solução

O sistema tem como base a recolha de informação relativa aos pontos de acesso Wi-Fi com o propósito de construir um mapa de rádio de forma autónoma. O processo de recolha de dados dos pontos de acesso é efetuado por um serviço da aplicação através da interface Wi-Fi dos dispositivos móveis. Para o efeito são necessários, de forma geral, os seguintes módulos:

- Módulo da aplicação móvel – capaz de recolher os dados dos pontos de acesso Wi-Fi e de prestar os oferecer as funcionalidades necessários ao utilizador.
- Módulo de suporte à aplicação móvel – responsável por disponibilizar serviços que garantam o correto funcionamento da aplicação.
- Motor de posicionamento (módulo já desenvolvido) – utiliza um mapa de rádio previamente construído para responder a pedidos de localização sobre um determinado dispositivo móvel.
- Módulo de construção do mapa de rádio – responsável por receber e processar as *fingerprints* recolhidas e os novos locais adicionados. Auxilia o motor de posicionamento no processo de construção do mapa de rádio e no posicionamento dos utilizadores.

A Figura 4.2 ilustra os principais módulos do sistema. A distribuição dos módulos confere ao sistema flexibilidade, permitindo que possam ser futuramente utilizados em outras aplicações.

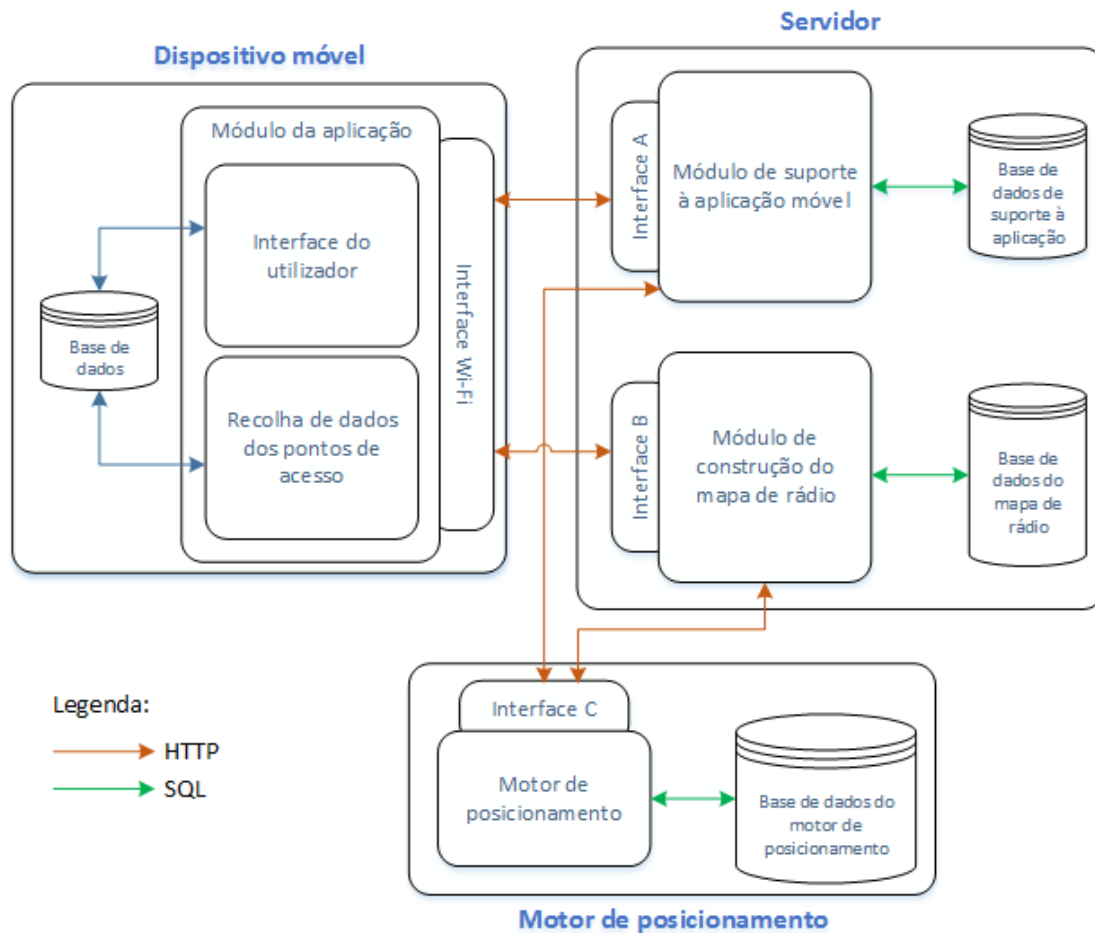


Figura 4.2 – Arquitetura geral do sistema

4.4.1 Aplicação móvel

A aplicação móvel é fundamental no funcionamento geral do sistema. É através desta que os utilizadores se envolvem no sistema e colaboram na construção do mapa de rádio.

Como tal, a aplicação deve compreender um conjunto de requisitos, enumerados na secção 4.2.2. Pretende-se que esta aplicação ofereça funcionalidades aos utilizadores de modo a que estes se sintam interessados e motivados a utilizar a aplicação, e,

essencialmente, sintam a necessidade, à medida que utilizam a aplicação, de adicionar novos locais ao sistema.

A aplicação, além de ser responsável pela interação com o utilizador, possui outra tarefa essencial para o funcionamento da aplicação e do sistema. É através desta que a localização dos utilizadores é mantida atualizada. Esta tarefa é conseguida com recurso ao módulo de recolha de dados dos pontos de acesso Wi-Fi.

Este, é responsável por periodicamente recolher o ambiente rádio Wi-Fi do local onde o dispositivo móvel se encontra e ainda por enviar para o servidor os dados recolhidos. A recolha é conseguida a partir da interface Wi-Fi dos dispositivos móveis. Caso não exista ligação à Internet, os dados são guardados na base de dados local criada para essa finalidade. Os dados guardados são posteriormente enviados para o servidor quando uma *fingerprint* for novamente recolhida e enviada com sucesso. Após o envio de uma *fingerprint*, a aplicação verifica se existem *fingerprints* guardadas na base de dados do dispositivo e, caso existam, são enviadas consecutivamente até um máximo de 10, de modo a não sobrecarregar o funcionamento do dispositivo e do servidor. O envio é feito a partir da *fingerprint* mais recente para a mais antiga (LIFO).

O serviço responsável pela recolha de dados é iniciado, por defeito, de 5 em 5 minutos. Este intervalo pode ser configurável através do ecrã de configurações da aplicação móvel. O serviço executa o processo de recolha em background, impedindo assim que o funcionamento das restantes aplicações e serviços em execução no dispositivo seja comprometido.

Um dos aspetos mais importantes, apesar de ser muitas vezes desvalorizado, é a interface do utilizador. A interface é a imagem do produto, é através dela que os utilizadores interagem com a aplicação. Esta deve ser simples e funcional, mas ao mesmo tempo atrativa para os utilizadores.

*“As far as the customer is concerned,
the interface is the product”*

Jef Raskin

4.4.2 Servidor

O servidor, à direita da Figura 4.2, está dividido em dois módulos distintos: o módulo que dá suporte e disponibiliza os serviços necessários à aplicação móvel e o módulo de construção do mapa de rádio, que interliga a aplicação móvel com o motor de posicionamento, já desenvolvido e independente do sistema.

De modo a desenhar um servidor eficiente e fiável, foi necessário idealizar uma solução que cumprisse os seguintes requisitos:

- Garantir a resposta a pedidos efetuados por vários dispositivos móveis através da Internet;
- Permitir a receção de múltiplos pedidos em simultâneo;
- Garantir a interoperabilidade do serviço, de forma a poder ser utilizado em aplicações futuras para diferentes plataformas, móveis ou *web*;
- Implementar mecanismos de segurança e privacidade;
- Implementar um protocolo de comunicação simples que não sobrecarregue o dispositivo móvel.

Módulo de suporte à aplicação móvel

O módulo de suporte à aplicação móvel é responsável por processar e dar resposta aos pedidos de gestão da aplicação móvel, desde contas de utilizadores, amigos e configurações, assim como disponibilizar a posição dos utilizadores. Os dados relativos aos utilizadores são armazenados na base de dados do servidor. A localização dos utilizadores é realizada aquando a receção de *fingerprints* pelo módulo de construção do mapa de rádio.

Módulo de construção do mapa de rádio

O módulo de construção do mapa de rádio não é apenas responsável por auxiliar o motor de posicionamento a construir o mapa de rádio. Como referido em cima, quando a aplicação móvel envia uma *fingerprint* ao servidor, este realiza um processo de localização do dispositivo móvel/utilizador. O servidor armazena na base de dados, o histórico de *fingerprints* dos utilizadores e todas as *fingerprints* anotadas. Ao armazenar todas as *fingerprints*, anotadas ou não, permite que posteriormente possam ser processadas e enviadas para o motor de posicionamento que as adiciona ao mapa de

rádio. Este processo de integração entre o sistema desenvolvido e o motor de posicionamento é importante para o funcionamento geral do sistema, uma vez que permite, com os dados recolhidos pelos dispositivos e pelos dados submetidos pelos utilizadores, aferir a credibilidade dos dados e excluir *fingerprints* confusas ou *fingerprints* anotadas por engano. Desta forma é possível enviar ao motor de posicionamento apenas *fingerprints* anotadas que cumpram determinados requisitos de qualidade.

O processo de localização do utilizador, descrito na secção 5.2.4, é realizado sempre que o servidor recebe uma *fingerprint*. Através de um algoritmo de processamento, a *fingerprint* é comparada com as *fingerprints* anotadas, e através de um processo de *ranking* é obtida a localização do utilizador.

O módulo é responsável ainda por disponibilizar à aplicação os locais armazenados pelo servidor, tanto os locais referentes à Universidade do Minho como os externos à mesma. A aplicação utiliza estes dados para fornecer um ecrã com um formulário (Figura 5.13), que os utilizadores preenchem quando pretendem corrigir a sua localização ou adicionar um local que ainda não é suportado pelo sistema.

Uma vez que o sistema conhece poucos locais fora da universidade (conhece apenas os locais que os utilizadores vão adicionando), o servidor recorre a dados de uma aplicação de terceiros, o Foursquare, para poder disponibilizar aos utilizadores um maior número de sugestões (locais próximos à posição do utilizador) no momento em que estes pretendam corrigir a sua localização. A integração com a aplicação Foursquare é descrita na secção 5.2.2.

4.4.3 Protocolos de comunicação

Foi adotada a solução de *Web Services* para a comunicação entre o servidor e a aplicação. Os *Web Services* são habitualmente usados em soluções Cliente – Servidor. Permitem a comunicação entre aplicações, independente da plataforma ou linguagem de programação utilizada no cliente. A comunicação é realizada através de pedidos HTTP. Os *Web Services* podem adotar diferentes tipos de interface:

- SOAP, *Simple Object Access Protocol*;
- REST, *Representational State Transfer*.

A escolha recai consoante o tipo de sistema que se pretende implementar. Existem inúmeros estudos relativamente a estas duas técnicas. Em [54], é apresentado um estudo que compara a utilização de *Web Services* REST e *Web Services* SOAP, bem como o uso de diferentes formatos de dados, JSON e XML. Este mesmo estudo apresenta as vantagens e desvantagens de ambas, a comparação das tecnologias utilizadas por ambas (XML e JSON) e um comparativo da utilização das técnicas ao longo do tempo. Em [54] a escolha recaiu sobre o uso de *Web Services* REST com os dados em formato JSON devido à reduzida complexidade dos pedidos e da estrutura dos dados necessários entre o cliente e o servidor.

Neste trabalho também foi adotada a abordagem REST+JSON para a interface do *Web Service*, utilizando o protocolo HTTP para a troca de mensagens. Esta interface foi escolhida por ser a que melhor se adapta às características e requisitos do sistema proposto.

Os *Web Services* REST possuem um conjunto de operações bem definidas:

- GET – Solicita dados ao servidor.
- POST – Utilizada para adicionar um novo recurso com os dados enviados no corpo (body) do pedido.
- PUT – Atualiza um determinado recurso especificado no URI.
- DELETE – Solicita a remoção de um recurso ao servidor.

Os pedidos são efetuados sob a forma de URI, em que constam os nomes dos recursos e o método indica a operação que se pretende efetuar sobre o mesmo. Habitualmente um URI é composto por:

{server}/{arg1}/{arg2}/{arg3}, em que:

- {server}: Endereço do servidor do *Web Service*;
- {arg1}: Nome do recurso;
- {arg2}: Identificador do recurso;
- {arg3}: Descritor adicional para especificar o pedido.

4.4.3.1 Interface A: módulo de suporte à aplicação

A interface do módulo de suporte à aplicação (Figura 4.2) inclui os serviços necessários para o correto funcionamento da aplicação móvel. São eles:

- Registo do utilizador – Serve para efetuar o registo do utilizador no sistema de forma a poder utilizar a aplicação.
- Autenticação do utilizador – Através da introdução das credenciais do utilizador permite que se autentique perante a aplicação.
- Cancelamento de conta – Permite ao utilizador cancelar o registo no sistema.
- Pedidos de amizade – À semelhança de uma rede social é possível convidar utilizadores para pertencer ao grupo de amigos.
- Atualização dos dados de conta – Permite que os utilizadores atualizem dados como password, alcunha e permissão da partilha da localização.
- Obtenção de informação relativa aos amigos – Permite que os utilizadores obtenham informação sobre o estado da relação mantida com outros utilizadores (lista de amigos e lista de pedidos de amizade pendentes).
- Localização do utilizador e amigos – Permite que os utilizadores obtenham a sua localização atual e a localização mais recente dos amigos.

4.4.3.2 Interface B: módulo de construção do mapa de rádio

A interface do módulo de construção do mapa de rádio (Figura 4.2) inclui os serviços para o processo de construção do mapa de rádio, assim como serviços que suportam a aplicação móvel relativos à recolha de dados e à disponibilização de locais ao utilizador. São eles:

- Envio de *fingerprints* – Serviço que o módulo de recolha de dados da aplicação móvel utiliza para enviar as *fingerprints* recolhidas.
- Obtenção dos locais pertencentes à Universidade do Minho – Serviço utilizado pela aplicação para disponibilizar os locais dos campi da Universidade do Minho ao utilizador.
- Obtenção de locais externos à Universidade do Minho – Este serviço permite que a aplicação disponibilize locais próximos ao utilizador quando este se

encontra fora da área universitária. Recorre a uma API de terceiros, Foursquare, para disponibilizar um conjunto maior de locais.

- Introdução de locais – Permite que o utilizador adicione novos locais ao sistema, pertencendo ou não à Universidade do Minho.

4.4.4 Modelo de dados

A adoção da tecnologia de bases de dados em sistemas de informação requer, obrigatoriamente, a modelação dos dados, porque várias aplicações partilham e usufruem do mesmo conjunto de dados. A base de dados deve ser idealizada com o mínimo de redundância, por forma a suportar os dados das aplicação do sistema em questão, estando preparada para alterações em caso de melhoramento das mesmas ou para ser utilizada em novas aplicações.

A modelação de dados é composta por três níveis de abstração:

- Nível concetual: Nível mais alto de abstração, que descreve os elementos constituintes da base de dados.
- Nível lógico: Descreve os dados e as relações existentes entre estes.
- Nível físico: Nível mais baixo, que descreve os ficheiros que constituem a base de dados e como os dados são fisicamente armazenados.

Bases de dados

O sistema inclui um total de duas bases de dados, uma no dispositivo móvel e outra no servidor do sistema, apesar de esta na Figura 4.2 esta dividida em duas. A base de dados do dispositivo móvel armazena temporariamente os dados recolhidos pela interface Wi-Fi do dispositivo, que não são enviados para o servidor por falta de conectividade à Internet, até que o envio dos dados para o servidor seja bem sucedido. Esta base de dados é composta por duas tabelas que registam o instante da recolha e a *fingerprint* recolhida. A base de dados do dispositivo móvel é abordada em detalhe na secção 5.1.3. A base de dados do servidor do sistema guarda os dados do sistema que dão suporte à aplicação móvel e ao motor de posicionamento. Esta é explicada em pormenor na secção 5.2.1.

A base de dados do servidor é composta por dez tabelas e armazena os dados relativos aos utilizadores, o mapa de rádio do motor de posicionamento, bem como os locais previamente inseridos e os locais que os utilizadores adicionam à medida que utilização a aplicação. De seguida são apresentados os principais atributos pertencentes às entidades mais relevantes.

- Informação do utilizador:
 - Identificador do utilizador;
 - Nome;
 - Email;
 - Password (mantida privada);
 - Alcunha;
 - Permissão de partilha da localização.
- Dispositivos móveis do utilizador:
 - Identificador do utilizador;
 - Endereço MAC da interface Wi-Fi do dispositivo;
 - Identificador do registo do dispositivo no serviço GCM.
- Relações de amizade entre utilizadores:
 - Identificador do utilizador que fez o convite;
 - Identificador do utilizador convidado;
 - Data relativa ao convite ou ao momento da confirmação da relação;
 - Estado da relação (confirmada, pendente ou terminada).
- Última localização conhecida do utilizador:
 - Identificador do utilizador;
 - Instante da última localização conhecida;
 - Identificador da última localização conhecida.
- Histórico de *fingerprints*:
 - Identificador da *fingerprint*;
 - *Fingerprint* (é guardada numa tabela própria onde regista o endereço MAC do ponto de acesso e a força do sinal registada);
 - Instante registado pelo dispositivo no momento da recolha da *fingerprint*;
 - Instante que o servidor recebeu a *fingerprint*;

- Identificador do local a que a *fingerprint* corresponde (caso corresponda a algum);
- Identificador do dispositivo.
- *Fingerprints* anotadas (mapa de rádio temporário):
 - Identificador da *fingerprint*;
 - *Fingerprint* (é guardada numa tabela própria onde regista o endereço MAC do ponto de acesso e a força do sinal registada);
 - Instante registado pelo dispositivo no momento da recolha da *fingerprint*;
 - Instante que o servidor recebeu a *fingerprint*;
 - Identificador do local a que a *fingerprint* corresponde;
 - Identificador do dispositivo.
- Locais (tabela dinâmica):
 - Identificador do local;
 - Nome do local;
 - Tipo de local (guardado numa tabela auxiliar);
 - Identificador do local hierarquicamente superior;
 - Identificador do utilizador que adicionou o local (caso tenha sido um utilizador a adicionar o local);
 - Coordenadas GPS do local (caso tenham sido recolhidas).

4.5 Estratégia de disseminação da aplicação

A divulgação deste projeto é essencial para o sucesso do mesmo, uma vez que se trata de uma abordagem colaborativa, em que o envolvimento dos utilizadores é fulcral para o funcionamento do sistema. Como tal, é preciso dar a conhecer a aplicação aos futuros utilizadores, que se entendem ser os utilizadores da Universidade do Minho.

Inicialmente, durante a fase de desenvolvimento, a distribuição da aplicação foi feita por convite direto a alguns utilizadores, de forma a manter o ambiente de teste controlado. Posteriormente foi divulgada a um lote de pessoas maior e por fim, foi anunciada a todos os utilizadores da Universidade do Minho. De modo a que os utilizadores conheçam a aplicação sem necessitarem de instalar a aplicação no dispositivo móvel, desenvolveu-se uma página *web* para mostrar a imagem do produto e expor as funcionalidades presentes no mesmo.

Outra forma que pode potenciar a adesão de novos utilizadores é através do pedido de amizade, realizado por um utilizador na aplicação, a alguém que não faça parte do sistema. O convite chega ao potencial utilizador por meio de email e convida este a instalar a aplicação e a pertencer à comunidade `where@UM`.

A implementação desta estratégia de divulgação é descrita na secção 6.1.

5. Implementação do sistema

Neste capítulo são especificados os elementos constituintes do sistema, bem como as tecnologias utilizadas para a implementação das funcionalidades presentes em cada componente.

5.1 Aplicação móvel

Esta secção explica o porquê da escolha da plataforma de desenvolvimento Android e descreve a implementação da aplicação.

5.1.1 Plataforma de desenvolvimento

Antes de iniciar o desenvolvimento da aplicação móvel foi necessário escolher a plataforma de desenvolvimento. Para tal, foi realizada uma pequena análise sobre as plataformas atuais, mais comuns, para dispositivos móveis. Pretendia-se com esta análise escolher a plataforma que mais se adequasse às características e requisitos do sistema e tivesse maior cobertura de utilização. Como tal, a análise foi realizada sobre as plataformas mais utilizadas:

- Android, da Google;
- iOS, da Apple Inc.

Ambas utilizam linguagens de programação diferentes (Java e Objective-C) e possuem, também, diferentes ambientes de desenvolvimento (Eclipse e XCode). O estudo não incidiu na comparação da dificuldade do desenvolvimento entre as duas plataformas, por cedo se decidir a plataforma que iria ser usada para o desenvolvimento da aplicação devido a condicionamentos tecnológicos impostos pela Apple. Em 2010, a Apple proibiu o lançamento de aplicações que recolhessem informações sobre as redes Wi-Fi [55][56]. Este fator foi decisivo para a escolha da plataforma móvel, uma vez que a recolha de informação sobre os pontos de acesso Wi-Fi é essencial para o

funcionamento do sistema. Já a Google não possui um controlo rígido sobre as aplicações que são colocadas no Google Play e permite todos os serviços necessários para o desenvolvimento da aplicação.

Ainda assim, a esta desvantagem que inviabiliza a escolha do sistema operativo iOS, junta-se outra que em nada favorece a Apple. Os desenvolvedores para a plataforma iOS necessitam de pagar uma quota anual de 80€ para poderem colocar aplicações na App Store. Já os desenvolvedores para a plataforma Android podem colocar gratuitamente aplicações na loja oficial da Google, Google Play. Necessitam apenas de efetuar o pagamento de 25\$ (aproximadamente 20€), aquando do registo como desenvolvedor no Google Play, para poder publicar aplicações na loja oficial. O registo como desenvolvedor é vitalício.

Após a análise efetuada foi decidido desenvolver a aplicação móvel para a plataforma Android, dado que preenche todos os requisitos necessários para o desenvolvimento da aplicação e permite-a publicar na loja oficial da Google com um custo reduzido.

5.1.2 Componentes da aplicação móvel where@UM

Antes de proceder à implementação do sistema foi necessário compreender a arquitetura e os componentes da plataforma Android. O estudo realizado sobre esta plataforma encontra-se no Anexo D.

A Figura 5.1 apresenta, de forma esquemática, os principais componentes utilizados no desenvolvimento da aplicação móvel.

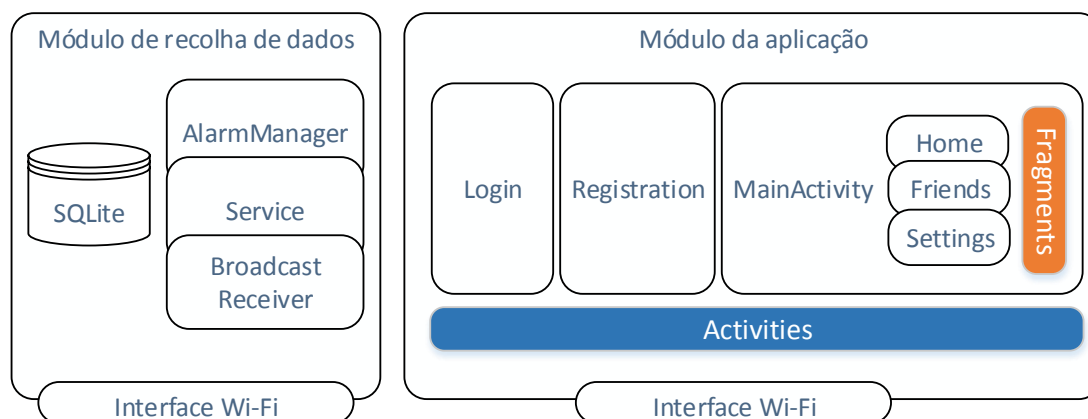


Figura 5.1 - Desenho dos módulos e respetivos componentes

Como referido na secção 4.4 e ilustrado pela Figura 5.1, a aplicação móvel está organizada em dois módulos distintos:

- O módulo responsável pela recolha de dados do ambiente rádio Wi-Fi através da interface Wi-Fi do dispositivo móvel;
- O módulo da aplicação, responsável pela interação com o utilizador.

Permissões de utilização

Aquando da instalação de uma aplicação Android, o utilizador aceita um conjunto de permissões que permite esta utilize determinados dados e funcionalidades, tanto do utilizador como do dispositivo. As permissões de utilização estão definidas no ficheiro XML “Manifest”. Para o desenvolvimento desta aplicação foi necessário definir um conjunto de permissões de utilização, presentes na Tabela 5.1.

Tabela 5.1 - Lista de permissões de utilização

Permissão	Descrição
INTERNET	Permite aceder à Internet.
ACCESS_NETWORK_STATE	Permite aceder ao estado da rede Wi-Fi.
ACCESS_WIFI_STATE	Permite saber o estado da interface Wi-Fi do dispositivo móvel.
WAKE_LOCK	Permite manter o processador em funcionamento, mesmo quando o dispositivo se encontra em modo <i>standby</i> .
ACCESS_FINE_LOCATION	Permite aceder à localização exata do dispositivo móvel através do GPS, rede móvel e Wi-Fi.
ACCESS_COARSE_LOCATION	Permite aceder a localização aproximada do dispositivo móvel através da rede móvel e Wi-Fi.
SET_ALARM	Permite à aplicação programar um alarme para o utilizador ou evento.
RECEIVE_BOOT_COMPLETED	Permite que a aplicação receba a mensagem “ACTION_BOOT_COMPLETED” que é enviada quando o sistema é iniciado.

5.1.3 Recolha de dados do ambiente rádio Wi-Fi

Pretende-se que o processo de recolha de dados das redes Wi-Fi seja executado periodicamente e que não comprometa o funcionamento das restantes aplicações do dispositivo móvel. Interessa que, à medida que os dados são recolhidos, sejam enviados para o servidor. Quando não existir ligação à Internet, a aplicação armazena os dados recolhidos até que exista nova ligação.

De modo a cumprir com estes requisitos foi desenvolvida uma solução que utiliza os componentes presentes na Figura 5.1.

- **AlarmManager**

Esta classe permite à aplicação utilizar os serviços de alarme do sistema. Permite definir que um alarme seja executado periodicamente através do método *setRepeating* que recebe o intervalo entre execuções e o processo que é executado quando o alarme é ativado. É possível introduzir alguns filtros ao alarme que permitem que seja ativado independentemente do estado do dispositivo móvel (*RTC_WAKEUP*), isto é, mesmo que este esteja suspenso (*standby*), o alarme é ativado.

Esta classe foi utilizada para programar um alarme que execute o processo (ou serviço) de recolha de dados periodicamente, com um intervalo de tempo configurável via interface de utilizador.

- **Service**

Um serviço, como explicado no Anexo D, permite executar um processo mais demorado em background. Permite que o processo seja executado sem comprometer o restante funcionamento do dispositivo móvel.

O serviço, iniciado pelo alarme programado, é responsável por iniciar a leitura (*scan*) do ambiente rádio Wi-Fi e registar o *BroadcastReceiver* (através do método *registerReceiver*) que recebe os dados dos pontos de acesso quando a interface Wi-Fi do dispositivo terminar a leitura rádio. Após os dados estarem disponíveis, o serviço envia-os para o servidor. Caso não haja conectividade à Internet, o serviço deve armazenar os dados recolhidos na base de dados do dispositivo móvel, SQLite, através da classe *SQLiteOpenHelper*.

- **BroadcastReceiver**

Permite, como explicado no Anexo D, receber dados provenientes de elementos do sistema. Neste caso o BroadcastReceiver é utilizado para receber os dados dos pontos de acesso Wi-Fi, recolhidos pela interface Wi-Fi do dispositivo móvel, e disponibilizá-los ao serviço que o registou.

- **SQLite**

A base de dados SQLite é utilizada para armazenar os dados que não são enviados para o servidor por falta de conectividade à Internet. A base de dados implementada, Figura 5.2, contém duas entidades que permitem guardar o instante (*timestamp*) em que a leitura foi realizada e a *fingerprint* recolhida. A tabela “aps” guarda a força de sinal dos pontos de acesso detetados. O atributo *motion* não é atualmente utilizado, mas pretende-se que seja implementado em futuras versões. O atributo *motion* servirá para saber se o dispositivo estava, ou não, em movimento no instante da leitura.

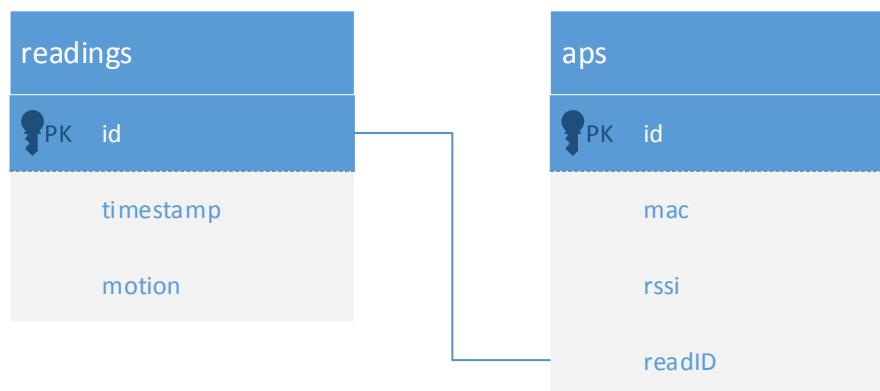


Figura 5.2 - Entidades da base de dados do módulo de recolha de dados

Algoritmo de recolha de dados dos pontos de acesso Wi-Fi

A Figura 5.3 apresenta o algoritmo de funcionamento do módulo de recolha de dados dos pontos de acesso Wi-Fi. Quando o serviço é iniciado, é realizada a inicialização de classes como WifiManager e PowerManager que permitem aceder a informações sobre a interface Wi-Fi e controlar o estado de energia do dispositivo móvel, respetivamente. Após inicializar as classes necessárias, são realizados os processos apresentados na figura. Uma vez que para realizar o processo de recolha de informação dos pontos de acesso é necessário a interface Wi-Fi do dispositivo móvel, é feita uma verificação sobre o seu estado. Apesar de facilmente ser possível alterar o

estado da interface Wi-Fi com recurso à classe WifiManager, vai contra a conduta de programação alterar as configurações do dispositivo sem a permissão e conhecimento do utilizador. Posto isto e tendo em conta que este processo é realizado periodicamente, com um curto intervalo de tempo, o serviço é interrompido caso a interface Wi-Fi esteja desativada.

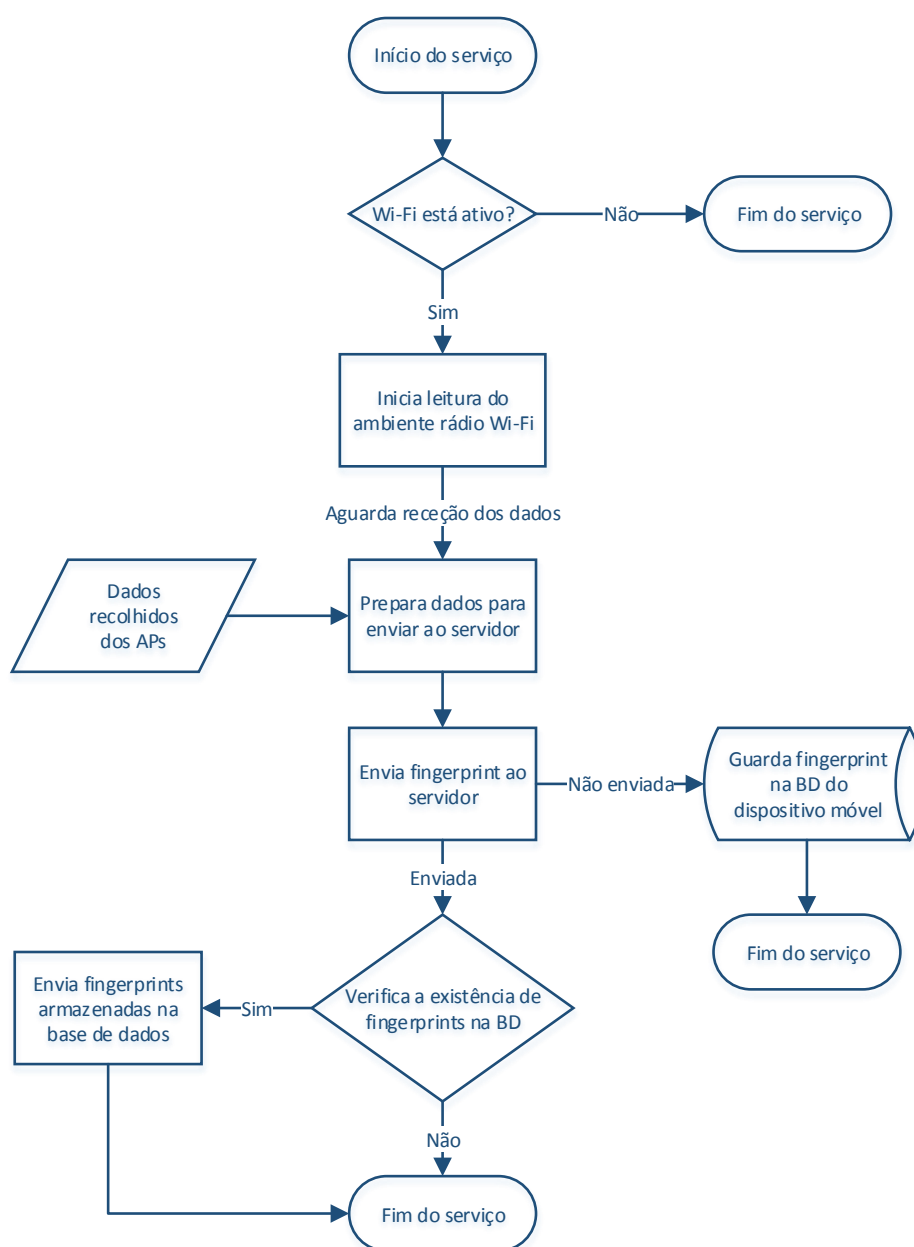


Figura 5.3 - Recolha de dados dos pontos de acesso Wi-Fi

Verificado o estado da interface Wi-Fi, é registado o BroadcastReceiver, que receberá os dados dos pontos de acesso Wi-Fi, e é iniciada a leitura do ambiente de

rádio Wi-Fi. Após recolher os dados, estes são colocados num objeto JSON juntamente com o tempo do momento da leitura (*timestamp*) e o endereço MAC do dispositivo móvel.

Os dados assim preparados são enviados para o servidor. Caso não exista conectividade à Internet ou tenha ocorrido alguma falha no envio, os dados são armazenados na base de dados (Figura 5.2) do dispositivo móvel.

```
{
  "idDevice": "c4:43:8f:cf:5f:8e",
  "timestamp": "2014-06-13 15:17:25",
  "fingerprint": {
    "00:12:43:8a:7f:c0": "-76",
    "00:18:84:81:2c:25": "-61",
    "88:43:e1:13:6d:70": "-85",
    "e0:cb:4e:a8:6c:01": "-86",
    "88:43:e1:13:6f:20": "-88",
    "c0:62:6b:b9:80:90": "-92",
    "00:0e:d7:cd:e7:90": "-88",
    "00:0e:83:cc:05:30": "-74",
    "00:0e:83:cb:f0:a0": "-77"
  }
}
```

Figura 5.4 - Exemplo de uma *fingerprint* em formato JSON

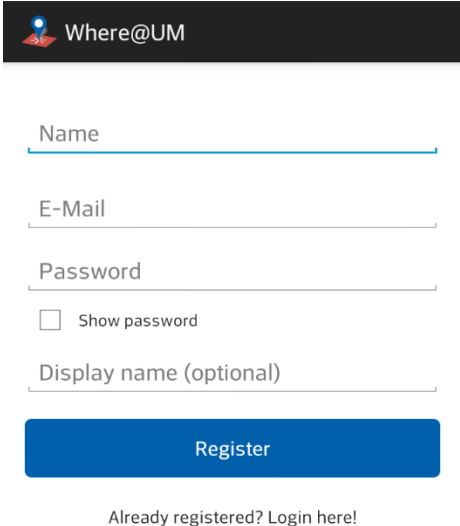
Após o envio da *fingerprint* para o servidor, o serviço verifica se existem *fingerprints* armazenadas na base de dados do dispositivo. Se existirem, é obtida a *fingerprint* mais recente, isto é, a última a ser guardada, e enviada para o servidor. Se o envio for bem sucedido, a *fingerprint* é removida da base de dados e o processo é repetido novamente até um limite máximo de dez *fingerprints* enviadas, de forma a não sobrecarregar tanto o dispositivo como o servidor. Se após este processo permanecerem *fingerprints* guardadas na base de dados do dispositivo, estas são enviadas quando houver nova recolha e posterior envio de *fingerprint* para o servidor bem sucedido.

5.1.4 Ecrãs e funcionalidades

Nesta secção são apresentados os ecrãs e funcionalidades presentes na aplicação móvel desenvolvida, assim como diagramas de sequência considerados pertinentes para a compreensão do sistema.

5.1.4.1 Registo

Atualmente, o registo dos utilizadores é realizado apenas via aplicação móvel (Figura 5.5). A aplicação pede ao utilizador os dados habituais deste tipo de aplicações, como o nome, email e password. Além destes dados, o utilizador pode, caso o pretenda, inserir um nome (*display name*) que é visível pelos amigos do utilizador. Este último pode ser alterado em qualquer altura através do ecrã de definições.



Where@UM

Name

E-Mail

Password

Show password

Display name (optional)

Register

Already registered? Login here!

Figura 5.5 - Ecrã de registo

No momento em que o utilizador submete os dados para o registo, a aplicação, através da classe `WifiManager`, obtém o endereço MAC do dispositivo móvel e junta-o aos restantes dados introduzidos pelo utilizador. Por fim, é enviado um pedido de registo ao servidor. Caso o registo seja efetuado com sucesso, a aplicação guarda os dados do utilizador, com exceção da password, enviados na resposta do pedido e redireciona o utilizador para o ecrã principal da aplicação. Por motivos de privacidade, o servidor armazena a password do utilizador de forma ilegível. O servidor calcula o *Hash* da password através do método “crypt” existente na linguagem PHP, este utiliza

um algoritmo baseado no método criptográfico DES. O texto resultante desta função de *Hash* deriva da combinação da password original e de uma palavra aleatória, designada por *Salt*. A utilização do *Salt* previne a repetição de passwords na base de dados, tornando os ataques *lookup tables*, *reverse lookup tables* e *rainbow* ineficientes. Para fins de comparação é utilizado o mesmo método, é gerado um *Hash* da password original combinada com o *Salt*, que neste caso é o *Hash* da password armazenada.

Este processo é ilustrado pela Figura 5.6.

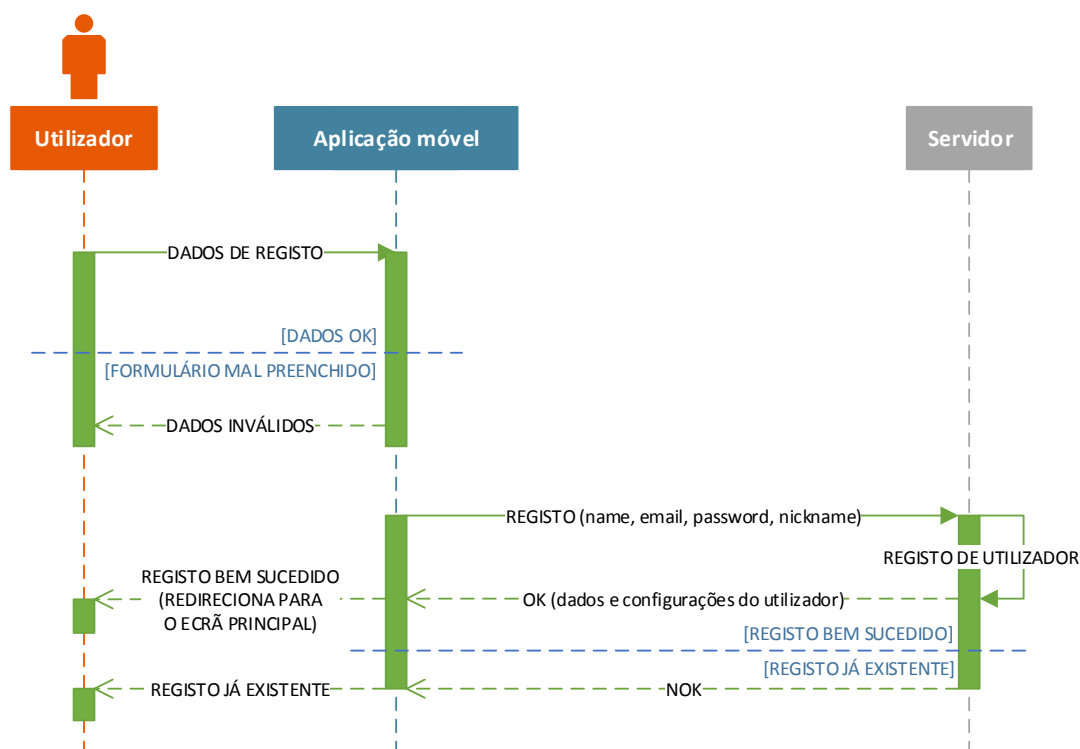


Figura 5.6 - Registo de um utilizador

Registo no GCM

Consumado o registo do utilizador, a aplicação realiza o registo do dispositivo móvel no serviço de mensagens da Google (Figura 5.7). Este serviço permite enviar mensagens do servidor para os dispositivos móveis dos utilizadores. É utilizado essencialmente para enviar notificações aos utilizadores. Neste caso foi utilizado para enviar notificações aos utilizadores quando recebem pedidos de amizade de outros (Figura 5.16).



Figura 5.7 - Arquitetura geral do serviço Google Cloud Messaging

5.1.4.2 Login

Ao iniciar a aplicação, é verificado se existem dados de sessão guardados, caso existam a aplicação redireciona o utilizador para o ecrã principal. O controlo de sessão é realizado com recurso à classe SharedPreferences, que permite guardar dados com um formato simples de pares chave-valor. A aplicação guarda os dados relativos ao utilizador (nome, email, alcunha, identificador do utilizador e configurações de partilha de posição) que vêm no corpo da resposta ao pedido de login e de registo. No caso de não existirem dados de sessão guardados é apresentado o ecrã de login, Figura 5.8.

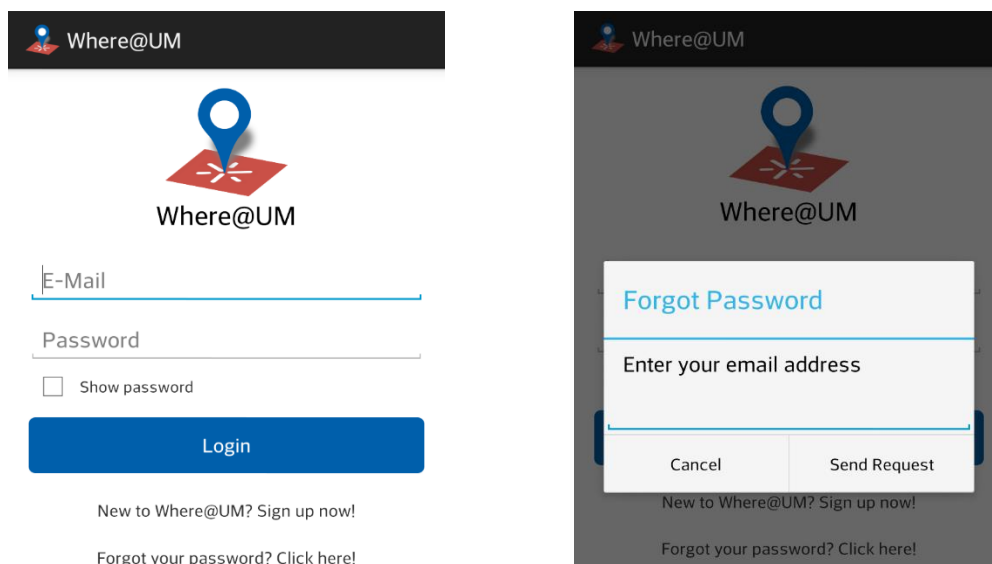


Figura 5.8 - Ecrã de login

Para utilizar a aplicação é necessário iniciar sessão, para tal o utilizador deve preencher os campos obrigatórios de email e password com as suas credenciais. Após inserir os dados de sessão, a aplicação envia um pedido de login para o servidor e, depois de este verificar e confirmar as credencias no sistema, a aplicação redireciona o utilizador para o ecrã principal da aplicação.

O ecrã de login permite ainda, caso o utilizar se tenha esquecido da password de acesso à aplicação, pedir que o sistema envie uma nova password para o seu endereço de email (à direita da Figura 5.8). Ao solicitar o envio de uma nova password o sistema gera uma palavra com cinco caracteres aleatórios, e esta palavra passa a ser a nova password do utilizador. No email, o sistema pede ao utilizador que este, após efetuar o login, altere a password através do menu de definições (Settings).

O processo de login é resumido pelo esquema da Figura 5.9.

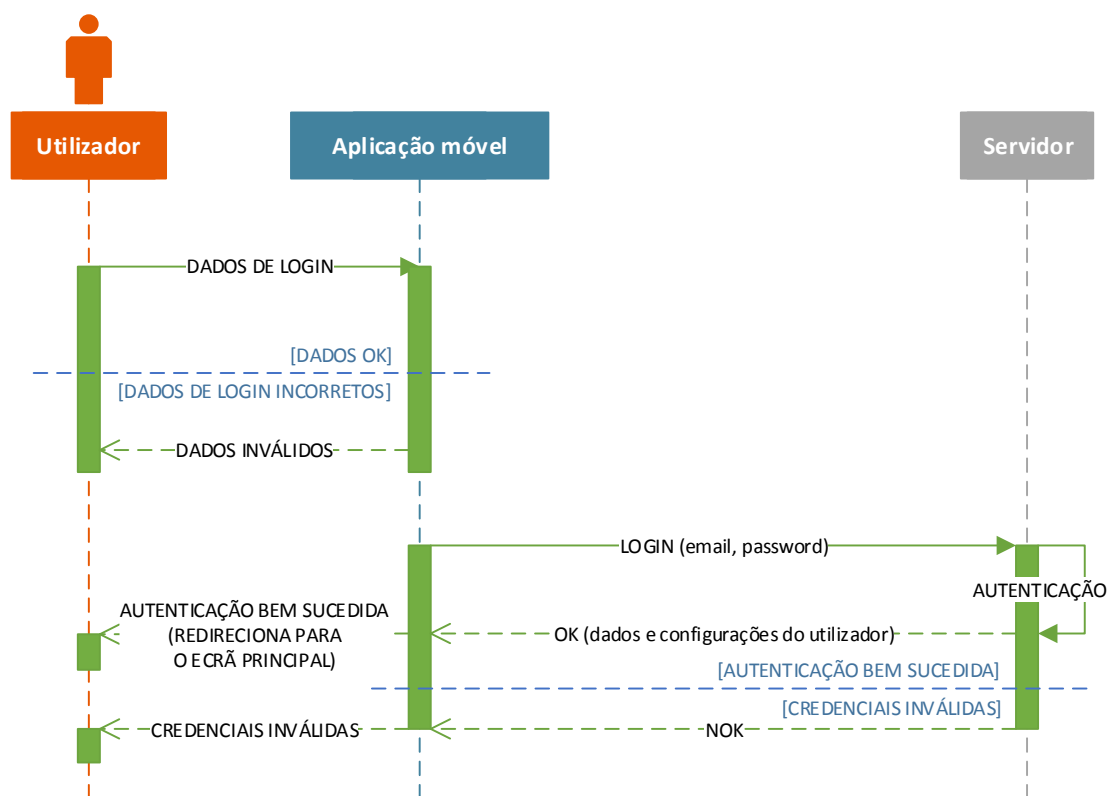


Figura 5.9 - Login de um utilizador

5.1.4.3 Ecrã principal

O ecrã principal (Figura 5.10), para o qual o utilizador é redirecionado após se autenticar, apresenta uma lista com os amigos e respetiva localização. Nesta lista aparecem apenas os amigos que estão a partilhar a sua posição. Através da figura seguinte é possível observar que os amigos do utilizador estão listados por data de atualização da localização, isto é, os utilizadores que se encontram no topo da lista são os utilizadores que, mais recentemente, partilharam uma localização conhecida. Caso um utilizador esteja a partilhar a localização, mas se encontre num local desconhecido para o sistema, a aplicação mostra a última localização conhecida por onde o utilizador passou. De modo a melhorar a perceção dos utilizadores, foi adicionado um círculo verde situado à esquerda dos nomes, cuja intensidade se altera consoante a data da última atualização. Esta particularidade permite aos utilizadores perceber de forma imediata se a informação que está a ver é recente ou não.

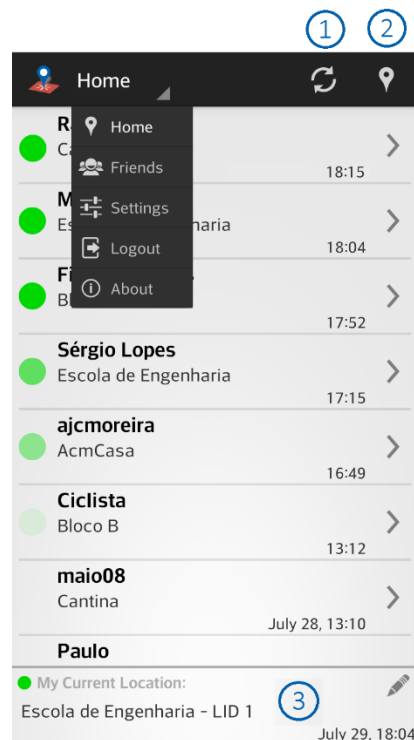


Figura 5.10 - Ecrã principal

Ao selecionar um utilizador é aberto um novo ecrã (Figura 5.11) que permite consultar mais informação sobre o local onde este se situa. A informação que aparece no ecrã difere consoante o local onde o utilizador selecionado se encontra: se estiver

dentro da Universidade do Minho o ecrã apresentado é semelhante ao que se encontra à esquerda na Figura 5.11, enquanto que se tiver fora da Universidade é idêntico ao ecrã da direita.

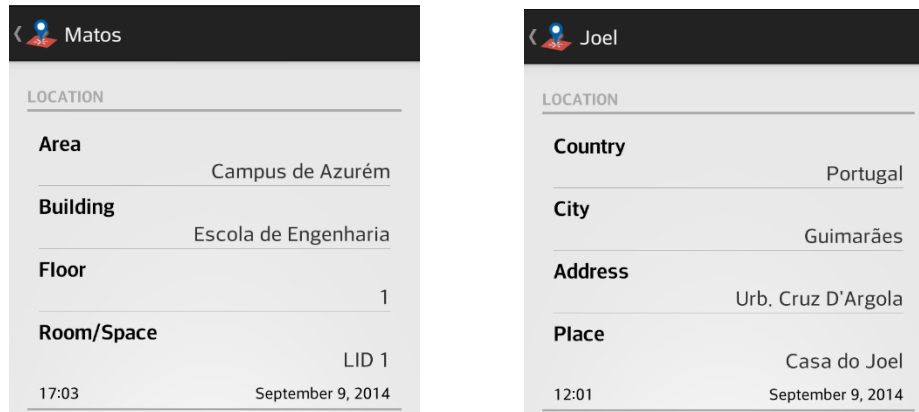


Figura 5.11 - Informação detalhada sobre a localização de um utilizador

Foi escolhido utilizar um menu da aplicação, que está sempre visível nos ecrãs principais, que possibilita a navegação entre os ecrãs da mesma. O menu foi colocado na *action bar* (em português barra de ação), onde também estão presentes botões relativos ao ecrã visível.

Através do botão de atualização (identificado com o número 1 na Figura 5.10) é possível atualizar o ecrã principal, ou seja, é enviado um pedido ao servidor para que este envie informação atualizada sobre os amigos. Todos os dados enviados pelo servidor para a aplicação são previamente preparados e estruturados adequando-se à estrutura da aplicação, reduzindo a carga computacional exercida pelo dispositivo móvel, tornado a aplicação mais fluida. O botão à direita (com o número 2), designado por *checkin*, serve para atualizar a posição do utilizador, isto é, realizar o processo de recolha do ambiente rádio que é feito periodicamente pela aplicação, mas de forma manual. A localização atual do utilizador é mostrada no fundo da Figura 5.10 (área identificada com o número 3). Caso o sistema não conheça o local em que o utilizador se encontra, a aplicação indica que o utilizador se situa numa zona desconhecida para o sistema, “Unknown”.

Nos casos em que o sistema desconhece o local onde o utilizador se encontra ou quando o sistema indica que este se situa num local onde na realidade não está, o utilizador pode corrigir a sua localização, selecionando a área identificada com o

número 3 da Figura 5.10 que o redirecionará para um ecrã específico, semelhante aos da Figura 5.11, mas com a possibilidade de corrigir a localização. Estes casos são ilustrados pela Figura 5.12.

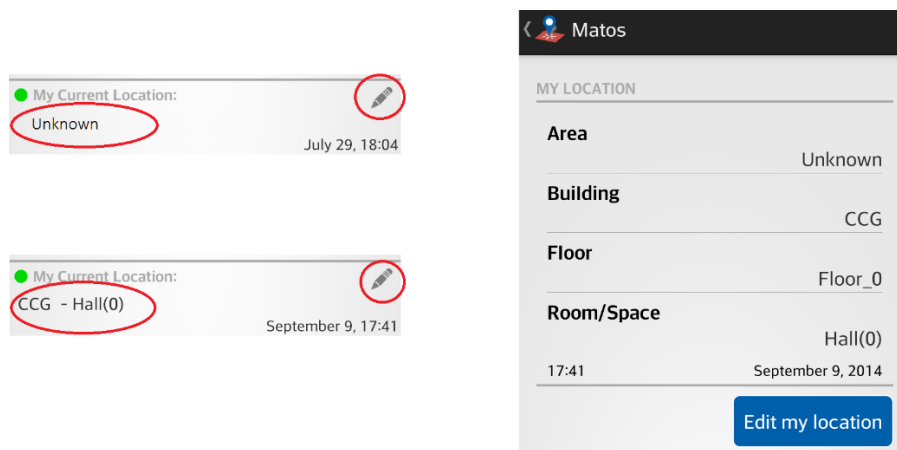


Figura 5.12 - Localização incorreta ou desconhecida. Ecrã que permite consultar a informação sobre o local em que o utilizador se encontra e ainda corrigir a sua localização

Ao pressionar o botão “Edit my location” (ver Figura 5.12), a aplicação redireciona o utilizador para um novo ecrã com dois formulários distintos, um para a Universidade do Minho e outro para fora da mesma, como referido atrás.

O processo de correção da localização é fulcral para o funcionamento do sistema, uma vez que é a única forma de anotar *fingerprints* e fazer com que o sistema cresça. É nesta fase que os utilizadores se envolvem e contribuem para o desenvolvimento do motor de posicionamento. Este método colaborativo vem substituir a fase de calibração dos sistemas comuns de posicionamento baseados em redes Wi-Fi.

A imagem da esquerda da Figura 5.13 permite ao utilizador selecionar a área, o edifício, o andar e o espaço onde se encontra. Todos os campos, à exceção do último, são preenchidos pelo sistema e não é possível adicionar uma nova opção para estes campos. Para o último campo, o sistema disponibiliza os locais que conhece, mas permite que o utilizador introduza novos locais.

The image displays two screenshots of a mobile application interface titled "Edit my location".

The left screenshot shows the "Inside UM" view. It features a header with a back arrow and a location icon, followed by a blue bar with "Inside UM" and "Outside UM" options. Below this, there are four dropdown menus: "Area" (Campus de Azurém), "Building" (Bloco A), "Floor" (0), and "Room/Space" (Salas de Estudo). A blue "Save location" button is at the bottom.

The right screenshot shows the "Outside UM" view. It has a similar header. Below the blue bar, there is a "Select place" dropdown menu (Fundação Cidade de Guimarães), followed by "Country" (Portugal), "City" (Guimarães), "Address" (Centro Cultural Vila Flor), and "Place" (Fundação Cidade de Guimarães). A blue "Save location" button is at the bottom.

Figura 5.13 - Formulários apresentados para corrigir a localização

À direita, na Figura 5.13, é apresentado o formulário que o utilizador deve preencher quando se encontra fora da Universidade do Minho. Para esta situação foi encontrada uma solução que possibilita oferecer um maior número de opções ao utilizador. Para isso o sistema recorre à API disponibilizada pelo Foursquare. Antes de apresentar o ecrã da Figura 5.13, a aplicação obtém as coordenadas GPS da posição do dispositivo e pede ao servidor os locais existentes nas proximidades. O servidor, explicado com maior pormenor na secção 5.2.2, pede ao Foursquare que lhe forneça os locais (designados por *venues*) próximos das coordenadas recolhidas. Os locais retornados pelo Foursquare são comparados com os locais, próximos, existentes na base de dados do sistema. Depois de comparados são enviados de volta à aplicação.

Para obter as coordenadas GPS foi utilizada a interface `LocationListener` e a classe `LocationManager`. A primeira permite receber notificações provenientes da segunda, que disponibiliza o acesso aos serviços de localização do sistema. A classe `LocationManager` dispõe de métodos que possibilita obter atualizações, periódicas ou não, da posição geográfica do dispositivo. A Figura 5.14 descreve as várias etapas para a obter as coordenadas GPS. Inicialmente é selecionado o fornecedor de localização, GPS ou rede (Wi-Fi ou GSM), e de seguida é verificado se a última localização conhecida (através do método `getLastKnownLocation`) pelo sistema é recente, se for a aplicação utiliza as coordenadas GPS da última atualização, caso contrário a aplicação solicita a atualização da localização (`requestLocationUpdates`). Quando é recebida uma

nova atualização (*onLocationChanged*), a classe disponibiliza as coordenadas para a restante aplicação.

Em 17 de Julho de 2013 [58] foi reportado um problema, no centro de gestão de erros da Google, sobre a atualização da localização dos dispositivos móveis em algumas versões Android. O método *onLocationChanged* deixa de ser chamado, por outras palavras, o dispositivo móvel deixa de receber atualizações sobre a sua localização. Este erro ocorre quando é utilizado o fornecedor de localização da rede (Wi-Fi e GSM). O problema fica temporariamente resolvido ao reiniciar o dispositivo móvel, mas apenas temporariamente, uma vez que passado alguns dias volta a acontecer. Até então, não foi encontrada uma solução definitiva que resolva este problema. Este problema já foi experienciado por alguns utilizadores da aplicação e pedir aos utilizadores para reiniciar o dispositivo não é uma solução viável. A versão atual da aplicação mantém-se com este problema que deve ser solucionado em versões futuras.

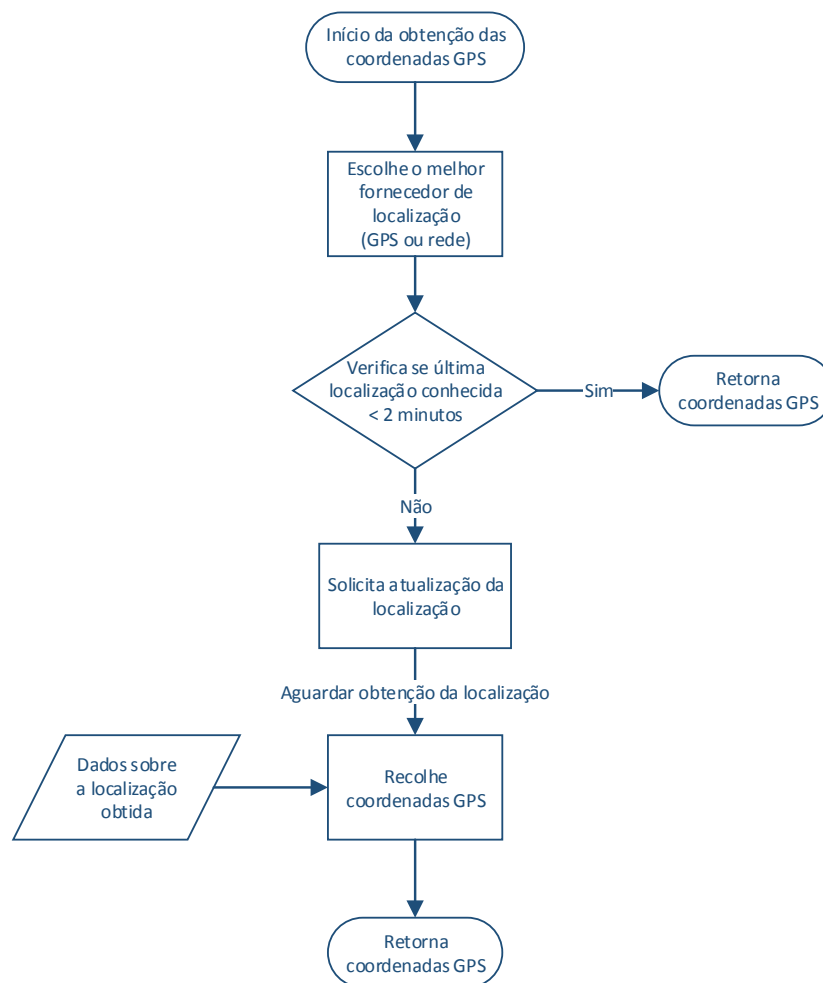


Figura 5.14 - Obtenção das coordenadas GPS

5.1.4.4 Ecrã com informação sobre os amigos

Por intermédio deste ecrã o utilizador pode consultar a sua lista de amigos (à esquerda na Figura 5.15), no qual pode remover a ligação que possui com outro utilizador, visualizar os pedidos que estão pendentes, isto é, pedidos de amizade realizados por outros utilizadores que aguardam por uma confirmação (figura do centro), e consultar os pedidos de amizade efetuados pelo próprio que aguardam a confirmação da outra pessoa.

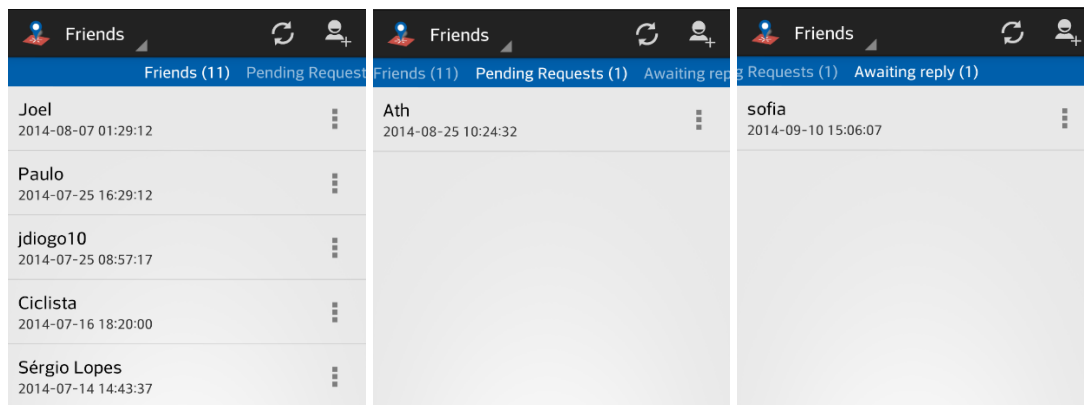


Figura 5.15 - Ecrã com informação sobre o estado das relações do utilizador

Para além das funcionalidades descritas em cima, o utilizador pode enviar convites a utilizadores para pertencer à sua lista de amigos. Ao pressionar no botão alusivo a esta funcionalidade (botão à direita na barra de ação), surge um formulário onde o utilizador deve inserir o endereço de email do amigo que pretende adicionar, e o convite é enviado. Caso o amigo, para o qual o convite foi enviado, não pertença à comunidade where@UM, o sistema envia-lhe um email a referir que foi convidado por um amigo para pertencer ao sistema, e é-lhe disponibilizado o *link* para efetuar o *download* da aplicação móvel. Se o utilizador convidado já pertencer ao sistema, recebe o convite de uma nova amizade em forma de notificação na aplicação, Figura 5.16.

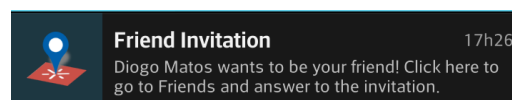


Figura 5.16 - Notificação de um pedido de amizade

5.1.4.5 Ecrã de definições

Através deste ecrã o utilizador pode configurar tanto a aplicação como alguns dados relativos à conta pessoal. O ecrã de definições, Figura 5.17, permite ao utilizador escolher se pretende ou não partilhar a sua localização com os amigos, configurar a frequência de atualização da localização, alterar o nome visível pelos amigos, alterar a password e cancelar a sua conta.

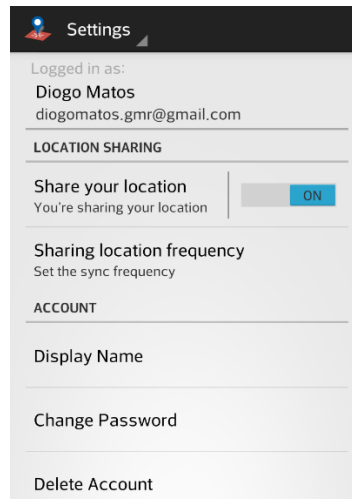


Figura 5.17 - Ecrã de definições

5.2 Servidor

De modo a conceber o servidor do sistema, surgiu a necessidade de escolher uma solução que melhor se adaptasse às características e requisitos do sistema. A solução adotada, referida e justificada na secção 4.4.3, baseia-se na utilização de *Web Services* REST com a integração da linguagem JSON para o formato dos dados. Por motivos de organização e flexibilidade, decidiu-se implementar dois *Web Services*, um para cada módulo do servidor. Esta abordagem permite que os módulos possam ser aproveitados, individualmente, por aplicações futuras necessitando apenas de alguns ajustes.

A linguagem de programação escolhida para o desenvolvimento dos *Web Services* foi o PHP. Como referido em [54], a linguagem PHP é considerada indicada para a implementação dos *Web Services* REST por ser uma linguagem simples e com bastante documentação sobre a implementação destes serviços. Além das vantagens enunciadas, o PHP possui uma fácil integração com as bases de dados. Para a comunicação entre a

base de dados e os *Web Services* foi utilizada uma classe da linguagem PHP específica para esta finalidade, designada por PDO.

A Figura 5.18 expõe as características e tecnologias dos componentes do servidor.

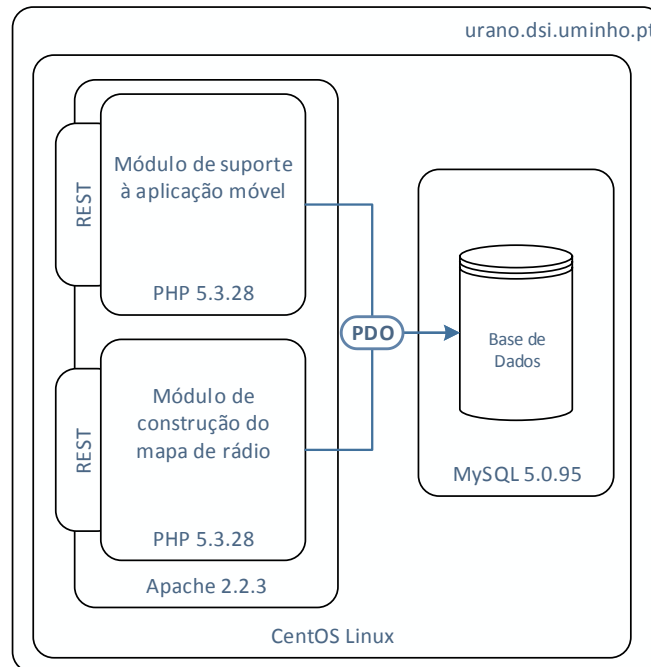


Figura 5.18 - Tecnologias do servidor

5.2.1 Base de dados

Nesta secção são apresentadas as entidades, e as relações entre as entidades, que constituem a base de dados implementada (Figura 5.19 e Figura 5.20), que dá suporte à aplicação móvel e ao módulo de construção do mapa de rádio.

A Figura 5.19 apresenta as entidades que armazenam as informações relativas aos utilizadores, como os dados de conta, os dispositivos, as relações mantidas entre utilizadores e a última localização conhecida. A Figura 5.20 apresenta as entidades associadas ao sistema de posicionamento. Estas guardam as *fingerprints* recolhidas pelos dispositivos móveis dos utilizadores, as *fingerprints* anotadas pelos utilizadores, os locais da Universidade do Minho, previamente inseridos, e os locais que os utilizadores adicionam à medida que utilizam a aplicação e se deparam com novos locais.

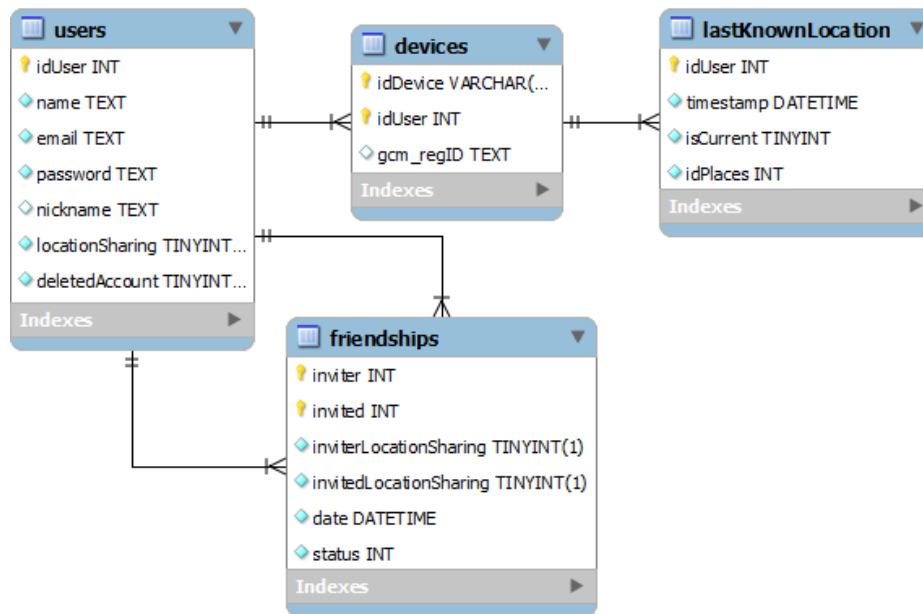


Figura 5.19 - Entidades com informações relativas aos utilizadores

Utilizadores

A tabela “users” guarda essencialmente os dados de conta dos utilizadores, o estado da conta (caso o utilizador elimine a conta, o servidor mantém os dados do utilizador) e o estado de partilha da localização. Este último permite que o utilizador possa escolher não mostrar a localização aos restantes utilizadores (amigos), mantendo a sua privacidade sempre que o desejar.

Os dispositivos móveis dos utilizadores são guardados na tabela “devices”. Apesar de não estar em funcionamento a base de dados está preparada para a utilização de múltiplos dispositivos pelo mesmo utilizador. Nesta tabela é guardado o endereço MAC da interface Wi-Fi do dispositivo móvel e o identificador do registo efetuado no serviço de mensagens do Google (GCM).

As relações entre utilizadores são guardadas na tabela “friendships”. Esta tabela guarda o identificador do utilizador que efetuou o pedido de amizade, o identificador do convidado e o estado da relação (confirmada, pendente ou terminada). A tabela “friendships” está preparada para uma futura funcionalidade que permite aos utilizadores escolher não mostrar a sua localização individualmente.

Quando a aplicação solicita a localização dos utilizadores, o servidor recorre à tabela “lastKnownLocation” que guarda a última localização conhecida dos

utilizadores. À medida que a aplicação móvel envia *fingerprints* para o servidor, este tenta associar um local à *fingerprint* e, caso seja bem sucedido, a tabela “lastKnownLocation” é atualizada e o atributo “isCurrent”, que indica se a informação guardada na tabela está atualizada ou não, é colocado a “1”.

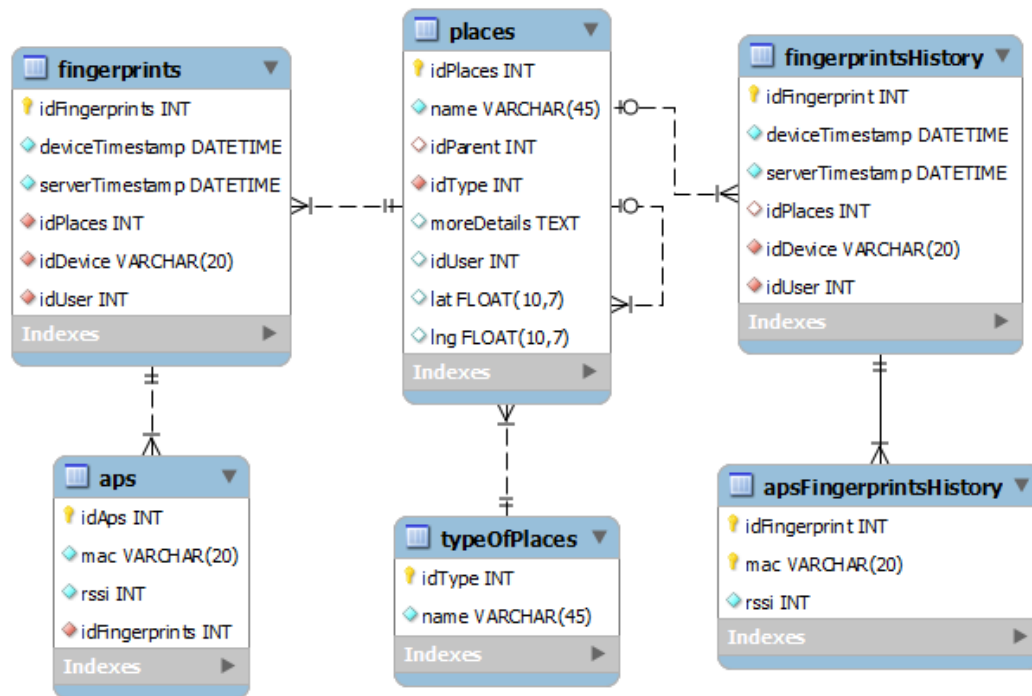


Figura 5.20 - Entidades com informações relativas ao sistema de posicionamento

Locais

A tabela “places”, presente na Figura 5.20, armazena os locais do sistema. É auxiliada pela tabela “typeOfPlaces” que contém os tipos de locais existentes na tabela “places” (país, cidade, universidade, área, edifício, andar e sala/local). Através do atributo “idParent” é possível construir uma sequência até um determinado local. A Figura 5.21 apresenta dois exemplos, um referente à Universidade do Minho e um outro local fora da universidade. Foi adotada uma abordagem que permite uma maior flexibilidade no armazenamento dos locais, uma vez que o sistema é essencialmente direcionado para o mapeamento da Universidade do Minho, mas ao mesmo tempo, permite adicionar locais fora da mesma.

A tabela “places” regista o identificador do utilizador que adicionou cada local por motivos estatísticos. Esta informação pode ser útil no futuro com a introdução de novas funcionalidades.

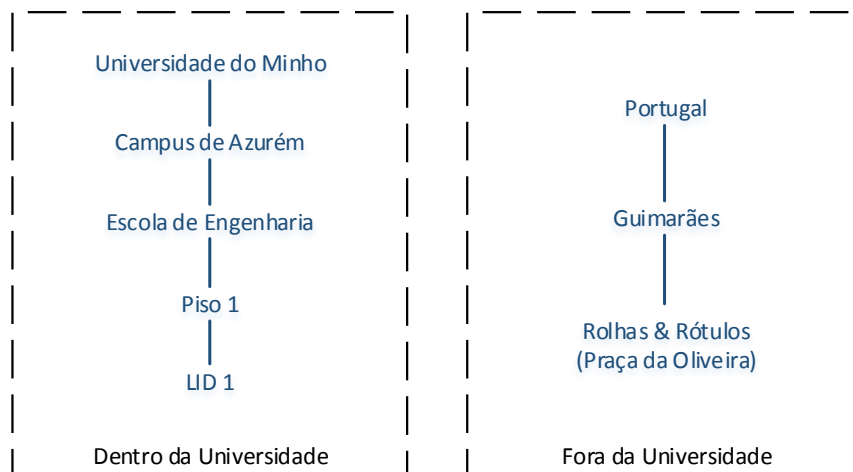


Figura 5.21 - Exemplo de locais, dentro e fora da Universidade

Fingerprints anotadas

A tabela “fingerprints” armazena as *fingerprints* anotadas pelos utilizadores. As *fingerprints* anotadas são também designadas por *fingerprints* de confiança, uma vez que são adicionadas quando os utilizadores pretendem corrigir a sua localização ou quando adicionam um novo local (associado à sua localização) ao sistema. Desta forma, o sistema confia que a *fingerprint* recolhida corresponde ao local indicado pelo utilizador.

Histórico de fingerprints

As *fingerprints* recolhidas periodicamente pelos dispositivos móveis dos utilizadores são armazenadas pelo servidor. O servidor ao receber uma *fingerprint*, mesmo que antiga, submete-a ao processo de comparação de *fingerprints* com o objetivo de lhe tentar associar um local.

O armazenamento de *fingerprints* pode ser útil para o processo de integração com o motor de posicionamento já existente. Com esta informação também é possível, se for considerado interessante para incluir na aplicação móvel, acrescentar a funcionalidade de histórico de locais visitados por cada utilizador.

5.2.2 Integração com o Foursquare

A integração com o Foursquare permitiu solucionar o problema sobre a falta de dados relativamente a locais fora da Universidade do Minho. Uma vez que, inicialmente, ao contrário dos espaços da Universidade, o sistema não continha uma lista de locais externos à Universidade, à exceção de alguns países e cidades que foram previamente inseridas, criou-se um mecanismo para facilitar a criação desta lista. A ideia de utilizar os dados do Foursquare surgiu na altura do levantamento das funcionalidades das aplicações, descritos na secção 2.4. Da mesma forma que a aplicação oferece aos utilizadores o mapa da Universidade (árvore de locais), era interessante também disponibilizar alguns locais quando o utilizador se encontra em locais públicos como cafés, restaurantes, museus e centros comerciais. Após explorar a documentação da API do Foursquare [59], verificou-se que existem métodos que permitem o acesso a informações sobre os locais armazenados na base de dados do Foursquare. Através das coordenadas GPS recolhidas pelo dispositivo móvel é possível recolher os locais próximos ao utilizador. O método utilizado pelo servidor para obter a lista de locais próximos ao utilizador é descrito pela Tabela 5.2.

Tabela 5.2 - Método de procura de locais próximos ao utilizador

Método HTTP	GET
URL	api.foursquare.com/v2/venues/search
Parâmetros	<ul style="list-style-type: none"> ▪ ll Coordenadas GPS da localização do utilizador ▪ limit Limite do número de resultados ▪ intent Tipo de pesquisa ▪ v Data que essencialmente representa a versão da API do Foursquare ▪ client_id Credenciais de utilização da aplicação ▪ client_secret
Resposta	Conjunto de objetos com a informação dos locais retornados

A resposta proveniente do Foursquare é composta por um conjunto de objetos (no máximo 15), em que cada um é relativo a um espaço ou estabelecimento e possui informação sobre o mesmo, como a localização, categoria, e estatísticas (número de *checkins*, número de dicas), entre outros. O servidor, após obter a resposta, processa os dados recebidos e extrai apenas alguns dados, de cada local, relevantes para posterior utilização por parte do sistema (Figura 5.22). Dos dados recebidos, são utilizados apenas os que contêm informação sobre o país e cidade, e estão a uma distância máxima de 150 metros da posição referida, isto porque para o contexto em questão não é relevante apresentar ao utilizador locais distantes do mesmo.

```
{
  "id": "4bd987cd0115c9b69a937780",
  "name": "Coconuts",
  "location": {
    "address": "Pr. da Oliveira",
    "lat": 41.442758,
    "lng": -8.292786,
    "distance": 20,
    "cc": "PT",
    "city": "Guimarães",
    "state": "Braga",
    "country": "Portugal"
  }
}
```

Figura 5.22 - Dados extraídos de um local proveniente do Foursquare

Depois de obter e processar a lista de locais do Foursquare, o servidor obtém uma lista com os locais próximos ao utilizador existentes na base de dados do sistema. Estes locais, externos à Universidade do Minho, foram previamente introduzidos pelos utilizadores. O servidor consegue calcular a distância entre o utilizador e os locais da base de dados através da função *Vicenty great-circle distance* (equação 2), cuja recorre às coordenadas associadas aos locais armazenados na base de dados do servidor e às coordenadas enviadas pelo dispositivo móvel do utilizador.

Após obter esta lista de locais, o servidor agrupa ambas as listas e exclui os locais duplicados (possíveis locais presentes nas duas listas). Finalmente, o servidor disponibiliza a lista resultante com os locais próximos ao utilizador apresentados por ordem de distância, menores distâncias primeiro (ecrã da direita da Figura 5.13).

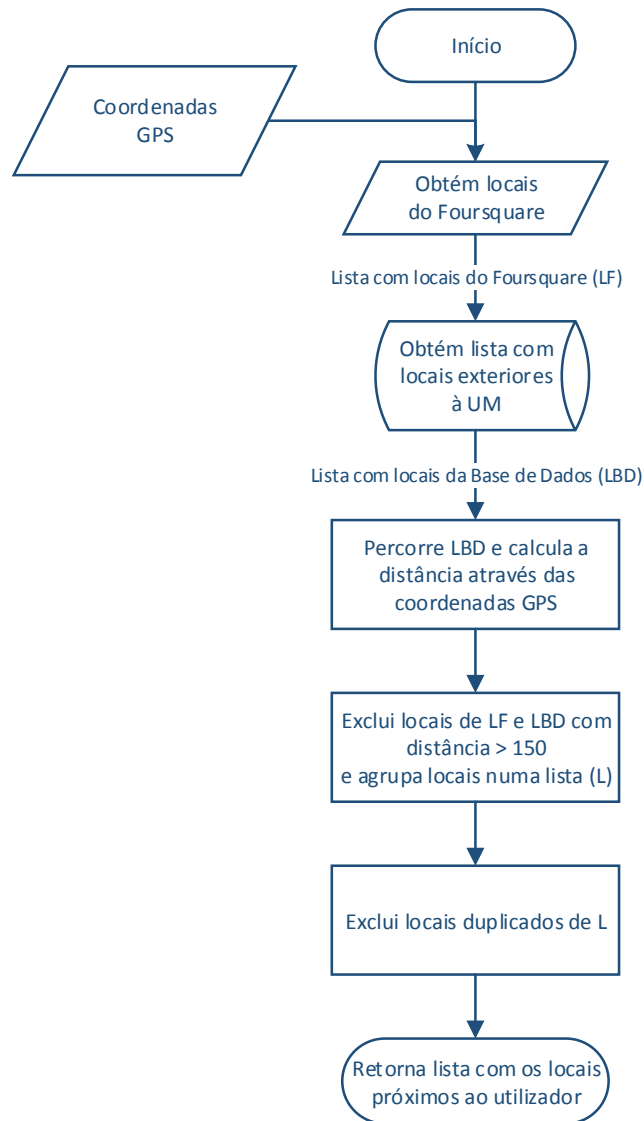


Figura 5.23 - Integração com o Foursquare

A distância entre os locais armazenados na base de dados do sistema e o utilizador, Figura 5.23, são calculados através da função *Vicenty great-circle distance*, que calcula a distância entre dois pares de coordenadas. A equação 2 apresenta a fórmula *great-circle distance*, em que a latitude é representada por λ e a longitude por ϕ .

$$\Delta\sigma = \arctan\left(\frac{\sqrt{(\cos\phi_2 \sin\Delta\lambda)^2 + (\cos\phi_1 \sin\phi_2 - \sin\phi_1 \cos\phi_2 \cos\Delta\lambda)^2}}{\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos\Delta\lambda}\right) \quad (2)$$

5.2.3 Obtenção da localização dos utilizadores

A obtenção da localização dos utilizadores é solicitada regularmente pela aplicação móvel ao servidor a fim de manter a informação dos utilizadores, presente no ecrã principal, atualizada (Figura 5.10). A localização do utilizador e dos amigos é solicitada através do mesmo pedido, podendo dividir-se o processo em duas fases.

- **Obtenção da própria localização**

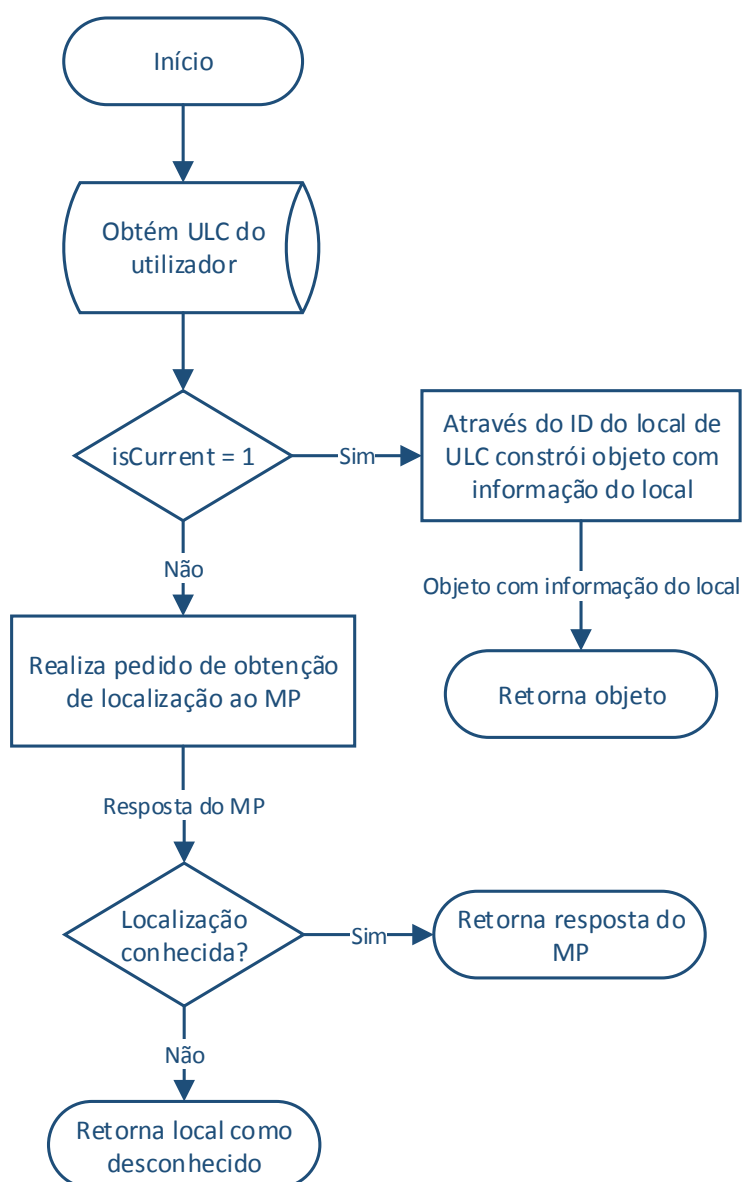


Figura 5.24 - Obtenção da própria localização

Inicialmente o servidor recorre à última localização conhecida (ULC) do utilizador através da tabela da base de dados “lastKnownLocation”. Verifica se a informação presente na tabela está atualizada através do atributo “isCurrent”. Se este atributo estiver a “0” significa que a aplicação enviou *fingerprints* mais recentes comparativamente à guardada em “lastKnownLocation”, mas o sistema não conseguiu identificar o local. Nesta situação o servidor envia um pedido de obtenção da localização do dispositivo móvel do utilizador ao motor de posicionamento (MP). Caso a conheça, o motor de posicionamento envia os dados sobre a localização do utilizador e o servidor utiliza essa informação. Caso contrário o servidor indica à aplicação que o sistema não conhece o local onde o utilizador se encontra (Figura 5.25), permitindo que este corrija a localização e contribua para o crescimento do mapa de rádio.

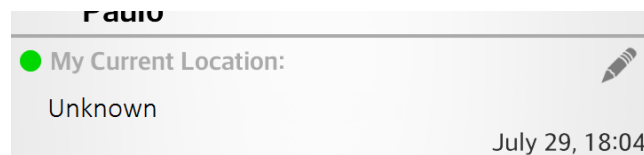


Figura 5.25 - Localização do utilizador desconhecida

- **Obtenção da localização dos amigos**

Após completar a localização do utilizador, o servidor obtém a lista de amigos do utilizador que estejam a partilhar a sua localização (atributo “locationSharing” a “1”). Esta informação é mantida na tabela de utilizadores da base de dados. Após obter a lista de amigos, o servidor obtém a última localização conhecida do amigo e verifica, como no processo anterior, se a informação está atualizada. Se estiver atualizada o servidor constrói um objeto com a informação do local e acrescenta-o à lista de amigos. Senão estiver atualizada, o servidor envia o endereço MAC da interface Wi-Fi do dispositivo móvel ao motor de posicionamento a fim de obter a localização do dispositivo. Se a resposta do motor de posicionamento não for positiva, isto é, não conseguiu localizar o dispositivo, o servidor assume a última localização conhecida, ainda que não seja a mais recente.

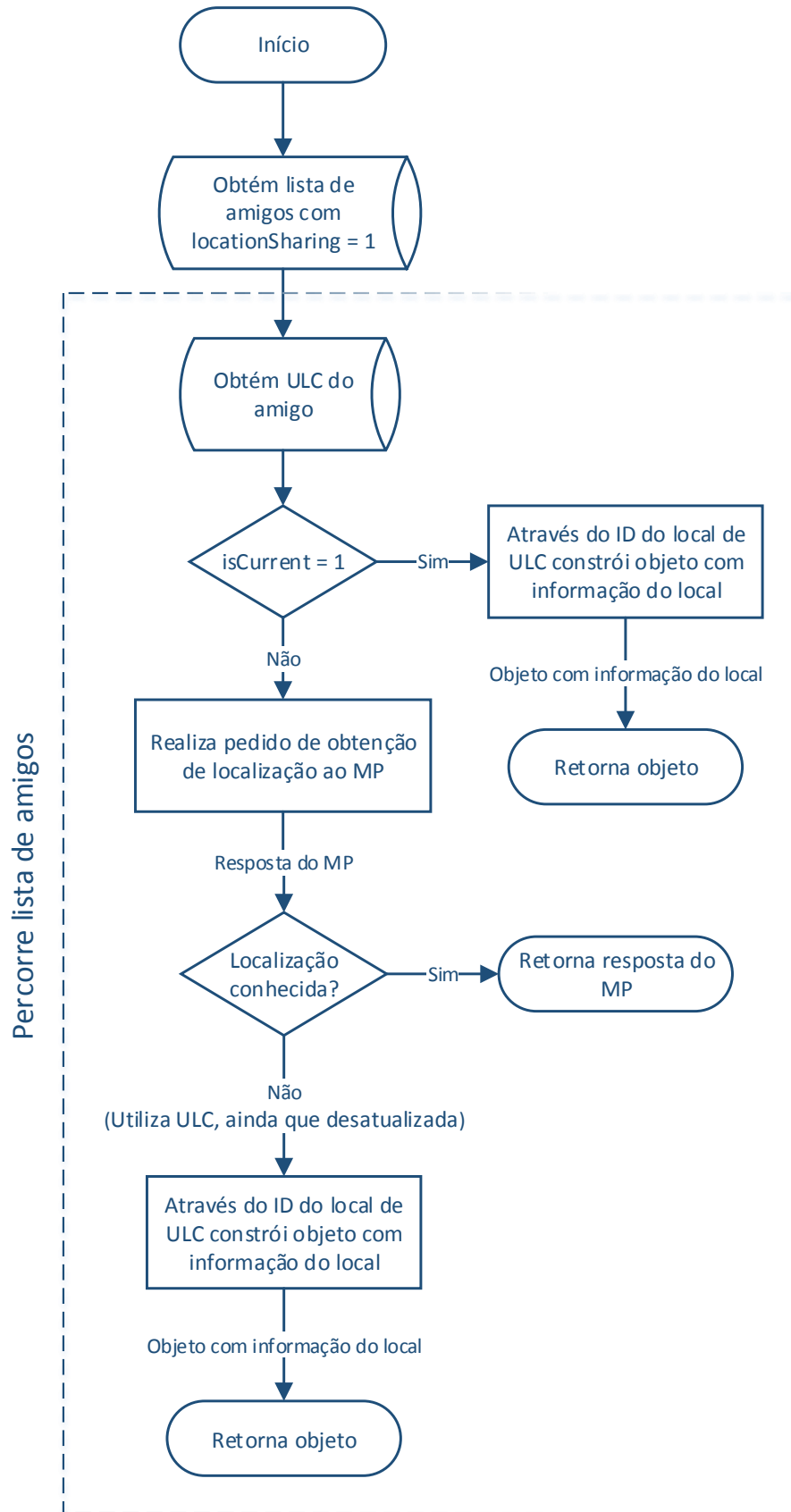


Figura 5.26 - Obtenção da localização dos amigos

5.2.4 Evolução do módulo de construção do mapa de rádio

Inicialmente o módulo de construção do mapa de rádio foi pensado para funcionar como encaminhador das *fingerprints* enviadas pelos dispositivos móveis para o motor de posicionamento. Armazenava apenas as *fingerprints* anotadas e posteriormente, através de um processo de integração, adicionava-as ao motor de posicionamento já desenvolvido.

Numa fase inicial não existia integração com o motor de posicionamento, ou seja, o motor de posicionamento não recebia os locais que iam sendo adicionados pelos utilizadores. Isto acontecia por não haver um processo que avaliasse a credibilidade das *fingerprints* anotadas.

Nesta altura a localização dos utilizadores era estimada sempre que a aplicação pedia ao servidor uma lista atualizada com a localização dos amigos do respetivo utilizador, abordagem que posteriormente foi alterada. Nesta altura, era utilizado um critério temporal, isto é, se a última *fingerprint* anotada tivesse sido submetida dentro de um intervalo de tempo pré-definido (ex.: 5 minutos), o sistema considerava que o utilizador estava no mesmo local. Como esta abordagem não era muito eficiente, uma vez que se o utilizador tivesse no mesmo local mais que o intervalo de tempo pré-definido o sistema já não sabia onde este se encontrava, surgiu a necessidade de adicionar algumas funcionalidades ao módulo de posicionamento.

- **Processo de localização dos utilizadores (versão 2)**

Para melhorar o processo de posicionamento foi necessário armazenar na base de dados, além das *fingerprints* anotadas, a última *fingerprint* enviada por cada utilizador. Nesta altura, a localização dos utilizadores ainda era estimada sempre que uma aplicação pedia a localização dos utilizadores. O objetivo da função passava por verificar se o utilizador ainda se encontrava no mesmo local através da comparação temporal ou de distância entre a última *fingerprint* (UF) enviada e a última *fingerprint* anotada (FA) pelo utilizador.

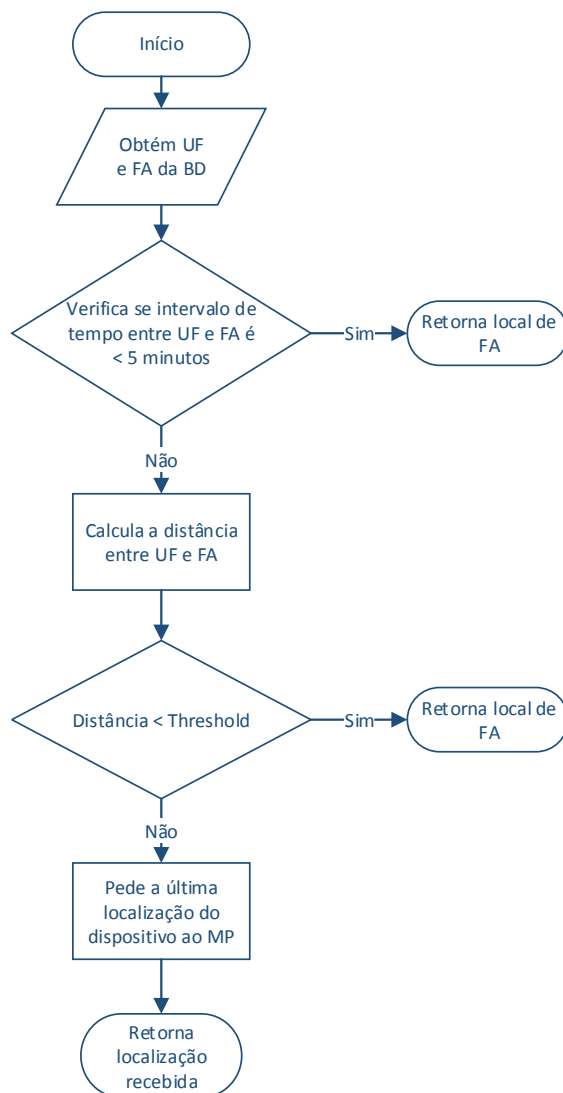


Figura 5.27 - Processo de localização dos utilizadores (versão 2)

A introdução da segunda versão do processo de localização dos utilizadores melhorou em parte o funcionamento do sistema e resolveu o problema inicial. Após algum tempo de utilização da aplicação com este processo implementado, surgiram novas dúvidas sobre o método adotado, uma vez que se o utilizador anotasse um local e depois se deslocasse para outro e o anotasse, quando voltasse ao anterior o sistema já não ia conhecer o local, porque este só comparava com a última *fingerprint* anotada pelo utilizador. Chegou-se à conclusão que era necessário atribuir maior responsabilidade e processamento ao servidor, tendo em conta que, para além do problema referido, o sistema estava a separar os locais por utilizador, isto é, não estavam a ser aproveitadas as *fingerprints* anotadas por outros utilizadores.

- **Processo de localização dos utilizadores (versão 3)**

Como referido em cima, houve a necessidade de melhorar o processo de localização, contrariamente ao idealizado inicialmente, e optou-se por dar maior preponderância ao servidor relativamente à localização dos utilizadores. Desta feita, o momento da localização dos utilizadores passou a ser realizado sempre que o servidor recebe uma *fingerprint*. Esta passou a ser comparada com as *fingerprints* anotadas por todos os utilizadores, contrariamente à versão anterior que só comparava com as anotadas pelo utilizador da *fingerprint* recebida. Estas alterações obrigaram a modificar a base de dados a tabela que antes guardava a última *fingerprint* dos utilizadores foi removida. Adicionou-se a tabela “lastKnownLocation” para guardar a última localização conhecida dos utilizadores, anotada ou estimada. Nesta altura, por motivos estatísticos e para possíveis benefícios futuros, foi decidido criar um histórico de *fingerprints* (“fingerprintsHistory”), armazenando todas as *fingerprints* enviadas pelos dispositivos móveis. Esta abordagem pode ser útil, ainda que algumas *fingerprints* não sejam totalmente fiáveis, para a implementação do processo de integração entre o motor de posicionamento do servidor e o motor de posicionamento anteriormente desenvolvido.

Nesta versão, inicialmente é verificado se a data de *fingerprint* recebida (F) é mais antiga que a da última localização conhecida (ULC) do respetivo utilizador. Ainda que mais antiga, é submetida ao processo de comparação de *fingerprints* de modo a tentar armazenar a *fingerprint* recebida, na tabela de histórico de *fingerprints*, com um local associado. Se for mais recente que a outra e se o intervalo de tempo entre as duas for inferior a 1 minuto, o sistema assume que o utilizador se mantém no local indicado pela última localização conhecida. Se esta última condição não se verificar, a *fingerprint* é submetida ao processo de comparação de *fingerprints* que a compara com as *fingerprints* anotadas e tenta obter uma localização. Se o processo for bem sucedido, a tabela “lastKnownLocation” é atualizado com o local estimado e o atributo “isCurrent” é colocado a “1”, indicando que é a última localização conhecido do utilizador. Caso contrário, a tabela “lastKnownLocation” não é atualizada com um novo local e o atributo “isCurrent” é colocado a “0”. Quando a *fingerprint* recebida não é correspondida, isto é, quando o servidor não consegue associar a *fingerprint* recebida a um local existente na base de dados, é enviada para o motor de posicionamento para

quando a aplicação móvel pedir a localização de utilizadores ao servidor, este perguntar ao motor de posicionamento se conhece a localização dos utilizadores que têm o atributo “isCurrent” a “0”.

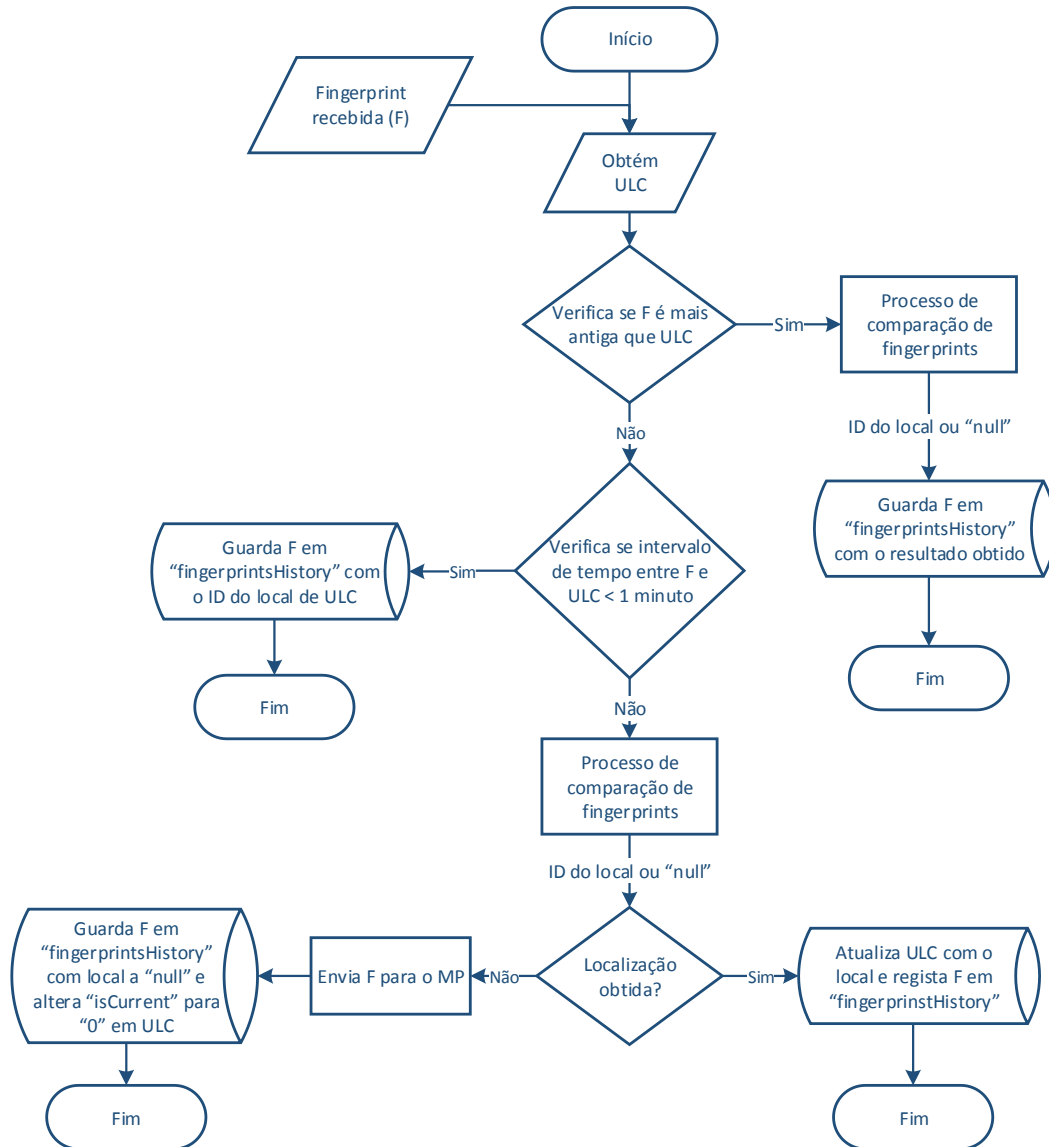


Figura 5.28 - Processo de localização dos utilizadores adotado (versão 3)

A Figura 5.28 explica, esquematicamente, o processo de localização dos utilizadores, explicado atrás, e usado atualmente. O processo de comparação de *fingerprints* presente neste último algoritmo foi submetido a testes com diferentes parâmetros para avaliar o seu desempenho, eficiência e tempo de execução. Este processo é explicado e discutido na secção seguinte.

5.2.5 Comparação de fingerprints

Antes de desenvolver o processo de comparação de *fingerprints* foi necessário compreender a função utilizada para calcular a distância entre *fingerprints*, estudar as *fingerprints* recolhidas em diversos cenários e aferir o nível de semelhança entre elas. O trabalho presente não inclui um estudo comparativo sobre as diferentes funções de cálculo de distância entre *fingerprints* por este já ter sido realizado anteriormente em [60], onde se concluiu que a função *Manhattan Distance* obteve melhor desempenho relativamente às restantes para o contexto em questão.

Mesmo que recolhidas consecutivamente no mesmo local, as *fingerprints* apresentam diferenças na sua constituição. Vários fatores influenciam a recolha de *fingerprints*, desde a presença e o movimento de pessoas, a existência de outros dispositivos e a forma como o dispositivo de recolha está posicionado. Estes e outros fatores provocam alterações nos sinais rádio e, por conseguinte, variações nos valores do RSSI.

Tabela 5.3 - Comparação entre *fingerprints* recolhidas no mesmo local por dispositivos diferentes

Ponto de acesso (endereço MAC)	RSSI	
	LG Nexus 5	LG 4X HD
00:0E:83:CB:F0:A0	-71	-67
00:0E:83:CB:F7:40	-82	-91
00:0E:83:CC:05:30	-84	N.D.
00:0E:D7:CD:E5:10	-78	-86
00:0E:D7:CD:E7:10	-82	N.D.
00:0E:D7:CD:E7:90	-76	-71
00:0E:D7:CD:E7:B0	N.D.	-86
00:12:43:8A:7F:C0	-81	-80
00:15:F2:EB:24:6E	-90	N.D.
00:16:B9:52:AF:3D	-86	N.D.
00:18:84:81:2C:25	-63	-52
88:43:E1:13:6D:70	N.D.	-80
88:43:E1:13:6F:20	-88	-85
88:43:E1:13:6F:B0	-86	N.D.
C0:62:6B:D4:55:80	-89	N.D.
E0:CB:4E:A8:6C:01	-84	-86

A Tabela 5.3 mostra a variação entre duas *fingerprints* recolhidas no mesmo local por diferentes dispositivos. É visível que ambas possuem pontos de acesso comuns, o que é normal por se tratar do mesmo local. Ainda assim, existem pontos de acesso que são apenas detetados, com fraco sinal (valores próximos ou inferiores a -90), por um dos dispositivos. Os pontos de acesso não detetados por um dos dispositivos estão assinalados como “N.D.” (não detetado) no campo RSSI. O principal fator que levou a esta discrepância de valores entre *fingerprints* deve-se ao facto de terem sido utilizados dispositivos com diferentes características a nível da placa de rede Wi-Fi. Ainda assim, se as *fingerprints* fossem recolhidas pelo mesmo dispositivo haveria uma variância, não tão substancial, da potência de sinal dos pontos de acesso detetados.

5.2.5.1 Distância entre fingerprints

De forma a saber se duas *fingerprints* pertencem ao mesmo local, foi utilizada uma função de cálculo de distância entre duas *fingerprints* (equação 1). A distância é definida pela soma da diferença absoluta entre os valores de RSSI de cada ponto de acesso, dividida pelo número total de pontos de acesso (n).

$$D(FP^A, FP^B) = \frac{\sum_{i=1}^n |RSSI_i^A - RSSI_i^B|}{n} \quad (1)$$

Como comprovado pela Tabela 5.3 a lista de pontos de acesso contida nas duas *fingerprints* pode não ser a mesma, nestes casos o sistema assume um valor pré-definido para o valor de RSSI em falta (defRSSI) e a distância é calculada pela diferença entre o valor pré-definido e o valor de RSSI presente na outra *fingerprint*.

5.2.5.2 Nível de similaridade entre fingerprints

Foi ainda necessário definir um valor *threshold* para saber se duas *fingerprints* pertencem ao mesmo local, isto é, se a distância calculada através da função utilizada for inferior ao valor pré-definido *threshold*, o sistema conclui que as *fingerprints* correspondem ao mesmo local, caso contrário o sistema assume que as *fingerprints* são de locais distintos. O valor pré-definido *threshold* define a distância limite entre duas *fingerprints* para ser considerado que representam ou não o mesmo local.

O valor ótimo para o *threshold* é o que conduz o sistema a identificar positivamente todas as *fingerprints* que correspondem a um mesmo local e que também faça o sistema diferenciar locais distintos. Se o *threshold* for inferior ao valor ótimo causa um número excessivo de falsos negativos, o contrário provoca um número excessivo de falsos positivos.

5.2.5.3 Definição dos parâmetros defRSSI e threshold

Foi desenvolvido um estudo para avaliar a similaridade entre *fingerprints* e encontrar o valor ótimo para os parâmetros defRSSI e *threshold*. Após realizar o estudo o valor obtido para defRSSI foi -109 e para o *threshold* foi 17,5. A experiência e os resultados obtidos encontram-se na secção 5.4.1.

5.2.5.4 Processo de comparação de fingerprints

Como referido na secção 5.2.4, foi necessário implementar um mecanismo de comparação de *fingerprints* que permitisse associar um local, previamente anotado, à *fingerprint* recolhida. O processo implementado baseia-se na comparação da *fingerprint* recebida com as *fingerprints* anotadas através do cálculo da distância entre *fingerprints* e posterior processo de ranking, que ordena a lista de *fingerprints* por distância, verifica se a primeira está abaixo do *threshold* pré-definido e retorna o identificador do local caso a verificação seja bem sucedida. Dado que comparar uma *fingerprint* com todas as *fingerprints* anotadas é um processo demorado e algo exigente, inicialmente não mas a longo prazo sim, foi idealizado um método que seleciona um conjunto de *fingerprints* anotadas para comparação. As *fingerprints* anotadas são escolhidas por ordem de data, isto é, são escolhidas preferencialmente as mais recentes e são apenas selecionadas um número pré-definido de *fingerprints* por local.

Esta solução foi adotada essencialmente por dois motivos: o motor de posicionamento implementado do lado do servidor ser apenas um motor que auxilia o motor de posicionamento anteriormente desenvolvido; e permite combater, em parte, o problema da alteração das redes Wi-Fi, problema referido no capítulo 3, através da utilização das *fingerprints* mais recentes. A primeira versão implementada utilizava apenas uma *fingerprint* por local, depois foi introduzida uma nova, com cinco

fingerprints por local (Figura 5.29). O aumento do número de *fingerprints* por local não influenciou significativamente o tempo de execução e melhorou a eficácia. Esta comparação encontra-se na secção 5.4.2.

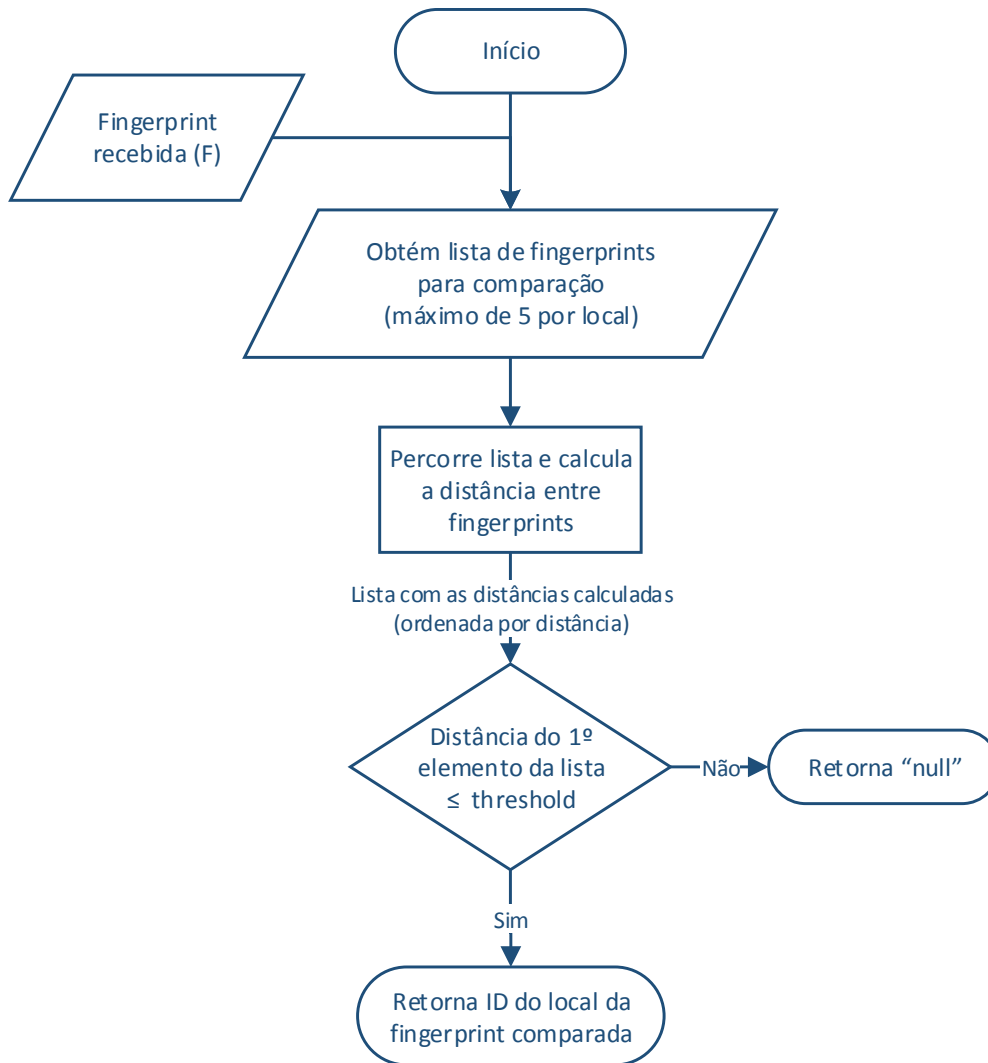
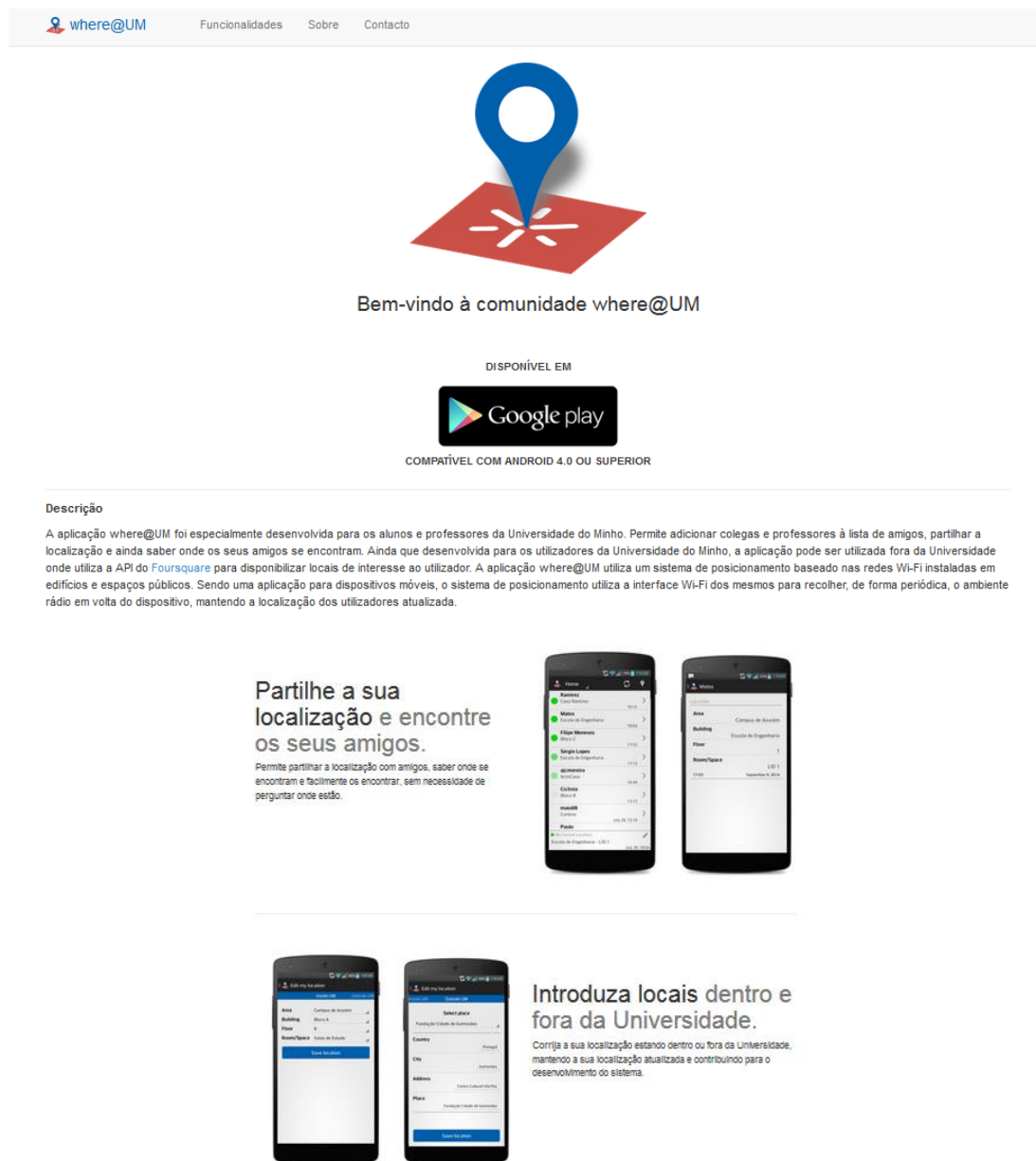


Figura 5.29 - Processo de comparação de fingerprints

5.3 Website

Como referido na secção 4.5, foi desenvolvido um *website* para a aplicação where@UM. O *website* concebido contém apenas uma página e é meramente informativo (Figura 5.30), descreve a finalidade da aplicação assim como as funcionalidades da mesma. A página foi construída a partir de um modelo já existente [61], foram apenas realizadas algumas alterações necessárias para se adequar à aplicação.



where@UM Funcionalidades Sobre Contacto

Bem-vindo à comunidade where@UM

DISPONÍVEL EM

Google play

COMPATÍVEL COM ANDROID 4.0 OU SUPERIOR

Descrição

A aplicação where@UM foi especialmente desenvolvida para os alunos e professores da Universidade do Minho. Permite adicionar colegas e professores à lista de amigos, partilhar a localização e ainda saber onde os seus amigos se encontram. Ainda que desenvolvida para os utilizadores da Universidade do Minho, a aplicação pode ser utilizada fora da Universidade onde utiliza a API do Foursquare para disponibilizar locais de interesse ao utilizador. A aplicação where@UM utiliza um sistema de posicionamento baseado nas redes Wi-Fi instaladas em edifícios e espaços públicos. Sendo uma aplicação para dispositivos móveis, o sistema de posicionamento utiliza a interface Wi-Fi dos mesmos para recolher, de forma periódica, o ambiente rádio em volta do dispositivo, mantendo a localização dos utilizadores atualizada.

Partilhe a sua localização e encontre os seus amigos.

Permite partilhar a localização com amigos, saber onde se encontram e facilmente os encontrar, sem necessidade de perguntar onde estão.

Introduza locais dentro e fora da Universidade.

Corrija a sua localização estando dentro ou fora da Universidade, mantendo a sua localização atualizada e contribuindo para o desenvolvimento do sistema.

Figura 5.30 - Website where@UM, retirado de [62]

5.4 Testes

Nesta secção são apresentados alguns estudos e testes singulares realizados durante o desenvolvimento deste trabalho. O primeiro estudo descrito serviu para aferir o nível de semelhança entre *fingerprints* e encontrar o valor ótimo para dois parâmetros usados no processo de comparação de *fingerprints*. O segundo serviu para avaliar e comparar o desempenho (tempo de execução e eficácia) do processo de comparação de *fingerprints* adotado, para um diferente número de *fingerprints* anotadas a comparar. Ainda no segundo, foi possível comparar as *fingerprints* recolhidas por diferentes dispositivos móveis. Por fim, o consumo de energia por parte da aplicação móvel na sua máxima utilização foi avaliado e os resultados foram apresentados.

5.4.1 Semelhança entre fingerprints

A introdução de processos de comparação de *fingerprints* conduziu à realização de um estudo para aferir o nível de semelhança entre *fingerprints* recolhidas em diversos locais e a definição dos parâmetros *defRSSI* e *threshold* necessários para o processo de comparação.

O desenvolvimento deste estudo incitou à elaboração de um artigo [63]. O artigo desenvolvido, além de descrever a solução apresentada no presente trabalho, aborda essencialmente o estudo sobre semelhança entre *fingerprints* recolhidas em diferentes cenários. O artigo foi publicado e apresentado no INForum 2014, simpósio de informática realizado no Porto, a 5 de Setembro de 2014.

A experiência contou com a recolha de 128 *fingerprints* em salas e espaços previamente seleccionados. Foram escolhidas 16 salas/espacos de 5 edifícios diferentes do Campus de Azurém da Universidade do Minho. A seleção de salas e espaços foi pensada de forma a abranger diferentes cenários do mundo real, foram escolhidas salas afastadas (isoladas), salas próximas/encostadas e salas com diferentes dimensões como é mostrado pela Figura 5.31. Em cada sala/espaco foram recolhidas 8 *fingerprints*, 4 em cada um de dois locais diferentes (geralmente em dois cantos das salas/espacos). Estes dois locais (marcados a laranja na Figura 5.31) foram preferidos para avaliar a diferença do ambiente rádio dos vários cenários. Importante referir que a recolha foi

realizada por um único dispositivo móvel, não sendo aqui considerado a diferenciação entre *fingerprints* recolhidas por diferentes dispositivos.

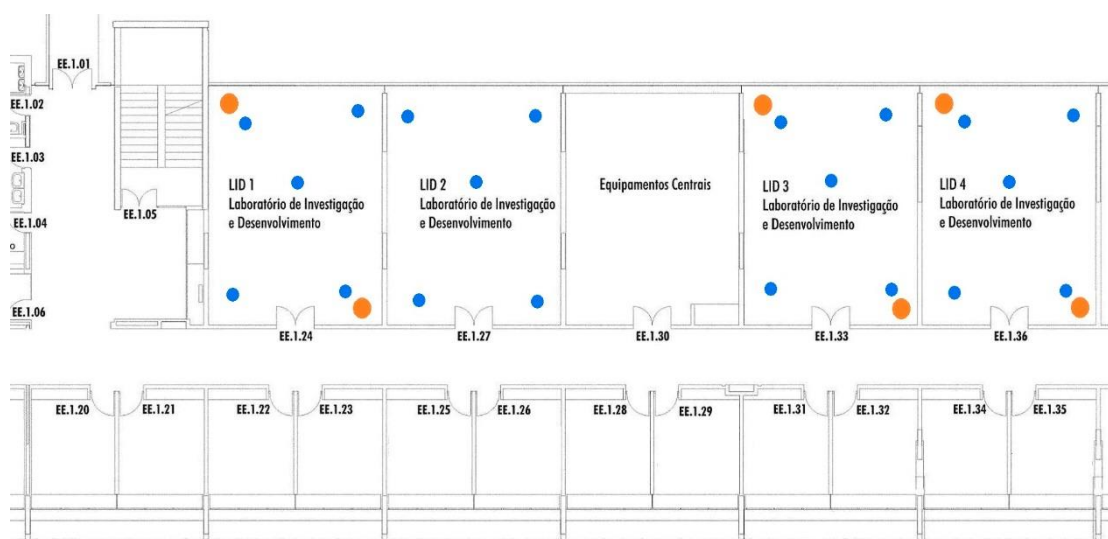


Figura 5.31 – Exemplo de 4 das salas e locais escolhidos para a recolha de fingerprints

As 128 *fingerprints* recolhidas foram processadas e avaliadas com intuito de aferir o nível de semelhança entre *fingerprints*, para poder concluir se duas *fingerprints* correspondem ou não ao mesmo local, e encontrar os valores ótimos para os parâmetros *defRSSI* e *threshold*. O número de falsos positivos (FP) e falsos negativos (FN) foram calculados em função do *threshold*. A Figura 5.32 mostra os falsos positivos e falsos negativos calculados para dois valores do parâmetro *defRSSI* (foram testados outros valores, não representados na Figura 5.32).

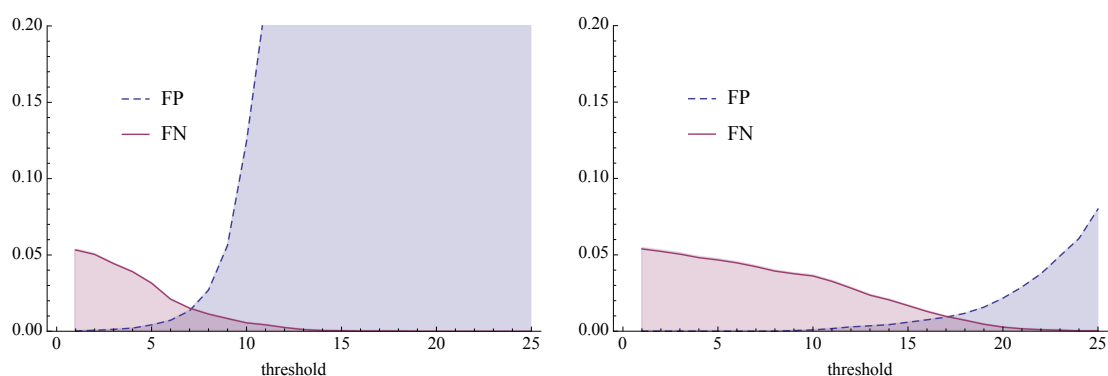


Figura 5.32 - Gráficos estatísticos dos valores de falsos positivos e falsos negativos em função do *threshold*: *defRSSI* = -110 (esquerda) e *defRSSI* = -90 (direita)

Os resultados apresentados pela Figura 5.32 mostram que a percentagem de falsos positivos é sempre inferior a 6%, para qualquer valor de *threshold* experimentado. Mostra ainda que o valor de defRSSI tem um impacto significativo, principalmente no número de falsos positivos. A partir dos gráficos da Figura 5.32 é possível concluir que o valor ótimo para o *threshold* deve ser escolhido de forma a minimizar o número de falsos positivos e falsos negativos.

O critério considerado para a escolha do *threshold* passa por calcular a soma dos falsos positivos e falsos negativos e definir um *threshold* que minimizasse esta métrica. Contudo, o *threshold* escolhido podia não significar o melhor resultado em termos de experiência de utilizador. Na perspetiva do utilizador, é preferível que o sistema indique que o utilizador se encontra num determinado local (próximo do utilizador) do que indique que este se encontra num local desconhecido. Em contrapartida, se o utilizador se encontra a uma distância considerável do local que o sistema indica que está, o utilizador pode questionar a credibilidade do sistema.

Ainda assim, considerou-se como valor ótimo para o *threshold* aquele que minimiza a soma dos falsos positivos e falsos negativos (Fmin). Contudo, como ilustrado pela Figura 5.32, este valor depende do valor tomado pelo parâmetro defRSSI.

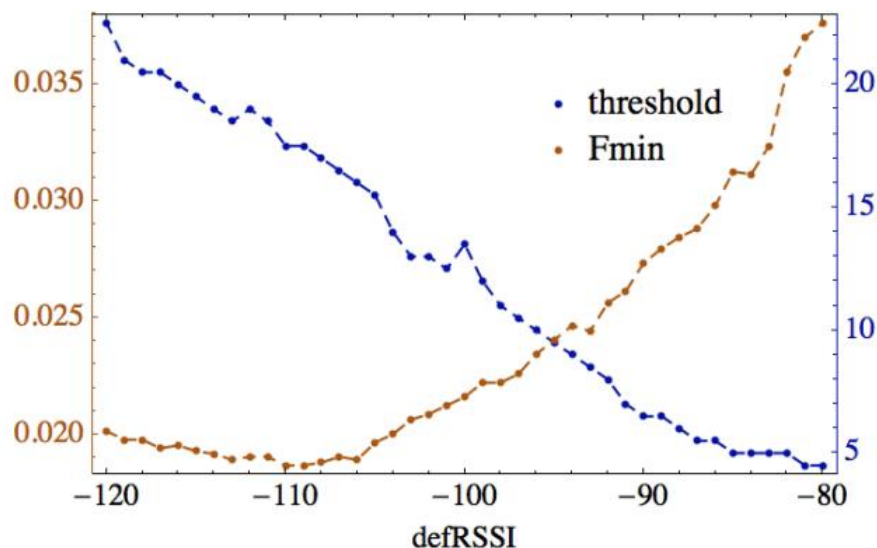


Figura 5.33 - Impacto do defRSSI no valor ótimo do *threshold*

A Figura 5.33 mostra o valor ótimo do *threshold* e o valor Fmin correspondente em função do parâmetro defRSSI. O valor mínimo de Fmin é obtido quando defRSSI toma

o valor de, aproximadamente, -109, correspondendo a um valor ótimo de *threshold* próximo de 17.5. Com estes parâmetros a percentagem de falsos positivos e falsos negativos é inferior a 1.9%, que para o contexto do sistema é um valor aceitável.

A Figura 5.34 apresenta o gráfico que ilustra o desempenho geral da experiência, também conhecido por matriz de confusão, onde é possível observar a correspondência das *fingerprints* relativamente aos locais escolhidos. Os pontos verdes representam as *fingerprints*, consideradas pelo sistema, que pertencem ao mesmo local. Os pontos laranjas representam os casos em que o sistema considerou que as *fingerprints* pertenciam a locais diferentes. Um resultado ideal seria se todas as distâncias calculadas entre as 8 *fingerprints* recolhidas dentro de cada sala/espço estivessem abaixo do *threshold* (a matriz de confusão ficaria com quadrados perfeitamente preenchidos a verde ao longo da diagonal), e todas as outras estivessem acima do *threshold*.

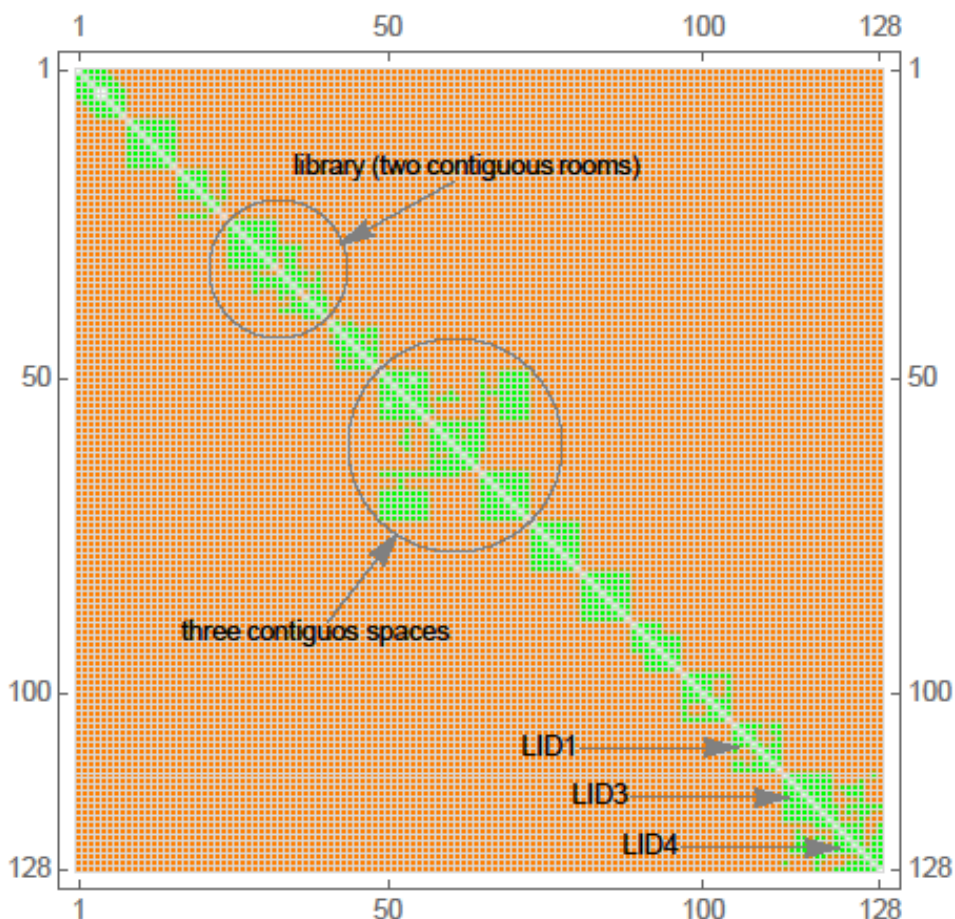


Figura 5.34 - Desempenho geral da experiência, retirado de [63]

Os resultados representados pela Figura 5.34 mostram uma baixa percentagem de falsos negativos (grande parte dos quadrados estão preenchidos a verde), mas apresentam alguns casos de falsos positivos. As zonas identificadas na Figura 5.34 indicam os casos mais conflituosos detetados durante a experiência. As duas salas da biblioteca, marcadas com um círculo e identificadas como “library”, estão separadas por uma parede fina e as *fingerprints* recolhidas próximas a esta parede, foram consideradas que pertenciam, erradamente, ao mesmo local. Uma das salas da biblioteca possui uma área consideravelmente superior à área dos restantes locais escolhidos, este fator provocou uma distância entre as *fingerprints* recolhidas nos diferentes locais da mesma sala superior ao *threshold* pré-definido, visível pela imperfeição do quadrado inferior do círculo identificado como “library”. A zona central da figura representa, essencialmente, uma sala com janelas em vidro e um corredor que dá para a mesma sala, aumentando o número de falsos positivos. Por último, as salas (LID 1, LID 3 e LID 4), cuja disposição é observável pela Figura 5.32, apresentam quadrados quase perfeitos, à exceção do LID4 que apresenta alguns falsos positivos e falsos negativos.

A evolução do módulo de construção do mapa de rádio, com a introdução de um processo de comparação mais complexo, permitiu minimizar a percentagem de falsos positivos e de falsos negativos, dado que as *fingerprints* são comparadas com um número maior de *fingerprints* melhorando a eficiência da correspondência de *fingerprints* aos locais existentes.

5.4.2 Análise do processo de comparação de fingerprints

A análise realizada sobre o processo de comparação de *fingerprints* serviu para avaliar o desempenho deste processo para diferentes números de *fingerprints* a comparar. As *fingerprints* utilizadas para comparação foram anotadas durante a utilização, normal, da aplicação. A primeira versão (V1) deste processo utiliza apenas uma *fingerprint* anotada por local: a mais recente. A outra versão (V2) avaliada utiliza no máximo as cinco *fingerprints*, anotadas por local, mais recentes. Aproveitou-se ainda para avaliar o desempenho do processo de comparação de *fingerprints* para diferentes dispositivos móveis.

A experiência envolveu 6 locais, dos quais 5 pertencem à Escola de Engenharia da Universidade do Minho e o outro a uma habitação com várias divisórias. Em cada local foram recolhidas 5 *fingerprints*, das quais 4 foram recolhidas nos cantos das salas e uma na zona central da mesma (marcados a azul na Figura 5.31), à exceção do último local em que as *fingerprints* foram recolhidas nas diversas divisórias. A recolha das *fingerprints* foi realizada por 3 dispositivos diferentes, sendo que no total foram recolhidas 90 *fingerprints* e posteriormente processadas pelas duas versões (representadas por V1 e V2).

Os resultados obtidos foram processados e traduzidos em tabelas, cada um apresenta os resultados referentes a cada dispositivo. As tabelas contêm o desempenho, a eficácia (que reflete o número de *fingerprints* que foram corretamente correspondidas sobre o número total de *fingerprints* recolhidas) e o tempo de execução médio das duas versões do processo de comparação. De forma a avaliar a influência do número de *fingerprints* anotadas por local no desempenho do processo de comparação de *fingerprints*, são indicados, para cada local, o número de *fingerprints* anotadas (FA) e ainda se o dispositivo em questão possui alguma *fingerprint* anotada.

Tabela 5.4 - Resultados referentes ao dispositivo Asus Memo Pad HD7

LOCAL	Nº de FA	V1		V2		Anotou alguma Fingerprint
		Eficácia	T. Execução (ms)	Eficácia	T. Execução (ms)	
LID 1	37	0/5	0.128	3/5	0.206	✗
LID 2	2	5/5	0.132	5/5	0.212	✓
LID 3	5	5/5	0.129	5/5	0.208	✓
LID 4	1	0/5	0.133	0/5	0.214	✗
BAR	13	0/5	0.130	1/5	0.210	✗
CASA	25	2/5	0.123	3/5	0.199	✗

Asus Memo Pad HD7

Tabela 5.5 - Resultados referentes ao dispositivo LG 4X HD

LOCAL	Nº de FA	V1		V2		Anotou alguma Fingerprint
		Eficácia	T. Execução (ms)	Eficácia	T. Execução (ms)	
LID 1	37	4/5	0.123	4/5	0.198	✓
LID 2	2	1/5	0.130	0/5	0.210	✗
LID 3	5	1/5	0.132	5/5	0.212	✓
LID 4	1	5/5	0.127	5/5	0.205	✓
BAR	13	2/5	0.122	5/5	0.197	✓
CASA	25	2/5	0.130	5/5	0.246	✓
LG 4X HD						

Tabela 5.6 - Resultados referentes ao dispositivo Acer Liquid E2

LOCAL	Nº de FA	V1		V2		Anotou alguma Fingerprint
		Eficácia	T. Execução (ms)	Eficácia	T. Execução (ms)	
LID 1	37	1/5	0.125	2/5	0.204	✗
LID 2	2	0/5	0.129	0/5	0.209	✗
LID 3	5	0/5	0.129	5/5	0.209	✗
LID 4	1	3/5	0.129	0/5	0.208	✗
BAR	13	3/5	0.123	5/5	0.198	✗
CASA	25	5/5	0.124	5/5	0.200	✗
Acer Liquid E2						

Estes resultados mostram uma tendência em que se obtêm melhores resultados quando utiliza um número maior de *fingerprints* por local. O tempo de execução em comparação com a versão 1 não é significativo, sendo que é possível optar por um número maior de *fingerprints* a comparar. Contudo, o aumento do número de *fingerprints* não resolve o principal problema encontrado. As *fingerprints* recolhidas por dispositivos diferentes são bastante distintas, influenciando, pela negativa, a eficiência do processo de comparação de *fingerprints*. Observando as tabelas é possível apurar que a eficácia melhora quando um dispositivo móvel contribuiu com uma ou mais *fingerprints* anotadas. Comparando os resultados relativamente ao local LID 4, em que possui apenas uma *fingerprint* anotada pelo dispositivo LG 4X HD, observa-se

que este dispositivo possui uma eficácia de 100% em oposição ao dispositivo Asus Memo Pad HD7, o que não significa que a distância calculada entre as *fingerprints* recolhidas por este último dispositivo e a *fingerprint* anotada pelo outro não se encontre abaixo do nível de *threshold* pré-definido. Pode acontecer, como foi o caso, que o processo de comparação de *fingerprints* calculou uma *fingerprint* mais próxima que a *fingerprint* correta, uma vez que se tratam de locais próximos (ver LID 3 e LID 4 na Figura 5.31) e o dispositivo Asus Memo Pad HD7 possui uma *fingerprint* anotada no local LID 3. Este exemplo comprova a disparidade existente nas *fingerprints* recolhidas por diferentes dispositivos, nos mesmos locais, que degrada a eficiência do processo de comparação de *fingerprints* e aumenta o número de falsos positivos. As percentagens, de falsos positivos e falsos negativos, obtidas para as diferentes versões podem ser consultadas na Figura 5.35.

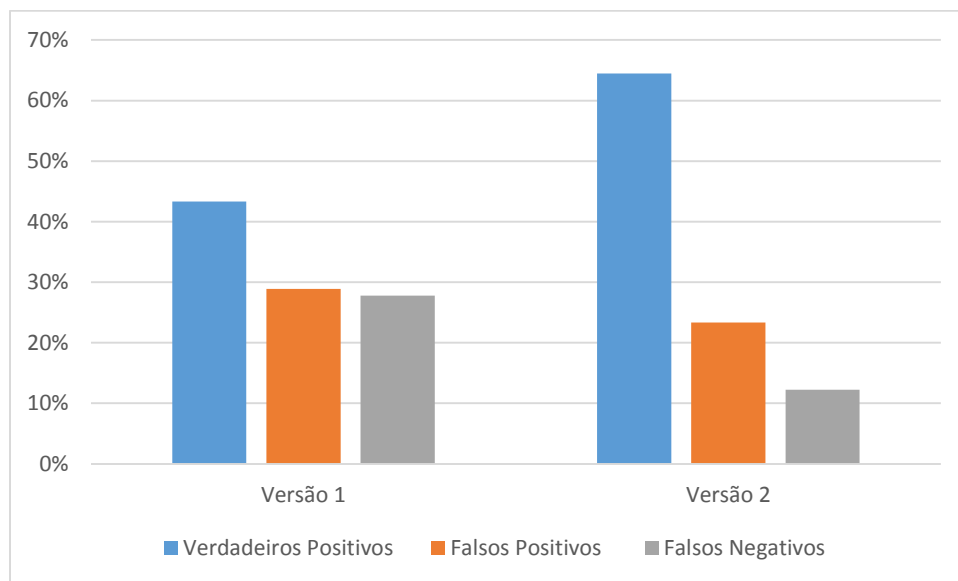


Figura 5.35 - Gráfico comparativo entre as duas versões do processo de comparação

Ainda que a percentagem de verdadeiros positivos seja significativamente superior para a versão 2, a percentagem de falsos positivos e falsos negativos, mas principalmente de falsos positivos, é considerável. Como já referido, esta situação deve-se ao facto de os dispositivos deterem *fingerprints* anotadas em locais próximos aos locais comparados, influenciando o processo de comparação de *fingerprints*. Este problema deve ser analisado e minimizado em futuras versões deste sistema.

5.4.3 Consumo de energia da aplicação móvel

Como referido na secção de requisitos não funcionais da aplicação móvel e sendo um tópico sempre discutido no contexto de desenvolvimento de aplicações para dispositivos móveis, o consumo de energia da aplicação móvel foi analisado.

Hoje em dia o consumo de energia por parte dos dispositivos móveis é elevado, sendo mesmo necessário carregar o dispositivo, pelo menos, a cada dois dias, isto considerando um funcionamento normal com a interface Wi-Fi ligada. Posto isto, a aplicação móvel foi testada durante o período diário, aproximadamente 12h, com uma recolha periódica do ambiente rádio Wi-Fi a cada 5 minutos, o que corresponde ao menor intervalo de tempo permitido pela aplicação. Os dados sobre o consumo de energia foram recolhidos através da aplicação GSam Battery Monitor [64].

Durante este período, a aplicação móvel, incluindo o serviço de recolha de dados, despertou 200 vezes, com um total aproximado de 9 minutos desperto e recolheu 141 *fingerprints*. A imagem da esquerda da Figura 5.36 apresenta os valores sobre o nível de utilização da aplicação móvel durante o período de teste. A imagem da direita da mesma figura apresenta a percentagem do consumo energia por parte da aplicação móvel. O valor da percentagem obtida é insignificante, o que permite concluir que é possível aumentar a frequência de recolha do ambiente rádio através da interface Wi-Fi do dispositivo móvel, principalmente quando este se encontra em movimento.

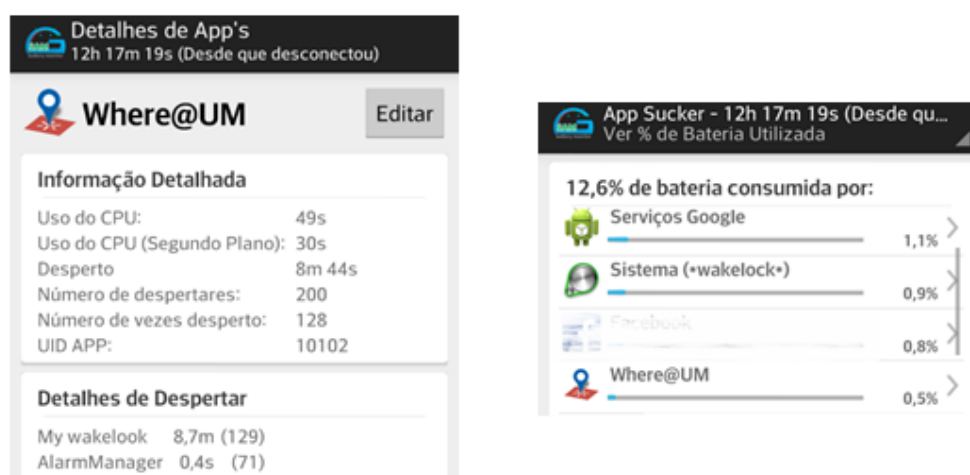


Figura 5.36 - Informação detalhada sobre a utilização da aplicação e a percentagem do consumo de bateria, captura de ecrã recolhida a partir da aplicação GSam Battery Monitor

6. Avaliação

O período de avaliação da aplicação móvel foi curto, principalmente após colocar a aplicação na loja oficial da Google onde houve maior adesão por parte dos utilizadores à aplicação. Antes desta data, a aplicação estava distribuída por alguns utilizadores com intuito de manter o ambiente de testes controlado. No entanto, e apesar da limitação temporal do período de testes e análise dos resultados, são apresentados os resultados obtidos ao longo do uso da aplicação móvel, assim como a experiência de utilização e opinião dos utilizadores relativamente à mesma.

6.1 Experiência no mundo real

A primeira versão da aplicação móvel implementada, ainda num estado precoce, foi distribuída aos orientadores deste trabalho no início de Junho de 2014. A partir dessa data o sistema de posicionamento sofreu um conjunto de testes e alterações, naturais para o momento em questão, até se obter uma versão estável e funcional para poder alargar o número de utilizadores. Em meados de Julho, do mesmo ano, a aplicação foi distribuída a novas pessoas com o mesmo intuito, aumentar o número de utilizadores mas mantendo o ambiente de testes controlado. Esta fase foi importante para avaliar o desempenho do sistema para um número maior, ainda que pequeno, de utilizadores com diferentes áreas de utilização e diferentes dispositivos móveis. O comportamento do sistema foi positivo, assim como o feedback recebido por parte dos utilizadores. Em Setembro, e coincidindo premeditadamente com o início do ano letivo para a maioria dos alunos da Universidade do Minho, a aplicação foi colocada na loja oficial da Google e disponibilizada para todos os alunos da universidade através do envio de emails personalizados com o link da página informativa da aplicação. O envio de emails personalizados a toda a universidade foi fundamental para a divulgação da aplicação. Muitos utilizadores quiseram experimentar e instalaram a aplicação, contribuindo com a introdução de novos locais, opinião e sugestões de melhoramento sobre a aplicação. A aplicação chegou ao conhecimento da Rádio Universitária do Minho (RUM) [65], ao

jornal nacional JN [66] e da redação de comunicação da Universidade do Minho (ComUM) [67] que contactaram o autor do trabalho e solicitaram uma reportagem sobre o projeto em questão.

6.1.1 Adesão e motivação dos utilizadores

Como esperado, motivar as pessoas a experimentar e utilizar uma aplicação em que o conceito base é localização, ainda que podendo manter a sua localização privada, foi um verdadeiro desafio. Além da questão da privacidade, persiste o facto de não ter sido implementada nenhuma funcionalidade, para além da localização, que fidelizasse os utilizadores à aplicação. Acredita-se que a solução para este problema pode passar pela introdução de novas funcionalidades de cariz social e de formas para recompensar os utilizadores por usarem a aplicação.

6.1.2 Experiência dos utilizadores

De modo a conhecer a opinião dos utilizadores sobre a aplicação móvel, foi-lhes enviado um email a solicitar uma pequena reflexão relativamente à aplicação, assim como sugestões sobre possíveis funcionalidades que poderiam enriquecer e tornar a aplicação imprescindível no dia a dia dos utilizadores, alargando a área de utilização e aumentando o número de utilizadores da comunidade where@UM.

Além de solicitar a opinião sobre a aplicação, o email enviado solicitava que os utilizadores escolhessem três funcionalidades, das enumeradas no email (legenda da Figura 6.1), que gostassem de ver implementadas numa versão futura da aplicação where@UM. Vinte e três utilizadores responderam ao email e o resultado deste pequeno inquérito é ilustrado pelo gráfico circular da Figura 6.1. O ideal seria implementar todas estas funcionalidades, mas os utilizadores responderam e deram preferência essencialmente à integração com o Facebook, permite facilitar o processo de adicionar amigos registados ou não no sistema, introdução de perfil e histórico dos utilizadores, que segundo os utilizadores é uma funcionalidade básica no contexto de aplicações sociais, e a introdução de um mapa com a localização dos amigos. Este último é considerado, em termos de implementação, o mais complicado de conseguir, uma vez que a utilização integral de mapas geográficos como o Google Maps ou o

OpenStreenMap numa aplicação direcionada para a localização em ambientes fechados não é a melhor solução. Estes mapas podem ser úteis na localização dos utilizadores quando estes se encontram fora da Universidade do Minho, em que não se exige um nível de precisão muito elevado. No entanto, seria necessário criar uma planta completa com todos os edifícios e andares da Universidade do Minho, e encontrar uma solução para associar os locais da planta aos locais adicionados no sistema.

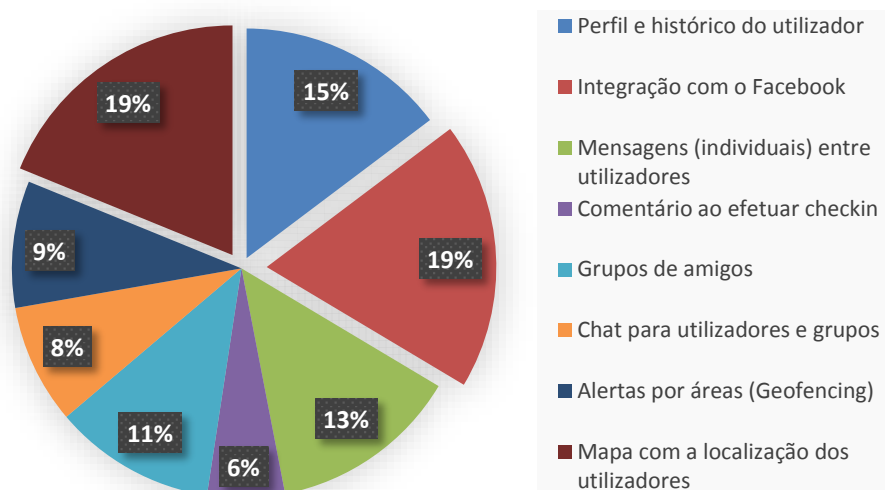


Figura 6.1 - Gráfico com a preferência dos utilizadores relativamente a funcionalidades futuras

De um modo geral a opinião dos utilizadores relativamente à aplicação foi positiva, alguns consideraram a aplicação inovadora e bem conseguida. A interface do utilizador também foi comentada por ser simples e de fácil compreensão. Um dos aspetos que segundo os utilizadores merece uma especial atenção é o processo de convidar amigos, e uma possível integração com o Facebook ou mesmo com os contactos telefónicos dos utilizadores, pode ser a solução para este problema.

6.2 Análise dos resultados

Apesar da dificuldade em encontrar pessoas dispostas a experimentar e utilizar a aplicação, o sistema conta neste momento com um total de 131 utilizadores registados no sistema, sendo que alguns não possuem um papel ativo na utilização da aplicação. O gráfico da Figura 6.2 não coincide com o número de utilizadores registados, uma vez que o gráfico inclui apenas as instalações realizadas através do Google Play, não

considerando as instalações efetuadas antes da colocação da aplicação where@UM no Google Play (dia 17 de Setembro de 2014). Através do gráfico é possível observar dois aumentos significativos no número de instalações. Entre os dias 5 e 7 de Outubro houve um crescente número de instalações devido ao email de divulgação enviado a todos os alunos da Universidade do Minho. Observou-se outro aumento entre os dias 19 e 20 de Outubro, motivado pela publicação da reportagem sobre a aplicação no jornal nacional JN.

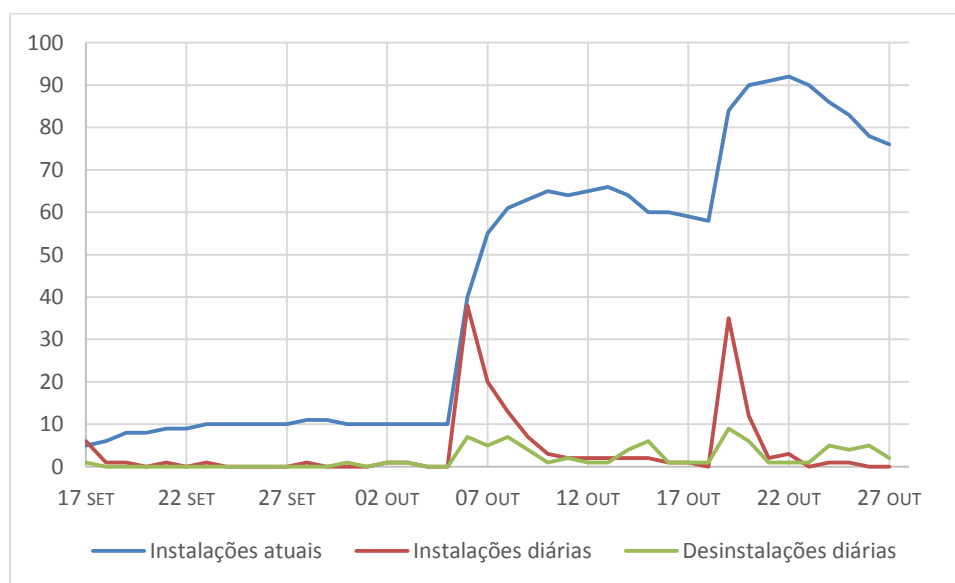


Figura 6.2 - Estatísticas de utilização da aplicação where@UM

Durante o período de testes e experimentação no mundo real foram recolhidas mais de 230 mil *fingerprints* pelos dispositivos móveis dos utilizadores, e aproximadamente 85 mil das *fingerprints* recolhidas possuem correspondência a um determinado local. Os utilizadores da aplicação contribuíram com 407 anotações (muitas foram registadas no mesmo local durante a fase inicial de testes), em 199 locais diferentes, dos quais 79 pertencem à Universidade do Minho e os restantes 129 estão distribuídos por diferentes cidades e alguns em diferentes países. O menor número de locais anotados dentro da Universidade do Minho não significa que os utilizadores não usem a aplicação na universidade, este valor explica-se por se tratarem de locais que os utilizadores, com um papel mais ativo no sistema, frequentam habitualmente. Os locais registados fora da universidade são, em norma, as casas dos utilizadores e amigos, cafés e restaurantes, centros comerciais e ainda salas de outras universidades.

7. Conclusões

O principal desafio deste trabalho de dissertação visava o desenvolvimento de um sistema de posicionamento, baseado em redes Wi-Fi, que através de uma solução colaborativa permitisse construir o mapa de rádio de edifícios de grande escala. Os sistemas de posicionamento mais comuns, que tiram partido da infraestrutura Wi-Fi dos edifícios e locais urbanos, possuem uma fase inicial de calibração demorada, às vezes impraticável em edifícios de grandes dimensões, e facilmente desatualizada devido às alterações das redes Wi-Fi.

Recorrendo ao estado da arte dos sistemas de posicionamento colaborativos baseados na tecnologia Wi-Fi considerou-se o desenvolvimento de uma aplicação de cariz social que permitisse envolver os utilizadores, inconscientemente, na construção do mapa de rádio. Optou-se por desenvolver uma aplicação móvel para os utilizadores da Universidade do Minho com o intuito de construir o mapa de rádio dos edifícios das várias áreas da Universidade do Minho e avaliar o comportamento tanto do sistema de posicionamento como o envolvimento dos utilizadores no processo de construção do mapa de rádio.

Os estudos realizados ao longo da dissertação, similaridade entre *fingerprints* recolhidas em diferentes cenários e a avaliação dos processos de localização implementados, contribuíram para o desenvolvimento do sistema de posicionamento e permitiram concluir que é possível utilizar as redes Wi-Fi como recurso para obter a localização dos dispositivos móveis e ao mesmo tempo a localização dos utilizadores.

Uma vez concluído este trabalho, verifica-se que uma parte significativa dos desafios iniciais foram abordados e que foram encontradas soluções adequadas para os resolver. No entanto, ficou por concluir a integração completa com o motor de posicionamento. Por outro lado, foram identificados novos desafios para os quais existem já propostas de solução.

7.1 Trabalho futuro

O trabalho desenvolvido pode ser melhorado em diversos aspetos, tendo em conta que ficaram por implementar algumas funcionalidades inicialmente definidas devido à limitação temporal para a realização da dissertação. Ficou essencialmente por implementar a integração do sistema desenvolvido com o motor de posicionamento anteriormente concebido, a implementação de funcionalidades para o melhoramento da aplicação de forma a fidelizar os utilizadores à aplicação *where@UM* e o módulo de deteção de movimento que permitiria enriquecer o processo de recolha de *fingerprints*. Ficou ainda por assegurar a confidencialidade e integridade dos dados durante a comunicação entre o servidor e a aplicação móvel, sendo relevante implementar o protocolo SSL para conferir segurança no momento da comunicação pela Internet.

É importante que no futuro seja realizado um estudo mais exaustivo sobre a comparação de *fingerprints* recolhidas por dispositivos móveis diferentes e seja encontrada uma solução para resolver este problema identificado durante a realização deste trabalho. A solução para este problema poderá passar por implementar uma das soluções já propostas por outros autores.

A atual página informativa pode evoluir para um *website* com outro tipo de fim, além de informativa pode-se tornar numa área em que os utilizadores possam consultar o seu perfil, visualizar o seu histórico, gerir a sua conta, conversar e acompanhar os amigos através do *website*. O registo de utilizadores poderia passar a ser realizado também pelo *website*.

7.1.1 Integração

Um dos desafios colocados na fase inicial deste trabalho foi integrar o motor de posicionamento no sistema de posicionamento colaborativo proposto. Durante o desenvolvimento deste trabalho foram feitas algumas alterações a nível de desenho do sistema e colocou-se em segundo plano a implementação da integração entre ambos. Assim sendo, o processo de integração passa a ser prioritário para um próximo trabalho que prossiga o trabalho presente.

O processo de integração permitiria completar o sistema global. Neste momento estão a ser utilizadas, maioritariamente, *fingerprints* cuja credibilidade não é conhecida, temporariamente guardadas no mapa de rádio do servidor. Após a implementação deste processo deverá ser possível analisar as *fingerprints* armazenadas e avaliar a sua credibilidade, construindo o mapa de rádio do motor de posicionamento.

Após experienciar o comportamento dos utilizadores e do sistema implementado ao longo do trabalho, é possível mencionar algumas ideias de como avaliar se as *fingerprints*, armazenadas no servidor, são ou não credíveis. Se quando um local possui apenas uma *fingerprint* anotada e entretanto outros utilizadores permaneceram no mesmo local e não corrigiram a localização, o sistema pode assumir que as *fingerprints* são credíveis, uma vez que nenhum utilizador corrigiu a localização. Se existirem várias *fingerprints* anotadas para um determinado local, é importante ter em conta quem anotou os locais, se foi apenas um utilizador ou vários utilizadores. O sistema pode calcular a distância entre as *fingerprints* anotadas sobre o mesmo local e caso a distância obtida esteja abaixo do *threshold* pré-definido, pode assumir-se que representam o mesmo local.

7.1.2 Aplicação móvel

Como já referido, a aplicação móvel pode evoluir em vários sentidos. Relativamente à componente social podem ser adicionadas algumas funcionalidades de forma a atrair os utilizadores para uma maior utilização da aplicação. O inquérito realizado aos utilizadores da aplicação móvel sobre funcionalidades gostariam de ver introduzidas numa futura versão da aplicação where@UM, secção 6.1.2, ajudou a definir as funcionalidades a implementar num trabalho futuro. Contudo, é importante continuar a explorar novas funcionalidades e possíveis integrações com outras aplicações de cariz social como o Facebook e o Foursquare. Relativamente ao processo de recolha de *fingerprints* pode ser melhorado com a introdução de um módulo de deteção de movimento. Isso permitiria detetar se o utilizador se encontra parado num determinado local ou se está em movimento. Caso se encontre parado, é possível obter um maior número de *fingerprints* credíveis. A introdução deste módulo deve ser considerado prioritário caso o projeto venha a ter continuação.

Referências

- [1] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger e James Collins, “Global Positioning System – Theory and Practice”, 1993.
- [2] “The Ekahau Real-Time Location System”. Disponível em: <http://www.ekahau.com>, visitado em 9 de Outubro 2014.
- [3] Paramvir Bahl e Venkata N. Padmanabhan, “Radar: An In-Building RF-based User Location and Tracking System”. Publicado em IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, 26 a 30 de Março de 2000.
- [4] Anthony LaMarca, “Place Lab: Device Positioning Using Radio Beacons in the Wild”. Publicado em Proceedings of the third International Conference on Pervasive Computing 2005, Munique, Alemanha, 8 a 13 de Março de 2005.
- [5] Anthony LaMarca. “Self-Mapping in 802.11 Location Systems”. Publicado em Proceedings of Ubicomp 2005, Tóquio, Japão, 11 a 14 de Setembro de 2005.
- [6] Rui Monteiro e Adriano Moreira, “Mapeamento Automático de Redes WiFi com base em Assinaturas Rádio”. Publicado em CRC’2010 – 10ª Conferência sobre Redes de Computadores, pp. 167-172, Universidade do Minho, Braga, Portugal, 2010.
- [7] Joel Carvalho, ”Localização de Dispositivos Móveis em Redes Wi-Fi”, Mestrado em Informática, Universidade de Trás-os-Montes e Alto Douro Vila Real, Portugal, 2007.
- [8] “Crux – Indoor Location & Dynamic Advertising”. Disponível em: <http://www.southmobile.com.ar/es/7/Crux-Indoor-Location-Dynamic-Advertising>, visitado em 18 de Outubro de 2013.
- [9] Roberto Casas, David Cuartielles, Álvaro Marco, Héctor J. Gracia e Jorge L. Falcó, “Hidden Issues in Deploying an Indoor Location System”, IEEE Pervasive

- Computing, vol. 6, n° 2, 2007, pp. 62-69.
- [10] Chunhan Lee, Yushin Chang, Gunhong Park, Jaeheon Ryu, Seung-Gweon Jeong, Seokhyun Park, Jae W. Park, Hee C. Lee, Keum-Shik Hong e Man H. Lee, “Indoor Positioning System Based on Incident Angles of Infrared Emitters”. Publicado em Industrial Electronics Society (IECON 2004), 30th Annual Conference of IEEE, vol. 3, pp. 2218-2222, 2 a 6 de Novembro de 2014.
- [11] “The Bat Ultrasonic Location System”. Disponível em: <http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/>, visitado em 9 de Outubro de 2014.
- [12] Hubert Piontek, Matthias Seyffer e Jörg Kaiser, “Improving the accuracy of ultrasound-based localization systems”. Publicado em Personal and Ubiquitous Computing, vol. 11 Issue 6, pp. 439-449, Agosto de 2007.
- [13] “Zebra Technologies”. Disponível em: <http://www.zebra.com/>, visitado em 9 de Outubro de 2014.
- [14] “Radianse – Intelligent Real time Location Solutions”. Disponível em: <http://www.radianse.com/>, visitado em 9 de Outubro de 2014.
- [15] P. Prasithsangaree, P. Krishnamurthy e P. K. Chrysanthis, “On indoor position location with wireless LANs”. Publicado em The 31th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 720-724, 15 a 18 de Setembro de 2002.
- [16] S. Kawakubo, A. Chansavang, S. Tanaka, T. Iwasaki, K. Sasaki, T. Hirota, H. Hosaka e H. Ando, “Wireless Network System for Indoor Human Positioning”. Publicado em 1st International Symposium on Wireless Pervasive Computing, pp. 1-6, 16 a 18 de Janeiro de 2006.
- [17] J. C. F. Michel, M. Christmann, M. Fiegert e M. Vossiek, “Multisensor Based Indoor Vehicle Localization System for Production and Logistic”. Publicado em IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 553-558, Alemanha, Setembro de 2006.
- [18] S. J. Ingram, D. Harmer e M. Quinlan, “UltraWideBand Indoor Positioning Systems and their Use in Emergencies”. Publicado em Position Location and Navigation Symposium (PLANS 2004), pp. 706-715, 26 a 29 de Abril de 2004.

-
- [19] F. H. Raab, E. B. Blood, T. O. Steiner e H. R. Jones, “Magnetic Position and Orientation Tracking System”. Publicado em IEEE Transactions on Aerospace and Electronic Systems, vol. AES-15, Issue 5, pp. 709-718, 20 de Fevereiro de 2007.
- [20] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale e S. Shafer, ”Multi-Camera Multi-Person Tracking for EasyLiving”. Publicado em Third IEEE International Workshop on Visual Surveillance, pp. 3-10, Dublin, 2000.
- [21] V. Paelke e C. Reimann, “Vision-Based Interaction – A First Glance at Playing MR Games in the Real-World Around Us”. Publicado em 3rd International Conference on Pervasive Computing (PERVASIVE 2005), 2005.
- [22] Yanying Gu, Anthony Lo e Ignas Niemegeers, “A Survey of Indoor Positioning Systems for Wireless Personal Networks”. Publicado em IEEE Communications Surveys & Tutorials, vol. 11, nº1, 2009.
- [23] Rainer Mautz, “Indoor Positioning Technologies”. Institute of Geodesy and Photogrammetry, Zurique, Suíça, Fevereiro de 2012.
- [24] Hui Liu, Houshang Darabi, Pat Banerjee e Jing Liu, “Survey of Wireless Indoor Positioning”. Publicado em IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 37, nº6, Novembro de 2007.
- [25] Mark Paciga e Hanan Lutfyya, “Herecast: An Open Infrastructure for Location-Based Services Using WiFi”. Publicado em IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob’2005), pp. 21-28, Montreal, Quebec, Canadá, 22 a 24 de Agosto de 2005.
- [26] Jonathan Ledlie, Jung-geun Park, Dorothy Curtis, André Cavalcante, Leonardo Camara, Afonso Costa e Robson Vieira, ”Molé: a Scalable, User-Generated WiFi Positioning Engine”. Publicado em International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-10, Guimarães, Portugal, 21 a 23 de Setembro de 2011.
- [27] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth teller e Jonathan Ledlie, “Growing an organic indoor location

- system”. Publicado em The 8th International Conference on Mobile Systems, Applications and Services (MobiSys '10), pp. 271-284, EUA, 2010.
- [28] P. Bolliger, “RedPin: Adaptive, Zero-Configuration Indoor Localization”. Em LoCA, Alemanha, 2008.
- [29] “infsoft – Indoor Positioning, Indoor Navigation”, <http://www.infsoft.com/>, visitado em 6 de Agosto de 2014.
- [30] “Navizon – Wifi location and cell tower triangulation for wireless devices”. Disponível em: <https://www.navizon.com/>, visitado em 6 de Agosto de 2014.
- [31] “Meridian – Indoor GPS, Indoor Positioning”. Disponível em: <http://www.meridianapps.com/>, visitado em 6 de Agosto de 2014.
- [32] “Insiteo”. Disponível em: <http://www.insiteo.com/joomla/index.php/en/>, visitado em 6 de Agosto de 2014.
- [33] “AlterGeo, global location technology provider”. Disponível em: <http://platform.altergeo.ru/>, visitado em 6 de Agosto de 2014.
- [34] “Combain – positioning solutions”. Disponível em: <http://www.combain.com/>, visitado em 6 de Agosto de 2014.
- [35] “Foursquare”. Disponível em: <https://pt.foursquare.com/>, visitado em 25 de Novembro de 2013.
- [36] “Mapbox”. Disponível em: <https://www.mapbox.com/>, visitado em 25 de Novembro de 2013.
- [37] “OpenStreetMap”. Disponível em: <http://www.openstreetmap.org/>, visitado em 25 de Novembro de 2013.
- [38] “Find my Friends!”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator>, visitado em 25 de Novembro de 2013.
- [39] “Life360 – Family Locator, Messaging and More”, aplicação iOS. Disponível em: <https://itunes.apple.com/us/app/life360-family-tracker/id384830320?mt=8>, visitado em 25 de Novembro de 2013.
- [40] “Crux Indoor Location”, aplicação Android. Disponível em: <https://>

-
- play.google.com/store/apps/details?id=com.sm.crux, visitado em 25 de Novembro de 2013.
- [41] “Vodafone Radar”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=com.vodafone.radar>, visitado em 25 de Novembro de 2013.
- [42] “Vodafone Radar”, Aplicação iOS. Disponível em: <https://itunes.apple.com/pt/app/vodafone-radar/id456617830?mt=8>, visitado em 25 de Novembro de 2013.
- [43] “Find My Friends”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=com.nliteapps.findmyfriends>, visitado em 25 de Novembro de 2013.
- [44] “Facebook Phone Tracker”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=br.com.webadger.peopletracker>, visitado em 25 de Novembro de 2013.
- [45] “Friend Mapper”, aplicação iOS. Disponível em: <https://itunes.apple.com/gb/app/friend-mapper/id295806081>, visitado em 25 de Novembro de 2013.
- [46] “Friend Pilot”, aplicação Windows Phone. Disponível em: <http://www.windowsphone.com/en-us/store/app/friend-pilot-beta/f5d17343-5288-4c24-b38a-6a250c8f10e5>, visitado em 25 de Novembro de 2013.
- [47] “App2Find”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=com.innovattic.app2find>, visitado em 25 de Novembro de 2013.
- [48] “App2Find”, aplicação iOS. Disponível em: <https://itunes.apple.com/pt/app/app2find-localizador-amigos/id519065152?mt=8>, visitado em 25 de Novembro de 2013.
- [49] “People Tracker”, aplicação iOS. Disponível em: <https://itunes.apple.com/us/app/people-tracker-pro/id539205975>, visitado em 25 de Novembro de 2013.
- [50] Yungeun Kim, Yohan Chon, Hojung Cha, MyungIn Ji e Sangjoon Park, “Scalable and Consistent Radio Map Management Scheme for Participatory Sensing-based Wi-Fi fingerprinting”.
- [51] Bernhard Krach e Patrick Robertson, “Cascaded Estimation Architecture for Integration of Foot-Mounted Inertial Sensors”. Publicado em IEEE/ION Position,

-
- Location and Navigation Symposium, pp. 112-119, Califórnia, EUA, 5 a 8 de Maio de 2008.
- [52] Patrick Robertson, Michael Angermann e Bernhard Krach, “Simultaneous Localization and Mapping for Pedestrians using only FootMounted Inertial Sensors”. Publicado em Ubicomp 2009, Flórida, EUA, 30 de Setembro a 3 de Outubro de 2009.
- [53] Hyojeong Shin e Hojung Cha, “Wi-Fi Fingerprint-Based Topological Map Building for Indoor User Tracking”. Publicado em IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pp. 105-113, Macau, China, 23 a 25 de Agosto de 2010.
- [54] Cristiano G. Pendão, “Recolha de Dados de Movimento em Dispositivos Móveis Pessoais”, Universidade do Minho, Guimarães, Portugal, Novembro de 2012.
- [55] Filip Truta, “After Purging Adult Apps, Apple Now Pulls Wi-Fi Finders”. Disponível em: <http://news.softpedia.com/news/After-Purging-Adult-Apps-Apple-Now-Pulls-Wi-Fi-Finders-136545.shtml>, visitado em 14 de Agosto de 2014.
- [56] Katie Marsal, “Apple removes Wi-Fi scanners, ‘minimum functionality iPhone apps’”. Disponível em: http://appleinsider.com/articles/10/03/04/apple_removes_wi-fi_scanners_minimum_functionality_iphone_apps, visitado em 14 de Agosto de 2014.
- [57] Tutorials Point, “Android Tutorial – Simply Easy Learning”. Disponível em: http://www.tutorialspoint.com/android/android_tutorial.pdf, visitado em 20 de Março de 2014.
- [58] “Issue 57707: ‘onLocationChanged’ is not called (LocationManager using Network Provider)”. Disponível em: <https://code.google.com/p/android/issues/detail?id=57707>, visitado em 6 de Junho de 2014.
- [59] Documentação da API do Foursquare. Disponível em: <https://developer.foursquare.com/docs/>
- [60] Nelson Marques, Filipe Meneses e Adriano Moreira, “Combining similarity functions and majority rules for multi-building, multi-floor, WiFi Positioning”.

-
- Publicado em Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-9, Sydney, Austrália, 13 a 15 de Novembro de 2012.
- [61] “Start Bootstrap – One Page Wonder Template”. Disponível em: <http://startbootstrap.com/template-overviews/one-page-wonder/>, visitado em 15 de Setembro de 2014.
- [62] “where@UM”. Disponível em: <http://urano.dsi.uminho.pt/whereatum/>, visitado em 14 de Outubro de 2014.
- [63] Diogo Matos, Adriano Moreira e Filipe Meneses, “Wi-Fi fingerprint similarity in collaborative radio maps for indoor positioning”. Publicado em INForum 2014 – Simpósio de Informática, Porto, Portugal, 4 e 5 de Setembro de 2014.
- [64] “GSam Battery Monitor”, aplicação Android. Disponível em: <https://play.google.com/store/apps/details?id=com.gsamlabs.bbm>, visitado em 2 de Outubro de 2014.
- [65] RUM – Rádio Universitária do Minho. Disponível em: <http://www.rum.pt/>, visitado em 19 de Outubro de 2014.
- [66] Delfim Machado, “‘Big Brother’ na universidade”, Jornal de Noticias (JN), pp. 25, 19 de Outubro de 2014.
- [67] Andreia Cunha e Florbela Caetano, “Aplicação ‘Where@UM’ sabe onde estás”, ComUM Online em Destaque, Sociedade, 23 de Outubro de 2014. Disponível em: <http://www.comumonline.com/?p=4039>, visitado em 24 de Outubro de 2014.

Anexo A

Tabela com informação detalhada sobre as aplicações móveis estudadas para diferentes plataformas móveis.

Tabela A.1 - Resumo das aplicações estudadas

Aplicação	Sistema Operativo	Funcionalidades	Técnicas de localização	Testado
Foursquare	Android iOS Windows Phone	<ul style="list-style-type: none"> · Check-in · Acompanhar check-ins de amigos · Procurar PI de determinados locais · Avaliar/comentar locais visitados · Integração com o Facebook e Twitter 	GPS Via API	✓
Find My Friends!	Android iOS	<ul style="list-style-type: none"> · Check-in · Acompanhar amigos através de um mapa · Pedir que estes partilhem a sua posição · Gerir grupos · Chat de amigos/grupos · Configurar partilha da posição · Áreas de notificação 	GPS Via API	✓
Crux Indoor Location	Android	<ul style="list-style-type: none"> · Acompanhar movimento, em tempo real, dentro de um edifício · Conhecer a planta do edifício e os seus PI com informações relevantes 	Wi-Fi	✗
Vodafone Radar	Android iOS	<ul style="list-style-type: none"> · Check-in via Facebook ou Foursquare · Acompanhar check-ins de amigos · Visualizar tabela de amigos associada à distância e tempo dos últimos check-ins · Procurar PI de determinados locais e ver os comentários de quem já os frequentou 	GPS Via API	✓

		<ul style="list-style-type: none"> · Avaliar/comentar locais visitados · Alertas de notificação (com custo de serviço associado) · Integração com o Facebook e Foursquare 		
Find My Friends	Android	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa · Observar no mapa mensagens partilhadas pelos amigos · Controlar quando e com quem se pretende partilhar a localização · Configurar partilha da posição · Integração com o Facebook, Google+ e LinkedIn 	GPS Via API	✓
Facebook Phone Tracker	Android	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa · Conhecer novas pessoas que se encontrem nas proximidades · Observar no mapa mensagens partilhadas pelos amigos · Integração com o Facebook 	GPS Via API	✓
Friend Mapper	iOS	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa (limitado a 23 pessoas em simultâneo) · Configurar partilha da posição · Observar última atualização dos amigos 	GPS Via API	✗
Friend Pilot	Windows Phone	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa básico · Configurar partilha de posição · Integração com o Facebook e Microsoft Account 	GPS Via API	✗
App2Find	Android iOS	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa · Escolher quem o segue e quem pretende seguir · Encontrar amigos através da câmara de Realidade Aumentada · Personalizar perfil dos amigos · Configurar partilha da posição 	GPS Via API	✓
People Tracker	iOS	<ul style="list-style-type: none"> · Acompanhar amigos, em tempo real, através de um mapa 	GPS Via API	✗

-
- Visualizar histórico dos movimentos de amigos
 - Enviar E-Mails com a posição exata
 - Gerir grupos
 - Áreas de notificação
-

Anexo B

Especificação dos serviços da interface do módulo de suporte à aplicação móvel.

- Registo do utilizador

Método HTTP POST

URL	{server}/users
Parâmetros	<ul style="list-style-type: none">▪ name Nome do novo utilizador▪ nickname Alcunha do novo utilizador▪ email <i>Email</i> do novo utilizador▪ password Senha do novo utilizador
Resposta	Em caso de sucesso é enviado um objeto com os dados de perfil do utilizador, em caso de insucesso é enviada uma mensagem de erro.

- Autenticação do utilizador

Método HTTP POST

URL	{server}/users/self/login
Parâmetros	<ul style="list-style-type: none">▪ email <i>Email</i> do utilizador▪ password Senha do utilizador
Resposta	Em caso de sucesso é enviado um objeto com os dados de perfil do utilizador, em caso de insucesso é enviada uma mensagem de erro.

- Cancelar conta

Método HTTP DELETE

URL	{server}/users/{idUser}	
Parâmetros	▪ idUser	Identificador do utilizador
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Convidar utilizador para pertencer ao grupo de amigos

Método HTTP POST

URL	{server}/users/{idUser}/request	
Parâmetros	▪ idUser	Identificador do utilizador
	▪ email	Email do utilizador que pretende convidar
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Cancelar pedido de amizade

Método HTTP DELETE

URL	{server}/users/{idUser}/cancel	
Parâmetros	▪ idUser	Identificador do utilizador que pretende cancelar o pedido de amizade
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Aceitar convite de um utilizador

Método HTTP POST

URL	{server}/users/{idUser}/approve
Parâmetros	▪ idUser Identificador do utilizador que fez o pedido
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

- Recusar convite de um utilizador

Método HTTP POST

URL	{server}/users/{idUser}/deny
Parâmetros	▪ idUser Identificador do utilizador que fez o pedido
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

- Remover utilizador do grupo de amigos

Método HTTP POST

URL	{server}/users/{idUser}/unfriend
Parâmetros	▪ idUser Identificador do utilizador que pretende remover do grupo de amigos
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

- Editar senha do utilizador

Método HTTP PUT

URL	{server}/users/{idUser}/password	
Parâmetros	▪ idUser	Identificador do utilizador
	▪ password	Nova senha do utilizador
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Editar alcunha do utilizador

Método HTTP PUT

URL	{server}/users/{idUser}/nickname	
Parâmetros	▪ idUser	Identificador do utilizador
	▪ nickname	Nova alcunha do utilizador
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Editar permissão de partilha de posição

Método HTTP PUT

URL	{server}/users/{idUser}/permission	
Parâmetros	▪ idUser	Identificador do utilizador que pretende alterar a permissão de partilha da posição
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Pedido de reenvio de uma password gerada automaticamente

Método HTTP POST

URL	{server}/users/{idUser}/password
Parâmetros	▪ idUser Identificador do utilizador
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

- Obter informação relativa aos amigos do utilizador

Método HTTP GET

URL	{server}/users/{idUser}/friends
Parâmetros	▪ idUser Identificador do utilizador
Resposta	Em caso de sucesso é enviado um objeto com a informação pessoal relativa aos amigos, em caso de insucesso é enviada uma mensagem de erro.

- Obter localização do utilizador e seus amigos

Método HTTP GET

URL	{server}/users/{idUser}/checkins
Parâmetros	▪ idUser Identificador do utilizador
Resposta	Em caso de sucesso é enviado um objeto com a informação da última localização conhecida do próprio e dos amigos, em caso de insucesso é enviada uma mensagem de erro.

Anexo C

Especificação dos serviços da interface do módulo de construção do mapa de rádio.

- Envio de *fingerprints*

Método HTTP POST

URL	{server}/fingerprints
	<ul style="list-style-type: none">▪ idDevice Endereço MAC da interface Wi-Fi do <i>smartphone</i> do utilizador▪ $MAC_1 = RSSI_1$ O nome do parâmetro é o endereço MAC do AP e o RSSI o valor RSSI do respetivo AP.▪ $MAC_{N-1} = RSSI_{N-1}$
Parâmetros	<ul style="list-style-type: none">▪ $MAC_N = RSSI_N$ Nota: N representa o número de AP detetados▪ motion Valor que indica se o utilizador está em movimento (campo opcional)▪ place Informação relativa ao local (campo opcional)▪ timestamp Data e hora da recolha da <i>fingerprint</i>
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.

- Obtenção de locais pertencentes à Universidade do Minho

Método HTTP GET

URL	{server}/places/um
Parâmetros	<i>Sem parâmetros</i>
Resposta	É enviado um objeto com os locais próximos ao utilizador após combinar os locais armazenados na base de dados com os fornecidos pela API do Foursquare.

- Obtenção de locais próximos ao utilizador

Método HTTP GET

URL	{server}/places/{coordinates}/search	
Parâmetros	▪ coordinates	Coordenadas GPS da posição do utilizador
Resposta	É enviado um objeto com os locais próximos ao utilizador após combinar os locais armazenados na base de dados com os fornecidos pela API do Foursquare.	

- Introdução de um local pertencente à Universidade do Minho

Método HTTP POST

URL	{server}/places/um	
Parâmetros	▪ fingerprint	<i>Fingerprint</i> representativa do local adicionado
	▪ location	Objeto com informação do local adicionado
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

- Introdução de um local exterior à Universidade do Minho

Método HTTP POST

URL	{server}/places	
Parâmetros	▪ fingerprint	<i>Fingerprint</i> representativa do local adicionado
	▪ location	Objeto com informação do local adicionado
Resposta	Em caso de sucesso é enviada uma mensagem de confirmação, em caso de insucesso é enviada uma mensagem de erro.	

Anexo D

Introdução à plataforma Android

Ao desenvolver a aplicação móvel, é importante conhecer e compreender a arquitetura e os componentes presentes na plataforma Android.

Arquitetura

Através da Figura D.1 é possível identificar que a arquitetura da plataforma Android está organizada em cinco secções e quatro camadas principais.

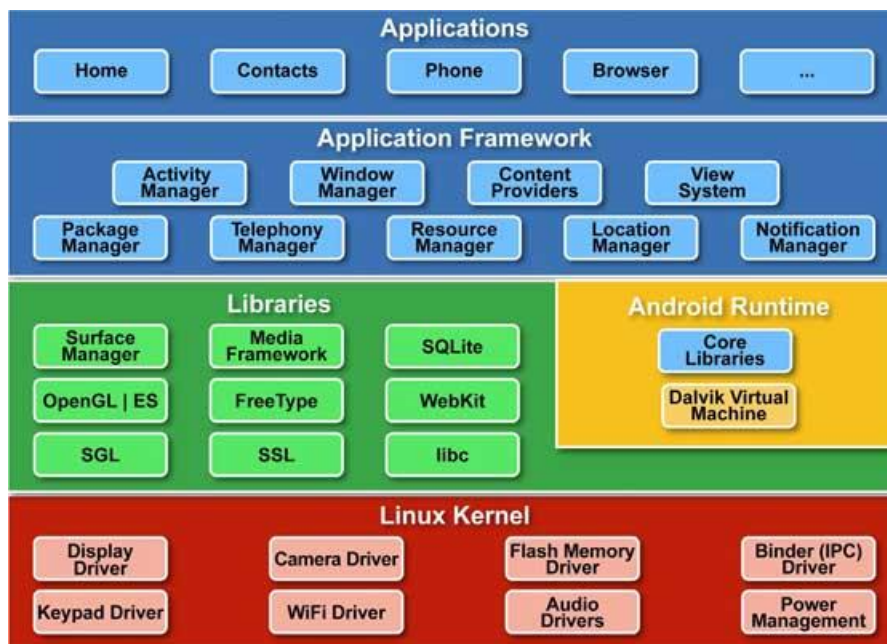


Figura D.1 - Arquitetura da plataforma Android, retirado de [57]

- Linux Kernel

Responsável pelo funcionamento básico do sistema, isto inclui: gestão de processos, gestão de memória, gestão de dispositivos como câmara, teclado, ecrã, Wi-Fi, etc.

- Libraries

Acima da camada Linux Kernel situa-se um conjunto de bibliotecas *open-source*, onde se incluem bibliotecas como: WebKit, SQLite, SSL, localização, entre outras.

- Android Runtime

Presente na segunda camada da arquitetura Android, a secção Android Runtime fornece um componente chave designado por Dalvik Virtual Machine, semelhante ao conhecido Java Virtual Machine mas desenhado e otimizado para Android. Dalvik VM utiliza os recursos básicos, de gestão de memória e multiprocessamento, do Linux. O Dalvik VM permite que as aplicações Android executem no seu próprio processo, com a sua própria instância.

- Application Framework

Esta camada fornece serviços de alto nível a aplicações em forma de classes da linguagem Java e permite a sua utilização no desenvolvimento das aplicações Android.

- Applications

Esta é a camada aplicacional, onde se encontram todas as aplicações nativas, instaladas ou desenvolvidas.

Componentes de uma aplicação Android

Os componentes são essências para o desenvolvimento de uma aplicação Android. Existem quatro componentes principais que podem ser utilizados no seu desenvolvimento. De seguida são apresentados e descritos estes quatro componentes, assim como o seu ciclo de vida.

- Activities

Uma *activity*, ou actividade, representa um ecrã que permite a interação com o utilizador. Habitualmente uma aplicação contém várias *activities*, em que todas são independentes e uma é distinguida como a *activity* principal (*Main Activity*). A navegação entre *activities* é possível assim como incluir troca de dados ao iniciar um nova *activity*. O início de novas atividades é realizado através da classe *Intent*. Uma *activity* é iniciada através da chamada de um método *callback* (*onCreate*). Existe uma

sequência de métodos *callback* definida para iniciar e parar uma *activity*. Esta sequência de métodos é designada por *lifecycle* (ciclo de vida). A Figura D.2 ilustra o ciclo de vida de uma *activity*.

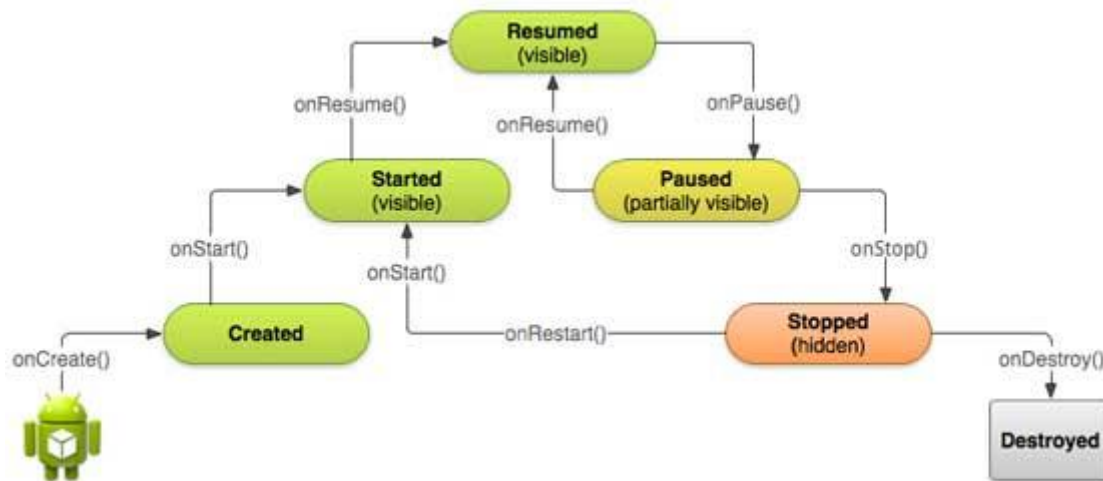


Figura D.2 - Ciclo de vida de uma *activity*, retirado de [57]

- Services

Os *services*, ou serviços, é um componente que executa em background operações com uma duração superior à habitual. Por exemplo, um *service* pode ser utilizado para processos de recolha de dados dos pontos de acesso Wi-Fi. Os *services*, ao contrário das *activities*, não possuem interface de utilizador. O *services* podem ser iniciados de formas diferentes, consoante o pretendido, através do método *startService* ou *bindService*, como ilustra a Figura D.3. A principal diferença entre ambas é que a primeira pode ser executada, mesmo que a *activity* que a chamou tenha sido destruída, enquanto a segunda permite a interação com outros componentes (por exemplo: com a *activity* que a chamou).

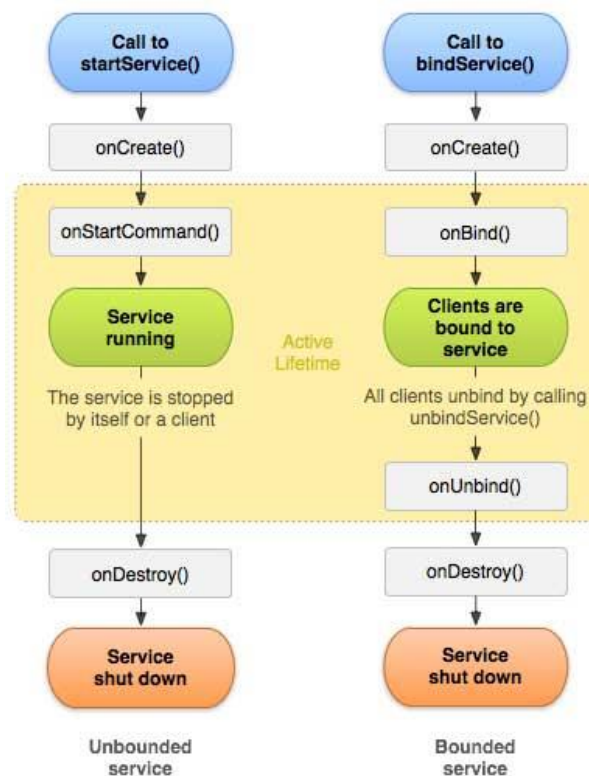


Figura D.3 - Ciclo de vida de um *service*, retirado de [57]

- Broadcast Receivers

O *broadcast receivers* respondem a mensagens provenientes de outras aplicações ou do sistema. Permite que uma aplicação possa efetuar uma determinada ação quando ocorre um determinado evento, que é escolhido, seja relativamente ao sistema ou à própria aplicação. É possível conciliar um *service* com um *broadcast receiver*, por exemplo (o mesmo exemplo dado em cima), é possível através de um *service* iniciar uma leitura do ambiente rádio Wi-Fi em redor do dispositivo móvel, e programar um *broadcast receiver* para quando receber uma informação completa sobre os pontos de acesso Wi-Fi realizar um determinado processo.

- Content Providers

Os *content providers* fornecem dados de uma aplicação para outras através de pedidos. Os dados podem ser armazenados no sistema de ficheiros ou em bases de dados (SQLite).

Existem outros componentes que são utilizados na construção das entidades descritas em cima. Através da Tabela D.1 é possível conhecê-los e compreender o seu propósito.

Tabela D.1 - Componentes adicionais de uma aplicação Android

Componentes	Descrição
Fragments	Representa um comportamento ou uma parte da interface do utilizador de uma <i>activity</i> . Por exemplo, uma <i>activity</i> pode incluir vários fragmentos, em que cada fragmento pode possuir um <i>layout</i> específico.
Views	As <i>views</i> representam a imagem da UI. As <i>views</i> são desenhadas consoante o pretendido.
Layouts	Os <i>layouts</i> controlam o formato do ecrã e a aparência das <i>views</i> .
Intents	São utilizados para interligar os componentes através de mensagens.
Resources	Compreende-se como <i>resources</i> , todos os elementos externos como <i>strings</i> , constantes, imagens.
Manifest	O <i>manifest</i> é o ficheiro (XML) de configuração da aplicação. <i>Activities</i> , <i>services</i> , permissões de utilização e versão da aplicação estão presentes neste ficheiro.