



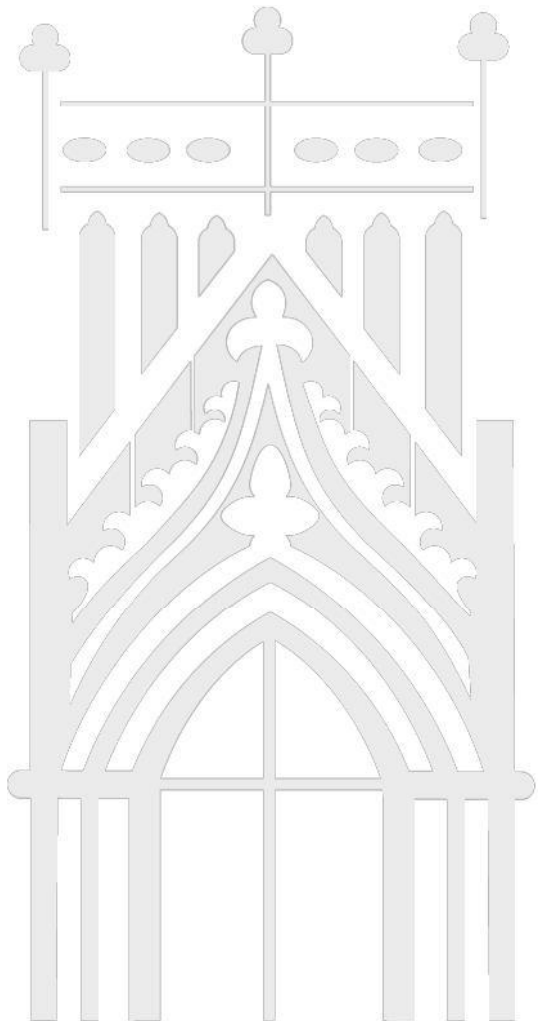
**IPG** Politécnico  
|da|Guarda  
Polytechnic  
of Guarda

Mestrado em Computação Móvel

Roadbook Digital com Detecção de Contextos  
para dispositivos móveis android

Nuno Miguel Pereira Martins

julho | 2015



Escola Superior  
de Tecnologia  
e Gestão



Instituto Politécnico da Guarda  
Escola Superior de Tecnologia e Gestão  
Av. Dr. Francisco Sá Carneiro nº50, 6300-559, Guarda, Portugal

***Roadbook Digital com Detecção de Contextos  
para Dispositivos Móveis Android***

Projeto Aplicado do Mestrado em Computação Móvel

**Nuno Miguel Pereira Martins**

**julho | 2015**



Instituto Politécnico da Guarda  
Escola Superior de Tecnologia e Gestão  
Av. Dr. Francisco Sá Carneiro nº50, 6300-559, Guarda, Portugal

***Roadbook Digital com Detecção de Contextos  
para Dispositivos Móveis Android***

Relatório de Projeto Aplicado submetido como requisito parcial para obtenção do grau de Mestre em Computação Móvel

Orientador: Professor Doutor Carlos Carreto

**Nuno Miguel Pereira Martins**

**julho 2015**



# Dedicatória

Dedico este trabalho à Heidy e à Sofia.

# Agradecimentos

A realização deste projeto de mestrado contou com importantes apoios e incentivos, sem os quais a sua realização não teria sido possível.

Agradeço ao Coordenador do Mestrado em Computação Móvel e orientador do projeto, Professor Doutor Carlos Carreto, pelos saberes transmitidos, pelo apoio, disponibilidade e por toda a colaboração prestada durante o desenvolvimento deste projeto.

Gostaria também de agradecer à Professora Doutora Clara Silveira, Diretora da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda, pela total colaboração e ajuda prestada.

Agradeço também à minha família pelo incentivo e apoio incondicional demonstrados ao longo do desenvolvimento deste projeto, aos quais dedico este trabalho.

# Resumo

Em atividades ao ar livre, como a realização de percursos em veículos todo-terreno, BTT, caminhada ou roteiros turísticos, é comum recorrer-se à navegação utilizando um guia que contém a descrição do trajeto a ser percorrido. Este guia, conhecido como *roadbook*, é elaborado através de notas tiradas, com alguma antecedência, aquando do reconhecimento e planificação do percurso.

Tipicamente, um *roadbook* é impresso em papel, mas com os atuais avanços da Computação Móvel e dos dispositivos móveis, torna-se muito interessante e útil ter um *roadbook* digital para *smartphones e tablets*, com o objetivo de facilitar a elaboração das notas, aquando do reconhecimento do percurso, assim como assistir o utilizador no uso dessas mesmas notas aquando da navegação.

Este relatório descreve o projeto aplicado, desenvolvido pelo autor, que consiste numa aplicação *roadbook* digital para dispositivos móveis *Android*, denominada de *Naturepathfinder*. Esta aplicação é composta por duas partes principais: Criar Navegação e Navegar.

A primeira parte - Criar Navegação – tem como objetivo permitir a criação dos *roadbooks*, recolhendo os dados necessários diretamente no terreno, a partir do *GPS* do dispositivo móvel e permitindo também ao utilizador elaborar os desenhos das notas através de um editor gráfico.

A segunda parte – Navegar – tem como objetivo ajudar o utilizador na navegação de um *roadbook* já existente numa base de dados, apresentando as notas de forma automática sempre que detetar a proximidade de um cruzamento. Ainda durante a navegação de um *roadbook*, a aplicação *Naturepathfinder* avisa o utilizador caso considere que este está a cometer um erro de navegação e faz automaticamente a correção da rota, guiando o utilizador de regresso ao caminho correto, utilizando mapas *online* ou uma seta indicadora da direção a tomar.

Em qualquer uma das partes descritas, a aplicação *Naturepathfinder* utiliza uma filosofia de deteção de contextos, pois a todo momento a aplicação conhece as variáveis do ambiente onde se localiza e adapta as suas próprias ações e as ações que o utilizador pretende desencadear de acordo com as mudanças no meio envolvente. O exemplo principal da deteção de contextos prende-se com a parte da navegação, pois a aplicação sabe exatamente em que cruzamento se encontra e mostra as indicações respetivas, assim como quando existe um erro de navegação a aplicação será capaz de saber onde se encontra e guiar o utilizador para o caminho correto, retomando de seguida o processo de navegação normal.

Foram também utilizados *Web Services*, para que toda a informação recolhida para a criação dos *roadbooks* seja armazenada numa base de dados *SQLite* e possa ser enviada para um servidor *web* onde ficará armazenada numa base de dados *MySQL*. O objetivo deste servidor é permitir a partilha dos *roadbook* criados por todos os utilizadores da aplicação *Naturepathfinder*.

Após efetuados os testes à aplicação com o público-alvo concluiu-se que obteve resultado positivo no que respeita à usabilidade e comportamento aquando da criação e navegação de um *roadbook*, pois a maior parte das pessoas que utilizaram a aplicação durante a fase de testes conseguiu criar um *roadbook* e navegar um *roadbook* criado por outra pessoa. Todas as pessoas admitiram que a aplicação foi ao encontro das suas expectativas e que este método de criação e navegação de *roadbooks* traz grandes vantagens relativamente à utilização dos tradicionais *roadbooks* em papel.

**Palavras-chave:** *Roadbook digital, Android, Mapas Online, Detecção de Contextos, Web Services*

# Abstract

In outdoor activities, such as managing paths in all-terrain vehicles, mountain biking, hiking or tourist itineraries, it is common to resort to the navigation by using a guide containing the description of the path that should be followed. This guide is known as a roadbook and it is prepared by taking notes in advance at the time of the recognition and planning of the route.

A roadbook is usually printed on paper, but with current advances in mobile computing and mobile devices, it has become very interesting and useful to have a digital roadbook for smartphones and tablets, in order to facilitate the preparation of notes upon recognition of the route, as well as to assist the user when they apply these same notes when navigating.

This report describes the applied project developed by the author, which consists of a digital roadbook application for Android mobile devices, called *Naturepathfinder*. This application consists of two main parts: Create Navigation and Navigate.

The first part - Create Navigation - aims to enable the creation of roadbooks by collecting the necessary data directly from the terrain on the mobile GPS and also by allowing the user to draw the note designs through a graphical editor.

The second part - Navigate- aims to assist the user in the navigation of an existing roadbook in a database, by presenting notes automatically when they detect the proximity of an intersection. While navigating a roadbook, the *Naturepathfinder* application warns the user if it considers that it is making a navigational error and automatically corrects the route, guides the user to the right path by using online maps or taking a direction indicated by an arrow.

In both parts described the *Naturepathfinder* application uses a philosophy of detection of contexts, because the application knows the environment variables where it is located the whole time and it adapts its own actions and the actions that the user wants to trigger according to the variations of environment/scenery. The prime example of contexts detection relates to the part of the navigation, because the application knows exactly where it is crossing and it shows the relevant instruction. Another example is when there is a navigation error, it will be able to identify the place and guide you to the correct path, and then return to the normal navigational process.

Web Services were also used so that all the information collected to create the roadbooks can be stored in a SQLite database and it can be sent to a web server where it will be stored in a MySQL database. The purpose of this server is to allow the sharing of the roadbook created by the users of *Naturepathfinder* application.

After analyzing the tests carried out with the target audience, it is concluded that the application achieved positive results with regard to usability and behavior in establishing and navigating a roadbook, because the vast majority of people who used the application during the testing phase managed to create a roadbook and navigate a road book created by someone else. Everyone enquired admitted that the application met their expectations and that this method of creating and navigating roadbooks has brought advantages in comparison to the use of traditional roadbooks on paper.

**Keywords:** Digital roadbook, Android, Online Maps, detection contexts, Web Services.



# Índice

<b>Dedicatória.....</b>	<b>i</b>
<b>Agradecimentos.....</b>	<b>ii</b>
<b>Resumo.....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>iv</b>
<b>Índice.....</b>	<b>v</b>
<b>Índice de figuras.....</b>	<b>viii</b>
<b>Índice de Tabelas.....</b>	<b>xi</b>
<b>Lista de siglas e acrónimos.....</b>	<b>xii</b>
<b>1. Introdução.....</b>	<b>1</b>
1.1 Enquadramento e Motivação.....	1
1.2 Descrição do projeto.....	2
1.2.1 Bases de dados.....	4
1.2.2 Editor gráfico para desenhar esquemas das notas.....	4
1.2.3 Módulo de gravação das notas de um roadbook.....	4
1.2.4 Módulo de navegação de um roadbook.....	5
1.2.5 Processo de correção de rota.....	6
1.2.6 Servidor web para partilha de roadbooks.....	7
1.2.7 Interface da aplicação.....	7
1.2.8 Detecção de contextos.....	7
1.2.9 Aplicação Naturepathfinder.....	8
1.3 Objetivos do projeto.....	9
1.4 Estrutura do Relatório.....	10
<b>2. Tecnologias de roadbooks.....</b>	<b>12</b>
2.1 Roadbook tradicional em papel.....	13
2.2 Dispositivos enroladores de roadbook.....	18
2.3 Resumo dos principais problemas da utilização dos roadbooks tradicionais em papel.....	21
2.4 Roadbooks digitais.....	22
2.5 Aplicações de roadbook para smartphones e tablets.....	25
<b>3. Análise de requisitos.....</b>	<b>32</b>
3.1 Permitir a criação de roadbooks.....	32
3.1.1 Marcação das coordenadas dos pontos online na própria aplicação.....	33

3.1.2	<i>Marcação das coordenadas dos pontos recorrendo ao sensor GPS do dispositivo móvel</i>	35
3.2	Necessidade de equipamento extra para marcação das coordenadas dos pontos.....	36
3.3	Esquemas das notas predefinidos .....	37
3.4	Desenhar esquemas das notas.....	37
3.5	Permitir Download de percursos da <i>web</i> .....	38
3.6	Permitir partilhar os <i>roadbooks</i> criados na <i>web</i> num serviço central .....	39
3.7	Permitir navegação manual.....	39
3.8	Permitir navegação automática.....	40
3.8.1	<i>Navegação através de distâncias percorridas</i> .....	40
3.8.2	<i>Navegação através das coordenadas dos pontos</i> .....	42
3.9	Necessidade de equipamento extra para a navegação.....	44
3.10	Detecção de erros de navegação e correção da rota.....	44
3.11	Utilização em veículos motorizados .....	45
3.12	Utilização em passeios de bicicleta e percursos pedestres.....	45
3.13	Resumo da análise .....	46
3.14	Estudo da importância que potenciais utilizadores atribuem às características analisadas.....	48
3.15	Estudo de opinião sobre a importância que potenciais utilizadores dão à existência de uma aplicação para dispositivos móveis com as características enunciadas .....	53
<b>4.</b>	<b>Implementação .....</b>	<b>55</b>
4.1	Arquitetura do Sistema .....	55
4.2	Bases de dados.....	57
4.2.1	<i>Base de dados SQLite – da aplicação</i> .....	58
4.2.2	<i>Base de dados MySQL – servidor web</i> .....	62
4.3	Editor gráfico para desenhar esquemas das notas.....	66
4.4	Módulo de gravação das notas de um <i>roadbook</i> .....	68
4.4.1	<i>Tecnologia utilizada para obtenção da localização</i> .....	69
4.4.2	<i>Registo de notas para criação de roadbooks</i> .....	72
4.4.3	<i>Gravar notas tendo em conta a velocidade e a exatidão</i> .....	79
4.4.4	<i>Inserir restantes dados da nota</i> .....	81
4.4.5	<i>Gravar ponto de controlo</i> .....	83
4.5	Módulo de navegação de um <i>roadbook</i> .....	85
4.5.1	<i>Configuração da frequência de leituras do GPS para poupança de energia</i> .....	87
4.5.2	<i>Rotação da Seta indicadora da direção</i> .....	88
4.5.3	<i>Definir o tipo de navegação (automática ou manual)</i> .....	90
4.5.4	<i>Calcular a distância à qual são detetados os cruzamentos</i> .....	92
4.5.5	<i>Passagem à nota seguinte</i> .....	96
4.5.6	<i>Verificar se o utilizador seguiu pelo caminho correto</i> .....	97
4.6	Processo de correção de rota.....	99

4.6.1	<i>Processo de correção de rota utilizando a seta</i> .....	101
4.6.2	<i>Processo de correção de rota utilizando mapa</i> .....	101
4.7	Detecção de Contextos .....	108
4.7.1	<i>Detecção de Contextos na aplicação desenvolvida</i> .....	109
4.8	Servidor <i>web</i> para partilha de <i>roadbooks</i> .....	110
4.8.1	<i>Envio de dados do dispositivo móvel para a web</i> .....	111
4.8.2	<i>Download de dados da web para o dispositivo móvel</i> .....	116
4.9	Interface da aplicação .....	119
4.9.1	<i>Densidade de pixéis</i> .....	119
4.9.2	<i>Diferentes tamanhos ecrãs</i> .....	122
4.9.1	<i>Títulos, textos e botões</i> .....	125
4.9.2	<i>Diferentes idiomas</i> .....	125
<b>5.</b>	<b>Testes e resultados</b> .....	<b>126</b>
5.1	Testes de desempenho .....	126
5.2	Teste do grau de satisfação da aplicação .....	130
5.3	Testes de usabilidade.....	135
5.4	Aspetos da implementação referentes a cada Heurística de Nielsen .....	137
<b>6.</b>	<b>Conclusões e perspetivas de desenvolvimento</b> .....	<b>141</b>
	<b>Referências bibliográficas</b> .....	<b>145</b>
	<b>Anexo A</b> .....	<b>149</b>
	<b>Anexo B</b> .....	<b>153</b>
	<b>Anexo C</b> .....	<b>155</b>
	<b>Anexo D</b> .....	<b>157</b>

# Índice de figuras

FIGURA 1 - EXEMPLO DE UMA PÁGINA DE UM ROADBOOK TRADICIONAL.....	12
FIGURA 2 - ESQUEMA DE UMA NOTA DE ROADBOOK .....	14
FIGURA 3 - UTILIZAÇÃO DE ROADBOOK DE FORMA TRADICIONAL, EM PAPEL .....	14
FIGURA 4 - DISPOSITIVO TERRATRIP PARA CONTROLO DE DISTÂNCIAS .....	15
FIGURA 5 - EXEMPLO DE UM ROADBOOK DESENHADO À MÃO .....	16
FIGURA 6 - UTILIZAÇÃO DE FOLHA DE CÁLCULO PARA CRIAR ROADBOOK.....	16
FIGURA 7 - INTERFACE DA APLICAÇÃO ROADBOOK ASSISTANT.....	17
FIGURA 8 - DISPOSITIVO ELETRÓNICO PARA ENROLAR UM ROADBOOK EM PAPEL, UTILIZADO EM MOTAS.....	18
FIGURA 9 - BOTÕES DE CONTROLO DAS NOTAS .....	18
FIGURA 10 - OPÇÃO PARA ENROLAR OU DESENROLAR O ROADBOOK MANUALMENTE .....	19
FIGURA 11 - DISPOSITIVO ARTESANAL ENROLADOR DE ROADBOOK .....	19
FIGURA 12 - LEIROR DE ROADBOOK ARTESANAL, MAS ELÉTRICO.....	20
FIGURA 13 - DISPOSITIVO ENROLADOR DE ROADBOOK COM GPS ACOPLADO .....	21
FIGURA 14 - ROADBOOK DIGITA PARA MOTOS DA EMPRESA "TRIPY".....	23
FIGURA 15 - CRIAÇÃO AUTOMÁTICA DA NOTA DE ROADBOOK ATRAVES DA DEFINIÇÃO DE PONTOS DO TRAJECTO NO SOFTWARE ROADTRACER PRO.....	23
FIGURA 16 - MENU DO TRIPY GPS EM QUE SE PODE VER O CAMINHO A PERCORRER.....	24
FIGURA 17 - APLICAÇÃO MOBILEROADBOOK .....	25
FIGURA 18 - NOTA DO ROADBOOK.....	25
FIGURA 19 - INTERFACE DA APLICAÇÃO GPS ROAD BOOK.....	26
FIGURA 20 - INTERFACE DA APLICAÇÃO ROAD BOOK DIGITAL TOTEM.....	27
FIGURA 21 - DISPOSITIVO COLOSSO EVO.....	27
FIGURA 22 - INTERFACE DO SOFTWARE TOTEM PLANILHA .....	28
FIGURA 23 - INTERFACE DA APLICAÇÃO ROADBOOK JSC v4 .....	29
FIGURA 24 - APLICAÇÃO ROADBOOK EM MODO DE NAVEGAÇÃO POR ESQUEMAS.....	30
FIGURA 25 - APLICAÇÃO ROADBOOK EM MODO DE NAVEGAÇÃO POR MAPA .....	30
FIGURA 26 - INTERFACE DA APLICAÇÃO TULIP - ROADBOOK EDITOR FOR IPAD .....	31
FIGURA 27 - IMAGEM GOOGLE MAPS - VISTA MAPA.....	33
FIGURA 28 - IMAGEM GOOGLE MAPS - VISTA SATÉLITE .....	33
FIGURA 29 - IMAGEM GOOGLE MAPS EM MODO "SATÉLITE" COM POUCO DETALHE DEVIDO ÀS NUVENS .....	34
FIGURA 30 - GPS GARMIN ETREX 10 .....	36
FIGURA 31 - EXEMPLOS DE ESQUEMAS INDICADORES DA DIREÇÃO A TOMAR.....	37
FIGURA 32 - DESENHAR UM ESQUEMA NO SOFTWARE TOTEM PLANILHA .....	38
FIGURA 33 - MARCAÇÃO DE PONTOS ATRAVÉS DE DISTÂNCIAS –CIRCULAR PELO LADO EXTERIOR DA CURVA.....	41

FIGURA 34 - NAVEGAÇÃO UTILIZANDO DISTÂNCIAS – CIRCULAR PELO LADO INTERIOR DA CURVA .....	41
FIGURA 35 - NAVEGAÇÃO USANDO AS COORDENADAS COMO REFERÊNCIA - CIRCULAR NORMALMENTE PELO CAMINHO.....	42
FIGURA 36 - NAVEGAÇÃO USANDO AS COORDENADAS COMO REFERÊNCIA – CIRCULAR DE FORMA IRREGULAR.....	43
FIGURA 37 - ESTUDO DA IMPORTÂNCIA QUE POTENCIAIS UTILIZADORES ATRIBUEM ÀS CARACTERÍSTICAS ANALISADAS .....	48
FIGURA 38 - ESTUDO EFETUADO NO QUE RESPEITA À OPINIÃO SOBRE A IMPORTÂNCIA DA EXISTÊNCIA DE UMA APLICAÇÃO PARA DISPOSITIVOS MÓVEIS COM AS CARACTERÍSTICAS ENUNCIADAS .....	53
FIGURA 39 - ARQUITETURA GERAL DO SISTEMA.....	56
FIGURA 40 - TABELAS DA BASE DE DADOS SQLITE NO DISPOSITIVO MÓVEL.....	58
FIGURA 41 - TABELAS DA BASE DE DADOS MYSQL NA WEB .....	62
FIGURA 42 - PAINEL DE CONTROLO DO XAMPP .....	63
FIGURA 43 - CRIAR TABELA “ROADBOOK” NO MENU DE ADMINISTRAÇÃO MYSQL DO XAMPP .....	64
FIGURA 44 - MENU DO ALOJAMENTO WEB REGISTRADO .....	64
FIGURA 45 - CRIAR TABELA “NOTA” NO MENU DE ADMINISTRAÇÃO MYSQL NA WEB .....	65
FIGURA 46 - LAYOUT DO EDITOR GRÁFICO.....	66
FIGURA 47 - TRIANGULAÇÃO DE SINAL GPS.....	69
FIGURA 48 - FLUXOGRAMA DOS PROCESSOS DE OBTENÇÃO DAS COORDENADAS, EXATIDÃO E VELOCIDADE DO DISPOSITIVO.....	73
FIGURA 49 - MENU DE GRAVAÇÃO DAS NOTAS .....	74
FIGURA 50 - DIFERENÇA ENTRE PRECISÃO E EXATIDÃO .....	77
FIGURA 51 - EXEMPLO DE UMA LEITURA EFETUADA PELO MÉTODO GETACCURACY .....	78
FIGURA 52 - FLUXOGRAMA DA VERIFICAÇÃO DA EXATIDÃO E VELOCIDADE PARA INÍCIO DO PROCESSO DE GUARDAR AS COORDENADAS .....	80
FIGURA 53 - INSERÇÃO DA INFORMAÇÃO E ESQUEMA DA NOTA GUARDADA .....	81
FIGURA 54 - MENU DE ESCOLHA DO ESQUEMA DA NOTA ATUAL .....	82
FIGURA 55 - GRAVAÇÃO DO PONTO DE CONTROLO.....	83
FIGURA 56 - FLUXOGRAMA REFERENTE À GRAVAÇÃO DO PONTO DE CONTROLO .....	84
FIGURA 57 - FLUXOGRAMA DOS PROCESSOS DE OBTENÇÃO DAS COORDENADAS E RESTANTE INFORMAÇÃO NECESSÁRIA PARA A NAVEGAÇÃO.....	86
FIGURA 58 - IMAGEM INDICADORA DA DIREÇÃO A TOMAR.....	88
FIGURA 59 - REPRESENTAÇÃO GRÁFICA DOS ÂNGULOS EM RELAÇÃO AO RUMO.....	89
FIGURA 60 - FLUXOGRAMA PARA CONTROLO DO MODO DE NAVEGAÇÃO .....	91
FIGURA 61 - ESQUEMA DE DETEÇÃO DE UM CRUZAMENTO.....	92
FIGURA 62 - REPRESENTAÇÃO DA EXATIDÃO SEM DETETAR CRUZAMENTO .....	93
FIGURA 63 - REPRESENTAÇÃO DA EXATIDÃO A DETETAR CRUZAMENTO .....	93
FIGURA 64 - ESQUEMA DA DETEÇÃO DO CRUZAMENTO.....	94
FIGURA 65 - ESQUEMA DE DETEÇÃO COM DOIS CRUZAMENTOS.....	94
FIGURA 66 - DIMINUIÇÃO DA DISTÂNCIA À QUAL SE DEVE DETETAR O CRUZAMENTO .....	95
FIGURA 67 - FLUXOGRAMA DO ALGORITMO PARA A PASSAGEM À NOTA SEGUINTE .....	97
FIGURA 68 - ERRO DE NAVEGAÇÃO .....	98

FIGURA 69 - ATIVAÇÃO DO PROCESSO DE CORREÇÃO DA ROTA.....	98
FIGURA 70 - CORREÇÃO DE ROTA USANDO O MAPA.....	99
FIGURA 71 - CORREÇÃO DE ROTA USANDO A SETA .....	99
FIGURA 72 - FLUXOGRAMA DO PROCESSO DE CORREÇÃO DA ROTA .....	100
FIGURA 73 - EXEMPLO DE UMA TILE.....	102
FIGURA 74 - COMPARAÇÃO ENTRE MAPNIK E MAPQUEST .....	103
FIGURA 75 - TILES MAPQUEST SATELLITE.....	103
FIGURA 76 - MAP SOURCE NO MOBILE ATLAS CREATOR .....	105
FIGURA 77 - ZOOM LEVELS NO MOBILE ATLAS CREATOR .....	105
FIGURA 78 - UTILIZAÇÃO DO MAPVIEW .....	106
FIGURA 79 - MARCADOR DO PONTO DE CONTROLO NO MAPA.....	107
FIGURA 80 - MARCADOR DO UTILIZADOR NO MAPA.....	107
FIGURA 81 - INTERFACE PARA PARTILHA DE ROADBOOKS .....	111
FIGURA 82 - VERIFICAR SE EXISTEM DADOS NO DISPOSITIVO QUE PODEM SER ENVIADOS PARA A WEB .....	112
FIGURA 83 - PASSAGEM DE DADOS ENTRE A BASE DE DADOS SQLITE E A BASE DE DADOS MYSQL.....	115
FIGURA 84 - DOWNLOAD DE ROADBOOKS A PARTIR DA WEB.....	116
FIGURA 85 - PASSAGEM DE DADOS ENTRE A BASE DE DADOS MYSQL E A BASE DE DADOS SQLITE.....	118
FIGURA 86 - DENSIDADE DE PIXELS .....	119
FIGURA 87 - EXEMPLO DE VISUALIZAÇÃO DA MESMA IMAGEM EM ECRÃS COM DIFERENTES DENSIDADES DE PIXELS .....	120
FIGURA 88 - DEFINIÇÃO DO TAMANHO DOS COMPONENTES EM DP .....	121
FIGURA 89 - ÍCONE CRIADOS PARA A APLICAÇÃO .....	122
FIGURA 90 - O MESMO LAYOUT EM DISPOSITIVOS COM ECRÃS DIFERENTES.....	123
FIGURA 91 - O MESMO LAYOUT EM DISPOSITIVOS COM ECRÃS DIFERENTES MAS COM ADAPTAÇÃO PARA ECRÃS GRANDES ..	124
FIGURA 92 - INTERFACE AQUANDO DOS TESTES DE COMPORTAMENTO.....	128
FIGURA 93 - TESTES DE DESEMPENHO DA APLICAÇÃO DIRETAMENTE NO TERRENO .....	129
FIGURA 94 - CLASSIFICAÇÃO DA IMPLEMENTAÇÃO DAS HEUÍSTICAS DE NIELSEN NA APLICAÇÃO NATUREPATHFINDER .....	135

# Índice de Tabelas

TABELA 1 - COMPARAÇÃO ENTRE CRIADORES/LEITORES DE ROADBOOK PARA DISPOSITIVOS MÓ-VEIS NO QUE RESPEITA ÀS SUAS CARACTERÍSTICAS.....	47
TABELA 2 - TIPOS DE DADOS DOS CAMPOS DA TABELA “ORGANIZACAO”.....	60
TABELA 3 - TIPOS DE DADOS DOS CAMPOS DA TABELA “ROADBOOK”.....	60
TABELA 4 - TIPOS DE DADOS DOS CAMPOS DA TABELA “NOTA”.....	61
TABELA 5 - TIPOS DE DADOS DOS CAMPOS DA TABELA “ESQUEMA”.....	61
TABELA 6 - TIPOS DE DADOS DOS CAMPOS DA TABELA “NAVEGACAO”.....	61
TABELA 7 - ANÁLISE DAS TECNOLOGIAS PARA OBTENÇÃO DA LOCALIZAÇÃO.....	71
TABELA 8 - VELOCIDADE PARA DETERMINAR O NÚMERO DE METROS E SEGUNDOS ENTRE CADA LEITURA DO GPS.....	75
TABELA 9 - DISTÂNCIA PARA DETERMINAR O NÚMERO DE METROS E SEGUNDOS ENTRE CADA LEITURA DO GPS.....	87
TABELA 10 - EXEMPLO DO CÁLCULO DA DISTÂNCIA À QUAL É DETETADO O CRUZAMENTO.....	96
TABELA 11 - CLASSES PHP PARA LIGAÇÃO ENTRE A APLICAÇÃO E A BASE DE DADOS MYSQL.....	113
TABELA 12 - CLASSES PHP PARA LIGAÇÃO ENTRE A BASE DE DADOS MYSQL E A APLICAÇÃO.....	117
TABELA 13 - LOGICAL PIXEL DENSITIES.....	120
TABELA 14 - LISTA DOS DISPOSITIVOS MÓVEIS USADOS DURANTE OS TESTES DE DESEMPENHO.....	127

# Lista de siglas e acrónimos

- APK - Android package
- BLOB - Binary large object
- BTT - Bicicleta todo-o-terreno
- DPI - Dots-per-inch
- FTP - File transfer protocol
- GPS - Global positioning system
- iOS - iPhone operating system
- JSON - JavaScript Object Notation
- PDF - Portable Document Format
- PHP - Hypertext Preprocessor
- PNG - Portable Network Graphics
- SDK - Software Development Kit
- SQL - Structured Query Language
- UHF - Ultra High Frequency
- Wi-Fi - Wireless Fidelity



## Capítulo 1

# 1. Introdução

Neste primeiro capítulo é definido o enquadramento e a motivação para o desenvolvimento deste projeto, é descrito o projeto no que respeita aos módulos que o compõem. São também definidos os objetivos gerais e específicos deste trabalho.

## 1.1 Enquadramento e Motivação

Este relatório descreve o projeto aplicado desenvolvido no âmbito do Curso de Mestrado em Computação Móvel do Instituto Politécnico da Guarda, que visa a criação de um *roadbook* digital para dispositivos móveis *Android*. Um *roadbook* é um guia usualmente utilizado em percursos em atividades ao ar livre, como Todo-Terreno, BTT ou Caminhada. Normalmente este guia é impresso em papel e contém a descrição das instruções do trajeto a ser percorrido. É elaborado através de notas registadas com alguma antecedência, aquando do reconhecimento e planificação do percurso.

Com os atuais avanços da Computação Móvel e dos dispositivos móveis, torna-se muito interessante e útil ter um *roadbook* digital para *smartphones e tablets*, com o objetivo de facilitar a elaboração das notas, aquando do reconhecimento do percurso, assim como assistir o utilizador no uso dessas mesmas notas aquando da navegação. Assim sendo, pretende-se que a aplicação permita ao utilizador elaborar as notas de um *roadbook* diretamente no terreno, recolhendo dados como coordenadas geográficas, diretamente a partir do *GPS* do dispositivo móvel e ao mesmo tempo assista o utilizador na navegação de *roadbooks* já existente numa base de dados do dispositivo móvel. Na fase da navegação tem-se como objetivo que a aplicação apresente as notas de forma automática, detetando a proximidade dos cruzamentos e avisando o utilizador caso considere que este está a cometer um erro de navegação e ajudando-o no regresso ao percurso correto. Trata-se portanto de uma aplicação que assenta numa metodologia de deteção de contextos, em que a mesma tem conhecimento do meio que a rodeia, tendo atenção às suas alte-

rações, nomeadamente exatidão do sinal *GPS*, velocidade, distâncias a determinados pontos, localização geográfica do próprio dispositivo e de pontos específicos ao longo do percurso. Tendo em conta as mudanças nestas variáveis do ambiente em que o dispositivo móvel se encontra, a aplicação vai adaptar o seu comportamento e as suas ações de acordo com o contexto e ao mesmo tempo interagir com o utilizador de maneira diferente de acordo com a situação atual.

A motivação principal para a realização deste trabalho resulta da tendência crescente de facilitar a criação de roadbooks assim como depois a sua navegação. Isto porque um roadbook tradicional em papel (Anexo A) requer muito cuidado e atenção no momento da sua criação assim como na respetiva navegação, especialmente no que se refere à interpretação das medições das distâncias entre cruzamentos. Assim sendo, a existência de uma aplicação para dispositivos móveis que possa simplificar o processo de criação das notas de um *roadbook* e posterior navegação apresenta grandes vantagens relativamente à criação dos *roadbooks* tradicionais em papel.

No que respeita a plataformas para dispositivos móveis, segundo estudos efetuados pela *International Data Corporation*, as plataformas *Android* e *iOS* são as mais difundidas por todo o mundo<sup>1</sup>, a partir do ano de 2011. No entanto, o número de vendas de dispositivos *Android* foi sempre maior do que o número de vendas de dispositivos com o sistema *iOS*. Após a análise deste estudo, optou-se por desenvolver a aplicação para a plataforma *Android*, pois esta plataforma revela ser a que mais utilizadores tem a nível mundial.

Foram analisadas algumas aplicações existentes para dispositivos móveis que tinham como função a criação/leitura de *roadbooks* digitais e a comparação entre as suas características é feita na secção 2.5. A inexistência de projetos semelhantes para dispositivos móveis *Android*, que permitam a navegação automática de um *roadbook*, conjugada com a implementação de deteção e contextos que permitisse a deteção de erros de navegação e respetiva correção da rota foi outra motivação para o desenvolvimento deste projeto.

Tratando-se de uma aplicação cujo objetivo principal é gerir percursos em atividades ao ar livre optou-se por lhe atribuir a designação de *Naturepathfinder*.

## 1.2 Descrição do projeto

Um *roadbook* tradicional, em papel, é constituído por um conjunto de esquemas que representam a direção que se deve tomar em cada cruzamento. Quando se cria um *roadbook* existe

---

<sup>1</sup> <http://www.idc.com/getdoc.jsp?containerId=prUS25450615>

a necessidade de registar manualmente todas as distâncias entre cada cruzamento, para que o utilizador ao navegar consiga saber a que distância pode esperar que se encontra o próximo cruzamento onde irá ter de escolher o caminho correto representado pelo esquema. Como foi explicado na secção 2.1, basta uma distância estar errada para que se comprometa toda a navegação, ou então basta o utilizador realizar uma leitura errada para por em causa a navegação. Pode acrescentar-se a isto o facto de o utilizador ter de estar muito atento às distâncias percorridas o que pode contribuir para que esteja menos atento ao esquema que representa a direção a tomar num determinado cruzamento, resultando disto um erro de interpretação e respetivo erro de navegação.

A aplicação *Naturepathfinder* vem resolver estes problemas, na medida que tanto no processo de criação de *roadbooks* como na posterior navegação a aplicação utiliza uma metodologia de deteção de contextos, sabendo a todo momento qual a sua localização atual. Assim sendo, na altura da criação de *roadbooks*, o utilizador não terá de registar manualmente nenhuma distância, mas concentrar-se apenas na introdução da informação e esquema referente a cada cruzamento ou nota que pretender registar. Na fase de navegação dos *roadbooks* a aplicação também resolve os problemas existentes na utilização dos *roadbooks* em papel, na medida que são apresentadas automaticamente todas as informações referentes a cada nota, assim que for detetado pela aplicação que o utilizador se está a aproximar de um cruzamento. Desta forma o utilizador apenas necessita de se concentrar na análise do esquema apresentado e fazer a escolha correta no que respeita ao caminho a seguir.

Como foi referido anteriormente, na navegação com um *roadbook* em papel existe a possibilidade de existência de erros de navegação, que são difíceis de detetar com a utilização deste tipo de *roadbooks*. Usualmente, depois de existir um erro de navegação usando esta técnica, o utilizador apenas deteta este erro depois de percorrida alguma distância e possivelmente depois de ter passado alguns cruzamentos. Nesta situação é muito complicado voltar atrás e corrigir a navegação, especialmente se o percurso for numa zona desconhecida, pelo que a navegação fica comprometida.

A aplicação *Naturepathfinder* pode evitar estes problemas, pois tem implementado um método que permite detetar se o utilizador seguiu pelo caminho correto ou efetuou um erro de navegação. Ao ser detetado um erro de navegação a aplicação procede à correção da rota, guiando o utilizador de volta ao caminho correto. Considera-se esta é uma grande evolução relativamente ao sistema de navegação de *roadbooks* tradicionais em papel.

Este projeto encontra-se dividido em vários componentes, que fazem parte integrante da aplicação, nomeadamente duas bases de dados para armazenamento de toda a informação, um editor gráfico que permite a criação dos esquemas dos cruzamentos, o módulo que corresponde

ao processo de criação de *roadbooks*, o módulo que permite o controlo da navegação de *roadbooks* criados anteriormente, o processo utilizado para deteção de erros de navegação e correção da rota, o servidor *web* utilizado para partilha dos *roadbooks* criados e finalmente a interface da aplicação.

### 1.2.1 Bases de dados

Foi criada uma base de dados relacional *SQLite* para armazenar dados no dispositivo, nomeadamente informação sobre os *roadbooks*, dados das notas criadas para cada *roadbook*, desenhos dos esquemas das notas, identificação e dados dos criadores dos *roadbooks* e dados de uma navegação específica.

Foi também criada uma base de dados *MySQL* num domínio *Web*, com o objetivo da partilha dos *roadbooks* criados com todos os utilizadores da aplicação.

Este tópico é abordado com mais detalhe na secção 4.1.

### 1.2.2 Editor gráfico para desenhar esquemas das notas

Cada nota gravada deverá ter uma imagem representativa do cruzamento em questão, assim como a indicação da direção a tomar. Foi implementada na aplicação *Naturepathfinder* uma ferramenta que permite ao utilizador desenhar o esquema pretendido ou utilizar um esquema que já exista previamente.

Para se implementar esta funcionalidade foi definido uma componente tipo “View” personalizada, como sendo a área onde se pode efetuar o desenho. Após definida a área de desenho foi implementada a interatividade que permite à aplicação detetar o toque no ecrã para a criação das linhas pretendidas seguindo no ecrã o toque do utilizador. Depois de estar desenhado, o esquema pode ser guardado na base de dados *SQLite* para que depois possa ser associado a uma nota.

Este tópico é abordado com mais detalhe na secção 4.3.

### 1.2.3 Módulo de gravação das notas de um *roadbook*

No processo de gravação das notas de um *roadbook* o utilizador deverá começar por criar o próprio *roadbook*, adicionar todas as informações necessárias referentes à identificação

do mesmo. Depois de concluir este processo passa-se à fase de adicionar as notas do *roadbook*, que irão identificar cada um dos cruzamentos que fazem parte do percurso, registando a localização geográfica de cada ponto.

Após análise efetuada e descrita na secção 4.4.1, a tecnologia selecionada para obtenção da localização foi o *GPS*. As coordenadas geográficas provenientes do sensor de *GPS* do dispositivo móvel serão guardadas em cada cruzamento que o utilizador pretenda registar no *roadbook*. Tratando-se de uma aplicação com deteção de contextos, a frequência das leituras do *GPS* será adaptada à velocidade atual, sendo que a velocidades reduzidas irão ser efetuadas mais leituras e a velocidades maiores irão ser efetuadas menos leituras, com o objetivo de poupança de energia.

Outro fator que irá ser monitorizado, tendo em conta a deteção de contextos, é a exatidão do sinal *GPS*, para que seja possível analisar se é viável a captura das coordenadas. Se o *GPS* apresentar uma exatidão de sinal reduzida poderá não ser permitido pela aplicação a gravação das coordenadas atuais, até que o sinal seja aceitável.

Também o controlo da velocidade é importante no que respeita à gravação das coordenadas de um ponto. Se o utilizador se deslocar acima da velocidade estabelecida pela aplicação não será permitida a captura das coordenadas, evitando-se assim possíveis erros no registo das coordenadas corretas para um cruzamento específico.

Após se ter guardado a latitude e longitude da nota, passa-se a introduzir a informação acerca da mesma e é também escolhido, ou desenhado, o esquema representativo do cruzamento.

Após o registo de todos os dados necessários o utilizador continua o percurso, sendo que passados alguns metros serão guardadas, de forma automática as coordenadas de um ponto de controlo. Este ponto vai ser utilizado no momento da navegação, para se verificar se depois de passar um cruzamento o utilizador segue pelo caminho correto. Todos os dados são guardados na base de dados *SQLite*.

Este tópico é abordado com mais detalhe na secção 4.4.

## **1.2.4 Módulo de navegação de um *roadbook***

Durante a navegação o utilizador irá ter acesso a informação referente à nota do cruzamento seguinte, especialmente ao desenho do esquema que lhe permite identificar por que caminho seguir quando encontrar o cruzamento. Nessa fase a aplicação também tem um modo de funcionamento baseado na deteção de contextos, na medida que vai analisar as variáveis do ambiente e realizar uma navegação automática. À medida que o utilizador se desloca é mostrada

automaticamente a nota do cruzamento seguinte assim como a informação respetiva. Se for cometido um erro de navegação a aplicação irá avisar o utilizador e iniciar o processo de correção da rota.

Também nesta etapa se recorreu ao sensor de *GPS* do dispositivo, de forma a se lerem as coordenadas dos pontos que definem o trajeto que se está a seguir. Para consultar e mostrar a informação de cada nota, a aplicação acede à base de dados *SQLite*, onde esta informação tinha sido armazenada durante a criação do *roadbook*.

Também nesta etapa irá ser feita a adaptação do número de leituras efetuadas pelo sensor de *GPS*, para que se possa poupar a bateria do dispositivo móvel, neste caso tendo em conta a distância em linha reta a que se encontra o utilizador do próximo cruzamento, sendo que quanto mais afastado estiver o utilizador do próximo cruzamento menor será o número de leituras efetuadas.

Para além do desenho do esquema representativo do cruzamento será também apresentada uma seta que indica a direção em que fica o próximo cruzamento e funcionará como auxiliar de navegação, cujo cálculo do ângulo de rotação é feito com base na direção que o utilizador está a tomar (rumo) em conjunto com a direção em que se encontra o próximo ponto registado.

Caso a aplicação detete que a exatidão do sinal *GPS* é baixa irá passar-se automaticamente para uma navegação manual, em que o utilizador terá de passar para a nota seguinte automaticamente. Este processo foi implementado para evitar que ocorressem erros de navegação quando o sinal *GPS* fosse muito fraco.

Depois da passagem por um cruzamento é sempre verificado se o utilizador seguiu o caminho correto, tentando-se encontrar o ponto de controlo que foi gravado aquando da criação do *roadbook*. Se percorrida uma determinada distância não for encontrado o ponto de controlo conclui-se que o caminho seguido não foi o correto e inicia-se o processo de correção da rota de uma forma automática.

Este tópico é abordado com mais detalhe na secção 4.4.

### **1.2.5 Processo de correção de rota**

Depois de iniciado o processo de correção da rota, o utilizador poderá escolher entre fazer a correção usando um mapa ou usando uma seta indicadora do caminho correto. Se for escolhida a opção do mapa irá ser apresentado um mapa onde se poderá ver a localização do dispositivo móvel e do caminho correto. Usando a informação presente neste mapa o utilizador deverá deslocar-se na direção do caminho correto, sendo que a sua localização irá mudar no mapa à medida que se for deslocando. Se for escolhida a opção alternativa, a navegação será feita tendo

como meio de orientação a referida seta que estará sempre a apontar para a localização do caminho correto. Ao mesmo tempo, o utilizador tem sempre disponível a distância em linha reta ao caminho correto.

Para o processo de correção utilizando um mapa foram usados mapas *OpenStreetMaps* em conjunto com as ferramentas *osmdroid*. O servidor de *Tiles* usado foi o *Mapquest*.

Relativamente ao processo de correção em que se utiliza a seta, o cálculo do ângulo de rotação é realizado com base no rumo do utilizador em conjunto com a direção em que se encontra o caminho correto.

Este tópico é abordado com mais detalhe na secção 4.5.

### **1.2.6 Servidor *web* para partilha de *roadbooks***

É possível o utilizador partilhar os *roadbooks* criados na *web*, utilizando um serviço central. Da mesma forma é possível o utilizador fazer *download* para o seu dispositivo móvel de *roadbooks* criados por outros utilizadores.

Foram utilizados *Web Services*, para que toda a informação recolhida para a criação dos *roadbooks* seja armazenada numa base de dados *SQLite* e possa ser enviada para um servidor *web* onde ficará armazenada numa base de dados *MySQL*.

Este tópico é abordado com mais detalhe na secção 4.8.

### **1.2.7 Interface da aplicação**

No que respeita à criação da interface da aplicação tiveram de se ter em conta a densidade de pixéis e os diferentes tamanhos dos ecrãs dos dispositivos móveis. Considerou-se também importante organizar os textos que fazem parte da interface em ficheiros de recursos próprios, de forma a permitir a criação de estilos de texto específicos, assim como a definição dos idiomas português e inglês. Este tópico é abordado com mais detalhe na secção 4.9.

### **1.2.8 Detecção de contextos**

Numa aplicação com detecção de contextos, sempre que houver uma mudança no ambiente em que a aplicação está a ser executada, a mesma deve adaptar-se a essas mudanças de forma automática. Através da combinação de diversos parâmetros do contexto em que se encontra o utilizador, poderão ser criados comportamentos diferentes para uma determinada aplicação. O comportamento da aplicação *Naturepathfinder* assenta numa metodologia de detecção de contextos, na medida que a aplicação faz a monitorização constante de parâmetros como a velo-

cidade, a localização geográfica e a exatidão do sinal *GPS*, que lhe permitem proceder a ações diferentes conforme ocorram mudanças nestes parâmetros. Desta forma é possível à aplicação realizar uma navegação automática, assim como controlar se ocorreram erros de navegação e proceder automaticamente à sua correção. Para além dos comportamentos visíveis para o utilizador, existem também outras ações que a aplicação realiza tendo em conta as alterações do meio envolvente, tais como a adaptação do número de leituras a partir do sensor *GPS* efetuadas num determinado período de tempo, com o objetivo de poupança de energia. Esta ação será controlada tendo em conta a velocidade atual do dispositivo e a sua distância ao cruzamento seguinte.

Este tópico é abordado com mais detalhe na secção 4.7.1.

## 1.2.9 Aplicação Naturepathfinder

A aplicação *Naturepathfinder* é direcionada para a plataforma *Android* e foi desenvolvida na linguagem de programação Java, com recurso ao *Android SDK* e ao *Android Development Tools*.

Durante o desenvolvimento da aplicação foram realizados testes de desempenho à aplicação, com o objetivo de se irem analisando aspetos a corrigir e assim se atingir o correto funcionamento da mesma. Foram também realizados testes de usabilidade em que se analisou a interação entre os utilizadores e a aplicação. Após finalizado o desenvolvimento da aplicação foi realizado o teste do grau de satisfação da aplicação final, com o objetivo de se verificar se a aplicação cumpria todos os objetivos propostos num cenário real.

Após efetuados os testes à aplicação com o público-alvo pôde concluir-se que se obteve resultado positivo no que respeita à usabilidade e comportamento durante o processo da criação e navegação de um *roadbook*, pois a maior parte das pessoas que utilizaram a aplicação durante a fase de testes conseguiu criar um *roadbook* e navegar um *roadbook* criado por outra pessoa. Todas as pessoas admitiram que a aplicação foi ao encontro das suas expectativas e que este método de criação e navegação de *roadbooks* traz grandes vantagens relativamente à utilização dos tradicionais *roadbooks* em papel.

A aplicação *Naturepathfinder* será disponibilizada no serviço *Google Play*<sup>2</sup> para que possa ser utilizada pelo público-alvo. A partir desta fase será possível obter feedback por parte de mais utilizadores e eventualmente proceder-se a alterações e correções que possam ser sugeridas.

---

<sup>2</sup> <https://play.google.com/store>



Foi desenvolvido o manual do utilizador da aplicação *Naturepathfinder*, que contém a descrição de todas as funcionalidades da aplicação. Este manual pode ser consultado pelo utilizador a partir da própria aplicação.

Para além das utilizações convencionais de um *roadbook*, em percursos na natureza, a aplicação pode também ser utilizada em percursos urbanos e na área do turismo, na medida que poderá ser utilizada para guiar os utilizadores através de roteiros turísticos.

Na secção seguinte serão apresentados os objetivos definidos para o projeto.

### 1.3 Objetivos do projeto

O objetivo geral do projeto consistiu no desenvolvimento de uma aplicação que permitisse a criação e posterior utilização de *roadbooks*, assim como a deteção e assistência ao utilizador para a correção de erros de navegação.

A lista a seguir apresenta os objetivos específicos do projeto que correspondem às funcionalidades que foram definidas e implementadas para o *roadbook* digital:

- Editor gráfico para criação de *roadbooks* no terreno. Pretende-se que o editor assista o utilizador na criação das notas de navegação através do uso de símbolos predefinidos ou criados pelo utilizador e dados provenientes dos sensores do dispositivo móvel. Sempre que o utilizador pretender gravar as informações referentes a um cruzamento irão ser armazenadas as coordenadas desse cruzamento e o utilizador irá inserir toda a informação necessária, nomeadamente um pequeno comentário e o desenho do esquema que representa a direção que se deve tomar nesse cruzamento. Toda esta informação é guardada em base de dados e será apresentada ao utilizador aquando da navegação. Serão também capturadas automaticamente as coordenadas de um ponto de controlo que serão usadas para verificar, na fase da navegação, se se está a seguir o caminho correto.
- Assistente de navegação com base em *roadbooks* existentes. Pretende-se que o assistente apresente as notas de navegação de forma automática através da deteção do contexto de navegação (localização, direção, erros de navegação). A aplicação verifica constantemente a localização atual do utilizador e assim que este se encontre perto de um ponto onde exista uma nota (cruzamento) irá avisar o utilizador e passar à nota seguinte. Caberá ao utilizador interpretar a imagem do esquema apresentada e tomar a direção correta.
- Assistente para deteção de erros de navegação e recuperação da navegação pelo caminho correto se houver um erro de navegação. Depois de cada cruzamento a aplicação irá

verificar se o utilizador seguiu pelo caminho correto, verificando se passou no ponto de controlo gravado automaticamente aquando da gravação das notas do *roadbook*. Se for detetado um erro de navegação a aplicação irá indicar o caminho correto ao utilizador, usando uma seta que indicará sempre a direção do caminho correto ou utilizando um mapa onde poderá consultar a posição atual e aposição do caminho correto. Todos os dados utilizados pela aplicação estão guardados na base de dados, durante a gravação das notas.

- Resolver problemas específicos dos sistemas tradicionais, tais como dificuldade e complexidade na medição das distâncias entre cada cruzamento, deteção de erros de navegação e respetiva correção da rota, dificuldade na identificação da nota/cruzamento atual, complexidade da criação de um *roadbook* no que respeita a todos os dados que se devem registar manualmente, necessidade de equipamento extra para cálculo de distâncias tanto na criação como na navegação de *roadbooks*.
- Implementar claramente uma aplicação com deteção de contextos, em que sempre que houver uma mudança no ambiente em que a aplicação está a ser executada a mesma deve adaptar-se a essas mudanças de forma automática, nomeadamente durante o processo de gravação das notas dos *roadbooks*, da navegação de *roadbooks* e durante a correção da rota depois de ter ocorrido um erro de navegação.
- Será importante que a aplicação tenha um elevado nível de usabilidade. Para se atingir este objetivo usou-se um método analítico para aferir a usabilidade de aplicações informáticas, neste caso as Heurísticas de Nielsen (Nielsen, 1994). Estas heurísticas estão representadas no Anexo D.

## 1.4 Estrutura do Relatório

Este trabalho encontra-se estruturado em 6 capítulos, dos quais o primeiro é composto pela introdução ao trabalho desenvolvido. Neste primeiro capítulo é definido o enquadramento e a motivação para o desenvolvimento deste projeto. É também descrito o projeto no que respeita aos módulos que o compõem. São ainda definidos os objetivos gerais e específicos deste trabalho.

No segundo capítulo são apresentadas algumas tecnologias de *roadbooks*, desde os tradicionais *roadbooks* em papel, passando por dispositivos artesanais de leitura de *roadbooks*,

*roadbooks* digitais, e aplicações para dispositivos móveis já existentes para criação e leitura de *roadbooks*.

Relativamente ao terceiro capítulo, vai ser realizada uma análise de requisitos de características no que respeita a aplicações para dispositivos móveis com o objetivo da criação e leitura de *roadbooks*. Será também feito neste capítulo um estudo e a respetiva análise da importância destas características para potenciais utilizadores da aplicação desenvolvida.

O quarto capítulo é referente à descrição de toda a implementação da aplicação desenvolvida, começando por se falar da arquitetura geral do sistema, passando pela explicação de como foram implementadas as bases de dados, o editor gráfico para desenhar as notas, o módulo de gravação das notas, o módulo para navegação de *roadbooks* e o processo de correção da rota. É também neste capítulo que se aprofunda o conceito de deteção de contextos e se referem as técnicas utilizadas para a criação da interface da aplicação.

No quinto capítulo são apresentados testes e respetivas análises, feitos durante o desenvolvimento da aplicação e na fase final do desenvolvimento do projeto.

O último capítulo contém as conclusões gerais deste trabalho, analisa os seus principais resultados, e apresenta algumas perspetivas de desenvolvimentos futuros.

## Capítulo 2

# 2. Tecnologias de *roadbooks*

Os *roadbooks* são esquemas em papel que servem de guia para um trajeto a percorrer, normalmente em atividades ao ar-livre. São elaborados através de notas registadas, com alguma antecedência, durante o reconhecimento e planificação do percurso.

Estes guias podem ser usados em passeios pedestres, passeios de bicicleta, passeios em viaturas todo-terreno, ou qualquer evento deste género. Também podem ser usados em eventos culturais, servindo como guias turísticos.

Tal como se pode ver na Figura 1, um *roadbook* deve disponibilizar informação referente ao percurso a percorrer, nomeadamente um esquema representando a direção a tomar na próxima nota, qual a distância entre a nota atual e a próxima nota, qual a distância total percorrida até à nota atual e, se necessário, uma observação referente a cada nota.








Nota Nº	PARCIAIS (Km)	TOTAIS (Km)	DIRECÇÃO	OBSERVAÇÕES
1	2,930	13,550		P. <sup>ma</sup> Lanhoso CABEZEIRAS    Segue P.P.
2	1,090	14,640		AMARES    Segue P.P.
3	0,620	15,260		Segue P.P.
4	1,420	16,680		MARCO S. João de Rei    Segue P.P.
5	0,440	17,120		Segue P.P.
6	0,420	17,540		Segue P.P.
7	0,470	18,010		Segue P.P.

Figura 1 - Exemplo de uma página de um *roadbook* tradicional<sup>3</sup>

<sup>3</sup> Retirado de: <http://landmania.pt/forum/viewtopic.php?f=33&t=801>

Um *roadbook* é criado durante o reconhecimento ou definição do percurso, normalmente “no terreno”, o que pode dificultar a tarefa, pois existe a possibilidade de acontecerem erros ao criar cada nota. A criação de um *roadbook* implica que tenham de se registar todas as distâncias entre cada ponto em que é necessário efetuar uma mudança de direção, assim como a distância total percorrida. Para além destes registos, é necessário criar o esquema que representa a direção a tomar em cada cruzamento, ou quando existe necessidade de informar o utilizador de algum detalhe do percurso.

É importante, depois de o *roadbook* estar concluído, voltar a realizar o percurso e confirmar se todas as distâncias e notas estão corretas, pois basta uma distância estar errada, ou um esquema de direção de uma nota estar mal desenhado, para comprometer todo o percurso e fazer com que o utilizador se perca. Quando acontece um utilizador de *roadbook* tradicional perder-se, este tem duas opções: tentar identificar um dos esquemas de direção no terreno e continuar o percurso a partir desse ponto, ou voltar ao início do *roadbook* e recomeçar o percurso. A primeira opção pode ser difícil, na medida que normalmente nos esquemas apenas aparece a direção a tomar nos cruzamentos, sem muitas referências ao terreno em redor, pelo que poderá ser penoso ao utilizador conseguir identificar um esquema. A segunda opção também poderá ser difícil, pois o utilizador poderá encontrar-se já distante da nota inicial.

Existem outras dificuldades associadas à criação de um *roadbook*. Por vezes na sua criação tenta-se que o desenho da nota corresponda tão fielmente quanto possível à realidade do terreno. No entanto, deve assumir-se que o desenho da nota não terá que corresponder exatamente à realidade do terreno, bastando que permita ter a noção de qual o caminho a seguir pois é praticamente impossível, através de esquemas, fazer corresponder a nota à representação real do ponto em que se vai realizar uma mudança de direção. Será pois fundamental, escrever uma pequena observação em notas mais complexas.

As secções a seguir apresentam as diferentes tecnologias usadas atualmente para a criação e utilização de *roadbooks*.

## 2.1 *Roadbook* tradicional em papel

O tradicional *roadbook*, como referido no capítulo anterior é um conjunto de indicações impressas que o utilizador deverá seguir para completar o percurso. Trata-se da forma tradicionalmente utilizada para realizar percursos, usando *roadbooks*.

A utilização de um *roadbook* implica a leitura e correta interpretação de cada uma das notas antes da chegada à localização respetiva. O utilizador deverá estar atento à distância per-

corrida e, quando atingir a distância apresentada na nota seguinte, deverá seguir as indicações apresentadas, seguindo o caminho indicado. Como exemplo pode ver-se na Figura 2 a representação de uma nota que indica ao utilizador que este deverá virar à direita.

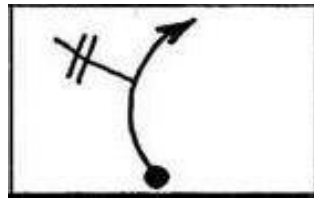


Figura 2 - Esquema de uma nota de roadbook

Sempre que exista, o utilizador deverá estar atento ao campo observações, que poderá ter indicações fundamentais, especialmente no caso de existir algum perigo no percurso a percorrer.

No caso de um passeio pedestre a leitura e interpretação de um *roadbook* é relativamente simples, pois existe muito tempo para analisar e avaliar as notas. Neste tipo de utilização, como o *roadbook* é em papel, este poderá danificar-se no decorrer do percurso ou ficar ilegível se for molhado.

No que respeita a percursos efetuados em veículos automóveis, é típico ser o passageiro a ler o *roadbook* e transmitir a informação ao piloto (Figura 3), orientando-o na direção correta, o que exige uma boa sincronização entre ambos para que não surjam erros de navegação.



Figura 3 - Utilização de roadbook de forma tradicional, em papel<sup>4</sup>

<sup>4</sup> Retirado de: <http://www.seriouswheels.com>

Neste caso existe a necessidade da utilização de um aparelho denominado de *terratrip* (Figura 4), cuja função principal é indicar as distâncias percorridas entre cada uma das notas, com uma medição de 10 em 10 metros. A utilização de um *terratrip* implica a aquisição e instalação da sonda que medirá a distância percorrida. Esta sonda será instalada no cabo do conta-quilómetros. No caso de veículos com um conta-quilómetros digital, a sonda é substituída por uma interface eletrónico que se instala diretamente no conta-quilómetros do veículo. Durante a navegação, este dispositivo terá de ser operado pelo copiloto, em conjunto com o *roadbook* em papel, que a cada nota terá de colocar as distâncias parciais a zero no *terratrip*, de forma a medir corretamente a distância até à próxima nota.



Figura 4 - Dispositivo terratrip para controlo de distâncias<sup>5</sup>

Devido aos custos financeiros deste aparelho e da respetiva sonda, normalmente este sistema é utilizado usualmente em provas profissionais, pelo que um utilizador casual de *roadbook* terá de realizar controlo das distâncias percorridas, recorrendo ao conta-quilómetros do próprio veículo. Esta situação dificulta bastante a operação de interpretar o *roadbook* corretamente, na medida que, normalmente, o copiloto não consegue ler os dados do conta-quilómetros do veículo, tendo de ser o próprio piloto a indicar qual a distância percorrida o que poderá gerar erros de navegação, ou mesmo de condução, pois o piloto, para além de estar atento ao caminho a percorrer, terá de estar atento ao conta-quilómetros do veículo.

Normalmente o registo destes dados é feito em folhas de papel onde se desenham os esquemas e se apontam os valores numa folha predefinida, criando diretamente no terreno o esquema pretendido, escrevendo a informação para cada cruzamento e adicionando manualmente as distâncias entre cada nota (Figura 5).

<sup>5</sup> Retirado de: <http://www.terratrip.com/dealer%20area.htm>

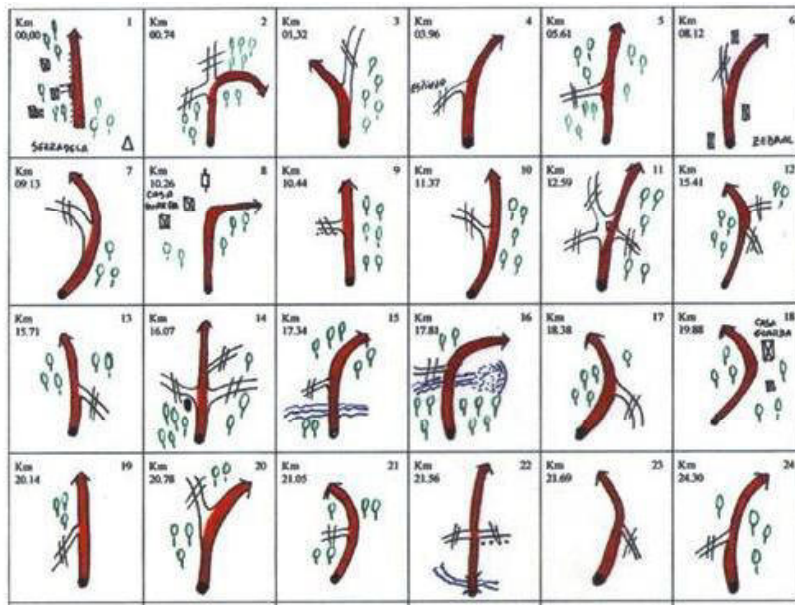


Figura 5 - Exemplo de um roadbook desenhado à mão<sup>6</sup>

Um *roadbook* criado utilizando apenas este tipo de registo pode levar a erros de interpretação aquando da utilização no terreno, pois os esquemas estarão dependentes da habilidade de desenho do criador. Poderá também ser difícil a interpretação das informações dos cruzamentos, pois foram escritas “à mão”. Para se ultrapassarem estes problemas, normalmente utiliza-se uma ferramenta informática que será usada para reproduzir digitalmente a informação registada anteriormente ao definir-se a rota. Uma das ferramentas informáticas mais utilizadas para este processo é uma simples folha de cálculo (Figura 6).

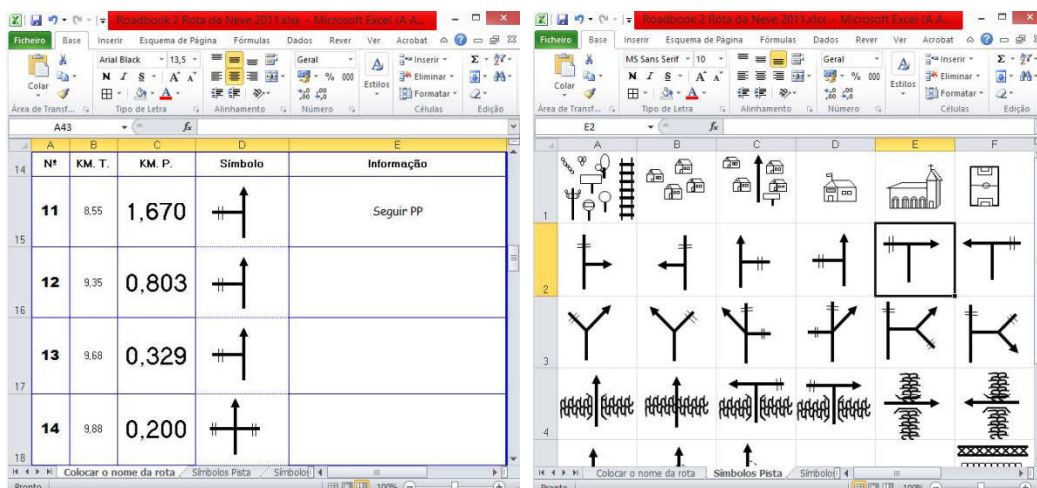


Figura 6 - Utilização de folha de cálculo para criar roadbook

<sup>6</sup> Retirado de: <http://www.cm-vminho.pt/9438>



Com este tipo de ferramentas é possível organizar corretamente e de uma forma bem estruturada toda a informação que interessa que esteja presente num *roadbook*. Para além disso podem ser utilizados esquemas que tenham sido desenhados previamente.

Existem também ferramentas informáticas específicas para a organização da informação e criação de *roadbooks*, como é o caso da Roadbook Assistant<sup>7</sup>. Como se pode ver na Figura 7 existem já esquemas pré desenhados que podem ser adicionados a uma nota específica, juntamente com as distâncias registadas previamente no terreno e a respetiva informação.

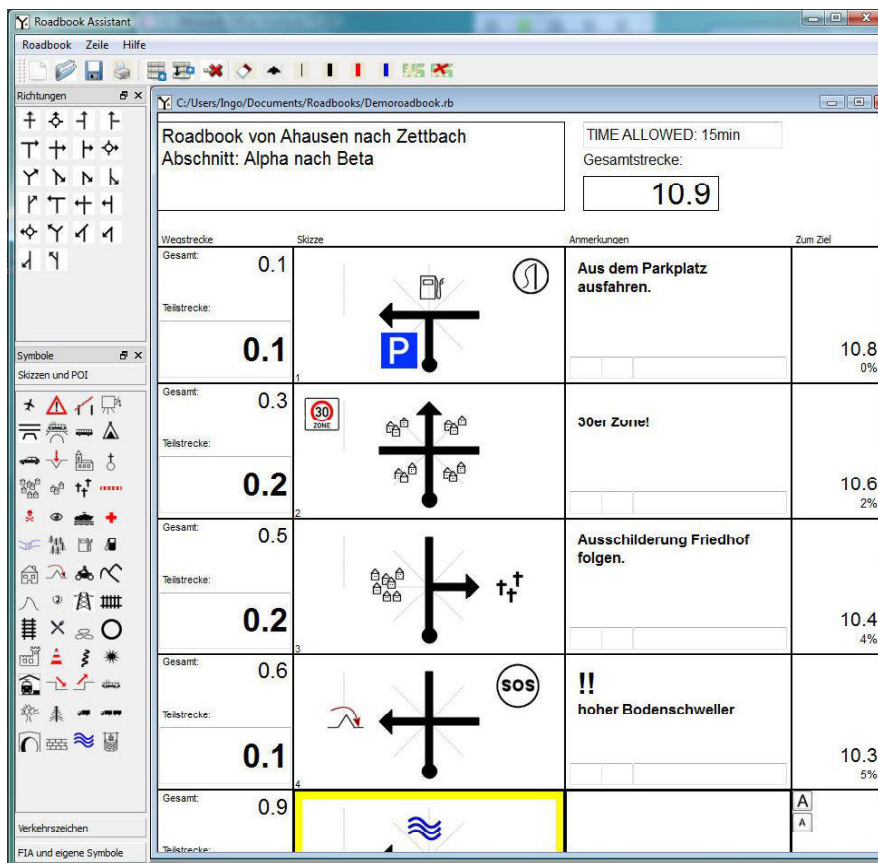


Figura 7 - Interface da aplicação roadbook Assistant<sup>8</sup>

Depois de utilizada uma ferramenta informática, num computador, para organizar a informação recolhida no terreno há necessidade de refazer o percurso e verificar se toda a informação está correta. Isto torna-se necessário, pois para além de erros que possam ter acontecido no momento do registo no terreno, existe a possibilidade de acontecerem erros quando se passa a informação do papel para uma destas aplicações informáticas.

<sup>7</sup> <http://www.thedownloadplanet.com/catalog/item/roadbook-assistant/>

<sup>8</sup> Retirado de: <http://www.wetexx.de/index.php/software/shareware/roadbook-assistant>

## 2.2 Dispositivos enroladores de *roadbook*

O processo de leitura e interpretação de um *roadbook* por um piloto de mota será mais difícil, na medida que não existe um copiloto. Para resolver este problema, existem dispositivos que permitem ao motociclista conduzir e, ao mesmo tempo, consultar o *roadbook* (Figura 8). Trata-se de um aparelho elétrico que permite ao utilizador pressionar um botão para avançar para as notas seguintes ou outro botão para recuar para as notas anteriores (Figura 9). Neste caso, o utilizador terá também de estar atento às distâncias percorridas, usando o contaquilómetros da mota, de forma a poder interpretar o *roadbook* corretamente. Pode concluir-se que este processo será algo complicado e poderá dar origem a erros de interpretação.



Figura 8 - Dispositivo eletrônico para enrolar um *roadbook* em papel, utilizado em motas<sup>9</sup>



Figura 9 - Botões de controlo das notas

Este tipo de dispositivo possui normalmente a opção de enrolar e desenrolar manualmente o *roadbook* em papel, tal como se pode ver na Figura 10, para resolver uma situação de avaria elétrica. São aparelhos que, apesar de usarem *roadbooks* em papel, são resistentes à água e ao pó. Os custos de aquisição e o tamanho dos dispositivos serão, sem dúvida, uma desvantagem para utilizadores casuais, sendo este tipo de leitores de *roadbook* utilizados apenas por pilotos profissionais.

<sup>9</sup> Retirado de:

[http://www.f2r.pt/epages/f2r.sf/en\\_GB/?ObjectPath=/Shops/f2r/Products/RB730/SubProducts/RB730-001](http://www.f2r.pt/epages/f2r.sf/en_GB/?ObjectPath=/Shops/f2r/Products/RB730/SubProducts/RB730-001)



Figura 10 - Opção para enrolar ou desenrolar o roadbook manualmente<sup>10</sup>

Existem também muitas versões artesanais, com custos de produção reduzidos, construídas pelos diversos utilizadores, que mais não são do que dispositivos manuais (Figura 11) que permitem movimentar o *roadbook* em papel para a frente e para trás, permitindo assim ao motociclista seguir o percurso. Estes aparelhos serão os mais difíceis de utilizar, pois implicam enrolar e desenrolar manualmente o *roadbook*, estar atento às distâncias marcadas no conta-quilómetros da mota e conduzir a mesma.

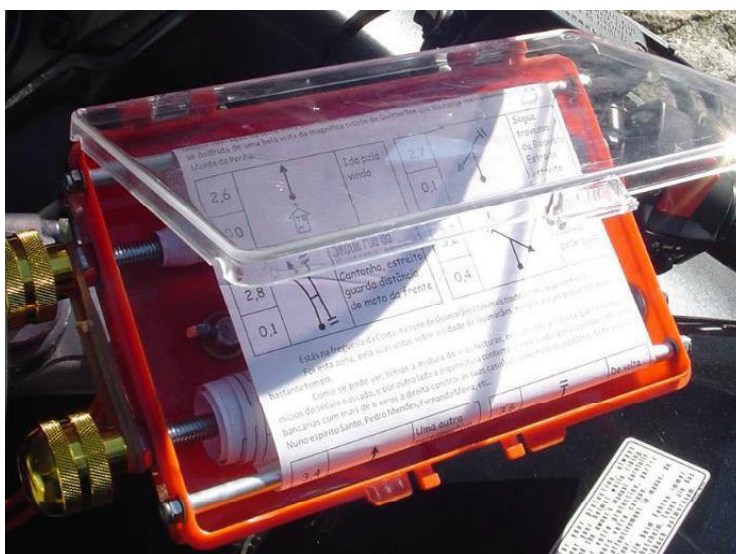


Figura 11 - Dispositivo artesanal enrolador de roadbook<sup>11</sup>

Neste caso o criador do leitor de *roadbook* usou uma caixa plástica com tampa transparente, cortou as divisórias interiores e utilizou dois ferros com rosca para servir como eixo de

<sup>10</sup> Retirado de: <http://www.motorallyservices.co.uk/products/f2r-roadbook>

<sup>11</sup> Retirado de: <http://www.forumtransalp.com/t1104-leitor-de-roadbook-caseiro>

rotação aos *roadbook* em papel e utilizou mais dois ferros lisos para servirem de esticador do papel. Trata-se de um sistema bastante arcaico, muito sujeito a avarias.

Existem também exemplos de leitores de *roadbook* elétricos artesanais, tal como se pode ver na Figura 12. Neste caso o criador utilizou uma caixa metálica como base para todo o sistema, que irá albergar o *roadbook*, os ferros guias e o motor que irá realizar a rotação do papel.

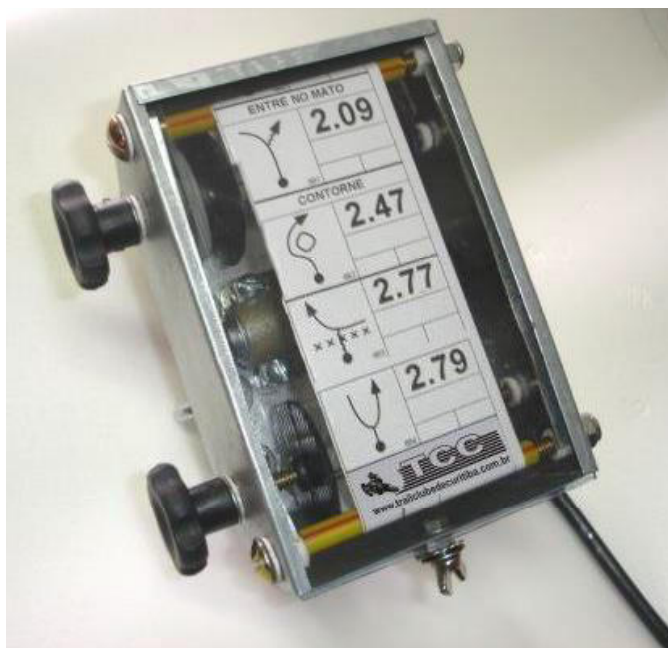


Figura 12 - Leitor de roadbook artesanal, mas elétrico<sup>12</sup>

O motor utilizado, neste exemplo, foi o de um aparafusador elétrico que vai permitir enrolar e desenrolar o papel do *roadbook*, possibilitando assim navegação entre as notas do mesmo. Como se trata de um dispositivo fabricado artesanalmente, poderão acontecer falhas e avarias na sua utilização.

Ainda dentro deste género de leitores de *roadbooks* (enroladores de papel), foram encontrados alguns em que os criadores decidiram acoplar-lhes um *GPS* (*Global Positioning System*) de navegação, que será responsável pelo cálculo das distâncias, tal como se pode ver na Figura 13. No entanto, não existe qualquer ligação entre o *GPS* e o sistema de leitura de *roadbooks*, sendo a passagem para a nota seguinte feita pelo utilizador, pressionando um botão para avançar e outro para recuar nas notas.

---

<sup>12</sup> Retirado de: <http://www.deitamato.com.br>



Figura 13 - Dispositivo enrolador de roadbook com GPS acoplado<sup>13</sup>

Todos estes dispositivos têm a desvantagem de utilizarem papel, que facilmente pode ser danificado, acrescentando a isto o facto de serem dispositivos com sistemas mecânicos, que podem avariar. Outra desvantagem destes aparelhos é o seu tamanho, pois não são fáceis de transportar e instalar. Outro ponto negativo a assinalar é o facto de que, todos estes tipos de leitores de *roadbook*, incluindo os tradicionais só em papel, necessitam de um sistema de luz auxiliar para percursos realizados durante à noite.

## 2.3 Resumo dos principais problemas da utilização dos *roadbooks* tradicionais em papel

Como neste tipo de *roadbooks* tradicionais em papel o registo dos dados é feito manualmente, poderão existir erros no cálculo das distâncias e é sempre necessário fazer o percurso diversas vezes para se verificar se todas as distâncias entre cruzamentos estão corretas. O sistema mais utilizado para efetuar as medições das distâncias é o conta-quilómetros da viatura utilizada, o que leva constantemente a erros, pois há muitos fatores que influenciam a distância medida fazendo com que conta-quilómetros do veículo indique distâncias diferentes das reais:

- Conta-quilómetros das viaturas mal calibrados;

---

<sup>13</sup> Retirado de: <http://portocity.olx.pt/pictures/leitor-de-roadbook-electrico-con-GPS-iid-12032145>

- Utilização de pneus com dimensões diferentes das de origem, que fará com que a distância percorrida não corresponda à distância medida pelo conta-quilómetros da viatura;
- Derrapagem dos pneus – o conta-quilómetros da viatura faz medições mas o veículo não andou a distância medida;
- Percorrer os caminhos encostado ao limite esquerdo ou direito, dará origem a diferentes medições das distâncias;
- O copiloto (que estará a apontar os dados em papel) não consegue ler os dados do conta-quilómetros do veículo, pelo que terá de ser o condutor a verificar estes valores, o que poderá levar a erros de leitura;
- Problemas com o registo de distâncias inferiores a 100 metros, que não são lidos pelos conta-quilómetros analógicos das viaturas.

Na criação ou navegação de um *roadbook* baseado em distâncias registadas manualmente poderão acontecer alguns, ou mesmo todos os casos descritos anteriormente, pelo que assume-se sempre que irá haver erros nas distâncias registadas entre cada cruzamento. Pode mesmo afirmar-se que se se fizerem vários reconhecimentos para verificação de um *roadbook* utilizando diferentes veículos e com diferentes pessoas a conduzir, irão sempre encontrar-se distâncias que se podem considerar erradas, pelo que as estas assumem-se sempre como distâncias aproximadas. Obviamente que, quanto mais pontos tiverem sido registados maior será a probabilidade de haver erros graves que levem a enganar na navegação.

## 2.4 *Roadbooks* digitais

O modo de funcionamento deste tipo de leitores *de roadbooks* é baseado nos *roadbooks* tradicionais em papel. No entanto os esquemas dos pontos de mudança de direção e todo o controlo das distâncias entre cada um dos pontos estão centralizados num único dispositivo, o que facilita muito a tarefa de percorrer o caminho estabelecido e diminui a probabilidade de ocorrência de erros de leitura. Este tipo de dispositivos possui *GPS* próprio.

Na Figura 14 pode ver-se um exemplo deste tipo de leitores, o *Tripy GPS II*, que tem grandes vantagens relativamente aos leitores de *roadbook* tradicionais. Este dispositivo tem como característica principal permitir carregar os *roadbooks*, previamente criados, de uma forma rápida, através da ligação a um computador. Desta forma, os utilizadores podem também criar os seus próprios *roadbooks* usando um computador e partilhar os mesmos na Internet.





Figura 14 - roadbook digita para motos da empresa "Tripy"<sup>14</sup>

Este dispositivo tem implementada uma característica bastante interessante, que é a possibilidade de criação de um trajeto, usando um mapa online, ou utilizando o software fornecido no ato da compra (RoadTracer Pro<sup>15</sup>), em que o utilizador seleciona os pontos por onde pretende passar e o software exporta o ficheiro do *roadbook*, com todos os cruzamentos assinalados com um esquema de setas, como se pode ver na Figura 15.



Figura 15 - Criação automática da nota de roadbook através da definição de pontos do trajeto no software RoadTracer Pro<sup>16</sup>

O utilizador poderá também optar por definir simplesmente um ponto de chegada e o *software* irá criar o *roadbook*, através de um trajeto calculado, analisando todos os cruzamentos

<sup>14</sup> Retirado de: <http://www.tripy.eu/en/tripy-2-GPS-roadbook-all-terrain-4x4-motorcycle-quad>

<sup>15</sup> <http://www.tripy.eu/en/tripy-2-GPS-roadbook-motorcycle-road/road-tracer/benefits>

<sup>16</sup> Retirado de: <http://www.tripy.eu/en/tripy-2-gps-road-book-motorcycle-road>

encontrados entre o ponto de origem e o ponto de destino. Para além do *roadbook*, a navegação poderá ser feita utilizando uma interface que mostra simplesmente o caminho a percorrer, como se pode ver na Figura 16.

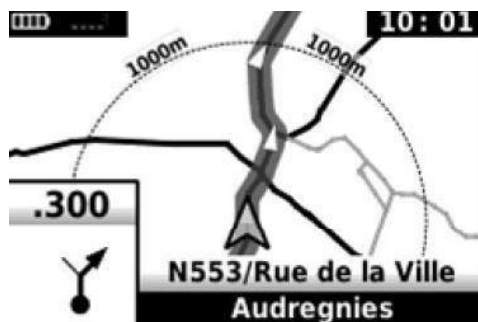


Figura 16 - Menu do Tripy GPS em que se pode ver o caminho a percorrer<sup>17</sup>

Esta última opção, que mostra o caminho a percorrer, será mais útil fora das localidades ou em zonas com poucos pontos de viragem, enquanto a opção *roadbook*, com o esquema de viragem nos cruzamentos, será mais útil em localidades ou zonas com grande número de cruzamentos.

Este aparelho possui também uma opção de gravação, que permite memorizar um caminho percorrido para que mais tarde se possa editar e criar um *roadbook*, usando o *software* fornecido para computador.

Outra opção que também se encontra implementada neste aparelho é a mudança automática para a nota seguinte, sempre que se chega a um cruzamento ou ocorra uma mudança de direção.

Existe também deteção de erros de navegação, pelo que se o utilizador sair do itinerário definido, o dispositivo apresenta uma linha reta entre o ponto atual e o próximo ponto de viragem definido no *roadbook*. Existe uma opção que permite recalcular o percurso de forma a corrigir o erro de navegação cometido e criar um novo itinerário para chegar ao destino. Esta opção pode estar ativa ou não, durante a navegação.

Este tipo de dispositivos tem necessidade de pouca manutenção, na medida que não possuem partes mecânicas. Sendo robustos e resistentes à água, têm óbvias vantagens relativamente aos tradicionais *roadbooks* em papel.

Este *GPS*, para além de apresentar os esquemas das notas, apresenta também as distâncias percorridas, evitando assim a necessidade de aquisição de um *terratrip*, ou da utilização do conta-quilómetros do veículo.

---

<sup>17</sup> Retirado de: <http://www.tripy.eu/en>



Uma desvantagem deste tipo de equipamentos é o seu preço tonando-os pouco acessíveis para utilizadores casuais de *roadbooks*.

## 2.5 Aplicações de *roadbook* para *smartphones* e *tablets*

Existem algumas aplicações para *smartphones* e *tablets* que permitem a leitura de notas de *roadbook* em formato digital.

Uma destas aplicações foi desenvolvida para dispositivos *Android*, pela empresa Marketing Mobilny<sup>18</sup>, da Polónia, intitulada *Mobileroadbook* na *TLC Camp'11* (Figura 17), cujo objetivo é disponibilizar informação aos participantes no evento todo-terreno *Toyota Land Cruiser Camp 2011*.



Figura 17 - Aplicação *Mobileroadbook*<sup>19</sup>

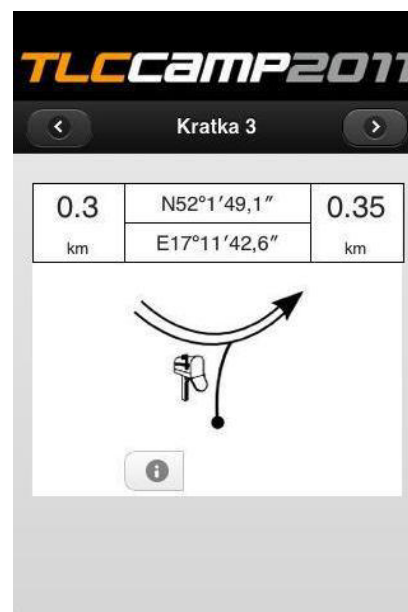


Figura 18 - Nota do *roadbook*

Incluído na informação fornecida existe o *roadbook* do evento, tal como se pode ver na Figura 18, que permite aos inscritos orientarem-se durante o percurso definido para o evento.

Trata-se, no entanto, de um *roadbook* cujas notas têm de ser passadas manualmente, pressionando um botão, quando o utilizador se apercebe que chegou à nota seguinte. Nesta aplicação não existem indicações de distâncias percorridas pelo utilizador, pelo que este terá de usar

<sup>18</sup> <http://www.w3mobi.pl/desktop/home.html>

<sup>19</sup> Retirado de: <http://www.w3mobi.pl/desktop/home.html>

o conta-quilómetros do veículo, ou utilizar um *terratrip*, de forma a saber quais as distâncias percorridas e decidir se deve, ou não, passar para a nota seguinte. Sendo assim, este *roadbook* digital para *smartphone* não apresenta grandes vantagens relativamente a um *roadbook* tradicional.

Para além disto, a aplicação apenas permite utilizar o *roadbook* predefinido para o evento em questão.

Outra aplicação analisada foi a *GPS Road Book*<sup>20</sup>, também para dispositivos *Android*. Relativamente a esta aplicação existe apenas numa versão de testes que desde 2012 até agora não chegou a evoluir para uma versão final. Esta aplicação foi desenvolvida pelo autor português João Simões. Trata-se de uma aplicação desenvolvida em Java, que não permite criar *roadbooks*, mas permite utilizar *roadbooks* previamente criados numa folha de cálculo e guardados em formato CSV (*Comma-Separated Values*). A Figura 19 apresenta um ecrã desta aplicação.



Figura 19 - Interface da aplicação *GPS Road Book*<sup>21</sup>

Também nesta aplicação o utilizador terá de, manualmente, passar para a nota seguinte ou regressar à anterior, pois apesar de este *software* informar das distâncias percorridas, não muda de nota automaticamente.

Outro exemplo estudado foi o *Road Book Digital Totem*<sup>22</sup> da empresa *Totem*, que fabrica computadores de bordo para veículos de Rally (Figura 20). Neste caso esta aplicação também é para dispositivos *Android*.

<sup>20</sup> <http://jsimoes.selfip.com/GPStrip/?file=GPS-road-book>

<sup>21</sup> Retirado de: <http://jsimoes.selfip.com/GPStrip/?file=GPS-roadbook>

<sup>22</sup> <http://www.rallyvirtual.com.br/sitenovo/>

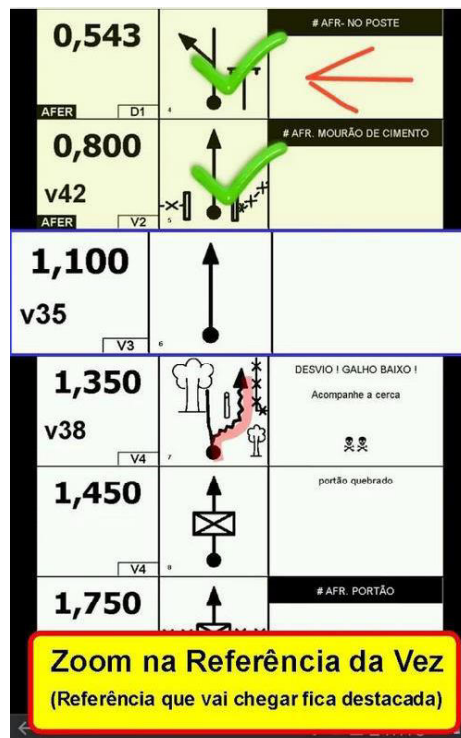


Figura 20 - Interface da aplicação Road Book Digital Totem<sup>23</sup>

Trata-se de um leitor de *roadbooks* que mostra as indicações e direções a seguir para se realizar um percurso. Neste caso a passagem de uma nota para a seguinte é feita automaticamente à medida que o veículo se vai deslocando, no entanto o dispositivo móvel deve estar conectado a um dispositivo fabricado pela mesma empresa, o Colosso EVO (Figura 21) que funciona como hodômetro, medindo as distâncias percorridas pelo veículo e mudando de nota quando for atingida uma distância predefinida no *roadbook* criado.



Figura 21 - Dispositivo Colosso EVO<sup>24</sup>

<sup>23</sup> Retirado de: <https://play.google.com/store/apps/details?id=br.com.totemonline.eroadAutoBlue>

Este é um sistema direcionado para utilização profissional, na medida que é necessário o dispositivo *Android* estar conectado ao Colosso EVO, o que implica custos extra na aquisição deste segundo equipamento. Como se trata de um equipamento que tem de estar ligado ao conta-quilómetros do veículo apenas poderá ser utilizado em veículos motorizados e não para atividades como passeios de bicicleta ou caminhadas. Os *roadbooks* são criados utilizando um *software* próprio - o Totem Planilha (Figura 22).

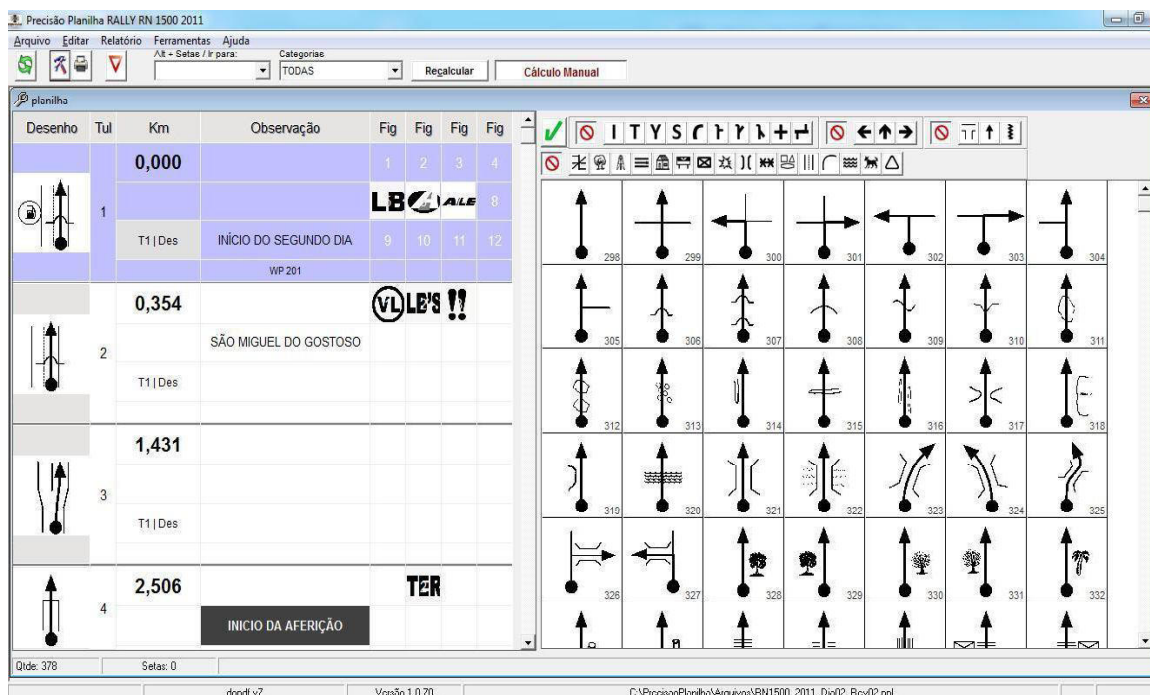


Figura 22 - Interface do software Totem planilha<sup>25</sup>

Foi também analisada a aplicação RoadBook JSC v4 para dispositivos móveis *Android* que permite a criação e utilização de *roadbooks* (Figura 23). Durante a navegação, as notas serão apresentadas automaticamente, sempre que o utilizador atinge as coordenadas de cada um dos cruzamentos guardados.

<sup>24</sup> Retirado de: <http://www.rallyvirtual.com.br/sitenovo/index.php/roadbook-digital>

<sup>25</sup> Retirado de: <http://www.rallyvirtual.com.br/sitenovo/index.php/totem-planilha>



Figura 23 - Interface da aplicação RoadBook JSC v4<sup>26</sup>

Quanto à criação de *roadbooks*, esta aplicação permite ao utilizador guardar as coordenadas *GPS* de cada cruzamento marcando as mesmas num mapa *online* (*Google Maps*) diretamente no dispositivo móvel. Aquando da gravação das coordenadas de cada cruzamento, o utilizador poderá desenhar o esquema representativo do mesmo, que será utilizado durante a navegação para se saber qual a direção a tomar. Nesta parte, o utilizador não poderá usar esquemas previamente construídos, mas terá de desenhar o esquema sempre que grava as coordenadas de um novo cruzamento. Esta técnica pode levar à ocorrência de problemas de navegação, na medida que ao serem marcados os pontos no mapa não há certeza absoluta se o cruzamento, ou o caminho que se está a definir é transitável. Outra coisa a ter em conta será a exatidão das coordenadas marcadas no mapa que será menor do que a exatidão das coordenadas *GPS* marcadas diretamente no terreno com o sensor *GPS* do dispositivo. Estas questões serão de novo abordadas no capítulo da análise final aos sistemas estudados (secção 3.1.1).

<sup>26</sup> Retirado de: [https://play.google.com/store/apps/details?id=jsanchezc.trackbook&hl=pt\\_PT](https://play.google.com/store/apps/details?id=jsanchezc.trackbook&hl=pt_PT)

Outra das aplicações analisadas foi uma, designada de *Roadbook*, da empresa Xdappfactory. Esta aplicação foi desenvolvida para os dispositivos iPhone e iPad e permite a navegação usando esquemas de *roadbook* (Figura 24).

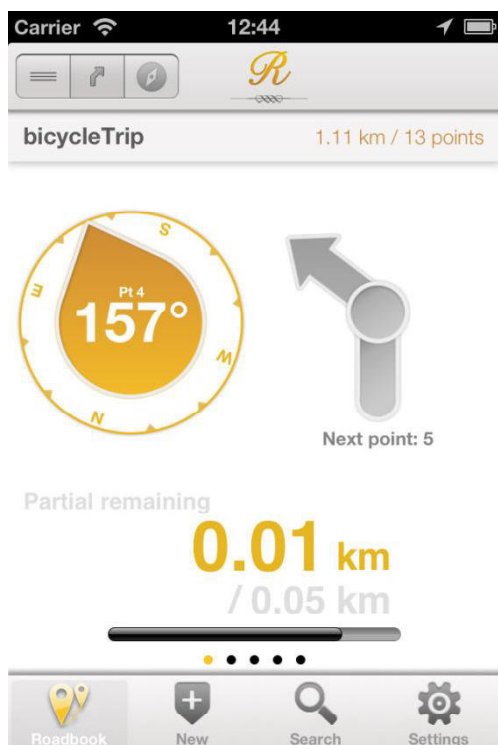


Figura 24 - Aplicação Roadbook em modo de navegação por esquemas<sup>27</sup>

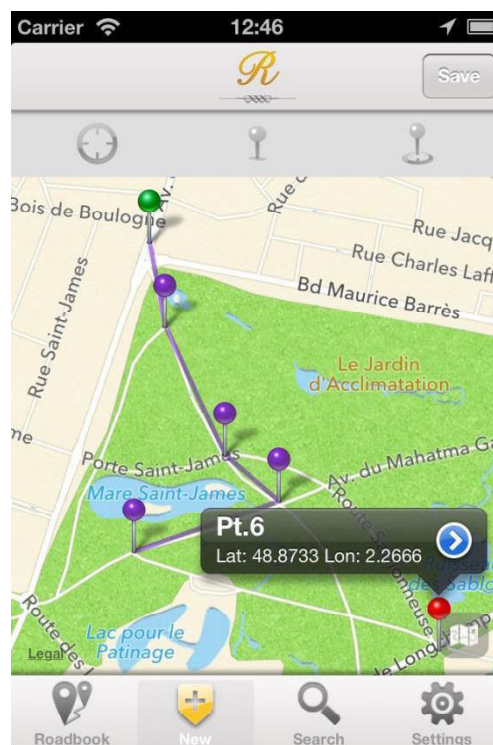


Figura 25 - Aplicação Roadbook em modo de navegação por mapa

Esta aplicação permite fazer o *download* dos diversos percursos diretamente a partir do dispositivo móvel. Tal como alguns sistemas já descritos, também permite ao utilizador a criação de *roadbooks* a partir do método de adicionar coordenadas *GPS* em cada cruzamento, utilizando um mapa, ilustrado na Figura 25. A esta técnica estão inerentes as desvantagens referidas no dispositivo analisado anteriormente.

Por outro lado, esta aplicação permite que sejam capturadas as coordenadas diretamente “no terreno” e com elas gravar o ponto referente ao cruzamento a guardar.

Para além de permitir navegar através da visualização de esquemas que representam os cruzamentos, também possibilita a navegação através da utilização de um mapa, com o respetivo caminho a percorrer assinalado.

Outro *software* analisado, também para iPad foi o Tulip - Roadbook Editor for iPad. Basicamente esta aplicação permite a criação de *roadbooks*, adicionando manualmente todas as

<sup>27</sup> Retirado de: <http://www.xdappfactory.com/wp/>



informações, distâncias e esquemas necessários (Figura 26). Trata-se pois apenas de um editor de *roadbooks*, que no final permite a sua gravação e divulgação em formato PDF.

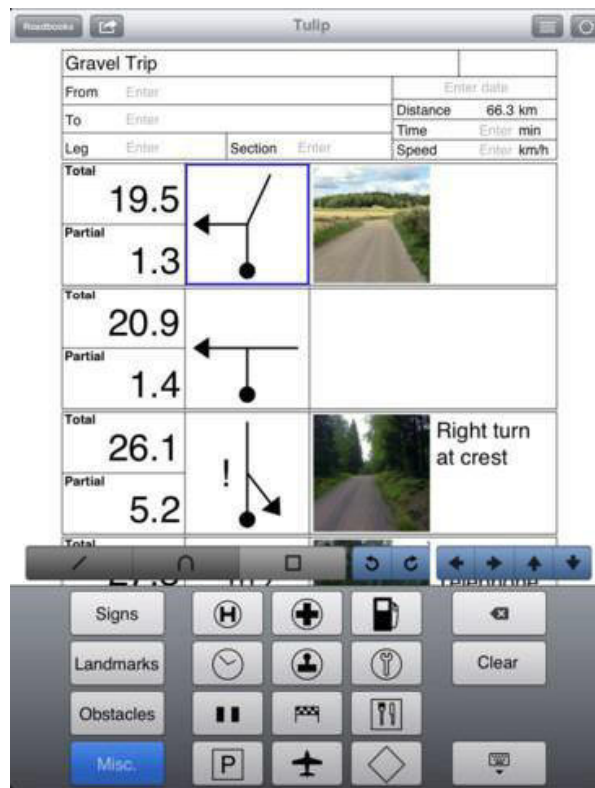


Figura 26 - Interface da aplicação Tulip - roadbook Editor for iPad<sup>28</sup>

A aplicação possui já diagramas e esquemas predefinidos que o utilizador poderá utilizar para criar um *roadbook*. O programa faz a gestão automática das distâncias totais sempre que uma distância parcial é adicionada, o que facilita a criação de *roadbooks*. Para além de se poder criar e editar um *roadbook* “no terreno” devido à portabilidade do iPad, esta aplicação não traz novidades relativamente às aplicações que existem para computador, que permitem criar um *roadbook* a ser impresso, inserindo todos os dados manualmente, como foi descrito na secção 2.1.

Como foi possível comprovar pelo estudo realizado, existem diversos tipos de sistemas de *roadbooks* disponíveis, os quais apresentam diferentes características e funcionalidades. O capítulo que se segue apresenta uma análise comparativa dessas características e funcionalidades com o objetivo de definir os requisitos da aplicação desenvolvida.

<sup>28</sup> Retirado de: [http://pt.appszoom.com/iphone\\_applications/navigation/tulip-roadbook-editor-for-ipad\\_gtkr.html](http://pt.appszoom.com/iphone_applications/navigation/tulip-roadbook-editor-for-ipad_gtkr.html)

## Capítulo 3

# 3. Análise de requisitos

Na atualidade existem algumas aplicações para dispositivos móveis com o objetivo da criação e leitura de *roadbooks*. Foram analisadas aplicações para o sistema *Android* e *iOS*. Todas as aplicações analisadas têm diferentes características, que irão influenciar o seu desempenho aquando da criação das notas e da posterior navegação. Nesta secção apresentam-se as principais características identificadas nos sistemas estudados e analisa-se a importância da sua implementação numa aplicação *roadbook* digital para dispositivos móveis tipo *smartphone*.

## 3.1 Permitir a criação de *roadbooks*

Uma característica muito importante é a possibilidade destas aplicações permitirem aos utilizadores criarem os *roadbooks* que depois serão utilizados na navegação. Tal como foi concluído na secção 2.2, a criação de um *roadbook*, inserindo os dados manualmente, é bastante trabalhosa e se não for feita com bastante cuidado poderá dar origem a erros de navegação. Tendo também em conta a secção 2.1, e com base na experiências feitas diretamente no terreno aquando da criação e navegação de *roadbooks* tradicionais em papel, pode afirmar-se que é bastante mais complexo criar um *roadbook* do que depois utilizar um *roadbook* para percorrer um percurso. Assim sendo, uma funcionalidade que permita a criação de *roadbooks* com o apoio de um dispositivo móvel é uma grande melhoria relativamente à criação de um *roadbook* em papel inserindo os dados manualmente. Das aplicações analisadas apenas duas não permitiam a criação de *roadbooks* e apenas possibilitavam aos utilizadores a navegação com base em *roadbooks* predefinidos. Esta funcionalidade que foi implementada na aplicação desenvolvida.



É também necessário analisar qual a técnica utilizada para adicionar as coordenadas dos pontos dos cruzamentos nos *roadbooks*. Esta análise irá ser feita nas secções seguintes.

### 3.1.1 Marcação das coordenadas dos pontos online na própria aplicação

Algumas das aplicações estudadas permitem a marcação de pontos *online*, diretamente na aplicação, utilizando um mapa. Pode definir-se um “ponto” como sendo as coordenadas geográficas latitude e longitude marcadas no globo terrestre. Os *pontos* são marcados em cada cruzamento. Assim, quando o utilizador se aproxima do cruzamento, aquando da navegação, a nota terá de mudar para dar as indicações do cruzamento seguinte. Esta técnica utilizada para a marcação dos pontos referentes aos cruzamentos diretamente no mapa poderá apresentar alguns problemas, pois poderá ser difícil marcar um percurso corretamente, especialmente para caminhadas, passeios BTT ou passeio em veículos todo terreno, na medida que pode ser difícil identificar todos os cruzamentos no mapa quando não existem estradas. As dificuldades surgem, por exemplo, se se utilizar a vista “Mapa” no *Google Maps*, em que os caminhos fora de estrada não são visíveis (Figura 27).

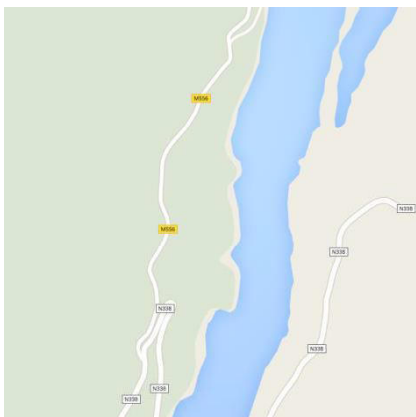


Figura 27 - Imagem Google Maps - vista mapa<sup>29</sup>

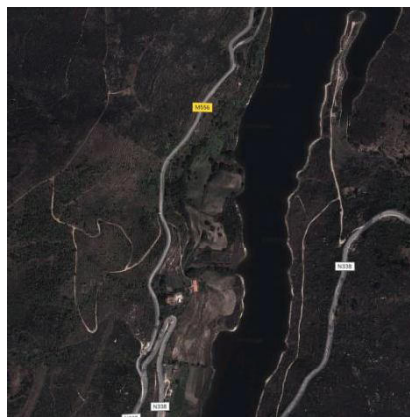


Figura 28 - Imagem Google Maps - vista satélite

Se se utilizar a vista “Satélite” (Figura 28) é possível ver os caminhos fora de estrada e podem ser marcadas os *pontos* e definir o caminho *online*, no entanto, poderão existir zonas em que é

---

<sup>29</sup> Retirado de: <https://www.google.pt/maps/@40.5192878,-7.3325387,16z>

difícil identificar caminhos e especialmente cruzamentos, que poderão estar numa zona com muita vegetação ou zonas onde houvesse muitas nuvens aquando da captura das imagens (Figura 29).



Figura 29 - Imagem Google Maps em modo "Satélite"compouco detalhe devido às nuvens<sup>30</sup>

Num tipo de navegação que se se usam esquemas de *roadbook*, todos os cruzamentos deverão estar sinalizados, pois se isto não acontecer será criada confusão e erros aquando da navegação.

Outra desvantagem da utilização desta técnica é que pode acontecer um caminho estar representado no mapa *online*, mas entretanto já não existe, está cortado, está intransitável devido ao crescimento da vegetação, já tem vedações ou outros obstáculos. No caso de acontecer um destes problemas num dos cruzamentos gravados, todo o *roadbook* fica inutilizado, pois na navegação o utilizador não tem acesso ao mapa e será praticamente impossível tentar encontrar o cruzamento que estaria a seguir ao cruzamento inacessível. Assim sendo, existem claras vantagens em marcar os cruzamentos percorrendo o caminho a definir, pois aí consegue-se saber ao certo como está o trilho que se pretende utilizar.

Ao serem marcadas as coordenadas utilizando um mapa *online*, estas terão menos exatidão do que se fossem marcadas diretamente "no terreno", utilizando o sensor *GPS* do dispositivo móvel. Segundo um estudo realizado sobre *Utilização do Google Earth para obtenção de mapas viários*

---

<sup>30</sup> Retirado de: <https://www.google.pt/maps/@40.8617579,-7.0325475,1616m/data=!3m1!1e3>

*urbanos para SIG*<sup>31</sup>, o Google Earth poderá apresentar erros de exatidão de até 21 metros, enquanto normalmente o sensor de *GPS* de um dispositivo móvel terá erros de exatidão de menos de metade deste valor<sup>32</sup>, dependendo obviamente de diversos fatores que foram alisados na seção 4.4.1.1. Neste tipo de navegação a exatidão é muito importante, especialmente se houver cruzamentos muito próximos uns dos outros, pois se a exatidão for baixa poderão ocorrer erros de navegação, tornando inútil o *roadbook* criado.

Devido às desvantagens apresentadas esta funcionalidade não foi implementada na aplicação desenvolvida.

### **3.1.2 Marcação das coordenadas dos pontos recorrendo ao sensor *GPS* do dispositivo móvel**

Utilizando esta técnica o utilizador vai percorrer todo o percurso e, para cada cruzamento, deverá guardar as suas coordenadas *GPS*. Desta forma o utilizador terá a certeza absoluta que o caminho é transitável e que guardou os pontos de todos os cruzamentos. Outra vantagem é que desta forma os pontos terão mais exatidão do que os gravados *online*, como descrito na seção anterior. Neste caso o utilizador poderá colocar-se no local onde realmente pretende capturar as coordenadas e, tal como se viu na seção 3.1.1, em condições ótimas as coordenadas terão mais exatidão quando marcadas usando o sensor de *GPS* do dispositivo móvel do que as marcadas recorrendo à internet.

Esta foi a técnica implementada na aplicação desenvolvida para gravação das coordenadas dos cruzamentos. Durante a definição do percurso, o utilizador irá indicar à aplicação que pretende guardar um determinado ponto onde se encontra fisicamente e a aplicação irá guardar em base de dados as coordenadas atuais, tal como está descrito na seção 4.4.2.

---

<sup>31</sup> <http://mundogeo.com/blog/2009/07/09/utilizacao-do-google-earth-para-obtencao-de-mapas-viarios-urbanos-para-sig/>

<sup>32</sup> <http://www.andygup.net/how-accurate-is-Android-GPS-part-1-understanding-location-data/>

## 3.2 Necessidade de equipamento extra para marcação das coordenadas dos pontos

Nesta secção foi verificado quais as aplicações necessitavam de equipamento extra para a gravação da localização de cada cruzamento. Uma das aplicações que necessitava de equipamento externo era a Road Book Digital Totem, que usava o *Colosso EV* para medição das distâncias percorridas, tal como descrito na secção 2.5. Sem dúvida que esta obrigatoriedade de usar equipamentos extra se torna numa desvantagem para utilizadores casuais deste tipo de aplicações, pois representa um custo monetário suplementar. Para além disso existe necessidade da aprendizagem das técnicas de utilização deste dispositivo.

Verificou-se que a aplicação Tulip - Roadbook Editor for iPad não fazia a leitura das distâncias percorridas e necessitava do apoio do conta-quilómetros do veículo para medição das distâncias. A medição de distâncias usando o conta-quilómetros de um veículo acaba por não ser muito precisa, na medida que os conta-quilómetros usuais analógicos medem distâncias de 100 em 100 metros. No caso da utilização da aplicação referida para marcação de um percurso pedestre ou de BTT, seria necessário usar um dispositivo *GPS* externo para a medição das distâncias, como por exemplo o *GPS* Etrex 10 (Figura 30).



Figura 30 - GPS Garmin Etrex 10<sup>33</sup>

---

<sup>33</sup> Retirado de: <https://buy.garmin.com/pt-PT/ES/nos-trilhos/portateis/etrex-10/prod87768.html>

Faz-se referência a este *GPS* apenas como exemplo, pois dentro da gama de equipamentos *GPS* que permitem medir distâncias é dos mais baratos, com valores a rondar os 100 euros. De qualquer forma, isto torna-se uma desvantagem, pois para além do manuseamento do dispositivo móvel o utilizador terá também de manipular o dispositivo *GPS*.

A aplicação desenvolvida não necessita de equipamento extra para a gravação das coordenadas dos pontos.

### 3.3 Esquemas das notas predefinidos

Para cada nota guardada num *roadbook* deverá existir um esquema que indica a direção a tomar no cruzamento em questão (Figura 31). Na análise às diferentes aplicações verificou-se que algumas permitiam utilizar esquemas já predefinidos, enquanto noutras o utilizador deveria desenhar todos os esquemas.

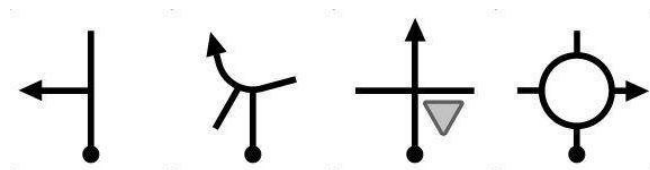


Figura 31 - Exemplos de esquemas indicadores da direção a tomar<sup>34</sup>

A aplicação desenvolvida permite a utilização de esquemas predefinidos que fazem parte da base de dados da própria aplicação.

### 3.4 Desenhar esquemas das notas

Para além da opção que permite a utilização de esquemas predefinidos foi verificado se as aplicações permitiam desenhar esquemas, como é o caso do *software* Totem Planilha (Figura 32), que é um editor gráfico que permite criar desenhos de esquemas para a aplicação RoadBook JSC v4. Esta funcionalidade é importante pois apesar de usualmente os esquemas serem genéricos e

---

<sup>34</sup> Retirado de: <http://scaterrally.car-id.biz/english/typicalrb.html>

poderem ser usados em cruzamentos semelhantes, por vezes surgem cruzamentos com características únicas que convém que sejam registados para identificar da melhor forma o cruzamento.

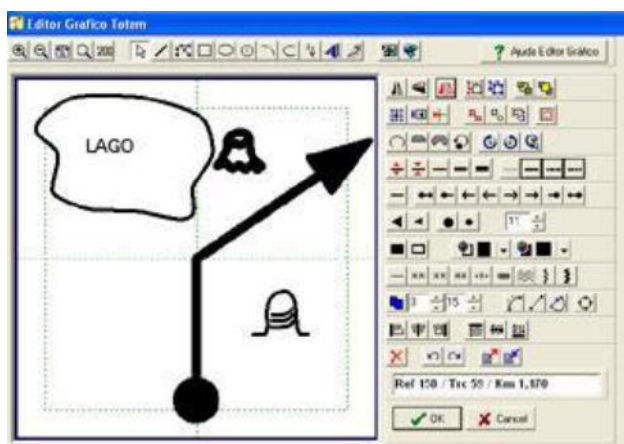


Figura 32 - Desenhar um esquema no software Totem Planilha<sup>35</sup>

Na aplicação desenvolvida foi implementada esta característica, tendo o utilizador a possibilidade de criar os desenhos dos esquemas antes de iniciar o processo de criação do *roadbook* ou durante o processo do registo das notas, caso se aperceba que não existe nenhum esquema predefinido para identificar um cruzamento específico. Este processo está descrito na secção 4.3.

### 3.5 Permitir Download de percursos da *web*

Apesar de não ser fundamental, esta opção também é interessante, na medida que permite aos utilizadores fazerem o *download*, a partir de um serviço central, de *roadbooks* previamente criados por outras pessoas.

A aplicação desenvolvida possui esta característica, pois é possível fazer o *download* de *roadbooks* existentes numa base de dados *MySQL* online, tal como está descrito na secção 4.8.2.

---

<sup>35</sup> Retirado de: <http://www.rallyvirtual.com.br/sitenovo/index.php/totem-planilha>

### 3.6 Permitir partilhar os *roadbooks* criados na *web* num serviço central

Esta funcionalidade tem como objetivo permitir a partilha dos *roadbooks* criados com outros utilizadores, enviando os mesmos para um serviço central que possa ser acedido por outras pessoas para *download* dos *roadbooks* partilhados.

A aplicação desenvolvida possui esta funcionalidade, pois é possível partilhar os *roadbooks* criados numa base de dados *MySQL online*, tal como está descrito na secção 4.8.1.

### 3.7 Permitir navegação manual

Esta funcionalidade dá ao utilizador a possibilidade de avançar e recuar nas notas do *roadbook* manualmente. Desta forma o utilizador irá usar uma aplicação *roadbook* digital de uma maneira muito semelhante à dos tradicionais *roadbooks* em papel, pois vai caber ao utilizador analisar o cruzamento e, se assim entender, passar para a nota seguinte de forma a ver o esquema e as indicações do próximo cruzamento que vai encontrar.

Tal como nos tradicionais *roadbooks* em papel, este método pode levar a erros de interpretação e respetivos erros de navegação, mas por outro lado representa um maior desafio que pode ser apreciado pelos utilizadores. Se esta funcionalidade puder ser complementada com a possibilidade de se fazer também uma navegação automática, a aplicação poderá ser mais interessante e valorizada.

No caso da aplicação desenvolvida é possível a realização de navegação manual sempre que a exatidão do sinal *GPS* for muito baixa e não for suficiente para que a navegação automática se possa realizar sem erros, tal como está descrito na secção 4.5.3.

## 3.8 Permitir navegação automática

Esta é uma funcionalidade que permite que a própria aplicação detete que o utilizador se encontra num determinado cruzamento e passe para a nota seguinte automaticamente. Existem duas maneiras de a aplicação saber se se encontra num cruzamento. A primeira é através da distância percorrida, em que a aplicação tem informação da distância que o utilizador deve percorrer para chegar a um determinado cruzamento e conseqüentemente muda automaticamente para a nota seguinte.

A outra forma é através das coordenadas *GPS*. A aplicação tem armazenadas as coordenadas *GPS* de todos os cruzamentos que foram registados aquando da gravação das notas do *roadbook* e assim que o utilizador se encontrar perto destas coordenadas passa para a nota seguinte automaticamente. Nas secções seguintes serão analisados estes dois métodos.

### 3.8.1 Navegação através de distâncias percorridas

Uma das características que poderá influenciar o desempenho da navegação é o método utilizado para a marcação dos cruzamentos e mudanças de direção.

Algumas aplicações guardam as distâncias percorridas para que quando se navegue usando o *roadbook* o utilizador se aperceba que está a chegar ao cruzamento e haja mudança de nota. Nos tradicionais *roadbooks* em papel utiliza-se também a técnica das distâncias percorridas.

As desvantagens deste tipo de navegação são que quando se cria o *roadbook* as distâncias ficam definidas pela passagem do veículo/pessoa no percurso pretendido, mas basta que durante a navegação se passe no mesmo caminho um pouco ao lado, ou se façam as curvas mais apertadas ou mais ao largo, para que as distâncias comecem a ficar alteradas. Isto pode dar origem a erros de navegação especialmente se houver cruzamentos próximos uns dos outros.

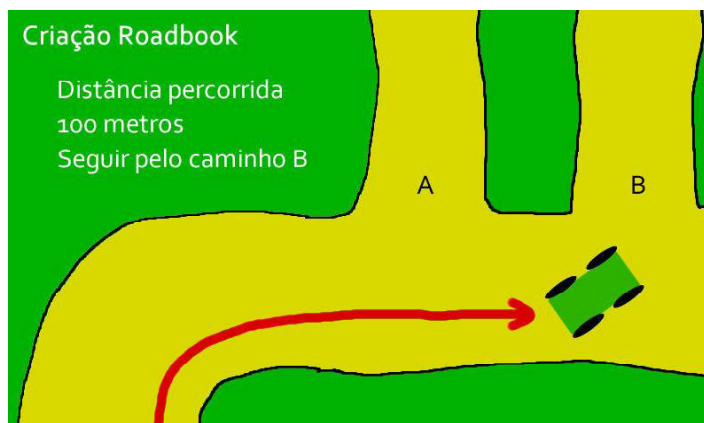


Na Figura 33 pode ver-se uma representação de um exemplo da marcação de um cruzamento. Durante a criação do *roadbook*, depois de percorridos 100 metros o utilizador irá guardar a nota referente ao cruzamento A, indicando que deverá ser efetuada uma curva à esquerda e seguindo por esse caminho que será o definido para o percurso. Repare-se que neste caso o veículo circulou pelo lado exterior da curva.



*Figura 33 - Marcação de pontos através de distâncias – circular pelo lado exterior da curva*

Aquando da navegação, podemos verificar na Figura 34 que o veículo circulou pelo lado interior da curva. O utilizador tinha indicação no *roadbook* que após percorridos 100 metros deveria virar a esquerda, pelo que acabaria por enveredar pelo caminho B, que levaria a um erro de navegação.



*Figura 34 - Navegação utilizando distâncias – circular pelo lado interior da curva*

Neste exemplo utilizou-se um veículo automóvel, no entanto se fosse analisado um exemplo referente a um *roadbook* para um percurso pedestre, ou de BTT, o exemplo seria ainda mais flagrante, na medida que num caminho desta natureza haveria ainda mais liberdade para se deslocar em diferentes partes do mesmo. Outra desvantagem é que se houver algum engano e for necessário recuar um pouco, as distâncias também ficam alteradas, pelo que será necessário voltar ao cruzamento anteriormente definido no *roadbook* e reiniciar a contagem das distâncias. Isto pode ser bastante moroso, especialmente se o cruzamento anterior estiver a alguns quilómetros de distância. Também é importante que durante a navegação o utilizador nunca saia do caminho, nem faça desvios e depois regresse ao caminho principal, pois se isso acontecer as distâncias ficarão também alteradas levando a outro erro de navegação. Devido a estes problemas optou-se por não se implementar este método na aplicação desenvolvida.

### 3.8.2 Navegação através das coordenadas dos pontos

Neste tipo de aplicações, quando é atingido um *ponto* passa-se à nota seguinte que indicará o esquema do próximo cruzamento. Esta técnica tem vantagens relativamente à anterior, na medida que para se saber que se está a chegar a um cruzamento apenas tem de se detetar a coordenada *GPS* do cruzamento, analisando se o utilizador entrou numa pequena área em redor da coordenada *GPS* do cruzamento em questão. Se assim for passa-se à nota seguinte (Figura 35).



Figura 35 - Navegação usando as coordenadas como referência - circular normalmente pelo caminho

O utilizador não necessita de fazer a mesma trajetória dentro de um caminho que foi utilizada aquando da marcação do percurso e se fizer as curvas mais apertadas ou mais ao largo não vai influenciar o resultado final, pois a única coisa que realmente vai influenciar a passagem à nota seguinte é o facto de o dispositivo móvel detetar que o utilizador está perto das coordenadas guardadas e referentes ao cruzamento atual (Figura 36).



Figura 36 - Navegação usando as coordenadas como referência – circular de forma irregular pelo caminho

Quando se utilizam *roadbooks* digitais que utilizam esta técnica é muito importante que aquando da criação do *roadbook* se identifiquem todos os cruzamentos e se guardem as coordenadas de todos, para que durante a navegação o utilizador não se confunda ao passar por um cruzamento sem que exista informação sobre ele. Será importante que o utilizador, aquando da navegação, nunca saia do caminho e volte a entrar no mesmo numa secção mais à frente, pois poderá não detetar um cruzamento e se isso acontecer toda a navegação fica comprometida, pois o dispositivo nunca irá detetar as coordenadas que estão a tentar encontrar.

No caso da aplicação desenvolvida foi implementado este método, usando uma metodologia de deteção de contextos. A aplicação sabe a todo momento onde se encontra o dispositivo móvel e deteta alterações do meio envolvente, analisando se o dispositivo móvel se está a aproximar de um cruzamento e adaptando o seu comportamento de acordo com isto. Assim que se “apercebe” que se está a aproximar de um cruzamento, a aplicação desenvolvida chama a atenção do utilizador através de um aviso sonoro, retira do ecrã toda a informação desnecessária para a tomada de decisão da

escolha do caminho seguir e mostra apenas o desenho do esquema da nota atual, para que o utilizador se concentre apenas na análise do esquema e diminua a possibilidade de se enganar e escolher o caminho errado. De seguida a aplicação passa a mostrar as informações e esquemas da nota do próximo cruzamento e continua-se a navegação para o próximo ponto.

### **3.9 Necessidade de equipamento extra para a navegação**

Nesta secção foi verificado quais as aplicações que necessitavam de equipamento extra para a navegação. Uma das aplicações que necessitava de um equipamento externo era a *Road Book Digital Totem*, que tem necessidade de usar o *Colosso EV* para medição das distâncias percorridas, tal como descrito na secção 2.5. Como foi referido anteriormente, isto é uma desvantagem para utilizadores casuais deste tipo de aplicações.

As aplicações *Mobileroadbook* na *TLC Camp'11* e *Tulip - Roadbook Editor for iPad* não fazem a leitura das distâncias percorridas e necessitam do apoio do conta-quilómetros do veículo para a medição das distâncias, ou então um dispositivo *GPS* extra. A necessidade de utilizar equipamento extra é uma desvantagem relativamente às aplicações que apenas utilizam o dispositivo móvel, pois o utilizador terá de lidar com dois equipamentos distintos ao mesmo tempo.

A aplicação desenvolvida não necessita de ser utilizada em conjunto com mais nenhum equipamento extra para navegação, na medida que esta é feita com recurso ao dispositivo móvel.

### **3.10 Detecção de erros de navegação e correção da rota**

Das aplicações analisadas nenhuma realiza a deteção de erros de navegação, nem corrigem a rota em caso de acontecerem estes erros. Esta é uma funcionalidade que se poderá ser considerada útil, especialmente se o tipo de navegação for automático, pois se acontecer o utilizador perder-se será difícil retomar a navegação corretamente.

O que se pretende que a aplicação faça é que detete se, num cruzamento, o utilizador seguiu pelo caminho correto ou se aconteceu um erro de navegação. No caso de ter acontecido um erro de navegação pretende-se que a aplicação ajude o utilizador a encontrar o caminho correto. Note-se que se estivéssemos a analisar uma aplicação cujo sistema de navegação fosse o próprio caminho marcado no mapa esta correção seria fácil, pois o utilizador apenas teria de analisar o mapa e navegar de forma a voltar a estar no caminho correto. Como se trata de uma navegação baseada em esquemas a correção torna-se mais difícil, no entanto esta é uma funcionalidade que adiciona valor a uma aplicação deste género, pois tal como já foi analisado, um erro de navegação pode arruinar completamente a navegação, fazendo com que o utilizador não consiga progredir mais no terreno sem ajuda do criador do *roadbook* ou sem voltar ao ponto inicial e reiniciar a navegação. Desta forma, considera-se que uma opção que permita uma ajuda automática na correção da navegação é importante.

### **3.11 Utilização em veículos motorizados**

No caso das aplicações analisadas todas podem ser usadas em veículos motorizados.

### **3.12 Utilização em passeios de bicicleta e percursos pedestres**

Uma das aplicações analisadas não pode ser usada em percursos pedestres ou de BTT, na medida que utiliza um dispositivo externo que deve estar conectado ao veículo de forma a efetuar a medição das distâncias. Por outro lado, existem algumas aplicações que podem ser usadas neste tipo de percursos, mas para tal deve ser usado um dispositivo *GPS* extra para realizar a leitura das distâncias percorridas. Devido a isto torna-se mais complicado realizar a navegação usando estas aplicações do que utilizando uma aplicação que não necessite equipamento extra para a navegação.

A aplicação desenvolvida pode ser utilizada em percurso de bicicleta e pedestres.

### 3.13 Resumo da análise

Na *Tabela 1* pode ver-se um resumo da comparação de cada uma das aplicações analisada no que respeita às suas características genéricas. Como se pode ver na referida tabela, a aplicação *Naturepathfinder* permite a criação de *roadbooks*, sendo que a marcação dos cruzamentos é feita utilizando a técnica de gravação de *pontos* diretamente no terreno. Não é necessário nenhum equipamento extra para a marcação do percurso do *roadbook*, pois a aplicação possui todas as funcionalidades necessárias para gravar as coordenadas *GPS* dos cruzamentos, assim como para a inserção de todas as informações necessárias. Aquando da gravação das notas o utilizador poderá utilizar esquemas predefinidos para representar os esquemas indicadores da direção a tomar, ou então poderá desenhar os seus próprios esquemas.

Também é possível realizar o *download* de *roadbooks* criados por outros utilizadores, a partir de um servidor central, assim como também é possível partilhar os *roadbooks* criados usando o mesmo serviço.

Durante a navegação é possível realizar uma navegação manual, permitindo ao utilizador passar para a nota seguinte ou recuar para a nota anterior (caso a exatidão do sinal *GPS* seja baixa), ou então realizar uma navegação automática, em que a própria aplicação deteta se o utilizador se encontra num cruzamento e passa automaticamente para a nota seguinte.

Na navegação é usado o sistema que utiliza as coordenadas dos *pontos* como forma de referência dos cruzamentos. Desta forma, independentemente da distância percorrida, é mostrada a informação da nota seguinte quando é detetado que o utilizador está próximo das coordenadas *GPS* do cruzamento para o qual se está a dirigir. Também na navegação a aplicação não necessita de nenhum dispositivo extra, pois a própria aplicação realiza todas as ações necessárias neste processo.

Uma outra funcionalidade que esta aplicação possui é a deteção de erros de navegação e a respetiva ajuda na correção para a rota correta. Assim sendo, sempre que acontecer um erro de navegação a aplicação irá avisar o utilizador e assisti-lo até que seja encontrada a rota correta.

Esta aplicação pode ser utilizada tanto em percursos a realizar em veículos motorizados, como em percursos de bicicleta ou pedestres, na medida que depende apenas do próprio dispositivo móvel para realizar a captura de notas, criação do percurso e respetiva navegação.

A aplicação *Naturepathfinder* foi disponibilizada gratuitamente.

Os detalhes das características referidas anteriormente serão abordados no capítulo 4.

Tabela 1 - Comparação entre criadores/leitores de roadbook para dispositivos mó-veis no que respeita às suas características

Características / Software	Aplicação Desenvolvida (Naturepathfinder)	Mobileroadbook na TLC Camp'11	GPS Road Book	Road Book Digital Totem	RoadBook JSC v4	Roadbook Xdappfactory	Tulip Roadbook Editor for iPad
Sistema Operativo	Android	Android	Android	Android	Android	iOS	iOS
Permite criação de roadbooks	Sim	Não	Não	Sim	Sim	Sim	Sim
Marcação dos pontos online (Mapa) na própria aplicação	Não	n.a.	n.a.	n.a.	Sim	Sim	n.a.
Marcação dos pontos com o sensor GPS do dispositivo móvel, diretamente no terreno	Sim	n.a.	n.a.	n.a.	Não	Sim	n.a.
Necessidade de equipamento extra para marcação das notas	Não	n.a.	Não	Sim	Não	Não	Sim
Permite usar esquemas das notas predefinidos	Sim	n.a.	n.a.	Sim	Não	Sim	Sim
Permite desenhar esquemas das notas	Sim	n.a.	n.a.	Sim	Sim	Não	Sim
Permite Download de roadbooks da web a partir de um serviço central	Sim	n.a.	Sim	Sim	Não	Sim	Sim
Permite partilhar os roadbooks na web num serviço central	Sim	n.a.	Não	Não	Sim	Não	Sim
Permite Navegação manual	Sim	Sim	Sim	Não	Não	Não	Não
Permite Navegação automática	Sim	Não	Não	Sim	Sim	Sim	Não
Navegação através de distâncias percorridas	Não	Sim	Sim	Sim	Não	Não	Sim
Navegação através das coordenadas dos pontos	Sim	Não	Não	Não	Sim	Sim	Não
Necessidade de equipamento extra para a navegação	Não	Sim	Não	Sim	Não	Não	Sim
Detecção automática de erros de navegação e correção automática da rota	Sim	Não	Não	Não	Não	Não	Não
Utilização em Veículos motorizados	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Utilização em passeios de bicicleta e percursos pedestres	Sim	Sim	Sim	Não	Sim	Sim	n.a.
Software gratuito	Sim	Sim	Sim	Não	Não	Não	Não

### 3.14 Estudo da importância que potenciais utilizadores atribuem às características analisadas

Foi feito um levantamento acerca da importância que potenciais utilizadores atribuem às características analisadas na secção 3.13 numa aplicação deste género.

Foi realizado um inquérito a 126 potenciais utilizadores da aplicação. Entre os entrevistados que realizaram o inquérito estavam organizadores de eventos ligados a atividades ao ar livre e outras pessoas que normalmente costumam criar e utilizar *roadbooks* tradicionais em papel. Na Figura 37 é possível ver-se o gráfico resultante do estudo efetuado.

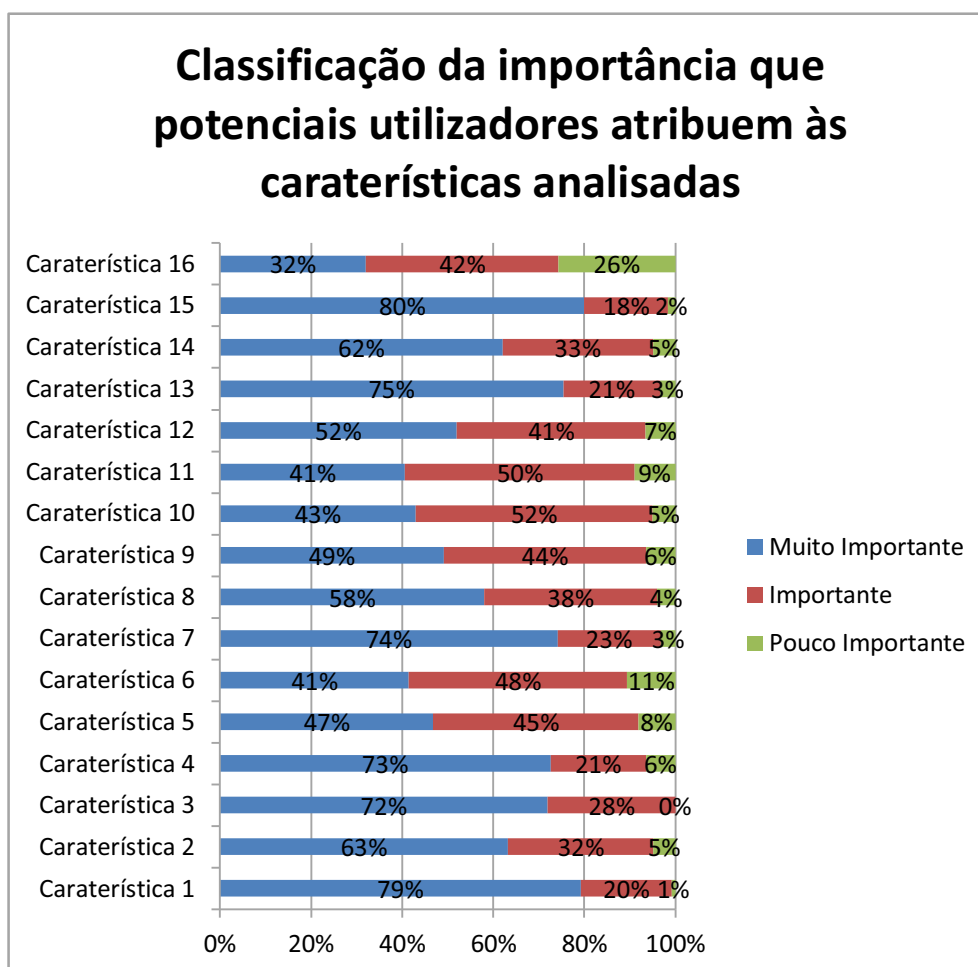


Figura 37 - Estudo da importância que potenciais utilizadores atribuem às características analisadas



A questão que foi colocada no inquérito foi: *Como considera a importância das características seguintes no que respeita a uma aplicação que permita criar/navegar roadbooks para dispositivos móveis?(Pouco Importante/Importante/Muito Importante)*

Analisa-se de seguida os resultados do inquérito tendo em conta cada uma das características:

**1. Permitir criar roadbooks;**

Relativamente a esta característica 79% dos inquiridos concordaram que é muito importante que uma aplicação para dispositivos móveis permita a criação de *roadbooks*, enquanto 20% afirmaram que é importante. Esta característica foi classificada de pouco importante por 1% dos inquiridos. Pode concluir-se que esta é sem dúvida uma característica que os potenciais utilizadores valorizam.

**2. Marcação dos pontos feita num mapa online, na própria aplicação;**

No caso deste tópico, 63% dos entrevistados afirmaram que é muito importante que uma aplicação deste género permita a marcação de pontos usando a internet, diretamente num mapa e 32% dos inquiridos concordou que é importante. Por outro lado 5% dos inquiridos afirmaram que é pouco importante esta funcionalidade. Esta é também uma funcionalidade bastante valorizada por potenciais utilizadores.

**3. Marcação dos pontos feita no terreno, com o sensor GPS do dispositivo móvel;**

Esta é uma funcionalidade a que os inquiridos dão valor, na medida que 72% disseram que é muito importante poder-se registar os dados diretamente no terreno, recorrendo ao *GPS* do dispositivo móvel e os restantes 28% afirmaram que é importante possuir esta funcionalidade. Esta é a única característica a que todos os inquiridos atribuíram uma classificação positiva, pelo que se pode considerar a característica mais valorizada pelos entrevistados.

**4. Não haver necessidade de equipamento extra para marcação das notas;**

Neste caso 73% dos inquiridos concordaram que é muito importante que não seja necessária a utilização de mais nenhum dispositivo para marcação das notas para além do dispositivo móvel onde está instalada a aplicação. Apenas 21% dos potenciais utilizadores afirmou que esta funcionalidade é importante. Por último, 6% dos inquiridos disse que é pouco importante que não haja necessidade de equipamento extra. Assim sendo é também uma característica valorizada pelos potenciais utilizadores de uma aplicação deste género.

#### **5. Permitir usar esquemas das notas predefinidos;**

Neste tópico 47% dos entrevistados concordaram que é muito importante que seja possibilitada a utilização de esquemas notas predefinidos e 45% afirma que é uma funcionalidade importante. Por outro lado 8% dos inquiridos não atribuiu importância à existência de esquemas de notas predefinidos. Apesar de a maioria dos entrevistados ter dado nota positiva a esta característica as opiniões dividem-se na classificação entre muito importante ou importante. No entanto, não deixa de ser uma funcionalidade a que os inquiridos dão valor. Apesar disso uma pequena percentagem dá pouco valor a esta funcionalidade.

#### **6. Permitir desenhar esquemas das notas;**

Segundo os resultados do inquérito, neste tópico 41% dos inquiridos esteve de acordo que é muito importante que seja permitido ao utilizador desenhar os seus próprios esquemas de notas e 48% dos inquiridos afirmaram que esta funcionalidade é importante. Apenas 11% disseram que é pouco importante que exista a possibilidade de se desenhar esquemas das notas. Também neste caso as opiniões dividem-se entre classificar esta característica de muito importante ou importante. Pode destacar-se que 11% classificaram esta característica de pouco importante, o que pode levar a pensar que alguns potenciais utilizadores não estarão interessados em ter de desenhar esquemas personalizados, mas contentam-se em poder utilizar esquemas predefinidos.

#### **7. Permitir fazer o *download* de *roadbooks* da *web* a partir de um serviço central;**

Neste tópico 74% dos entrevistados concordaram que é muito importante que seja possível fazer o *download* de *roadbooks* a partir da *web*, de um serviço central e 23% considerou que esta funcionalidade é importante. Por outro lado 3% dos inquiridos deu pouco valor a esta possibilidade, classificando-a de pouco importante. Esta é uma característica a que a generalidade dos inquiridos deu valor.

#### **8. Permitir a partilhar os *roadbooks* na *web* num serviço central;**

Neste tópico a maioria dos entrevistados (58%) concordou que é muito importante que seja possível a partilha dos *roadbooks* criados usando um serviço na Internet, enquanto 38% considerou que esta funcionalidade é importante. Esta característica foi classificada de pouco importante por 4% dos inquiridos. Pela análise dos resultados e comparando-os com os da característica 7, pode afirmar-se que os inquiridos dão mais valor à possibilidade de fazerem *download* de *roadbooks* da *web* do que à hipótese de poderem partilhar *roadbooks*, apesar de darem valor a esta funcionalidade.

**9. Permitir navegação manual;**

No caso deste tópico, 49% dos entrevistados disseram que é muito importante a existência desta característica, 44% dos entrevistados classificaram esta funcionalidade de importante, enquanto 6% concordaram que é pouco importante. Também aqui as opiniões estão divididas, apesar de ser uma funcionalidade valorizada, na medida que apenas uma minoria de 6% classificou esta funcionalidade de pouco importante.

**10. Permitir Navegação automática - controlada pela aplicação;**

Neste tópico 43% dos entrevistados concordaram que é muito importante que a aplicação permita uma navegação automática dos *roadbooks*, 52% afirmaram que é importante e 5% disseram que esta funcionalidade é pouco importante. Contrariamente ao que era espectável são mais os inquiridos a classificar esta funcionalidade de importante do os que a classificam de muito importante. No entanto é uma característica que é valorizada pela maioria dos inquiridos. Ainda assim há uma minoria que classificou esta característica de pouco importante.

**11. Navegação ser feita através de distâncias percorridas;**

Neste caso 41% dos entrevistados afirmaram que é muito importante que exista a possibilidade de se navegar utilizando as distâncias percorridas, enquanto 50% disseram que esta opção é importante. Neste caso 9% das pessoas afirmaram que esta característica é pouco importante. Nesta caso as opiniões estão divididas entre uma classificação de importante ou muito importante. Destaca-se os 9% de inquiridos que responderam que esta característica é pouco importante.

**12. Navegação ser feita através das coordenadas dos pontos;**

Nesta opção 52% dos entrevistados concordou que é muito importante que a navegação seja feita através das coordenadas dos pontos e 41% afirmou que é importante que assim aconteça. Esta característica foi classificada de pouco importante por 7% dos entrevistados. Os resultados deste tópico são bastante semelhantes aos da característica 11. No entanto, são mais os inquiridos que classificaram de uma forma positiva a característica que permite que a navegação seja feita através das coordenadas dos pontos.

**13. Não haver necessidade de equipamento extra para a navegação;**

Nesta característica 75% dos inquiridos concordou que é muito importante que não seja necessário mais nenhum dispositivo extra para se realizar a navegação de um *roadbook* e 18% afirmaram que esta característica é importante. Por outro lado 3% dos inquiridos afirmaram que esta caracte-

rística é pouco importante. Neste caso a maioria dos entrevistados deu bastante valor ao facto de não ser necessário equipamento extra para a realização da navegação.

#### **14. Detecção automática de erros de navegação e correção automática da rota;**

Neste caso 62% dos entrevistados afirmaram que é muito importante a existência de uma funcionalidade que permita detetar erros de navegação e permita fazer a correção automática da rota e 33% disseram que esta funcionalidade é importante. Por outro, lado 5% classificaram esta característica como sendo pouco importante. Contrariamente ao que poderia ser espectável, existe uma minoria que dá pouco valor a uma característica deste género.

#### **15. Utilização em Veículos motorizados;**

Neste tópico 80% dos entrevistados concordaram que é muito importante que uma aplicação que permita criar e navegar *roadbooks* possa ser utilizada em percursos recorrendo a veículos motorizados. Esta característica foi classificada de importante por 18% dos entrevistados e 2% afirmaram que esta funcionalidade é pouco importante. Neste caso a grande maioria valorizou bastante esta possibilidade.

#### **16. Utilização em passeios de bicicleta e percursos pedestres;**

Relativamente a esta característica 32% dos inquiridos concordaram que é muito importante que uma aplicação que permita criar e navegar *roadbooks* possa ser utilizada em percursos pedestres e de bicicleta, 52% afirmaram que é importante que isso aconteça e 26% disseram que esta possibilidade é pouco importante. Esta é a característica que dividiu mais a opinião dos inquiridos, apesar de ser uma característica valorizada. Mesmo assim 26% das pessoas não se importa com a possibilidade de poder utilizar uma aplicação deste género em passeios de bicicleta ou percursos pedestres, dando a mais importância à possibilidade de usar uma aplicação deste género em percursos com veículos motorizados.

Tendo em conta os resultados do inquérito realizado pode concluir-se que os inquiridos demonstraram bastante interesse na possibilidade da existência das características enumeradas na secção 3.13 numa aplicação deste género. Foi realizado ainda um estudo sobre a importância que potenciais utilizadores dão à existência de uma aplicação para dispositivos móveis com as características enunciadas anteriormente. A análise a este estudo é feita na secção seguinte.

### 3.15 Estudo de opinião sobre a importância que potenciais utilizadores dão à existência de uma aplicação para dispositivos móveis com as características enunciadas

Neste estudo foi realizado um inquérito aos 126 potenciais utilizadores que já tinham também respondido ao inquérito analisado na secção 3.14.

Neste caso foi colocada a seguinte pergunta: *Na sua opinião, considera importante a existência de uma aplicação para dispositivos móveis com as características enunciadas?*

Na Figura 38 é possível ver-se o gráfico resultante do estudo efetuado à resposta dada pelos inquiridos. Como se pode observar, 96% dos inquiridos considerou importante a existência de uma aplicação para dispositivos móveis com as características enunciadas anteriormente. No entanto 2% dos questionados não têm certeza, enquanto os outros 2% afirmam que não consideram isso importante.

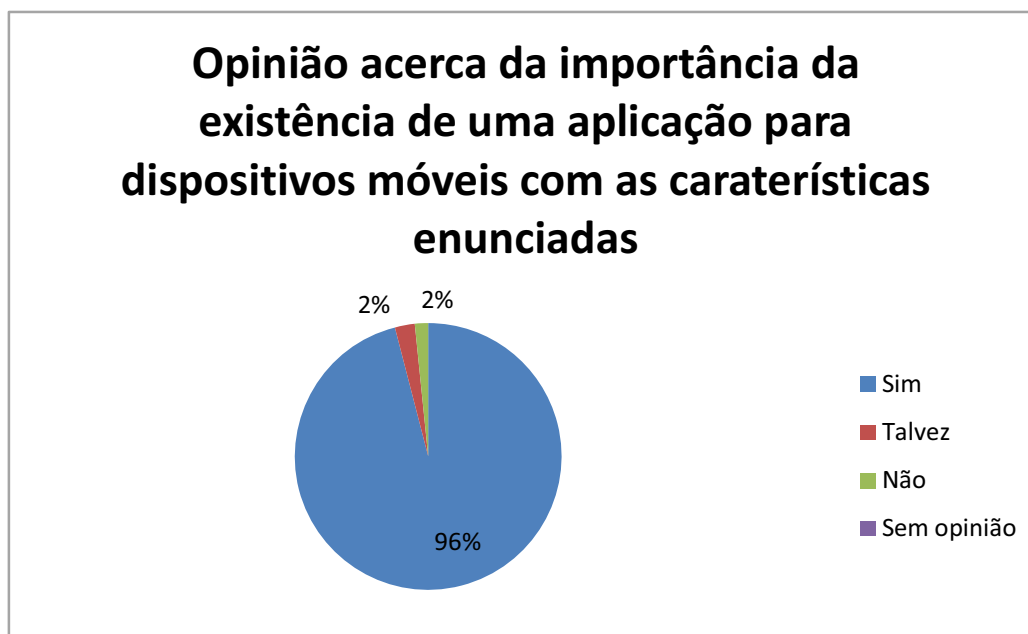


Figura 38 - Estudo efetuado no que respeita à opinião sobre a importância da existência de uma aplicação para dispositivos móveis com as características enunciadas

No questionário realizado dava-se a possibilidade aos inquiridos de justificar a resposta, caso não considerassem importante a existência de uma aplicação para dispositivos móveis com as características enunciadas. Um dos inquiridos não justificou a sua opção, enquanto o outro afirmou que continua a preferir utilizar os tradicionais *roadbooks* em papel.

## Capítulo 4

# 4. Implementação

Neste capítulo vai ser feita a descrição da implementação da aplicação desenvolvida, começando por se falar da arquitetura geral do sistema.

De seguida é feita a explicação de como foram implementadas as bases de dados, o editor gráfico para desenhar as notas, o módulo de gravação das notas, o módulo para navegação de *roadbooks* e o processo de correção da rota.

É também neste capítulo que se aprofunda o conceito de deteção de contextos e se refere as técnicas utilizadas para a criação da interface da aplicação *Naturepathfinder*.

## 4.1 Arquitetura do Sistema

Na Figura 39 pode consultar-se um diagrama geral da arquitetura do sistema desenvolvido, em que se podem ver diversos diagramas que representam os diferentes módulos que formam a aplicação cliente e a aplicação servidora.

Relativamente à aplicação cliente, que é referente à aplicação propriamente dita, podem destacar-se o módulo da gravação das notas de um *roadbook* e o módulo da navegação de um *roadbook*, pois são os processos base em que assentam todos os restantes comportamentos da aplicação *Naturepathfinder*. Toda a informação é armazenada numa base de dados no próprio dispositivo móvel.

A aplicação servidora é constituída por classes PHP que fazem a ponte entre a aplicação cliente e a base de dados no servidor, cujo objetivo é a partilha de roadbooks.

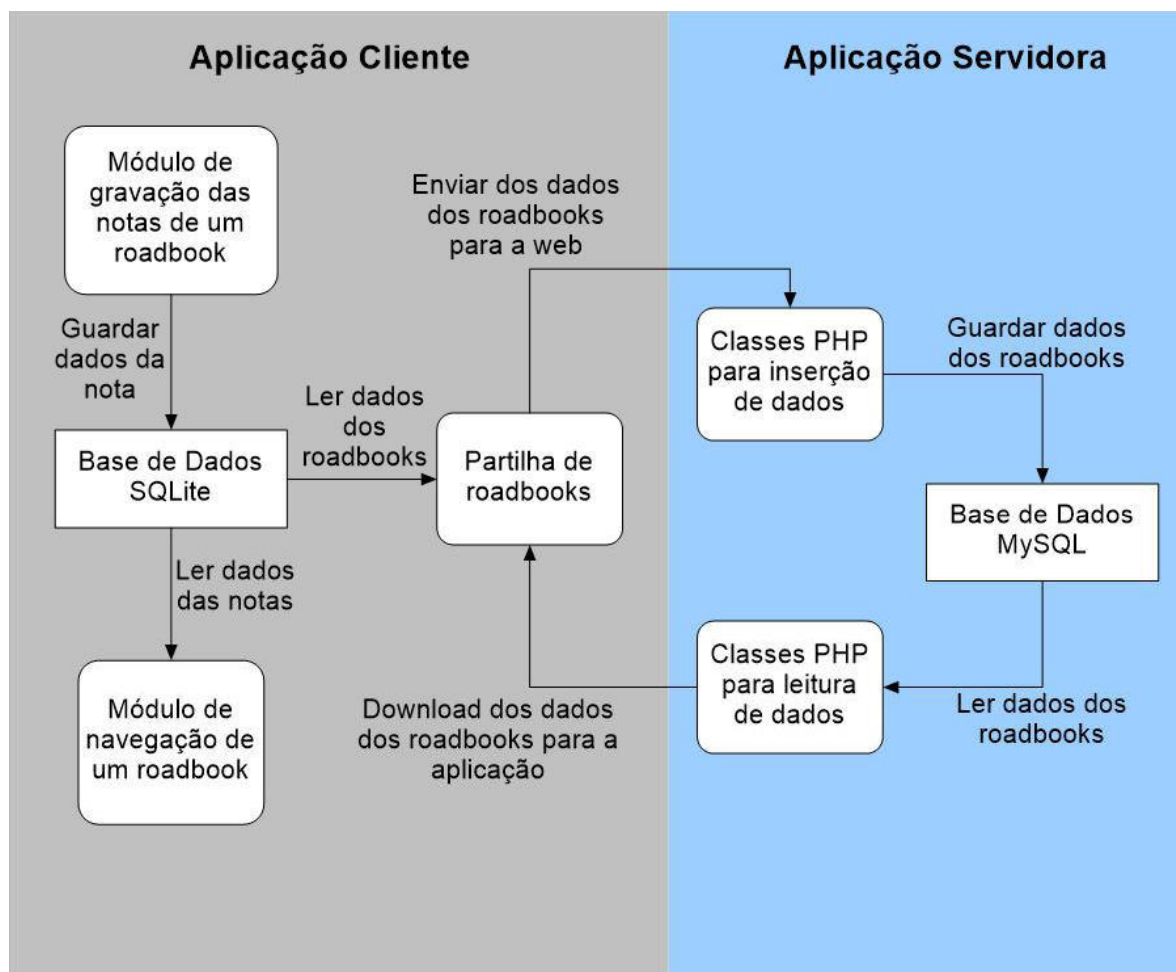


Figura 39 - Arquitetura geral do sistema

A função principal dos módulos da aplicação cliente são permitir ao utilizador criar *roadbooks* e navegar usando *roadbooks*. Do módulo de gravação das notas de um *roadbook* faz parte o editor gráfico que permite a criação dos desenhos dos esquemas das notas. Relativamente ao módulo da navegação de um *roadbook*, é complementado pelo processo de correção da rota, que será ativado quando houver um erro de navegação, respeitando uma metodologia de deteção de contextos. Ambos os módulos e respetivas ferramentas e processos assentam numa base de dados *SQLite*, onde ficam armazenados todos os dados registados aquando da criação de um *roadbook* e que serão usados durante da sua navegação.

No que respeita à aplicação servidora, um dos seus objetivos é permitir a partilha de *roadbooks* pelos utilizadores da aplicação. Na sua implementação foram criadas diversas classes *PHP*



que têm como função fazer a ligação entre os dados que são enviados da aplicação para o servidor, inserindo os dados dos *roadbooks* na base de dados *MySQL*. O outro objetivo desta aplicação servidora é permitir ao utilizador fazer o *download* de *roadbooks* existentes na base de dados *MySQL* para a base de dados da aplicação Cliente (*SQLite*), de forma a poder utiliza-los na navegação. Esta funcionalidade está também garantida por classes *PHP* existentes no servidor.

Nas Secções seguintes será feita a descrição de cada um destes componentes que fazem parte da estrutura geral da aplicação desenvolvida. Os módulos foram implementados com base nas Heurísticas de Nielsen, que podem ser consultadas no Anexo D, como forma de garantir que a aplicação tenha um bom nível de usabilidade.

## 4.2 Bases de dados

Foi criada uma base de dados relacional *SQLite* para armazenar todos os dados no dispositivo, na medida que é a base de dados suportada pelo sistema *Android* e contém os recursos essenciais para o desenvolvimento de uma base de dados para a aplicação em questão, nomeadamente criação de múltiplas tabelas e relacionamentos entre elas com recuso a chaves primárias e chaves externas. Trata-se de um sistema de gestão de bases de dados bastante simples, que não necessita de ser instalado antes de ser usado, nem necessita de um servidor para ser executado, como se pode consultar no *web* site oficial<sup>36</sup>.

Foi também criada uma base de dados *MySQL* num domínio *Web*, com o objetivo da partilha dos *roadbooks* criados com todos os utilizadores da aplicação. Optou-se por um sistema de bases dados *MySQL* pois trata-se de um sistema fácil de instalar e administrar, com uma interface bastante acessível. Como se pode consultar no *web* site oficial<sup>37</sup> atualmente trata-se de um dos sistemas de gestão de bases de dados mais rápidos e seguros, características fundamentais para uma utilização online. Acrescenta-se a isto o facto de se tratar de *software* livre.

---

<sup>36</sup> <https://www.sqlite.org/different.html>

<sup>37</sup> <https://www.mysql.com/why-mysql/>

### 4.2.1 Base de dados *SQLite* – da aplicação

Como se pode ver na Figura 40 foram criadas 5 tabelas na base de dados *SQLite* que constituem a base de dados da própria aplicação.

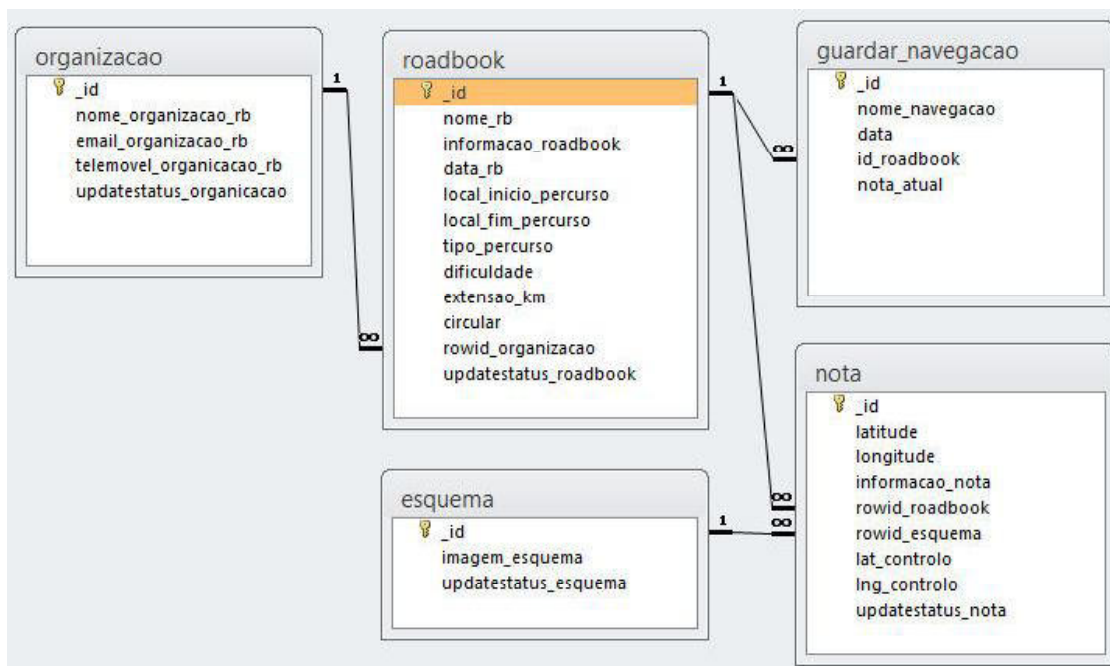


Figura 40 - Tabelas da base de dados *SQLite* no dispositivo móvel

A tabela “organizacao” tem como objetivo armazenar os dados do registo de uma organização, ou pessoa individual que irá criar *roadbooks*. Para além da chave primária (*\_id*), foram criados os campos “nome\_organizacao”, “email\_organizacao” e “telemovel\_organizacao” que irão ser usados aquando da necessidade que poderá existir de o utilizador contactar a organização do evento para esclarecimento de dúvidas. Quanto ao campo “updatestatus\_organizacao” será usado para armazenar informação relativa à sincronização dos dados entre a base de dados *SQLite*, no dispositivo móvel, e a base de dados criada online, permitindo que sejam identificados os dados que ainda não foram enviados para a *web*. Ao serem criados os registos da tabela “organizacao” o campo “updatestatus\_organizacao” será preenchido automaticamente com o valor “no”, de forma a se saber que este registo ainda não foi enviado para a base de dados online. Quando este registo for enviado para

a base de dados no servidor *web*, o valor do campo “updatestatus organizacao” passa a ter o valor “yes”.

No que respeita à tabela “roadbook” existem os campos que têm como objetivo armazenar a informação de cada *roadbook* criado, nomeadamente “nome\_rb”, “informacao\_roadbook”, “data\_roadbook”, “local\_inicio\_percurso”, “local\_fim\_percurso”, “tipo\_percurso”, “dificuldade”, “extensao\_km” e “circular”. Nesta tabela existe também a chave externa “rowid\_organizacao” que servirá para garantir da ligação com os dados da tabela “organização” associando um ou vários *roadbooks* a uma organização ou pessoa individual. O campo “updatestatus\_roadbook” irá ser usado para controlar se os dados desta tabela foram enviados para a base de dados na *web* para partilha de *roadbooks* com outros utilizadores da aplicação, utilizando a mesma técnica referida para a tabela “organizacao”.

Quanto à tabela “nota” esta tem como função armazenar os dados relativos a todas as notas guardadas para cada *roadbook* diretamente no terreno, usando os campos “latitude”, “longitude”, “informação nota”, “lat controlo” e “lng controlo”. No que respeita ao campo “updatestatus nota” irá ser usado para controlar se os dados desta tabela foram enviados para a base de dados na *web* para partilha de *roadbooks* com outros utilizadores da aplicação, utilizando a mesma técnica referida para a tabela “organizacao”. Existe também o campo chave externa “rowid\_roadbook” para ligação com a tabela *roadbook*, na medida que cada *roadbook* corretamente construído e finalizado terá associadas várias notas. Outra chave externa que existe é a “rowid\_esquema” que irá assegurar a ligação com a tabela “esquema”, para que a cada nota guardada possa estar associado um esquema. O mesmo esquema poderá estar associado a várias notas, na medida que os esquemas serão os desenhos dos cruzamentos e das direções a tomar, representando apenas as características principais do que se pretende mostrar. Sendo assim irão existir muitos cruzamentos que irão partilhar estas características e que poderão ter os mesmos esquemas representativos.

Como foi referido, a tabela “esquema” irá armazenar as imagens referentes aos esquemas criados pelo utilizador para cada cruzamento, no campo “imagem\_esquema”. Mais uma vez o campo “updatestatus\_esquema” irá armazenar informação para permitir à aplicação controlar se os dados estão sincronizados com a base de dados *online*.

Sempre que um utilizador estiver a percorrer um *roadbook* terá a possibilidade de interromper a navegação a qualquer momento e guardar a informação referente à nota onde se encontra para mais tarde retomar a navegação. Esta informação será guardada na tabela “guardar\_navegacao”.

usando os campos “nome\_navegacao” (que será introduzido pelo utilizador), “data”, “nota\_atual” e a chave externa “id\_roadbook”, que permitirá a identificação do *roadbook* que estava a ser usado, fazendo ligação com a tabela “roadbook”.

#### 4.2.1.1 Tipos de dados

Nas seguintes tabelas podem ser consultados os tipos de dados dos campos de todas as tabelas criadas.

*Tabela 2 - Tipos de dados dos campos da tabela “organizacao”*

<b>Tabela: organizacao</b>	
Campo	Tipo de dados
_id	integer
nome_organizacao_rb	text
email_organizacao_rb	text
telemovel_organizacao_rb	integer
updatestatus_organizacao	text

*Tabela 3 - Tipos de dados dos campos da tabela “roadbook”*

<b>Tabela: roadbook</b>	
Campo	Tipo de dados
_id	integer
nome_rb	text
informacao_roadbook	text
data_rb	date
local_inicio_percurso	text
local_fim_percurso	text
tipo_percurso	text
dificuldade	text
extensao_km	float
circular	text
rowid_organizacao	integer
updatestatus_roadbook	text

Tabela 4 - Tipos de dados dos campos da tabela "nota"

<b>Tabela: nota</b>	
Campo	Tipo de dados
_id	integer
latitude	float
longitude	float
informacao_nota	text
rowid_roadbook	integer
rowid_esquema	integer
lat_controlo	float
lng_controlo	float
updatestatus_nota	text

Tabela 5 - Tipos de dados dos campos da tabela "esquema"

<b>Tabela: esquema</b>	
Campo	Tipo de dados
_id	integer
imagem_esquema	blob
updatestatus_esquema	text

Tabela 6 - Tipos de dados dos campos da tabela "navegacao"

<b>Tabela: guardar_navegacao</b>	
Campo	Tipo de dados
_id	integer
nome_navegacao	text
data	date
id_roadbook	integer
nota_atual	integer

Na aplicação desenvolvida foi criada uma classe java com todos os métodos necessários para criar a base de dados e as respetivas tabelas e campos, assim como inserção, atualização, consulta e eliminação de dados específicos (SQLiteManageBD.class). Esta classe vai ser acedida a partir de qualquer uma das outras classe sempre que seja necessário efetuar alguma destas ações na base de dados.

## 4.2.2 Base de dados *MySQL* – servidor *web*

Relativamente à base de dados *MySQL*, que foi criada num servidor *web*, foram criadas as mesmas tabelas que já tinham sido criadas na base de dados *SQLite*, exceto a tabela “guardar\_navegacao”, como se pode ver na Figura 41. Optou-se por não se criar esta tabela, nem permitir o envio destes dados para a *web* para partilha, na medida que na tabela “guardar\_navegação” vão ficar guardados os dados da posição em que o utilizador estava quando interrompeu a navegação, para que a possa retomar mais tarde. Esta informação terá pouco valor numa partilha, na medida que um utilizador pretende apenas realizar o percurso completo de um *roadbook* ou, depois de abandonar uma navegação a meio, continuar a sua própria navegação.

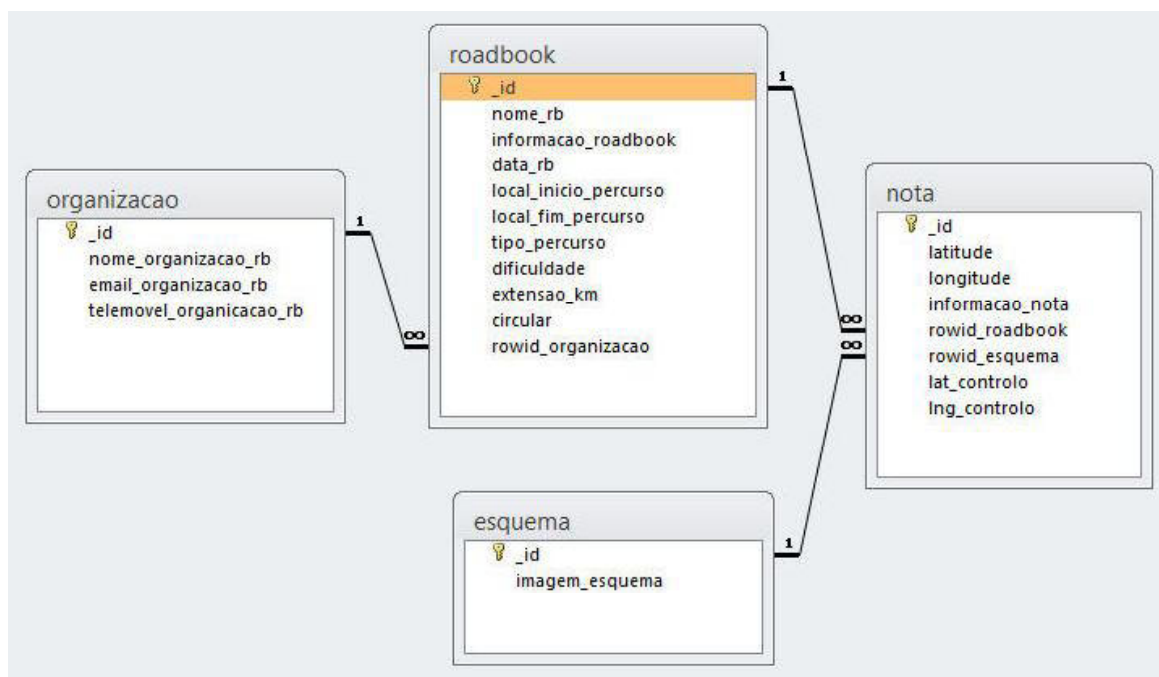


Figura 41 - Tabelas da base de dados *MySQL* na *web*

Todos os tipos de dados dos campos que foram criados nesta base de dados são os mesmos dos campos das tabelas da base de dados *SQLite* e podem ser consultados na secção 4.2.1.1. Utilizou-se a linguagem de Script *PHP* para gerir a base de dados a partir do servidor remoto.

Para os testes *offline* foi utilizado o XAMPP<sup>38</sup>, um ambiente de desenvolvimento que permite a utilização de um servidor *web* Apache com interpretadores *PHP* e base de dados *MySQL*. Trata-se de um *software* fácil de instalar, gratuito, e com suporte para vários sistemas operativos.

Como se pode ver na Figura 42, depois de instalado pode aceder-se ao painel de controlo do XAMPP, que é bastante intuitivo, permitindo neste caso iniciar-se o servidor *web* Apache e o sistema de gestão de bases de dados *MySQL* para que a aplicação possa aceder à base de dados utilizando a linguagem *Apache*.

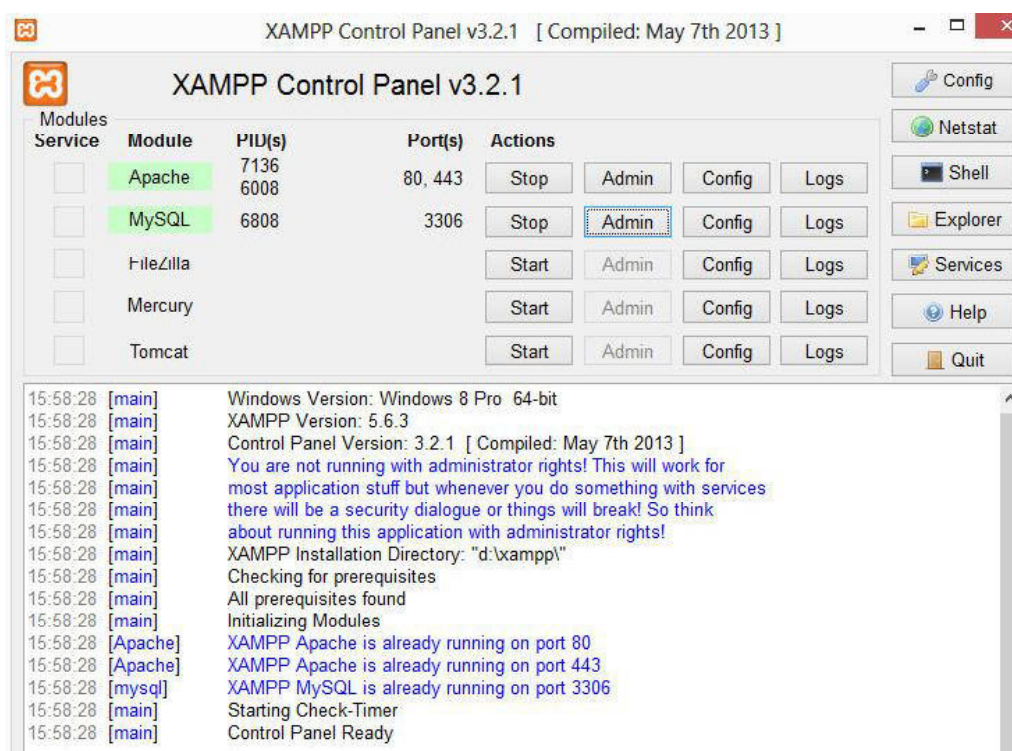


Figura 42 - painel de controlo do XAMPP

Pode também iniciar-se a página de administração *MySQL*, que irá permitir criar e administrar a base de dados. Pode ver-se na Figura 43 o exemplo da criação de uma das tabelas (“*roadbook*”) usando *SQL*.

<sup>38</sup> [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)

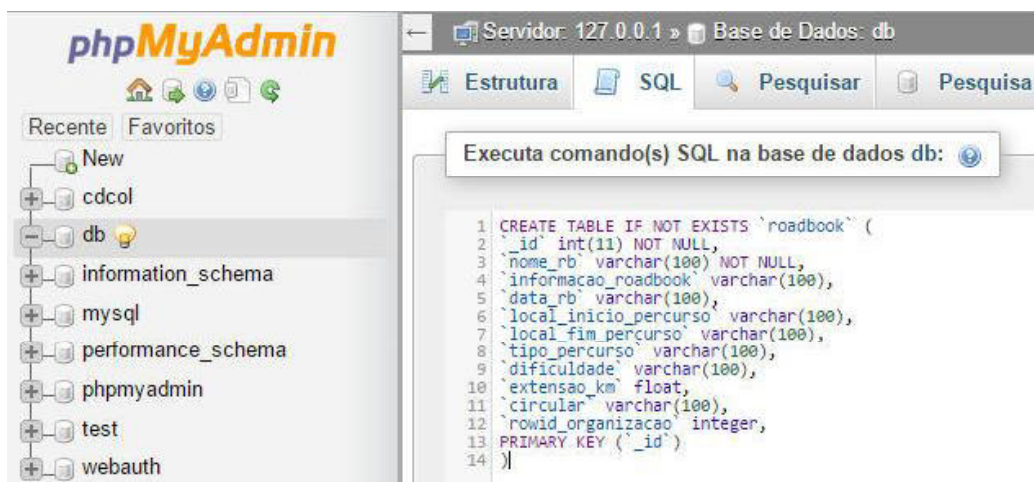


Figura 43 - Criar tabela “roadbook” no menu de administração MySQL do XAMPP

Tal como no exemplo anterior todas as outras tabelas da base de dados serão criadas a partir deste menu. Será também possível eliminar ou alterar campos e registos das tabelas utilizando esta interface.

Depois de efetuados todos os testes *offline* utilizando o XAMPP foi registado um alojamento *web* (Figura 44) e utilizando um sistema de gestão de bases de dados MySQL, com o PHP como meio de acesso e controlo da informação presente na base de dados.



Figura 44 - Menu do alojamento web registado



O menu de administração do *MySQL*, no domínio registado, é bastante semelhante ao do *XAMPP*, pelo que as operações para criação das tabelas da base de dados e respetiva gestão são iguais. Pode ver-se o exemplo da criação da tabela “nota” na Figura 45, utilizando *SQL*.



Figura 45 - Criar tabela “nota” no menu de administração *MySQL* na web

Depois de criadas as tabelas da base de dados, quer seja *online* ou utilizando o servidor *XAMPP*, o acesso e controlo dos dados presentes nas tabelas será feito através de scripts *PHP*, cujos ficheiros deverão ser enviados para o alojamento *web* utilizando um software de *FTP* (*File Transfer Protocol*). Neste caso específico o *software* utilizado foi o Filezilla<sup>39</sup>, que terá de aceder ao alojamento *web* e permitir o *upload* dos ficheiros *PHP*.

Na secção 4.8 irá ser explicado o processo que permite o acesso a este servidor a partir do dispositivo móvel, assim como a troca de informação entre a base de dados *SQLite* no dispositivo móvel e a base de dados *MySQL* no servidor *web*.

<sup>39</sup> <https://filezilla-project.org/>

## 4.3 Editor gráfico para desenhar esquemas das notas

Cada nota gravada deverá ter uma imagem representativa do cruzamento em questão, assim como a indicação da direção a tomar. Na base de dados irão existir já alguns esquemas desenhados, no entanto o utilizador poderá desenhar os seus próprios esquemas. Foi desenvolvido o menu que se pode ver na Figura 46, em que o utilizador poderá escolher o tamanho e a cor da ferramenta de desenho e realizar o esquema pretendido. Tem também a opção de escolher uma ferramenta que permite apagar partes do desenho. Finalmente poderá gravar o desenho, que ficará armazenado na base de dados e passar à criação de um novo esquema.

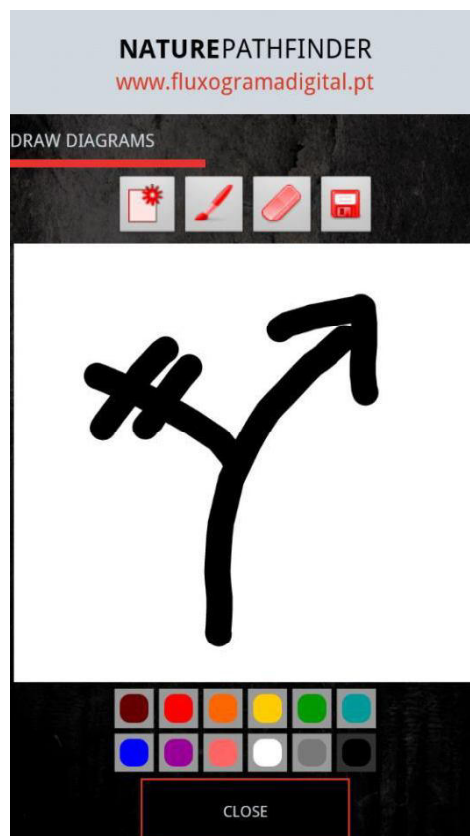


Figura 46 - Layout do editor gráfico

O *Android* permite a criação de desenhos diretamente numa componente tipo *View* a partir do *layout* definido<sup>40</sup>. Esta é a maneira mais eficaz de criar desenhos simples e estáticos. A outra opção seria desenhar diretamente no *Canvas*, o que iria permitir um controlo maior, na eventualidade da necessidade de criação de animações ou no caso do desenvolvimento de um videojogo em que fosse necessário redesenhar a imagem constantemente. Como no caso da aplicação desenvolvida apenas se pretende desenhar e guardar uma imagem representativa de uma nota foi utilizada a técnica de desenhar diretamente numa componente tipo *View*.

Foi definida uma componente *View* personalizada, como sendo a área onde se pode efetuar o desenho. Para tal foi criada uma nova classe java (*DesenharNotas\_DrawingView*), tendo como superclasse *android.view.View*, de forma a permitir controlar as ações numa componente tipo *View*. Esta nova classe java vai ser a responsável por detetar o toque no ecrã e controlar onde serão desenhadas as linhas. A partir desta nova classe foi então criada, no layout, a componente *View* personalizada e foi adicionado ao *layout* a componente *View* personalizada como se se tratasse de uma das componentes predefinidas, com as propriedades de uma componente *View* usual, apenas com a diferença que existe uma referência à classe java *DesenharNotas\_DrawingView* e consequentemente aos métodos que lá forem definidos para controlo dos desenhos. Esta será a área onde se poderão realizar os desenhos dos esquemas.

Foram definidos os componentes para os botões que irão permitir criar um novo desenho, escolher o tipo de pincel, escolher a borracha e gravar o desenho, assim como os botões da seleção das cores, como se pode ver na Figura 46.

O passo seguinte foi a criação da interação, permitindo à aplicação detetar o toque no ecrã para a criação das linhas pretendidas. Isto será feito na classe *DesenharNotas\_DrawingView*.

Foi utilizada a classe *Path* e para se seguir o toque do utilizador no ecrã e consequentemente criar o caminho percorrido com a função *drawPaint*.

De seguida o desenho vai ser criado usando a função *canvasPaint*, tendo em conta o caminho definido pela função *drawPaint*.

Foi utilizada o método *onTouchEvent*, com o parâmetro *MotionEvent* para se verificar se houve algum tipo de toque sobre a componente *View* que tinha sido definida anteriormente.

---

<sup>40</sup> <http://developer.android.com/guide/topics/graphics/2d-graphics.html>

Neste caso foi analisado se o utilizador tocou no ecrã, se deixou de tocar no ecrã, ou se realizou um arrastamento sobre o ecrã. Em cada um destes casos será programado um comportamento. Se a ação for clicar no ecrã irá ser definido início da linha nessas coordenadas. Se o utilizador arrastar o dedo na zona da *View* irá ser definido o caminho nessa zona. Ao ser levantado o dedo do ecrã será desenhada a linha no caminho que foi definido previamente e será reiniciada a configuração para que se inicie uma nova linha assim que o utilizado volte a clicar no ecrã, na zona da *View*.

Depois de estar desenhado o esquema, este pode ser guardado na base de dados *SQLite* para que posteriormente possa ser associado a uma nota. O tipo de dados do campo da base de dados que vai armazenar a imagem será o *blob*<sup>41</sup>. Um campo definido como *blob* pode armazenar imagens ou ficheiros multimédia. Este tipo de dados armazena a informação em formato binário em *arrays*.

Em primeiro lugar o desenho efetuado irá ser guardado em *cache* e guardado como bitmap. De seguida vai ser criado o objeto *byte array* e a imagem será comprimida em formato *PNG*. Finalmente esta imagem vai ser enviada para a base de dados *SQLite*.

## 4.4 Módulo de gravação das notas de um *roadbook*

Nesta secção vai ser descrito o processo de gravação das notas em cada cruzamento para um *roadbook* específico que o utilizador já tenha criado. Inicialmente o utilizador deverá criar um *roadbook* e adicionar todas as informações necessárias à sua identificação. Depois de concluir este processo passa-se à fase de adicionar as notas do *roadbook*, que irão identificar cada um dos cruzamentos que fazem parte do percurso, registando a localização geográfica de cada ponto.

---

<sup>41</sup> <http://developer.android.com/reference/java/sql/Blob.html>

## 4.4.1 Tecnologia utilizada para obtenção da localização

Foi feita uma análise das diversas tecnologias disponíveis nos dispositivos *Android* que poderiam ser utilizadas para a obtenção da localização. As tecnologias analisadas foram sensor *GPS* do dispositivo, Wi-Fi e Torres de Células.

### 4.4.1.1 *GPS - Global Positioning System*

O *GPS* é um dos sistemas utilizado em dispositivos móveis para obtenção da posição geográfica. Esta tecnologia é baseada numa rede de 24 satélites geostacionários que se encontram na órbita terrestre. Para obtenção correta do sinal de *GPS* o dispositivo deve obter sinal de pelo menos 3 satélites, de forma a calcular corretamente a latitude e a longitude da posição do dispositivo. Se obtiver sinal de mais do que 3 satélites será possível ao dispositivo calcular também a altitude. Esta localização é calculada através da medição das distâncias entre o dispositivo e os satélites, tendo em conta a sua posição em órbita. Desta forma é criada uma triangulação que permite calcular as coordenadas do dispositivo recetor de sinal *GPS* (Figura 47).

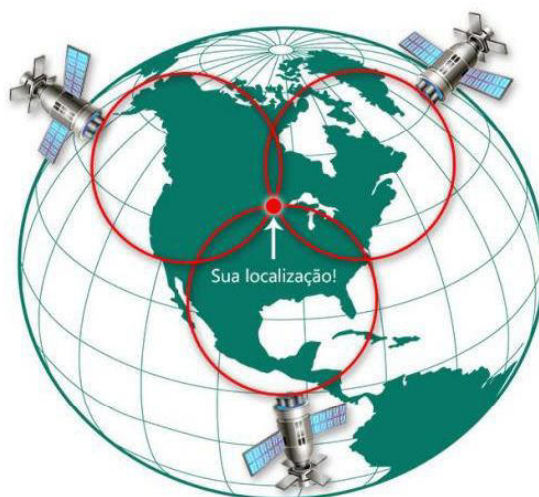


Figura 47 - Triangulação de sinal GPS<sup>42</sup>

---

<sup>42</sup> Retirado de: <http://www.oficinadanet.com.br/post/12406-como-funciona-o-gps>

O Sinal de *GPS* enviado pelos satélites é constituído por sinal rádio de baixa frequência na banda UHF que atravessam (com relativa dificuldade dependendo da espessura do material) materiais como vidro ou plástico, mas não atravessam objetos mais sólidos, como paredes, edifícios, montanhas, ou grandes densidades de vegetação. Existem outros fatores que influenciam a exatidão do sinal de *GPS*, tais como a geometria dos satélites, o *multitrajeto* e efeitos atmosféricos (Morgado, 2009).

A Geometria dos satélites tem a ver com o posicionamento espacial dos satélites em relação uns aos outros. O dispositivo recetor de *GPS* pode captar sinal de 4 satélites que se encontrem muito próximos, o que irá dificultar o cálculo da triangulação, pois os 4 sinais *GPS* podem ser interpretados apenas como um sinal *GPS*.

Quanto ao *multitrajeto* a interferência resulta da reflexão do sinal em objetos que se encontrem junto ao dispositivo recetor de sinal, como por exemplo edifícios, montanhas ou vegetação. O sinal acaba por chegar ao recetor, mas como demora mais tempo a atingi-lo isto pode levar a erros nos cálculos da triangulação (Sousa, 2005).

Quanto aos efeitos atmosféricos, o sinal *GPS* é afetado pela atmosfera terrestre, pois o sinal ao atravessar a ionosfera, que é composta por partículas elétricas, vai sofrer um atraso provocado pela diminuição da velocidade de propagação do sinal, dando origem também a erros nas medições das distâncias. Também quando o sinal atravessa a troposfera sofre um atraso, pois nesta zona existe uma maior concentração de nuvens, que também afetam a propagação do sinal.

A acumulação destes erros irá provocar erros na localização das coordenadas latitude e longitude lidas. No entanto, se analisarmos o estudo realizado pela ArcGIS Resources<sup>43</sup> podemos concluir que esse erro não será muito significativo, se forem tidos em consideração alguns detalhes. Segundo este estudo, independentemente do fabricante do sensor *GPS* do dispositivo móvel, a exatidão do mesmo é bastante semelhante. Segundo este estudo, num dia sem nuvens muito intensas, em locais sem muitas árvores ou edifícios altos que bloqueassem o sinal *GPS*, aproximadamente 90% das posições registadas com diversos dispositivos móveis teriam um desvio de apenas 3 metros relativamente à localização correta. Os mesmos dispositivos móveis foram conectados a antenas *GPS* externas e nas mesmas condições foi verificado que em 99% das localizações registadas teriam um erro de 3 metros e aproximadamente 70% das localizações teriam um desvio de apenas 1 metro.

---

<sup>43</sup> <http://blogs.esri.com/esri/arcgis/2013/07/15/smartphones-tablets-and-GPS-accuracy/>

Em condições mais adversas, como em localizações com muitos edifícios, ou em dias de condições atmosféricas com muitas nuvens, os resultados não foram tão bons, mas o erro obtido não foi considerado significativo. Em situações mais desfavoráveis e utilizando um máximo de 4 satélites a exatidão andará a rondar os 15 metros.

Outra desvantagem da utilização do *GPS* do dispositivo móvel é o grande consumo de energia da bateria, que é mais elevado do que quando se utiliza apenas Wi-Fi, a antena Rádio do *smartphone* ou o *Bluetooth* para obter a localização atual (Silva R. J., 2013).

#### 4.4.1.2 Torres de Células

A tecnologia utilizada pelas torres de células, tais como *GSM* ou *4G* permite a localização do dispositivo móvel. Nos *smartphones* atuais, esta tecnologia é mais usada com a técnica da análise de cenários, usando um registo de posições conhecidas das *Cell-ID* ou da *Cell Tower-ID*, de forma semelhante ao WPS (Silva R. J., 2013). No entanto, a exatidão da localização é baixa, podendo haver erros de 1000 metros. Para além disso existe a necessidade do registo de todas as torres de células e as respetivas localizações, sendo que o facto de serem propriedade dos operadores de telecomunicações poderá ser um entrave a este registo.

Relativamente à cobertura de sinal, esta técnica poderá ter vantagens relativamente ao *GPS*, especialmente em zonas urbanas, mas a maior vantagem prende-se mesmo com o facto de com esta técnica haver menos consumo de energia da bateria do dispositivo.

#### 4.4.1.3 Análise do estudo efetuado

Tabela 7 - Análise das tecnologias para obtenção da localização

Tecnologia	Exatidão do sinal	Segurança	Local de utilização ideal	Consumo de bateria
<i>GPS</i>	Aproximadamente entre 3 e 15 metros	Alta	Zonas com menos concentração de edifícios altos	Alto
Torres de Células	Aproximadamente 1000 metros	Alta	Zonas Urbanas	Baixo

Após a análise das vantagens e desvantagens das diversas tecnologias que poderiam ser utilizadas para a obtenção da localização do dispositivo móvel (Tabela 7) optou-se por utilizar a tecno-

logia *GPS*, na medida que, no âmbito da utilização da aplicação desenvolvida, será a que mais vantagens apresenta, pois tratando-se de uma aplicação para criar e ler *roadbooks*, a utilização maioritária será em percursos ao ar-livre. Das tecnologias estudadas foi a tecnologia que mostrou possuir uma exatidão de sinal mais elevada e maior segurança no que respeita à preservação da privacidade dos utilizadores.

Relativamente ao consumo de bateria, esta é a tecnologia que apresenta maior desvantagem, no entanto se se tiver em consideração as vantagens, esta desvantagem não é significativa. Durante o desenvolvimento da aplicação *Naturepathfinder*, foram implementados alguns algoritmos para diminuir o consumo de energia, que são abordados na secção 4.5.1. A adicionar a isto, existe a possibilidade de o dispositivo móvel ser carregado durante o percurso, caso se esteja a usar um veículo para criar/navegar um *roadbook*.

Pode concluir-se também que as condições ideais para utilização do sensor *GPS* do dispositivo móvel para leitura das coordenadas latitude e longitude serão sempre ao ar livre, de preferência longe de edifícios altos ou zonas de árvores grandes e muito densas. Quanto às condições atmosféricas seria preferível que não houvesse nuvens densas. Obviamente que nem sempre se poderão verificar as condições ideais, pelo que a aplicação desenvolvida usa algumas técnicas para melhorar a gravação das notas e a posterior leitura das mesmas aquando da navegação, evitando ao máximo que o utilizador se perca. Estas técnicas vão ser abordadas nas próximas secções.

Nos testes efetuados durante o desenvolvimento da aplicação atual, mesmo em condições mais adversas tendo em conta o que foi explicado anteriormente neste capítulo, a exatidão do sinal *GPS* variou sempre entre os 3 e os 12 metros, utilizando os dispositivos móveis identificados na Tabela 14, sendo que as leituras com menor erro foram obtidas em condições atmosféricas perfeitas e as leituras com maior erro foram obtidas em dias com nuvens intensas. Não foi valorizado o erro respeitante a localizações em zonas de vegetação intensa, pois nunca se revelou significativo tendo em conta as necessidades da aplicação desenvolvida.

#### **4.4.2 Registo de notas para criação de *roadbooks***

A Figura 48 apresenta o fluxograma do algoritmo implementado para a gravação das coordenadas dos pontos, recorrendo-se ao sensor de *GPS* do dispositivo.



Após as configurações iniciais e depois de o *GPS* do dispositivo estar ativo, começou-se o processo de obtenção de sinal de *GPS*. Enquanto não seja detetado sinal de *GPS* estará sempre visível uma mensagem que indica ao utilizador que ainda não há sinal.

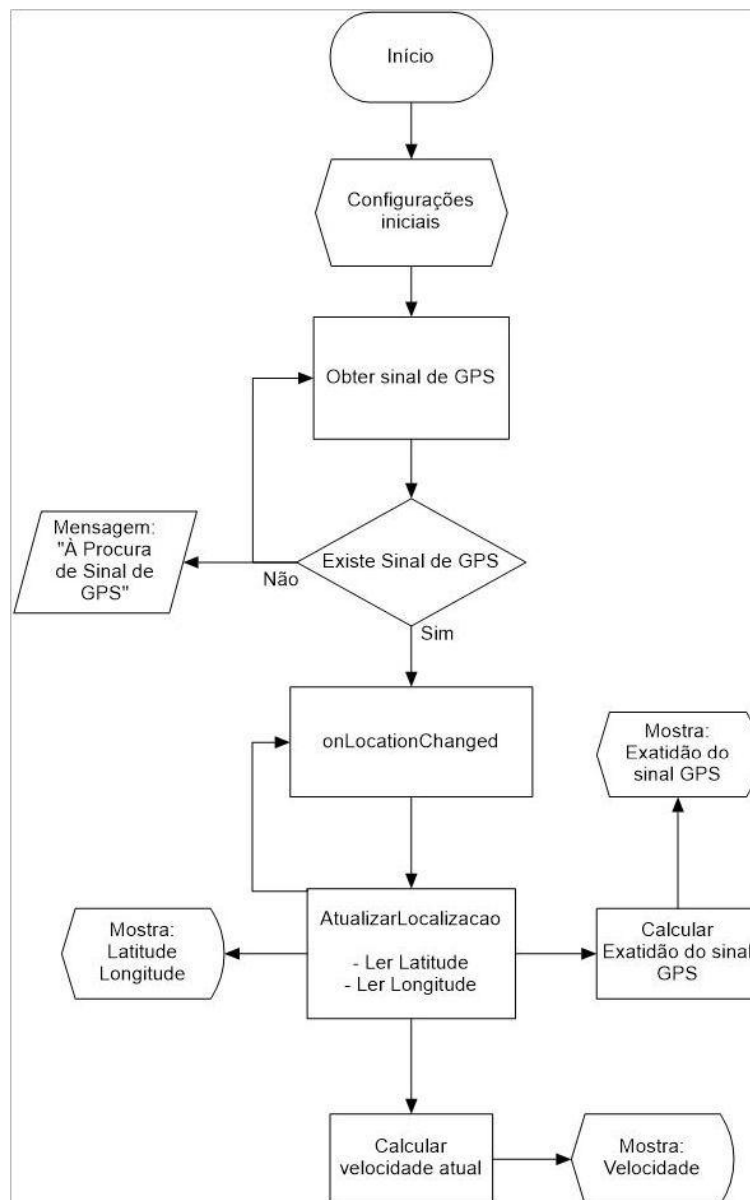


Figura 48 - Fluxograma dos processos de obtenção das coordenadas, exatidão e velocidade do dispositivo

A partir do momento em que é detetado sinal *GPS*, irão ser mostradas as coordenadas da latitude e da longitude, e será calculado o valor da exatidão e da velocidade atual, valores estes que também serão visíveis no *layout* atual (Figura 49). Sempre que seja detetada uma alteração na posição estes valores vão ser atualizados.

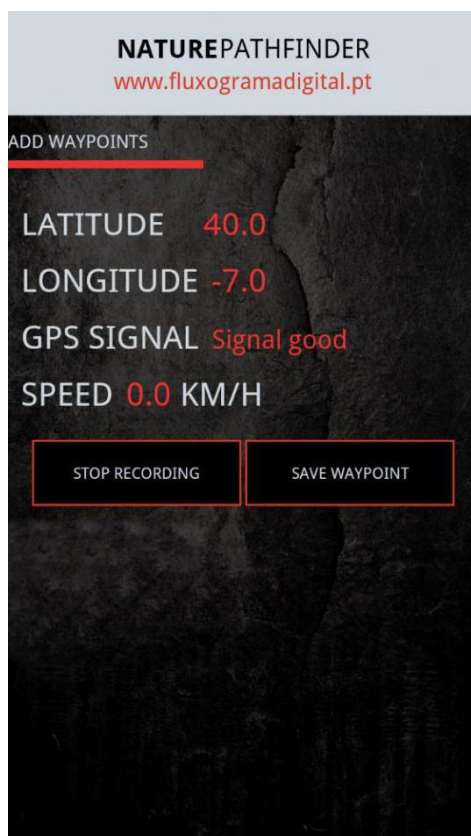


Figura 49 - Menu de gravação das notas

Foi utilizada a classe *Location*<sup>44</sup>, que é responsável por representar a localização geográfica atual do dispositivo móvel, tendo em conta as coordenadas latitude e longitude, assim como permitir saber a velocidade de deslocamento e a exatidão do sinal *GPS*. Nesta classe foram usados os métodos *getLatitude* e *getLongitude*, que irão fornecer as coordenadas em graus. Quanto à velocidade, esta foi obtida com recurso ao método *getSpeed* (da mesma classe), que fornece a velocidade em metros por segundo. A exatidão do sinal *GPS* é obtida a partir do método *getAccuracy*.

---

<sup>44</sup> <http://developer.android.com/reference/android/location/Location.html>

### 4.4.2.1 Configuração da frequência das leituras do *GPS* para poupança de bateria

Como a aplicação atual tem a necessidade de saber quase constantemente qual a localização atual, para que a possa guardar sempre que o utilizador o desejar, é necessário que sejam efetuadas leituras constantes a partir do sensor de *GPS*. Para se obter estas leituras são pedidas atualizações da localização ao sensor depois de um intervalo de tempo definido ou depois de percorrida uma certa distância, utilizando o método *requestLocationUpdates*<sup>45</sup> referente à classe *LocationManager*. Esta classe que tem como objetivo permitir o acesso aos serviços de localização, possibilitando a obtenção de atualizações de localização do dispositivo móvel.

Dois dos parâmetros desta função são precisamente os que milissegundos devem passar para que se volte a efetuar uma leitura e quantos metros se devem percorrer para se realizar também nova leitura. Neste caso, o tempo definido tem precedência relativamente à distância percorrida, ou seja, será feita uma nova leitura depois de ter sido percorrida a distância definida no parâmetro, mas apenas se tiver passado o tempo definido<sup>46</sup>.

Como foi analisado anteriormente na secção 4.4.1, uma das desvantagens da utilização do sensor de *GPS* dos dispositivos móveis para obtenção da localização é o seu grande consumo de energia, sendo que esta desvantagem se agrava se for necessária a sua utilização durante todo o tempo de execução da aplicação. Para minimizar este problema decidiu-se reduzir o número de leituras a partir do sensor *GPS*, especialmente quando não se esteja a realizar gravação de notas, aumentando o intervalo entre cada leitura. Para se controlar o número de metros e o número de segundos entre cada leitura optou-se por utilizar a velocidade atual, como se pode ver na Tabela 8.

*Tabela 8 - Utilização da velocidade para determinar o número de metros e segundos entre cada leitura do GPS*

<b>Velocidade (metros por segundo)</b>	<b>Menor ou igual a 5</b>	<b>Maior que 5 e menor ou igual a 11</b>	<b>Maior que 11</b>
Metros entre leituras	0	10	40
Segundos entre leituras	0	5	40

<sup>45</sup> <https://developer.android.com/training/location/receive-location-updates.html>

<sup>46</sup>

<http://developer.android.com/reference/android/location/LocationManager.html#requestLocationUpdates%28long,%20float,%20android.location.Criteria,%20android.app.PendingIntent%29>

Depois de realizados alguns testes de desempenho da aplicação diretamente no terreno (secção 5.1), utilizando vários valores de segundos e metros entre leituras, decidiu-se que o máximo valor para cada um destes parâmetros seria o que está registado na Tabela 8.

Relativamente aos valores da velocidade menores ou iguais a 5 metros por segundo, optou-se por fazer o número máximo de leituras a partir do sensor *GPS*, pois é a velocidades reduzidas que são gravadas as notas dos *roadbooks*. Se se tiver em consideração que, tal como vai ser visto na secção 4.4.3, apenas será permitido gravar notas quando a velocidade for menor que 3 metros por segundo, sempre que o utilizador se deslocar a velocidades maiores não há necessidade que sejam realizadas mais leituras a partir do sensor de *GPS*. Nesta tabela pode ver-se que irão ser realizadas maior número de leituras quando a velocidade é menor que 5 metros por segundo e não apenas quando é menor que 3 metros por segundo, para que haja uma pequena margem de transição e adaptação antes que seja permitida a gravação de notas pelo programa (menos de 3 metros por segundo).

Relativamente aos valores de velocidade maiores que 5 metros por segundo e menores ou iguais a 11 metros por segundo, trata-se de uma velocidade considerada normal (tendo em conta os testes efetuados) para um utilizador num veículo que percorra caminhos de “terra batida”. Assim sendo, foram definidos os valores dos metros entre leitura e segundos entre leituras representados na Tabela 8, por se considerar que são valores que permitem uma atualização da frequência das leituras do *GPS* equilibrada entre a necessidade de atualização do sinal *GPS* e a poupança de energia. Quando o utilizador se desloca a estas velocidades não estará perto de um ponto que pretenda registar, pois ao aproximar-se de um cruzamento terá de diminuir a velocidade.

Para valores de velocidade maiores que 11 metros por segundo não serão necessárias tantas atualizações da frequência das leituras do *GPS* como nos casos anteriores, pois de acordo com a experiência resultante dos testes de desempenho da aplicação (secção 5.1), um utilizador da aplicação que se desloque a uma velocidade superior a 11 metros por segundo certamente não estará perto de um cruzamento.

#### **4.4.2.2 Exatidão e precisão**

Tratando-se de uma aplicação baseada em deteção de contextos, necessita de monitorizar constantemente a exatidão do sinal *GPS*, para poder adaptar o seu comportamento de acordo com mudanças neste parâmetro, nomeadamente não permitindo a gravação de uma nota ou passando para navegação manual sempre que seja detetada uma exatidão do sinal *GPS* muito reduzida. Como

foi referido na secção 4.4.2, é utilizado o método *getAccuracy* para se obter a exatidão do sinal. Para a definição do termo exatidão é importante referir que, muitas vezes, existe confusão entre a definição de exatidão e precisão, quando se interpreta o termo *Accuracy*.

Como exatidão pode entender-se o quão próximo o valor lido está do valor real, ou seja, é a distância entre a medida que se obteve e o seu verdadeiro valor.

Quanto à precisão este valor tem a ver com a dispersão das medidas lidas que foram repetidas nas mesmas condições. Se depois de efetuadas diversas medidas os valores obtidos variarem muito então estamos perante um caso de baixa precisão (Silva H. D., 2006).

Como se pode ver na Figura 50, no primeiro caso todos os valores estão longe do alvo, logo não existe exatidão.

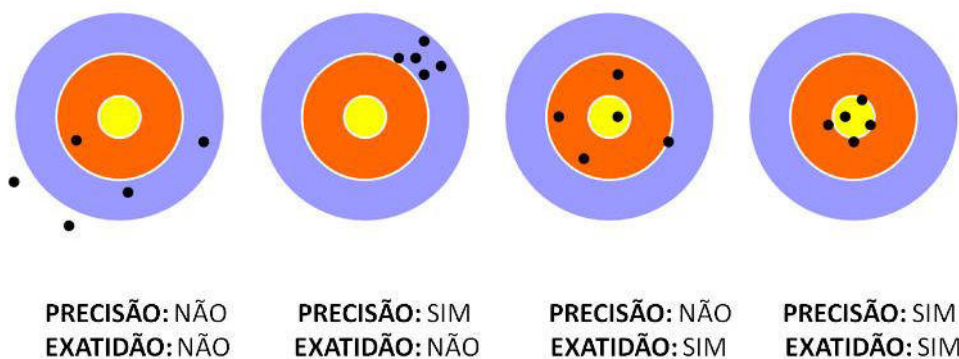


Figura 50 - Diferença entre precisão e exatidão<sup>47</sup>

Vê-se também que os valores são todos muito diferentes uns dos outros, logo também não existe precisão. No segundo exemplo não existe exatidão porque os valores estão longe do alvo pretendido, mas existe precisão porque os valores lidos são muito próximos uns dos outros. Já no terceiro caso acontece o inverso, existe exatidão pois os valores estão próximos da leitura ideal que seria o centro do alvo, mas não existe precisão porque mesmo assim os valores são muito diferentes uns dos outros. No último caso existe bastante precisão e exatidão, pois os valores lidos estão próximos dos valores considerados corretos e nas leituras que foram feitas obtiveram-se valores muito próximos.

<sup>47</sup> Retirado de: <http://calibraend.blogspot.pt/2013/02/voce-conhece-diferenca-entre-precisao-e.html>

### 4.4.2.3 Exatidão do sinal *GPS*

Na aplicação desenvolvida, a exatidão tem um papel muito importante, pois irá ser o fator essencial para que um *roadbook* seja criado corretamente e depois possa ser utilizado para a navegação sem erros ou com erros reduzidos. Como foi referido anteriormente, depois de ser detetado o sinal de *GPS* vai ser calculada a exatidão do respetivo sinal, de forma a se verificar se é viável proceder-se à gravação dos pontos, tendo em conta que só se deverá gravar um *roadbook* quando a exatidão do sinal for considerada média ou boa.

Como referido na secção 4.4.2, foi usado o método *getAccuracy*, para se obter o valor da exatidão do sinal. Tal como definido no site oficial do desenvolvimento para dispositivos *Android*<sup>48</sup>, a exatidão é o raio de confiança de 68% em redor do ponto onde o *GPS* se encontra (latitude e longitude). Este raio é definido em metros. Ou seja, existe a probabilidade de 68% da localização correta estar dentro do círculo cuja exatidão é dada pelo método *getAccuracy*. Analisando a Figura 51 pode ver-se um exemplo de uma leitura registada pelo método *getAccuracy*. Neste caso a leitura teve o valor de 5 metros, o que significa que existe uma probabilidade de 68% de a localização correta estar dentro de um raio de 5 metros em redor do dispositivo móvel.



Figura 51 - Exemplo de uma leitura efetuada pelo método *getAccuracy*

---

<sup>48</sup> [http://developer.android.com/reference/android/location/Location.html#getAccuracy\(\)](http://developer.android.com/reference/android/location/Location.html#getAccuracy())

### 4.4.3 Gravar notas tendo em conta a velocidade e a exatidão

Tendo em conta o fluxograma da Figura 48, e tal como referido anteriormente, o utilizador terá acesso aos valores da latitude, longitude, velocidade e exatidão. O próximo passo será o utilizador encontrar um cruzamento e clicar no botão *Save Waypoint* (Figura 49) que permite registar os dados desse cruzamento, criando uma nota do *roadbook*.

No entanto, esta ação irá depender da velocidade atual e da exatidão do sinal. Como se pode ver no fluxograma Figura 52, depois de o utilizador clicar no botão *Save Waypoint*, irá ser verificado se a velocidade é menor ou igual a 3 metros por segundo. Se isto não se verificar vai ser indicado ao utilizador que deve reduzir a velocidade para se poder guardar a localização da nota atual. Isto deve-se ao facto de se o utilizador estiver em movimento com uma determinada velocidade pode acontecer que as coordenadas que ficam guardadas não corresponderão ao cruzamento, mas a uma localização já fora do mesmo.

O valor desta velocidade de controlo foi obtido através de diversos testes no terreno em que se verificou que a partir desta velocidade seria difícil que as coordenadas ficassem guardadas corretamente e correspondessem ao cruzamento em questão.

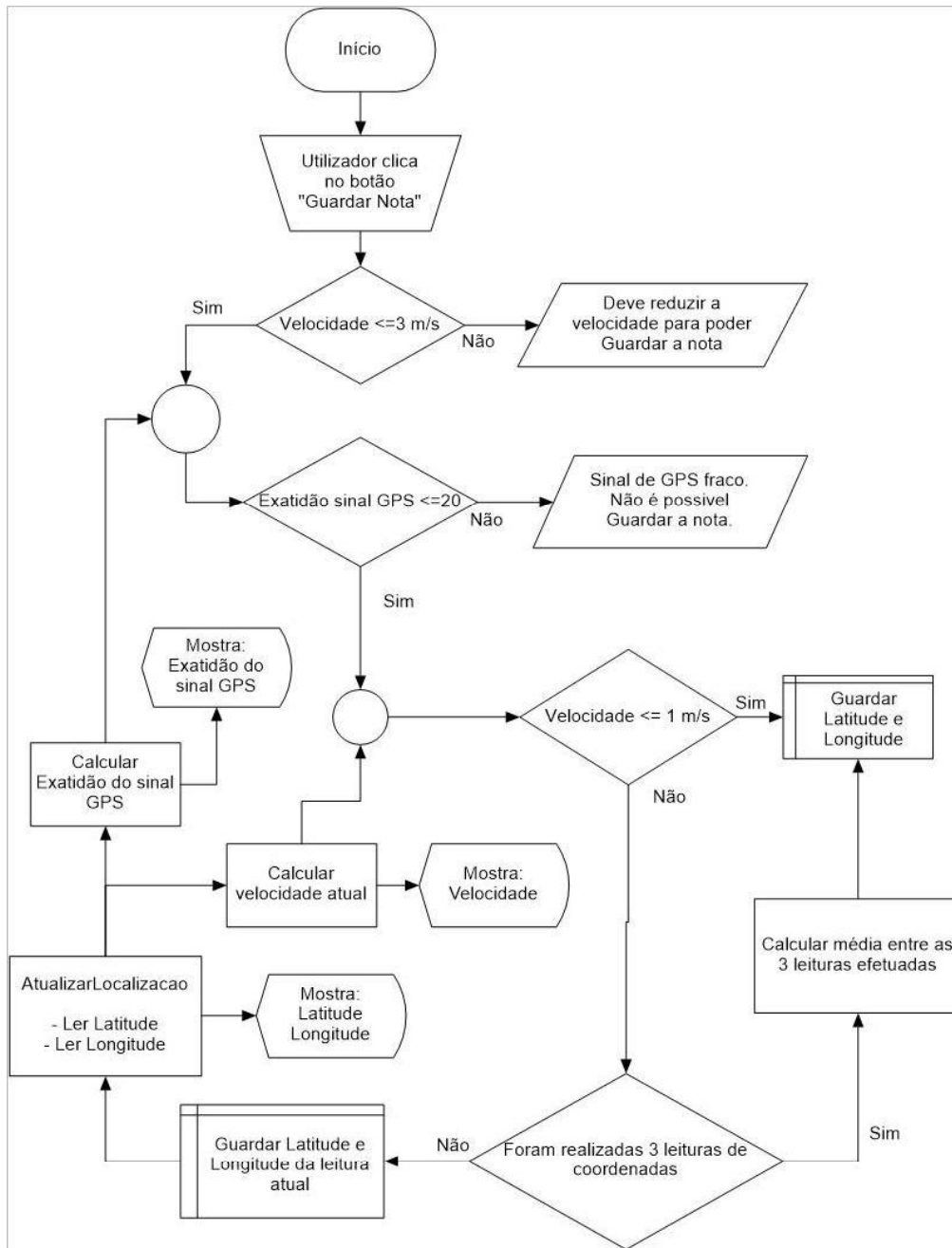


Figura 52 - Fluxograma da verificação da exatidão e velocidade para início do processo de guardar as coordenadas

Caso se verifique que a exatidão seja menor ou igual a 20 metros, inicia-se o processo de registo das coordenadas latitude e longitude. Se o utilizador estiver praticamente parado (velocidade menor ou igual a 1 metro por segundo) então serão guardadas as coordenadas latitude e longitude



atuais. Caso o utilizador se esteja a deslocar a uma velocidade maior que 1 metro por segundo serão feitas 3 leituras das coordenadas *GPS* e feita a média entre elas. Será este o valor final que será guardado como sendo a localização do ponto atual. Optou-se por utilizar esta estratégia pois mesmo estando o utilizador a deslocar-se muito lentamente corria-se o risco de se registar uma coordenada afastada do centro do cruzamento. Desta forma garante-se que se vai registar um valor mais próximo da coordenada central do cruzamento.

#### 4.4.4 Inserir restantes dados da nota

Como se pode ver no fluxograma da Figura 53, depois de se ter guardado a latitude e longitude da nota, passa-se a introduzir a informação acerca da mesma. De seguida passa-se a escolher o desenho representativo do esquema da nota.

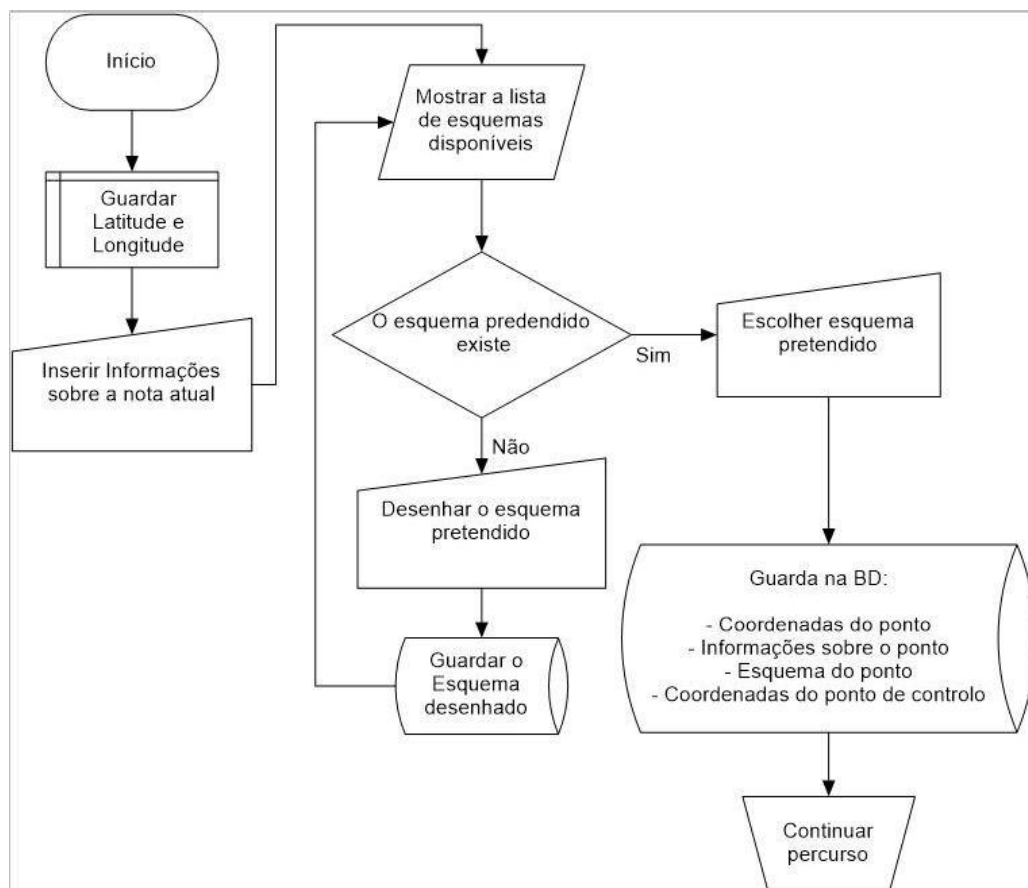


Figura 53 - Inserção da informação e esquema da nota guardada

Caso exista já um esquema que possa ser usado, o utilizador poderá escolhe-lo (Figura 54). Se o esquema pretendido não existir, o utilizador poderá cria-lo e guarda-lo, tal como foi descrito na secção 4.3. Poderá então ser escolhido o esquema que se acabou de desenhar.

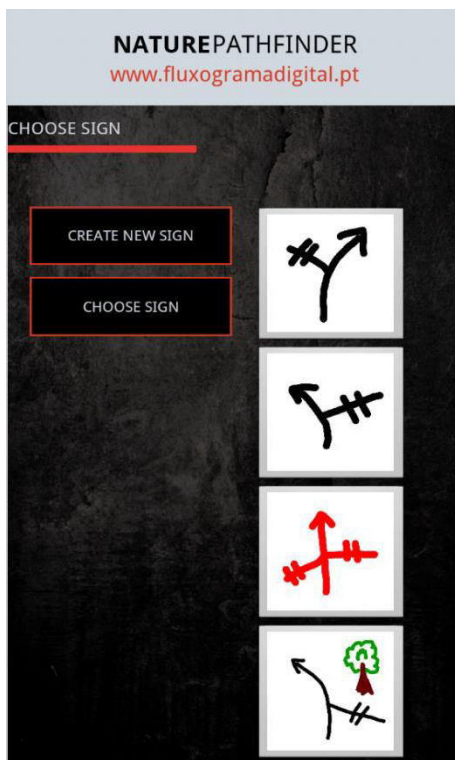


Figura 54 - Menu de escolha do esquema da nota atual

Ao escolher um esquema, o utilizador deverá responder a uma mensagem de confirmação do esquema escolhido. Após este passo serão efetivamente guardados os dados relativos à nota, nomeadamente as coordenadas, a informação, o desenho do esquema e as coordenadas de controlo (secção 4.4.5), que serão enviados para a base de dados *SQLite*. No caso da informação do cruzamento, esta poderá conter avisos de perigo ou outras indicações a que o utilizador deva estar atento aquando da navegação do *roadbook*. Se não for inserida nenhuma informação pelo utilizador, será guardada automaticamente uma mensagem a indicar que se deve seguir o percurso principal. Este tipo de mensagem é usual dos *roadbooks* tradicionais e serve como mensagem indicadora para que o utilizador saiba que deve seguir sempre o percurso principal, se não houver indicações em contrário. Após esta etapa o utilizador irá continuar o percurso, procurando o próximo ponto que irá gravar numa nova nota.

### 4.4.5 Gravar ponto de controlo

O ponto de controlo vai ser utilizado, aquando da navegação, para se verificar se depois de passar um cruzamento o utilizador segue pelo caminho correto. É de referir que o ponto de controlo de um cruzamento é sempre gravado automaticamente pela aplicação, a partir do afastamento que é feito desde o cruzamento anterior. Como se pode ver no exemplo representado na Figura 55, durante a gravação dos pontos do *roadbook* o utilizador desloca-se até ao ponto A. Neste ponto pode parar ou reduzir a velocidade até menos de 3 metros por segundo e de seguida grava o ponto referente a esse cruzamento. Depois de efetuar esta gravação retoma a marcha e, passados alguns metros, no ponto B, a aplicação irá gravar automaticamente o ponto de controlo, referente ao próximo cruzamento. A gravação deste ponto de controlo é feita sem que o utilizador se aperceba ou tenha qualquer intervenção.

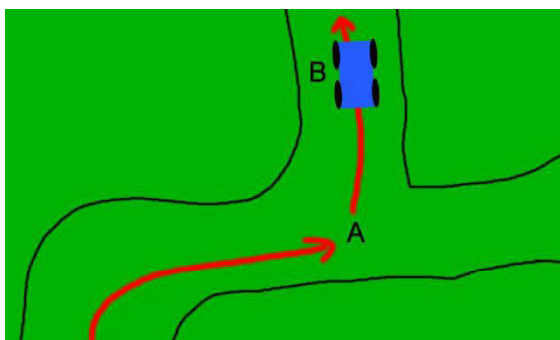


Figura 55 - Gravação do Ponto de Controlo

Como se pode ver na Figura 56, sempre que são gravados os dados de um cruzamento, vai ser calculada a distância percorrida em linha reta. Após o utilizador se ter afastado 40 metros irão ser guardadas automaticamente as coordenadas atuais como sendo as do ponto de controlo. Caso se atinja outro cruzamento e o utilizador pressione o botão “Guardar Nota” sem que antes tenha sido guardado o ponto de controlo, significa que esse novo cruzamento está muito próximo do anterior e não há necessidade do ponto de controlo, pelo que o mesmo fica com as mesmas coordenadas do cruzamento anterior. Depois de guardadas as coordenadas do ponto de controlo, as mesmas são adicionadas à base de dados durante o registo dos dados do cruzamento seguinte. Relativamente à distância de 40 metros, que foi definida como sendo a distância que seria necessário ser percorrida a partir do cruzamento anterior para que fosse guardado o ponto de controlo, a mesma foi definida

com base nos testes realizados no terreno. Trata-se de uma distância que permite que o utilizador se afaste significativamente do cruzamento anterior e percorra alguma distância do caminho correto, permitindo que o ponto de controlo seja marcado no novo caminho e ao mesmo tempo não se trata de uma distância muito grande, que fizesse com que eventualmente pudesse haver muitos cruzamentos que se encontrassem antes de o utilizador se afastar o suficiente do cruzamento para gravar o ponto de controlo. Se um cruzamento estiver tão próximo do anterior que não seja guardado o ponto de controlo, significa que o utilizador nunca se irá perder nesta parte do percurso e não há necessidade de um ponto de controlo.

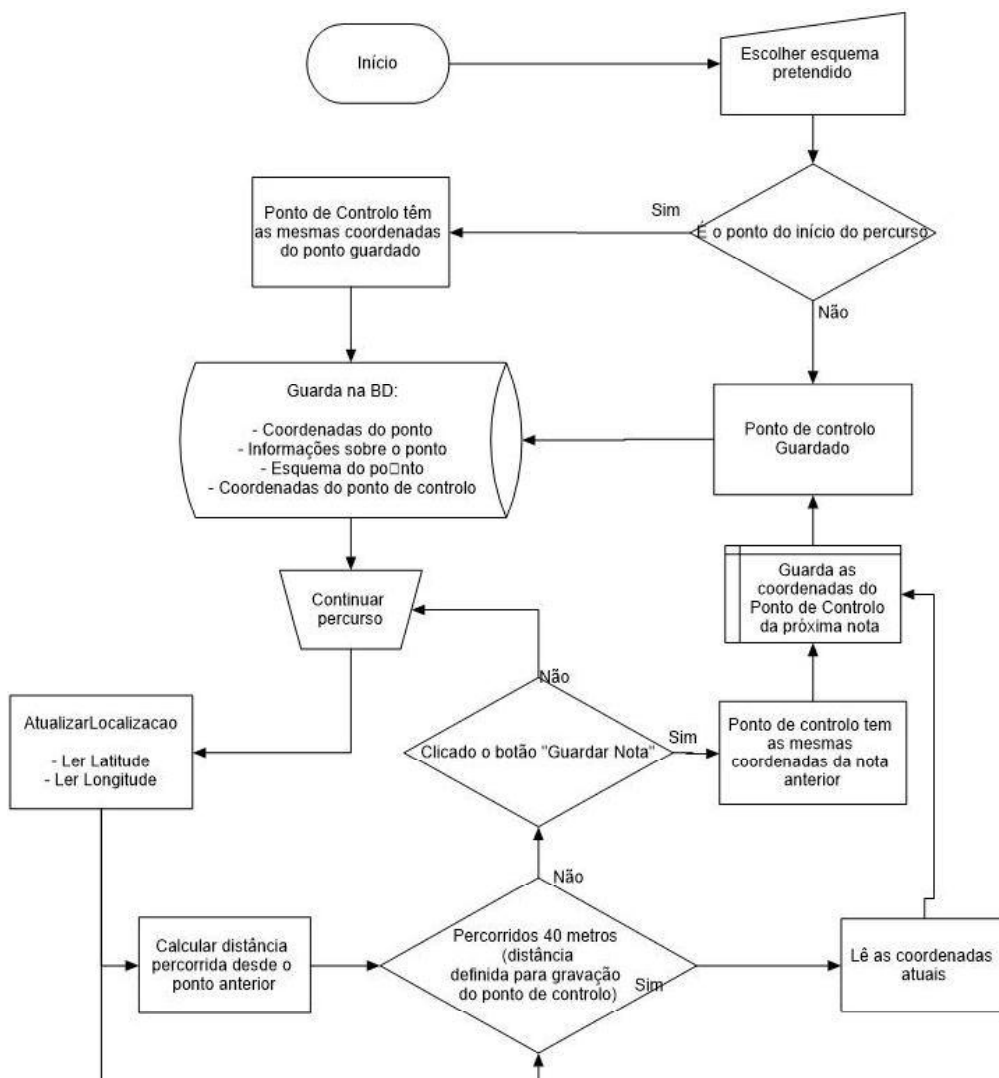


Figura 56 - Fluxograma referente à gravação do ponto de controlo

O fluxograma completo do módulo de gravação das notas dos *roadbooks* pode ser consultado no Anexo B.

## 4.5 Módulo de navegação de um *roadbook*

Nesta secção vai ser explicado o processo de navegação utilizando um *roadbook* previamente criado e armazenado na base de dados da aplicação.

O utilizador irá ter acesso à informação referente à nota do cruzamento seguinte, nomeadamente ao desenho do esquema que lhe permite identificar por que caminho seguir quando encontrar um cruzamento. Nessa altura, a aplicação irá passar automaticamente para a nota do cruzamento seguinte, mostrando a informação respetiva. Se for cometido um erro de navegação a aplicação irá avisar o utilizador e iniciar o processo de correção da rota.

Também nesta etapa recorreu-se ao sensor de *GPS* do dispositivo, de forma a serem obtidas as coordenadas dos pontos que definem o trajeto que se está a seguir. Caso o *GPS* do dispositivo não esteja ativo a aplicação irá mostrar uma mensagem ao utilizador indicando que o deve ativar.

Como se pode ver no fluxograma da Figura 57, após as configurações iniciais começa-se o processo de obtenção de sinal de *GPS*. Enquanto não seja detetado sinal de *GPS* estará sempre visível uma mensagem que indica ao utilizador que ainda não há sinal.

A partir do momento que é detetado sinal *GPS*, irão ser mostradas as coordenadas da latitude e da longitude, assim como será calculado o valor da exatidão e da velocidade atual, valores estes que também serão visíveis no *layout* atual (Figura 56). Sempre que seja detetada uma alteração na posição estes valores vão ser atualizados.

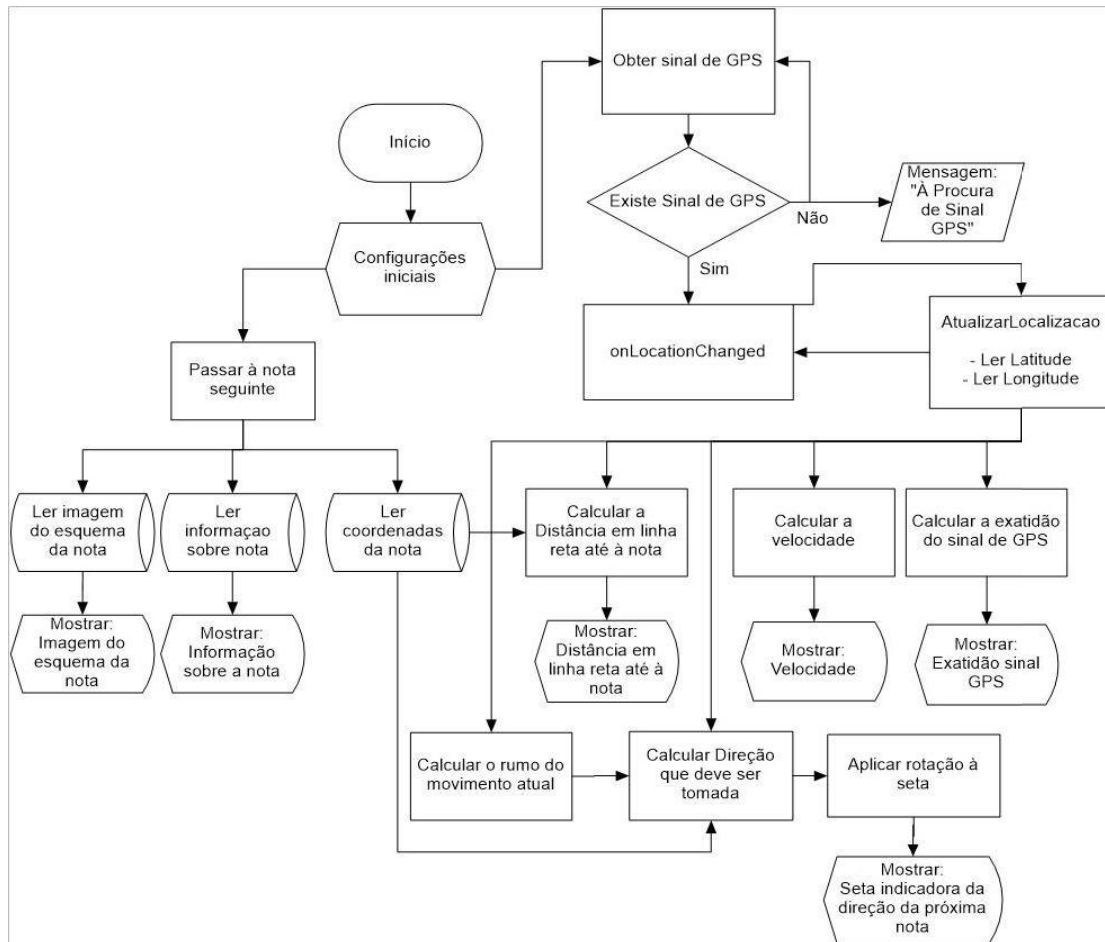


Figura 57 - Fluxograma dos processos de obtenção das coordenadas e restante informação necessária para a navegação

Para além destes dados, a aplicação irá aceder à base de dados onde estão guardadas as notas do *roadbook*, de onde serão lidos dados referentes à nota seguinte. Será mostrado o desenho do esquema da próxima nota, assim como a informação referente a essa nota. São também consultadas, a partir da base de dados, as coordenadas da latitude e longitude da mesma, para que possa ser calculada a distância em linha reta a que se encontra a próxima nota, assim como a direção para lá chegar. Esta direção é representada por uma seta indicadora que está constantemente a girar na direção em que se encontra o próximo cruzamento. Não se trata do método principal de navegação, mas sim de um método auxiliar, que complementa o método de navegação principal – imagens dos esquemas dos cruzamentos.

### 4.5.1 Configuração da frequência de leituras do *GPS* para poupança de energia

Durante esta fase irá ser feita a adaptação do número de leituras efetuadas pelo sensor de *GPS*, para que se possa poupar bateria do dispositivo móvel, tal como foi descrito na secção 4.4.2.1. No entanto, neste caso em que se está a proceder à navegação, optou-se por fazer a adaptação das leituras tendo em conta a distância em linha reta a que se encontra o utilizador do próximo cruzamento. Basicamente, quanto mais afastado estiver o utilizador do próximo cruzamento, menor será o número de leituras efetuadas.

Neste caso, para se controlar o número de metros e o número de segundos entre cada leitura (secção 4.4.2.1), optou-se por utilizar os valores que se podem ver na Tabela 9.

*Tabela 9 - Utilização da distância para determinar o número de metros e segundos entre cada leitura do GPS*

<b>Distância em linha reta ao próximo cruzamento (metros)</b>	<b>Menor que 100</b>	<b>Maior igual a 100 e menor que 500</b>	<b>Maior igual a 500 e menor que 2000</b>	<b>Maior ou igual a 2000</b>
Metros entre leituras	0	10	20	60
Segundos entre leituras	0	10	20	60

Depois de realizados os testes de desempenho da aplicação diretamente no terreno (secção 5.1), utilizando vários valores de segundos e metros entre leituras, decidiu-se que os valores para cada um destes parâmetros seriam os que estão registados na Tabela 9. Desta forma, reduzindo o número de leituras quando o utilizador se encontra ainda longe do próximo cruzamento, é possível reduzir o consumo de energia da aplicação.

Relativamente a valores de distâncias menores que 100 metros, em linha reta até ao próximo ponto, optou-se por se usar a maior frequência de leituras do *GPS*, pois estando perto de um ponto registado é importante que existam o máximo número de leituras a partir do *GPS* para que não ocorram falhas de deteção dos pontos.

Com distâncias maiores ou iguais a 100 metros e menores que 500 metros optou-se por reduzir um pouco a frequência das leituras do *GPS* para poupança de bateria, na medida que estando mais longe do cruzamento não haverá necessidade de tantas atualizações do sinal.

O mesmo princípio foi aplicado para valores maiores ou iguais a 500 metros e menores que 2000 metros. Relativamente a valores de distância maiores que 2000 metros optou-se por só se fazer atualizações de 60 em 60 metros ou passados 60 segundos, pois neste caso o utilizador estará bastante longe do cruzamento e a atualização da leitura do sinal do *GPS* não será tão importante, permitindo uma maior poupança de energia. Estes valores resultaram das experiências efetuadas aquando da realização dos testes de desempenho da aplicação ao longo do desenvolvimento da aplicação. Os valores dos metros entre leituras e segundos entre leituras são valores empíricos, definidos como válidos durante os vários testes realizados.

## 4.5.2 Rotação da Seta indicadora da direção

Tal como foi referido, para além do desenho do esquema representativo do cruzamento será também apresentada uma seta que indica a direção em que fica o próximo cruzamento e funcionará como auxiliar de navegação, tal como se pode ver na Figura 58.

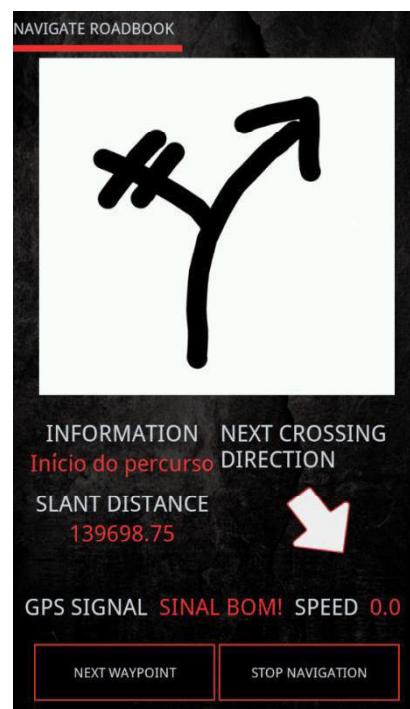


Figura 58 - Imagem indicadora da direção a tomar



Para a implementação da rotação desta seta, em primeiro lugar é calculada a direção que o utilizador está atualmente a tomar, utilizando o método *getBearing*, que faz parte da classe *Location*<sup>49</sup>. Este método devolve a direção horizontal do deslocamento em graus, independentemente da orientação do dispositivo móvel<sup>50</sup>.

Se for analisado o conceito de rumo (*bearing*) pode dizer-se que se trata de uma medida de direção entre dois pontos, que é apresentado num de dois formatos – azimute ou quadrante. O rumo baseado em azimute utiliza 360 graus para indicar a direção, enquanto o rumo por quadrante divide cada um dos quatro quadrantes em 90 graus<sup>51</sup>. No caso do método usado (*getBearing*) devolve o rumo no formato azimute, ou seja, representando a direção de 0 a 360 graus, sendo que a direção de referência é a linha norte, em que o ângulo é zero. Na Figura 59 pode ver-se as relações entre os ângulos e os pontos cardeais. Isto significa, por exemplo, que se o utilizador estiver a deslocar-se e obtiver um rumo de 45 graus está a deslocar-se para nordeste.

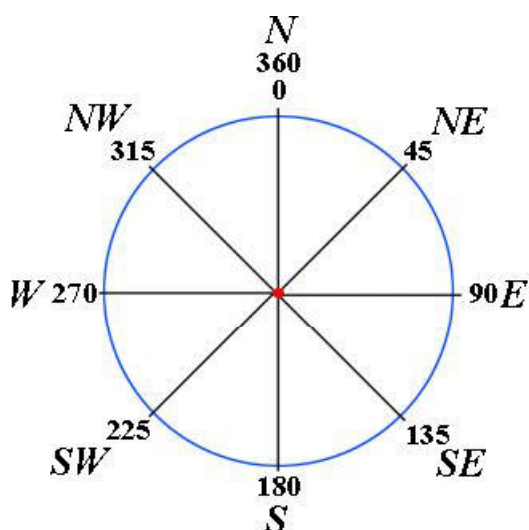


Figura 59 - Representação gráfica dos ângulos em relação ao rumo<sup>52</sup>

<sup>49</sup> <http://developer.android.com/reference/android/location/Location.html>

<sup>50</sup> [http://developer.android.com/reference/android/location/Location.html#getBearing\(\)](http://developer.android.com/reference/android/location/Location.html#getBearing())

<sup>51</sup> [http://geology.isu.edu/geostac/Field\\_Exercise/topomaps/bearing.htm](http://geology.isu.edu/geostac/Field_Exercise/topomaps/bearing.htm)

<sup>52</sup> Retirado de: <http://academic.brooklyn.cuny.edu>

Depois de se saber o rumo atual, passa-se ao passo seguinte que consiste em calcular a direção entre as coordenadas atuais e as coordenadas do próximo cruzamento, utilizando o método *bearingTo*<sup>53</sup>, da classe *Location*. Este método devolve o rumo aproximado, em graus, entre a localização atual e a localização de destino.

Para se calcular a direção da seta que vai apontar para o próximo cruzamento não basta usar o valor calculado da direção entre a localização atual e a localização de destino. Isso seria apenas viável se o utilizador estivesse a deslocar-se originalmente para norte (rumo=0°), mas como o utilizador poderá estar a deslocar-se em qualquer direção terá de se subtrair o rumo atual ao rumo entre a localização atual e o cruzamento (Equação 1).

$$A = Rc - Ra \quad (1)$$

Onde *A* é o angulo de rotação, em graus, que a seta deve ter em relação à vertical (norte), *Rc* é o rumo entre o ponto atual e o próximo cruzamento e *Ra* é o rumo atual.

### 4.5.3 Definir o tipo de navegação (automática ou manual)

Tratando-se de uma aplicação baseada em deteção de contextos, uma das suas principais características é permitir uma navegação automática. Com esta funcionalidade, a aplicação deteta que se encontra num cruzamento, avisa o utilizador que deverá escolher um caminho a seguir baseando-se no desenho do esquema apresentado e passa automaticamente para a nota seguinte, mostrando as suas informações. No entanto, nos casos em que a exatidão do sinal *GPS* seja muito baixa, a aplicação irá sugerir ao utilizador que proceda a uma navegação manual, sendo o próprio utilizador a realizar o processo de passagem para a nota seguinte assim que chegue a um cruzamento. Isto acontece para prevenir possíveis erros de navegação que poderiam acontecer no caso de se captar pouco sinal *GPS*.

---

<sup>53</sup> <http://developer.android.com/reference/android/location/Location.html>

Analisando o fluxograma da Figura 60, pode ver-se que, caso a exatidão seja maior que 25 metros irá ser indicado ao utilizador que deverá proceder à navegação manual, pois não existe exatidão suficiente para que se possa utilizar o *roadbook* sem que haja muitos erros de navegação. Obviamente que isto tratar-se ia de uma situação extrema, pois tal como foi analisado na secção 4.4.1.1, para que um sinal de *GPS* tenha uma exatidão tão baixa deverá haver um conjunto de condicionantes que afetem ao mesmo tempo o sinal de *GPS*.

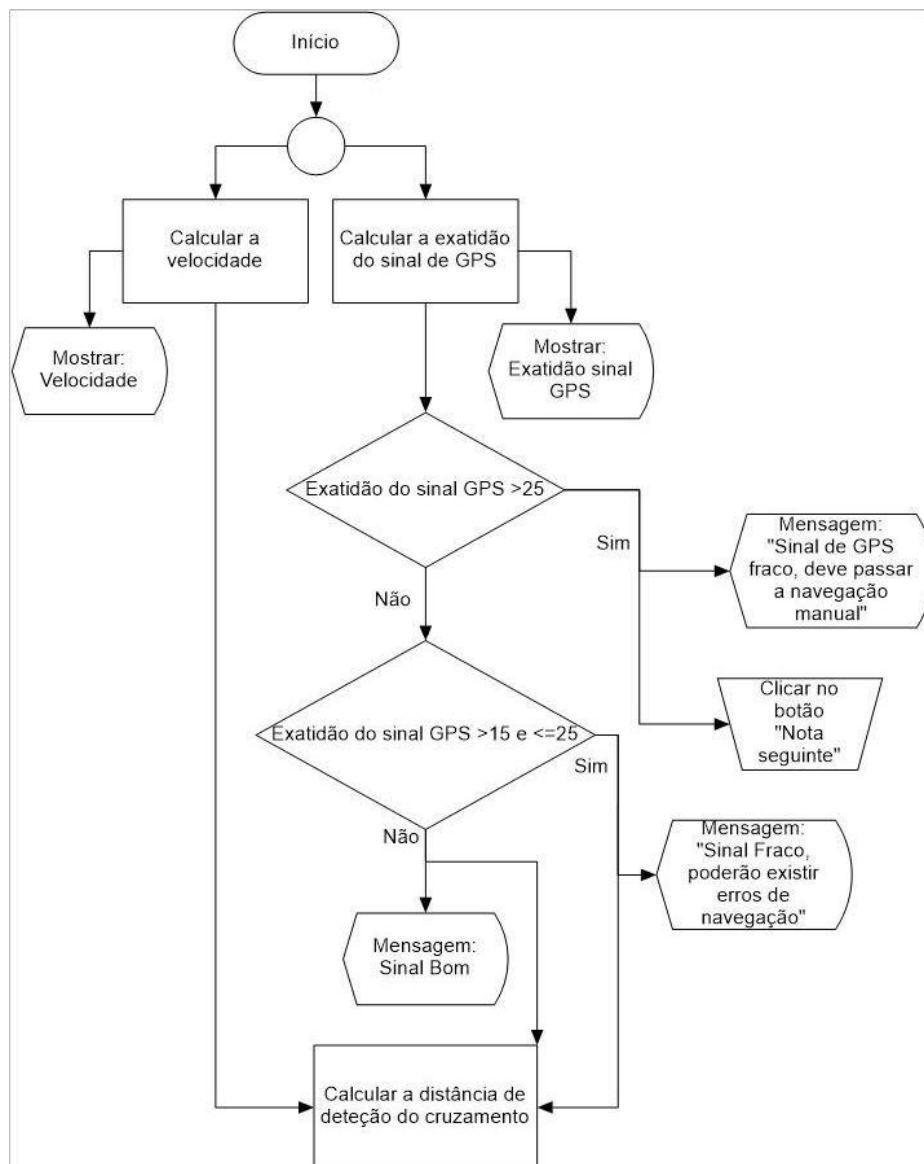


Figura 60 - Fluxograma para controlo do modo de navegação

Se a exatidão do sinal *GPS* se encontrar em valores entre 15 e 25 metros a navegação poderá ser feita de modo automático, no entanto o utilizador será advertido que poderão existir erros de navegação. Caso o sinal seja inferior a 15 metros, será considerado que existe um tipo de sinal bom e poderá proceder-se à navegação automática sem previsão de problemas.

A fase seguinte consiste no processo de cálculo da distância à qual um cruzamento irá ser detetado, para que depois a aplicação possa passar automaticamente para a nota seguinte.

#### 4.5.4 Calcular a distância à qual são detetados os cruzamentos

A Figura 61 ilustra a situação de chegada a um novo cruzamento. Sempre que o utilizador se aproxime de um cruzamento e se encontre a uma determinada distância (neste caso 10 metros) irá ser ouvido um aviso sonoro indicando que se encontra perto do cruzamento e que as informações e o desenho do esquema irão mudar para os do próximo cruzamento.

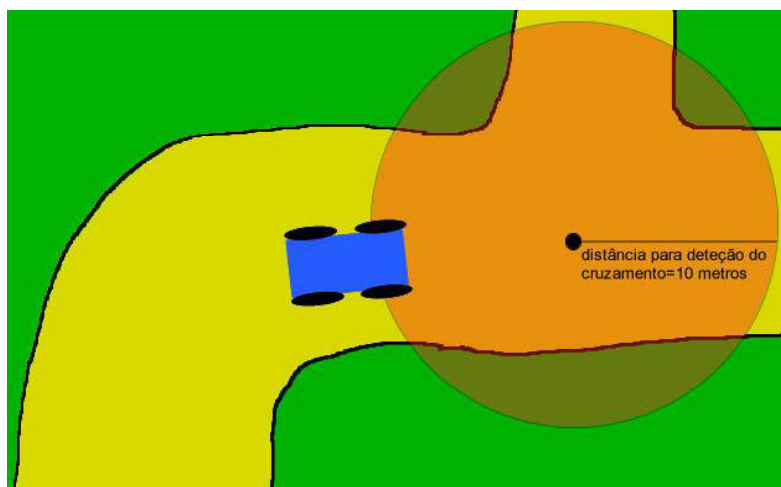
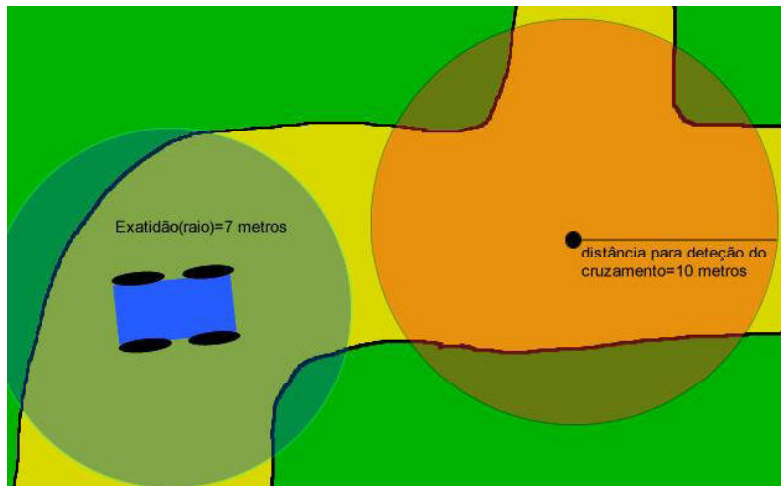


Figura 61 - Esquema de deteção de um cruzamento

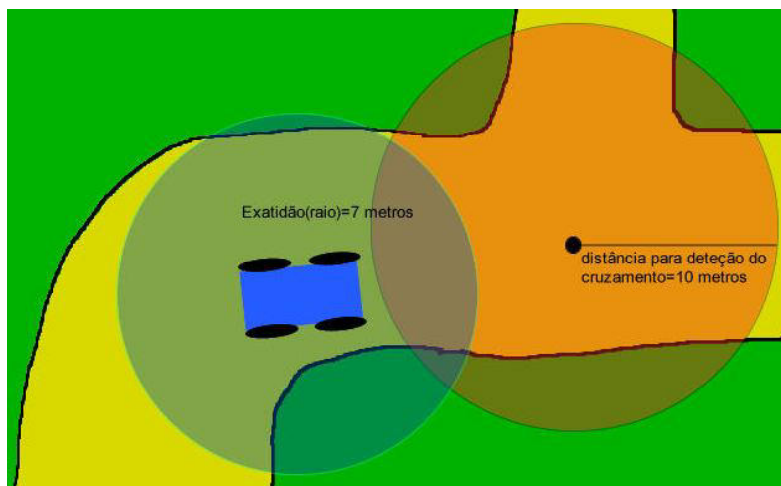
Na realidade a situação não é tão simples quanto a do esquema apresentado anteriormente, pois tem de se ter em consideração a exatidão do sinal e a velocidade de deslocamento do utilizador. Como foi analisado na secção 4.4.2.3, a exatidão é o raio de confiança de 68% em redor do ponto onde o dispositivo *GPS* se encontra. No caso a apresentado na Figura 62 pode ver-se que a exatidão do dispositivo *GPS* é de 7 metros, significando isso que nesse raio existe 68% de pro-

habilidades de estar a coordenada atual correta. Como a distância para deteção do cruzamento é de 10 metros ainda não fui detetado o cruzamento.



*Figura 62 - Representação da exatidão sem detetar cruzamento*

Na situação da Figura 63 já existe interseção entre o raio que representa a distância para deteção do cruzamento e o raio da exatidão do dispositivo, pelo que o cruzamento já pode estar a ser detetado se as coordenadas detetadas se situarem dentro dessa interseção. No entanto, se as coordenadas detetadas se situarem na zona atrás do veículo, mesmo que dentro do círculo que representa a exatidão, o cruzamento ainda não foi detetado.



*Figura 63 - Representação da exatidão a detetar cruzamento*

À medida que o utilizador se desloca para o centro do local das coordenadas dos cruzamentos maior será a probabilidade de ser detetado o cruzamento, pois a interseção entre os dois círculos vai ser maior, como se pode ver na Figura 64.

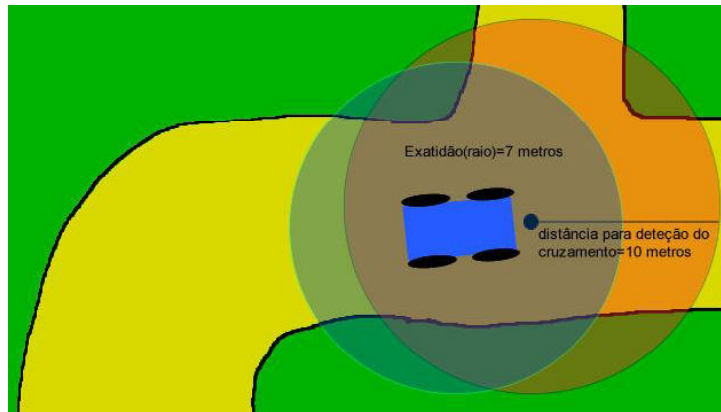


Figura 64 - Esquema da deteção do cruzamento

Nos exemplos apresentados nas figuras anteriores, a distância para deteção do cruzamento estava definida como 10 metros, no entanto esta distância não poderá ter sempre o mesmo valor, pois poderão haver erros de navegação.

Analisando o esquema da Figura 65, no caso de haver dois cruzamentos próximos a aplicação poderá detetar o primeiro cruzamento e detetar imediatamente o segundo, podendo conduzir a um erro de navegação, na medida que o utilizador não se chega a aperceber que houve duas mudanças na informação.

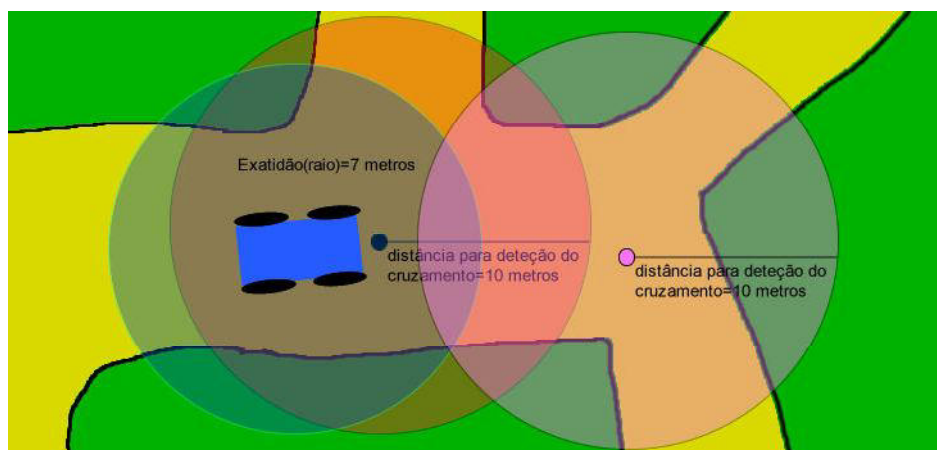


Figura 65 - Esquema de deteção com dois cruzamentos

Podemos então concluir que o ideal seria definir um raio da distância para a detecção do cruzamento o menor possível. No entanto, se o utilizador passar no cruzamento com uma velocidade mais elevada poderá não ser possível detetar o cruzamento, na medida em que foi reduzida a área de interseção entre o raio da exatidão do dispositivo e a distância à qual o cruzamento é detetado e poderão não ser feitas leituras a tempo de ler as coordenadas enquanto há interseção entre as duas circunferências (Figura 66).

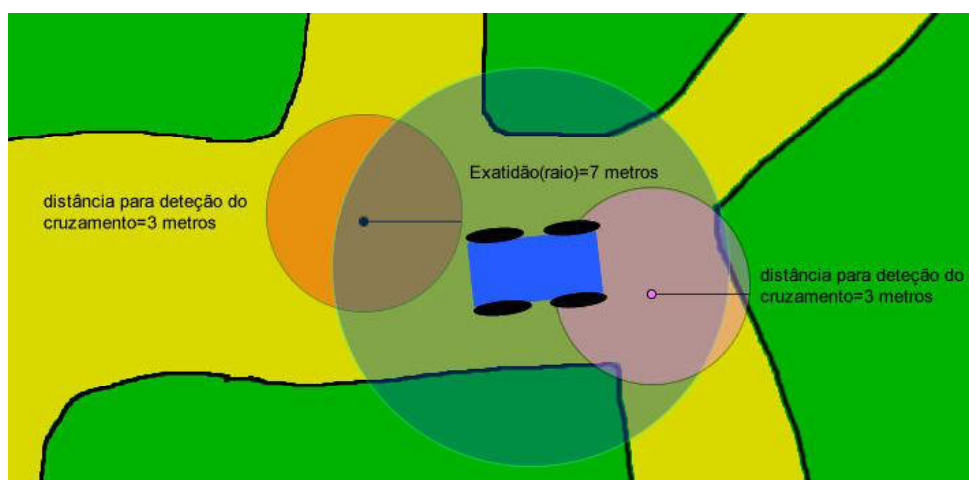


Figura 66 - Diminuição da distância à qual se deve detetar o cruzamento

Para resolver este problema foi usada a Equação 2 que calcula a distância a usar para o raio em função da velocidade.

$$DC = V + V * 0,03 \quad (2)$$

Onde  $DC$  é a distância para detecção do cruzamento e  $V$  é a velocidade atual. Assim é estabelecida uma relação entre a velocidade atual e a distância à qual será detetado o cruzamento. Quanto maior for a velocidade maior também será o raio desta distância. A partir desta equação será calculada constantemente a distância a que vai ser detetado o cruzamento, tendo em conta da velocidade do utilizador.

Na Tabela 10 pode ver-se o exemplo do cálculo da distância a que pode ser detetado um cruzamento, tendo em conta velocidades típicas do utilizador. Estes valores foram testados no terreno e revelaram-se viáveis para esta utilização.

Tabela 10 - Exemplo do cálculo da distância à qual é detetado o cruzamento

Velocidade (km/h)	≤5	10	15	20	25	30	35	≥40
Velocidade (m/s)	1,4	2,8	4,2	5,6	6,9	8,3	9,7	11,1
Distância à qual é detetado o cruzamento (m)	5,2	10,3	15,5	20,6	25,8	30,9	36,1	41,2

Foi também definido que sempre que a velocidade fosse maior que 40 km/h a distância seria sempre 42 metros, pois não haveria necessidade de aumentar muito este valor, mesmo para velocidades maiores, na medida que se este valor aumentar muito um cruzamento poderá ser detetado longe de mais e conduzir a erros de navegação.

Relativamente a valores de velocidade inferiores a 5 km/h, o valor da distância à qual será detetado o cruzamento será sempre 5,2 metros. Esta regra foi definida pois em média o ser humano desloca-se a pé a uma velocidade de 4 km/h, pelo que facilmente quando se procura algo (neste caso o cruzamento) o ser humano poderá deslocar-se a velocidades mais baixas.

Se o valor desta distância não fosse constante para estas velocidades mais baixas, o raio de deteção das coordenadas do cruzamento poderia ser tão pequeno que a probabilidade de essas coordenadas serem encontradas seria reduzida. Foram efetuados testes “no terreno” que comprovaram esta teoria.

#### 4.5.5 Passagem à nota seguinte

Depois de calculada a distância a que vão ser detetados os cruzamentos, a aplicação irá verificar constantemente se a distância em linha reta até à nota seguinte é menor ou igual à distância que foi calculada para se detetarem os cruzamentos.

Se isso acontecer irá passar-se à nota seguinte e os respetivos dados dessa nota serão mostrados no ecrã. Também será ouvido um aviso sonoro que indica que se chegou ao cruzamento e se passa a mostrar a informação do cruzamento seguinte (Figura 14).



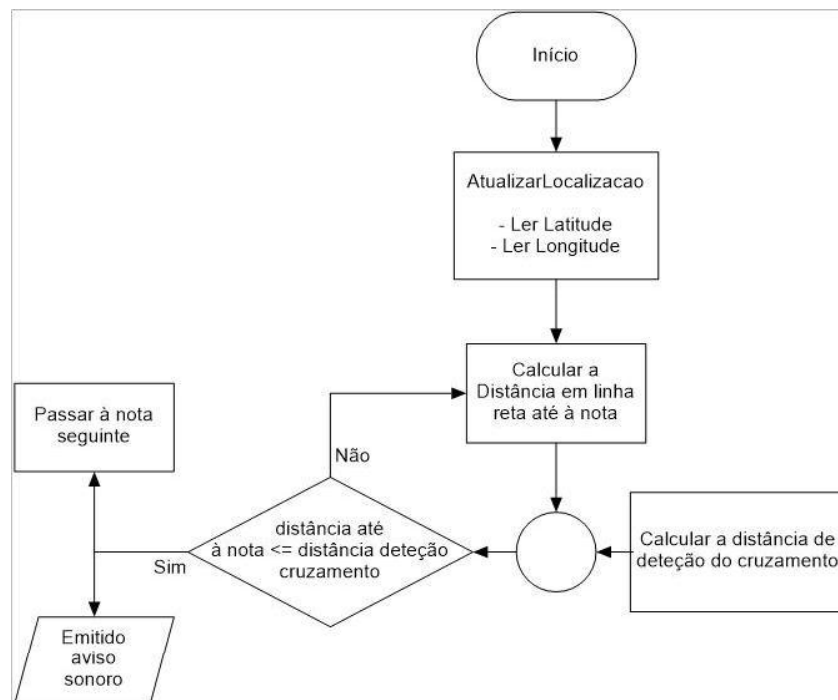


Figura 67 - Fluxograma do algoritmo para a passagem à nota seguinte

### 4.5.6 Verificar se o utilizador seguiu pelo caminho correto

Depois de se detetar um cruzamento e surgir a nota do cruzamento seguinte, o utilizador irá seguir pelo caminho que era indicado na nota anterior. A aplicação irá então verificar se é detetado o ponto de controlo, que tinha sido registado automaticamente durante da gravação do percurso, tal como explicado na secção 4.4.5.

Como se pode observar no exemplo da Figura 68, o utilizador chegou a um cruzamento (ponto A) e de seguida seguiu em frente, quando o caminho correto seria virar à esquerda. Como passados alguns metros não é detetado o ponto de controlo a aplicação assume que houve um erro de navegação. Este processo é realizado sem que haja intervenção ou conhecimento do utilizador. O utilizador será apenas avisado que ocorreu um erro e seguiu por um caminho errado, através de um aviso sonoro e uma mensagem. O processo de correção da rota será explicado na secção 4.6.



Figura 68 - Erro de navegação

Tal como referido anteriormente, depois detetado um cruzamento irá ser calculada a distância em linha reta entre esse cruzamento e a localização atual do dispositivo. Se o utilizador se afastar 60 metros do cruzamento anterior e a aplicação não detetar as coordenadas do ponto de controlo, então isto significa que o utilizador estará a percorrer um caminho errado. A técnica usada para detetar o ponto de controlo é a mesma usada quando da localização de um cruzamento, tal como descrito na secção 4.5.4. Como se pode ver na Figura 69, se não for encontrado o ponto de controlo depois de se afastar 60 metros do cruzamento então será emitido um aviso sonoro, será enviada uma mensagem ao utilizador indicando que deverá fazer inversão de marcha e será iniciado o processo de correção da rota.

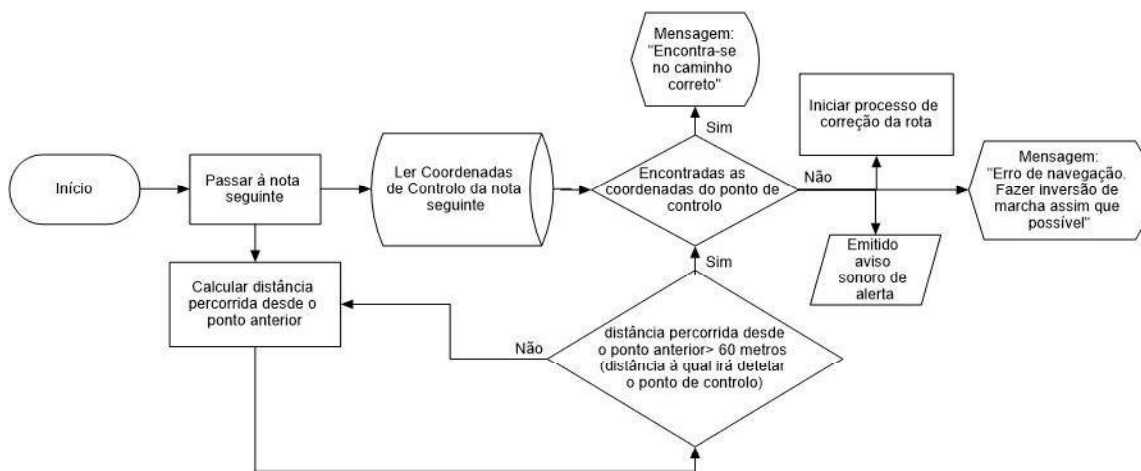


Figura 69 - Ativação do processo de correção da rota

Foi definida uma distância de 60 metros pois quando da gravação, o ponto de controlo foi gravado quando o utilizador que gravou as notas se afastou 40 metros de cada cruzamento guardado. Desta forma, se na navegação se afastar 60 metros e não for encontrado o ponto de controlo conclui-se que o caminho escolhido foi o errado. Foram escolhidos estes valores com base em testes

de desempenho realizados no terreno (secção 5.1), em que foram usadas diversas medidas, sendo que estas demonstraram permitir obter bons resultados. Se dois cruzamentos estiverem tão perto um do outro que não exista ponto de controlo irá passar-se logo para a nota seguinte e não vai ser verificado se se está a percorrer o caminho correto, pois é assumido que se chegou ao cruzamento seguinte e a navegação está correta. O fluxograma completo do módulo de navegação de *roadbooks* pode ser consultado no Anexo C.

## 4.6 Processo de correção de rota

Depois de iniciado o processo de correção da rota o utilizador poderá escolher entre fazer a correção usando um mapa (Figura 70) ou usando uma seta indicadora do destino (Figura 71).

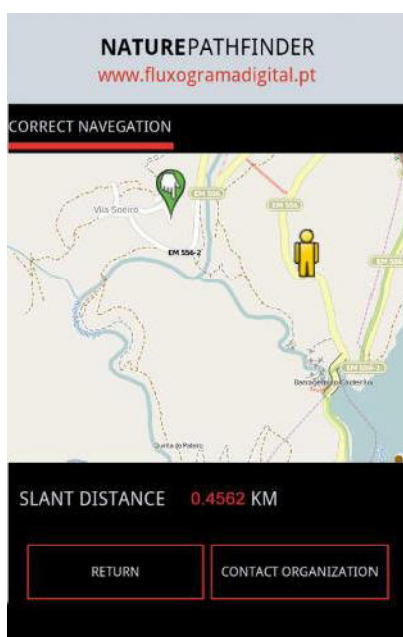


Figura 70 - Correção de rota usando o mapa

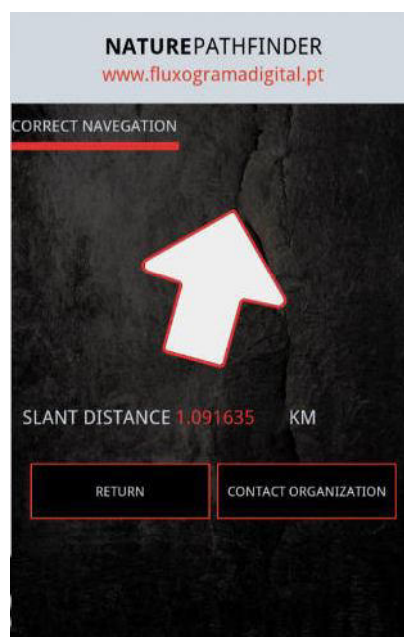


Figura 71 - Correção de rota usando a seta

Se for escolhida a opção do mapa, irá ser apresentado um mapa onde se poderá ver a localização dos dispositivo móvel e do ponto de controlo (que indica o caminho correto). Usando a informação presente neste mapa, o utilizador deverá deslocar-se na direção do caminho correto, sendo que a sua localização irá mudar no mapa à medida que se for deslocando. Se for escolhida a opção da seta, a navegação será feita tendo como meio de orientação a referida seta que estará sempre a

apontar para a localização do ponto de controlo. Ao mesmo tempo, o utilizador tem sempre disponível a distância em linha reta ao caminho correto. Após o utilizador ter feito inversão de marcha e iniciado o processo de correção, será calculada a distância em linha reta até ao ponto de controlo, usando a mesma técnica descrita na secção 4.5.4. Quando é detetado que a distância até ao ponto de controlo é menor que a distância mínima para deteção do ponto, isto significa que o utilizador se encontra no caminho correto e então passa-se de novo para a navegação normal do *roadbook* (Figura 72).

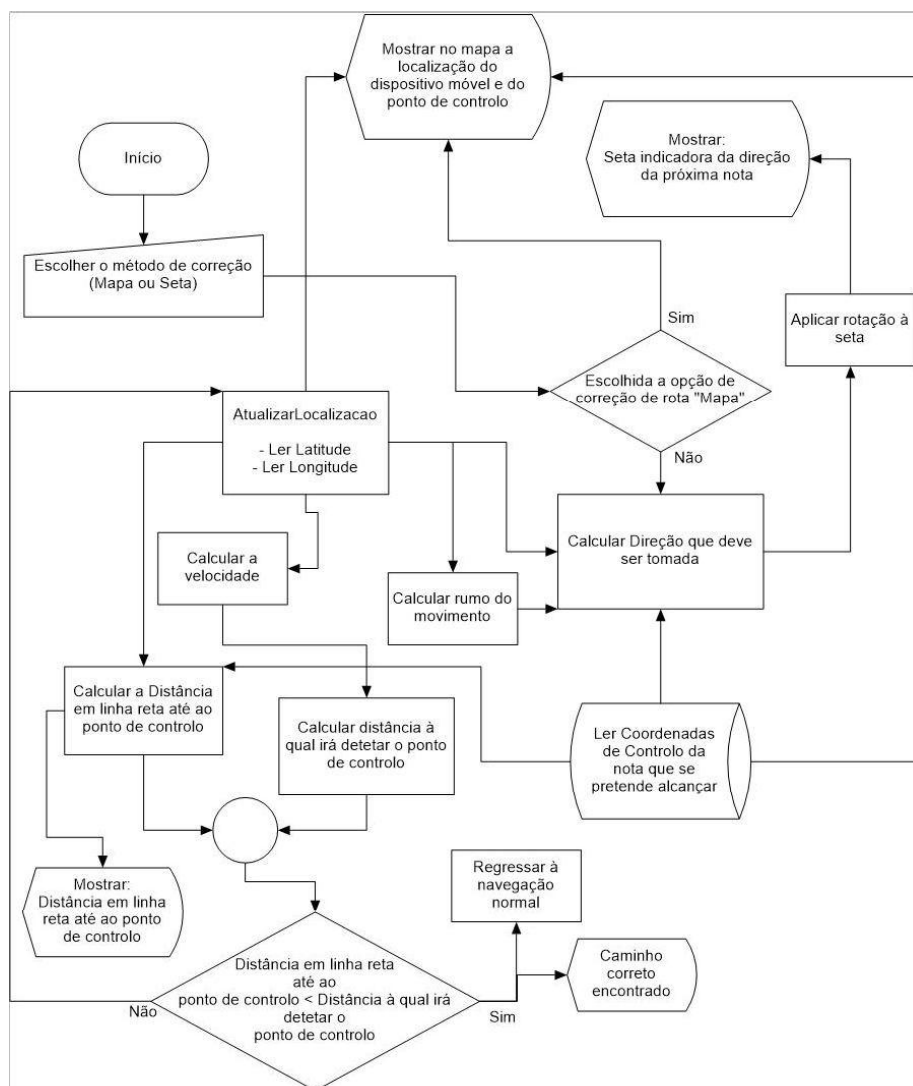


Figura 72 - Fluxograma do processo de correção da rota

Apesar de ser usada a mesma técnica para deteção do ponto de controlo em ambas as abordagens, o método para guiar o utilizador para o caminho correto é diferente, como se explica nas secções seguintes.

### 4.6.1 Processo de correção de rota utilizando a seta

No que respeita à rotação da seta para a orientação na direção do ponto de controlo foi utilizado o mesmo processo que foi descrito na secção 4.5.2, em que é calculado o rumo atual e em conjunto com a direção entre a posição atual e a posição do ponto alvo (ponto de controlo) é calculado o ângulo de rotação da seta.

### 4.6.2 Processo de correção de rota utilizando mapa

O conceito fundamental deste algoritmo de correção da rota assenta na disponibilização no ecrã do mapa, onde se pode ver a localização do dispositivo móvel assim como a localização do ponto de controlo, que indica o caminho correto. Foram utilizados mapas *OpenStreetMaps*<sup>54</sup>. Este tipo de mapas é criado por uma comunidade que pretende fazer o mapeamento de todo o mundo. Os dados dos mapas são do tipo aberto, sob a licença Open Data Commons Open Database License<sup>55</sup>, pelo que os dados podem ser copiados, distribuídos e transmitidos, sempre que se faça referência à OpenStreetMaps. Este projeto é gerido pela OpenStreetMap Foundation<sup>56</sup>.

Para a utilização de mapas OpenStreetMaps na aplicação *Android* foram utilizadas as ferramentas *osmdroid*<sup>57</sup> que permitem a criação da interação com os mapas. Para a utilização desta ferramenta foi necessário instalar a biblioteca *osmdroid-android*<sup>58</sup>, que permite ter acesso e manipular os referidos mapas, assim como a biblioteca *slf4j-android*<sup>59</sup>, que tem como objetivo simplificar a ligação entre o Java e os *osmdroid*.

---

<sup>54</sup> <https://www.openstreetmap.org/#map=12/40.5141/-7.2664&layers=N>

<sup>55</sup> <http://opendatacommons.org/licenses/odbl/>

<sup>56</sup> [http://wiki.osmfoundation.org/wiki/Main\\_Page](http://wiki.osmfoundation.org/wiki/Main_Page)

<sup>57</sup> <https://github.com/osmdroid/osmdroid>

<sup>58</sup> <https://code.google.com/p/osmdroid/downloads/detail?name=osmdroid-android-3.0.8.jar>

<sup>59</sup> <http://www.slf4j.org/android/>

As ferramentas *osmdroid* utilizam mapas definidos por um conjunto de *Tiles*<sup>60</sup>, cada um deles composto por uma imagem PNG com dimensão típica de 256 x 256 pixéis. No entanto, para apresentação de mapas em dispositivos móveis poderão ter dimensões de 64 x 64 pixéis. Cada uma destas imagens representa uma secção do mapa. Na Figura 73 pode ver-se um exemplo de uma destas imagens.



Figura 73 - Exemplo de uma Tile<sup>61</sup>

Vai ser o conjunto de uma grande quantidade destas imagens que depois irá formar o mapa completo, havendo a possibilidade de existirem 19 níveis de *zoom*.

Usualmente, para que o mapa seja apresentado, deve existir conexão à internet para que a aplicação possa aceder a um servidor de *Tiles* e possa fazer o *download* da zona em que foi detetado que o utilizador se encontra. Cada servidor possui *Tiles* de mapas com características diferentes.

O Servidor *standard*, associado ao OpenStreetMap Project é o *Mapnik*<sup>62</sup>, que procura mostrar um mapa geral, com objetos e indicações genéricas que não confundam muito o utilizador. Este é o servidor mais utilizado.

Mas para além deste existem outros servidores, mais específicos, que se focam em determinadas informações. Um servidor muito semelhante ao *Mapnik* é o *Mapquest*<sup>63</sup>, que já apresenta

---

<sup>60</sup> <http://wiki.openstreetmap.org/wiki/Tiles>

<sup>61</sup> Retirado de: <http://wiki.openstreetmap.org/wiki/Tiles>

<sup>62</sup> <http://www.openstreetmap.org/#map=14/40.5463/-7.2424&layers=Q>

menos informação. Na Figura 74 pode ver-se a comparação entre um mapa *Mapnik* e um *Mapquest*. Pode ver-se que são bastante semelhantes, no entanto o *Mapnik* mostra mais informação, enquanto no *Mapquest* se conseguem ver melhor as estradas, pois está mais “limpo” de informações superfluas.

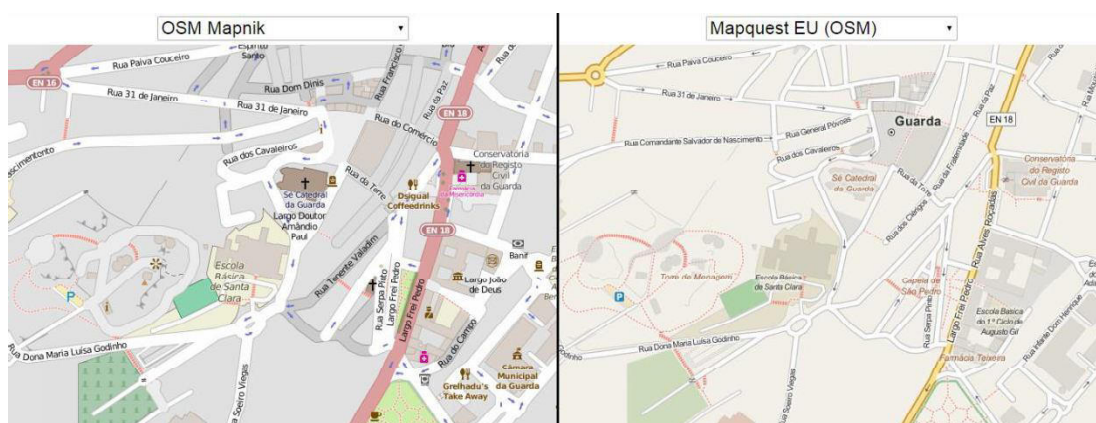


Figura 74 - Comparação entre Mapnik e Mapquest<sup>64</sup>

A *Mapquest* possui também um tipo de mapa com vista em modo satélite<sup>65</sup>, que poderia ser ideal para aplicações com navegação em caminhos de terra batida, no entanto como se pode ver na Figura 75, estes *Tiles* possuem poucos níveis de detalhe.



Figura 75 - Tiles Mapquest Satellite

<sup>63</sup> <http://www.openstreetmap.org/#map=14/40.5445/-7.2410&layers=Q>

<sup>64</sup> Comparação realizada através do site:

<http://mc.bbbike.org/mc/?num=2&mt0=mapnik&mt1=mapquest-eu>

<sup>65</sup> <http://www.mapquest.com/satellite-maps/>

Para além destes três, existem muitos outros servidores de *Tiles*, cada um com as suas características próprias<sup>66</sup>, no entanto o *Mapnik* e *Mapquest* são os mais divulgados e utilizados. Todos os servidores baseados em OpenStreetMaps utilizam a mesma informação, mas os *Tiles* estão armazenados em servidores diferentes.

Como os *Mapnik* (tiles oficiais OpenStreetMaps) são os mais utilizados, este uso excessivo de recursos tem posto em causa a capacidade para os colaboradores editarem os mapas, devido à largura de banda suportada pelos servidores. Devido a este facto, foram criados requisitos mínimos que os utilizadores devem possuir para utilizarem as *Tiles* do *Mapnik*, tais como limitar o acesso a aplicações que façam uso excessivo dos mapas, necessidade de criação de um *Valid User Agent*, proibição de *download* excessivo de mapas para uso offline especialmente com níveis de zoom elevado.

Devido a estes fatores e a estas limitações, optou-se pela utilização de um servidor de *Tiles* *Mapquest*, na medida que os seus *Tiles* são bastante semelhantes aos do *Mapnik*, sendo que possuem um apeto mais simples e de melhor leitura e não têm tantas limitações na utilização<sup>67</sup>. Deve também ter-se em conta que a utilização de mapas na aplicação desenvolvida apenas será feita durante a correção da rota e apenas no caso de o utilizador escolher essa opção, pois a aplicação pode ser utilizada sem o recurso a mapas.

#### 4.6.2.1 Mapas offline

Na aplicação implementada, à semelhança com o que acontece com muitas aplicações que utilizam mapas OpenStreetMaps, é também possível utilizar mapas *offline*, tendo o utilizador que fazer a instalação manual dos mapas. Para isso basta que se faça a cópia dos *Tiles* pretendidos para a pasta *osmdroid*, no dispositivo móvel.

Para o *download* de mapas OpenStreetMaps para uso *offline*, poderá ser utilizada uma aplicação como o OSMtiledownloader<sup>68</sup>, que permite que se faça o download de *Tiles* específicos a partir de uma zona seleccionada no mapa online. Pode ver-se na Figura 76 a interface desta aplicação durante o processo de *download* de mapas.

---

<sup>66</sup> <http://wiki.openstreetmap.org/wiki/Tiles>

<sup>67</sup> [http://wiki.openstreetmap.org/wiki/Tile\\_usage\\_policy](http://wiki.openstreetmap.org/wiki/Tile_usage_policy)

<sup>68</sup> <http://wiki.openstreetmap.org/wiki/OSMtiledownloader>



Outra aplicação (mais completa que a anterior) que pode ser usada para o *download* de *Tiles* OpenStreetMaps é a Mobile Atlas Creator<sup>69</sup>. Trata-se de *software open source* que possibilita o *download* de mapas que poderão ser usados em aplicações móveis. A aplicação referida permite o *download* de *Tiles* PNG do OpenStreetMaps.

A utilização deste tipo de *software* é bastante simples. Começa-se por se definir a fonte dos mapas, como se pode ver na Figura 76.



Figura 76 - Map Source no Mobile Atlas Creator

Depois de escolhida a fonte dos mapas pode definir-se o nível de zoom dos mesmo (Figura 77), sendo que quanto maior for o nível de zoom, mais detalhe vai ter o mapas, mas também mais vão ser os ficheiros PNG dos *Tiles*, pelo que mais espaço irá ser ocupado.

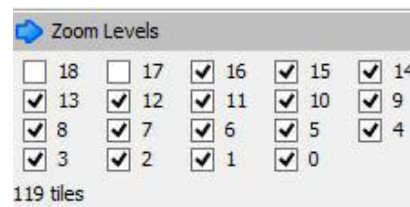


Figura 77 - Zoom levels no Mobile Atlas Creator

<sup>69</sup> <http://mobac.sourceforge.net/>

Depois de definidas estas configurações basta criar o mapa. Nesta fase será feito o *download* de todos os *Tiles* tendo em conta a fonte escolhida, assim como o nível de zoom definido. Depois de feito o *download*, tal como referido anteriormente, basta copiar a pasta dos *Tiles* para a pasta *osmdroid* e os mapas ficarão disponíveis quando a aplicação for executada. Se houver mapas na pasta *osmdroid* estes serão mostrados, senão houver mapas e se houver conexão à internet será feito o *download* dos *Tiles* da zona onde o utilizador se encontra e estes serão mostrados no ecrã.

#### 4.6.2.2 Implementação da navegação por mapa

Para a implementação da navegação utilizando o mapa foi utilizada a classe *MapView*, que faz parte da biblioteca *osmdroid*. Esta classe vai permitir a definição da fonte de *Tiles* para que assim seja mostrado o mapa através do método *setTileSource*. No caso da aplicação desenvolvida, tal como referido anteriormente na secção 4.6.2, foi usado o servidor de *Tiles Mapquest*. Como se pode ver na Figura 78, vai ser usado o componente *MapView*, onde vai ser apresentado o mapa.

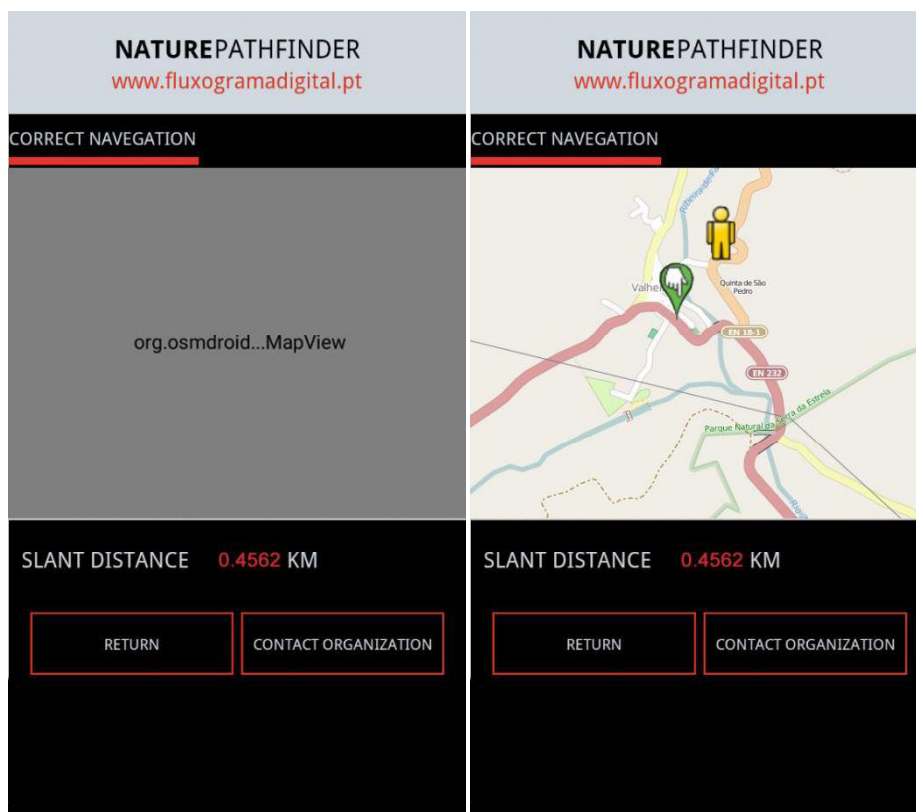


Figura 78 - Utilização do MapView

Esta classe *MapView* possui métodos de controlo do mapa, nomeadamente do controlo do zoom que é possível aplicar-se ao mapa, com o método *setBuiltInZoomControls*.

Depois de mostrado o mapa, foi adicionado um marcador ao mapa (Figura 79), representando a localização do ponto de controlo do caminho correto, através da classe *OverlayItem* usando as coordenadas latitude e longitude do referido ponto.

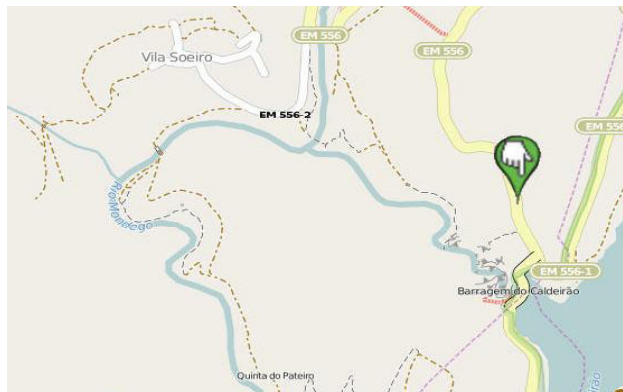


Figura 79 - Marcador do Ponto de Controlo no mapa

Para se adicionar a localização do utilizador no mapa é usada a classe *MyLocationOverlay*, que caso o utilizador se encontre parado será mostrada uma imagem que representa uma pessoa e caso o dispositivo se encontre em movimento será mostrada a imagem de uma seta apontando a direção do movimento, tal como se pode ver na Figura 80.

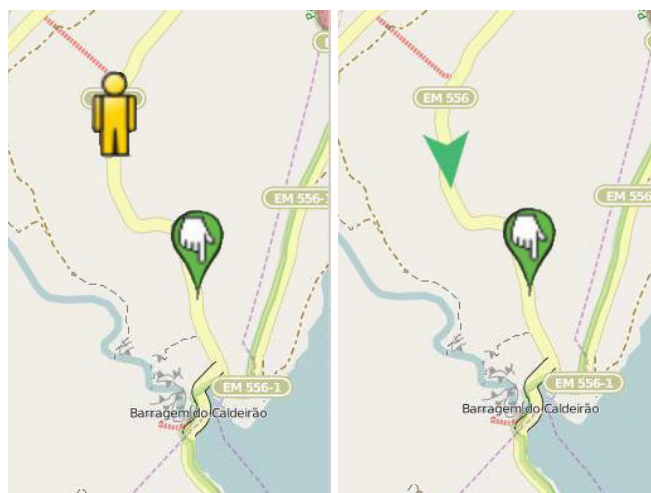


Figura 80 - Marcador do utilizador no mapa

Através do marcador do ponto de controlo e do marcador da localização do dispositivo no mapa, o utilizador poderá deslocar-se até que os marcadores se encontrem e, tal como descrito na secção 4.6, encontrar o caminho correto e retomar a navegação normal. Ao ser encontrado o ponto de controlo a navegação normal é retomada automaticamente.

## 4.7 Detecção de Contextos

Pode definir-se contexto como sendo as condições de interdependência nas quais alguma coisa existe ou acontece. No entanto, partindo da afirmação anterior, pode dizer-se que a definição de contexto pode ser considerada um pouco vaga, na medida que tudo no mundo acontece num determinado contexto. No entanto, segundo (Kotz, 2000), pode dizer-se que contexto são todas as situações que nos rodeiam e que geram informação que pode ser utilizada para caracterizar o estado de uma entidade. Esta informação pode depois ser usada, por exemplo, por uma aplicação interagir com o utilizador ou com o meio envolvente. Quanto mais informação estiver disponível, maior será o conhecimento que existe acerca do meio envolvente e mais precisa poderá ser a interação com ele.

Numa aplicação com uma metodologia de deteção de contextos, sempre que houver uma mudança no ambiente em que a aplicação está a ser executada, esta deve ter conhecimento do ambiente que a rodeia e adaptar-se a essas mudanças de forma automática. No que se refere a uma aplicação para dispositivos móveis, pode definir-se contexto como sendo o conhecimento sobre o utilizador e o estado do dispositivo móvel, incluindo o ambiente à sua volta e a sua localização (Schmidt, 1999). Através da combinação de diversos parâmetros do contexto em que se encontra o utilizador, poderão ser criados comportamentos mais complexos para uma determinada aplicação. Por exemplo, numa aplicação em que se saiba a localização do dispositivo, a hora atual e informação sobre a agenda do utilizador, a aplicação poderá saber o que está o utilizador a fazer e adaptar-se a essa situação (Kotz, 2000). Se, por exemplo, o utilizador estiver numa reunião a aplicação poderá automaticamente desligar o som do dispositivo móvel. Segundo Kotz (2000), pode dividir-se deteção de contextos em duas categorias *Active Context Awareness* e *Passive Context Awareness*. Na Detecção de Contextos Ativa uma aplicação adapta-se automaticamente ao novo contexto do ambiente, alterando um comportamento. Na Detecção de Contextos Passiva, a aplicação apresenta ao utilizador as novas alterações que ocorreram no contexto e deixa à descrição do utilizador a adaptação às mesmas.

### 4.7.1 Detecção de Contextos na aplicação desenvolvida

A aplicação desenvolvida é uma aplicação que deteta contextos e se adapta ao ambiente de acordo com as necessidades.

Na secção 4.4.2.1, foi descrito que a aplicação adapta automaticamente a frequência de leituras a partir do sensor de *GPS* com o objetivo da poupança de energia. Esta adaptação é feita tendo em conta uma alteração no contexto da aplicação, que é a velocidade a que o utilizador se está a deslocar. Neste caso trata-se de deteção de contextos de uma forma ativa.

Aquando da gravação das notas, quando o utilizador clica no botão *Save Waypoint* para registar os dados de uma nota específica, a aplicação irá analisar a velocidade atual e apenas vai permitir a gravação dos mesmos se a velocidade for reduzida, como explicado na secção 4.4.3. Esta situação trata-se de uma utilização passiva da deteção de contextos, na medida que ao detetar que a velocidade não é a indicada para a gravação das notas, a aplicação informa o utilizador dessa situação, cabendo ao utilizador fazer a respetiva adaptação à situação atual, reduzindo a velocidade. Mesmo que o utilizador se desloque dentro da velocidade permitida para que seja gravada a nota, a aplicação irá detetar se o utilizador está em movimento ou efetivamente parado. Se estiver mesmo parado irá gravar uma coordenada *GPS*, se estiver em movimento (mesmo que lentamente) irá gravar 3 coordenadas e depois fazer a média entre estas leituras, para que consiga obter um ponto de coordenado mais preciso e central.

Também aquando da gravação do ponto de controlo, descrito na secção 4.4.5, a aplicação terá também de saber qual a distância percorrida depois do cruzamento de forma a saber em que altura deverá guardar as coordenadas do ponto de controlo. Neste caso trata-se de uma deteção de contextos ativa, sendo que neste caso o utilizador não se chega a aperceber quais as ações que foram tomadas nesta adaptação ao ambiente, nem em que circunstâncias.

Também aquando da navegação de um *roadbook* existe deteção de contextos. Nesta situação irá ser feita a adaptação da frequência de leituras efetuadas pelo sensor do *GPS*, tendo em conta a distância a que se encontra o próximo cruzamento, para que se possa poupar energia do dispositivo móvel. Quanto mais longe o utilizador estiver do próximo cruzamento menos leituras irá fazer o sensor de *GPS*.

Outro exemplo da utilização de detecção de contextos é o facto de a todo o momento a aplicação saber em que direcção fica o próximo cruzamento e indicar essa direcção ao utilizador através de uma seta.

Através do valor da exatidão do sinal *GPS*, a aplicação vai decidir se a navegação pode ser automática ou tem de ser manual, no caso de a exatidão do sinal ser reduzida.

A aplicação adapta a distância à qual são detetados os cruzamentos tendo em conta a velocidade, sendo este outro exemplo de detecção de contextos. Também ao se chegar a um cruzamento a aplicação irá ter perceção disso e mudar automaticamente para as informações da nota seguinte.

Depois de passado um cruzamento a aplicação irá analisar se o utilizador seguiu o caminho correto e se assim não for irá iniciar automaticamente o processo de correção da rota.

Tal como foi descrito na secção 4.6, depois de haver um erro de navegação o utilizador pode optar por um de dois métodos para correção da rota: o mapa ou a seta. Optando pela técnica do mapa, a aplicação sabe sempre qual a posição do utilizador e do caminho correto mostrando ambas estas localizações no mapa. Se for escolhida a utilização da seta como técnica de correção da rota, a aplicação sabe em que direcção fica o caminho correto e aponta a seta sempre nessa direcção. Em ambos os casos a aplicação irá detetar quando o utilizador se encontra no caminho correto e passar automaticamente para a navegação normal do *roadbook*.

Pode dizer-se que a maior parte dos comportamentos da aplicação aquando da gravação das notas e especialmente aquando da navegação de um *roadbook* são automáticos e baseados no meio envolvente, fazendo com que o utilizador se consiga concentrar no processo de identificar os desenhos das notas dos cruzamentos e tente encontrar o caminho correto, tornando esta aplicação num assistente de criação e utilização de *roadbooks* com detecção de contextos.

## 4.8 Servidor *web* para partilha de *roadbooks*

Utilizando a aplicação *Naturepathfinder* é possível partilhar *roadbooks* na *web*, utilizando uma base de dados *MySQL* que foi criada num servidor (como descrito na secção 4.2.2). Da mesma forma é possível o utilizador fazer *download* para o dispositivo móvel de *roadbooks* criados por outros utilizadores.

### 4.8.1 Envio de dados do dispositivo móvel para a *web*

Quando se acede ao menu de partilha de *roadbooks*, representado na Figura 81, é possível saber-se se todos os *roadbooks* gravados no dispositivo móvel já foram enviados para a *web* ou se há possibilidade de os enviar. No caso específico da imagem referida pode ver-se que todos os *roadbooks* já foram enviados para a *web*. Caso houvesse *roadbooks* que ainda não tivessem sido enviados eles iriam aparecer listados e bastava clicar-se no botão *Share roadbooks* para que estes fossem enviados para a *web*. Como foi descrito na secção 4.2.1, nas tabelas *roadbook*, *organizacao*, *nota*, e *esquema* da base de dados *SQLite* existem os campos *updatestatus\_roadbook*, *updatestatus\_organizacao*, *updatestatus\_nota* e *updatestatus\_esquema* respetivamente. Ao ser criado um novo registo é atribuído a estes campos o valor “no”, ficando assim identificado que este registo ainda não foi enviado para a *web*. Depois de ter sido enviado para a *web* o valor deste campo é atualizado, passando a “yes”.

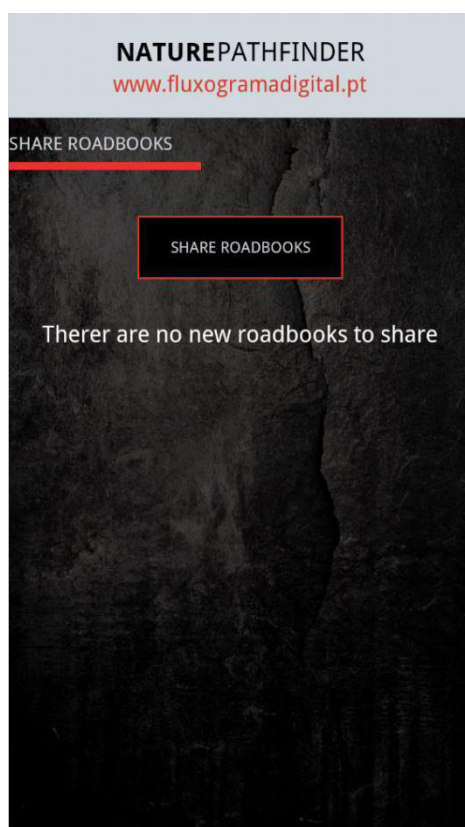


Figura 81 - Interface para partilha de roadbooks

Desta forma, tal como se pode ver no fluxograma da Figura 82, quando o utilizador acede a este menu é realizada uma consulta à base de dados e caso algum registo não tenha sido enviado surge o respetivo aviso e é permitido ao utilizador enviar os dados da sua base de dados para a *web*, clicando no botão *Share Roadbooks*.

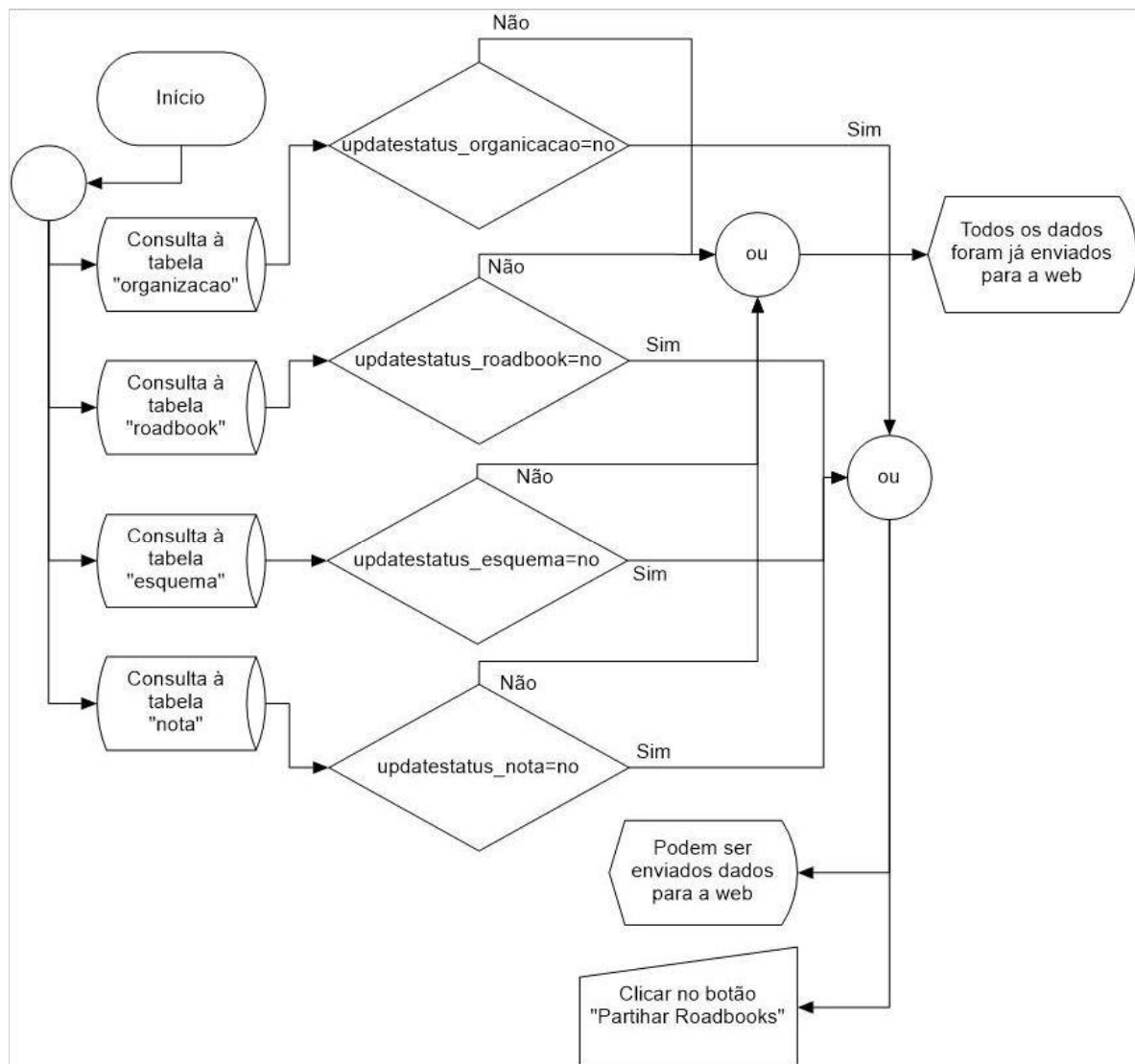


Figura 82 - Verificar se existem dados no dispositivo que podem ser enviados para a web

Depois de clicado o botão *Share Roadbooks*, serão então enviados os dados para a *web*. Passa-se de seguida a explicar como é feita esta ligação entre a aplicação *Android* e a base de dados *MySQL* na *web*.



### 4.8.1.1 Ligação entre a aplicação e a base de dados *MySQL* para envio de dados

Como tinha sido explicado na secção 4.2.2, foi criada a base de dados *MySQL* num alojamento *web* com as tabelas *organizacao*, *roadbook*, *nota* e *esquema*. Depois de as tabelas estarem criadas foram criadas as classes *PHP* que irão ser responsáveis por fazer a ligação entre a aplicação e a base de dados *MySQL* (Tabela 11).

Tabela 11 - Classes *PHP* para ligação entre a aplicação e a base de dados *MySQL*

Classes <i>PHP</i>	Objetivo
config.php	Classe para a configuração de variáveis, como o nome da base de dados, <i>password</i> e <i>web host</i> .
db_connect.php	Classe com os métodos para realizar e terminar a ligação à base de dados <i>MySQL</i> .
db_functions.php	Classe com os métodos para efetuar ações sobre a base de dados <i>MySQL</i> , nomeadamente inserir registos e fazer consultas.
insert_esquema.php	Classe que vai ser acedida a partir da aplicação <i>Android</i> para inserção dos registos da tabela <i>esquema</i> na base de dados <i>MySQL</i> .
insert_nota.php	Classe que vai ser acedida a partir da aplicação <i>Android</i> para inserção dos registos da tabela <i>nota</i> na base de dados <i>MySQL</i> .
insert_organizacao.php	Classe que vai ser acedida a partir da aplicação <i>Android</i> para inserção dos registos da tabela <i>organizacao</i> na base de dados <i>MySQL</i> .
insert_roadbook.php	Classe que vai ser acedida a partir da aplicação <i>Android</i> para inserção dos registos da tabela <i>roadbook</i> na base de dados <i>MySQL</i> .

Será nas classes *PHP* *insert\_esquema*, *insert\_nota*, *insert\_organizacao* e *insert\_roadbook* que vai ser usado o formato de dados *JSON*<sup>70</sup> para se fazer o intercâmbio de informação entre a aplicação *Android* e a base de dados *MySQL*, através do envio do nome do campo e do valor do registo respetivo. Todas as classes *PHP* descritas na Tabela 11 serão colocadas no servidor *web*.

<sup>70</sup> <http://json.org/>

Como foi referido, apenas serão enviados os dados dos *roadbooks* que ainda não tenham sido enviados anteriormente, controlando-se o campo *updatestatus* de cada registo. Esta verificação é feita na classe Java *Sinc\_NPFinder\_SmartWeb\_DBController*, que faz parte do projeto da própria aplicação e que é responsável por gerir a interação com os dados em formato *JSON* e a ligação e passagem de dados da base de dados *SQLite* no dispositivo móvel e a base de dados *MySQL* na *web*. É então feita uma consulta para se verificar que dados podem ser enviados e é construída a estrutura de dados *JSON* que contém os dados a passar para a base de dados *MySQL*, relativos a cada uma das tabelas.

Estes dados serão então enviados para a classe *PHP* respetiva, dependendo de que tabela se pretende atualizar.

Já do lado da classe *PHP*, na *web*, os dados em formato *JSON* são decodificados em arrays *PHP* e inseridos na base de dados *MySQL* usando a classe *db\_functions*.

O processo descrito para se efetuar passagem de dados da base de dados *SQLite* no dispositivo móvel para a base de dados *MySQL* na *web* está representado no fluxograma da Figura 83.

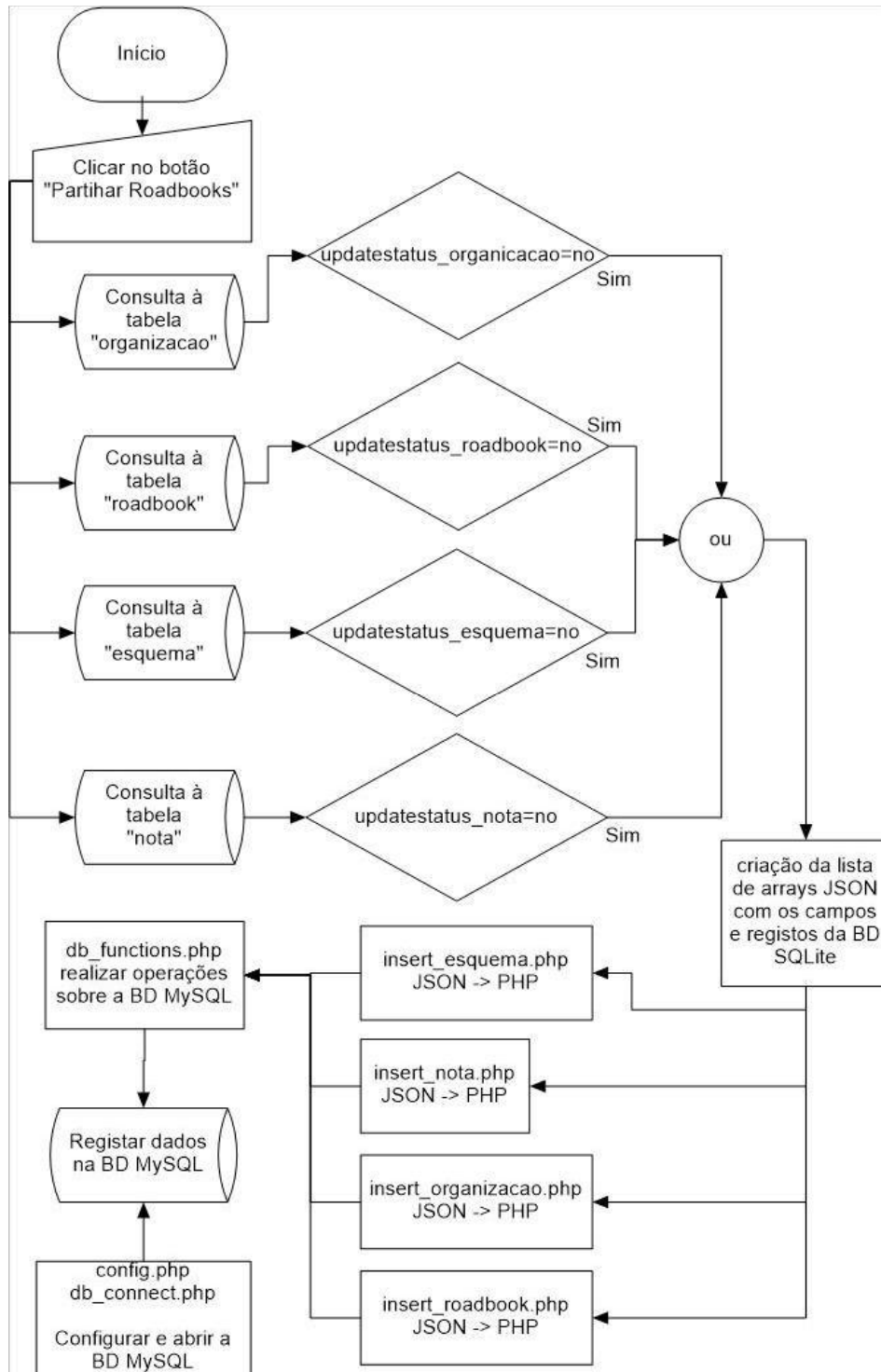


Figura 83 - Passagem de dados entre a base de dados SQLite e a base de dados MySQL

## 4.8.2 *Download* de dados da *web* para o dispositivo móvel

Quando o utilizador acede ao menu de *download* de *roadbooks*, representado na Figura 84, tem à sua disposição dois botões. Um permite verificar quais os *roadbooks* que se encontram no dispositivo móvel (*Roadbooks list in the device*) e outro permite consultar quais os *roadbooks* que se encontram na base de dados na *web* (*Roadbook list in the web*). Se se clicar neste último botão será preenchida uma lista com os nomes de todos os *roadbooks* disponíveis para *download* e se o utilizador clicar no nome de um dos *roadbooks* que se encontram na *web* será feito o *download* do mesmo para o dispositivo móvel.

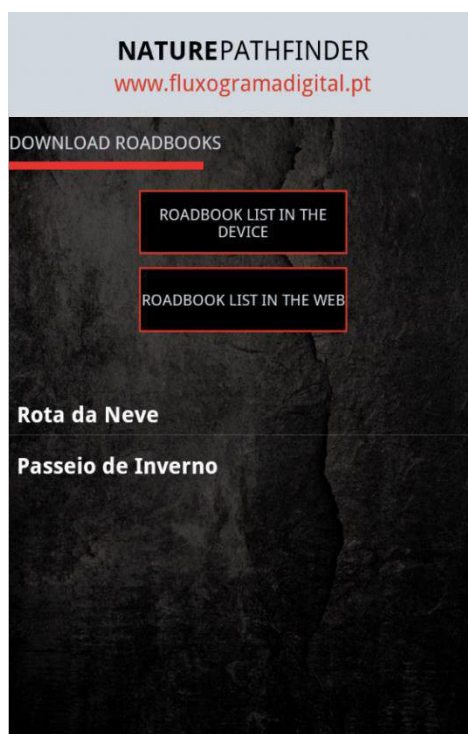


Figura 84 - *Download* de *roadbooks* a partir da *Web*

Depois de criada a base de dados *MySQL*, tal como descrito na secção 4.2.2 foram criadas classes *PHP* com o objetivo de realizarem a ligação entre a base de dados *MySQL* e a aplicação *Android*. Na Figura 12 pode ver-se a descrição de todas as classes *PHP* que foram criadas.

Tabela 12 - Classes PHP para ligação entre a base de dados MySQL e a aplicação

Classes PHP	Objetivo
config.php	Classe para configuração de variáveis, como o nome da base de dados, password e <i>web host</i>
db_connect.php	Classe com os métodos para realizar e terminar a ligação à base de dados Mysql
db_functions.php	Classe com os métodos para efetuar ações sobre a base de dados <i>MySQL</i> , nomeadamente fazer consultas
getAllroadbook.php	Classe responsável por criar a lista de arrays <i>JSON</i> que vão ser usados pela classe java, a partir da tabela <i>roadbook</i> da base de dados <i>MySQL</i>
get_Dados_roadbook_id_selecionado	Classe responsável por criar a lista de arrays <i>JSON</i> a partir dos dados da tabela <i>roadbook</i> da base de dados <i>MySQL</i> que correspondam ao <i>roadbook</i> escolhido para <i>download</i>
get_Dados_Organizacao_id_RB_selecionado	Classe responsável por criar a lista de arrays <i>JSON</i> a partir dos dados da tabela <i>organizacao</i> da base de dados <i>MySQL</i> que correspondam ao <i>roadbook</i> escolhido para <i>download</i>
get_Dados_Esquema_id_RB_selecionado	Classe responsável por criar a lista de arrays <i>JSON</i> a partir dos dados da tabela <i>esquema</i> da base de dados <i>MySQL</i> que correspondam ao <i>roadbook</i> escolhido para <i>download</i>
get_Dados_Nota_id_RB_selecionado	Classe responsável por criar a lista de arrays <i>JSON</i> a partir dos dados da tabela <i>nota</i> da base de dados <i>MySQL</i> que correspondam ao <i>roadbook</i> escolhido para <i>download</i>

Ao ser clicado o botão que permite consultar os *roadbooks* que se encontram guardados na base de dados *MySQL*, é utilizada a classe PHP *getAllroadbook*, que vai criar a lista de arrays

JSON a partir dos dados da tabela *roadbook*. Esta informação vai ser interpretada pela aplicação *Android* que cria uma lista com todos os nomes dos *roadbooks* e mostra-os no *layout* da aplicação. Tal como referido anteriormente, é possível clicar nos objetos desta lista, para se escolher o *roadbook* que se pretende copiar para o dispositivo móvel.

Nesta etapa, a aplicação efetua outro pedido à base de dados *MySQL*, tendo como referência a chave primária do *roadbook* que o utilizador escolheu da lista. Para tal, vão ser usadas as classes *PHP* *get\_Dados\_roadbook\_id\_selecionado*, *get\_Dados\_Organizacao\_id\_RB\_selecionado*, *get\_Dados\_Eschema\_id\_RB\_selecionado* e *get\_Dados\_Nota\_id\_RB\_selecionado*, que vão criar os objetos *JSON* com os dados de cada uma das tabelas. Estes dados vão ser interpretados pela aplicação *Android* que os vai usar para inserir na base de dados *SQLite*.

O processo descrito para se efetuar passagem de dados da base de dados *MySQL* na *web* para a base de dados *SQLite* no dispositivo móvel está representado no fluxograma da Figura 85.

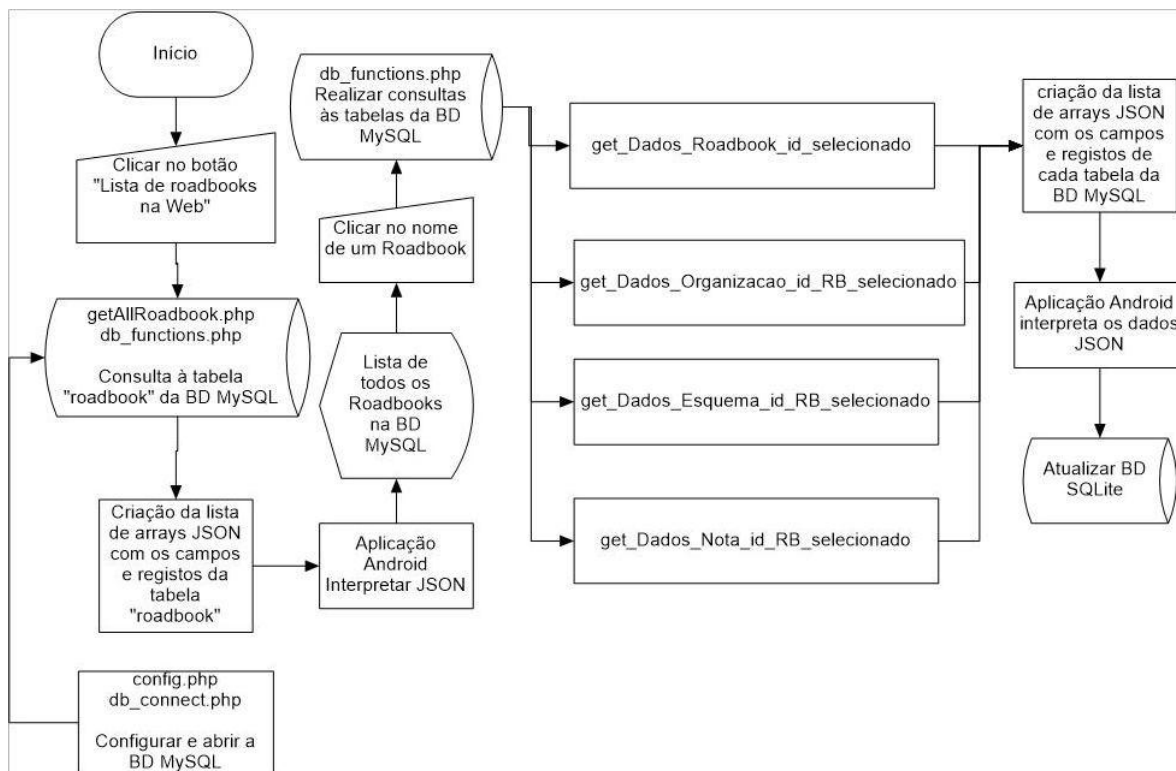


Figura 85 - Passagem de dados entre a base de dados *MySQL* e a base de dados *SQLite*

## 4.9 Interface da aplicação

No que respeita à criação da interface da aplicação tiveram de se ter em conta dois fatores que influenciam a forma como as interfaces são apresentadas num dispositivo móvel: a densidade de pixéis e os diferentes tamanhos dos ecrãs. Para além desses fatores, considerou-se importante organizar os textos dos títulos, conteúdos e botões, que fazem parte da interface, em ficheiros de recursos próprios, de forma a permitir a criação de estilos de texto específicos, assim como a definição dos idiomas português e inglês<sup>71</sup>.

### 4.9.1 Densidade de pixéis

Diferentes dispositivos móveis *Android* têm diferentes densidades de pixéis independentemente do tamanho de ecrã que possuem. A unidade de medida da densidade de pixéis é o dpi (*dots-per-inch*). Esta medida define quantos pixéis vão caber dentro de uma polegada no ecrã do dispositivo (Figura 86).

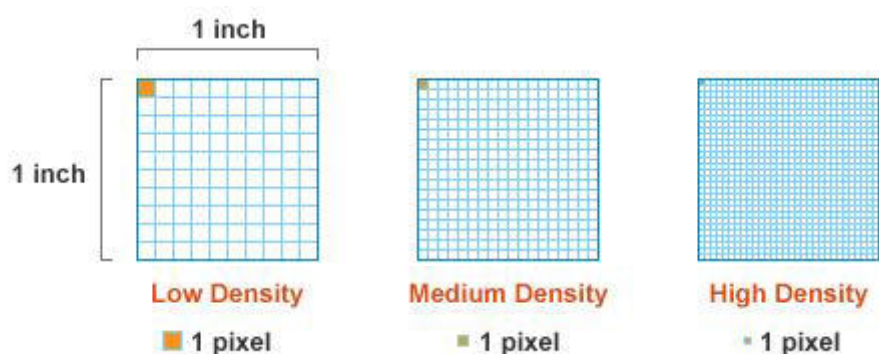


Figura 86 - Densidade de pixéis<sup>72</sup>

Quanto maior for a densidade de pixéis mais perfeitos serão os gráficos apresentados, mantendo as imagens com o mesmo tamanho, como se pode ver na Figura 87.

<sup>71</sup> [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)

<sup>72</sup> Retirado de: <https://laaptu.wordpress.com/2013/12/12/android-view-basics-coordinatesmarginpaddingdipx/>

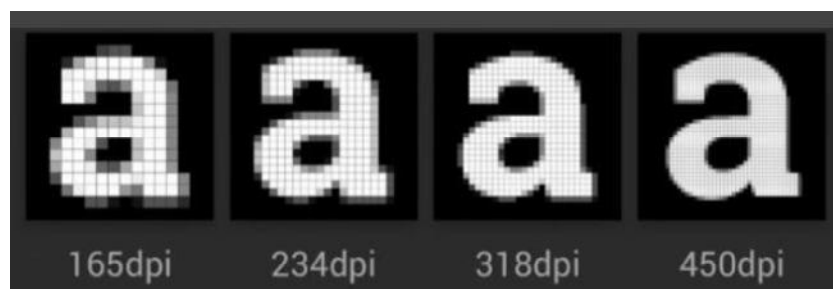


Figura 87 - Exemplo de visualização da mesma imagem em ecrãs com diferentes densidades de pixels<sup>73</sup>

Uma interface desenvolvida com um determinado valor de pixéis (em valores absolutos) pode ser apresentada de maneira diferente em diferentes dispositivos. A plataforma *Android* define uma lista de densidades de base, chamadas *Logical Pixel Densities*, que servem como valores de referência para a criação dos *layouts*, de acordo com diversas densidades de pixéis. Esta lista pode ser consultada na Tabela 13.

Tabela 13 - *Logical Pixel Densities*

Densidade de Pixéis (DPI)	Nome	Escala
160	MDPI	1x
240	HDPI	1.5x
320	XHDPI	2x
480	XXHDPI	3x

Nesta tabela estão representados valores de referência para a criação dos *layouts*, para que apenas seja necessário ter-se em atenção estes valores e não valores intermédios. No que respeita ao valor 160 dpi trata-se do valor base que servirá de referência para todos os outros valores.

---

<sup>73</sup> Retirado de: <http://blog.raffaeu.com/archive/2015/03/04/understand-density-independent-pixels-dpi.aspx>



#### 4.9.1.1 *Density-Independent Pixels – dp*

Trata-se de uma medida virtual que equivale a 1 pixel num ecrã com 160 dpi – MDPI. Se este pixel tiver de ser representado num ecrã maior, por exemplo XHDPI este seria representado por 4 pixéis (dois na vertical e dois na horizontal). Esta medida usa-se para se garantir que quando se criam os *layouts* de uma aplicação tenha de se garantir que os objetos que fazem parte da interface (por exemplo botões) mantenham o mesmo tamanho físico aproximado, mesmo em dispositivos com diferentes densidades de pixéis. Desta forma, sendo o *layout* e os seus componentes definidos em *dp*, sabe-se que em ecrãs com diferentes densidades de pixéis o tamanho se vai adaptar, mantendo o mesmo tamanho físico. Se não fosse assim, teria de se especificar os tamanhos de todos os elementos do *layout* para todas as possíveis densidades de pixéis existentes.

Como exemplo concreto pode falar-se dos botões. Em média o espaço clicado pelo dedo humano no ecrã de um dispositivo móvel é de 50 *dp*, pelo que se deve garantir que os botões não fiquem pequenos demais e têm pelo menos este tamanho, de altura e largura. Deve também ter-se em atenção o espaço definido entre componentes, de forma a evitar erros no processo de clicar nos botões e assim tornar a utilização da aplicação confortável. Como se pode ver na Figura 88, na aplicação desenvolvida os botões têm uma altura de 50*dp* e os espaços entre eles foram definidos com um valor de 40*dp*.



Figura 88 - Definição do tamanho dos componentes em *dp*

Quanto à largura, neste caso específico, a mesma é de 160*dp*, no entanto existem botões com outros tamanhos, sendo que nenhum é inferior a 50*dp* para se garantir que o utilizador consegue clicar nos botões, independentemente da densidade de pixéis do ecrã.

Também no que se refere às imagens tem de se ter em atenção os tamanhos em *dp* e devem inclui-se no projeto imagens de diferentes tamanhos para diferentes densidades de pixéis. No caso específico do ícone da aplicação desenvolvida, e tendo em atenção as proporções definidas na Tabela 13, foram definidas imagens PNG com tamanhos em *dp* para cada uma das densidades de pixéis padrão, como se pode ver na Figura 89.

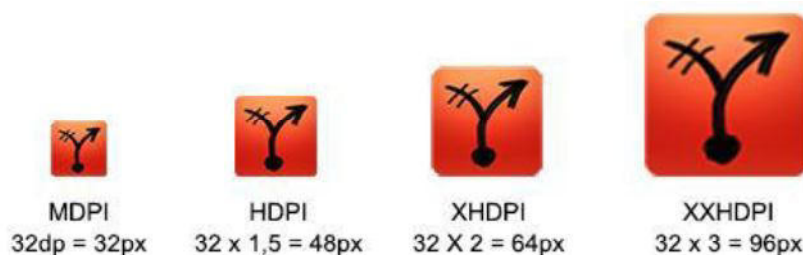


Figura 89 - Ícone criados para a aplicação

Da mesma forma foi realizado o mesmo procedimento para as imagens de fundo do *layout*.

Cada uma das imagens deve foi colocada na pasta específica, tendo em conta a que resolução se destina. Na hierarquia de pastas do projeto *Android* existe a pasta *res* (recursos). Dentro desta pasta existem as pastas *drawable-xhdpi*, *drawable-hdpi*, *drawable-mdpi* e *drawable-ldpi*. É nestas pastas que devem ser colocadas as imagens que vão fazer parte do *layout* da aplicação. A mesma imagem, com resolução diferente, deve ter o mesmo nome em qualquer uma destas pastas.

## 4.9.2 Diferentes tamanhos ecrãs

Mesmo tendo em conta o que foi descrito na secção 4.9.1, no que respeita à criação e configuração dos *layouts* para adaptação correta a diversos tamanhos e densidades de ecrãs, pode ser necessária uma configuração extra de forma a aperfeiçoar a interface.

Como se viu no capítulo anterior, é possível que em diferentes dispositivos, com diferentes densidades de pixéis, os diversos componentes sejam mostrados com o mesmo tamanho físico. No entanto, o *layout* poderá ainda não estar perfeito, se por exemplo se pretender utilizar um *layout* que foi desenvolvido para um *smartphone* num *tablet* que tenha um ecrã consideravelmente maior. Na pática, os componentes são mostrados com o mesmo tamanho. No entanto, poderá ser vantajoso que

sejam mostrados com tamanho maior, aproveitando o maior espaço existente num dispositivo com maior tamanho de ecrã.

Como se pode ver na Figura 90, o *layout* representado num dispositivo cujo ecrã possui 3.2 polegadas parece diferente do mesmo *layout* apresentado num *tablet* com 10.1 polegadas. Na prática os botões têm as mesmas dimensões físicas, utilizando as técnicas descritas no capítulo anterior, no entanto, no caso do *tablet*, como o ecrã é maior os botões poderiam ser maiores, de forma a aproveitar melhor o espaço.

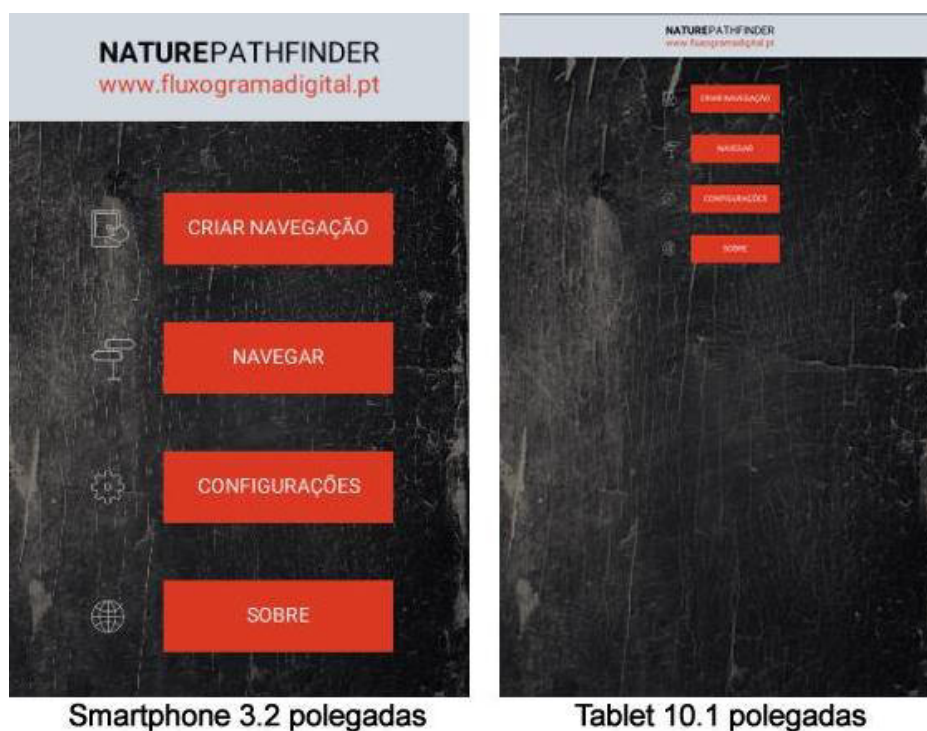


Figura 90 - O mesmo *layout* em dispositivos com ecrãs diferentes

Para se ultrapassar esta questão e se maximizar a usabilidade da aplicação, deve criar-se um ficheiro *XML* de *layout* para cada tamanho de ecrã que se pretende suportar. Cada um destes ficheiros deve ter os mesmos componentes adaptados para o tamanho do ecrã alvo. Todos os ficheiros *XML* de *layout* devem ter o mesmo nome, mas devem ser colocados em pastas diferentes, dentro da pasta *res*<sup>74</sup>.

<sup>74</sup> <http://developer.android.com/training/basics/supporting-devices/screens.html>

Na prática, não será efetivamente necessário criar-se um *layout* para cada tamanho de ecrã existente, mas poderá construir-se um *layout* genérico tendo em conta as classificações *small normal*, *large* e *xlarge*. O *Android* encarrega-se de efetuar o redimensionamento para ecrãs de tamanho intermedio. De uma forma geral, poderá colocar-se o ficheiro *XML* referente aos *smartphones* na pasta *layout* e o ficheiro *XML* para *tablets* na pasta *layout-large*.

Será o próprio *Android* a analisar automaticamente que ficheiro *XML* de *layout* vai utilizar tendo em conta o dispositivo móvel onde a aplicação está a ser executada.

No caso da aplicação *Naturepathfinder* foi utilizada a técnica descrita anteriormente e como se pode ver na Figura 91, para além do *layout* normal para *smartphones*, foi criado um segundo *layout* direcionado para *tablets*, onde os botões e o cabeçalho foram definidos com medidas maiores. Este *layout* foi colocado na pasta *layout-large*.

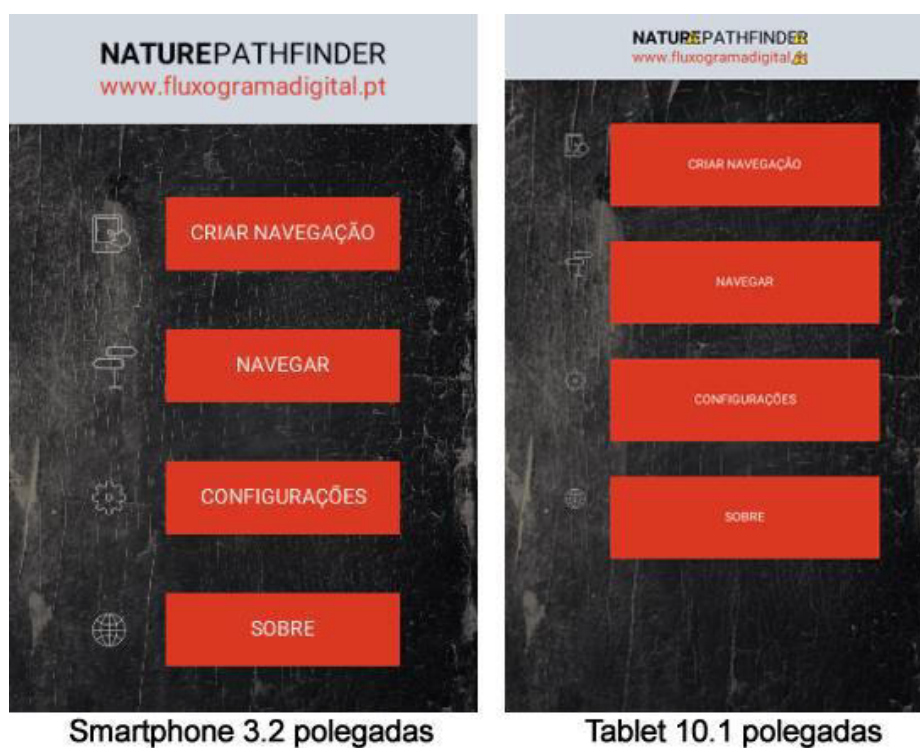


Figura 91 - O mesmo *layout* em dispositivos com ecrãs diferentes mas com adaptação para ecrãs grandes

Quando a aplicação for usada num dispositivo “pequeno” será apresentado automaticamente o *layout* da esquerda. Quando for utilizada num *tablet* será carregado o *layout* da direita. Este processo tem de ser realizado para todos os *layouts* de todas as *activitys* da aplicação.

### 4.9.1 Títulos, textos e botões

No que respeita aos textos usados numa aplicação *Android*, nomeadamente títulos, informações e botões, estes devem ser organizados num ficheiro *XML* próprio, denominado de *strings.xml*, que se encontra na pasta dos recursos. Neste ficheiro todas as *strings* têm uma identificação, que depois vai ser utilizada para representar a *string* num ficheiro de *layout* ou então numa *Activity*. Para além de ser possível criar estilos próprios para as *strings*, outra das vantagens que existem em ter todos os textos organizados desta maneira vai ser a possibilidade de utilização de diversos idiomas, tal como é explicado na secção 4.9.2.

### 4.9.2 Diferentes idiomas

Para além de diferentes densidades de pixéis, tamanho de ecrãs e *strings* dos textos é possível ter em conta mais recursos alternativos no desenvolvimento de uma aplicação *Android*, como por exemplo o idioma. Na aplicação desenvolvida existe a possibilidade de os *layouts* serem em português ou em inglês. Por defeito o idioma da aplicação é inglês, cujos textos estão armazenados no ficheiro *strings.xml* na pasta *values*, como foi analisado na secção 4.9.1. No caso do ficheiro *strings.xml* com os textos em português, este é guardado na pasta *values-pt*. Desta forma, se nas configurações do dispositivo o idioma definido for o português serão carregadas as *strings* em português. Se estiver definido outro idioma serão carregadas as *strings* em inglês<sup>75</sup>.

---

<sup>75</sup> <http://developer.android.com/guide/topics/resources/providing-resources.html#BestMatch>

## Capítulo 5

# 5. Testes e resultados

Neste capítulo são descritos os testes realizados para analisar o desempenho e a usabilidade da aplicação desenvolvida. Foram realizados testes de desempenho ao longo do desenvolvimento da aplicação, com o objetivo de se irem detetando aspetos a corrigir e assim se atingir o correto funcionamento da mesma. Na fase final do desenvolvimento da aplicação *Naturepathfinder*, foram realizados testes do grau de satisfação da aplicação final, com o objetivo de se verificar se a aplicação cumpria todos os objetivos propostos num cenário real. Foram também realizados testes de usabilidade com o objetivo de se detetar problemas que pudessem existir na interação entre o utilizador e a aplicação.

Os intervenientes nos testes realizados recorreram ao manual do utilizador para terem conhecimento do modo de funcionamento da aplicação.

As secções seguintes descrevem os testes realizados, os resultados obtidos e a análise dos mesmos.

## 5.1 Testes de desempenho

Durante o desenvolvimento da aplicação foram realizados testes que permitiram a correção e adaptação do desenvolvimento da aplicação com o fim de se atingir o correto funcionamento da mesma e alcançar os objetivos definidos para o projeto.

Os testes de desempenho foram realizados pelo criador da aplicação durante o seu desenvolvimento, com o objetivo de verificar o comportamento dos algoritmos utilizados no processo de gravação das notas do *roadbook*, navegação de um *roadbook*, deteção de erros de navegação e correção da rota. Todos estes processos que foram testados assentam numa metodologia de deteção de

contextos. Seguindo esta metodologia, a aplicação deve ter conhecimento do ambiente que a rodeia e das variações que nele ocorrem, adaptando-se automaticamente a estas alterações. Assim sendo, foi fundamental que estes testes fossem feitos em ambiente real, semelhante ao que os utilizadores finais da aplicação encontrarão.

Para a realização dos testes efetuados foram utilizados os dispositivos móveis apresentados na Tabela 14.

*Tabela 14 - Lista dos dispositivos móveis usados durante os testes de desempenho*

<b>Dispositivo</b>	<b>Versão Sistema</b>	<b>Processador</b>	<b>Polegadas</b>	<b>Resolução</b>	<b>RAM</b>
TMN Sapo A5	<i>Android</i> 2.3	800 MHz	3.5 Polegadas	800x480 px	512 Mb
Samsung Galaxy S4	<i>Android</i> 4.2.2	1.6 GHz 8 core	5 Polegadas	1920x1080 px	2 GB
Samsung Tab 4 10.1	<i>Android</i> 4.4.2	Quad-core 1.2 GHz	10.1 Polegadas	1280 x 800 px	1,5 GB

Nestes testes foram analisados alguns parâmetros de configuração usados em algoritmos implementados, nomeadamente para poupança de energia aquando da gravação das notas de um *roadbook*, descrito na secção 4.4.2.1 e da navegação de um *roadbook*, analisado na secção 4.5.1.

Também durante estes testes foram analisados os parâmetros de configuração usados no algoritmo utilizado para implementar o cálculo da distância à qual são detetados os cruzamentos e permite a passagem automática para a nota seguinte, durante da navegação dos *roadbooks*, tal como descrito na secção 4.5.4.

Os testes tiveram também como objetivo a análise das técnicas usadas para deteção de erros de navegação (secção 4.5.6) e posterior correção da rota (secção 4.6). O método utilizado para a realização dos testes foi criar um *roadbook* e depois navegar o mesmo diversas vezes, de forma a verificar o comportamento da aplicação, recolher os dados relativos ao desempenho dos algoritmos e depois analisar os mesmos para realização das alterações e correções necessárias.

Na Figura 92 podem ver-se duas representações do *layout* da aplicação aquando da realização dos testes de desempenho da aplicação. Nesta fase a interface era constituída por alguns campos de teste que permitiram avaliar as possibilidades e o comportamento da aplicação tendo em conta diversos parâmetros.

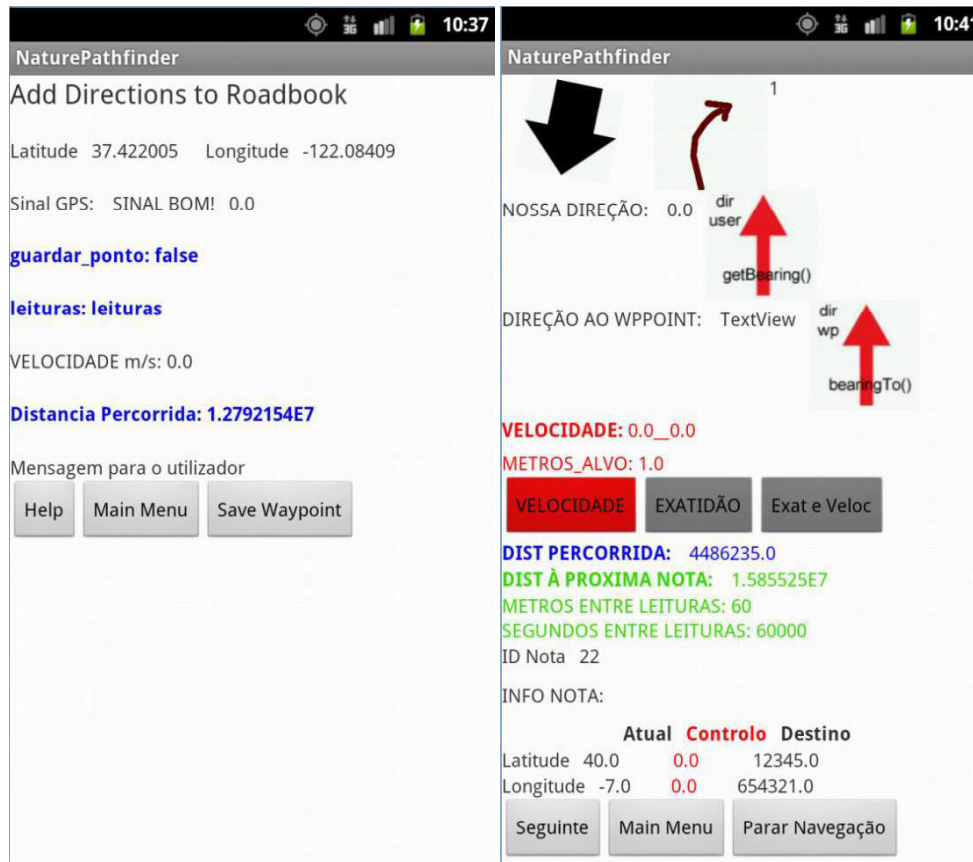


Figura 92 - Interface aquando dos testes de comportamento

Os testes foram realizados em condições reais, semelhantes às que o utilizador final irá encontrar quando utilizar a aplicação. Foram realizados testes em percursos ao ar-livre em caminhos rurais e em percursos urbanos, com recurso a veículos ou a pé. Tratando-se de uma aplicação cujo objetivo é ser usada no exterior, estes testes de comportamento teriam de ser realizados nesta situação, tendo sido tomado em conta o comportamento do principal sensor responsável pelo comportamento da aplicação: o *GPS*. Tal como referido na secção 4.4.1.3, durante os testes de desempenho a exatidão do sinal *GPS* variou sempre entre os 3 e os 12 metros, que são valores aceitáveis para a utilização correta da aplicação *Naturepathfinder*.

Por vezes foram usados dois dispositivos móveis ao mesmo tempo (um *smartphone* e um *tablet*), para que pudessem ser comparados os comportamentos dos dois dispositivos e detetar possíveis erros ou falhas (Figura 93). Os dois dispositivos sempre apresentaram comportamento muito semelhante, pelo que não existem variações de comportamento da aplicação. Obviamente, nesta



fase do teste foi possível analisar as alterações que se deviam fazer às interfaces da aplicação, tendo em conta o tamanho dos ecrãs, tal como referido na secção 4.9.2.



*Figura 93 - Testes de desempenho da aplicação diretamente no terreno*

A realização destes testes diretamente no terreno foi bastante demorada e trabalhosa, na medida que sempre que fosse feita uma alteração que houvesse necessidade de testar, o criador da aplicação teria de se deslocar ao exterior e percorrer um determinado percurso, a pé ou de carro, para efetuar os testes específicos. Cada um dos testes, envolvendo marcação de um percurso e depois navegação desse mesmo percurso, demorou em média 1,5 horas.

Durante o desenvolvimento da aplicação e com as alterações e adaptações que foram feitas durante o processo de realização dos testes de desempenho, o comportamento da aplicação foi evoluindo no que respeita à deteção de contextos. Numa fase inicial do desenvolvimento da aplicação, esta apresentava bastante falhas e alguns pontos gravados não eram detetados durante a navegação. Na fase final do desenvolvimento da aplicação, não foram detetadas falhas de navegação.

A partir destes testes de comportamento não foi feita uma avaliação formal da aplicação, mas foi antes feita uma análise constante da evolução da aplicação e foi-se corrigindo e adaptando o seu comportamento. Apesar da morosidade da metodologia utilizada para a realização dos testes de comportamento, esta revelou-se eficaz, pois para além da correção de comportamentos da aplica-

ção, conseguiu-se definir os parâmetros usados nos algoritmos implementados na deteção de contextos.

## 5.2 Teste do grau de satisfação da aplicação

Na fase final do desenvolvimento da aplicação foram realizados testes com o público-alvo, onde pôde ser avaliado o comportamento global da aplicação e definidos aspetos que poderão ser melhorados.

Nesta fase foi realizado um teste em cenário real com um conjunto de 10 pessoas, com idades compreendidas entre os 20 e os 65 anos. Durante o teste realizado, o autor da aplicação acompanhou cada um dos indivíduos durante a realização das tarefas pretendidas (criar e navegar um *roadbook*), de forma a perceber se a aplicação era usada corretamente ou se haveria dificuldades ou erros na sua utilização. Pretendeu-se que este teste de grau de satisfação da aplicação não servisse apenas para coletar dados para análise, mas também que se pudesse usar para tirar conclusões práticas sobre o funcionamento da aplicação e conseqüentemente fazerem-se possíveis alterações para melhorar a mesma. Assim sendo, optou-se pela realização de um teste bastante personalizado. Aquando da divulgação da aplicação na *Internet*, o autor da aplicação poderá obter um *feedback* de um maior número de indivíduos e usar também essa informação para melhorar a aplicação.

Todos indivíduos que participaram no teste atual já tinham anteriormente utilizado *roadbooks* tradicionais em papel para percorrer percursos, mas apenas 3 destas pessoas tinham já criado *roadbooks* no seu formato tradicional.

Antes do início do teste, foi disponibilizado aos utilizadores o ficheiro APK<sup>76</sup> da aplicação para que pudessem instalar a mesma nos seus dispositivos móveis.

Foram utilizados diferentes modelos de *smartphones* e não foi usado nenhum *tablet*, de forma a criar maiores dificuldades na utilização da aplicação, devido aos tamanhos dos ecrãs, e assim conseguir-se avaliar melhor a facilidade/dificuldade da utilização da aplicação em ecrãs mais pequenos.

Cada pessoa que participou no teste criou um *roadbook* completo de um percurso escolhido pelo próprio. Mais tarde, todas as pessoas utilizaram um *roadbook*, não criado por si, e procederam

---

<sup>76</sup> <https://developer.android.com/tools/building/index.html>

à sua navegação. Desta forma, foi possível aos utilizadores testarem a aplicação nas suas duas vertentes (criação de *roadbooks* e navegação de *roadbooks*).

O teste foi realizado durante 4 dias, sendo que a primeira fase (criação de *roadbooks*) demorou em média 1 hora a realizar por cada individuo e a segunda fase (navegação de *roadbooks*) demorou em média 30 minutos. Cada teste foi realizado individualmente, na medida que houve um acompanhamento personalizado por parte do autor da aplicação ao desempenho de cada individuo.

No que respeita ao terreno foram utilizados caminhos de terra para a criação e navegação de 8 *roadbooks*, que foram percorridos utilizando veículos todo-terreno. Os restantes 2 *roadbooks* foram criados em ambiente urbano e o percurso foi realizado a pé.

Relativamente às notas criadas, em média os *roadbooks* tinham 14 notas.

Após completado este teste foi realizado um inquérito aos participantes, cujos resultados se passam a analisar.

**Questão 1. Como classifica a aplicação desenvolvida no que respeita à criação de *roadbooks*? (Boa/Satisfatória/Má)**

Nesta questão apenas 10% dos utilizadores classificaram a aplicação de satisfatória, no que respeita à parte da criação de *roadbooks* sendo que todos os outros atribuíram a classificação de boa. Como qualquer outra aplicação, também esta necessita de um tempo de adaptação, especialmente num tipo de aplicação em que tem de se ter em consideração conceitos externos relativos, neste caso, aos *roadbooks*.

**Questão 2. Como classifica a aplicação desenvolvida no que respeita à navegação de *roadbooks*? (Boa/Satisfatória/Má)**

Neste ponto 100% os utilizadores atribuíram a classificação máxima à aplicação, na medida que nesta seção a interação do utilizador será mínima e o mesmo terá apenas de tomar a decisão da direção a tomar em cada cruzamento, sendo que a aplicação se encarrega de todos os restantes processos de navegação.

**Questão 3. Utilizando a aplicação conseguiu criar um *roadbook* para um percurso? (Sim/Não/Parcialmente)**

Nesta questão 20% dos inquiridos não conseguiram completar a criação do *roadbook*, sendo que os restantes conseguiram completar esta tarefa sem problemas e sem ajuda. No caso das 2 pessoas que apenas conseguiram criar o *roadbook* parcialmente, requisitaram a ajuda do autor da aplicação, pois estavam com dificuldades em entender que pontos deveriam registar, devido a possuírem pouca experiência na utilização de *roadbooks* e nunca terem criado um *roadbook* tradicional em papel. Após a explicação e ajuda fornecida, conseguiram completar a criação do *roadbook*. Tal como analisado na secção 3.13, o processo de criação de *roadbooks* estar bastante simplificado relativamente aos tradicionais *roadbooks* em papel. No entanto, os utilizadores da aplicação *Naturepathfinder* devem ter algumas noções básicas sobre como funciona um *roadbook* e entender a filosofia de navegação utilizando os mesmos.

**Questão 4. Qual o grau de dificuldade/facilidade na criação de um *roadbook* usando a aplicação? (Fácil/Média/Difícil)**

Apesar de na questão anterior se ter visto que 20% dos inquiridos não conseguiram completar a criação do *roadbook* sem ajuda, apenas 10% dos utilizadores afirmaram que a dificuldade de criação de um *roadbook* é média. Os restantes classificaram esta tarefa como sendo fácil.

**Questão 5. Como classifica a aplicação desenvolvida no que respeita ao desenho dos esquemas das notas? (Boa/Satisfatória/Má)**

Nesta questão 20% das pessoas classificaram a aplicação como sendo satisfatória devido ao facto de por vezes ser difícil desenhar em *Smartphones* com ecrãs mais pequenos. Apesar disto as restantes 80% classificaram a aplicação como sendo boa.

**Questão 6. Utilizando a aplicação conseguiu realizar a navegação de um *roadbook*? (Sim, Não/Parcialmente)**

Nesta etapa todos os utilizadores conseguiram efetuar a navegação de um *roadbook* até ao final.

**Questão 7. Durante a navegação ocorreu algum erro de navegação, tendo seguido um caminho errado? (Sim/Não)**

Durante a navegação ocorreram erros de navegação a 20% dos utilizadores, sendo que seguiram por caminhos errados, não seguindo a indicação apresentada pelo esquema.

**Questão 8. Caso tenha respondido afirmativamente à pergunta anterior, conseguiu retomar a rota correta usando um dos métodos de correção? Se sim, qual foi o método utilizado? (Não/Sim, usando a opção do mapa/Sim, usando a opção da seta)**

O método escolhido pelos utilizadores para retomar à rota correta foi o mapa. Todos conseguiram regressar ao percurso correto. Afirmaram que fizeram esta escolha por se tratar de um método mais visual e direto.

**Questão 9. Qual o grau de dificuldade/facilidade na navegação de um *roadbook* usando a aplicação? (Fácil/Média/Difícil)**

Todos os utilizadores afirmaram que é fácil utilizar a aplicação aquando da navegação de um *roadbook*.

**Questão 10. Na sua opinião esta aplicação apresenta claras vantagens relativamente à criação de um *roadbook* tradicional em papel? (Sim, apresenta muitas vantagens/Não apresenta vantagens/Apresenta poucas vantagens)**

Neste tópico todos os utilizadores afirmaram que a aplicação apresenta muitas vantagens relativamente à criação dos tradicionais *roadbooks* em papel.

**Questão 11. Na sua opinião esta aplicação apresenta claras vantagens relativamente à navegação de um *roadbook* tradicional em papel? (Sim, apresenta muitas vantagens/Não apresenta vantagens/Apresenta poucas vantagens)**

Também neste tópico todos os utilizadores afirmaram que a aplicação apresenta muitas vantagens relativamente à navegação utilizando os tradicionais *roadbooks* em papel.

**Questão 12. A utilização desta aplicação foi ao encontro das suas expectativas? (Sim/Não)**

Pode concluir-se que a aplicação foi ao encontro das expectativas dos utilizadores, na medida que 100% dos inquiridos responderam afirmativamente a esta questão.

**Questão 13. Que aspetos gostaria de ver melhorados na aplicação? (Resposta Aberta)**

Neste caso um utilizador sugeriu que a aplicação poderia ter mais sons de alerta. Assim sendo acabou por ser implementado um aviso sonoro quando tivesse sido atingido um cruzamento e outro aviso sonoro diferente quando houvesse um erro de navegação.

Outra sugestão fornecida por três utilizadores foi que o teclado virtual poderia recolher sempre que o utilizador clicasse em qualquer parte do ecrã que não uma caixa de texto. Isto foi sugerido porque quando o utilizador acabava de inserir um texto numa caixa de texto o teclado virtual continuava ativo até que se clicasse no botão retroceder do dispositivo móvel. Esta opção não era intuitiva pelo que foi programada também esta alteração.

Outra das sugestões de seis utilizadores foi que ao ser escolhido o desenho do esquema houvesse algum tipo de confirmação da mesma, para que se tivesse a certeza que foi escolhido o esquema pretendido. Também esta funcionalidade foi implementada.

Por último, dois utilizadores sugeriram que seria útil que, além da ajuda que já existia na aplicação, se pudesse ter acesso ao manual da aplicação em formato PDF. Esta opção foi implementada, sendo possível neste momento fazer-se o *download* do manual da aplicação em formato PDF a partir da própria aplicação.

Todas as melhorias propostas pelos inquiridos foram implementadas na aplicação.

**Questão 14. Na sua opinião que novas funcionalidades devem ser implementadas? (Resposta Aberta)**

Um dos utilizadores sugeriu que se poderia implementar uma opção que permitisse fazer a impressão em papel dos *roadbooks* criados, para que pudessem ser utilizados por pessoas que não possuíssem dispositivos móveis *Android*. Esta sugestão é bastante interessante e, apesar de não fazer parte dos objetivos da aplicação atual, poderá ser implementada em trabalho futuro.

Outra sugestão, de outro utilizador, foi a implementação desta aplicação para dispositivos *iPhone* e *iPad*. Será sem dúvida também uma sugestão a ter em conta para trabalho futuro, pois é bastante pertinente.

## 5.3 Testes de usabilidade

Nestes testes da usabilidade tentou apurar-se e resolver problemas que pudessem existir na interação entre o utilizador e a aplicação. Estes testes foram realizados usando as Heurísticas de Nielsen (Nielsen, 1994), que estão representadas no Anexo D.

Estas heurísticas representam um método que permite analisar a usabilidade de aplicações informáticas.

Os testes de usabilidade e a avaliação foram realizados por um conjunto de 10 pessoas, com idades compreendidas entre os 20 e os 65 anos. Este teste de usabilidade foi realizado no final da realização do teste do grau de satisfação da aplicação (secção 5.2), pelos mesmos participantes, em que cada um criou e navegou um *roadbook* em cenário real.

Foi dado conhecimento das Heurísticas de Nielsen aos utilizadores que participaram no teste e foi-lhes pedido que classificassem a implementação de cada heurística na aplicação, com a escala de boa, suficiente ou má. Na Figura 94 pode ver-se o gráfico do resultado do teste realizado.

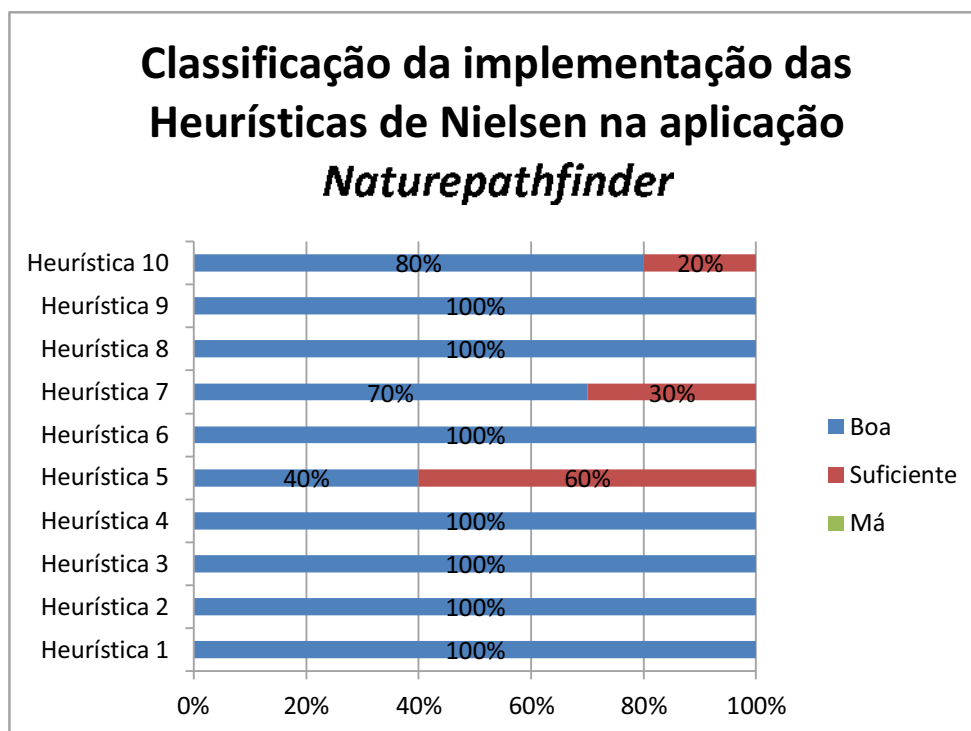


Figura 94 - Classificação da implementação das Heurísticas de Nielsen na aplicação *Naturepathfinder*

Analisando o gráfico anterior pode ver-se que a aplicação obteve classificação de boa, na maioria das heurísticas implementadas.

Relativamente à heurística 5 - *Prevenção de erro*, 60% dos inquiridos atribuiu uma classificação suficiente, tendo afirmado que ao ser escolhido o desenho de um esquema de um cruzamento, durante a gravação das notas, deveria haver algum tipo de confirmação do mesmo, para que se tivesse a certeza que foi escolhido o esquema pretendido. Depois de feita esta sugestão a funcionalidade foi implementada na aplicação. Os restantes 40% de utilizadores classificaram a implementação desta heurística de boa.

No que respeita à heurística 7 - *Flexibilidade e eficiência de utilização*, 30% dos inquiridos classificaram a implementação desta heurística como sendo suficiente, sugerindo que o teclado virtual poderia recolher, sempre que o utilizador clicasse em qualquer parte do ecrã que não uma caixa de texto. Isto foi sugerido porque quando o utilizador acabava de inserir uma informação numa caixa de texto o teclado virtual continuava ativo até que se clicasse no botão retroceder do dispositivo móvel. Também esta sugestão acabou por se implementada na aplicação. Os restantes 70% dos inquiridos classificaram a implementação desta heurística de boa.

Quanto à heurística 10 - *Ajuda e documentação*, foi atribuída uma classificação de suficiente à sua implementação por parte de 20% dos inquiridos, pois afirmaram que seria bastante interessante que, para além da ajuda que já existe na aplicação, se pudesse ter acesso ao manual da aplicação em formato PDF. Esta opção foi implementada, sendo possível neste momento fazer-se o *download* do manual da aplicação em formato PDF a partir da própria aplicação. Os restantes 80% de inquiridos classificaram a implementação desta heurística de boa.

Pode concluir-se que este tipo de testes é bastante útil, na medida que permitiu a deteção de diversas falhas relativas à usabilidade que foram corrigidas, permitindo à aplicação ter um bom nível de usabilidade. Na secção seguinte vão-se enumerar aspetos da implementação da aplicação *Naturepathfinder* relacionados com cada uma das Heurísticas de Nielsen.



## 5.4 Aspectos da implementação referentes a cada Heurística de Nielsen

Tal como foi referido nos objetivos do projeto (secção 1.3), pretendeu-se que a aplicação tivesse um elevado nível de usabilidade. Para o efeito descrevem-se as características da aplicação correspondentes a cada heurística.

### **Heurística 1 - Visibilidade do estado do sistema**

Aquando da criação do *roadbook* os utilizadores sabem a todo momento as coordenadas *GPS* do local onde se encontram, qual a velocidade atual e qual a exatidão do sinal *GPS*. Quando o utilizador está a navegar, usando um *roadbook*, o utilizador sabe qual a direção que deve tomar quando chegar ao próximo cruzamento, assim como em que direção fica o mesmo e a que distância fica em linha reta. Para além disso, sabe qual a velocidade atual e qual a exatidão do sinal de *GPS*. Poderá também ter acesso a informação específica da nota a que se refere o próximo cruzamento, apesar de esta informação não ser obrigatória.

Sempre que exista um erro de navegação o utilizador será informado disso mesmo. Durante o processo de correção da rota se o utilizador estiver a utilizar a opção do mapa sabe a todo o instante qual a sua localização e a localização do caminho correto, assim como a distância em linha reta até esse caminho correto. Se estiver a ser usada a técnica da correção da rota pela seta, o utilizador saberá a todo o instante em que direção fica o caminho correto e qual a distância em linha reta a que se encontra o início desse caminho. Ao ser encontrado o caminho correto o utilizador será informado disso mesmo.

### **Heurística 2 - Equivalência entre o sistema e o mundo real**

Foi utilizada terminologia familiar ao utilizador, sendo que existem alguns termos mais técnicos no que respeita à navegação utilizando *roadbooks*, tal como latitude, longitude e *roadbook*. No entanto estes termos são a parte fundamental desta aplicação e dos conceitos base que qualquer pretensão utilizador de um *roadbook* tradicional em papel conhece, pelo que não foram considerados termos técnicos e foram considerados termos familiares ao utilizador, tendo em conta o contexto da aplicação.

**Heurística 3 - Controle do utilizador e liberdade**

Em qualquer instante o utilizador pode escolher quais as ações que quer tomar e nunca é obrigado a realizar determinada tarefa. Em todas as secções da aplicação o utilizador poderá retroceder para o menu anterior, quer seja utilizando botões como “Cancelar”, “Voltar”, “Fechar”, “Parar Gravação” e “Parar Navegação”. Em algumas secções onde podem haver listas muito extensas de dados, como por exemplo na listagem de *roadbooks* que o utilizador tem à disposição, não haverá nenhum botão que permita voltar ao menu anterior, com o objetivo de manter a interface simples, mas será possível ao utilizador efetuar essa ação, intuitivamente, usando o botão de retroceder do dispositivo móvel.

**Heurística 4 - Consistência e padrões**

Em toda a aplicação os mesmos comandos, ações ou conceitos têm sempre os mesmos efeitos e significados. Por exemplo, aquando da navegação existe uma seta que indica a direção em que fica o próximo cruzamento, assim como a indicação da distância em linha reta a que se encontra desse mesmo cruzamento. Nesta secção trata-se apenas de informação auxiliar ao esquema do próximo cruzamento, que será o meio principal de orientação. Já na parte da correção da rota (quando houve um erro de navegação) volta a ser usada a seta e a ser indicada a distância em linha reta, mas agora como meio principal de orientação. No entanto em ambos os casos tanto a seta como a distância apresentada transmitem exatamente o mesmo tipo de informação.

**Heurística 5 - Prevenção de erro**

Sempre que se está a realizar uma tarefa importante e o utilizador clica num botão que irá alterar completamente a tarefa que está a realizar, será mostrada uma mensagem que permitirá ao utilizador confirmar a ação ou regressar à tarefa que estava a realizar. Por exemplo, aquando da gravação dos pontos de um *roadbook* o utilizador poderá clicar inadvertidamente em *Stop Recording*. Neste caso será enviada uma mensagem de alerta para que o utilizador confirme ou cancele esta ação. Da mesma forma, durante a navegação de um *roadbook* se o utilizador clicar em *Stop Recording* será também enviada uma mensagem de alerta para confirmar a ação, permitindo ao utilizador gravar a navegação atual para a poder continuar mais tarde.

Aquando da gravação de um percurso, depois de o utilizador escolher um esquema de uma nota, surge uma mensagem que mostra o esquema escolhido, para que o utilizador confirme que foi aquele o esquema pretendido e se evite que seja adicionado ao *roadbook* um esquema errado.

**Heurística 6 - Reconhecer ao invés de relembrar**

Na aplicação desenvolvida foram utilizados ícones com significado como por exemplo a seta indicadora da direção certa, assim como os ícones usados para desenhar os esquemas das notas. Todos os botões e títulos das secções possuem nomes simples e diretos que facilitam o entendimento por parte do utilizador.

**Heurística 7 - Flexibilidade e eficiência de utilização**

Aquando da criação dos dados de um *roadbook*, antes do início da captura das notas, alguns dos campos estão já preenchidos de forma a facilitar a introdução de dados e ao mesmo tempo facilitar a perceção por parte do utilizador de que tipos de dados se pretendem que sejam introduzidos. Quando se efetua uma pesquisa de *roadbooks* ou organização irá surgir uma lista com todos os dados. No entanto será possível ao utilizador escrever, numa caixa de texto, o nome do *roadbook* ou organização que procura, para que seja aplicado um filtro e sejam apresentados os dados procurados, caso existam. Quando se pretende contactar a organização de um *roadbook*, serão automaticamente preenchidos os campos de *email* e do número de telefone da organização (para envio de mensagem *SMS* ou telefonema). No caso específico em que se pretende contactar a organização depois de ter acontecido um erro de navegação, o campo do corpo da mensagem *SMS* e de *email* vai ser preenchido automaticamente com uma mensagem genérica de ajuda, à qual são adicionadas automaticamente as coordenadas atuais do dispositivo móvel, de forma a simplificar o processo de pedido de ajuda.

Outro exemplo de um acelerador de tarefa é o processo utilizado para minimizar o teclado virtual quando se termina de inserir dados numa caixa de texto. Para efetuar esta ação basta o utilizador clicar em qualquer parte do ecrã, que não a própria caixa de texto.

**Heurística 8 - Estética e design minimalista**

Foi desenvolvido para a aplicação um *layout* simples e direto, em que o utilizador apenas tem acesso a informação que ele necessita em determinada situação. Optou-se por deixar ficar em praticamente em todos os *layouts* o cabeçalho com o nome da aplicação, pois concluiu-se que não iria contribuir para distrair a atenção do utilizador, na medida que nunca muda de espeto ou localização, acabando por passar despercebido e continuando a cumprir o seu objetivo. Apenas se optou por retirar o cabeçalho naquele que é considerada a parte mais sensível e importante da aplicação e na que não pode haver distração nenhuma, que é a parte da navegação de *roadbooks*. Nesta secção apenas aparece a informação que o utilizador irá necessitar para realizar a navegação do *roadbook*.

**Heurística 9 - Auxiliar os utilizadores a reconhecer, diagnosticar e recuperar ações erradas**

Como foi explicado na secção 4.6 foi desenvolvido um processo para deteção de quando o utilizador fez um erro de navegação, assim como depois foram desenvolvidos dois métodos de correção da rota e atingir de novo o caminho correto, usando um mapa ou uma seta indicadora da direção do caminho correto. O utilizador será sempre informado de todas as ações que ocorreram, nomeadamente quando é efetuado o erro e quando é recuperada a navegação correta.

**Heurística 10 - Ajuda e documentação**

A ajuda fornecida pela aplicação existe em todos os menus de navegação, bastando para isso o utilizador clicar no botão *Help*. Em cada menu, a informação fornecida será relacionado com a secção atual. Na secção do menu principal quando o utilizador clica no botão *About* poderá ter acesso a uma explicação geral do modo de funcionamento da aplicação. Neste mesmo menu será possível fazer-se o *download* do manual do utilizador da aplicação em formato *PDF*, que tem descrito todas as características e modo de funcionamento de todas as funcionalidades da aplicação.

## Capítulo 6

# 6. Conclusões e perspectivas de desenvolvimento

Foi desenvolvida uma aplicação para dispositivos móveis *Android* designada por *Naturepathfinder* que permite a criação e navegação de *roadbooks* digitais. A aplicação está dividida em duas partes, a secção referente à criação de *roadbooks* diretamente no terreno em que o utilizador é assistido pela aplicação e é feita a gravação de todos os pontos necessário para identificar o percurso a percorrer, usando dados capturados pelo sensor *GPS* do dispositivo e esquemas de notas que poderão ser desenhadas diretamente pelo utilizador. A outra parte refere-se à navegação de *roadbooks criados anteriormente*, que será feita de uma forma automática pela aplicação, em que o utilizador apenas tem a função de decidir qual o caminho a seguir, tendo em conta a informação fornecida pela aplicação. A aplicação foi desenvolvida com base em deteção de contextos, de modo a que sempre que acontecer um erro de navegação e o utilizador seguir por um caminho errado a aplicação irá informar o utilizador e indicar o processo para retomar o caminho correto. O utilizador poderá sempre partilhar os *roadbooks* criados na *web*, usando um serviço que foi criado para a aplicação, assim como fazer *download* de *roadbooks* criados por outros utilizadores da aplicação. A aplicação possui um manual do utilizador que poderá ser consultado a partir da própria aplicação.

Tal como nos *roadbooks* tradicionais em papel, quem cria um *roadbook* utilizando esta aplicação deverá ter noção dos conceitos base de um *roadbook*, tais como o facto de terem de se registar todos os cruzamentos ou situações que se pretenda que o utilizador final tenha acesso durante a navegação. Obviamente que se o criador de um *roadbook* não tiver em conta estes conceitos, o *roadbook* poderá levar a erros de navegação ou ao surgimento de dúvidas por parte do utilizador que esteja a navegar um *roadbook* digital com recurso à aplicação desenvolvida.

No que se refere à criação de *roadbooks* digitais, a aplicação apresenta uma grande vantagem relativamente aos tradicionais *roadbooks* em papel, pois na gravação das notas o utilizador não terá de apontar todas as distâncias percorridas entre cada ponto/cruzamento. Medidas de distâncias erradas em *roadbooks* tradicionais em papel são o fator que mais leva a erros de navegação, pois os métodos usados tradicionalmente para a medição das distâncias são pouco precisos.

Se o processo de criação de um *roadbook* está bastante facilitado relativamente aos métodos tradicionais então o processo de navegação de um *roadbook* está verdadeiramente simples e direto, quando comparado com a navegação por *roadbooks* em papel. Basicamente o utilizador apenas tem de estar atento às indicações da aplicação e seguir os caminhos apresentados.

Num *roadbook* tradicional em papel, muitas vezes o utilizador só se apercebe que está perdido e não está a fazer o percurso definido alguns cruzamentos a seguir ao erro e depois torna-se muito difícil, ou quase impossível, retomar a navegação correta, especialmente se se estiver numa zona desconhecida. Isto está resolvido também com esta aplicação, pois alguns metros a seguir a um erro de navegação o utilizador é informado do erro e podem ser usados dois métodos independentes que permitem recuperar o caminho correto. O utilizador poderá escolher o método que desejar, usando um a seta indicadora do caminho correto ou um mapa onde se pode ver a localização do caminho correto e a localização atual do dispositivo móvel.

No que respeita ao método de navegação, em que se tem de analisar o desenho do esquema do cruzamento com as indicações, pode ser considerado antiquado por potenciais utilizadores, pois hoje em dia o método de navegação mais famoso é sem dúvida a utilização de *tracks* diretamente desenhados em cima de um mapa, bastando ao utilizador seguir esse percurso. No entanto, a navegação por *roadbooks* continua a ser utilizado por muitas pessoas, pois para além de se pretender chegar ao destino, o objetivo deste tipo de navegação é a exploração e o desafio de conseguir encontrar todos os pontos assinalados e conseguir com sucesso percorrer o percurso proposto.

Pretende-se também com esta aplicação que a navegação por *roadbook* não seja posta de parte em detrimento da navegação por *track* num mapa, mas que contribua para um “rejuvenescimento” desta técnica e que possa passar a ser mais utilizada, não só pelos utilizadores dos *roadbooks* em papel, mas também por pessoas que nunca tenham realizado navegação com um *roadbook* tradicional.

Para além da tradicional utilização deste tipo de sistema de navegação em percursos ao ar livre, a aplicação poderá também ser utilizada em outras vertentes que impliquem a definição de um

percurso, nomeadamente na área do turismo, pois poderá ser utilizada para guiar os utilizadores através de roteiros turísticos predefinidos.

Foram realizados testes de desempenho ao longo do desenvolvimento da aplicação com o objetivo da análise e correção do comportamento da aplicação. Na fase final do desenvolvimento da aplicação, foram realizados testes do grau de satisfação da aplicação final, com o objetivo de se verificar se a aplicação cumpria todos os objetivos propostos num cenário real. Foram também realizados testes de usabilidade com o objetivo de se analisar a interação entre o utilizador e a aplicação. Após a realização destes testes e da respetiva análise foi possível efetuar alterações e correções na aplicação, especialmente no que se refere à usabilidade.

No que se refere aos objetivos gerais do projeto estes foram cumpridos, pois como foi referido, foi desenvolvida uma aplicação que permite a criação e posterior utilização de *roadbooks*, assim como a deteção e assistência ao utilizador para a correção de erros de navegação.

Relativamente aos objetivos específicos, estes também foram cumpridos, pois foi implementado um editor gráfico para criação de *roadbooks* no terreno, que assiste o utilizador na criação das notas de navegação através do uso de símbolos predefinidos ou criados pelo utilizador e dados provenientes dos sensores do dispositivo móvel. Foi também implementado um assistente de navegação com base em *roadbooks* existentes, que apresenta as notas de forma automática através da deteção do contexto de navegação. Outro dos objetivos específicos era implementar um assistente para deteção de erros de navegação e recuperação do caminho correto. Também este objetivo foi cumprido, pois se for detetado um erro de navegação a aplicação irá indicar o caminho correto ao utilizado, utilizando uma seta que apontará na direção do caminho correto ou utilizando um mapa onde poderá consultar a posição atual e aposição do caminho correto. Todos os dados utilizados pela aplicação estão guardados na base de dados, adicionados aquando da gravação das notas.

Tal como foi referido anteriormente nesta secção, a aplicação resolve problemas específicos dos sistemas tradicionais, tais como dificuldade e complexidade na medição das distâncias entre cada cruzamento, deteção de erros de navegação e respetiva correção da rota, dificuldade na identificação da nota/cruzamento atual, complexidade da criação de um *roadbook* no que respeita a todos os dados que se devem registar manualmente, necessidade de equipamento extra para cálculo de distâncias tanto na criação como na navegação de *roadbooks*, que também constituía um dos objetivos específicos.

Um dos objetivos específicos principais era a implementação de uma aplicação que assentasse numa metodologia de deteção de contextos. Este objetivo também foi cumprido, pois sempre que aconteça uma alteração no ambiente em que a aplicação está a ser executada, a mesma adaptar-se a essa mudança de forma automática, nomeadamente durante o processo de gravação das notas dos *roadbooks*, da navegação de um *roadbook* e durante a correção da rota depois de ter ocorrido um erro de navegação.

Tendo em conta o que foi analisado na secção 5.3, pode concluir-se que também o objetivo de criação de uma aplicação com um bom nível de usabilidade foi cumprido.

Relativamente às perspectivas de desenvolvimento, existem várias funcionalidades e características que poderão ser implementadas. Futuramente poderá também incluir-se no manual do utilizador um capítulo referente aos conceitos base do modo de utilização de um *roadbook*, de maneira a complementar a informação apresentada e permitir a utilizadores menos experientes entenderem melhor a filosofia de navegação utilizando esta técnica e assim tenham uma melhor experiência na utilização da aplicação *Naturepathfinder*.

No projeto atual a aplicação foi desenvolvida para a plataforma *Android*. No entanto, num projeto futuro poderá ser desenvolvida esta aplicação para outras plataformas.

Na secção 3.14 foi feito o estudo da importância que potenciais utilizadores atribuíram a determinadas características propostas. Neste caso os inquiridos valorizaram bastante todas estas características, mas uma delas não foi implementada, por não fazer parte do âmbito do projeto e também porque pode dar origem a erros nas marcações dos pontos: a possibilidade de os pontos de um *roadbook* poderem ser marcados num mapa *online* diretamente na aplicação. Esta poderá ser uma funcionalidade a implementar futuramente, para complementar o método de gravação de coordenadas diretamente “no terreno”. Apesar de, no referido estudo, as características apresentadas terem agradado aos inquiridos, haverá sempre utilizadores mais puristas que poderão preferir continuar a utilizar os tradicionais *roadbooks* em papel, pelo que futuramente poderá ser implementada uma opção que permita a impressão dos *roadbooks* criados.

A aplicação *Naturepathfinder* será disponibilizada no serviço *Google Play*, sendo que será possível obter feedback por parte de mais utilizadores e eventualmente proceder-se a alterações e correções que possam ser sugeridas.



# Referências bibliográficas

- Android. (s.d.). *Android Developers*. Obtido em 2014-2015, de <http://developer.android.com/>
- Android View Basics: Coordinates, Margin, Padding, Dip, Px*. (s.d.). Obtido em janeiro de 2015, de <https://laaptu.wordpress.com/2013/12/12/android-view-basics-coordinatesmarginpaddingdippx/>
- ArcGIS. (s.d.). *ArcGIS Resources*. Obtido em julho de 2014, de Smartphones, Tablets and GPS Accuracy: <http://blogs.esri.com/esri/arcgis/2013/07/15/smartphones-tablets-and-gps-accuracy/>
- CalibraEND*. (s.d.). Obtido em julho de 2014, de Diferença entre Precisão e Exatidão: <http://calibraend.blogspot.pt/2013/02/voce-conhece-diferenca-entre-precisao-e.html>
- Chen, G., & Kotz, D. (2000). A Survey of Context-Aware Mobile Computing Research. (pp. 2-9). Hanover: Dartmouth Computer Science Technical Report TR2000-381.
- Filezilla - The free ftp solution*. (s.d.). Obtido em junho de 2014, de FileZilla Features: <https://filezilla-project.org/>
- Foundation, O. (s.d.). *OpenStreetMap Foundation*. Obtido em fevereiro de 2014, de OpenStreetMap: [http://wiki.osmfoundation.org/wiki/Main\\_Page](http://wiki.osmfoundation.org/wiki/Main_Page)
- Gamin. (s.d.). *Garmin*. Obtido em maio de 2014, de eTrex: <https://buy.garmin.com/pt-PT/ES/nos-trilhos/portateis/etrex-10/prod87768.html>
- GmbH, w. (s.d.). *Software.Informer*. Obtido em abril de 2014, de Roadbook Assistant: <http://wetexx-gmbh.software.informer.com/>
- Hubsch, E. (2012). Uma Abordagem Comparativa do desenvolvimento de aplicações para dispositivos móveis. *Monografia submetida como exigência de Tecnólogo em Processamento de Dados* (pp. 42-59). São Paulo: Faculdade de Tecnologia de São Paulo.
- JSON. (s.d.). *JSON*. Obtido em janeiro de 2015, de Introducing JSON: <http://json.org/>
- Knutsen, J. (2009). Web Service Clients on Mobile Android - A Study on Architectural Alternatives and Client Performance. (pp. 11-15). Noruega: Norwegian University of Science and Technology - Department of Computer and Information Science.

- Mapquest. (s.d.). *Mapquest*. Obtido em agosto de 2014, de <http://www.mapquest.com/satellite-maps/>
- Millette, G., & Stroud, A. (2012). *Professional Android Sensor Programming*. (pp. 27-44). Indianapolis: John Wiley & Sons, Inc.
- Mobile Atlas Creator*. (s.d.). Obtido em agosto de 2014, de MOBAC: <http://mobac.sourceforge.net/>
- Morgado, P. J. (2009). Avaliação da precisão de posicionamento inerente à utilização de sistemas GPS de baixo custo, receptores utilizados para navegação. (pp. 23-25). Braga: Universidade do Minho.
- MundoGeo. (s.d.). *MundoGeo*. Obtido em maio de 2014, de Utilização do Google Earth para obtenção de mapas viários urbanos para SIG:  
<http://mundogeo.com/blog/2009/07/09/utilizacao-do-google-earth-para-obtencao-de-mapas-viarios-urbanos-para-sig/>
- MySQL*. (s.d.). Obtido em junho de 2014, de Why MySQL?: <https://www.mysql.com/why-mysql/>
- Nielsen, J. (1994). Enhancing the Explanatory Power of Usability Heuristics. (p. 156). New York: CHI '94 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.
- Oficina da Net*. (s.d.). Obtido em julho de 2014, de Como funciona o GPS?:  
<http://www.oficinadanet.com.br/post/12406-como-funciona-o-gps>
- Open Data Commons*. (s.d.). Obtido em agosto de 2014, de Database License:  
<http://opendatacommons.org/licenses/odbl/>
- Openstreetmap. (s.d.). *Openstreetmap*. Obtido em julho-agosto de 2014, de  
<https://www.openstreetmap.org>
- Osmdroid. (s.d.). *Osmdroid*. Obtido em agosto de 2014 , de  
<https://code.google.com/p/osmdroid/downloads/detail?name=osmdroid-android-3.0.8.jar>
- Raffaeu's Blog*. (s.d.). Obtido em fevereiro de 2015, de Understand Density Independent Pixels:  
<http://blog.raffaeu.com/archive/2015/03/04/understand-density-independent-pixels-dpi.aspx>
- Riesterer, J. (s.d.). *Introduction to Topographic Maps*. Obtido em julho de 2014, de Get a Bearing:  
[http://geology.isu.edu/geostac/Field\\_Exercise/topomaps/bearing.htm](http://geology.isu.edu/geostac/Field_Exercise/topomaps/bearing.htm)

- Rubke. (s.d.). *Openstreetmap*. Obtido em setembro de 2014, de OSMtiledownloader:  
<http://wiki.openstreetmap.org/wiki/OSMtiledownloader>
- SA, T. (2014). *TripY II - User's manual*. Bélgica: TripY SA.
- SASU, X. (s.d.). *XDAppfactory*. Obtido em maio de 2014, de Roadbook:  
<http://www.xdappfactory.com/wp/>
- Schilit, B. N., LaMarca, A., Borriello, G., Griswold, W. G., McDonald, D., Lazowska, E., et al. (2003). *Challenge: Ubiquitous Location-Aware Computing and the "Placc Lab" Initiative*. (pp. 56-62). Seattle: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots.
- Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Laerhoven, K. V., & Velde, W. V. (1999). *Advanced interaction in context*. (pp. 2-9). Karlsruhe, Germany: Proceedings of First International Symposium on Handheld and Ubiquitous Computing.
- Seidler, K. ', Vogelgesang, K., Rueda, B., & Ridruejo, D. L. (s.d.). *Friends, Apache*. Obtido em junho de 2014, de XAMPP: [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)
- Silva, H. D. (2006). *Precisão, Exatidão, e a Terminologia das Medições*. (pp. 8-12). São Paulo: Universidade Estadual Paulista.
- Silva, R. J. (2013). *Aplicação Móvel Android de Apoio a Percursos Pedestres Outdoor*. (pp. 6-24). Lisboa: Instituto Superior Técnico.
- Simões, J. (s.d.). *Gpstrip*. Obtido em abril de 2014, de <http://jsimoes.selfip.com/gpstrip/?file=gps-road-book>
- Sousa, C. R. (2005). *Interferidores de GPS: Análise do sistema e de potenciais fontes de interferência*. (pp. 53-71). Rio de Janeiro: Instituto Militar de Engenharia.
- SQLite*. (s.d.). Obtido em junho de 2014, de Distinctive Features Of SQLite:  
<https://www.sqlite.org/different.html>
- Totem. (s.d.). *Totem - Apuração e Cronometragem - Computadores de Bordo*. Obtido em maio de 2014, de Road Book Digital: <http://www.rallyvirtual.com.br/sitenovo/>
- TripY. (s.d.). *TripY*. Obtido em abril de 2014, de The GPS for memorable trips:  
<http://www.tripy.eu/en>

*Tulip - Roadbook Editor for iPad.* (s.d.). Obtido em 2014 de maio, de <http://www.appato.com/long-haul-software/tulip-roadbook-editor-for-ipad/>

## **Anexo A**

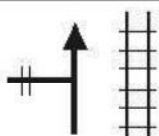
### **Exemplo de um roadbook tradicional em papel**

# Rota da Neve

**17/12/2011**  
**Road-Book 2**

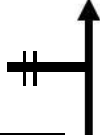

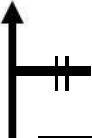
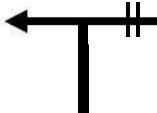

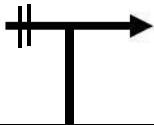
<b>Realizada por:</b> Clube Latitude40	
<b>Dificuldade:</b> Média/Baixa	<b>Distância total:</b> 57 km
<b>Origem / Destino:</b> Videmonte --> Guarda	
<b>Comentários:</b> Percurso agradável em zonas do Parque Natural da Serra da Estrela	

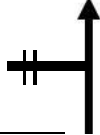
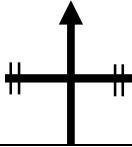
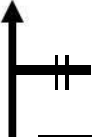
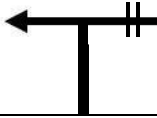

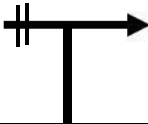
Distancia quilométrica desde o início

Nº	KM. T	KM. P.	Símbolo	Informação
21	15,6	3,6		P.P. Seque paralelo linha comboio

Nº de ordem na folha

Distancia em quilómetros desde a última nota

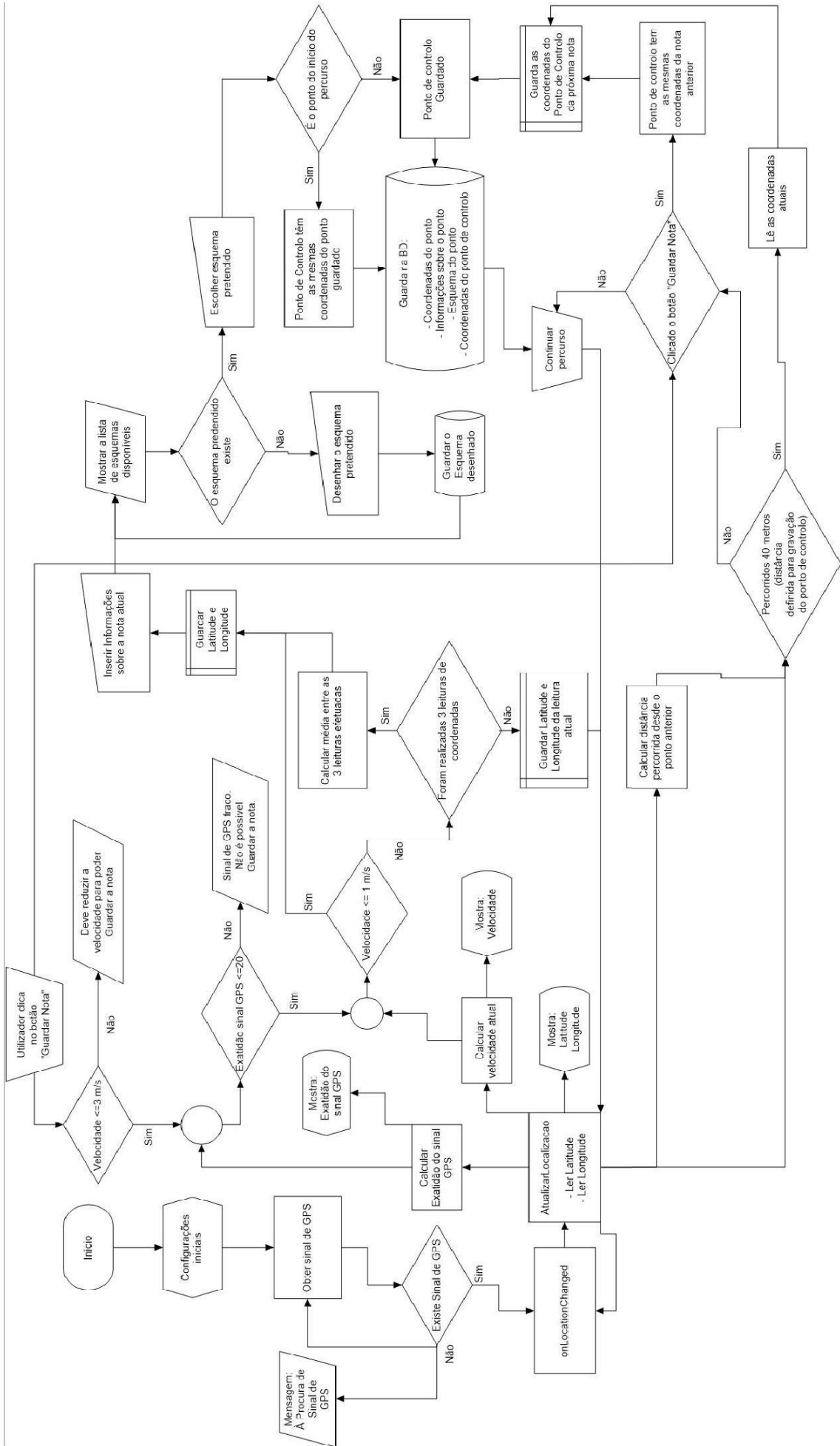
Nº	KM. T.	KM. P.	Símbolo	Informação
<b>1</b>	0,00	0,000		
<b>2</b>	0,88	0,879		
<b>3</b>	1,02	0,136		
<b>4</b>	1,69	0,671		Seguir PP
<b>5</b>	3,46	1,770		
<b>6</b>	3,61	0,154		Seguir PP

Nº	KM. T.	KM. P.	Símbolo	Informação
<b>7</b>	4,79	1,180		
<b>8</b>	5,31	0,515		
<b>9</b>	5,69	0,385		
<b>10</b>	6,88	0,190		Seguir PP
<b>11</b>	8,55	1,670		
<b>12</b>	9,35	0,803		Seguir PP
...				



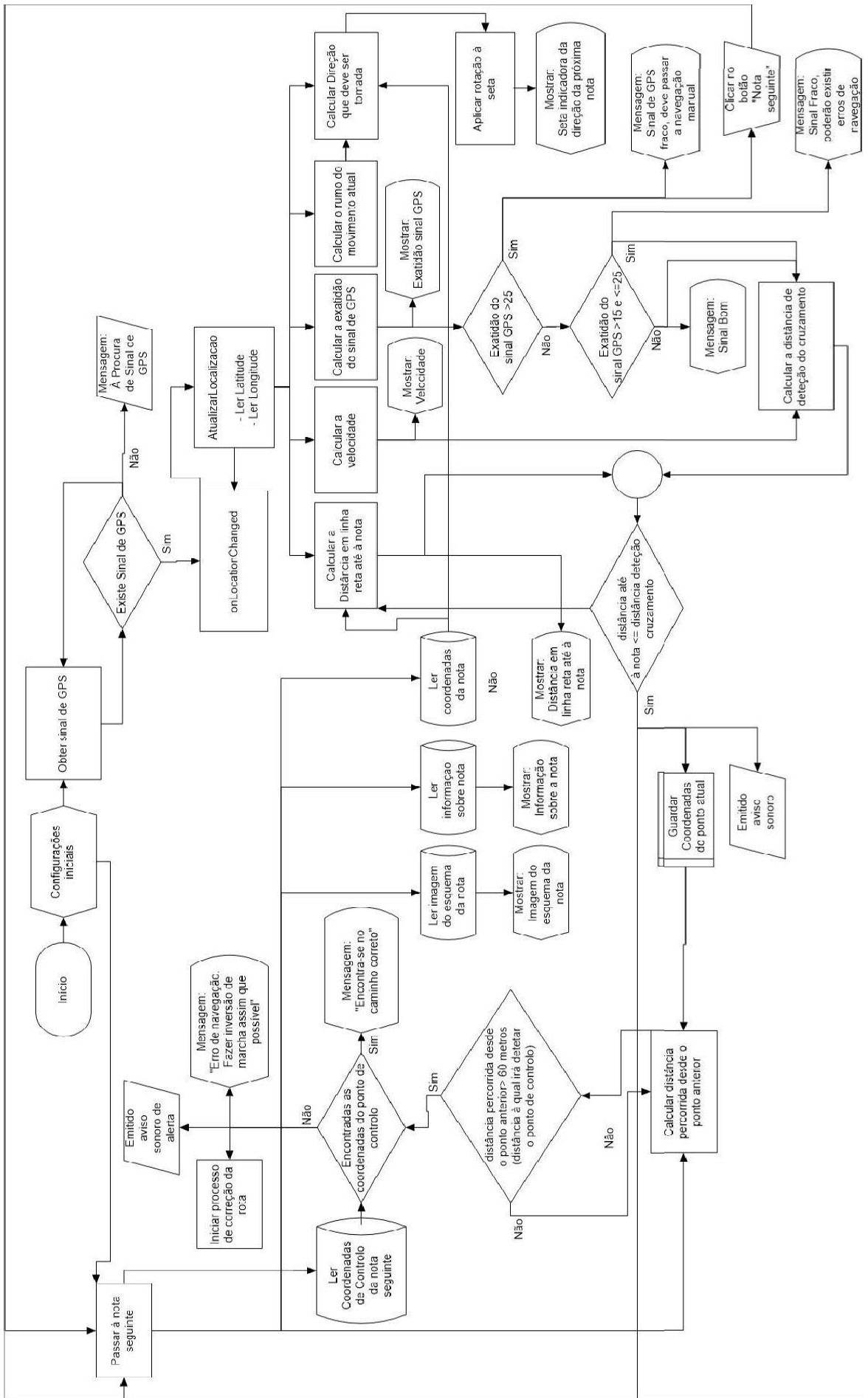
## **Anexo B**

### **Fluxograma do módulo de gravação das notas dos *roadbooks***



## **Anexo C**

### **Fluxograma do módulo de navegação de *roadbooks***



## **Anexo D**

# **Heurísticas de Nielsen**

**Heurística 1 - Visibilidade do estado do sistema**

O sistema deve sempre manter o utilizador informado a respeito do que está a acontecer, por meio de feedback apropriado em tempo razoável.

**Heurística 2 - Equivalência entre o sistema e o mundo real**

O sistema deve falar a linguagem do utilizador, com palavras, frases e conceitos familiares ao utilizador, ao invés de utilizar termos orientados ao sistema. Deve-se seguir convenções do mundo real, mostrando a informação numa ordem natural e lógica.

**Heurística 3 - Controlo do utilizador e liberdade**

Os utilizadores poderão escolher opções do sistema por engano e irão necessitar de uma “saída de emergência” bem definida para deixarem o estado não desejado sem terem de passar por um extenso diálogo.

**Heurística 4 - Consistência e padrões**

Os utilizadores não devem duvidar se palavras, situações, ou ações diferentes significam a mesma coisa. Devem-se seguir as convenções da plataforma. Usar palavras de forma consistente no conteúdo e nos botões.

**Heurística 5 - Prevenção de erro**

Muito melhor que boas mensagens de erro é um projeto cuidadoso que, em primeiro lugar, previna a ocorrência de problemas através de orientação e apresentação de recursos que facilitem a navegação.

**Heurística 6 - Reconhecer ao invés de relembrar**

Deve-se tornar objetos, ações e opções visíveis, diminuindo a necessidade de o utilizador se lembrar de muitas coisas. O utilizador não deve ter que se lembrar de informações de uma parte do diálogo numa outra parte. Instruções para o uso do sistema devem estar visíveis ou facilmente recuperáveis sempre que necessário.

**Heurística 7 - Flexibilidade e eficiência de utilização**

Criação de aceleradores de tarefas que passem despercebidos a utilizadores inexperientes mas que possam aumentar a velocidade de interação, tanto a utilizadores experientes como inexperientes.

**Heurística 8 - Estética e *design* minimalista**

Os diálogos não devem conter informação que seja irrelevante ou raramente necessária. Qualquer informação irrelevante num diálogo vai competir com a informação relevante e diminuir a sua visibilidade.

**Heurística 9 - Auxiliar os utilizadores a reconhecer, diagnosticar e recuperar ações erradas**

As mensagens de erro devem ser expressas em linguagem clara (sem códigos), indicar precisamente o problema e sugerir construtivamente uma solução.

**Heurística 10 - Ajuda e documentação**

Ainda que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Qualquer informação deste tipo deve ser fácil de procurar, ser focada na tarefa do utilizador, relacionar passos concretos a serem desenvolvidos e não ser muito longa.