

Universidade do Minho

Escola de Engenharia

Diogo Daniel de Sousa Oliveira

**Monitorização Georreferenciada de
Dados no Ciclismo utilizando
Smartphones**

Dissertação de Mestrado

Ciclo de Estudos Integrados Conducentes

ao grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do

Professor Doutor José Augusto Afonso

outubro de 2014

DECLARAÇÃO

Nome: Diogo Daniel de Sousa Oliveira

Endereço electrónico: a52591@alunos.uminho.pt

Telemóvel: 911551252

Número do Bilhete de Identidade: 12815750

Título dissertação: Monitorização Georreferenciada de Dados no Ciclismo utilizando Smartphones

Orientador: José Augusto Afonso

Ano de conclusão: 2013/2014

Designação do Mestrado: Ciclo de Estudos Integrados conducentes ao Grau de Mestre em Engenharia de Comunicações

Área de Especialização: Engenharia de Comunicações

Escola: Escola de Engenharia da Universidade do Minho

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE

Universidade do Minho, ___/___/_____

Assinatura: _____

Agradecimentos

A conclusão deste ciclo de estudos não seria possível sem a ajuda de algumas pessoas que direta ou indiretamente me foram dando motivação para continuar.

Em primeiro lugar gostaria de mencionar que foi um privilégio realizar esta dissertação sob a orientação do Professor Doutor José Augusto Afonso. Agradeço a sua ajuda e disponibilidade, sendo os seus conhecimentos, conselhos e orientações de extrema relevância para o resultado obtido.

Agradeço à minha mãe, Maria da Conceição, por todo o apoio e motivação extra dado no decorrer do meu percurso académico, assim como à minha irmã Bárbara pela demonstração de coragem e sacrifício no árduo caminho que tem percorrido para se tornar uma investigadora de sucesso.

Agradeço à minha namorada Ana por toda a paciência e apoio demonstrados nestes últimos meses e a todos os meus amigos e companheiros de mestrado, que estiveram mais presentes, pelo seu apoio e incentivo.

Finalmente, dedico todo o trabalho e esforço em memória do meu pai, Armando Emídio, pelos sacrifícios ultrapassados juntos em prol da minha formação académica e pela sua força de vontade e esperança demonstradas, mesmo nos piores momentos. Sem ele não me teria tornado a pessoa que sou hoje, o meu muito obrigado pai.

Resumo

Nos últimos anos tem-se assistido a uma grande evolução e adesão a aplicações para dispositivos móveis que disponibilizam serviços baseados na localização. Os dispositivos disponíveis no mercado fazem-se acompanhar de um conjunto de sensores com uma precisão satisfatória. Há também a possibilidade de adquirir dados de sensores externos utilizando protocolos como o Bluetooth.

O propósito desta dissertação centra-se no desenvolvimento de uma aplicação para dispositivos móveis que permite a recolha de dados através dos diversos sensores e o seu envio para um servidor, para posterior consulta através da aplicação ou de um website. Com este sistema pretende-se melhorar e facilitar a acessibilidade do ciclista aos dados recolhidos durante um percurso. Os dispositivos móveis fazem parte da rotina diária de um utilizador, pelo que foi dada especial atenção à seleção do sistema operativo móvel que traria maiores vantagens ao sistema. A aplicação desenvolvida permite aos utilizadores partilhar os seus dados com os amigos, aderir/criar eventos, localizar amigos, consultar gráficos e ver percursos efetuados num mapa. De forma a facilitar a utilização do sistema pelos ciclistas, foi desenvolvido um website adaptável a diferentes dispositivos.

É necessário ter em consideração questões de privacidade, sendo necessária a integração de mecanismos que garantam a privacidade e segurança dos utilizadores. É fundamental minimizar o consumo de recursos e energia pela aplicação de modo a maximizar a autonomia da bateria do smartphone. Estas questões podem ser ultrapassadas através da criação de algoritmos eficientes para a recolha dos dados, recorrendo às tecnologias do dispositivo, como os diversos sensores, e utilizando o Wi-Fi no envio dos dados para o servidor. Este documento descreve a implementação do sistema, abordando as dificuldades encontradas e a forma como foram ultrapassadas, para atingir os objetivos propostos. Para validar a solução desenvolvida

a aplicação foi testada num ambiente real com recurso a uma bicicleta, sendo obtidos resultados satisfatórios quanto ao propósito para o qual o sistema foi projetado.

Abstract

In recent years there has been a great evolution regarding applications for mobile devices that provide location-based services. The devices available on the market provide a set of sensors with satisfactory precision. It is also possible to acquire data from external sensors using protocols such as Bluetooth.

The purpose of this dissertation focuses on the development of an application for mobile devices that allows the collection of data using the different sensors and, ultimately, sending it to a server where it can be accessed later using the application or via a Web page. With this system we aim to improve and facilitate the cyclist's access to data produced during a specific pathway. Mobile devices are part of the daily routine, so special attention was given to the selection of the operating system that would bring greater benefits to the system. The application developed allows users to share their data with friends, join or create events, find friends, see and consult graphs and previous pathways in a map. We also developed a Web page compatible with different devices in order to facilitate the system use by cyclists.

It is necessary to consider privacy issues which requires the integration of mechanisms to ensure the privacy and security of users. It is crucial to minimize the consumption of resources and energy by the application in order to maximize the battery of the smartphone. These issues can be overcome by creating efficient algorithms for data collection, using the device resources, such as its various sensors, and using Wi-Fi to send data to the server. This work describes the system implementation, addressing the difficulties encountered and how they were overcome to achieve our goals. For validation purposes the application was tested in a real environment using a bicycle and, ultimately, we were able to achieve the purpose for which the system was designed.

Índice de conteúdos

Agradecimentos	iii
Resumo	v
Abstract	vii
Índice de conteúdos	ix
Lista de figuras	xiii
Lista de tabelas	xvii
Lista de abreviaturas	xix
1. Introdução	1
1.1 Enquadramento e motivação	1
1.2 Objetivos	3
1.3 Estrutura da dissertação	4
2. Estado da arte	7
2.1 Sistemas operativos móveis.....	7
2.1.1 Arquitetura Android	8
2.1.2 Versões da plataforma Android.....	9
2.1.3 Ciclo de vida de uma <i>Activity</i>	13
2.1.4 Ciclo de vida de um <i>Fragment</i>	15
2.2 Sistemas de gestão de bases de dados.....	17
2.2.1 MySQL.....	17
2.2.2 SQLite.....	18
2.3 Linguagens de programação Web	18

2.3.1	PHP.....	19
2.3.2	JavaScript.....	19
2.4	Linguagem de marcação	20
2.4.1	HTML.....	20
2.4.2	XML.....	21
2.5	Trabalho relacionado	21
2.5.1	BikeNet	23
2.5.2	Endomondo	26
2.5.3	Roadroid	30
3.	Implementação do sistema proposto	35
3.1	Requisitos.....	35
3.2	Arquitetura do sistema	37
3.3	Servidor.....	38
3.3.1	Serviço Web.....	39
3.3.2	Base de dados.....	40
3.4	Software.....	43
3.5	Hardware.....	44
3.6	Aplicação móvel	45
3.6.1	Manifesto Android.....	45
3.6.2	Google Maps.....	46
3.6.3	Base de dados.....	47
3.6.3.1	Upload SQLite para MySQL.....	48
3.6.4	Registo e autenticação	49
3.6.4.1	Preservação da sessão iniciada	52

3.6.5	Navegar entre <i>Activities</i>	53
3.6.6	Navegação na aplicação	54
3.6.6.1	Amigos	55
3.6.6.2	Eventos	58
3.6.6.3	Histórico.....	60
3.6.6.4	Localizar Amigos	61
3.6.6.5	Gráficos.....	62
3.6.7	Menu de overflow	64
3.6.8	Modo de navegação	66
3.6.8.1	GPS.....	68
3.6.8.2	Altitude	69
3.6.8.3	Distância	70
3.6.8.4	SMS de segurança.....	70
3.6.8.5	Trepidação	71
3.7	Website	72
3.7.1	Página de apresentação.....	73
3.7.2	Página principal.....	76
3.7.2.1	Amigos	77
3.7.2.2	Eventos	79
3.7.2.3	Histórico.....	81
3.7.2.4	Gráficos.....	81
3.7.2.5	Barra de navegação	82
3.8	NFC.....	83
4.	Resultados e discussão	85

4.1	Teste da altitude	85
4.2	Teste da distância	89
4.3	Teste de velocidade	89
4.4	Teste da trepidação	90
4.5	Teste da navegação.....	93
4.6	Bateria e dados móveis.....	96
5.	Conclusões	99
5.1	Trabalho futuro	100
	Referências	103
	Apêndice A – Principais classes Android	109
	Apêndice B – Protocolo de comunicação	111
	Apêndice C – NFC	117
	NFC Tools.....	117
	NFC Tasks.....	117

Lista de figuras

Figura 1 – Previsão de vendas de bicicletas elétricas nos próximos anos [2].	2
Figura 2 - Venda de smartphones por sistema operativo a nível mundial [22].	8
Figura 3 - Arquitetura Android (adaptado de [32]).	9
Figura 4 - Ciclo de vida da <i>Activity</i> [14].	14
Figura 5 - Ciclo de vida do <i>Fragment</i> [15].	16
Figura 6 - Visão global do sistema BikeNet [7].	24
Figura 7 - Representação lógica da Bicycle Area Networking (BAN) [7].	25
Figura 8 - Portal Web para consulta de dados e submissão de <i>queries</i> [7].	26
Figura 9 - Menu principal disponível ao utilizador na aplicação Endomondo para Android.	27
Figura 10 – Diferentes opções de visualização no decorrer do treino.	28
Figura 11 – Consulta de uma rota efetuada no histórico do utilizador.	29
Figura 12 - Consulta de uma rota do histórico de um utilizador.	30
Figura 13 – Exemplo de visualização do mapa na página Web [9].	31
Figura 14 – Modelo de bicicleta testado [8].	33
Figura 15 - Diagrama global do sistema.	38
Figura 16 - Diagrama global do servidor.	39
Figura 17 – Diagrama do serviço Web.	40
Figura 18 – Diagrama E-R.	42
Figura 19 – Versão Java instalada na nossa máquina.	43
Figura 20 - Permissões da aplicação desenvolvida.	46

Figura 21 - <i>Key</i> para o Google Maps.	46
Figura 22 – Referência ao OpenGL ES v2 no <i>AndroidManifest.xml</i>	46
Figura 23 - Diagrama E-R SQLite.	47
Figura 24 – Associar a entidade <i>localizacao</i> com a entidade <i>trajeto</i>	48
Figura 25 - Diagrama de blocos alusivo à sincronização SQLite com MySQL.	49
Figura 26 - Processo de registo.	50
Figura 27 - Diagrama de sequência relativo ao registo e à autenticação.	51
Figura 28 - Diagrama de blocos referente à preservação da sessão.	53
Figura 29 – Representação do início de uma nova <i>Activity</i> sem que a anterior seja destruída.	54
Figura 30 - Diagrama geral do <i>spinner</i>	55
Figura 31 - <i>Tab</i> AMIGOS.	56
Figura 32 - <i>Tab</i> ADICIONAR.	57
Figura 33 - <i>Tab</i> REMOVER.	57
Figura 34 – <i>Tab</i> onde se encontram todos os eventos.	58
Figura 35 - <i>Tab</i> para visualizar os eventos que o utilizador aderiu.	59
Figura 36 - <i>Tab</i> para criar um evento.	60
Figura 37 – Apresentação do histórico de um utilizador.	61
Figura 38 – Localizar amigos.	62
Figura 39 – Gráfico de linhas (gasto de bateria da bicicleta).	63
Figura 40 - Gráfico de barras (quilómetros efetuados).	63
Figura 41 - Diagrama geral do <i>action overflow</i>	64
Figura 42 - Modo de navegação.	67
Figura 43 – Linhas imaginárias da latitude e longitude [51].	69

Figura 44 – Diagrama de sequência da obtenção da pressão ao nível do mar.	70
Figura 45 – Exemplo de um SMS de pedido de ajuda.	71
Figura 46 - Sistema de coordenadas no Android [57].	72
Figura 47 - Esquema geral do website.	73
Figura 48 - Página de apresentação.	74
Figura 49 - Página de contato.	75
Figura 50 - Página de login com credenciais inseridas incorretamente.	76
Figura 51 - Página principal.	77
Figura 52 - Página principal (calendário e gráfico do top4).	77
Figura 53 – Lista de amigos.	78
Figura 54 – Histórico de um amigo.	78
Figura 55 – Trajeto de um amigo.	79
Figura 56 – Lista de eventos de um utilizador.	79
Figura 57 – Informação relativa a um evento que o utilizador aderiu.	80
Figura 58 – Criar um evento.	81
Figura 59 – Gráficos disponíveis ao utilizador.	82
Figura 60 – Dados alusivos ao perfil do utilizador.	83
Figura 61 – <i>Smart tag</i> NFC em autocolante.	84
Figura 62 - Obtenção da altitude recorrendo ao EarthTools.	86
Figura 63 - Obtenção da altitude recorrendo ao Google Earth.	86
Figura 64 - Altitude na aplicação desenvolvida.	87
Figura 65 - Altitude na aplicação OsmAnd vs altitude na aplicação Phone Tester.	88
Figura 66 - Teste da altitude.	88
Figura 67 - Teste relativo à distância percorrida.	89

Figura 68 - Teste relativo à velocidade.	90
Figura 69 – Suporte utilizado para o smartphone no teste da trepidação.....	91
Figura 70 - Teste em paralelepípedo com suspensão a uma velocidade de 10 km/h durante 58 segundos.	92
Figura 71 - Teste em paralelepípedo sem suspensão a uma velocidade de 10 km/h durante 58 segundos.	92
Figura 72 - Teste em asfalto com suspensão a uma velocidade de 10 km/h durante 45 segundos.	93
Figura 73 - Teste em asfalto sem suspensão a uma velocidade de 10 km/h durante 45 segundos.	93
Figura 74 - Navegação sem A-GPS, com A-GPS interrompido e com A-GPS constantemente ligado.	94
Figura 75 – Percurso Endomondo à esquerda e percurso da aplicação desenvolvida à direita.....	95
Figura 76 - Gradiente de cores utilizado para a altitude.	96
Figura 77 – Trajeto com recurso ao gradiente de cores para a altitude.	96
Figura 78 – Bateria e dados gastos durante um percurso com a aplicação desenvolvida.	97
Figura 79 - Bateria e dados gastos durante um percurso com a aplicação Endomondo.	98
Figura 80 – Alerta de proximidade de amigo.....	102
Figura 81 – Seleção de uma aplicação a ser escrita numa <i>tag</i>	118
Figura 82 – Processo de escrita numa <i>tag</i>	119

Lista de tabelas

Tabela 1 - Versões da plataforma e respetiva API (adaptado de [13]).	12
Tabela 2 - Evolução do HTML (adaptado de [29]).	21
Tabela 3 - Principais classes utilizadas no desenvolvimento da aplicação.	109
Tabela 4 – Métodos da classe AsyncTask.	110

Lista de abreviaturas

ADT	Android Development Tools
A-GPS	Assisted-Global Positioning System
API	Application Programming Interface
AVD	Android Virtual Device
BD	Base de Datos
BAN	Bicycle Area Networking
DEM	Digital Elevation Model
GPRS	General Packet Radio Service
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IRI	International Roughness Index
JSON	JavaScript Object Notation
NASA	National Aeronautics and Space Administration
NFC	Near Field Communication
NGA	National Geospatial-Intelligence Agency
PC	Personal Computer
PHP	PHP: Hypertext Preprocessor

SAP	Sensor Access Point
SDK	Software Development Kit
SGBD	Sistema de Gestão de Base de Dados
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SMS	Short Message Service
SO	Sistema Operativo
SQL	Structured Query Language
SRTM	Shuttle Radar Topography Mission
URL	Uniform Resource Locator
XML	eXtensible Markup Language

1. Introdução

Neste capítulo é feito um breve enquadramento e motivação ao tema da dissertação, assim como os objetivos a cumprir. No final do capítulo é descrita a estrutura adotada para a escrita desta dissertação.

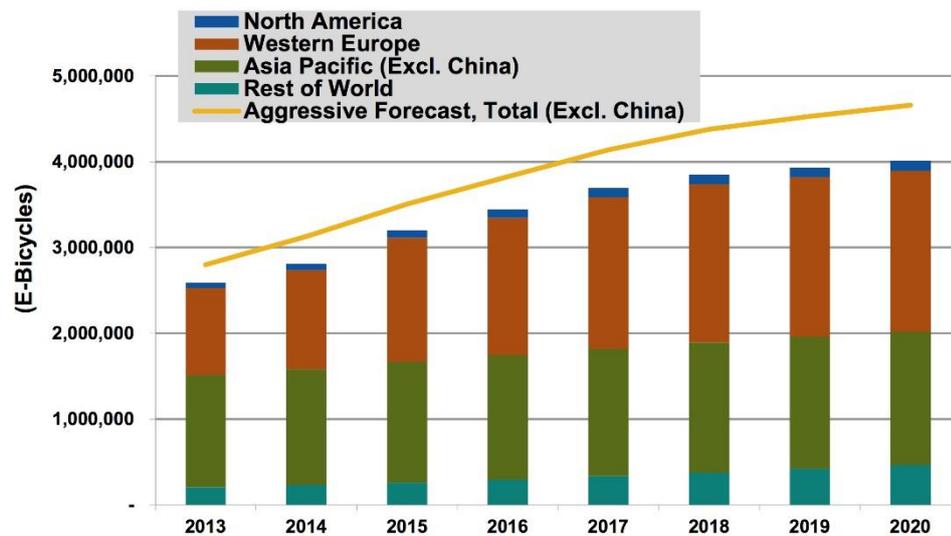
1.1 Enquadramento e motivação

Os smartphones estão se tornando cada vez mais os dispositivos centrais de comunicação e processamento na vida das pessoas, o que faz com que o estudo e desenvolvimento de sistemas de sensorização móvel seja uma atrativa área emergente de investigação [1].

Relativamente a aplicações direcionadas para a locomoção por bicicletas elétricas, ainda existe uma vasta quota de mercado a preencher; existem, no entanto, aplicações direcionadas para as bicicletas convencionais que já tiram um bom partido dos vários parâmetros monitorizáveis [3],[4],[5].

Podemos observar na Figura 1, o aumento significativo do número de bicicletas elétricas previsto para os próximos anos [2]. De referir que desta previsão se encontra excluída a China, um país onde existe também uma possível elevada aderência devido ao número de habitantes.

Annual E-Bicycle Sales by Region and Forecast Scenario, World Markets: 2013-2020



(Source: Navigant Research)

Figura 1 – Previsão de vendas de bicicletas elétricas nos próximos anos [2].

Cada vez mais os ciclistas gostam de monitorizar vários parâmetros no decorrer do seu exercício, e com as bicicletas elétricas não será diferente. Um dos entraves na monitorização de certos parâmetros é o preço de certos dispositivos de monitorização, mas uma vez que o sistema proposto parte de uma utilização do smartphone do próprio utilizador, isso representa uma mais-valia, pois não é necessário esse investimento extra.

A principal motivação deste trabalho consiste em satisfazer os requisitos do ciclista, que pode, nomeadamente no decorrer do seu percurso, recolher parâmetros tais como a velocidade da bicicleta, a cadência, a distância, a altitude, a postura, a frequência cardíaca, e a potência. A georreferenciação desta informação com a sua localização GPS permite a visualização, armazenamento e partilha de mapas dos percursos e uso de gradientes de cor para fornecer informação visual associada às variações dos parâmetros monitorizados.

Com o intuito de melhorar o seu desempenho ou até mesmo o da bicicleta, o ciclista cada vez mais pretende aceder às informações recolhidas, que no sistema

desenvolvido estão disponíveis localmente no smartphone e através da Internet acedendo a um website [6].

O ciclista hoje em dia gosta de aderir a eventos onde estejam presentes outros ciclistas, tornando assim os seus passeios mais interativos, possibilitando a partilha de experiencias relativas a alguns locais que conheçam e que achem interessantes. No sistema proposto o utilizador tem, não só a opção de aderir a um evento, como a de criar um evento, sendo que quem agenda o evento, tem de inserir algumas informações gerais do percurso, para que outros ciclistas vejam se é do seu agrado aderir ao evento.

Após alguma pesquisa e pensando na segurança do utilizador em caso de acidente, não foi encontrada nenhuma aplicação que enviasse qualquer tipo de alerta se o mesmo se encontrasse parado por um tempo delimitado. Surgiu então a ideia de, ao fim de um determinado tempo ao detetar que o dispositivo se encontra parado, emitir um alerta para um número de telemóvel predefinido anteriormente contendo as suas coordenadas geográficas. A opção encontra-se ativa por definição, podendo o utilizador desativá-la a qualquer momento.

Com base nos dados recolhidos pelos sensores é possível também determinar características do terreno como a rugosidade, trepidação e inclinação, permitindo assim aos ciclistas escolher ou evitar percursos de acordo com as características da superfície desejada.

1.2 Objetivos

Pretende-se no âmbito desta dissertação a realização de um conjunto de tarefas no âmbito da programação móvel, de base de dados e da programação Web com recurso a um servidor. Quanto à aplicação móvel, é necessário que seja executada com o menor número de falhas possível, desde que o dispositivo (smartphone) cumpra os requisitos mínimos exigidos (secção 3.7.1). Deverá ser efetuada a recolha de informação relativa a um trajeto, permitindo a sua partilha com um grupo restrito de amigos e que o utilizador a possa consultar no seu histórico e com recurso a um mapa

possa visualizar pormenorizadamente o seu percurso. Com a utilização de um gradiente de cores pode-se, por exemplo, distinguir a altitude durante determinado trajeto. Deverá ainda ser permitida a criação/adesão de eventos, localizar amigos e tratar questões de segurança relacionadas diretamente com o utilizador.

Relativamente ao website deve fornecer uma compatibilidade entre os diferentes *browsers* e permitir a sua utilização em diversos dispositivos (computador, tablet ou smartphone) adaptando-se às diferentes dimensões do ecrã. Deve ser assegurada a comunicação entre a aplicação móvel e o servidor, assim como o website e o servidor, desta forma é possível manter os dados atualizados, podendo ser feita a sua consulta por parte do utilizador ou pelo seu grupo de amigos no website/aplicação.

Na parte final, testar ambas as aplicações num cenário real, recorrendo a um utilizador registado.

1.3 Estrutura da dissertação

No primeiro capítulo é feito um breve enquadramento ao tema da dissertação. É também neste capítulo que é descrita a motivação assim como os objetivos a cumprir. No final do capítulo é descrita a estrutura adotada para a escrita desta dissertação.

O capítulo dois contém o estado da arte onde é efetuado um estudo dos diferentes sistemas operativos móveis disponíveis atualmente no mercado. É efetuada uma análise à arquitetura do Android e são dadas a conhecer as diferentes versões da plataforma, assim como as principais evoluções entre elas. São abordadas a base de dados do dispositivo e a base de dados do servidor, as linguagens de programação e as linguagens de marcação. No final são mencionadas algumas soluções relacionadas com o tema da dissertação que servirão de base e de referência ao desenvolvimento da mesma.

No capítulo três é abordado o processo de implementação do sistema, definindo os requisitos e descritas as ferramentas utilizadas tanto a nível de software como de hardware. É efetuada uma descrição geral do sistema e dos componentes envolvidos no servidor. São descritas todas as envolventes da aplicação móvel e do website. No

final é apresentado o propósito da utilização do Near Field Communication (NFC) no sistema desenvolvido.

O capítulo quatro documenta os resultados obtidos relativos aos diferentes testes realizados, sendo efetuada uma discussão sobre os mesmos. Foram efetuados testes relacionados com a altitude, distância, velocidade, trepidação, navegação (vantagens da utilização do A-GPS, utilização de gradiente de cores, entre outros) e consumo de bateria e dados móveis.

No quinto e último capítulo são apresentadas as conclusões referentes a todo o trabalho desenvolvido. Adicionalmente, são apresentadas algumas sugestões de trabalho futuro.

2.Estado da arte

Neste capítulo é feita uma análise aos sistemas operativos móveis com maior sucesso no mercado, sendo dada uma maior ênfase ao sistema operativo Android, que foi o escolhido para o desenvolvimento da aplicação. É feito um estudo aos sistemas de gestão de bases de dados, às linguagens de programação Web e às linguagens de marcação. No final do capítulo são abordados alguns sistemas relacionados com o tema desta dissertação e que serviram de base para a implementação do sistema desenvolvido.

2.1 Sistemas operativos móveis

No mercado mundial existe uma expressiva concorrência em relação ao desenvolvimento de aplicações ricas em funcionalidades para smartphones, oferecendo uma melhor experiência e satisfação nas necessidades do utilizador final.

Neste momento os principais sistemas operativos (SO) para smartphones, e sendo mesmo grandes concorrentes entre si são: o Android, da Google [19]; iOS, da Apple [20]; e o Windows Phone, da Microsoft [21]. Quanto ao Android e ao Windows Phone, os mesmos não são vinculados a um único hardware, podendo vários fabricantes licenciar os seus smartphones com um destes sistemas operativos. O Android tem ainda a grande vantagem de que a sua plataforma também é livre e de código aberto, sendo a sua licença flexível, permitindo que cada fabricante possa realizar alterações no código-fonte para customizar os seus produtos. Já o iOS caracteriza-se por não permitir que o seu SO seja executado em hardware de terceiros, sendo de uso exclusivo da Apple.

Na Figura 2 pode-se observar a ascensão do Android, um pouco devido ao elevado número de marcas que querem dar o seu toque pessoal aos seus dispositivos, e verificar quais os SOs que têm tido um maior destaque nos últimos anos. Após uma

análise destes três SOs, chegou-se á conclusão que o que correspondia melhor aos nossos objetivos era o Android.

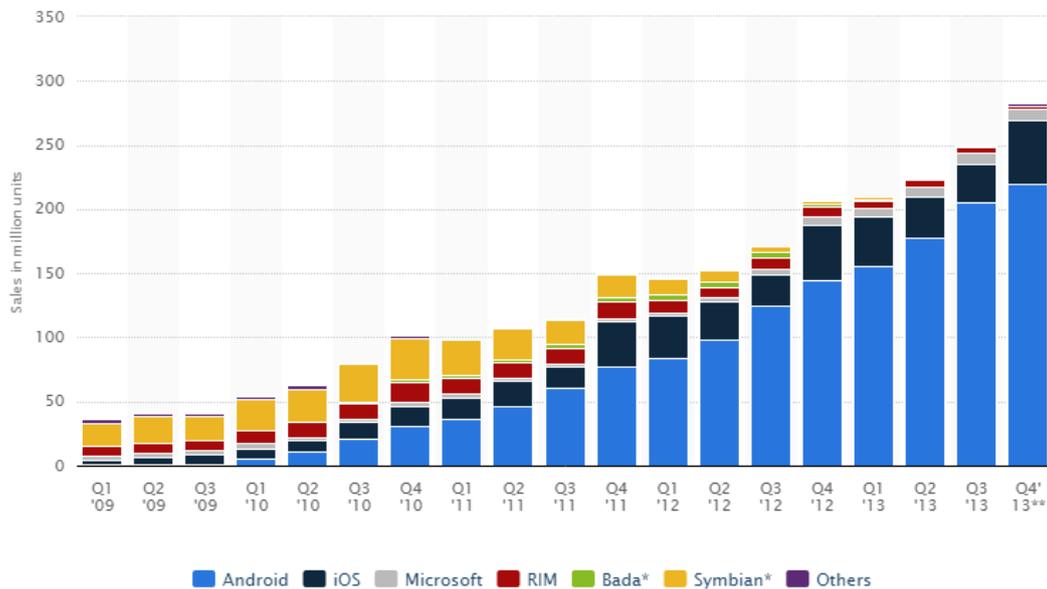


Figura 2 - Venda de smartphones por sistema operativo a nível mundial [22].

2.1.1 Arquitetura Android

Na Figura 3 é apresentada a arquitetura geral do Android, onde se pode observar na camada superior da pilha (camada Aplicações) as principais funcionalidades para um usuário básico, isto é, que pretende dar uso a funcionalidades básicas como telefonar, navegar na Web e aceder às suas aplicações. As aplicações interagem diretamente com a próxima camada (camada Quadro de Aplicações), sendo que é nela onde se gerenciam as funções básicas do telemóvel, tais como gestão de recursos, gestão de chamadas, etc. Na próxima camada (Bibliotecas) é onde se encontram as bibliotecas nativas do Android, e ainda se encontra o ambiente de execução Java. Esta inclui um conjunto de bibliotecas Java e a Máquina Virtual Dalvik, que é o software responsável pela execução das aplicações no dispositivo Android. Na parte inferior da pilha está o Núcleo Linux, sendo que este nunca interage realmente com os utilizadores e os programadores.



Figura 3 - Arquitetura Android (adaptado de [32]).

2.1.2 Versões da plataforma Android

Na Tabela 1 pode-se observar todas as versões da plataforma Android, bem como as respectivas APIs e seus nomes, até à data de hoje, sendo o Android 4.4 a versão mais recente. Apresenta-se a seguir um destaque de algumas funcionalidades das versões mais utilizadas, assim como as suas evoluções relativamente à versão anterior:

Android 1.5

- Suporte ao teclado QWERTY;
- Gravação de vídeos com envio para o YouTube;
- Possibilidade de inserir *widgets*;
- Performance na Interface de Aplicações melhorada;
- Suporte ao Bluetooth A2DP;
- Função copiar/colar.

Android 2.1

- Suporte a novos hardwares, permitindo smartphones com uma maior capacidade de memória, maior resolução e processadores mais rápidos;
- Suporte até nove telas principais;
- Papéis de parede animados;
- Suporte ao HTML 5;
- Bluetooth 2.1;
- Gestor de bateria.

Android 2.2

- Velocidade de processamento melhorada;
- Adobe Flash 10.1;
- Instalação de aplicativos no cartão de memória;
- Atualizações automáticas;
- Sincronização com um PC.

Android 2.3

- Melhoria no consumo de energia;
- Melhoria na interface do utilizador;
- Teclado mais rápido;

- Suporte ao protocolo SIP (Session Initiation Protocol).

Android 4.1

- Opção de desativar alertas aplicação a aplicação;
- Reorganização automática de atalhos e *widgets*;
- Melhorias na busca por voz;
- Menor consumo de energia;
- Aplicação Google Now.

Android 4.4

- Aplicações com uso de memória otimizada;
- Maior fluidez;
- Gestão centralizada de arquivos;
- Par de sensores virtuais que reconhecem e contam os passos;
- Google Cloud Print.

Tabela 1 - Versões da plataforma e respetiva API (adaptado de [13]).

Versão da Plataforma	API	Nome
Android 1.0	1	BASE
Android 1.1	2	BASE_1_1
Android 1.5	3	CUPCAKE
Android 1.6	4	DONUT
Android 2.0	5	ECLAIR
Android 2.0.1	6	ECLAIR_0_1
Android 2.1.x	7	ECLAIR_MR1
Android 2.2.x	8	FROYO
Android 2.3 Android 2.3.1 Android 2.3.2	9	GINGERBREAD
Android 2.3.3 Android 2.3.4	10	GINGERBREAD_MR1
Android 3.0.x	11	HONEYCOMB
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.2	13	HONEYCOMB_MR2
Android 4.0 Android 4.0.1 Android 4.0.2	14	ICE_CREAM_SANDWICH
Android 4.0.3 Android 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.1 Android 4.1.1	16	JELLY_BEAN
Android 4.2 Android 4.2.2	17	JELLY_BEAN_MR1
Android 4.3	18	JELLY_BEAN_MR2
Android 4.4	19	KITKAT

2.1.3 Ciclo de vida de uma *Activity*

As aplicações correm os seus próprios processos, sendo que estes arrancam e param conforme são necessários. Caso haja necessidade de reclamar recursos, estes processos podem ser eliminados. Há que ter em conta ainda que sempre que acionamos uma nova Atividade (*Activity*) a anterior é interrompida.

Na Figura 4 é possível observar o ciclo de vida da *Activity*. O `onCreate()` é o primeiro método a ser executado, e quando a *Activity* vai ficar visível para o utilizador é executado o método `onStart()`. O método `onResume()` ocorre quando a *Activity* é executada, e significa que a aplicação está a interagir com o utilizador. O método `onPause()` ocorre quando é preciso poupar energia ou quando, por exemplo, recebemos uma chamada. Quando a *Activity* está a ser encerrada ocorre o método `onStop()` e, caso seja necessária a aplicação novamente, é executado o método `onRestart()`, que também ocorre quando uma *Activity* para por algum motivo. O método `onDestroy()` ocorre quando a aplicação é fechada, e se o Android precisar de recursos também pode finalizar a *Activity*.

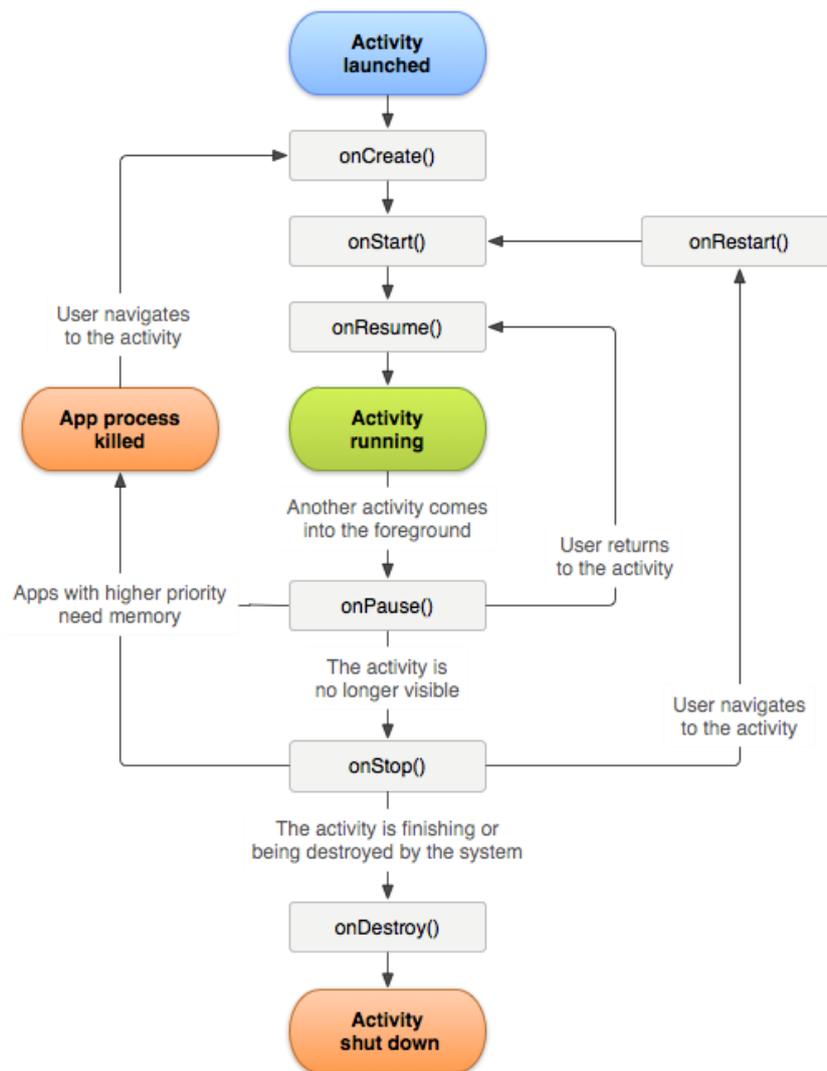


Figura 4 - Ciclo de vida da *Activity* [14].

Podemos concluir que existem três fases principais num ciclo de vida de uma aplicação, sendo elas:

- *Entire life time* – todo o ciclo de vida entre o início e a destruição, ou seja, entre o `onCreate()` e o `onDestroy()`;
- *Visible life time* – entre o `onStart()` e o `onStop()`, sendo o tempo em que o utilizador pode visualizar e interagir com a aplicação.
- *Foreground life time* – durante este tempo a *Activity* está á frente de todas as outras no ecrã (em primeiro plano), encontra-se entre o `onResume()` e o `onPause()`.

2.1.4 Ciclo de vida de um *Fragment*

Os *Fragments* surgiram após a versão 3.0 do Android chamada HoneyComb (API 11). Um *Fragment* é uma classe que implementa uma fração de uma *Activity*, representando uma operação em particular ou interface correndo dentro de uma *Activity* [15]. Permitem um *design* de *Activities* mais modular, fazendo com que uma aplicação se adapte facilmente a diferentes orientações e resoluções.

Um *Fragment* deve estar embutido numa *Activity*, ou seja, eles não podem correr sem estarem associados a uma *Activity*, sendo comum um *Fragment* definir o seu próprio *layout*. Possui o seu próprio ciclo de vida, que é muito próximo do ciclo de vida da *Activity* que o contém, podendo ser adicionados e removidos dinamicamente. Pode-se concluir que o ciclo de vida da *Activity* na qual o *Fragment* se encontra afeta diretamente o ciclo de vida do *Fragment*, funcionando ambos em conjunto.

Como classes associadas surgem:

- `android.app.Fragment` – Classe base para definição de *Fragments*.
- `android.app.FragmentManager` – classe para interagir com os *Fragments* dentro de uma *Activity*.
- `android.app.FragmentTransaction` – classe para suportar um conjunto de operações atômicas com *Fragments*, por exemplo: adicionar, remover e substituir.

Quanto ao suporte para versões anteriores à API 11, este pode ser efetuado utilizando a classe `android.support.v4.Fragment`.

Na Figura 5 é possível observar o ciclo de vida de um *Fragment*. O `onAttach()` é chamado quando o *Fragment* é adicionado à *Activity*, e o `onCreate()` está relacionado com a criação inicial do *Fragment*. O `onCreateView()` é chamado para associar o *layout* ao *Fragment*, enquanto o `onActivityCreated()` é chamado quando o método `onCreate()` da *Activity* retorna. Quando o `onStart()` ocorre, o *Fragment* é colocado visível ao utilizador, e com o `onResume()` o *Fragment* permite interação com o utilizador. Quanto ao `onPause()`, o *Fragment* já não interage com o utilizador, e no `onStop()` o *Fragment* já não se encontra visível ao utilizador. O `onDestroyView()` avisa

o *Fragment* que a sua *view* está a ser destruída, para que este possa limpar todos os recursos associados, enquanto no `onDestroy()` é efetuada a limpeza do estado do *Fragment*. O `onDetach()` é chamado quando o *Fragment* esta a ser desassociado da *Activity*.

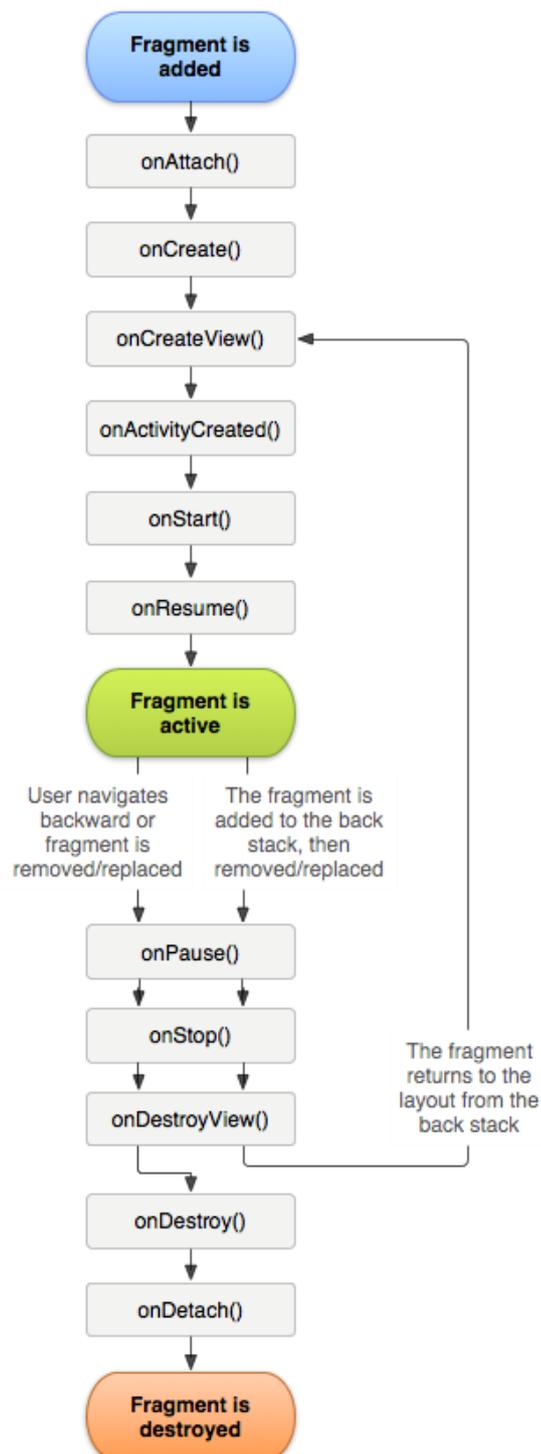


Figura 5 - Ciclo de vida do *Fragment* [15].

2.2 Sistemas de gestão de bases de dados

Um Sistema de Gestão de Base de Dados (SGBD) é uma aplicação informática que fornece a *interface* entre os dados que são armazenados fisicamente na base de dados e o utilizador [33]. Um SGBD necessita de uma linguagem para permitir aos utilizadores operarem sobre a base de dados, sendo usada maioritariamente a linguagem Structured Query Language (SQL).

Dentro dos mais conhecidos temos o MySQL, o PostgreSQL, o Oracle, o DB2 e o SQL-Server. No nosso sistema é utilizado o MySQL no servidor e no dispositivo móvel é utilizado a biblioteca SQLite, que não precisa executar um SGBD.

2.2.1 MySQL

O MySQL é um SGBD relacional, sendo a sua vertente gratuita aplicável para fins não comerciais [16]. A sua maior valia é ser um software livre, que conta com um rápido desenvolvimento de toda a comunidade.

A versão mais recente disponível é a 5.6, estando em fase de testes a versão 5.7 [17]. A partir da versão 5.5 o MySQLServer suporta conexões TCP/IP de clientes que se conectam através de IPv6.

Algumas das principais características são [18]:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (*drivers* e módulos de *interface* para diversas linguagens de programação);
- Variedade quanto ao tipo de dados (FLOAT, DOUBLE, CHAR, VARCHAR, BINARY, VARBINARY, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM);
- Pouco exigente a nível de recursos (hardware);
- Segurança (sistema de privilégios e senhas flexível e seguro);
- Escalabilidade e limites (suporte para bases de dados de grandes dimensões e um suporte até 64 índices por tabela);

2.2.2 SQLite

O SQLite é uma pequena biblioteca, desenvolvida em linguagem C [23], que implementa um amplo subconjunto do *standard* SQL-92 e pode ser integrada em programas escritos em diferentes linguagens, com o intuito de possibilitar a manipulação de dados através de instruções SQL. A sua principal diferença é que não necessita de acesso a um SGBD.

Encontra-se atualmente na versão 3.8.6 [24] e algumas das suas principais características são:

- Software gratuito;
- Toda a base de dados é guardada localmente (junto com a aplicação);
- Suporte até 2 terabytes de dados;
- Não necessita de instalação, configuração ou administração.

2.3 Linguagens de programação Web

São linguagens de programação com maior intuito para o desenvolvimento de páginas Web e aplicações que corram na Internet. Cada linguagem possui as suas próprias características e diferentes formas de se realizarem determinadas ações, por isso, existe a necessidade de as estudar e compreender, para escolher aquela em que se possa extrair ao máximo o seu potencial.

As linguagens Web estão divididas entre as que correm do lado do cliente (*client-side*) e as que correm do lado do servidor (*server-side*). Nas *client-side* temos como exemplo: JavaScript, VBScript, ActionScript e Dart. Nas *server-side* temos como exemplo: C; C++; C#; Java; PHP; Python; ASP.NET; Ruby on Rails e a Perl.

Após algum estudo, foram escolhidas, dentre as linguagens acima mencionadas, o JavaScript e o PHP.

2.3.1 PHP

O PHP é uma tecnologia muito popular no desenvolvimento de aplicações Web [25]. O seu código é executado do lado do servidor e pode gerar HTML, assim como criar um padrão de texto XML e enviar para o cliente. Pode interagir com bases de dados e aplicações existentes no servidor, com a vantagem de não expor o código-fonte para o cliente, o que é útil quando se trata de informação confidencial (password).

O PHP possui também suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e muitos outros.

Algumas das suas principais características são [26]:

- Portabilidade (pode ser utilizado em vários SO);
- Utilização gratuita;
- Exige poucos recursos do sistema;
- Execução rápida e estável;
- Código-fonte aberto (permite uma melhor ajuda na resolução de problemas através da comunidade);
- Suportado pela maioria dos Servidores Web atuais (Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet Servers, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd, e muitos outros);
- Intercâmbio de dados complexos (Web Distributed Data eXchange (WDDX)).

2.3.2 JavaScript

O JavaScript é uma linguagem de programação baseada em *scripts*, interpretada e orientada a objetos baseada em protótipos e em *first-class functions* [27], e encontra-se atualmente na versão 1.8.5 [28]. Permite tanto a criação de aplicações

que correm no lado do cliente (aplicações de *front-end*), como aplicações que correm do lado do servidor (aplicações *server-side*, como Node.js). Não funciona tanto como uma linguagem independente, mas é facilmente incorporada em outras aplicações, permitindo por exemplo a criação de aplicações que correm a partir do *browser*. O JavaScript utiliza uma tipagem dinâmica, ou seja, se o tipo de conteúdo mudar durante a execução do programa a variável muda de tipo dinamicamente.

2.4 Linguagem de marcação

As linguagens de marcação são projetadas para o processamento, definição e apresentação do texto, especificando código para formatação do *layout* e do estilo dentro de um ficheiro de texto. A marcação normalmente é omitida da versão de texto apresentada ao utilizador, e o código para especificar a formatação é denominado de *tag*.

Como exemplos de linguagens de marcação temos: SGML; MathML; HTML; XML e XHTML (reformulação do HTML, baseada em XML). As linguagens utilizadas foram HTML e XML.

2.4.1 HTML

O HTML é uma linguagem de marcação para a produção de páginas na Web, sendo que a maioria dos seus elementos são escritos com um marcador (*tag*) de abertura e outro para fechar, ficando o conteúdo entre os dois. Pode ser criado a partir de qualquer editor de texto e é interpretado pelos navegadores. Não é *case-sensitive*, o que significa que as *tags* tanto podem ser escritas em maiúsculas como em minúsculas. Na Tabela 2 é possível observar a evolução do HTML desde a sua primeira versão em 1991.

Tabela 2 - Evolução do HTML (adaptado de [29]).

Versão	Ano
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2012

2.4.2 XML

O XML foi desenvolvido com a finalidade de solucionar algumas das limitações do HTML, sendo o seu formato baseado em texto simples para representar as informações estruturadas [30]. Um dos seus principais propósitos é a facilidade de compartilhamento de informação através da Internet [31], por exemplo, para que páginas Web ou aplicações que atualizam frequentemente o seu conteúdo, como páginas de notícias ou *blogs*, possam fornecer um *feed* XML para que seja possível acompanhar as mudanças de conteúdo.

2.5 Trabalho relacionado

Com os smartphones a fazerem cada vez mais parte do cotidiano das pessoas a sensorização móvel torna-se uma área emergente para a investigação. Os sensores contidos nos smartphones serão o centro da próxima revolução nas redes sociais, monitorização ambiental global, saúde pessoal e comunitária, sistemas de transportes inteligentes e nos jogos e divertimento. Cada vez mais os smartphones são utilizados para recolha, processamento e distribuição de dados, vindo equipados com sensores como: sensor de luminosidade, acelerómetro, barómetro, giroscópio, magnetómetro, sensor de frequência cardíaca, GPS, sensor de proximidade e sensores de uso geral como microfone e câmara. Estes podem ser utilizados, por exemplo, para controlo do tráfego, monitorizando as condições da estrada e do trânsito. Os smartphones têm um imenso potencial, podendo melhorar substancialmente a vida das pessoas

(recorrendo aos sensores) e em determinados projetos pode até mesmo salvar vidas. Como em todos os sistemas que lidam com dados pessoais, existem questões como a segurança e privacidade, havendo ainda uma série de desafios e problemas que precisam ser resolvidos [55].

Podemos encontrar na literatura uma série de trabalhos que envolvem a monitorização de sinais de sensores. Como exemplo, em [10] é apresentada uma rede de sensores em tempo-real baseada em Bluetooth. Na área da medicina, temos um sistema de monitorização, armazenamento e acesso remoto de sinais de sensores [11] e ainda um sistema de monitorização dos sinais vitais adquiridos por sensores biométricos recorrendo a um smartphone Windows Mobile [12].

O MobiSens [53] tem como objetivo o reconhecimento da atividade humana baseada em dados recolhidos através dos sensores do smartphone. É uma plataforma de deteção móvel versátil para uma variedade de aplicações de sensorização móvel da vida real (registo do quotidiano, cuidados com idosos, saúde mental, relatórios do terreno, segurança orientada ao comportamento e sensorização social). Após 5 meses no Google Play, a plataforma MobiSens, conseguiu recolher 13993 horas de dados de 310 utilizadores.

Com uma visão para cidades mais inteligentes e seguras, o algoritmo iSafe [54] pretende prever futuros desastres, baseando-se em experiências do quotidiano do utilizador em tempo real. O iSafe foi testado utilizando dados relacionados com a criminalidade registados na base de dados do condado de Miami-Dade desde 2007 e dados recolhidos através do Yelp (rede geosocial onde os utilizadores efetuam *reviews* dos locais). Cada informação tem associado o tipo de crime (7 categorias), a data e a localização geográfica. Para garantir, por exemplo, que uma determinada rua problemática se encontra em segurança, podem ser recolhidas imagens através dos smartphones dos utilizadores. Foi desenvolvida uma aplicação em Android e um Plugin para o *browser*, tendo o utilizador acesso ao nível de segurança atribuído (1 a 5) ao local onde se encontra, assim como estatísticas de crime desse mesmo local.

Numa vertente focada no ciclismo, o Biketastic [52] apresenta como principal objetivo a eficiência dos trajetos efetuados, isto é, viajar da forma mais rápida possível

entre dois pontos com o menor número de paragens possível. A qualidade de um percurso pode ser influenciada pela segurança, eficiência ou apreciação em geral. Relativamente à segurança pode ser uma rota que evite ruas congestionadas optando por rotas com menos trânsito. Quanto à apreciação em geral (no decorrer de um percurso) pode ser uma rota que passa por determinados pontos de interesse (como por exemplo monumentos), partilhados através de conhecimentos pela comunidade. O Biketastic visa facilitar essa troca de conhecimentos entre os ciclistas, criando uma plataforma onde os participantes possam partilhar as suas rotas e experiências. Os dados são recolhidos através dos sensores integrados no smartphone. Através do GPS são recolhidas informações como a velocidade, latitude e longitude, através do acelerómetro e do microfone é possível medir a rugosidade do terreno e o nível de ruído sonoro.

2.5.1 BikeNet

O BikeNet é um sistema monitorização para mapear a experiência do ciclista que utiliza uma série de sensores embutidos na bicicleta para coletar os dados sobre as rotas do ciclista e disponibiliza um portal Web que permite a partilha de informação em tempo real ou acedendo a informação armazenada num servidor [7].

Pode-se dizer que a nível de hardware, este se encontra dividido em três partes:

- Sensores móveis, que se encontram colocados no ciclista e na bicicleta;
- Ponto de Acesso de Sensores (SAP - Sensor Access Point), responsáveis por comunicar com o servidor, transferindo os dados recolhidos e fazendo pedidos de informação solicitados pelo ciclista. No caso de um SAP móvel (Nokia N80) o ciclista poderá ter uma comunicação contínua com o servidor se assim o pretender.
- Servidor, onde se encontra o poder computacional e o armazenamento de todos os dados referentes a cada ciclista.

Como se pode observar na Figura 6, o BikeNet utiliza um paradigma de rede oportunista. Os sensores coletam dados ambientais e do ciclista ao longo do percurso,

sendo que uma tarefa a nível aplicacional com necessidade de contato com o servidor ou um upload de dados ocorre quando os sensores entram num raio abrangido por um SAP estático ou através de um SAP móvel ao longo da rota. O transporte dos dados via “muling” pode ocorrer quando os ciclistas se encontram dentro da mesma faixa de rádio, e pode ou não ser ativado pelo ciclista através das preferências. A comunicação entre o SAP e o servidor é feita via TCP/IP para os SAPs estáticos e via GPRS/GSM para os SAPs móveis.

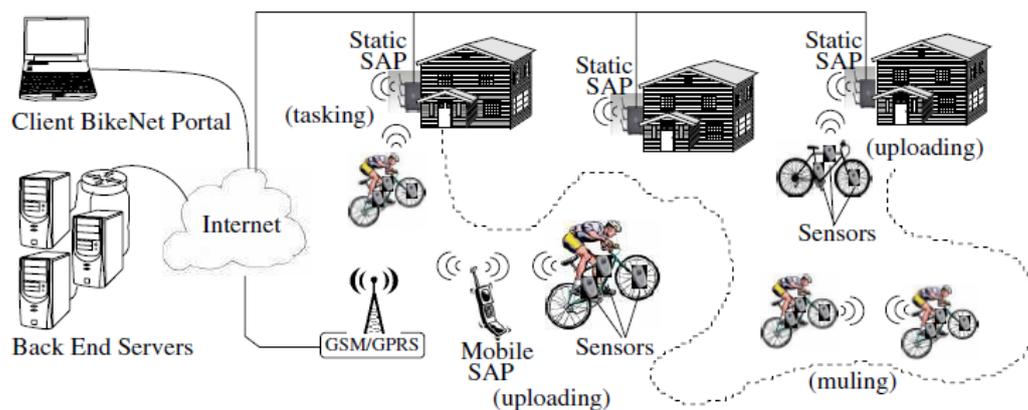


Figura 6 - Visão global do sistema BikeNet [7].

Para que o sistema funcione de forma correta e com medições mais precisas é necessária uma calibração para cada bicicleta. Quando o ciclista pretende efetuar uma rota, depois de estar completamente equipado, são estabelecidas ligações com todos os sensores necessários para atender as especificações das preferências do utilizador e um LED mostra o estado “Ready” no dispositivo transportado pelo utilizador. Pressionando um botão aquando do estado “Ready”, é enviada uma mensagem de início de transmissão, indicando o início da viagem, e os sensores começam a coletar os dados.

O ciclista, através de um perfil de preferências, pode personalizar que dados são recolhidos pelo sistema, quando são recolhidos, onde são recolhidos e em que condições são recolhidos. Podem existir dados que estão em constante envio e outros que apenas são enviados acontecendo determinado evento.

O sistema recolhe e armazena dados como a velocidade atual, velocidade média, distância percorrida, calorias queimadas e ainda dados mais específicos, tais como informação relativa à inclinação, batimento cardíaco, cadência, resposta galvânica da pele (para medir níveis de stress), níveis de poluição, níveis de alérgenos, níveis de ruído e rugosidade do terreno, todos eles referenciados pelo tempo e localização. Todos estes dados são recolhidos pelos vários sensores que partilham um canal IEEE 802.15.4 comum. É possível observar todos os sensores intervenientes nessa recolha na Figura 7.

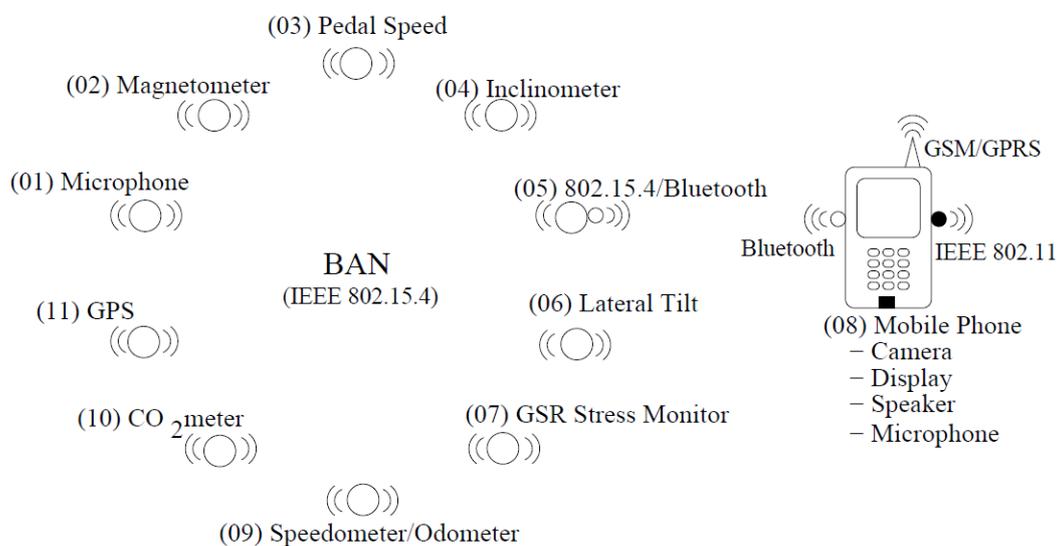


Figura 7 - Representação lógica da Bicycle Area Networking (BAN) [7].

Quanto ao portal Web, como se pode observar na Figura 8, contém gráficos do desempenho do ciclista, o mapa com a rota efetuada e os dados recolhidos pelos sensores. Através do portal Web são permitidas ainda *queries* em tempo real usando a conexão GPRS do Nokia N80 (se o ciclista estiver equipado com o mesmo), *queries* sobre a localização de um ciclista, capturar uma imagem através da câmara no Nokia N80 e ainda obter uma amostra de áudio do microfone.

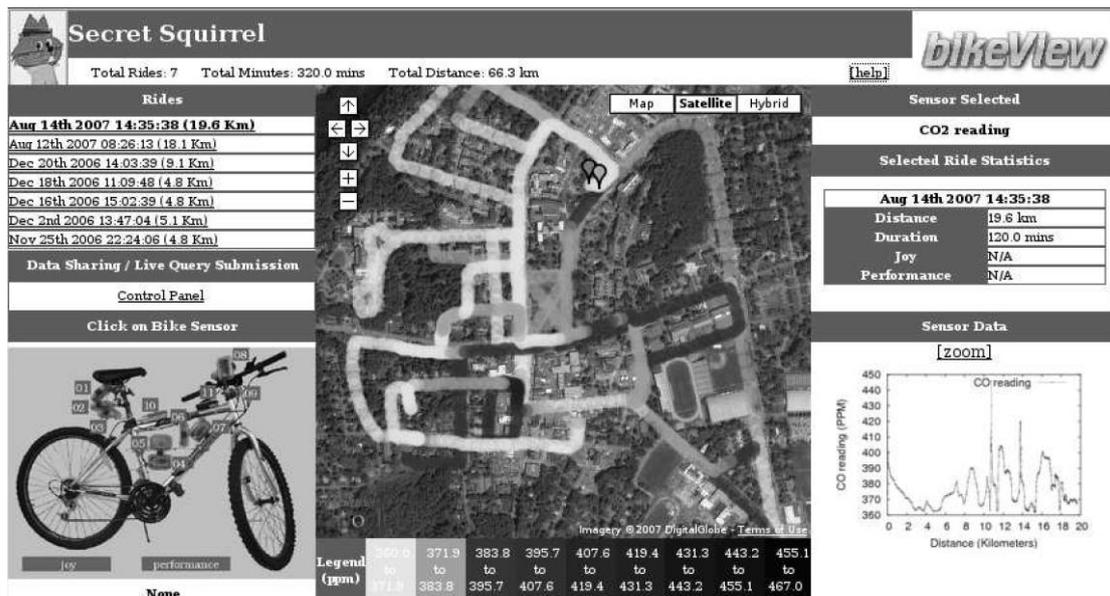


Figura 8 - Portal Web para consulta de dados e submissão de *queries* [7].

2.5.2 Endomondo

Lançado em Setembro de 2008, o Endomondo [3] é um software usado por muitos e que tem vindo a aumentar o número de utilizadores, mas não se restringe apenas ao ciclismo, dispondo de apoio para muitos outros desportos. A sua visão é tornar os diferentes desportos mais emocionantes e unir um grande grupo de pessoas fisicamente ativas. Monitorizando o seu progresso e lançando desafios a outros utilizadores, o utilizador vai assim buscar uma motivação extra.

O Endomondo utiliza o GPS para a localização e os recursos das redes sociais para uma maior motivação. Usando o GPS, esta aplicação rastreia a rota, distância total, distância percorrida, altitude atual, altitude mínima, altitude máxima, total de subida, total de descida, duração da rota, consumo de calorias, hidratação, velocidade atual, velocidade média, velocidade máxima e ritmo médio em minutos por quilómetro. Providência ainda um *feedback* de áudio durante o desempenho, e permite continuar a receber chamadas enquanto o utilizador se desloca, colocando a aplicação em pausa. Com o recurso às redes sociais, é possível adicionar amigos e seguir o seu histórico de atividades.

Esta aplicação permite ainda a criação de eventos e a adesão a eventos criados por outros utilizadores ou até mesmo por grandes marcas, como por exemplo a Samsung, que recentemente criou um evento em que oferecia um smartphone da sua marca para os dez utilizadores que utilizassem mais tempo a aplicação durante o exercício.

Podem ser guardados percursos através de uma rota efetuada pelo utilizador, podendo outros utilizadores seguir essa mesma rota e até tentar bater o *record* atual em termos de tempo.

Quanto à aplicação para Android, que foi a testada e que apresenta especial interesse para o desenvolvimento deste projeto, é colocado o foco apenas no ciclismo nesta secção. Sendo assim, quando o utilizador entra com a sua conta, tem disponível o menu, como se pode observar na Figura 9.

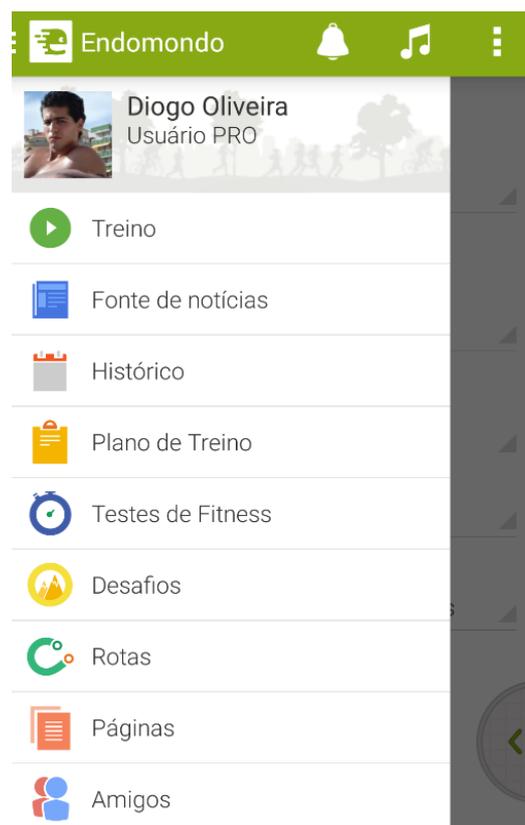


Figura 9 - Menu principal disponível ao utilizador na aplicação Endomondo para Android.

Escolhendo a opção de treino, é possível escolher o tipo de exercício pretendido e os vários parâmetros a serem exibidos ao utilizador no ecrã do seu smartphone.

Iniciando a sua atividade, o utilizador tem a opção de visualizar apenas os parâmetros escolhidos (Figura 10 à esquerda) ou o mapa do seu percurso com 2 parâmetros à sua escolha (Figura 10 à direita).

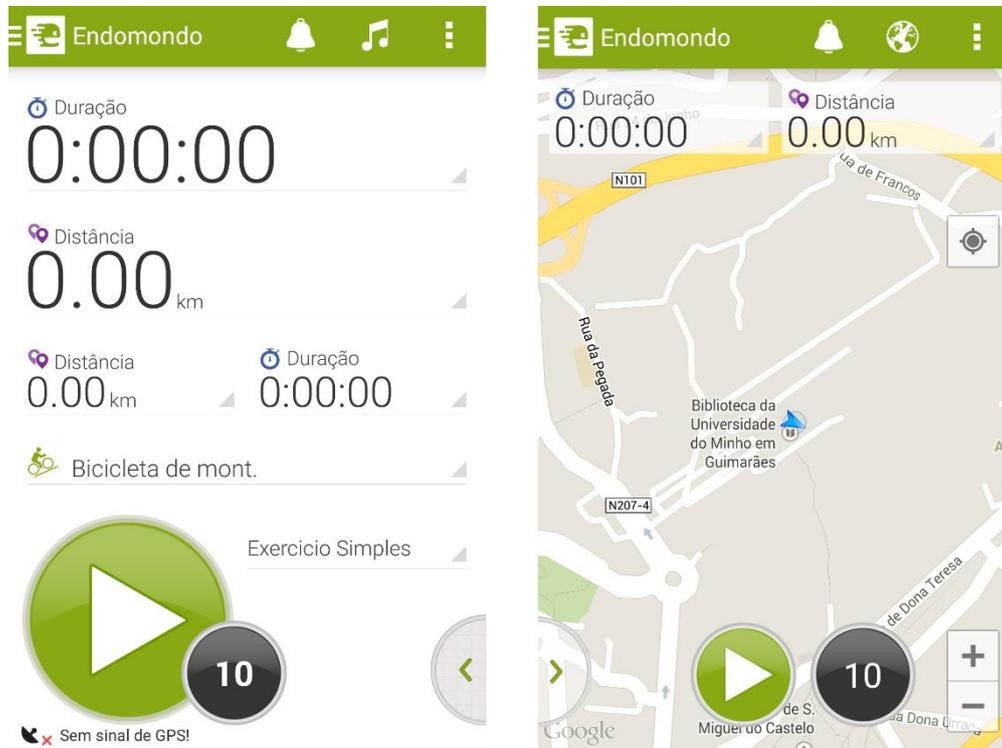


Figura 10 – Diferentes opções de visualização no decorrer do treino.

Consultando o histórico, o utilizador pode observar todas as informações disponíveis referentes à sua rota (Figura 11 à esquerda). Clicando sobre o mapa, obtém informação pormenorizada ao longo do percurso (Figura 11 à direita), tal como: o tempo decorrido até aquele momento, a distância percorrida, a velocidade a que se encontrava e a altitude. É possível ao utilizador mover-se ao longo do gráfico, obtendo assim as informações relativas a outro ponto do mapa. É possível ainda observar o tempo total e o tempo parcial por cada quilómetro, com informação do quilómetro onde foi mais lento e onde foi mais rápido, selecionando-se o separador voltas que se encontra a seguir ao separador zonas de RC (referência cardíaca). Esta funcionalidade é muito útil se o utilizador efetuar várias vezes o mesmo trajeto, pois permite-lhe assim saber onde poderá melhorar o seu desempenho.

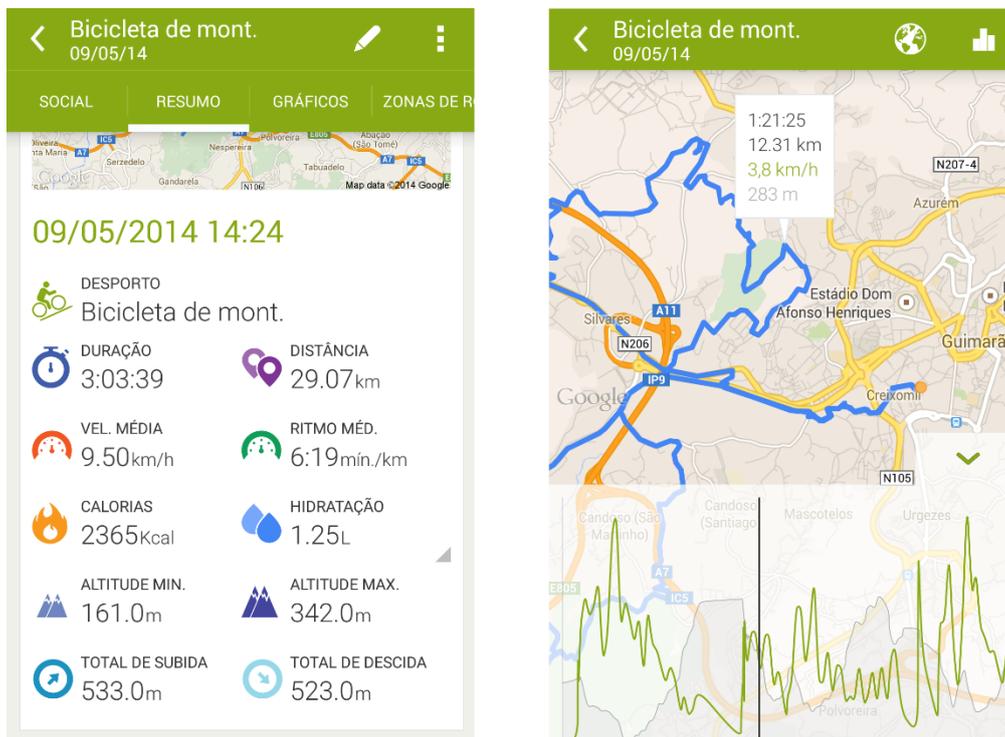


Figura 11 – Consulta de uma rota efetuada no histórico do utilizador.

O utilizador tem ainda disponível uma página Web onde pode consultar o histórico de todos os seus treinos, estatísticas, planos de treino, todos os desafios disponíveis, desafios a que aderiu, criar um desafio, consultar rotas guardadas por outros utilizadores para que possa tentar bater o recorde, aceder aos eventos e a uma lista com os seus amigos, para que possa ver o histórico deles, e ainda tem opção de aceder e alterar o seu perfil.

A Figura 12 apresenta uma consulta de uma rota do histórico do utilizador, onde se pode observar um sumário de todo o percurso, incluído até o estado do tempo, que não se encontra disponível na aplicação Android, o traçado da rota no mapa recorrendo ao Google Maps e, por fim, um gráfico com a velocidade e a altitude ao longo da distância ou do tempo, conforme selecionado, onde o utilizador pode mover um ponto de referência que vai disponibilizando informação ao longo do percurso, tal como: distância, duração, velocidade e altitude. Ao mesmo tempo, é mostrado um

ponto vermelho no mapa que simboliza o local onde o utilizador se encontrava relativamente a essa informação.

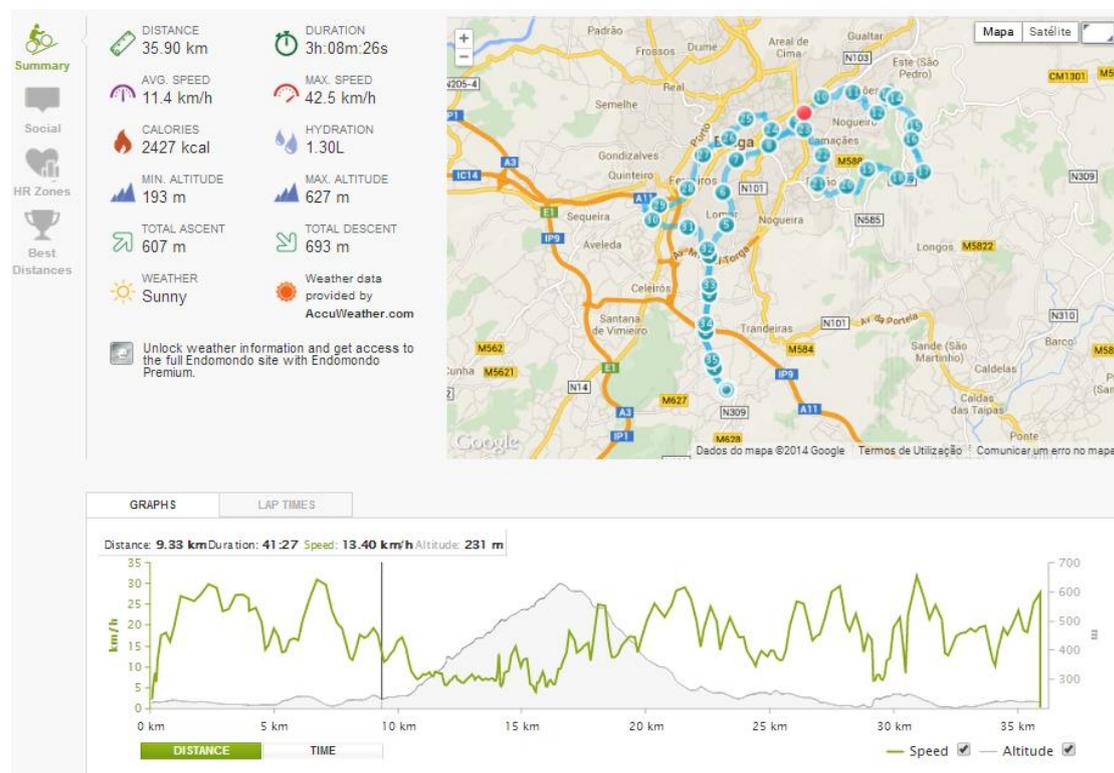


Figura 12 - Consulta de uma rota do histórico de um utilizador.

Oferecendo funcionalidades semelhantes, há ainda o Runtastic, o Strava e o Sports Tracker.

2.5.3 Roadroid

O Roadroid [8] é um sistema de monitorização das condições da estrada, recorrendo ao uso de um smartphone, baseando-se em resultados de alguns anos de investigação pela administração de estradas nacionais Sueca em cooperação com os engenheiros e investigadores do Roadroid.

Consiste numa aplicação, que capta as vibrações da estrada através do acelerómetro do smartphone e utiliza o seu GPS para o posicionamento, à semelhança da aplicação desenvolvida nesta dissertação. As vibrações são analisadas 100 vezes por segundo, e dessas 100 amostras resultantes é calculado um valor que é guardado

juntamente com as coordenadas GPS. Como a maioria dos smartphones incorpora uma câmara de boa qualidade, a aplicação faz uso da câmara para tirar fotos da estrada, que são, em conjunto com as coordenadas GPS, colocadas no mapa da página Web, funcionalidade útil para confirmação da análise feita, mas inexistente na aplicação desenvolvida nesta dissertação. Os dados são armazenados no smartphone e depois comprimidos e transferidos via Wi-Fi ou 3G para um servidor Web na nuvem da Amazon. Na aplicação desenvolvida nesta dissertação o processo é o mesmo, sendo que, os dados não são comprimidos e o servidor não se encontra na nuvem (servidor dedicado).

Na página Web os dados podem ser monitorizados num mapa, onde existem quatro cores para definir as condições da estrada: Verde para Bom, Amarelo para Satisfatório, Vermelho para Insatisfatório e Preto para Mau, pode-se observar um exemplo na Figura 13. Os dados são agregados por definição em secções de 100 metros, podendo assim ser facilmente transferidos para uma melhor análise, por exemplo, criando gráficos em Excel.

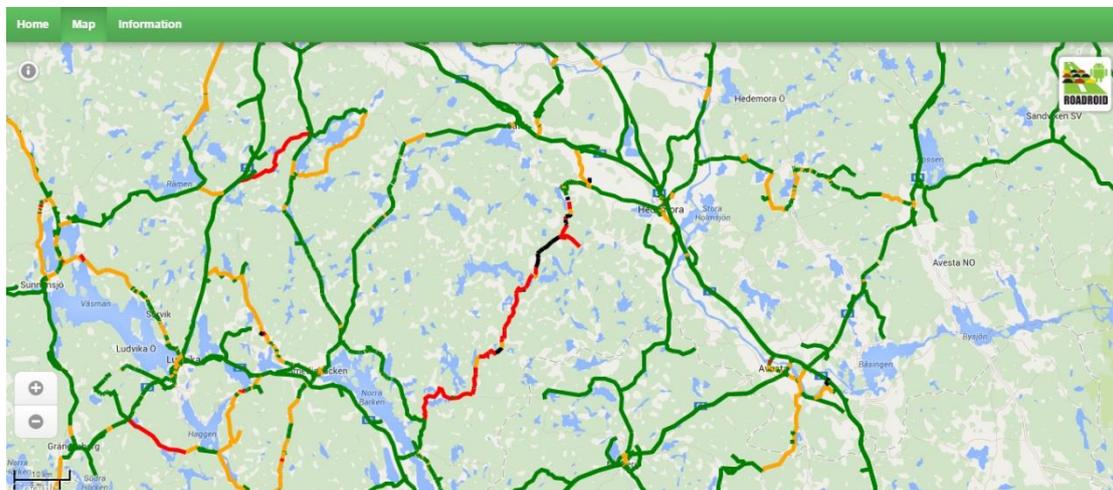


Figura 13 – Exemplo de visualização do mapa na página Web [9].

Como referido em [34] existem quatro classes genéricas para os métodos de medição da rugosidade das estradas:

- Classe 1 – Perfis de precisão;
- Classe 2 – Outros métodos de perfil métrico;

- Classe 3 – Estimativas do índice de rugosidade internacional (IRI) através de equações de correlação;
- Classe 4 – Avaliações subjetivas.

Um smartphone encontra-se na classe 2/3 e pode fornecer dados de boa qualidade relativos à rugosidade para um servidor durante 24 horas sem custos elevados, em contraste com um cliente de software caro com dados “exatos” de classe 1, e vem preencher uma lacuna entre a classe 1 e a classe 4.

Foram feitos testes recorrendo a diferentes smartphones e diferentes carros, cinco vezes em velocidades diferentes (10, 40, 60, 80, 100 e 120 km/h), para entender o funcionamento da aplicação, denotando-se que havia uma diferença no tipo de veículo, especialmente a velocidades baixas, surgindo a necessidade de implementar um modelo para calcular a influência da velocidade para 3 tipos de classes de carros. No uso de diferentes smartphones, foi notória uma diferença nos dados obtidos dos sensores, obrigando assim a um processo de calibragem prévio. Para que o teste resulte perfeitamente, os smartphones tem de estar num suporte bem fixo e de preferência que deixe a lente da câmara apontar para a estrada.

Quanto ao uso do Roadroid em bicicletas, pode ser usado o mesmo sistema. O teste foi efetuado numa bicicleta com reboque, sendo montado o smartphone firmemente sobre o eixo da roda, como se pode observar na Figura 14. A aplicação desenvolvida no âmbito desta dissertação tem uma finalidade um pouco diferente, pois a intenção é reconhecer o tipo de terreno e não uma avaliação do estado do terreno.



Figura 14 – Modelo de bicicleta testado [8].

Teve de ter sido em atenção o IRI, e foi desenvolvida uma correlação entre os valores de condição da estrada medidos e o IRI, que foi denominada ROADROID INDEX (RI). Existiu um trabalho contínuo de forma a melhorar a flexibilidade e a escalabilidade, para que pudessem ser capazes de adicionar dados a partir de várias medições ao longo do tempo e comparar os resultados ao longo do tempo de uma forma flexível.

3. Implementação do sistema proposto

Nesta secção é feito um levantamento dos requisitos para a implementação do sistema, apresentada a arquitetura do sistema, e descritos quais os procedimentos para a implementação do servidor, do serviço Web e da BD. São expostos todos os processos envolvidos no desenvolvimento da aplicação móvel e da página Web, referenciando cada uma das suas funcionalidades. É ainda abordada a utilização da tecnologia NFC, e o que traz de útil à aplicação desenvolvida.

3.1 Requisitos

Os requisitos podem ser divididos entre requisitos do sistema e requisitos do utilizador, sendo os últimos considerados requisitos de alto nível, ou seja, estão relacionados com funcionalidades disponíveis ao utilizador.

Quanto aos requisitos do sistema desenvolvido, estes encontram-se divididos entre requisitos funcionais e requisitos não funcionais.

Requisitos funcionais:

- Recolha de dados através do smartphone com recurso ao sensor de GPS, acelerómetro e barómetro;
- Armazenamento dos dados na base de dados (BD) do smartphone;
- Permitir o acesso ao servidor a partir da Internet;
- Envio dos dados para o servidor;
- Armazenamento dos dados na BD do Servidor.

Requisitos não funcionais:

- Garantir a segurança, privacidade e integridade dos dados;
- Garantir a disponibilidade do serviço a vários utilizadores em simultâneo;

- Barrar o acesso aos conteúdos sem que haja autenticação;
- Otimizar o consumo de recursos no envio de dados;
- Minimizar os consumos de energia.

Relativamente aos requisitos do utilizador, estes também se encontram divididos entre requisitos funcionais e requisitos não funcionais.

Requisitos funcionais:

- Registo e autenticação;
- Terminar sessão;
- Dar permissão para que os amigos sigam o seu histórico;
- Pedir para que um amigo lhe dê permissão para que siga o seu histórico;
- Remover amigos;
- Criar eventos;
- Adicionar eventos à sua agenda;
- Remover eventos da sua agenda;
- Visualização do seu histórico;
- Visualização do histórico dos seus amigos;
- Localizar amigos;
- Visualização de gráficos.

Requisitos não funcionais:

- Fluidez entre *layouts*;
- Interface de fácil navegação;
- Sincronização do histórico em vários dispositivos;
- Otimização do consumo de dados no acesso ao servidor.

É apresentada de seguida a arquitetura do sistema e explicado em detalhe cada um dos componentes.

3.2 Arquitetura do sistema

A solução para o sistema desenvolvido (Figura 15) consiste nos seguintes componentes:

- Um servidor Web capaz de atender todos os pedidos efetuados pelo utilizador quer seja pelo smartphone, quer seja através de uma página Web, integrando uma BD própria onde se encontra armazenada toda a informação relevante ao funcionamento do sistema.
- Um serviço Web externo que permite obter a pressão ao nível do mar para o cálculo da altitude e ainda a temperatura onde o utilizador se encontra, através de um pedido HTTP, efetuado unicamente através do smartphone. Há ainda um serviço Web externo para que seja efetuado o download do mapa.
- Uma página Web em HTML, JavaScript e CSS, capaz de correr em qualquer dispositivo (PC, tablet ou smartphone) e comunicar com o servidor para eventuais pedidos.
- Um smartphone com uma BD própria em SQLite (relativa à aplicação), capaz de comunicar com os satélites utilizando o sensor de GPS do smartphone e obter informação georreferenciada, podendo comunicar com o servidor, para que seja armazenada na BD (do servidor) toda a informação relevante recolhida por este, ou sejam efetuados pedidos de informação que já se encontrem armazenados na BD.

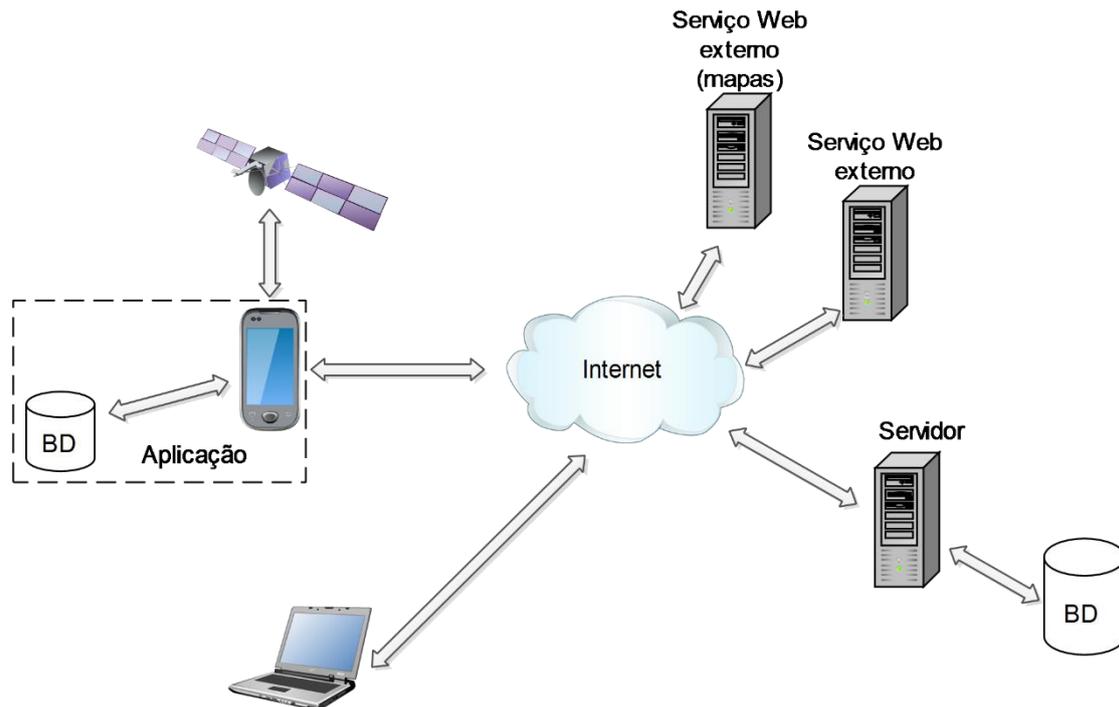


Figura 15 - Diagrama global do sistema.

Nas próximas secções será explicado com maior detalhe cada um dos componentes envolvidos no diagrama global do sistema, assim como todas as opções tomadas no decorrer do seu desenvolvimento.

3.3 Servidor

Como servidor HTTP foi escolhido o servidor Apache devido a uma fácil instalação e configuração. Foi ainda utilizada a aplicação gratuita XAMPP, que possibilita, além do acesso a um servidor Apache, o acesso a um servidor MySQL. Na Figura 16 é possível observar um diagrama global do servidor, onde o cliente acede ao servidor Apache, que por sua vez estabelece contato com o motor PHP. Através do motor PHP é feito um pedido ao SGBD MySQL, consultando este a BD.

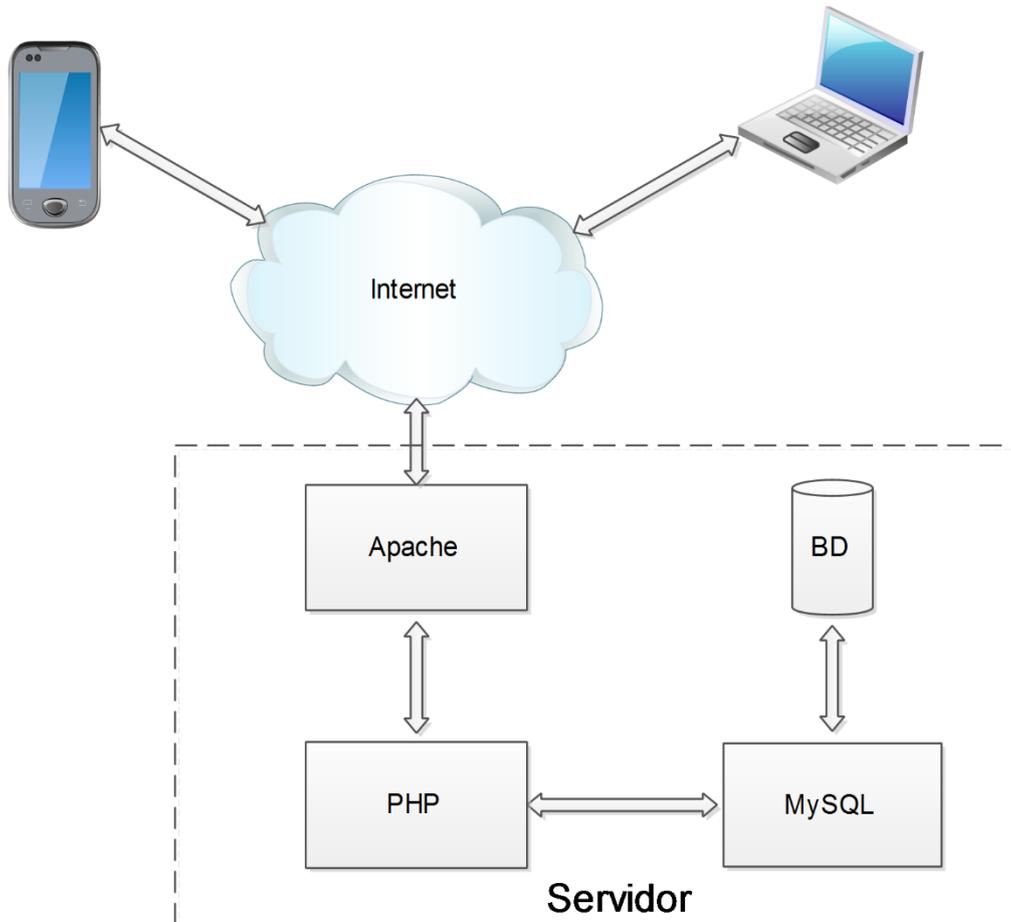


Figura 16 - Diagrama global do servidor.

3.3.1 Serviço Web

Para a comunicação entre o smartphone/computador e o servidor, optou-se por um serviço simples e viável (Figura 17). No exemplo do smartphone, é utilizado o protocolo HTTP para a transferência dos dados (HTTP POST em JSON). É efetuado o pedido ao servidor contendo o URL (http://193.136.12.191/android/ws_geobike/WS_Geobike.php/método) com os parâmetros a serem interpretados, onde o último campo do URL é o método a ser executado. O servidor executa o pedido acedendo ao método contido no ficheiro `WS_Geobike.php` e executa uma ação com os parâmetros recebidos, que normalmente inclui sempre uma consulta à BD. De seguida devolve uma resposta, em JSON, enviada por HTTP para o smartphone (Apêndice B – Protocolo de comunicação).

A escolha dos dados serem enviados com o formato JSON deve-se ao facto de consumir pouca largura de banda e exigir pouco processamento. Nos dispositivos móveis trata-se de uma característica fundamental para que se poupe ao máximo um recurso tão importante como a bateria e se possa reduzir o consumo de dados móveis (3G).

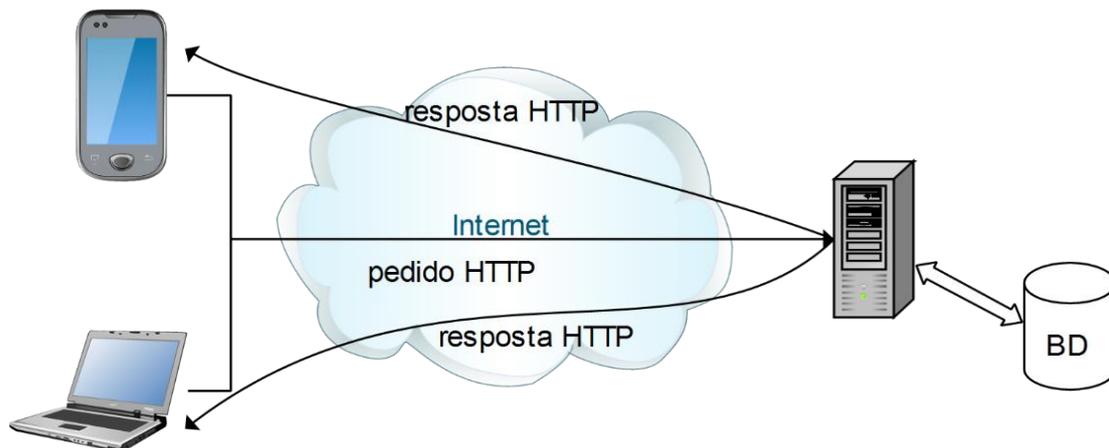


Figura 17 – Diagrama do serviço Web.

3.3.2 Base de dados

A base de dados contida no servidor é onde se encontra armazenada toda a informação para o funcionamento do sistema. Como SGBD optou-se pelo MySQL como referido em 2.2. Na Figura 18 encontra-se o diagrama entidade-relação (E-R) desenvolvido para a BD, sendo este constituído pelas seguintes entidades:

- utilizador – onde se encontra toda a informação relativa ao utilizador;
- localiza_utilizador – contém a informação para que seja possível obter a última localização do utilizador;
- trajeto – possui a informação relativa a cada trajeto efetuado pelo utilizador;
- localização – contém toda a informação armazenada no decorrer de um trajeto;
- utilizadores_bicicleta – onde se encontra a informação sobre que utilizador utilizou qual bicicleta;

- bicicleta – contém a informação relativa a uma bicicleta em cada instante de tempo;
- utilizador_evento – onde se encontra a informação sobre quais os eventos a que o utilizador se encontra associado;
- evento – contém a informação sobre o evento;
- utilizador_amigos – onde se encontra a informação sobre quais os amigos associados ao utilizador;
- amigos – contém os emails dos amigos para que se possa aceder posteriormente a informações relativas a esse amigo;
- cardio – possui a informação recolhida do sensor cardíaco.

Relativamente á entidade cardio e à entidade bicicleta, os seus atributos deveriam se encontrar agregados na entidade localização, se estes fossem recolhidos através da mesma aplicação. Porém, uma vez que as aplicações que recolhem esses dados são diferentes e pertencem a trabalho realizado por outros investigadores, decidiu-se criar as entidades em separado, para que eles possam utilizar a mesma BD. Os valores dos atributos associados a essas entidades, podem facilmente ser atribuídos a um trajeto, uma vez que cada entidade possui a data, hora e email do utilizador.

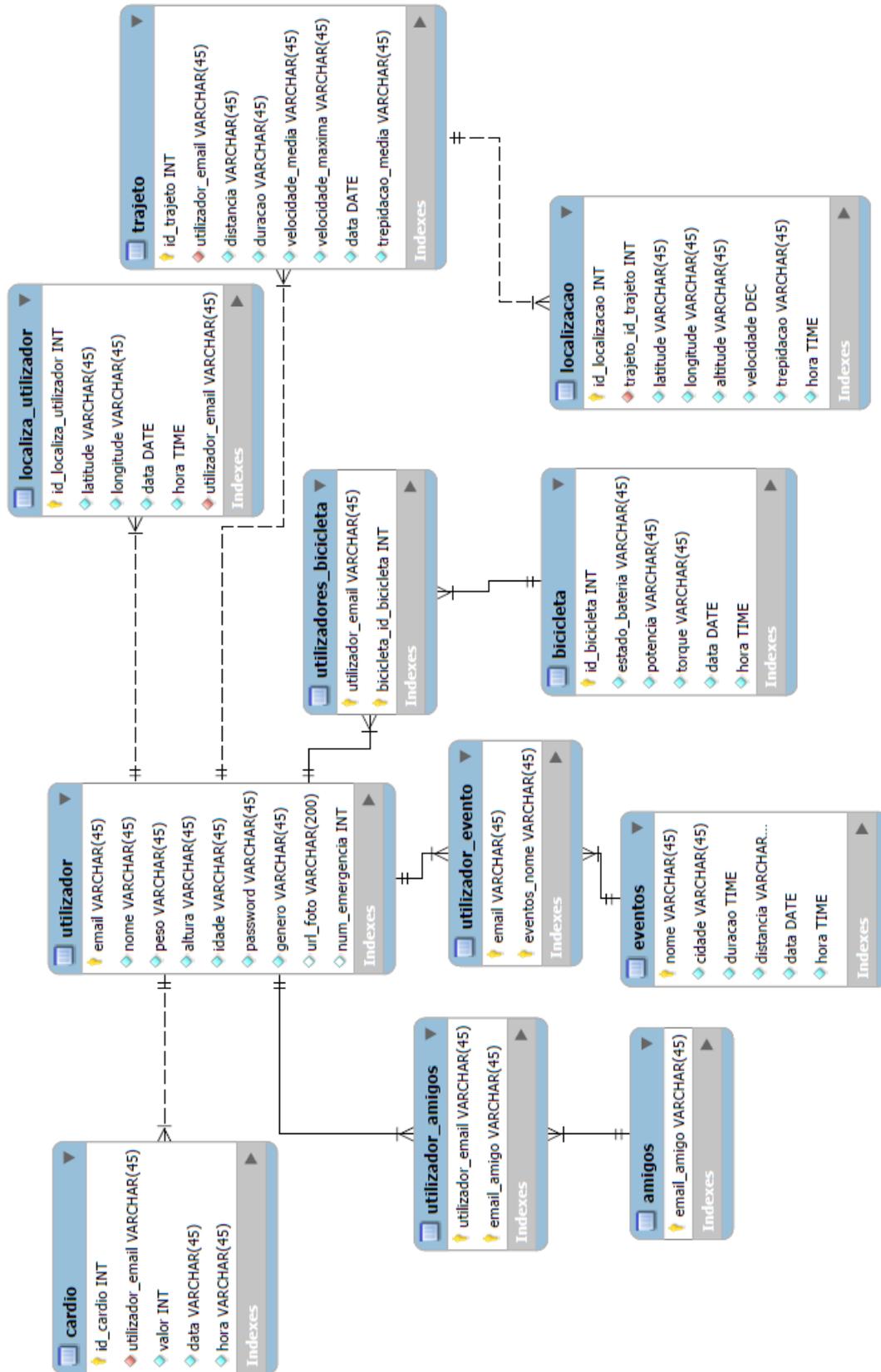
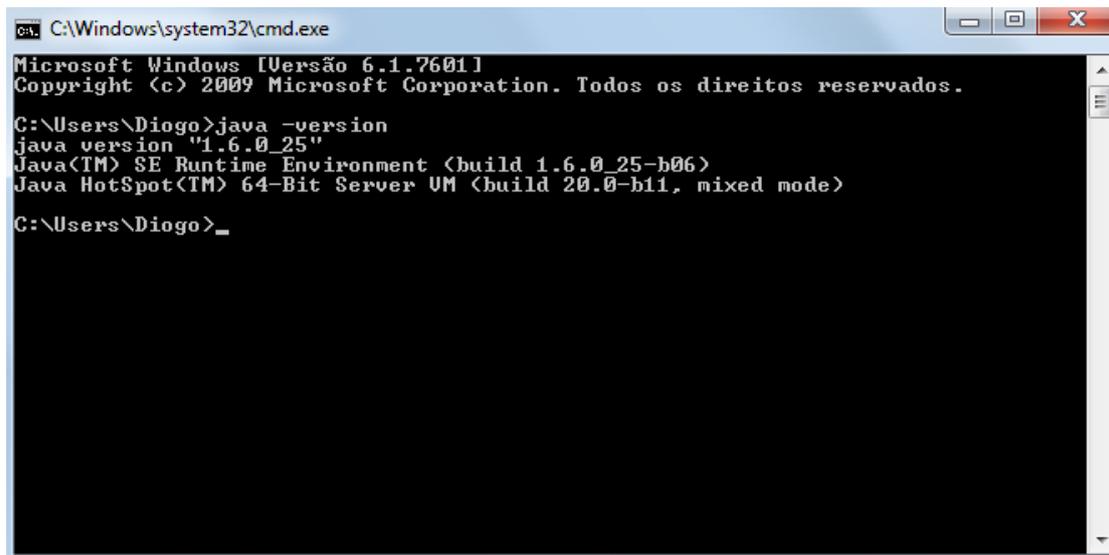


Figura 18 – Diagrama E-R.

3.4 Software

Relativamente à criação da aplicação Android as ferramentas utilizadas foram:

- Java – como é utilizada a linguagem de programação Java, foi necessário verificar a sua existência na nossa máquina, como podemos observar na Figura 19, sendo que para o JDK 6 é preciso utilizar no mínimo a versão Gingerbread do Android (API 9);



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Diogo>java -version
java version "1.6.0_25"
Java(TM) SE Runtime Environment (build 1.6.0_25-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.0-b11, mixed mode)

C:\Users\Diogo>_
```

Figura 19 – Versão Java instalada na nossa máquina.

- Eclipse – Plataforma de desenvolvimento de software livre, baseada em Java;
- SDK Manager – Fornece as ferramentas e as APIs necessárias para o desenvolvimento de aplicações;
- ADT (Android Development Tools) – *Plugin* Android para o Eclipse, estendendo as suas capacidades, permitindo assim criar rapidamente um projeto Android;
- AVD (Android Virtual Device) Manager – Fornece uma interface gráfica ao utilizador, onde este pode criar e gerenciar dispositivos Android virtualmente, sendo posteriormente requisitados pelo emulador do Android.

Relativamente à biblioteca, para que fosse possível o acesso a um mapa, foram analisados o OpenStreetMap [38] e o Google Maps [39]. No Google Maps qualquer programador que pretenda desenvolver uma aplicação recorrendo ao seu mapa terá de obedecer a determinadas políticas de utilização; já no OpenStreetMap, o programador é livre para utilizar toda a informação existente de uma forma mais permissiva. Devido ao facto de ser nativo do Android a escolha recaiu sobre o Google Maps API v2.

Quanto à criação da página Web foi utilizada a plataforma de desenvolvimento de software livre NetBeans IDE 8.0, com recurso à *framework* Bootstrap [40], que fornece um vasto conjunto de modelos HTML, CSS, formulários, botões, navegação e outros componentes de interface bem como extensões JavaScript opcionais.

Do lado do servidor foi utilizado o MySQL Workbench, para a criação da base de dados, e a ferramenta XAMPP, para criação do servidor Apache e do servidor MySQL. Para garantir um acesso remoto ao servidor sem a necessidade de estar permanentemente no mesmo local em que este se encontrava sempre que fossem precisas alterações, foi utilizada a aplicação TeamViewer 9, sendo assim possível efetuar qualquer configuração, ou transferência de ficheiros de configuração remotamente.

3.5 Hardware

Durante o desenvolvimento do sistema foi utilizado um smartphone Nexus 5, permitindo o acesso aos seguintes sensores: GPS (para recolha da latitude e longitude), acelerómetro (para medição e cálculos relativos à trepidação) barómetro (para obtenção da pressão atmosférica, necessária para o cálculo da altitude) e NFC. A escolha de um smartphone desenvolvido pela própria Google permitiu a utilização da última versão do SO Android lançada até ao momento (4.4.4 KitKat) de uma forma estável. Quanto ao servidor foi utilizado um portátil Sony Vaio para esse efeito.

3.6 Aplicação móvel

Nesta secção são abordados todos os componentes relacionados com a aplicação móvel, incluindo as pré-configurações efetuadas antes do uso da aplicação, referentes à utilização da API v2 do Google Maps, o processo relacionado com a BD contida no smartphone, a sincronização com a BD do servidor, como é efetuado o registo e a autenticação de um utilizador e como a sua sessão é mantida, como é feita a navegação entre *Activities* para um caso específico e as diferentes opções que o utilizador tem ao seu dispor durante a utilização da aplicação.

3.6.1 Manifesto Android

Todas as aplicações desenvolvidas em Android têm obrigatoriamente um ficheiro *AndroidManifest.xml*. É neste ficheiro que se encontra informação essencial acerca da aplicação para o sistema Android. Encontram-se também declaradas todas as permissões e a mínima versão do Android que o dispositivo tem de satisfazer para executar a aplicação. Todas as *Activities* a serem executadas têm de estar declaradas, e uma *Activity* tem de ser declarada como sendo a principal. É neste ficheiro onde se encontra a *key* referente à nossa aplicação para obtenção do mapa da Google. Na Figura 20 é possível observar todas as permissões da aplicação desenvolvida, tais como: permissão de acesso à Internet, permitir o uso da API para download do mapa, do servidor do Google Maps, permitir que a API verifique o estado da conexão para determinar se os dados podem ser transferidos, permitir que a API armazene da cache para o armazenamento externo, permitir que a API utilize o GPS para determinar a localização do dispositivo e permissão de envio de SMS.

```

<permission
  android:name="com.example.maps.permission.MAPS_RECEIVE"
  android:protectionLevel="signature"/>
<uses-permission android:name="com.example.maps.permission.MAPS_RECEIVE"/>

<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<!-- The following two permissions are not required to use
  Google Maps Android API v2, but are recommended. -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.SEND_SMS"></uses-permission>

```

Figura 20 - Permissões da aplicação desenvolvida.

3.6.2 Google Maps

Para que fosse possível usufruir do mapa da Google, foi necessário primeiramente adicionar o Google Play Services, através do SDK Manager, ao Eclipse. Sendo a API key do Google Maps baseada numa forma abreviada do certificado digital da nossa aplicação, conhecida como *fingerprint* SHA-1, foi necessária a sua obtenção para assim ter o acesso à API key, como se pode observar na Figura 21, sendo necessário posteriormente adicionar a API key no ficheiro AndroidManifest.xml

Key for Android apps (with certificates)

API key: AIzaSyAzhzySBmxe9SKftPtvixGq6e7pR0CVkgQ
 Android apps: 5E:19:5D:B3:DC:09:99:1B:8E:C3:C5:70:18:10:74:45:75:7F:9D:DC;com.example.aplicacao
 Activated on: Mar 20, 2014 3:40 AM
 Activated by: diogodani85@gmail.com – you

Figura 21 - Key para o Google Maps.

A API do Google Maps para Android utiliza o OpenGL ES v2 para apresentar o mapa no dispositivo. Para que a sua visualização fosse possível, foi necessário adicionar o código da Figura 22 no ficheiro AndroidManifest.xml.

```

<uses-feature
  android:glEsVersion="0x00020000"
  android:required="true"/>

```

Figura 22 – Referência ao OpenGL ES v2 no AndroidManifest.xml.

3.6.3 Base de dados

Como referido em 2.2, foi utilizada a biblioteca SQLite para a elaboração da base de dados contida no smartphone, que é onde se encontra armazenada a informação relativa aos trajetos efetuados pelo utilizador. Na Figura 23 encontra-se o diagrama E-R desenvolvido para a BD do smartphone, sendo as entidades que o constituem: o trajeto, que contém a informação relativa a um trajeto efetuado pelo utilizador; e a localização, que contém toda a informação armazenada no decorrer de um trajeto.

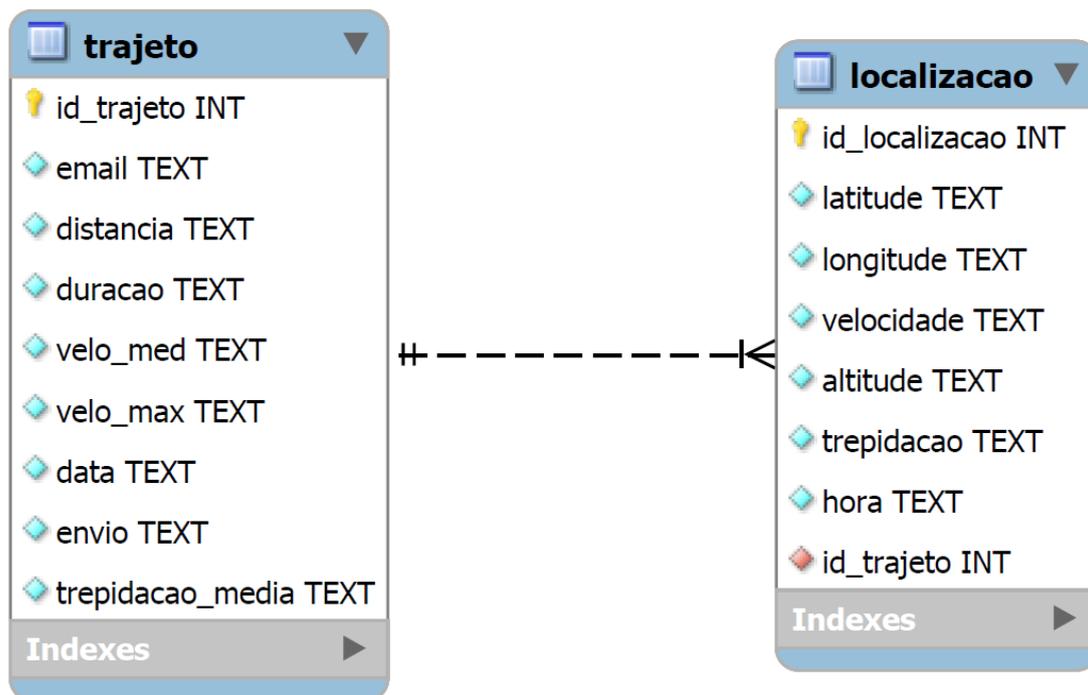


Figura 23 - Diagrama E-R SQLite.

Como se pode observar na figura, estas duas entidades são muito parecidas com as duas entidades do diagrama E-R do servidor; a ideia é facilitar a sincronização entre ambas as BDs. Na entidade trajeto pode-se observar um atributo diferente, o “envio”. Este funciona como uma *flag* podendo receber como valores zero ou um. No SQLite a forma de relacionar as duas entidades é feita de uma forma diferente, tendo de se declarar o atributo e de seguida referir que se trata de uma chave estrangeira, fazendo referência à entidade de onde ele é proveniente, neste caso o trajeto. É possível observar o código para a criação dessa relação na Figura 24.

```
private static final String CREATE_TABLE_LOCALIZACAO = "create table "  
+ TABELA_LOCALIZACAO + " ("  
+ ID_LOCALIZACAO + " integer primary key autoincrement, "  
+ LATITUDE + " text not null, "  
+ LONGITUDE + " text not null, "  
+ VELOCIDADE + " real not null, "  
+ ALTITUDE + " text not null, "  
+ TREPIDACAO + " text not null, "  
+ HORA + " text not null, "  
+ ID_TRAJETO + " integer, "  
+ " foreign key("+ID_TRAJETO + ") references " +TABELA_TRAJETO +"(" + ID_TRAJETO+"));
```

Figura 24 – Associar a entidade localizacao com a entidade trajeto.

3.6.3.1 Upload SQLite para MySQL

Como referido em 3.6.3, na entidade trajeto existe o atributo envio. Inicialmente este recebe o valor zero e, sempre que existe uma sincronização, é feito um *update* do valor para um, o que permite que na próxima sincronização sejam tidos em conta apenas as novas inserções que estão com o valor zero. Sempre que a aplicação é inicializada é verificado este atributo e, se existir algum com o valor zero, são devolvidos todos os atributos com esse id_trajeto e inseridos na BD do servidor. Assim fica já inserido o id_trajeto na entidade trajeto da BD do servidor, podendo de seguida inserir os valores associados aos atributos da entidade localizacao através do id_trajeto.

O utilizador tem à sua disposição duas formas de sincronizar os dados (Figura 25). A primeira é feita de forma automática – sempre que este inicialize a aplicação e tenha acesso à Internet através de Wi-Fi – e a segunda pode ser efetuada de forma manual (com recurso a um botão) – sempre que o utilizador pretender, utilizando o modo como este se encontra conectado à Internet (Wi-Fi/3G).

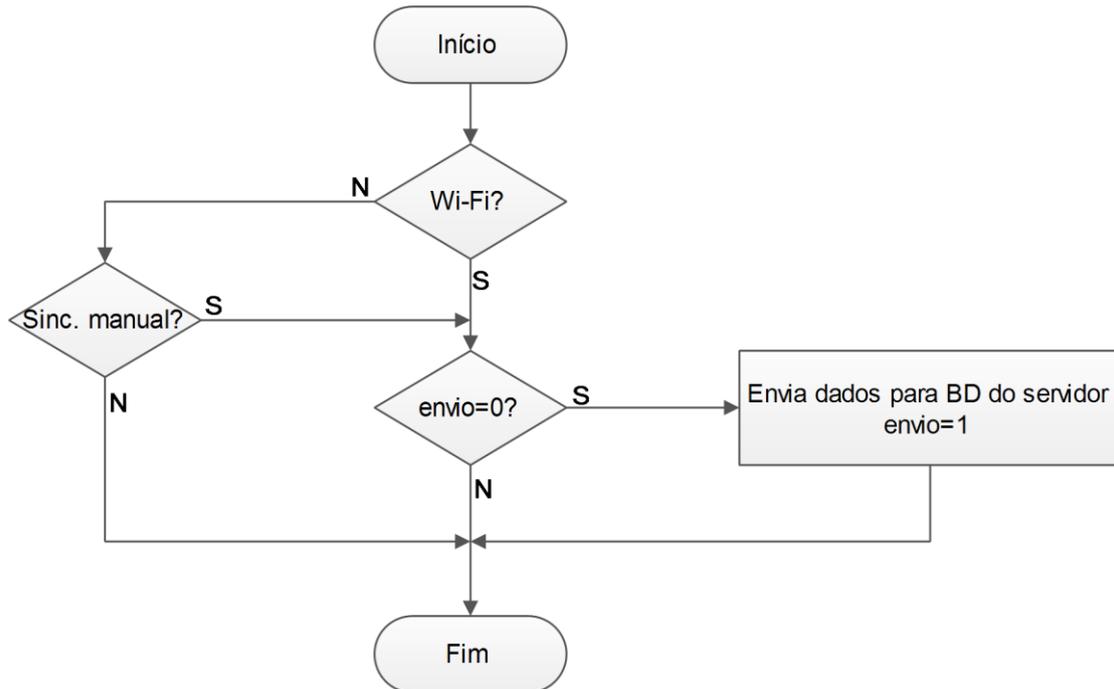


Figura 25 - Diagrama de blocos alusivo à sincronização SQLite com MySQL.

3.6.4 Registo e autenticação

Inicializada a aplicação, pela primeira vez, é apresentada a *Activity* da Figura 26 à esquerda. Quando o utilizador pretende efetuar um registo ou o login, primeiramente é verificada a conexão à Internet, pois em ambos os casos esta é necessária. Caso não exista, é apresentada uma mensagem para que este se conecte à Internet.

No caso do registo, e existindo a conexão à Internet, o utilizador tem acesso à *Activity* de registo (Figura 26 ao centro), onde todos os campos têm de ser devidamente preenchidos. De notar que, em alguns campos, o conteúdo tem de ser inserido num formato específico. No exemplo da data de nascimento, é apresentada ao utilizador uma janela de *pop-up* (Figura 26 à direita) para que este escolha a sua data de nascimento, e o valor adquire automaticamente o formato pretendido. Após pressionar o botão guardar, dá-se a verificação do formato do email. Caso este esteja num formato correto, é estabelecida uma conexão com o servidor para verificar se este não consta na BD. Caso este não se encontre registado é efetuada a inserção na BD e apresentada uma mensagem ao utilizador, notificando-o que foi efetuado o registo com sucesso e retornando-o para a *Activity* de login (Figura 26 à esquerda). Se

o email já se encontrar registado é exibida uma mensagem ao utilizador alertando-o do mesmo. O registo só é possível na aplicação para garantir que, se um utilizador se regista, é porque possui a aplicação.

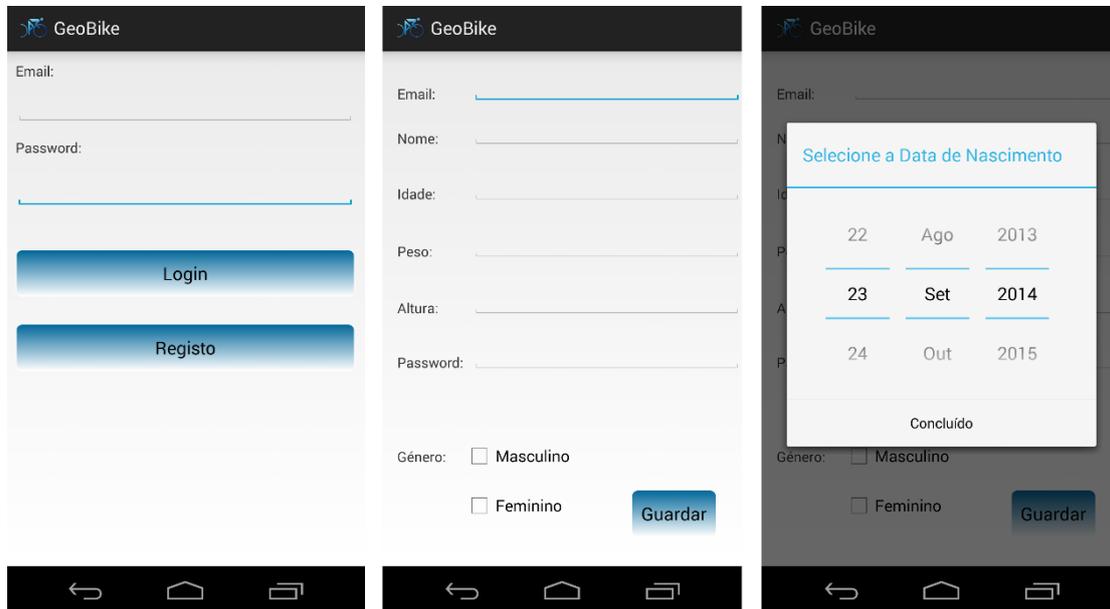


Figura 26 - Processo de registo.

Relativamente ao login, após introdução do email e da password, é efetuado um pedido ao servidor para verificar se a password introduzida corresponde ao email. Caso não corresponda é exibida uma mensagem ao utilizador alertando-o de a password ou o email estarem incorretos. No caso de o email e a password corresponderem é chamada a *Activity* principal da aplicação. Na Figura 27 é possível observar o diagrama de sequência referente ao registo do utilizador seguido de um pedido de autenticação.

Devido ao facto da requisição de um registo, deve ser salvaguardada a privacidade dos dados dos utilizadores. Assim, a password introduzida pelo utilizador é cifrada, não sendo possível ao administrador do sistema, ou a um possível intruso, a obtenção da mesma. Para a cifragem dos dados foi utilizado o algoritmo SHA-1, sendo suficiente para garantir um nível de proteção adequado sem que fosse introduzido um excessivo consumo de recursos ao dispositivo. É importante que os utilizadores estejam a par da implementação de mecanismos de segurança, para assim, atrair a sua confiança.

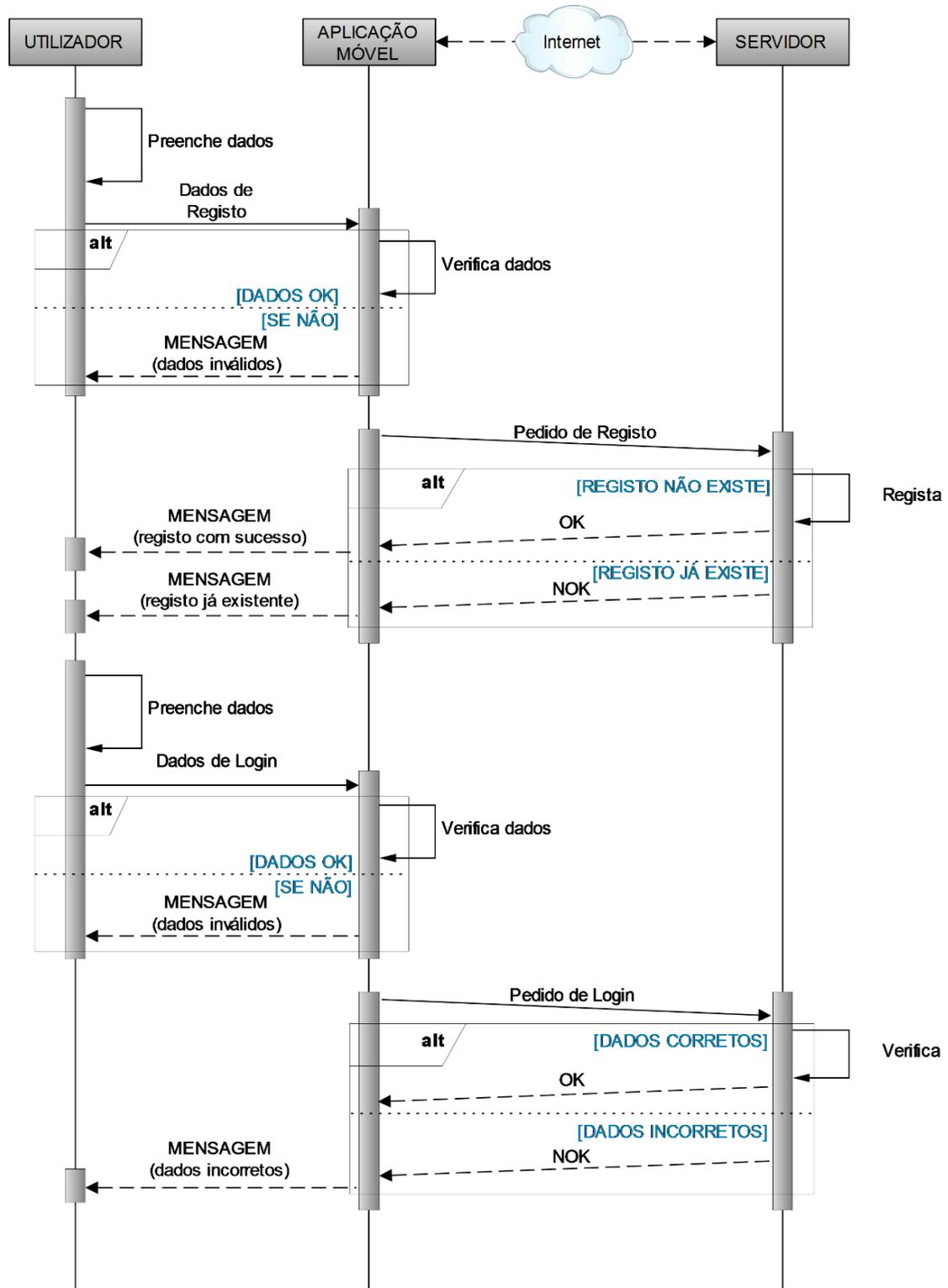


Figura 27 - Diagrama de sequência relativo ao registo e à autenticação.

3.6.4.1 Preservação da sessão iniciada

Para evitar que, cada vez que o utilizador iniciasse a aplicação, fosse pedido novamente o login foi criada uma classe `SessionManager`, que utiliza a classe `SharedPreferences` (Apêndice A – Principais classes Android), para que possam ser guardadas variáveis como o email e o estado da sessão e estas possam ser acedidas em qualquer *Activity*.

Caso o login seja efetuado com sucesso, é chamado um método que coloca a *flag* `IS_LOGIN` a *true* e guarda o email do utilizador numa variável. Sempre que se inicializa a aplicação, na *Activity* declarada no ficheiro `AndroidManifest.xml` como sendo a principal, é invocado um método que verifica o estado da *flag* `IS_LOGIN`. Se esta se encontrar a *true*, o programa mantém-se na *Activity* principal; caso seja *false*, é redirecionado para a *Activity* do login (Figura 28).

O utilizador tem a oportunidade de terminar a sessão a qualquer momento, sendo invocado um método que coloca a *flag* `IS_LOGIN` a *false*, limpa a variável que contém o email e redireciona-o para a *Activity* de login.

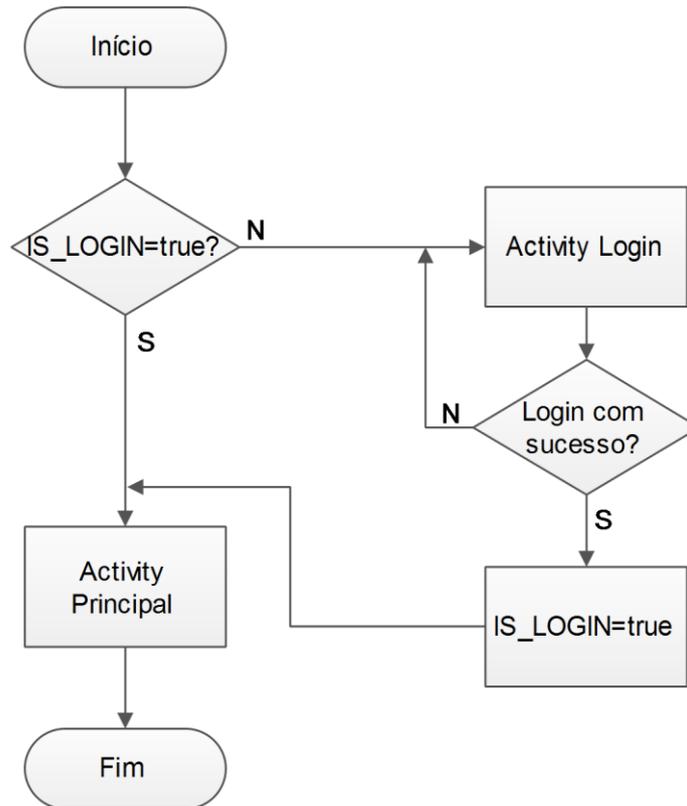


Figura 28 - Diagrama de blocos referente à preservação da sessão.

3.6.5 Navegar entre *Activities*

Para que a navegação entre as *Activities* fosse efetuada de uma forma fluida, adotou-se a abordagem da Figura 29. Inicialmente, o utilizador encontra-se na *Activity 1*, que se apresenta em primeiro plano; quando o utilizador inicializa a *Activity 2*, esta vem para o topo da pilha, passando para primeiro plano. A *Activity 1* permanece na pilha, mas é parada (o sistema mantém o estado atual da sua interface de utilizador); o mesmo acontece quando é inicializada a *Activity 3*, onde a *Activity 2* e a *Activity 1* permanecem na pilha paradas. Quando o utilizador retrocede, a *Activity 3* é destruída e a *Activity* anterior (*Activity 2*) é retomada (o estado anterior da sua interface de utilizador é restaurado). Pode-se dizer, então, que o seu funcionamento é do tipo LIFO (*last in, first out*), onde o último a entrar é sempre o primeiro a sair. Este exemplo encontra-se presente na parte do histórico do utilizador na aplicação desenvolvida (secção 3.6.6.3), onde o utilizador tem oportunidade de passar, por precisamente, três *Activities*.

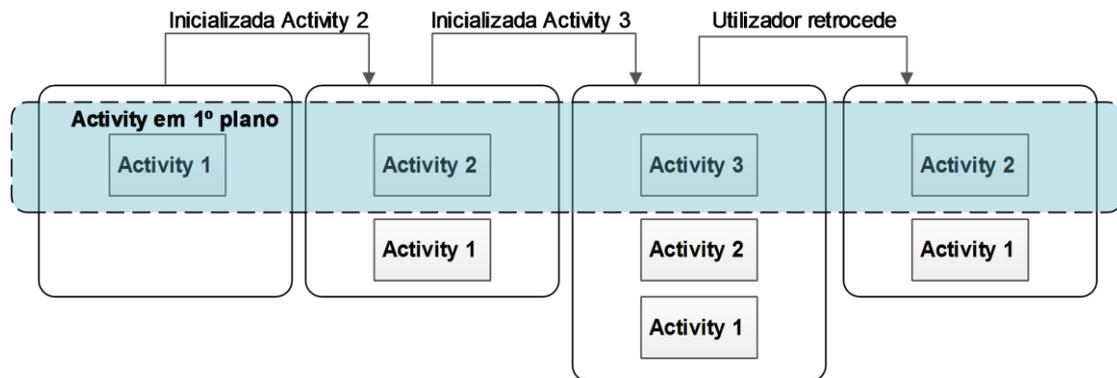


Figura 29 – Representação do início de uma nova Activity sem que a anterior seja destruída.

3.6.6 Navegação na aplicação

Enquanto o utilizador navega pela aplicação, tem disponível um *spinner* com diferentes opções, sendo que a cada opção escolhida estão associadas outras opções (Figura 30). Cada uma dessas opções é analisada pormenorizadamente de seguida.

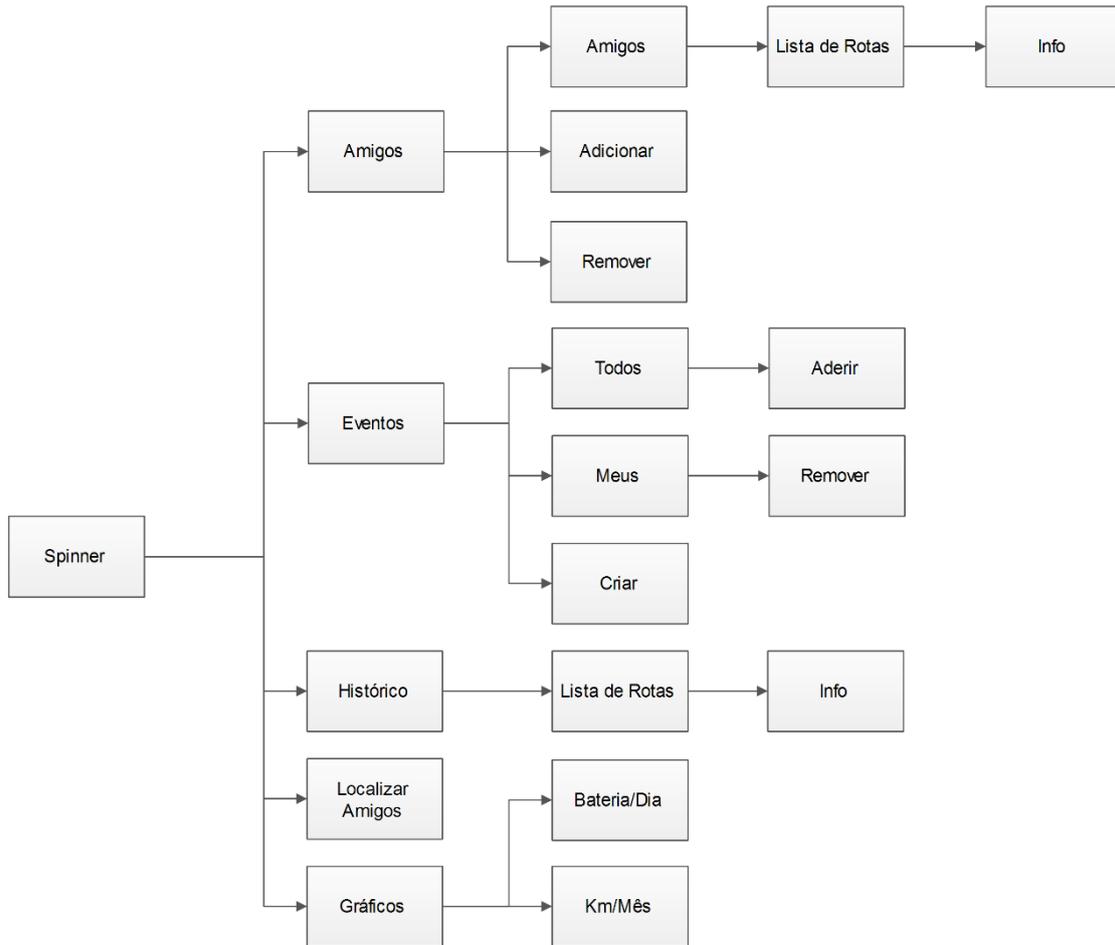


Figura 30 - Diagrama geral do spinner.

3.6.6.1 Amigos

Selecionada a opção Amigos é apresentado ao utilizador uma *swipe view* (recorrendo a *Fragments*) com três *tabs*: AMIGOS, ADICIONAR e REMOVER. Na *tab* AMIGOS, são apresentados todos os amigos do utilizador numa *list view* dinâmica contendo: foto, nome e email (Figura 31 à esquerda). Selecionando um desses amigos é apresentada novamente uma *list view*, que contém o histórico armazenado no servidor até ao momento (referente ao amigo selecionado), apresentando: data, duração e distância de cada trajeto (Figura 31 ao centro). Selecionando um desses trajetos, o utilizador tem acesso a informação mais pormenorizada do trajeto como: hora a que iniciou o trajeto, velocidade média, velocidade máxima e a trepidação média (Figura 31 à direita). O utilizador tem ainda acesso à rota efetuada, podendo navegar no mapa para uma melhor visualização. Para tal, basta um clique sobre a

imagem do mapa. A componente do mapa é abordada também no histórico do utilizador, ficando assim para essa secção uma maior descrição do seu funcionamento.

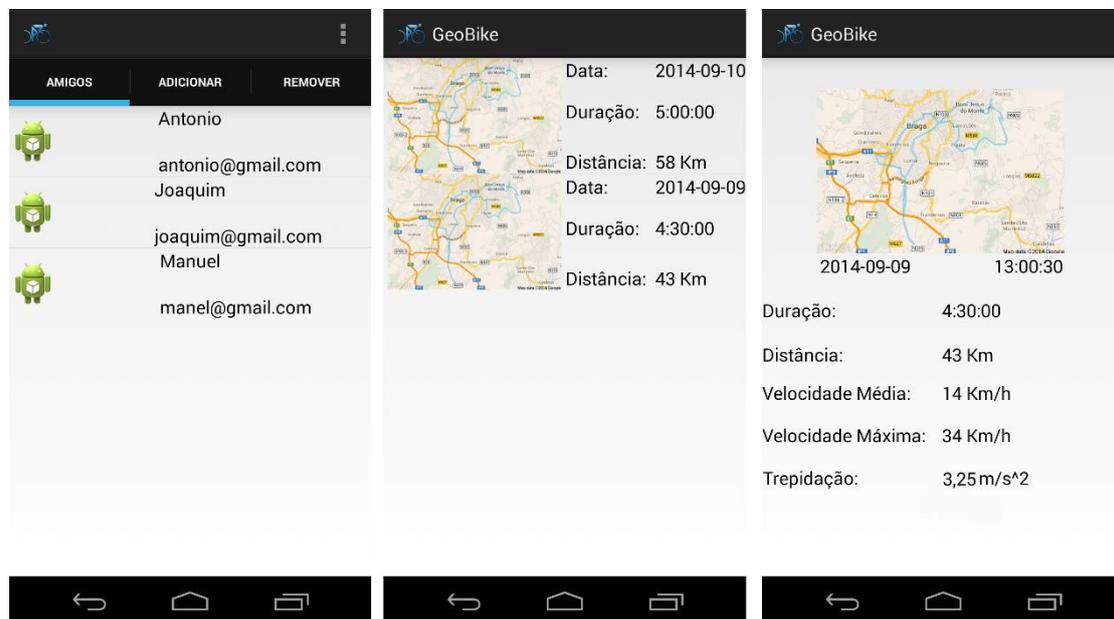


Figura 31 - Tab AMIGOS.

Na *tab* ADICIONAR o utilizador tem duas opções: dar permissão a um utilizador para que aceda à sua informação ou efetuar um pedido para que outro utilizador o adicione como amigo (Figura 32 à esquerda). Sempre que o utilizador preenche o email é verificado se este se encontra no formato correto, notificando o utilizador em caso de erro. Na Figura 32 ao centro pode-se observar um pedido de acesso a um utilizador cujo email não foi encontrado na BD. Perante esta situação, o utilizador tem a opção de o convidar para aderir à aplicação, ou pode simplesmente tratar-se de um engano na introdução do email. Na Figura 32 à direita é possível observar o caso em que o utilizador escolheu a opção de convidar o amigo. É então enviado um email a esse utilizador, com um conteúdo pré-inserido, onde é identificado o utilizador que o convida e o endereço da página Web onde é possível descarregar a aplicação.

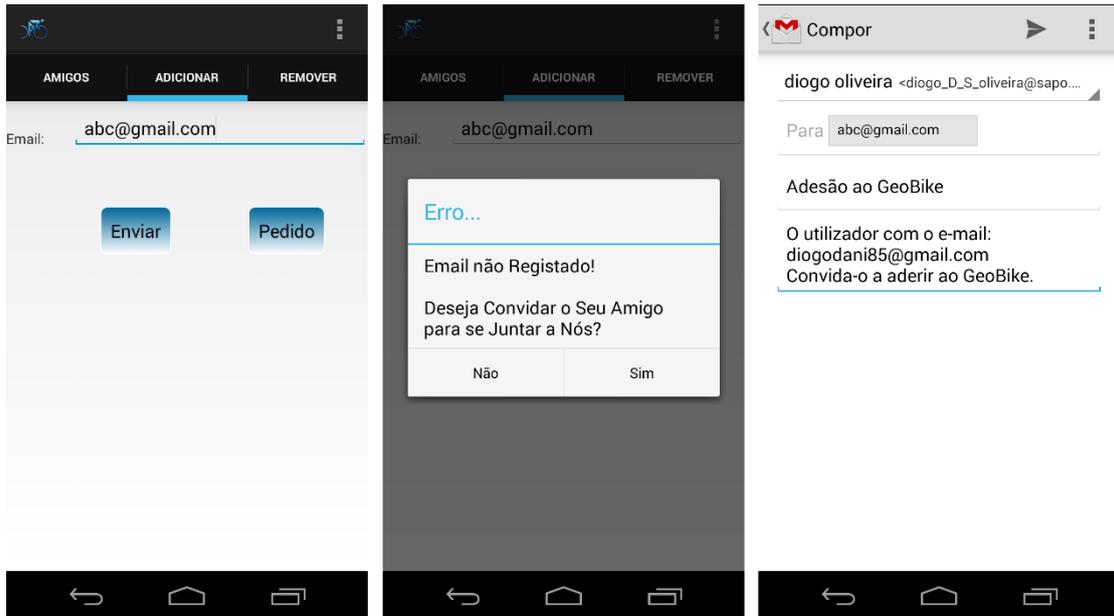


Figura 32 - Tab ADICIONAR.

Na *tab* REMOVE são apresentados todos os amigos do utilizador, tal como na *tab* AMIGOS (Figura 33 à esquerda), seleccionando um deles obtém-se informação relativa a esse amigo, como: nome, email, peso, altura e data de nascimento (Figura 33 à direita). Para além da informação sobre o amigo é também possível removê-lo da lista de amigos.

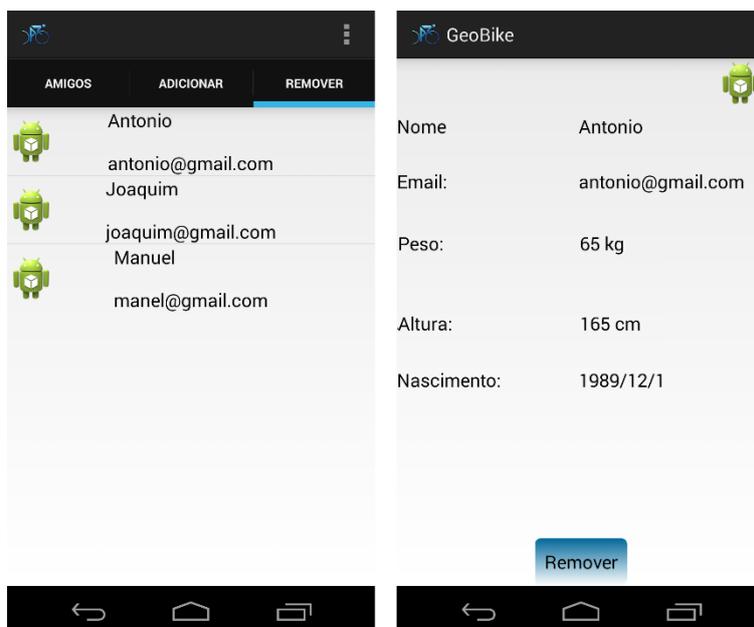


Figura 33 - Tab REMOVE.

3.6.6.2 Eventos

Selecionada a opção Eventos é apresentada ao utilizador, tal como na opção Amigos, uma *swipe view* (recorrendo a *Fragments*) com três tabs: TODOS, MEUS e CRIAR.

Na *tab* TODOS são apresentados, numa *list view* dinâmica, todos os eventos a que o utilizador pode aderir (Figura 34 à esquerda). Todos esses eventos são listados por ordem cronológica de realização, sendo apresentados apenas os eventos da data corrente ou superior. Na *list view* o utilizador tem acesso ao nome do evento, assim como a cidade onde se realiza e a data. No caso de o utilizador escolher um dos eventos, tem acesso a uma outra *Activity* onde são apresentados outros pormenores do evento, tais como: duração prevista do evento, distância e hora de início (Figura 34 à direita). Nessa *Activity* o utilizador tem ainda a possibilidade de aderir ao evento.

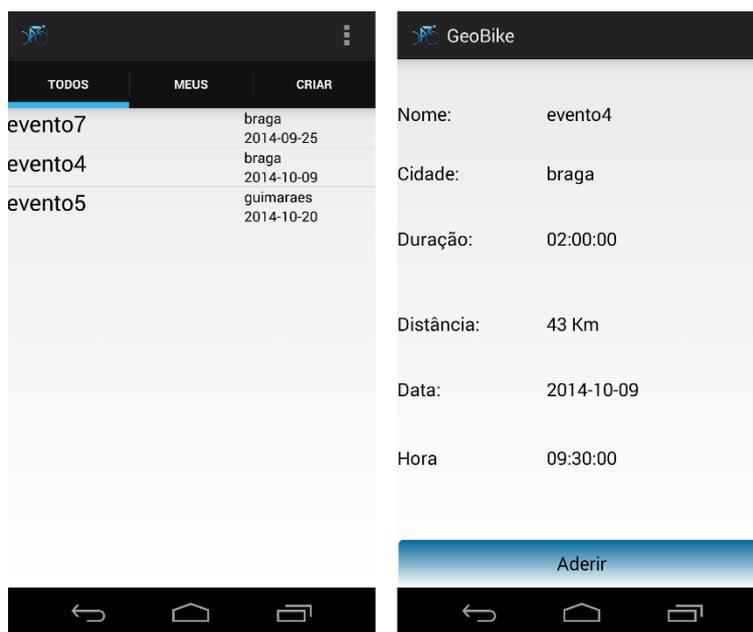


Figura 34 – *Tab* onde se encontram todos os eventos.

Relativamente à *tab* MEUS esta é em tudo idêntica à *tab* TODOS, com a diferença de conter apenas os eventos a que o utilizador aderiu (Figura 35 à esquerda). Após a seleção de um desses eventos, além da informação relativa ao evento, é apresentada a oportunidade de desistir do evento, removendo-o da lista de eventos do utilizador (Figura 35 à direita).

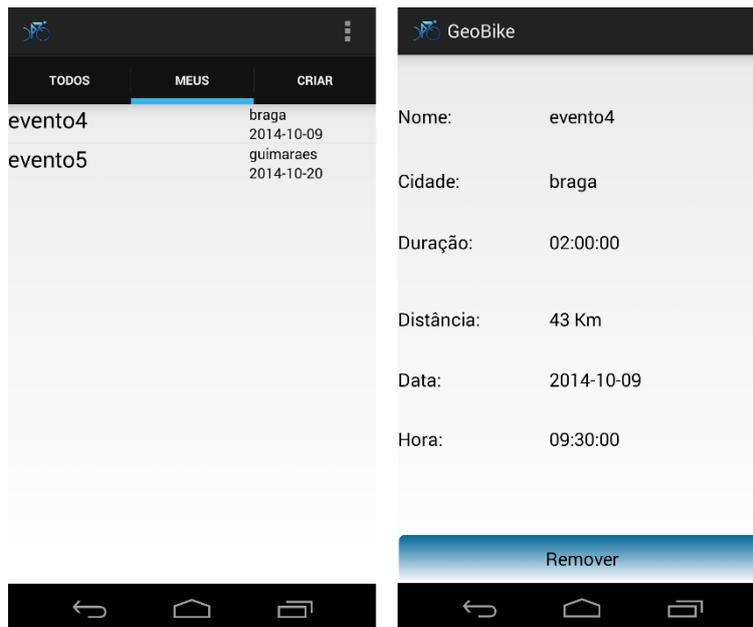


Figura 35 - *Tab* para visualizar os eventos que o utilizador aderiu.

Na *tab* CRIAR é dada a possibilidade ao utilizador de criar um evento, sendo que o utilizador tem de saber de antemão informações relativas ao evento, como: cidade onde se realiza, previsão da duração, distância em quilómetros a percorrer e data e hora de início (Figura 36 à esquerda). Tal como na *Activity* do registo, alguns campos têm de ter um formato específico, sendo apresentada ao utilizador uma janela de *popup* para que este escolha a opção desejada (Figura 36, ao centro e à direita). Quando o utilizador seleciona a opção guardar, é verificado se todos os campos estão preenchidos e, em caso afirmativo, é pedido ao servidor que efetue uma inserção na BD. No caso de o nome do evento ainda não ter sido inserido na BD, é apresentada ao

utilizador uma mensagem de que o evento foi criado com sucesso. Caso já exista um evento com o nome pretendido, o utilizador é notificado para que este seja alterado.

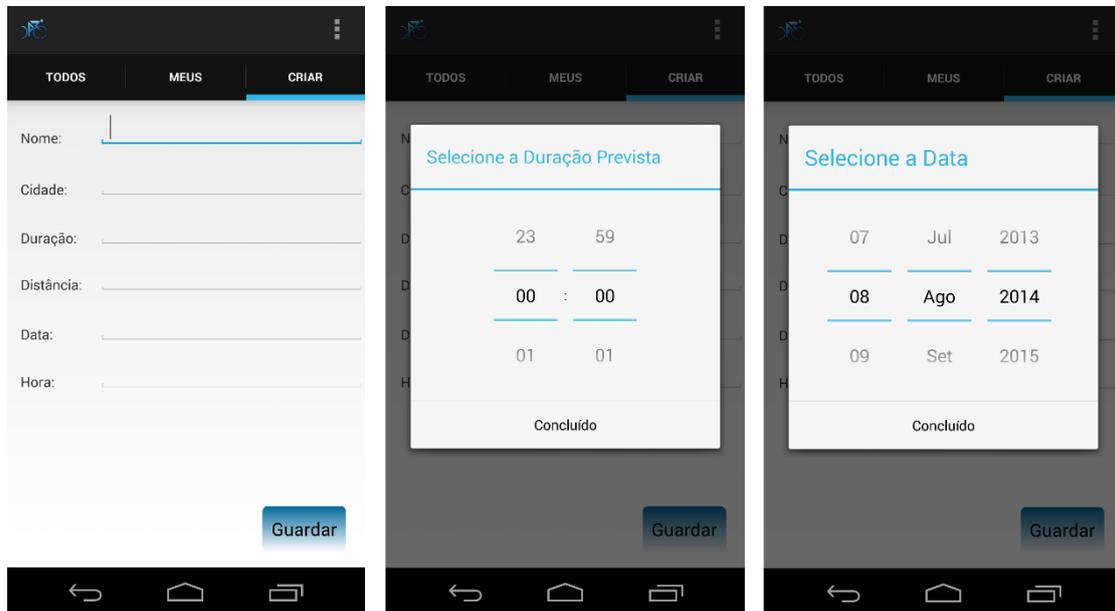


Figura 36 - Tab para criar um evento.

3.6.6.3 Histórico

Selecionada a opção Histórico é apresentada ao utilizador uma *list view* dinâmica contendo todos os seus trajetos efetuados, ordenados pela sua ordem de inserção. Cada trajeto tem como apresentação: data, duração e distância percorrida pelo utilizador (na Figura 37 à esquerda). Quando o utilizador seleciona um dos trajetos é-lhe apresentado um resumo referente a esse trajeto, contendo informações como: data, hora em que iniciou o trajeto, duração total, distância total percorrida, velocidade média, velocidade máxima, trepidação média e o acesso a um mapa (Figura 37 ao centro). Como referido na secção 3.6.6.1 o utilizador pode, através de um clique sobre o mapa, ter acesso a uma *Activity* que contém o seu percurso traçado no mapa (Figura 37 à direita). Para traçar o seu percurso no mapa é utilizada a API do Google Maps v2 para Android, que, para além de apresentar o mapa – podendo o utilizador navegar sobre ele e fazer zoom de uma forma interativa – permite o uso da classe *Polyline*, que define um conjunto de segmentos de linha conectados no mapa e permite a escolha da sua cor e largura. Para que seja possível essa conexão, é

necessário obter todas as latitudes e longitudes referentes ao trajeto ordenadas por ordem de sequência. Para tal é feita uma *query* à BD do smartphone, pedindo as latitudes e longitudes referentes ao id do trajeto. De forma que seja possível posicionar logo de início o percurso efetuado no mapa, para que fique visível ao utilizador, foi criado um método que permite o posicionamento da câmara centrada no início do trajeto e com o zoom adequado para a sua visualização.

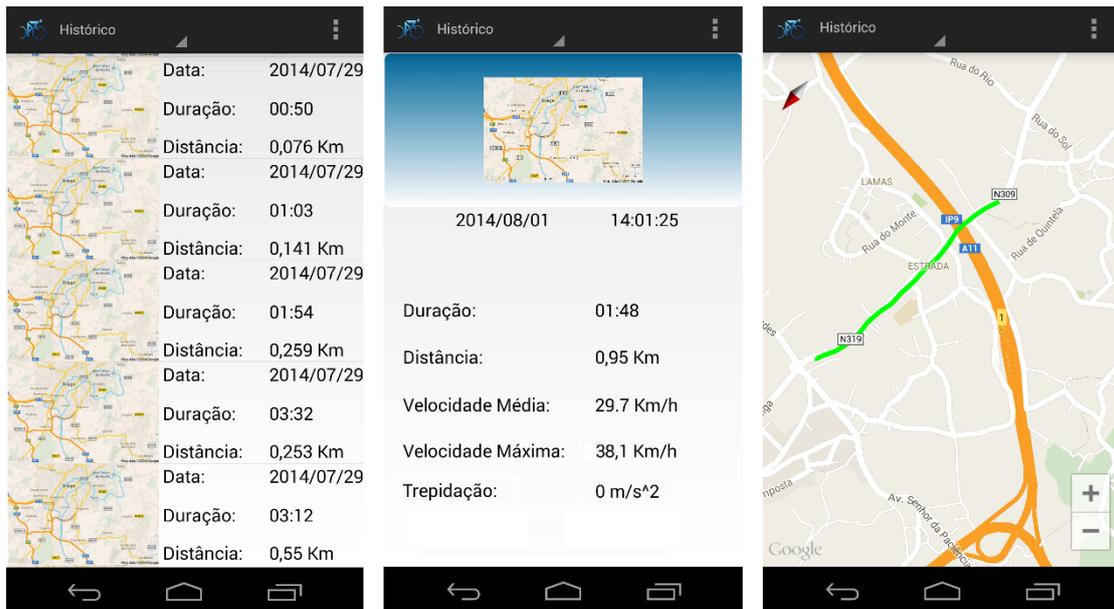


Figura 37 – Apresentação do histórico de um utilizador.

3.6.6.4 Localizar Amigos

Selecionada a opção Localizar Amigos primeiramente é verificado se o utilizador possui conexão à Internet, para que possa ser feito o download do mapa, estabelecida a conexão com o servidor e seja possível saber a última posição dos seus amigos. Se existir conexão à Internet é apresentado ao utilizador o mapa e, com recurso a um marcador, os seus amigos são posicionados no mapa. Clicando sobre o marcador, é possível ao utilizador saber o nome e o instante em que o amigo se encontrava naquela posição, recorrendo à data e hora (Figura 38).

Para que fosse possível rastrear a localização do utilizador com um intervalo de tempo aceitável, foi criado um método que, quando o utilizador se encontra no modo de navegação, guarda a sua posição (latitude e longitude), data e hora, de cem em

cem segundos. Quando o utilizador termina a aplicação, o local onde este se encontra é guardado como sendo a sua última localização conhecida e a informação relativa à data e hora é armazenada, para que o utilizador possa saber se existe probabilidade de um amigo ainda se encontrar naquela posição.

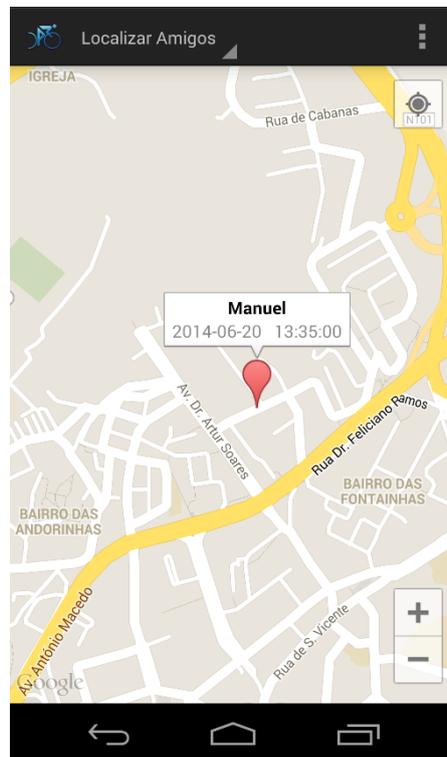


Figura 38 – Localizar amigos.

3.6.6.5 Gráficos

Selecionada a opção gráficos, o utilizador tem acesso a dois tipos de gráficos: um que permite ver a bateria gasta da bicicleta durante o dia corrente e comparar com o dia anterior (Figura 39) e outro que permite ao utilizador ter acesso aos quilómetros efetuados mensalmente durante o ano (Figura 40). Em ambos os gráficos, no início, é exibida uma mensagem ao utilizador sugerindo que este coloque o seu smartphone na posição horizontal, para uma melhor visualização, o que possibilita ao utilizador uma melhor interação com o mesmo. O utilizador, aquando da visualização do gráfico, pode navegar e aumentar ou diminuir o zoom, acedendo ao menu no canto inferior direito ou recorrendo aos gestos, tal como no mapa.

Para a criação de gráficos optou-se pelo uso da biblioteca AChartEngine versão 1.1.0 [56], dado ter disponíveis diferentes tipos de gráficos e ainda pela forma como o utilizador interage com gráfico (por permitir ao utilizador alterar o zoom e mover-se ao longo do gráfico).

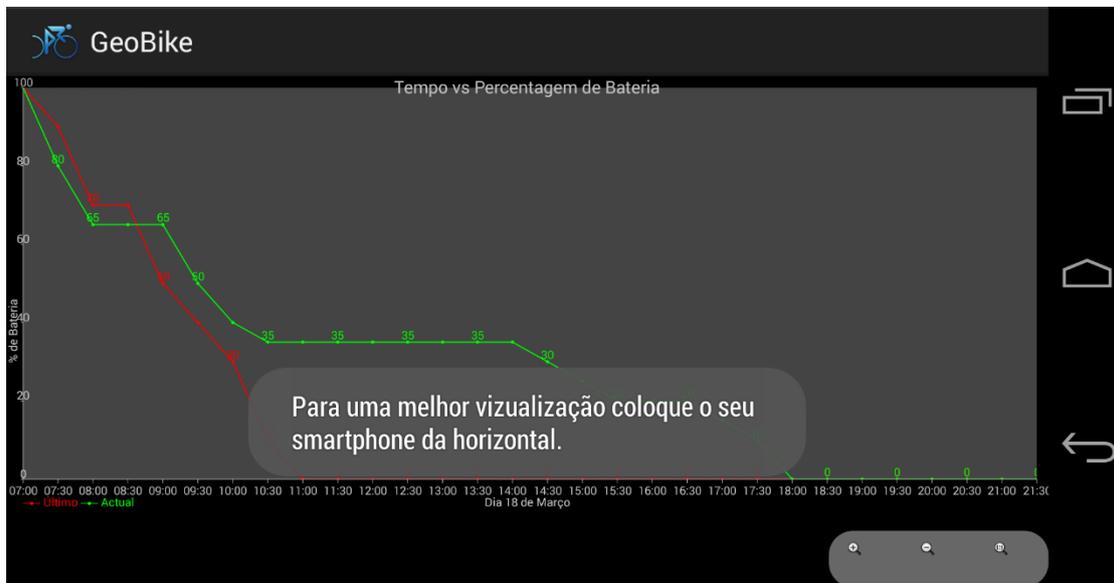


Figura 39 – Gráfico de linhas (gasto de bateria da bicicleta).

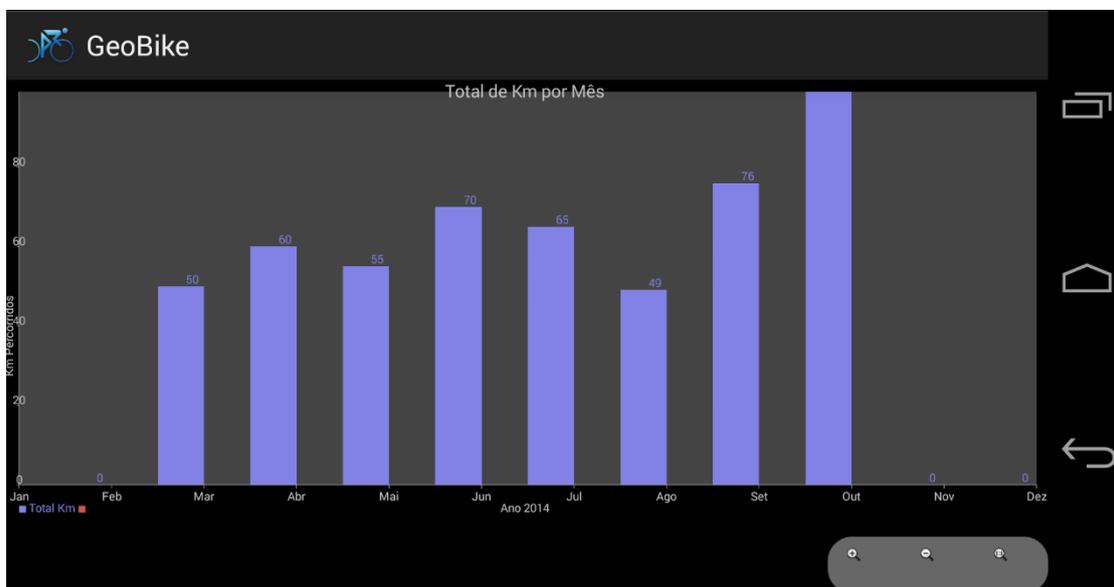


Figura 40 - Gráfico de barras (quilómetros efetuados).

3.6.7 Menu de overflow

O utilizador tem disponível um menu de overflow que lhe permite o acesso às definições, que termine a sua sessão e ainda que saia da aplicação (Figura 41). De seguida são analisadas as diferentes opções a que o utilizador tem acesso.

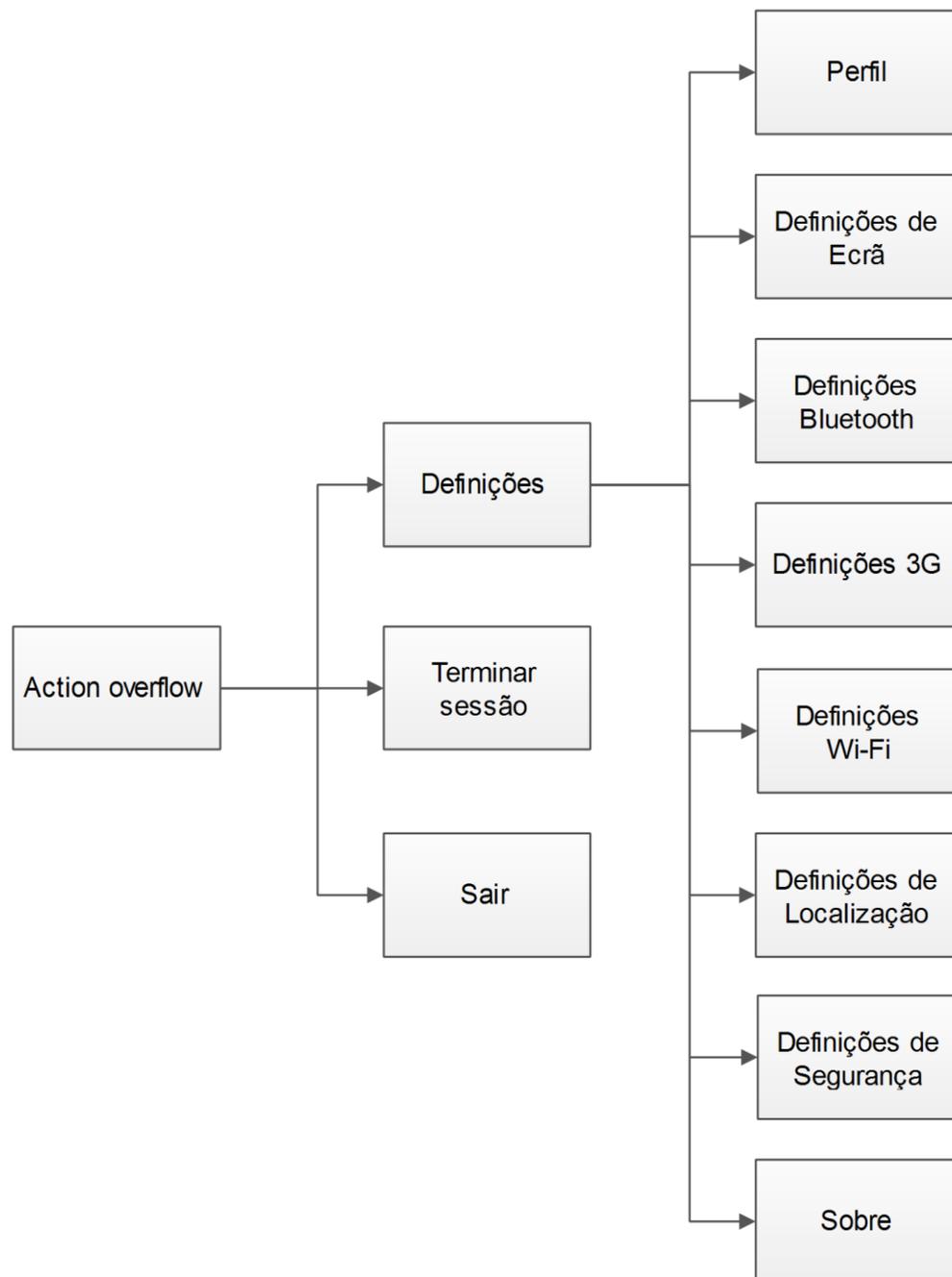


Figura 41 - Diagrama geral do *action overflow*.

Nas Definições é permitido ao utilizador efetuar uma série de configurações que afetam diretamente o uso da aplicação. Podem ser pedidas determinadas configurações no decorrer do uso da aplicação, podendo o utilizador aceder a estas sem a necessidade de sair da aplicação (por exemplo: ligar 3G ou GPS). Se o utilizador não necessitar de alguma configuração poderá desativa-la, como por exemplo o A-GPS, para poupar bateria.

O utilizador também pode aceder ao seu perfil, onde lhe é apresentado: a sua foto, nome, email, peso, altura, data de nascimento e género. Para que seja apresentada essa informação é efetuado um pedido HTTP ao servidor que devolve a informação referente ao utilizador. Na aplicação Android o utilizador apenas pode visualizar a sua informação. Caso pretenda efetuar alterações, poderá fazê-lo recorrendo à página Web.

Nas Definições de Ecrã o utilizador pode reduzir ou aumentar o brilho do ecrã e escolher o tempo para a suspensão do ecrã após um período de inatividade. Quando o utilizador se encontra no modo de navegação a suspensão de ecrã encontra-se bloqueada, permanecendo o ecrã sempre ligado, a não ser que o utilizador pressione o botão de bloqueio.

Nas Definições de Bluetooth o utilizador pode ativá-lo e, caso seja a primeira vez que efetua um trajeto numa determinada bicicleta, pode sincronizar o smartphone com a bicicleta.

Nas Definições 3G o utilizador pode ativar o acesso à Internet no smartphone e escolher o tipo de rede (3G, 4G), enquanto nas Definições Wi-Fi o utilizador pode ativar o Wi-Fi e ainda escolher a rede a que se pretende conectar.

Nas Definições de Localização o utilizador pode ativar o GPS e escolher o modo de localização:

- Alta precisão – acesso à Internet do smartphone, Wi-Fi e GPS;
- Poupança de bateria – acesso à Internet do smartphone e Wi-Fi;
- Apenas no dispositivo – utiliza apenas o GPS.

O modo de alta precisão é o aconselhado para uma localização mais precisa.

Nas Definições de Segurança o utilizador pode ativar ou desativar o serviço de emergência e pode, ainda, alterar o contato de emergência, sendo feito um pedido ao servidor para que efetue um *update* do contato de emergência referente ao utilizador na BD.

A informação relativa à versão da aplicação, por quem foi desenvolvida, em que âmbito foi desenvolvida e contato de email para eventuais dúvidas ou esclarecimentos encontra-se na opção Sobre.

No caso de o utilizador terminar sessão, como referido em 3.6.4.1, é invocado um método que coloca a *flag* IS_LOGIN a *false*, é limpo o conteúdo da variável que guarda o email do utilizador e a aplicação é redirecionada para a *Activity* de login. Embora o smartphone seja um dispositivo pessoal, a BD foi estruturada de forma a permitir que outro utilizador possa efetuar login sem que o histórico de utilizador fique associado sempre ao mesmo utilizador, permitindo desta forma que vários utilizadores possam usar o mesmo smartphone.

Houve alguma indecisão quanto ao método a abordar quando o utilizador pretende sair da aplicação. A primeira alternativa era invocar o método KillProcess, onde é fornecido o pid do processo como parâmetro. Esta solução implicava que o processo fosse forçado a encerrar, o que deve ser função do próprio gestor de processos do Android. Optou-se por utilizar o método finish() em conjunto com o System.exit(0), permitindo assim que a aplicação fosse inicializada com maior rapidez e o gestor do Android terminasse o processo caso achasse necessário.

3.6.8 Modo de navegação

Este é o modo principal da aplicação, é nele que são recolhidos todos os dados através dos sensores (GPS, acelerómetro e barómetro) do smartphone e armazenados no mesmo, para que sejam mais tarde sincronizados com a BD do servidor.

Enquanto o utilizador se encontra nesta janela tem acesso a: um cronómetro referente ao tempo total contabilizado até ao momento; sua velocidade atual; data e a hora; distância total percorrida; altitude a que se encontra; temperatura em graus

Celsius e mapa do local (Figura 42). A interação entre o utilizador e o mapa encontra-se bloqueada, quando este se encontra em movimento, devido ao facto da posição do mapa estar definida para seguir os seus movimentos e por questões de segurança. Quando o utilizador se encontra parado tem total liberdade para percorrer o mapa, pesquisar locais do seu interesse, aumentar e diminuir o zoom. Inicializado o modo de navegação é verificada a conexão à Internet e o estado do GPS; se estes não se encontrarem ativos o utilizador é notificado e, se pretender, é diretamente redirecionado para as definições, para que os possa ativar. O utilizador tem disponíveis dois botões, o Avança (iniciar trajeto) e o Parar (terminar trajeto). Inicialmente apenas é apresentado o Avança, pressionando-o, este é automaticamente colocado num estado invisível ao utilizador e o botão Parar assume o estado visível, dando assim apenas a possibilidade do utilizador dar por terminada a sua rota.

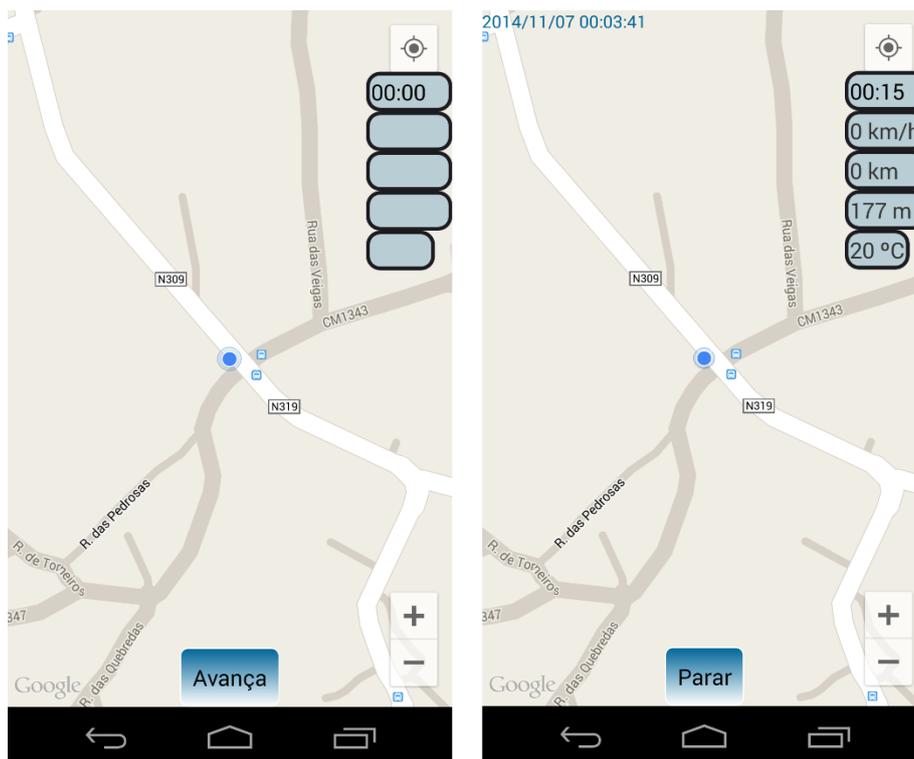


Figura 42 - Modo de navegação.

3.6.8.1 GPS

O GPS é um sistema de posicionamento global baseado em satélites. Consiste numa constelação de 24 satélites em 6 planos de órbita com 4 satélites em cada plano que permite uma total cobertura em todo o mundo. A técnica fundamental do GPS é a medição das distâncias entre o recetor e os satélites observáveis simultaneamente naquele instante. As posições dos satélites são previstas e transmitidas juntamente com o sinal de GPS ao utilizador. Através de várias posições conhecidas (dos satélites) e das distâncias entre o recetor e os satélites é possível determinar a posição do recetor, sendo apenas necessários 4 satélites para fornecer informações de localização [50].

A latitude e longitude são o sistema de coordenadas de mapas mais antigo utilizado para a localização na Terra [51]. A latitude é a distância angular medida a norte e sul do Equador (Figura 43). Tomando como ponto de referência a linha do Equador (0 graus de latitude), à medida que se desloca para norte, a latitude aumenta até 90 graus (chegada ao Pólo Norte), e à medida que se desloca para sul, a latitude diminui até -90 graus (chegada ao Pólo Sul). A longitude é a distância angular medida a este e oeste do Meridiano de Greenwich. Tomando como ponto de referência a linha do Meridiano de Greenwich (0 graus de longitude), à medida que se desloca para este, a longitude aumenta até 180 graus, e à medida que se desloca para oeste, a longitude diminui até -180 graus. Um grau de latitude ou longitude é equivalente a aproximadamente 112,65 km. Existem diferentes formas de representar a latitude e a longitude (exemplo do Campus de Azurém):

- Graus, minutos e segundos – $41^{\circ}27'07.6''N$ $8^{\circ}17'27.1''W$;
- Graus e minutos decimais – $41^{\circ}27.1283''N$ $8^{\circ}17.4516''W$;
- Graus decimais – 41.452107, -8.290872

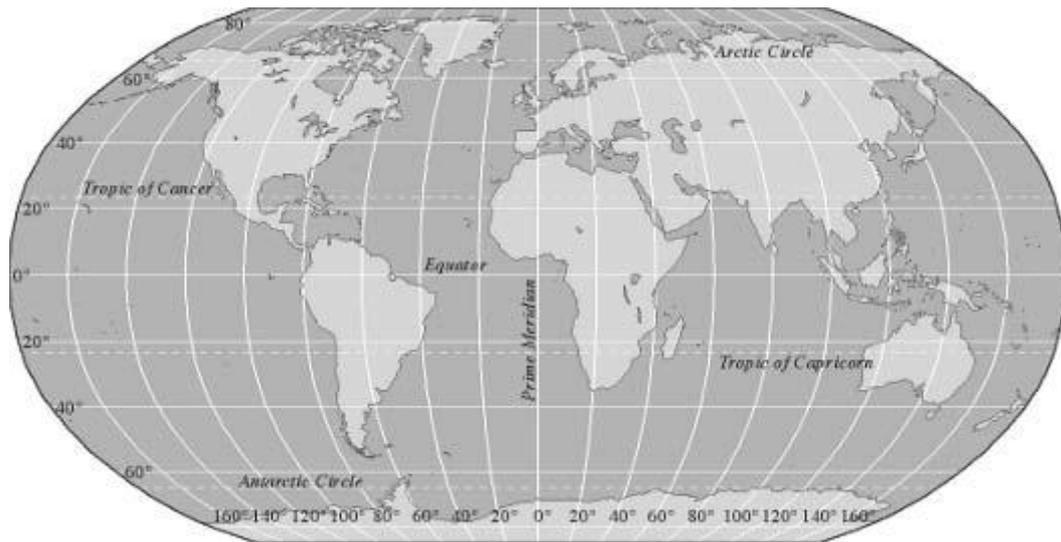


Figura 43 – Linhas imaginárias da latitude e longitude [51].

Com recurso à classe `LocationManager` foi possível o acesso aos serviços de localização do sistema, permitindo à aplicação obter atualizações periódicas sobre a localização geográfica do smartphone. Recorrendo à classe `Location` foi possível recolher, com o auxílio do sensor de GPS do smartphone, a latitude, a longitude (em graus decimais) e ainda a velocidade.

3.6.8.2 Altitude

Para obter uma altitude o mais próxima possível da real, existem dois fatores a ter em consideração: a qualidade do hardware do smartphone e a autenticidade da pressão ao nível do mar obtida. Para o cálculo da altitude foi utilizado o método `getAltitude(float p0, float p)`, que calcula a altitude em metros a partir da pressão ao nível do mar (p_0) e da pressão atmosférica (p). Para a pressão atmosférica foi utilizado o sensor do smartphone. Para a pressão ao nível do mar foi necessário recorrer a um serviço Web externo, do servidor `api.openweathermap.org`, onde é efetuado um pedido HTTP, dando como parâmetros a latitude e a longitude, para a obtenção do valor em hectoPascals (hPa), sendo obtida, também, a temperatura referente às coordenadas do utilizador. É possível, recorrendo a um diagrama de sequência (Figura 44), observar o exemplo da obtenção da pressão ao nível do mar.

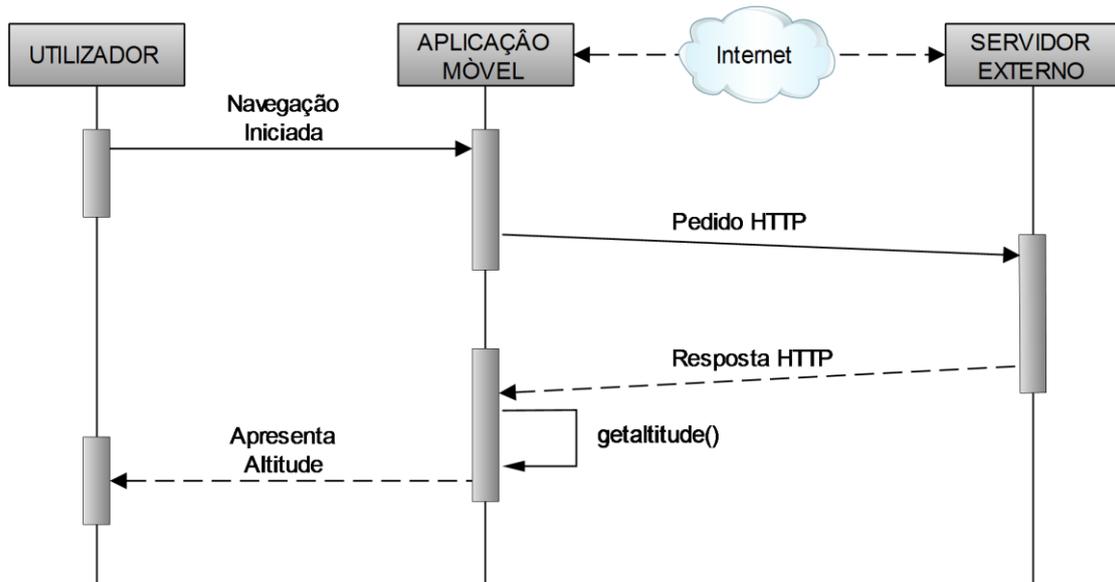


Figura 44 – Diagrama de sequência da obtenção da pressão ao nível do mar.

3.6.8.3 Distância

Existem várias equações importantes no que diz respeito à navegação. Uma delas é a fórmula de Haversine, que devolve uma distância credível entre dois pontos na superfície de uma esfera a partir das suas latitudes e longitudes [41], [42]. É possível obter a distância (d) usando a função *arcsin* (arco-seno), como se pode observar na equação (1):

$$d = 2r \operatorname{arcsin} \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (1)$$

Onde d é a distância entre dois pontos, r é o raio da esfera (6371 km), ϕ_1 corresponde à latitude 1, ϕ_2 corresponde à latitude 2, λ_1 corresponde à longitude 1, λ_2 corresponde à longitude 2.

3.6.8.4 SMS de segurança

Com recurso às SharedPreferences é verificado se a ajuda se encontra ativa. Estando ativa, é feito um pedido HTTP ao servidor, para que este devolva o contato de emergência associado ao utilizador, sendo este pedido feito uma única vez quando o utilizador inicializa a aplicação. Se a soma da velocidade nos últimos dez minutos for

igual a zero, é enviado um SMS para o contato de emergência, fornecendo a sua localização (latitude e longitude) (Figura 45). No caso do utilizador que recebe o SMS de emergência possuir um smartphone com GPS, ao selecionar a hiperligação enviada no SMS é imediatamente acionado o modo de navegação para o local através, por exemplo, da aplicação do Google Maps.



Figura 45 – Exemplo de um SMS de pedido de ajuda.

3.6.8.5 Trepidação

Para a obtenção dos valores relativos à trepidação foi utilizado o acelerómetro do smartphone e, recorrendo à classe `SensorManager`, foi escolhido como tipo de sensor o acelerómetro, selecionando ainda a opção para que o seu atraso fosse o menor possível (`SENSOR_DELAY_FASTEST`).

O acelerómetro mede a aceleração, em m/s^2 , aplicada a um dispositivo nos três eixos físicos (x , y e z) incluindo a aceleração da gravidade. Através da classe `SensorEvent` foi possível medir os três eixos (x , y e z), e calcular o módulo da aceleração exercida sobre ele (de 100 em 100 ms) através da equação (2), sendo a variável a correspondente à aceleração em cada eixo. Quando o dispositivo se encontra em repouso, esta equação retorna a aceleração da gravidade.

$$|v| = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2)$$

A troca de eixos aquando da mudança de orientação do ecrã foi considerada, dado que existe distinção, por exemplo, entre smartphones e tablets, no que diz respeito aos eixos. Uma vez que nesta dissertação se trata do uso de um smartphone, foi assumido que a posição do *layout* é sempre na vertical, ficando os eixos como se pode observar na Figura 46 (necessária configuração no `AndroidManifest.xml` para que o *layout* permaneça sempre na vertical).

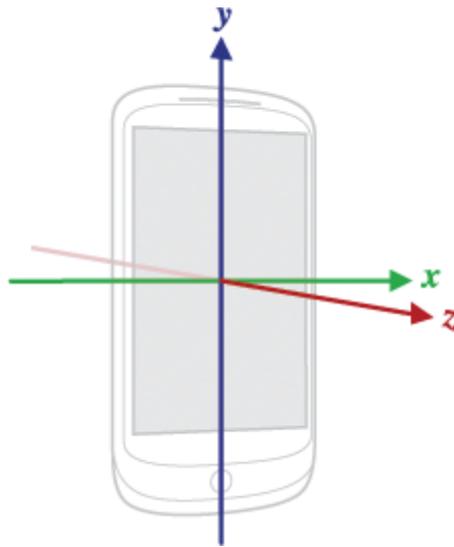


Figura 46 - Sistema de coordenadas no Android [57].

3.7 Website

Com o intuito de divulgar a aplicação, e para que fosse possível ao utilizador a consulta de toda a sua informação, assim como acompanhar os seus amigos, foi construído um website (Figura 47). Este permite ao utilizador aceder à informação a partir de qualquer dispositivo, bastando para tal, possuir uma ligação à Internet e um *browser*. Recorreu-se à *framework* Bootstrap [40], que fornece um vasto conjunto de formulários, botões, tabelas, barras de navegação, barras laterais, *tabs*, alertas (sucesso/insucesso), barras de progresso, listas, diversos *templates*, e possibilita, ainda, a navegação com dispositivos de diferentes tamanhos, ajustando a página a cada dispositivo. A versão 3 da API JavaScript do Google Maps [43] permite a incorporação do Google Maps no website, oferecendo diversos utilitários para manipulação de mapas e para a adição de conteúdos ao mapa; por exemplo, com recurso à classe *Polyline* foi possível desenhar o trajeto efetuado pelo ciclista. O Google Charts [44] permite a criação desde o gráfico mais básico até ao gráfico mais complexo com um simples JavaScript embutido na página Web. Os gráficos podem ser customizados (alteração da cor, tamanho e títulos), para uma melhor visualização e para que sejam mais apelativos ao utilizador. É permitido, ainda, que o utilizador interaja com um gráfico; por exemplo, no caso de um gráfico de barras, onde um valor

não seja perceptível em um dos eixos, o utilizador pode colocar o rato sobre a barra e é exibida a informação de ambos os eixos, como é possível observar na secção 3.7.2. Os componentes do website desenvolvida são abordados ao pormenor nas secções seguintes.

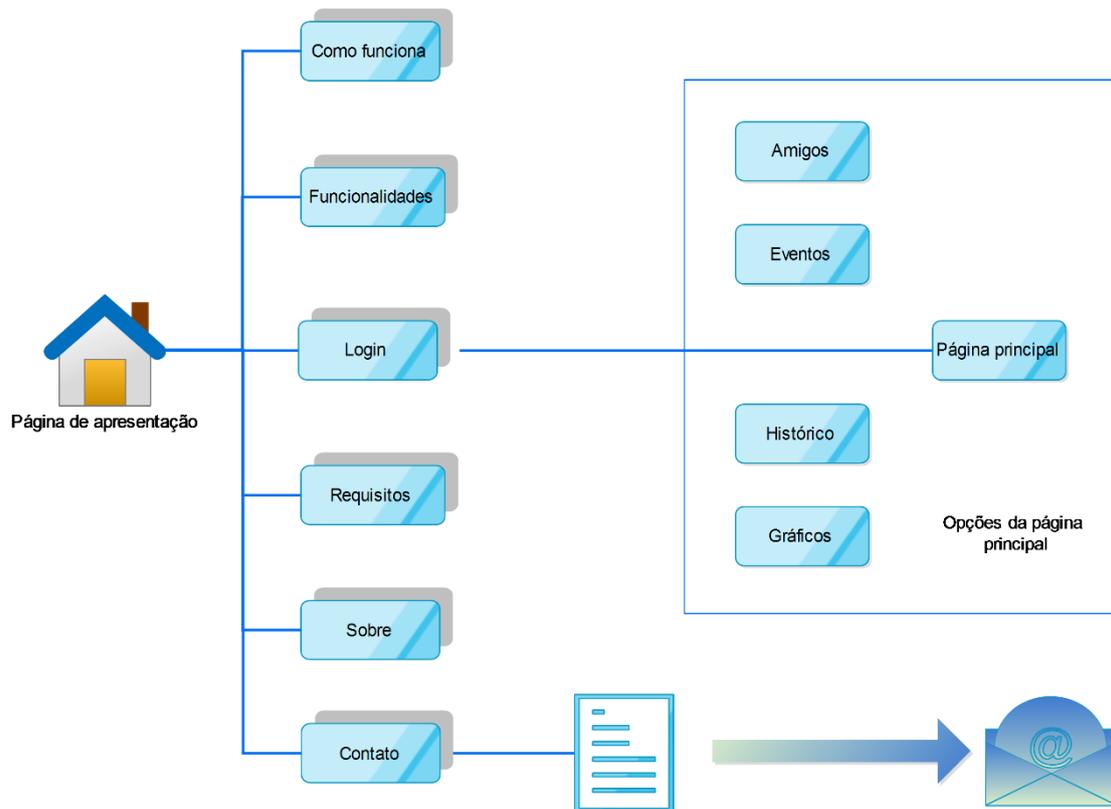


Figura 47 - Esquema geral do website.

3.7.1 Página de apresentação

Uma vez introduzido o endereço do website (<http://193.136.12.191/geobike>) no *browser*, o utilizador tem acesso à página de apresentação (Figura 48), onde pode entender melhor o funcionamento da aplicação, as suas funcionalidades, os requisitos necessários para a sua utilização (GPS, acesso 3G ou superior e dispositivo Android 4.0 ou superior) e, caso pretenda aderir à aplicação, com um simples clique no botão “Adira já” é efetuado o download. O utilizador pode aceder ao website diretamente do seu smartphone e, ao fazer o download, a aplicação pode ser instalada de imediato no seu dispositivo. Em alternativa, o utilizador pode fazer o download para outro dispositivo (por exemplo PC) e mais tarde transferir para o seu smartphone. Através

do separador Sobre, na barra de navegação, o utilizador tem acesso ao âmbito em que foi desenvolvida a aplicação e quais os objetivos.

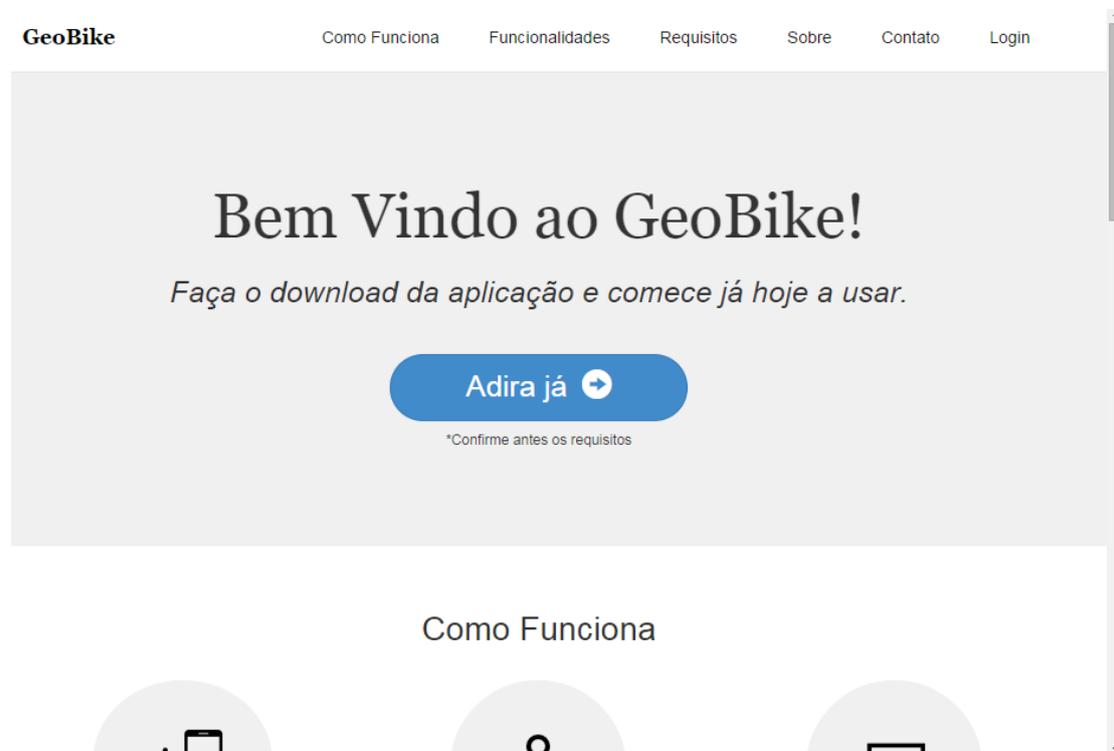


Figura 48 - Página de apresentação.

No separador Contato o utilizador pode submeter as suas dúvidas ou até mesmo sugestões para eventuais atualizações, através de um formulário (Figura 49), bastando introduzir o nome, endereço de email e a sua mensagem. Quando o utilizador clica em submeter, é verificado se todos os campos se encontram preenchidos e se o email está no formato correto. Clicando na hiperligação, “Departamento de Eletrónica Industrial”, o *browser* é redirecionado para o Google Maps, posicionando-se no local de desenvolvimento deste trabalho.

GeoBike Como Funciona Funcionalidades Requisitos Sobre Contato Login

Envie-nos a Sua Mensagem:

Nome
 ✓

Email
 ✓

Mensagem

Universidade do Minho Polo de Guimarães
[Departamento de Eletrónica Industrial](#)

Figura 49 - Página de contato.

No separador de login da página de apresentação, que se destina a utilizadores registados, o utilizador é redirecionado para a página de login, onde é necessária a introdução do email e password com que se registou. Após introdução das credenciais de acesso, quando o utilizador clica no botão Login, é estabelecida uma conexão com o servidor que, através de uma consulta à BD, verifica se a password corresponde ao email introduzido. No caso de a resposta vinda do servidor ser afirmativa, é apresentada ao utilizador a sua página principal, caso contrário, é apresentada uma mensagem de erro (Figura 50) para que o utilizador volte a introduzir os seus dados novamente.

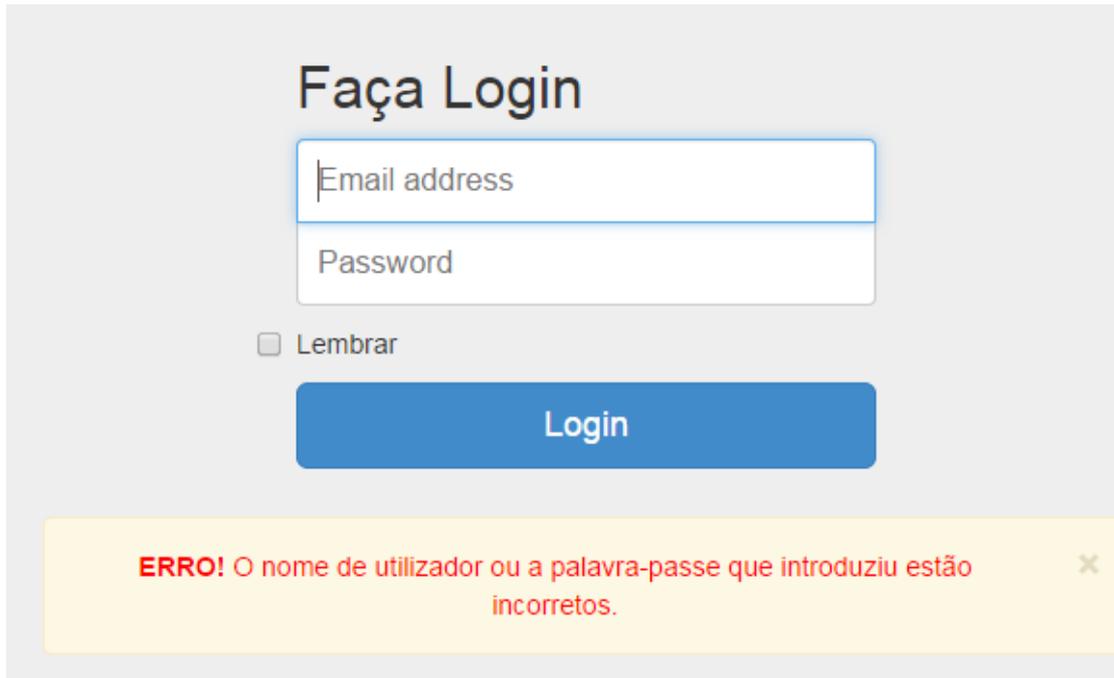


Figura 50 - Página de login com credenciais inseridas incorretamente.

3.7.2 Página principal

Uma vez na página principal, o utilizador tem acesso, através da barra de navegação, ao Perfil e ao Logout e, na barra lateral, às opções: Amigos, Eventos, Histórico e Gráficos (Figura 51). Ambas as barras encontram-se sempre disponíveis, independentemente da página onde o utilizador se encontre. Recorrendo à API JavaScript do Google Maps v3 é apresentado ao utilizador um mapa com o seu último trajeto, onde o utilizador pode interagir com este (fazer zoom, mover-se pelo mapa e alterar a vista do mapa entre mapa ou satélite). Junto ao rodapé do mapa, o utilizador encontra informação relativa a esse trajeto, tal como: data, hora de início, duração total, velocidade média e máxima em km/h e a trepidação média do terreno. O utilizador possui ainda acesso a um calendário feito em JavaScript (recorrendo à *framework* Bootstrap) com todas as marcações dos eventos que aderiu (Figura 52), onde pode navegar entre os meses passados e os seguintes e, com um clique numa das datas em que possui um evento (marcação azul), é redirecionado para esse evento, para que possa ver todos os detalhes relacionados com o mesmo. Juntamente com o calendário o utilizador tem disponível um gráfico de barras, desenvolvido

através da API do Google Charts, onde pode visualizar os seus quilómetros totais percorridos em simultâneo com os quatro amigos que mais quilómetros fizeram.

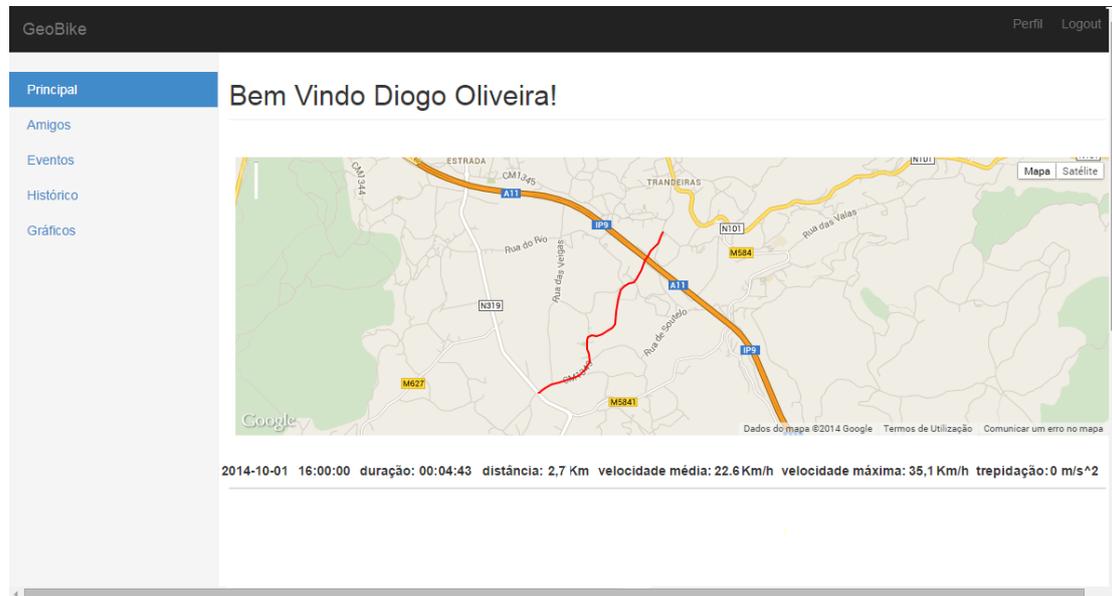


Figura 51 - Página principal.

Calendário dos Meus Eventos

Setembro 2014						
D	S	T	Q	Q	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

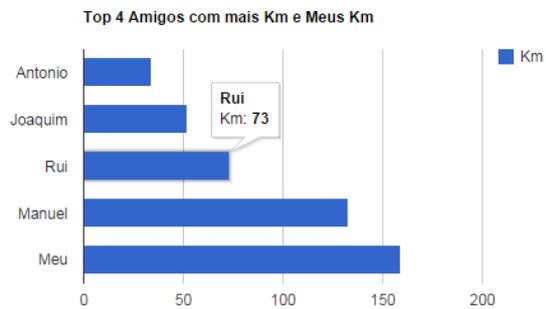


Figura 52 - Página principal (calendário e gráfico do top4).

3.7.2.1 Amigos

Aquando da escolha da opção Amigos, na barra lateral (Figura 53), é feito um pedido ao servidor para que consulte a BD e devolva todos os amigos referentes ao utilizador (através do email), tendo este acesso a uma lista com todos os seus amigos, onde é apresentada uma foto, o nome e o email para cada amigo. No caso de o utilizador pretender aceder ao histórico de um dos seus amigos, basta seleccioná-lo através da lista. Um novo pedido ao servidor é efetuado para que devolva todos os

trajetos referentes ao amigo selecionado, sendo apresentada uma lista ao utilizador com todo o histórico referente a esse amigo (Figura 54). Nessa lista, cada trajeto tem associado uma data de realização, a duração e a distância total percorrida. O utilizador tem ainda a opção de selecionar um desses trajetos, obtendo assim mais informação relativa a esse trajeto (hora de início, velocidade média, velocidade máxima e trepidação média) e um mapa como o da página principal, que apresenta o trajeto feito pelo amigo e possibilita a interação do utilizador com o mesmo (Figura 55).

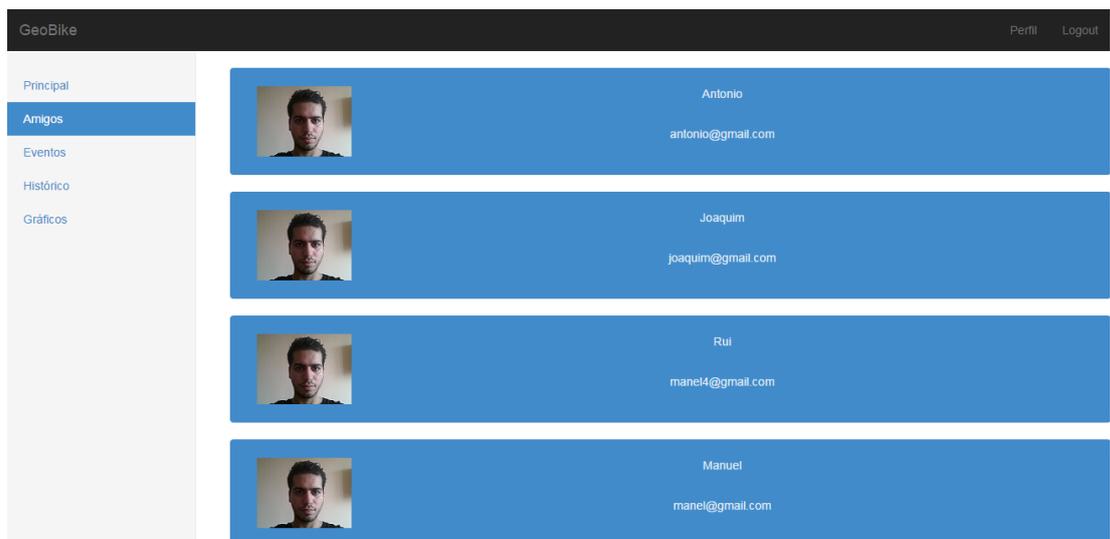


Figura 53 – Lista de amigos.



Figura 54 – Histórico de um amigo.

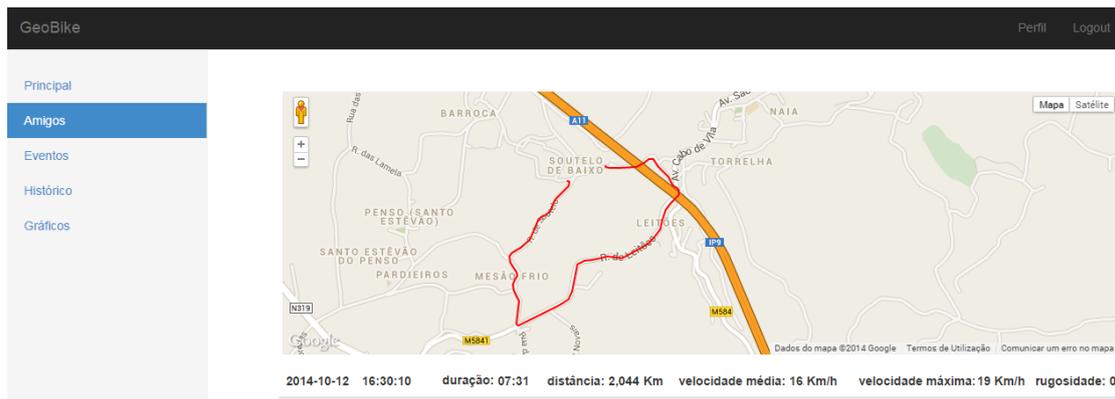


Figura 55 – Trajeto de um amigo.

3.7.2.2 Eventos

Selecionada a opção Eventos na barra lateral (Figura 56), o utilizador tem três opções: uma lista de todos os eventos a que pode aderir (Todos), os seus eventos (Meus) e criar um evento (Criar). Na opção Todos encontram-se todos os eventos a partir da data corrente, sendo apresentados ao utilizador numa lista, contendo o nome do evento, a data e a cidade onde se realiza. Escolhendo um deles é apresentada ao utilizador informação mais detalhada sobre o evento e dada a oportunidade de aderir a esse mesmo evento.



Figura 56 – Lista de eventos de um utilizador.

Na opção Meus encontram-se listados todos os eventos aos quais o utilizador aderiu. Selecionando um dos eventos é apresentada ao utilizador toda a informação relativa a esse evento (Figura 57).

The screenshot shows the GeoBike web application interface. On the left is a navigation menu with options: Principal, Amigos, Eventos (highlighted), Histórico, and Gráficos. The top right of the header contains 'Perfil' and 'Logout'. The main content area has three tabs: 'Todos', 'Meus', and 'Criar'. The 'Criar' tab is active, displaying a form with the following fields:

Field Label	Value
Nome	Evento
Cidade	Braga
Duração	02:30:00
Distância	30
Data	2014-08-07
Hora	09:30:00

Figura 57 – Informação relativa a um evento que o utilizador aderiu.

Na opção Criar é dada a possibilidade ao utilizador de criar um evento, tendo obrigatoriamente de preencher todos os campos no formato recomendado (Figura 58). Quando o utilizador clica no botão Criar é então verificado se todos os campos se encontram preenchidos e no formato correto, notificando o utilizador caso não se verifique esta condição. Caso todos os dados se encontrarem inseridos no formato correto, é estabelecida uma comunicação com o servidor requisitando a inserção dos dados na BD.

The screenshot shows the 'GeoBike' web application interface. On the left is a sidebar with navigation options: 'Principal', 'Amigos', 'Eventos' (highlighted in blue), 'Histórico', and 'Gráficos'. The top right of the header contains 'Perfil' and 'Logout'. The main content area is titled 'Criar' and contains a form with the following fields:

- Nome:** 'Nome do Evento' with a checkmark.
- Cidade:** 'Cidade' with a checkmark.
- Duração:** 'Duração do Evento ex: 01:30:00' with a checkmark.
- Distância:** 'Distância do Evento em Km ex: 34' with a checkmark.
- Data:** 'Data ex: 2014-09-23 ou 2014/09/23' with a checkmark.
- Hora:** 'Hora ex: 14:30' with a checkmark.

A blue 'Criar' button is located at the bottom right of the form.

Figura 58 – Criar um evento.

3.7.2.3 Histórico

A opção Histórico da barra lateral é idêntica à seleção do histórico de um amigo, mas referente ao próprio utilizador, apresentando uma lista com todos os seus trajetos efetuados: data, duração e distância total percorrida de cada um. Caso o utilizador selecione um deles é apresentada a informação referente a esse mesmo trajeto (data, hora de início, duração, distância total percorrida em quilómetros, velocidade média e máxima em km/h e a trepidação média em m/s^2) e o mapa com o seu percurso, dando também a oportunidade de o utilizador interagir com o mapa, tal como referido em 3.7.2.

3.7.2.4 Gráficos

Na opção Gráficos da barra lateral (Figura 59) são apresentados ao utilizador dois tipos de gráficos: um gráfico de linhas onde é possível observar a percentagem de

bateria da bicicleta gasta ao longo do tempo para um dia e comparar com o dia anterior e um gráfico de colunas que apresenta o total de quilómetros efetuados mensalmente pelo utilizador para um ano completo.

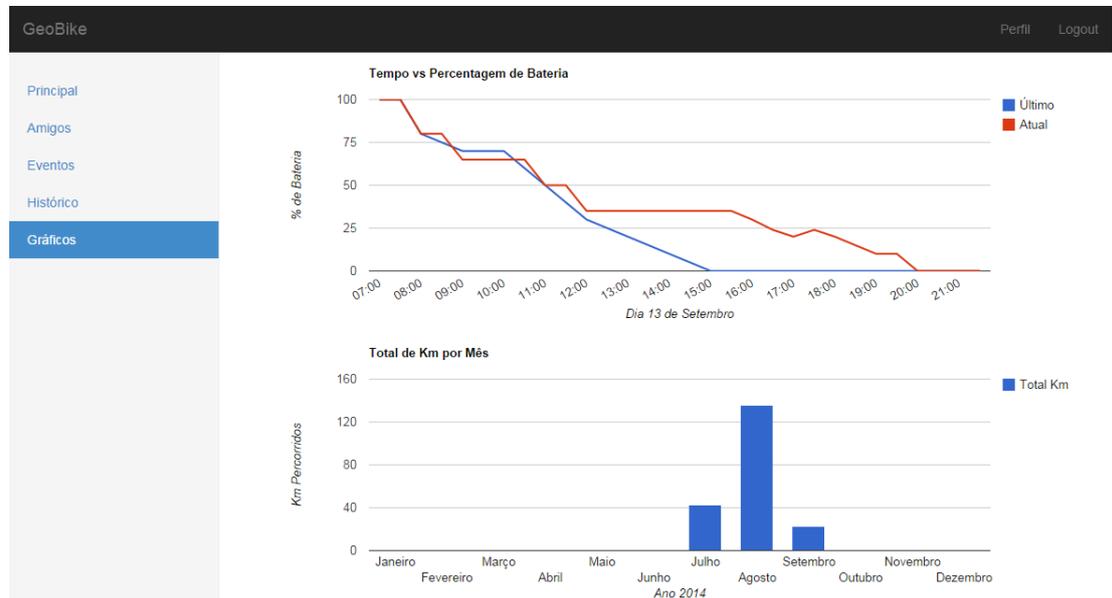
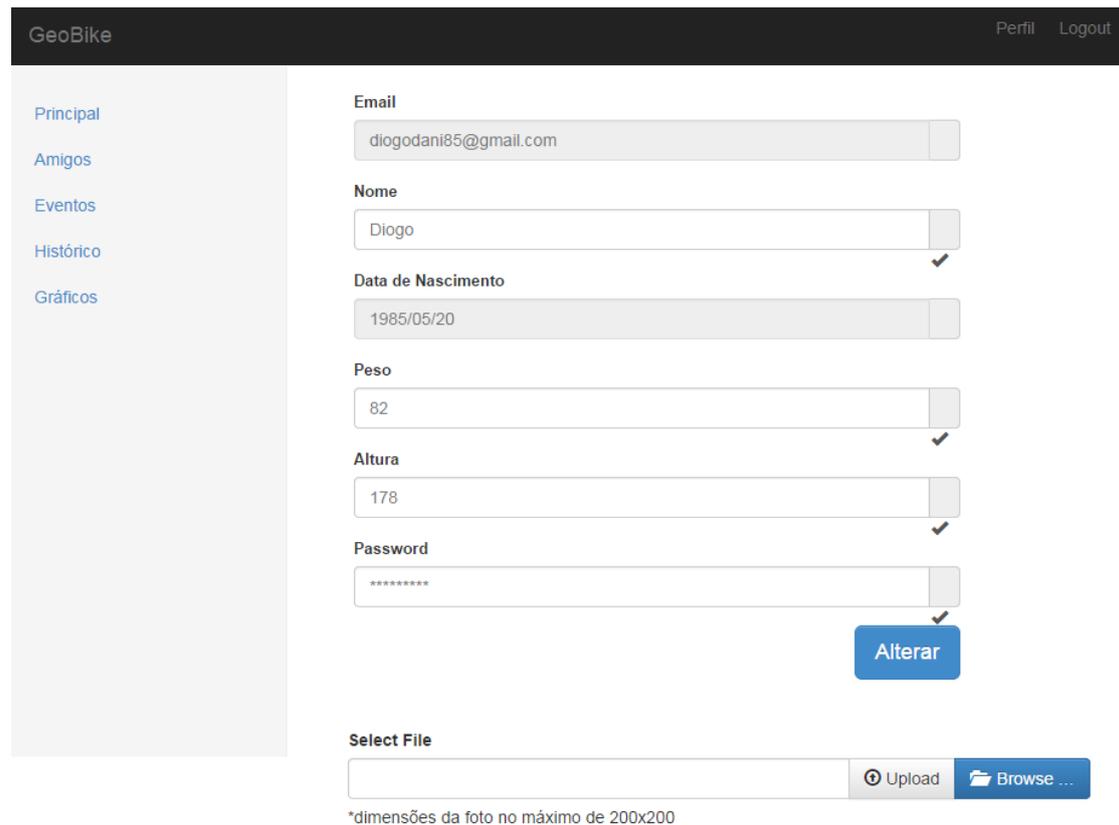


Figura 59 – Gráficos disponíveis ao utilizador.

3.7.2.5 Barra de navegação

Na barra de navegação o utilizador tem acesso a duas opções: o Perfil e o Logout. O Logout permite que o utilizador termine a sua sessão, obrigando a que da próxima vez que pretenda aceder ao website tenha de efetuar novamente o login. No Perfil (Figura 60) é feita uma consulta à BD e são listados todos os dados referentes ao utilizador que efetuou o login. Os dados como o email e data de nascimento são inalteráveis. Os restantes dados podem ser alterados na opção Alterar, onde, após a introdução dos novos dados, é efetuado o update da BD. O utilizador tem ainda a possibilidade de associar uma foto ao seu perfil, escolhendo o ficheiro a partir do seu dispositivo, que não deve exceder as dimensões de 200x200 pixéis.



The screenshot shows the user profile page for 'GeoBike'. The page has a dark header with 'GeoBike' on the left and 'Perfil Logout' on the right. A light gray sidebar on the left contains navigation links: 'Principal', 'Amigos', 'Eventos', 'Histórico', and 'Gráficos'. The main content area displays several form fields for personal information, each with a checkmark to its right, indicating they are filled: 'Email' (diogodani85@gmail.com), 'Nome' (Diogo), 'Data de Nascimento' (1985/05/20), 'Peso' (82), 'Altura' (178), and 'Password' (masked with asterisks). A blue 'Alterar' button is positioned below the password field. At the bottom, there is a 'Select File' section with an 'Upload' button and a 'Browse ...' button. A note below this section states: '*dimensões da foto no máximo de 200x200'.

Figura 60 – Dados alusivos ao perfil do utilizador.

3.8 NFC

Near Field Communication (NFC) é um conjunto de tecnologias sem fios de curto alcance, geralmente exigindo uma distância nunca superior a 4 cm para que seja possível iniciar uma conexão [35]. Permite a partilha de pequenas quantidades de dados entre uma *tag* NFC (Figura 61) e um dispositivo compatível. Devido à curta distância exigida para que possam ser transferidos os dados, não é necessária uma grande preocupação ao nível de segurança.

Recorreu-se ao uso desta tecnologia com o intuito de simplificar a interação do ciclista com a aplicação e a bicicleta, podendo este, sempre que inicia uma nova rota, ter um conjunto de pré-configurações efetuadas automaticamente (ligar Bluetooth e emparelhar com a bicicleta automaticamente, ligar GPS, ligar 3G e inicializar a aplicação).

Para que se possa escrever para uma *tag* e ler através do smartphone, foram escolhidas duas aplicações, a NFC Tools [36] e a NFC Tasks [37], respetivamente (Apêndice C – NFC).



Figura 61 – *Smart tag* NFC em autocolante.

4. Resultados e discussão

Neste capítulo são apresentados e discutidos os resultados obtidos, assim como decisões relevantes para o desempenho da aplicação. Foram efetuados testes no sentido de averiguar diferenças no hardware de diferentes dispositivos. Conclui-se que, dependendo do modelo do dispositivo, os resultados obtidos são perceptivelmente diferentes, sendo assim, os testes foram efetuados recorrendo sempre ao mesmo dispositivo (Nexus 5). Nas seguintes secções são descritos testes relativos à altitude, distância, velocidade, trepidação, navegação e aos gastos de bateria e dados móveis.

4.1 Teste da altitude

Para que fosse possível comparar uma altitude de referência com a fornecida pela aplicação desenvolvida, foi efetuada uma pesquisa por aplicações ou páginas Web que fornecessem este tipo de serviço. De entre as várias encontradas, destacam-se o EarthTools [45] e o Google Earth [46]. O EarthTools recorre ao Shuttle Radar Topography Mission (SRTM) [47] para obter a pressão ao nível do mar, num projeto conjunto entre a National Geospatial-Intelligence Agency (NGA) e a National Aeronautics and Space Administration (NASA). No caso do Google Earth, os valores de altitude são obtidos através de um modelo de elevação digital (DEM - Digital Elevation Model) e os dados são coletados através do SRTM tal como no EarthTools.

A altitude podia ser obtida diretamente através de várias APIs disponíveis, mas que implicavam o envio constante das coordenadas geográficas do dispositivo para a sua obtenção. Na solução desenvolvida é necessário apenas efetuar um pedido da pressão ao nível do mar uma vez, permitindo assim uma grande poupança no consumo de dados móveis.

Os diferentes testes para a altitude foram efetuados no mesmo local, onde o EarthTools devolveu uma altitude de 210 metros (Figura 62), o Google Earth uma altitude de 218 metros (Figura 63) e a aplicação desenvolvida uma altitude de 216 metros (Figura 64). Os três valores referentes à altitude encontram-se muito próximos, sendo a maior proximidade entre a aplicação desenvolvida e o Google Earth, com uma diferença de apenas 2 metros.



Figura 62 - Obtenção da altitude recorrendo ao EarthTools.

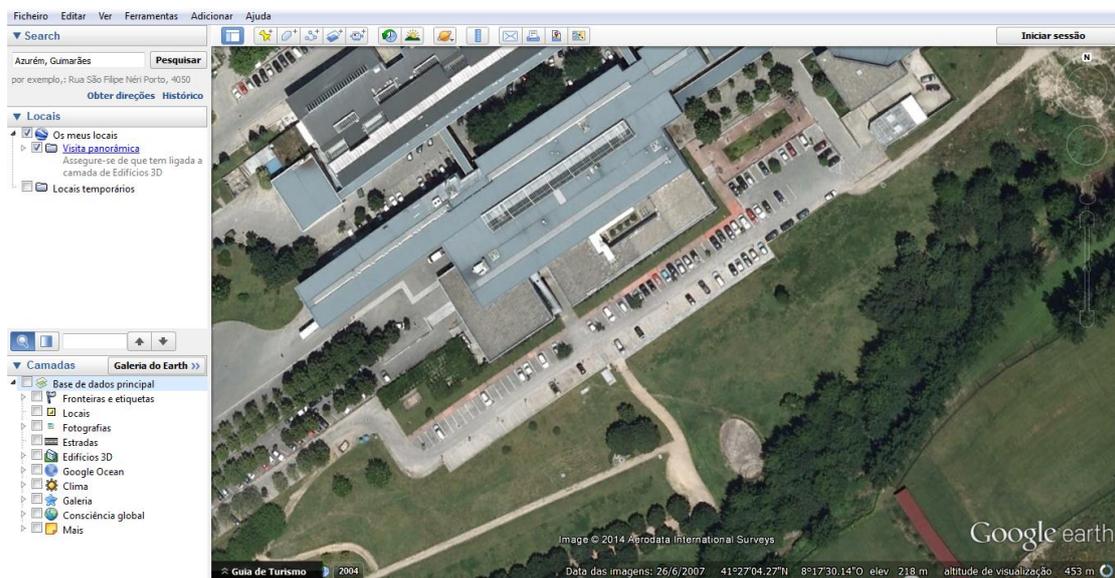


Figura 63 - Obtenção da altitude recorrendo ao Google Earth.

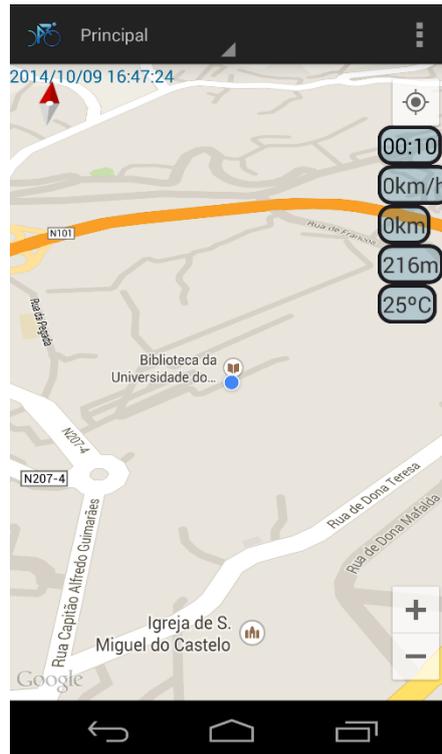


Figura 64 - Altitude na aplicação desenvolvida.

Foram efetuados ainda testes recorrendo às aplicações OsmAnd [48] e Phone Tester [49], sem que estas se encontrassem com acesso à Internet – utilizando apenas o GPS do smartphone. Nestas condições não se efetua qualquer consulta quanto à pressão ao nível do mar, utilizando apenas a triangulação de satélites para obter a altitude. O OsmAnd devolveu uma altitude de 258 metros e o Phone Tester uma altitude de 259 metros (Figura 65), sendo valores muito próximos um do outro, mas mais elevados relativamente aos valores obtidos previamente. Foi efetuado um teste entre a aplicação desenvolvida e o Phone Tester, num percurso de 2 km, para uma melhor perceção da diferença de altitude (Figura 66). A diferença na altitude registada por ambos manteve o mesmo padrão, independentemente das subidas e descidas de altitude. Para a aplicação desenvolvida não é tão relevante a diferença na altitude registada, pois o importante é a referência inicial, sendo que as variações são registadas corretamente de ambas as formas. No entanto, a abordagem escolhida foi a que mais se aproximou de altitudes credíveis, como as obtidas no Google Earth.

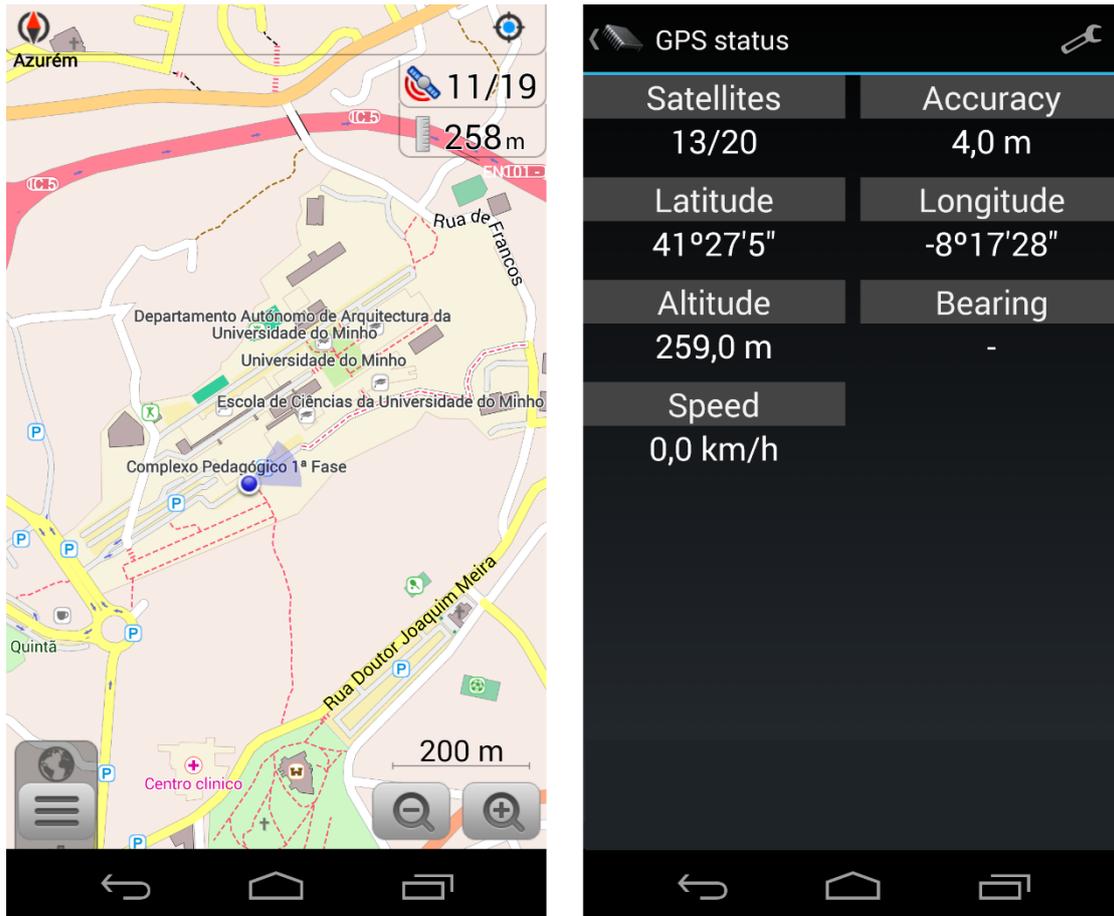


Figura 65 - Altitude na aplicação OsmAnd vs altitude na aplicação Phone Tester.

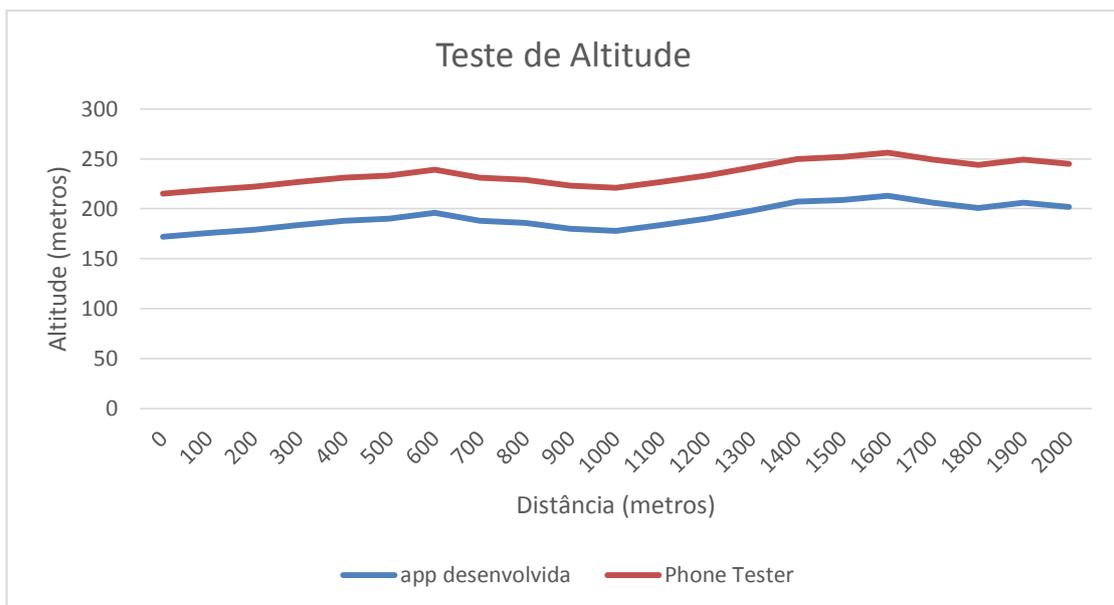


Figura 66 - Teste da altitude.

4.2 Teste da distância

Para o teste da distância percorrida optou-se por efetuar a comparação recorrendo a um automóvel convencional (Toyota Aygo). Foi efetuado um percurso durante 8 minutos ao longo de 2 km com uma velocidade constante de 15 km/h, tendo sido recolhidas amostras de 24 em 24 segundos. Os valores de distância percorrida registados pela aplicação desenvolvida estiveram muito próximos dos registados pelo odómetro do automóvel (Figura 67), exceto inicialmente, onde, durante a obtenção dos satélites, foi introduzido um erro de 20 metros. Esse erro é mantido constante até ao final do teste, sendo o valor final registado na aplicação de 2020 metros, enquanto o automóvel registou 2000 metros. Sendo assim, verificou-se que a fórmula de Haversine é capaz de fornecer dados relativos à distância com uma margem de erro reduzida.

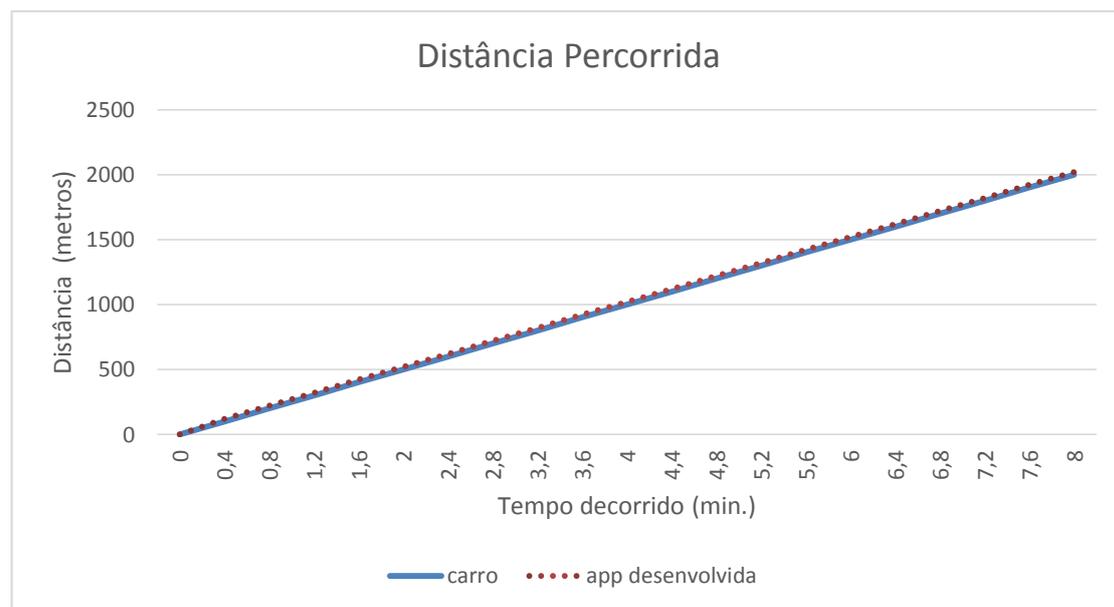


Figura 67 - Teste relativo à distância percorrida.

4.3 Teste de velocidade

Para o teste da velocidade foi comparada a aplicação desenvolvida com a aplicação Endomondo e com o velocímetro do mesmo automóvel utilizado na secção anterior. O teste foi efetuado num percurso com uma duração de aproximadamente

um minuto. A aplicação desenvolvida e o Endomondo registaram os mesmos valores de velocidade no decorrer do percurso, exceto no caso de acelerações ou paragens bruscas, onde foi notada uma ligeira vantagem na aplicação desenvolvida quanto ao tempo de resposta à variação da velocidade (Figura 68). Em comparação com o automóvel, é notória a diferença nas curvas do gráfico aquando do aumento ou diminuição repentino de velocidade; já quando a velocidade é constante os valores mantêm-se muito próximos. Este teste foi efetuado num automóvel onde foram, propositadamente, efetuados paragens e arranques bruscos da velocidade. Numa bicicleta o utilizador não consegue produzir a mesma variação de velocidade de uma forma tão rápida, pelo que é expectável que os valores da velocidade medidos pela aplicação desenvolvida mantenham-se mais próximos da velocidade real durante todo o trajeto.

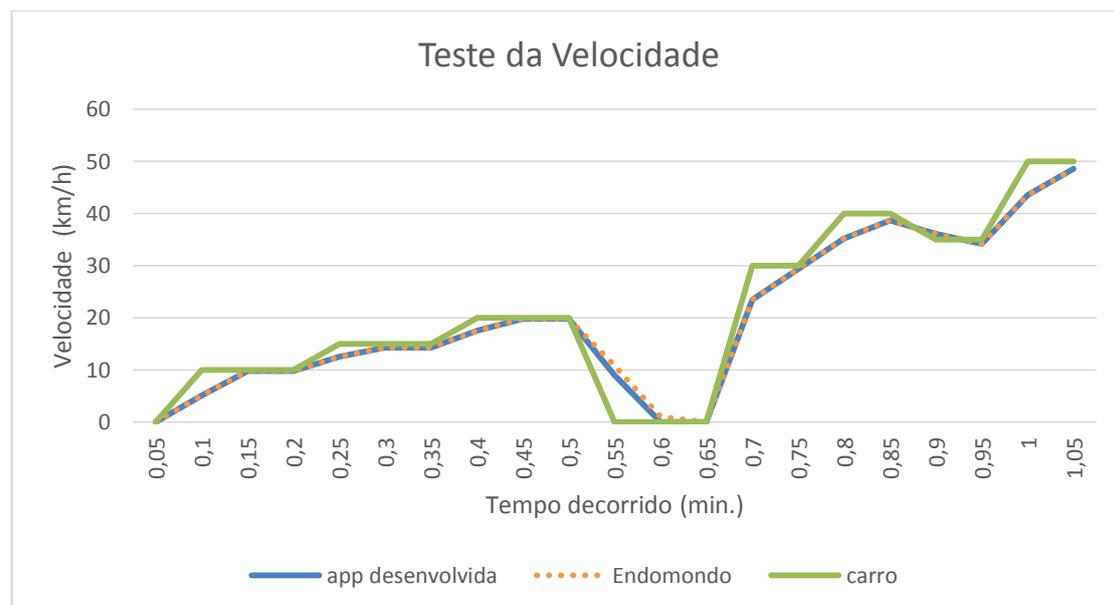


Figura 68 - Teste relativo à velocidade.

4.4 Teste da trepidação

Para que fosse possível a recolha dos valores da trepidação, foi necessário colocar um suporte bem fixo centrado no volante da bicicleta (Figura 69) e prender o smartphone ao suporte de forma que os valores fossem lidos com precisão. A bicicleta utilizada possuía uma suspensão frontal, com a possibilidade do seu bloqueio.



Figura 69 – Suporte utilizado para o smartphone no teste da trepidação.

Através da aceleração é possível observar variações entre os diferentes pisos (paralelepípedo e asfalto) com ou sem suspensão. A raiz do valor quadrático médio foi calculado após a subtração da aceleração da gravidade, sendo esta de $9,6 \text{ m/s}^2$ segundo a leitura do sensor. No teste referente ao paralelepípedo chega-se a obter amplitudes máximas na ordem dos 25 m/s^2 e uma raiz do valor quadrático médio de $6,9336 \text{ m/s}^2$ com a utilização da suspensão (Figura 70), enquanto sem a suspensão a amplitude máxima chega aos 30 m/s^2 e a raiz do valor quadrático médio foi de $7,8453 \text{ m/s}^2$ (Figura 71). Já no teste referente ao asfalto, com suspensão, a amplitude máxima obtida foi de $14,38 \text{ m/s}^2$ e a raiz do valor quadrático médio de $1,7888 \text{ m/s}^2$ (Figura 72), enquanto sem suspensão a amplitude máxima chegou aos $15,2 \text{ m/s}^2$ e a raiz do valor quadrático médio foi de $1,6062 \text{ m/s}^2$ (Figura 73). Pode-se concluir que, no percurso efetuado no asfalto, as diferenças entre ter a suspensão bloqueada ou desbloqueada não são tão notórias como no percurso efetuado em paralelepípedo, onde existe uma diferença considerável. Relativamente aos valores da trepidação entre os diferentes tipos de terreno existe uma diferença notável. O facto de no asfalto a raiz do valor quadrático médio ter sido superior com suspensão do que sem, ao contrário do que

se sucedeu no paralelepípedo, deve-se a um pequeno amortecimento do volante aquando da pedalada.

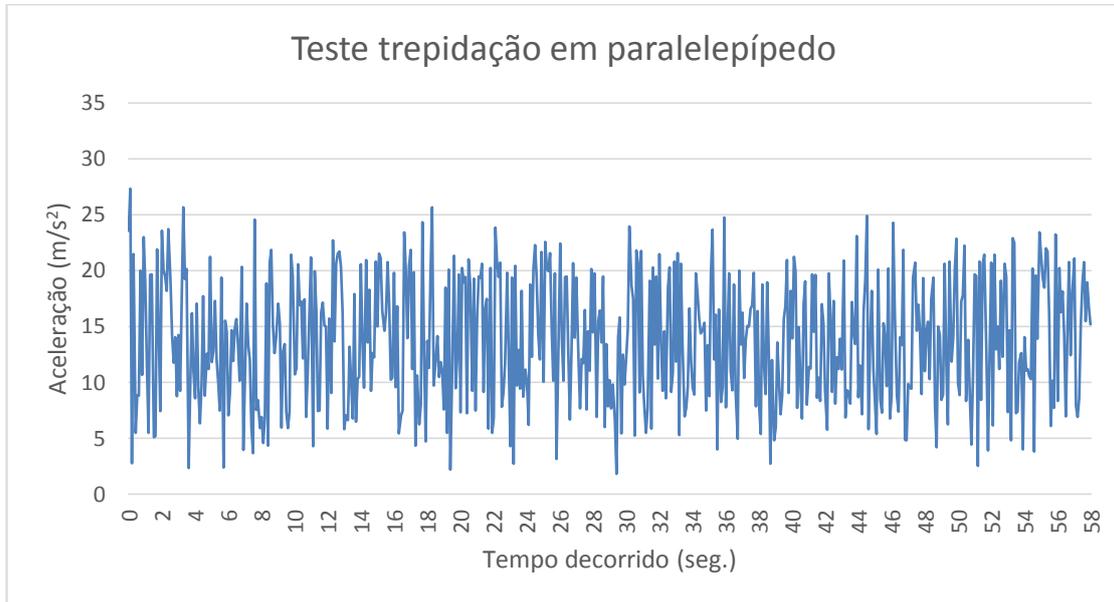


Figura 70 - Teste em paralelepípedo com suspensão a uma velocidade de 10 km/h durante 58 segundos.

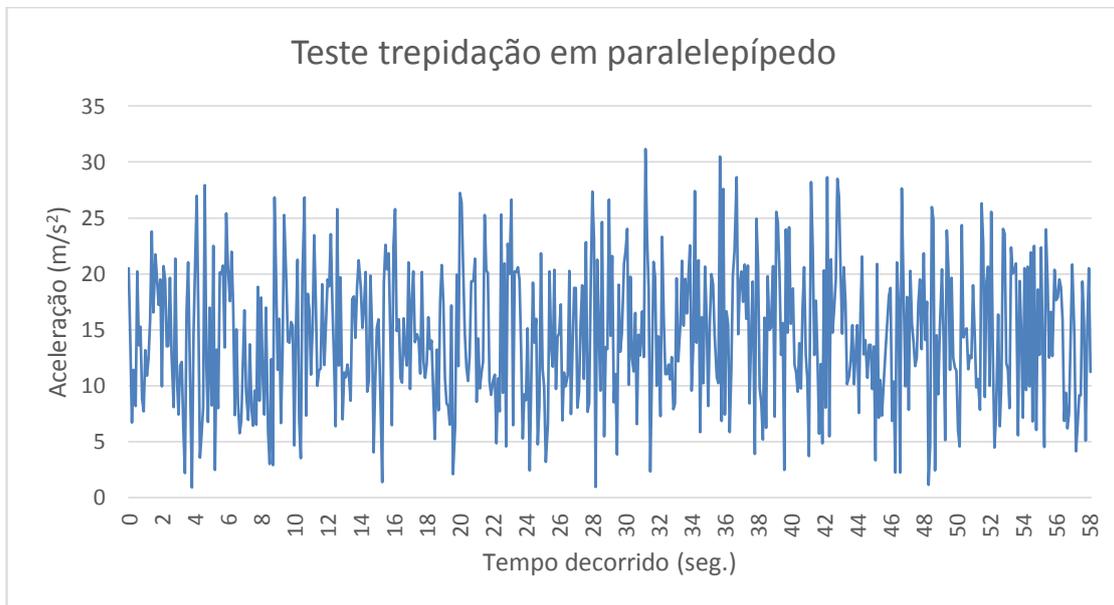


Figura 71 - Teste em paralelepípedo sem suspensão a uma velocidade de 10 km/h durante 58 segundos.

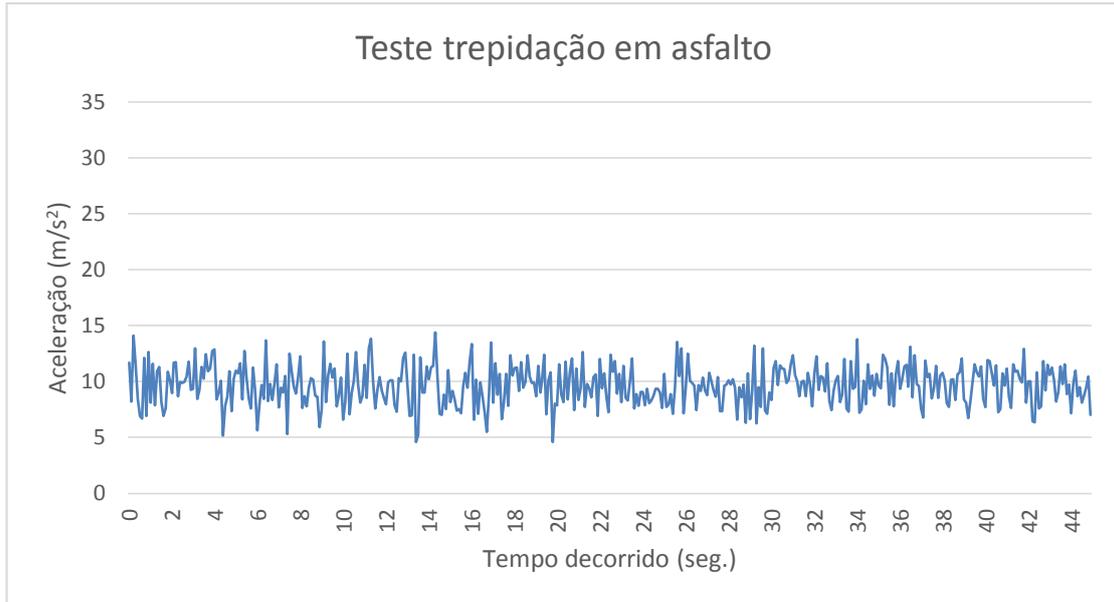


Figura 72 - Teste em asfalto com suspensão a uma velocidade de 10 km/h durante 45 segundos.

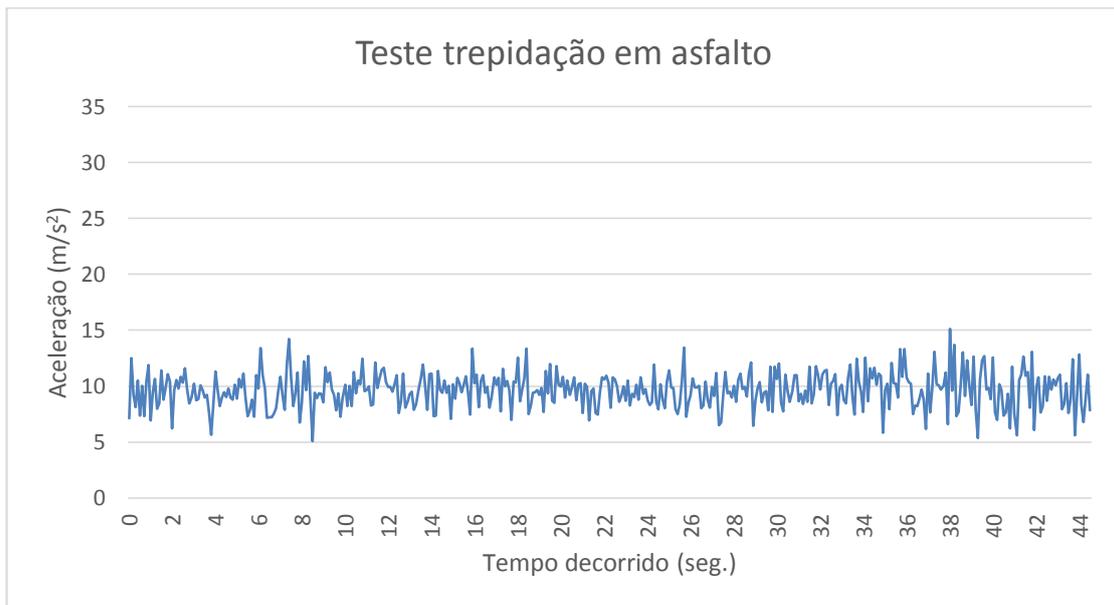


Figura 73 - Teste em asfalto sem suspensão a uma velocidade de 10 km/h durante 45 segundos.

4.5 Teste da navegação

No que concerne à navegação no mapa, primeiramente foi efetuado um teste para que fosse perceptível a importância da utilização do GPS assistido (A-GPS). O teste foi efetuado numa zona rodeada por edifícios com uma altura equivalente a três

andares, onde a conexão com os satélites era difícil. No primeiro percurso (Figura 74 à esquerda) não foi utilizado o A-GPS e, como se pode observar pelo traçado, as coordenadas encontram-se desviadas do que foram as coordenadas reais (sobre a estrada identificada a branco no centro). No segundo percurso (Figura 74 ao centro) foi utilizado o A-GPS, mas a determinada altura este foi desabilitado, sendo perceptível um desvio das coordenadas reais a meio do percurso, voltando a estabilizar aquando a ligação do A-GPS. No terceiro percurso (Figura 74 à direita) o percurso foi efetuado na totalidade com o A-GPS ativo, mantendo-se assim coordenadas bem mais próximas das reais.

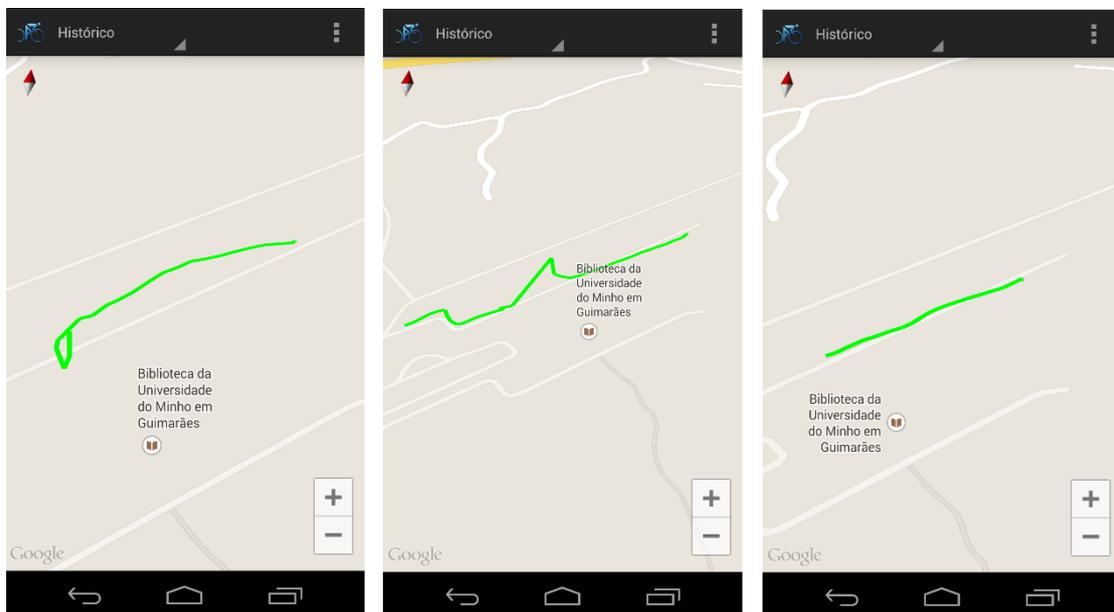


Figura 74 - Navegação sem A-GPS, com A-GPS interrompido e com A-GPS constantemente ligado.

A aplicação desenvolvida regista as coordenadas geográficas (latitude e longitude) a cada segundo (caso o utilizador se mantenha em movimento), para que seja possível, posteriormente, efetuar o desenho do trajeto. O Endomondo, no decorrer do percurso, desenha o trajeto, efetuando a ligação entre dois conjuntos de coordenadas geográficas, de sete em sete segundos. A decisão adotada na aplicação desenvolvida permite que a linha referente ao percurso não saia fora dos limites da estrada, principalmente a velocidades elevadas, quando o utilizador se encontra numa curva (Figura 75 à direita). Isto não se verifica frequentemente no Endomondo, uma vez que o tempo entre recolha de dados é maior (Figura 75 à esquerda). Uma das

desvantagens de uma atualização com um tempo tão curto é o facto de ser guardada mais informação. Como o utilizador tem a opção de sincronizar os seus dados com o servidor via Wi-Fi, não foi obstáculo ter uma maior quantidade de dados.

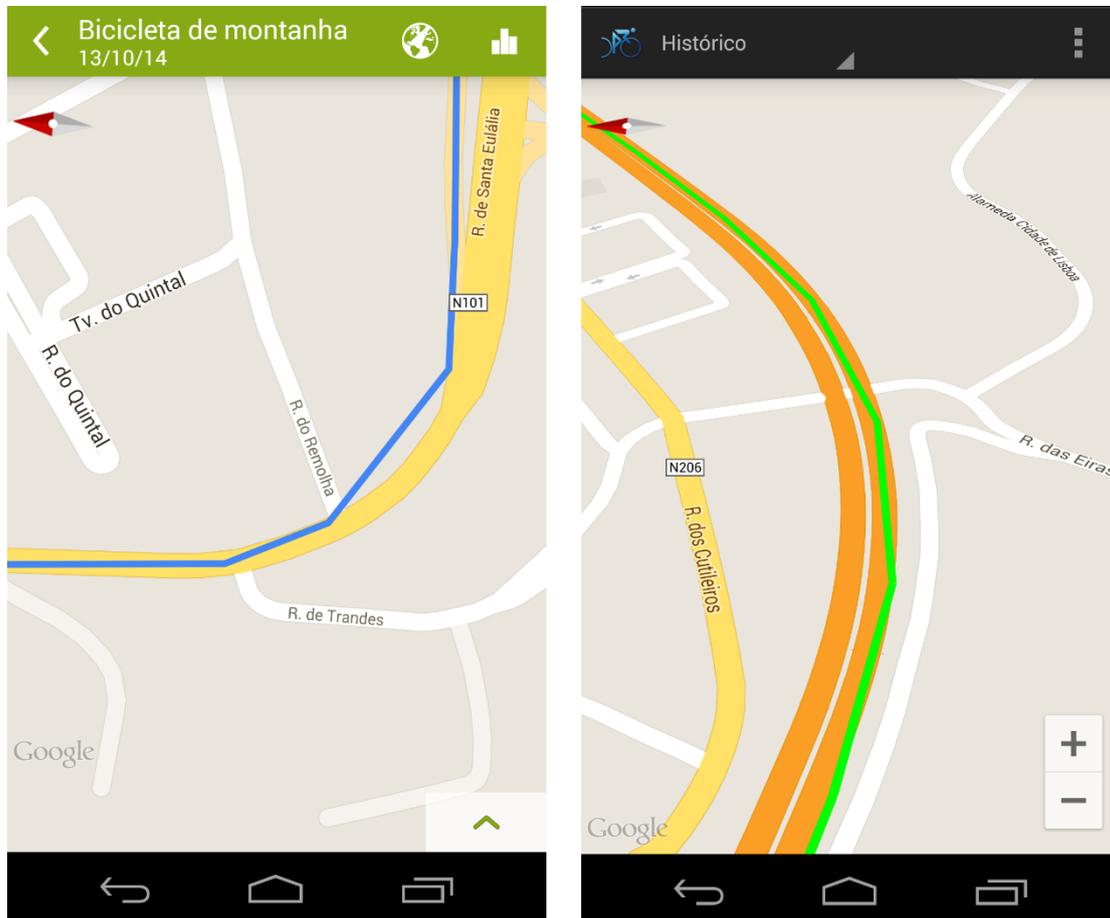


Figura 75 – Percurso Endomondo à esquerda e percurso da aplicação desenvolvida à direita.

É possível observar um trajeto efetuado, recorrendo a um gradiente de cores, para diferentes altitudes (Figura 77), sendo as cores mais escuras reservadas para as altitudes superiores e as cores mais claras para as altitudes inferiores. É obtida a altitude inicial e guardada como valor de referência (mantendo a mesma cor entre os -10 e os +10 metros). Existem 10 cores correspondentes ao gradiente de cor além da cor reservada ao valor de referência (Figura 76). A primeira mudança de cor, para altitudes superiores, ocorre acima dos 10 metros de altitude e as restantes acima dos 30, 50, 70 e 90 metros. Para altitudes inferiores a primeira mudança de cor ocorre abaixo dos -10 metros de altitude e as restantes abaixo dos -30, -50, -70 e -90 metros.

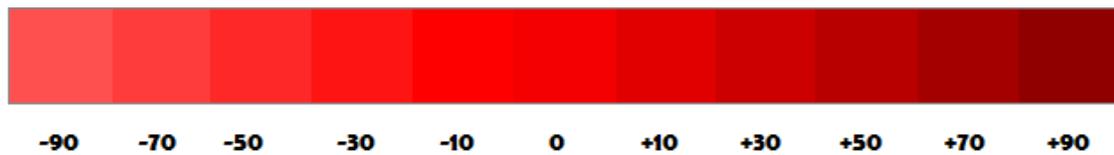


Figura 76 - Gradiente de cores utilizado para a altitude.

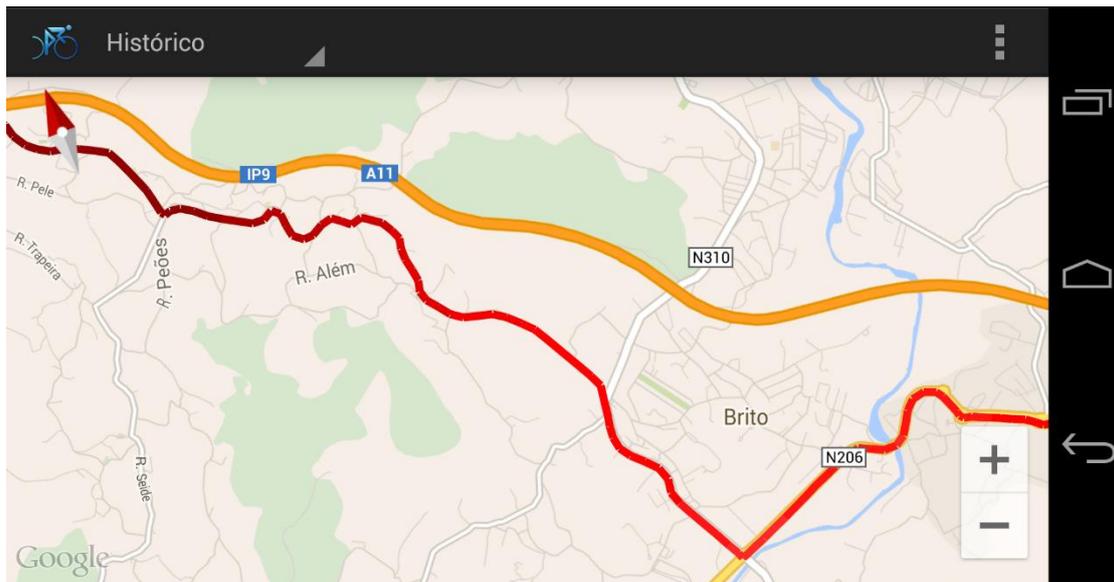


Figura 77 – Trajeto com recurso ao gradiente de cores para a altitude.

4.6 Bateria e dados móveis

Com o intuito de avaliar o desempenho da aplicação relativamente ao consumo de bateria e dados móveis, foi efetuado um percurso duas vezes, a primeira com a aplicação desenvolvida e a segunda com o Endomondo (versão 10.4.2).

No primeiro teste (Figura 78), com a bateria do smartphone a 100%, o percurso teve uma duração aproximada de 27 minutos. No final do percurso o smartphone encontrava-se ainda com 92% de bateria, onde, 44% da bateria gasta foi com a aplicação, ou seja, esta teve um consumo de 3,52%. O percurso foi efetuado durante o dia tendo exigido uma maior luminosidade do ecrã (cerca de 25% da capacidade máxima). Relativamente aos dados móveis consumidos durante o percurso (enviados e recebidos), estes foram no total 1,05 MB. Os 231 KB referentes aos dados móveis

enviados deve-se ao facto de, inicialmente ter sido executado o modo de navegação três vezes, efetuando um pedido ao servidor externo contendo as coordenadas geográficas. Por cada pedido ao servidor foram obtidas a pressão ao nível do mar e a temperatura. O pedido foi efetuado três vezes precisamente para uma perceção da influência do consumo de dados móveis enviados. O consumo de dados em tempo real deve-se ao facto da utilização do A-GPS e do download do mapa quando necessário (não estando armazenado em cache).



Figura 78 – Bateria e dados gastos durante um percurso com a aplicação desenvolvida.

No segundo teste (Figura 79) e com a bateria novamente a 100%, foi efetuado o mesmo percurso com a mesma duração, de aproximadamente 27 minutos. No final do percurso o smartphone encontrava-se com 90% de bateria, onde 56% da bateria foi gasta com o Endomondo, ou seja, este teve um consumo de 5,6%. O percurso foi efetuado durante a noite tendo exigido uma menor luminosidade do ecrã (cerca de 15% da capacidade máxima). Quanto aos dados móveis consumidos, estes foram no total 1,77 MB, sendo 1,59 MB para os dados móveis recebidos e 184 KB para os enviados.



Figura 79 - Bateria e dados gastos durante um percurso com a aplicação Endomondo.

Estes resultados indicam que a aplicação desenvolvida apresenta um menor consumo de energia da bateria que o Endomondo e, mesmo efetuando o pedido ao servidor externo três vezes no mesmo teste, o consumo total de dados móveis foi menor.

5. Conclusões

Tendo em conta os objetivos idealizados no início do projeto, é possível concluir que estes foram superados com sucesso. Existem, no entanto, diversas funcionalidades que podem acrescentar, futuramente, um maior valor ao projeto.

Tal como proposto, foi desenvolvida uma aplicação Android capaz de monitorizar os dados referentes a um trajeto efetuado por um ciclista (latitude, longitude, altitude, velocidade, distância e duração). Sendo a primeira vez que foi desenvolvida uma aplicação para dispositivos móveis, teve de existir um processo de habituação à programação em Android, processo este facilitado devido aos conhecimentos de Java adquiridos previamente. Foi desenvolvido com sucesso um website permitindo o acesso a toda a informação referente ao utilizador e possíveis amigos. Configurou-se um servidor Apache capaz de responder aos requisitos necessários para o funcionamento do sistema. Surgiu a necessidade de encontrar uma forma de manter os dados o mais atualizados possível no servidor, sem que recursos como dados móveis (3G) fossem desperdiçados, sendo então utilizado o Wi-Fi para a sincronização de dados menos críticos temporalmente.

Devido ao fato do AVD exigir demasiados recursos de processamento, optou-se por correr a aplicação diretamente no dispositivo móvel. Esta escolha permitiu que o código desenvolvido fosse testado no smartphone de uma forma muito mais rápida em comparação com um dispositivo virtual. A utilização de um smartphone produzido pela própria Google permitiu tirar um maior partido do sistema operativo, nomeadamente a utilização da última versão (4.4 KitKat) desde o início do desenvolvimento da aplicação.

Quanto aos testes efetuados, recorreu-se sempre ao mesmo dispositivo, pois devido aos diferentes fabricantes e diferentes gamas existiam variações dos valores obtidos. Este problema foi observado tanto na aplicação desenvolvida como em todas

as outras aplicações testadas. Foram observadas melhorias entre a aplicação desenvolvida e a utilizada para comparação a nível de consumo de bateria e consumo de dados móveis (3G), e obteve-se um traçado do percurso no mapa com maior precisão. Foram efetuados testes de viabilidade para o caso de não ser possível a utilização constante do consumo de dados móveis (3G), mas quando eram percorridos caminhos rodeados por edifícios altos (mais de três andares) os valores (latitude e longitude) obtidos apresentavam uma elevada variação quando comparados com o valor real. Optou-se então pelo uso contínuo do 3G durante um percurso.

5.1 Trabalho futuro

Relativamente à aplicação móvel, os parâmetros apresentados ao utilizador enquanto este se encontra no modo de navegação são fixos, sem que este possa optar apenas por aqueles que lhe são relevantes. Uma funcionalidade interessante seria o utilizador ter a opção de selecionar apenas os parâmetros que lhe são úteis.

Quando o utilizador se encontra no modo de navegação o 3G encontra-se sempre ativo. Para que fosse possível poupar bateria do smartphone, seria interessante o estudo de um algoritmo capaz de desabilitar o 3G enquanto o utilizador se encontra parado, voltando a habilitá-lo assim que este retomasse o seu percurso, sem que isto afetasse o funcionamento da aplicação.

O utilizador tem acesso a trajetos percorridos por ele ou pelo seu grupo de amigos, sendo que apenas é apresentado o gradiente de cores para a altitude. Seria interessante a utilização de gradientes de cores para outras variáveis, como a velocidade, a cadência e o nível de bateria da bicicleta, dando a opção ao utilizador de escolher qual o que pretende visualizar.

O utilizador tem apenas acesso a um gráfico referente aos quilómetros percorridos e um outro referente ao nível de bateria da bicicleta, sendo este último simulado devido à leitura do nível de bateria da bicicleta fazer parte do trabalho de outro investigador. Seria interessante a integração de ambas as aplicações, podendo

assim obter valores reais e possibilitar a utilização de outras variáveis a serem representadas graficamente.

Uma funcionalidade interessante a ser adicionada é, enquanto o utilizador se encontra parado e no modo de navegação, ter a possibilidade de inserir pontos de interesse com informação à sua rota, podendo mais tarde outros utilizadores visualizar os mesmos.

Outra variável interessante de se trabalhar seria a frequência cardíaca do utilizador, podendo ser apresentada na tela principal enquanto este efetua o seu percurso ou ser apresentada graficamente no mapa com recurso ao gradiente de cores.

Foi testada, numa aplicação criada à parte, uma forma de alertar o utilizador sempre que este se encontre a uma distância pré-definida de um marcador associado a um utilizador do seu círculo de amigos. Quando o ponto azul que simboliza a posição do utilizador entra no raio a vermelho em volta do marcador, é enviado um alerta ao utilizador (Figura 80). Um próximo passo seria integrar a funcionalidade de localizar amigos com o modo de navegação, permitindo assim que no decorrer do percurso o utilizador fosse notificado e pudesse observar no mapa de quem se estava a aproximar. Outra vantagem seria um sistema de partilha de mensagens, onde o utilizador pudesse notificar o seu amigo que se dirige ao seu encontro ou perguntar a sua disponibilidade para se juntarem no mesmo percurso.

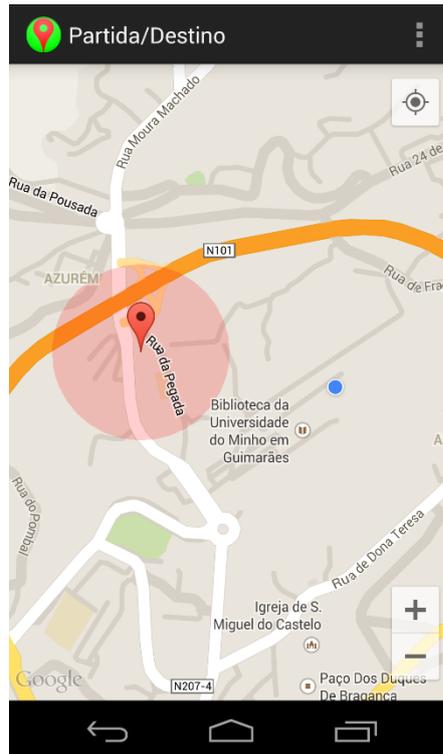


Figura 80 – Alerta de proximidade de amigo.

Cada vez mais as aplicações encontram-se de alguma forma conectadas com as redes sociais, uma funcionalidade possível seria a partilha de circuitos efetuados com o consentimento do utilizador.

Não foi abordado qualquer tipo de mecanismo para recuperação de password, estando o utilizador impossibilitado de recuperar a sua em caso de esquecimento. Uma possibilidade era a recuperação da password através do seu envio para o endereço de email introduzido no registo.

No que diz respeito aos valores da trepidação obtidos, o próximo passo seria o reconhecimento do tipo de terreno, ficando um vasto trabalho a se desenvolver devido à sua complexidade. A próxima etapa seria encontrar um padrão característico para cada um dos diferentes tipos de terreno, averiguar se diferentes velocidades têm algum impacto nesse padrão e verificar a necessidade de uma calibragem prévia do smartphone devido à possibilidade dos valores obtidos dos sensores oscilarem de dispositivo para dispositivo.

Referências

- [1] N.Lane, M. Mohammad, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, A. Campbell, "BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing", 5th International ICST Conference on Pervasive Computing Technologies for Healthcare, Paris, France, 2011.
- [2] T. Tucker, "Why Ride An Ebike? Why Not Ride An Ebike?", [Online] Available: <http://electricvehicle.ieee.org/2014/05/15/ride-ebike-ride-ebike> [consulted on August 2014].
- [3] Endomondo, "Endomondo Sports Tracker", [Online] Available: <http://www.endomondo.com/about> [consulted on August 2014].
- [4] B.iCycle, "B.iCycle GPS Cycling Computer", [Online] Available: <http://www.bicycle.com/android/home.html> [consulted on August 2014].
- [5] Sports Tracker, "Sports Tracker", [Online] Available: <http://www.sports-tracker.com/blog/about> [consulted on August 2014].
- [6] M. Singhal, A. Shukla, "Implementation of Location based Services in Android using GPS and Web Services", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.
- [7] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn, and A. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping", 5th International Conference on Embedded Networked Sensor Systems, Sydney, Australia, 2007.
- [8] H. Jones and L. Forslof, "Roadroid Continuous Road Condition Monitoring With Smart Phones", 17th IRF World meeting, Saudi Arabia, 2014.
- [9] Roadroid, "Roadroid", [Online] Available: <http://www.roadroid.com/Map> [consulted on September 2014].

-
- [10] E. T. Coelho, P. Carvalhal, M. J. Ferreira, L. F. Silva, H. Almeida, C. Santos, J. A. Afonso, "A Bluetooth-based wireless distributed data acquisition and control system", Proceedings of IEEE International Conference on Robotics and Biomimetics - ROBIO 2006, Kunming, China, December 2006.
- [11] H. F. López, J. A. Afonso, J. H. Correia, R. Simões, "HM4All: A Vital Signs Monitoring System based in Spatially Distributed ZigBee Networks", Proceedings of 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2010), Munchen, Germany, March 2010.
- [12] B. Fernades, J. Afonso, R. Simões, "Vital Signs Monitoring and Management using Mobile Devices", 6th Iberian Conference on Information Systems and Technologies (CISTI), Chaves, Portugal, June 2011.
- [13] Google, "Android Developers", [Online] Available: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels> [consulted on November 2013].
- [14] Google, "Android Developers", [Online] Available: <http://developer.android.com/reference/android/app/Activity.html> [consulted on November 2013].
- [15] Google, "Android Developers", [Online] Available: <http://developer.android.com/guide/components/fragments.html> [consulted on May 2014].
- [16] ORACLE, "MySQL", [Online] Available: <http://www.mysql.com> [consulted on September 2014].
- [17] ORACLE, "MySQL", [Online] Available: <http://dev.mysql.com/doc> [consulted on September 2014].
- [18] ORACLE, "MySQL", [Online] Available: <http://dev.mysql.com/doc/refman/5.0/en/features.html> [consulted on September 2014].
- [19] Google, "Android", [Online] Available: <http://www.android.com> [consulted on

- September 2014].
- [20] Apple, “iOS7”, [Online] Available: <https://www.apple.com/pt/ios/features> [consulted on September 2014].
- [21] Microsoft, “Windows Phone”, [Online] Available: <http://www.windowsphone.com/pt-pt/features> [consulted on September 2014].
- [22] Statista, “Global smartphone sales by operating system 2009-2013”, [Online] Available: <http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system> [consulted on September 2014].
- [23] SQLite, “SQLite”, [Online] Available: <http://www.sqlite.org/about.html> [consulted on September 2014].
- [24] SQLite, “SQLite”, [Online] Available: <http://www.sqlite.org/news.html> [consulted on September 2014].
- [25] The PHP Group, “PHP”, [Online] Available: <https://php.net/manual/en/intro-what-is.php> [consulted on September 2014].
- [26] The PHP Group, “PHP”, [Online] Available: <https://php.net/manual/en/intro-whatcando.php> [consulted on September 2014].
- [27] MDN, “JavaScript”, [Online] Available: <https://developer.mozilla.org/pt-PT/docs/Web/JavaScript> [consulted on September 2014].
- [28] MDN, “JavaScript”, [Online] Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript [consulted on September 2014].
- [29] W3Schools.com, “HTML”, [Online] Available: http://www.w3schools.com/html/html_intro.asp [consulted on September 2014].
- [30] W3C, “XML”, [Online] Available: <http://www.w3.org/standards/xml/core#summary> [consulted on September 2014].

-
- [31] Ullas Nambiar, Zoé Lacroix, Stéphane Bressan, Mong Li Lee, Ying Guang Li, “Efficient XML Data Management: An Analysis”, EC-Web 2002, Berlin, 2002.
- [32] Microsoft, “Porting Android (Java) applications to Qt”, [Online] Available: [http://developer.nokia.com/community/wiki/Archived:Porting_Android_\(Java\)_applications_to_Qt](http://developer.nokia.com/community/wiki/Archived:Porting_Android_(Java)_applications_to_Qt) [consulted on September 2014].
- [33] L. Damas, “SQL - Structured Query Language” (12th edition), FCA, 2005.
- [34] M. Sayers, T. Gillespie, C. Queiroz, “The international road roughness experiment: establishing correlation and a calibration standard for measurements”, Technical Paper No. 45, World Bank, Washington DC, 1986.
- [35] Google, “Android Developers”, [Online] Available: <http://developer.android.com/guide/topics/connectivity/nfc/index.html> [consulted on September 2014].
- [36] Google Play, “NFC Tasks”, [Online] Available: <https://play.google.com/store/apps/details?id=com.wakdev.wdnfc> [consulted on September 2014].
- [37] Google Play, “NFC Tools”, [Online] Available: <https://play.google.com/store/apps/details?id=com.wakdev.nfctasks> [consulted on September 2014].
- [38] OpenStreetMap, “OpenStreetMap”, [Online] Available: <http://www.openstreetmap.org/> [consulted on March 2014].
- [39] Google, “Android Developers”, [Online] Available: <http://developer.android.com/google/play-services/maps.html> [consulted on March 2014].
- [40] Bootstrap, “Bootstrap”, [Online] Available: <http://getbootstrap.com/> [consulted on September 2014].
- [41] D. AbdElminaam, H. Kader, M. Hadhoud, S. El-Sayed, “GPS Test Performance: Elastic Execution Applications between Mobile Device and Cloud to Reduce Power Consumption”, IJCSNS International Journal of Computer Science and

-
- Network Security, VOL.13 No.12, Egypt, December 2013.
- [42] P. Ratsameethammawong, M. Kasemsan, “Mobile Phone Location Tracking by the Combination of GPS, Wi-Fi and Cell Location Technology”, Communications of the IBIMA, Thailand, 2010.
- [43] Google, “API Javascript Google Maps v3”, [Online] Available: <https://developers.google.com/maps/documentation/javascript/> [consulted on September 2014].
- [44] Google, “Google Charts”, [Online] Available: <https://developers.google.com/chart/interactive/docs/index> [Consulted on September 2014].
- [45] J. Stott, “EarthTools”, [Online] Available: <http://www.earthtools.org/> [consulted on October 2014].
- [46] Google, “Google Earth”, [Online] Available: <https://www.google.com/earth/> [consulted on October 2014].
- [47] USGS, “Shuttle Radar Topography Mission”, [Online] Available: <http://srtm.usgs.gov/index.php>, [consulted on October 2014].
- [48] Google Play, “OsmAnd”, [Online] Available: https://play.google.com/store/apps/details?id=net.osmand&hl=pt_PT [consulted on October 2014].
- [49] Google Play, “Phone Tester”, [Online] Available: https://play.google.com/store/apps/details?id=net.osmand&hl=pt_PT [consulted on October 2014].
- [50] G. Xu, “GPS Theory, Algorithms and Applications”, (2nd edition), Springer, 2007.
- [51] J. McNamara, “GPSSS for DUMMIES”, (1st edition), Wiley Publishing, Inc., 2004.
- [52] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, M. Srivastava “Biketastic: Sensing and Mapping for Better Biking”, CHI 2010: Bikes and Buses, USA, April 2010.

-
- [53] P. Wu, J. Zhu, J. Zhang, “MobiSens: A Versatile Mobile Sensing Platform for Real-World Applications”, *Journal Mobile Networks and Applications*, Volume 18 Issue 1, Pages 60-80, February 2013.
- [54] J. Ballesteros, B. Carbunar, M. Rahman, N. Rische, S. Iyengar, “Towards Safe Cities: A Mobile and Social Networking Approach”, *Parallel and Distributed Systems*, *IEEE Transactions*, Volume: 25 Issue 9, Pages 2451 – 2462, August 2013.
- [55] W. Khan, Y. Xiang, M. Aalsalem, Q. Arshad, “Mobile Phone Sensing Systems: A Survey”, *IEEE Communications Surveys & Tutorials*, Volume: 15, Issue 1, Pages 402 – 427, February 2013.
- [56] The 4ViewSoft Company, “AChartEngine”, [Online] Available: <http://www.achartengine.org/> [consulted on September 2014].
- [57] Google, “Android Developers”, [Online] Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html [consulted on October 2014].

Apêndice A – Principais classes Android

Tabela 3 - Principais classes utilizadas no desenvolvimento da aplicação.

Classe	Descrição
SharedPreferences	Providencia uma <i>framework</i> geral que permite guardar e recuperar valores-chave de tipos de dados primitivos (boolean, float, int, long e string). Estes dados persistem, mesmo que a aplicação seja encerrada ou o processo terminado.
ViewPager	Gestor de <i>layout</i> que permite ao utilizador navegar para a esquerda e para a direita através de páginas contendo dados. Utilizada maioritariamente em conjunto com fragments.
PagerAdapter	Classe base que fornece o adaptador para preencher páginas dentro de um ViewPager. Foram utilizadas implementações mais específicas como FragmentPagerAdapter ou FragmentStateAdapter.
ActionBar	Opção de janela no topo da Activity que pode conter o título, modos de navegação e outros itens interativos.
LocationManager	Providencia o acesso aos serviços de localização do sistema. Permite a aplicação obter atualizações periódicas da localização geográfica do dispositivo.
Location	Classe de dados representando a localização geográfica. A localização pode consistir na latitude, longitude, tempo e outras informações como a velocidade.
GoogleMap	Classe principal da API do Google Maps, sendo o ponto de acesso a todos os métodos relacionados com o mapa.
SupportMapFragment	Permite, de uma forma simples, colocar um mapa numa aplicação recorrendo ao método <code>getMap()</code> .
Polyline	Lista de pontos, onde os segmentos de linha são traçados entre pontos consecutivos.
SensorManager	Permite o acesso a todos os sensores disponíveis no dispositivo.
Sensor	Classe representativa de um sensor.
AsyncTask	Permite realizar operações em background e publicar os resultados sem a necessidade de manipular threads.

Tabela 4 – Métodos da classe AsyncTask.

AsyncTask	
onPreExecute()	Invocado antes da tarefa ser executada. Por exemplo para atribuir valores a variáveis antes de executar o doInBackground().
doInBackground()	Invocado na thread de background, depois do onPreExecute() terminar a sua execução. Utilizado para computação em background que demore um longo período de tempo.
onProgressUpdate()	Utilizado para apresentar informação enquanto a computação em background continua em execução.
onPostExecute()	Quando a computação em background termina, recebe o resultado como parâmetro.

Apêndice B – Protocolo de comunicação

Login()

HTTP_POST	
campo	valor
email	"email do utilizador"
password	"password do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição

Verifica se o email corresponde à password e notifica em caso de sucesso ou insucesso.

Registo()

HTTP_POST	
campo	valor
email	"email do utilizador"
nome	"nome do utilizador"
peso	"peso do utilizador"
altura	"altura do utilizador"
idade	"idade do utilizador"
password	"password do utilizador"
genero	"género do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição

Insere os dados na BD e notifica em caso de sucesso ou insucesso.

ListarTelemovelEmergencia()

HTTP_POST	
campo	valor
email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
num_emergencia	"telemóvel de emergência"

Descrição

Devolve o número de telemóvel de emergência associado ao utilizador.

UpdateTelemovelEmergencia()

HTTP_POST	
campo	valor
email	"email do utilizador"
num_emergencia	"telemóvel de emergência"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição

Efetua a inserção/update do número de telemóvel de emergência na BD e notifica em caso de sucesso ou insucesso.

CriaEvento()	
HTTP_POST	
campo	valor
nome	"nome do evento"
cidade	"cidade onde se realiza"
duracao	"duração do evento"
distancia	"distância total"
data	"data de realização"
hora	"hora de início"

ListarMeusEventos()	
HTTP_POST	
campo	valor
email	"email do utilizador"

ListarMeusEventosInfo()	
HTTP_POST	
campo	valor
nome	"nome do evento"

ListarTodosEventos()	
HTTP_POST	
campo	valor
data	"data atual"

ListarEventoInfo()	
HTTP_POST	
campo	valor
nome	"nome do evento"

AdereEvento()	
HTTP_POST	
campo	valor
email	"email do amigo"
eventos_nome	"nome do evento"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Insere os dados na BD e notifica em acaso de sucesso ou insucesso.

HTTP_RESPONSE(JSON)	
campo	valor
eventos_nome	"nome dos eventos do utilizador"

Descrição
 Devolve o nome de todos os eventos associados ao email do utilizador.

HTTP_RESPONSE(JSON)	
campo	valor
cidade	"cidade do evento"
data	"data de realização"

Descrição
 Devolve a cidade e a data do evento.

HTTP_RESPONSE(JSON)	
campo	valor
nome	"nome do evento"
cidade	"cidade do evento"
data	"data de realização"

Descrição
 Devolve nome, cidade e data dos eventos (superior ou igual à data atual).

HTTP_RESPONSE(JSON)	
campo	valor
nome	"nome do evento"
cidade	"cidade do evento"
duracao	"duração total"
distancia	"distância total"
data	"data de realização"
hora	"hora de realização"

Descrição
 Devolve toda a informação referente ao evento.

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Devolve se o evento ficou associado ao utilizador com sucesso ou não.

VerificaExiste()	
HTTP_POST	
campo	valor
email	"email do amigo"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Devolve se o amigo se encontra registado na BD ou não.

AdicionaAmigo()	
HTTP_POST	
campo	valor
email_amigo	"email do amigo"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Devolve se o amigo foi introduzido na tabela amigos com sucesso ou não.

RelacionaAmigo()	
HTTP_POST	
campo	valor
email_amigo	"email do amigo"
utilizador_email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Devolve se o amigo foi relacionado com o utilizador com sucesso ou não.

ListarMeusAmigos()	
HTTP_POST	
campo	valor
email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
email_amigo	"email do amigo"

Descrição
 Devolve o email de todos os amigos associados ao utilizador.

ListarInfoMeusAmigos()	
HTTP_POST	
campo	valor
email	"email do amigo"

HTTP_RESPONSE(JSON)	
campo	valor
nome	"nome do amigo"
email	"email do amigo"
peso	"peso do amigo"
altura	"altura do amigo"
idade	"idade do amigo"

Descrição
 Devolve a informação referente ao amigo.

RemoverAmigos()	
HTTP_POST	
campo	valor
email_amigo	"email do amigo"
utilizador_email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
resposta	"OK ou NOK"

Descrição
 Devolve se o amigo ficou desassociado do utilizador com sucesso ou não.

NomeMeusAmigos()	
HTTP_POST	
campo	valor
email	"email do amigo"

HTTP_RESPONSE(JSON)	
campo	valor
nome	"nome do amigo"

Descrição
 Devolve o nome de todos os amigos associados ao utilizador.

ListarTrajetoAmigos()	
HTTP_POST	
campo	valor
email	"email do amigo"

HTTP_RESPONSE(JSON)	
campo	valor
data	"data"
duracao	"duração total"
distancia	"distância total"
id_trajeto	"id do trajeto"

Descrição
 Devolve data, duração, distância e o id dos trajetos do amigo do utilizador.

ListarTrajetoAmigosInfo()	
HTTP_POST	
campo	valor
id_trajeto	"id do trajeto"

HTTP_RESPONSE(JSON)	
campo	valor
data	"data"
duracao	"duração total"
distancia	"distância total"
velocidade_media	"velocidade média"
velocidade_maxima	"velocidade média"
trepidacao_media	"rugosidade média"
hora	"hora"

Descrição
 Devolve a informação associada a um trajeto.

ListarInfoUser()	
HTTP_POST	
campo	valor
email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
nome	"nome do utilizador"
email	"email do utilizador"
peso	"peso do utilizador"
altura	"altura do utilizador"
idade	"idade do utilizador"
url_foto	"url da foto do utilizador"
genero	"género do utilizador"

Descrição
 Devolve informação referente ao utilizador.

ListarInfoTrajeto()	
HTTP_POST	
campo	valor
id_trajeto	"id do trajeto"

HTTP_RESPONSE(JSON)	
campo	valor
latitude	"latitude"
longitude	"longitude"
altitude	"altitude"
velocidade	"velocidade"
hora	"hora"

Descrição
 Devolve informação referente ao decorrer de todo o trajeto.

LocalizarAmigos()

HTTP_POST	
campo	valor
email	"email do amigo"

HTTP_RESPONSE(JSON)

campo	valor
nome	"nome do amigo"
data	"data"
hora	"hora"
latitude	"latitude"
longitude	"longitude"

Descrição

Devolve informação referente à última localização conhecida do amigo.

LocalizarAmigosInsere()

HTTP_POST	
campo	valor
utilizador_email	"email do utilizador"
data	"data"
hora	"hora"
latitude	"latitude"
longitude	"longitude"

HTTP_RESPONSE(JSON)

campo	valor
resposta	"OK ou NOK"

Descrição

Devolve se os dados referentes à localização do utilizador foram introduzidos na BD com sucesso ou não.

DevolveID()

HTTP_POST	
campo	valor

HTTP_RESPONSE(JSON)

campo	valor
id_trajeto	"id do trajeto"

Descrição

Devolve o último id do trajeto inserido na BD.

InsereTrajeto()

HTTP_POST	
campo	valor
id	"id do trajeto"
email	"email do utilizador"
distancia	"distância total"
duracao	"duração total"
velocidade_media	"velocidade média"
velocidade_maxima	"velocidade máxima"
data	"data do trajeto"
trepidacao_media	"trepidação média"

HTTP_RESPONSE(JSON)

campo	valor
resposta	"OK ou NOK"

Descrição

Inserir os dados relativos ao trajeto na BD e notificar em caso de sucesso ou insucesso.

InsereLocalizacao()

HTTP_POST	
campo	valor
id_trajeto	"id do trajeto"
latitude	"latitude"
longitude	"longitude"
altitude	"altitude"
velocidade	"velocidade"
trepidacao	"trepidação"
hora	"hora"

HTTP_RESPONSE(JSON)

campo	valor
resposta	"OK ou NOK"

Descrição

Inserir os dados relativos à localização na BD e notificar em caso de sucesso ou insucesso.

ListarKm()	
HTTP_POST	
campo	valor
email	"email do utilizador"

HTTP_RESPONSE(JSON)	
campo	valor
janeiro	"total de km de janeiro"
fevereiro	"total de km de fevereiro"
março	"total de km de março"
abril	"total de km de abril"
maio	"total de km de maio"
junho	"total de km de junho"
julho	"total de km de julho"
agosto	"total de km de agosto"
setembro	"total de km de setembro"
outubro	"total de km de outubro"
novembro	"total de km de novembro"
dezembro	"total de km de dezembro"

Descrição
Devolve o total de km efetuados em cada mês.

Apêndice C – NFC

NFC Tools

NFC Tools é uma aplicação utilizada para a leitura e escrita de uma *tag*. Na Figura 81 pode-se observar os passos a seguir para a escolha de uma aplicação a ser escrita numa *tag*. No canto superior esquerdo é escolhida a opção de escrita, no campo superior direito é escolhida a opção referente às aplicações, no canto inferior esquerdo é introduzido o nome do *package* da aplicação e finalmente no canto inferior direito é listada a nossa aplicação, bastando agora selecioná-la.

Na Figura 82 pode-se observar na parte superior as opções selecionadas a serem escritas para a *tag*, no canto inferior esquerdo é pedido ao utilizador que aproxime a *tag* para que possa ser efetuado o processo de escrita e no canto inferior direito obtém-se a confirmação que a escrita foi efetuada com sucesso.

NFC Tasks

NFC Tasks é uma aplicação adicional para a execução das tarefas, onde é possível escolher se queremos ser notificados ou não da ação acionada, caso a escolha seja ser notificado, existe a possibilidade de ser através de uma janela de *pop-up* ou pela barra de notificações. Escolhendo a notificação através de uma janela de *pop-up* existem três opções de visualização no ecrã: no topo, no centro ou em baixo. Esta aplicação possui um contador de leitura, permitindo ao utilizador saber quantas vezes executou uma tarefa recorrendo à leitura de uma *tag* com o seu smartphone.

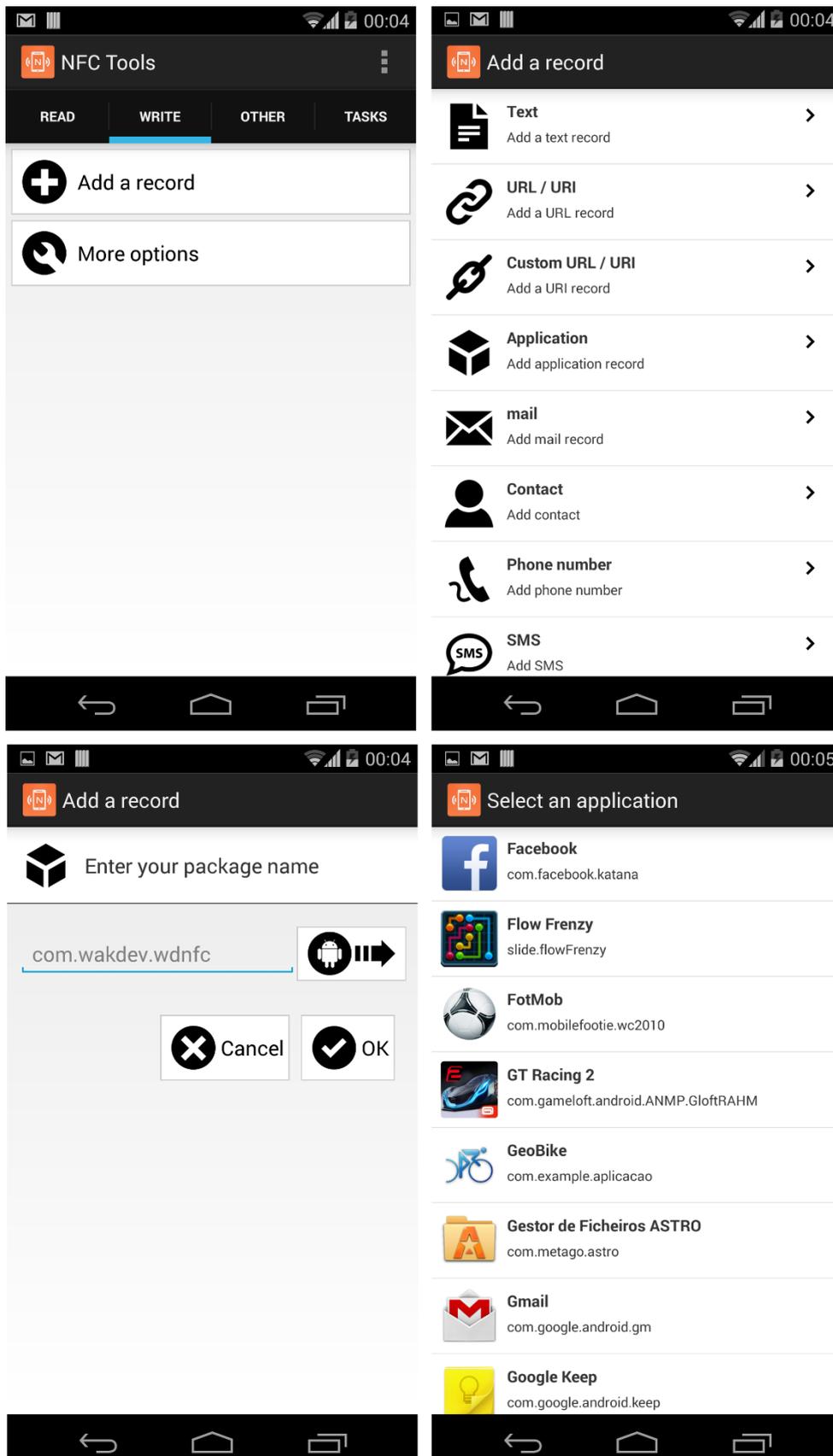


Figura 81 – Seleção de uma aplicação a ser escrita numa *tag*.

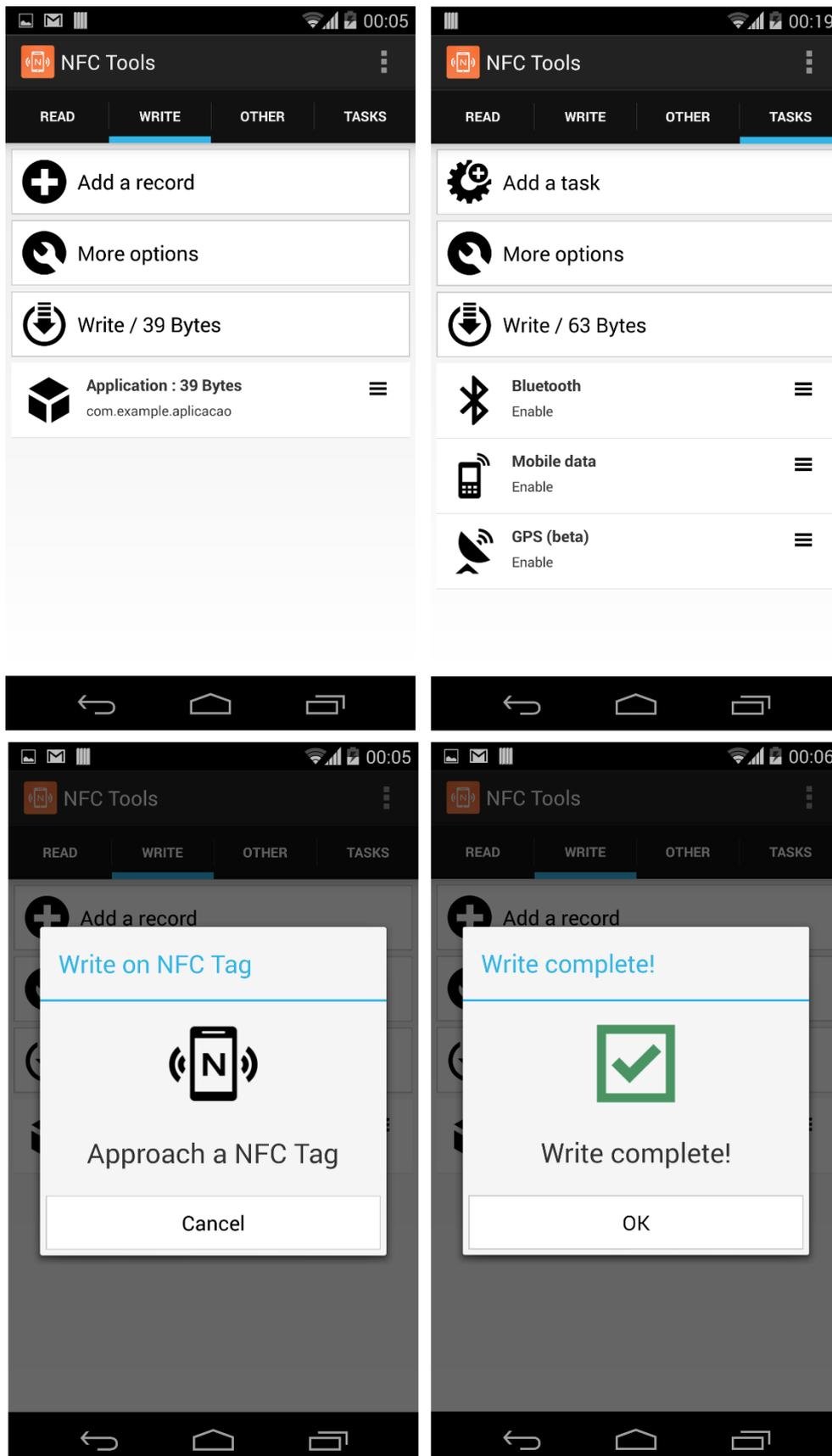


Figura 82 – Processo de escrita numa tag.