

**TocáMexer: Aplicação Móvel de Monitorização de
Actividade Física em Contexto Clínico com Recurso a
GNSS**

Bernardo Miguel Antunes Carapito

Dissertação para obtenção do Grau de Mestre em

**Mestrado Integrado em Engenharia Electrotécnica e de
Computadores**

Orientador: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino

Júri

Presidente: Prof. Paulo Luís Serras Lobato Correia
Orientador: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Vogal: Prof. Francisco António Bucho Cercas

Novembro de 2021

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Agradecimentos

Em primeiro lugar, agradeço ao meu Professor orientador, Prof. José Eduardo Charters Ribeiro da Cunha Sanguino, pela proposta de realização desta dissertação que me fez descobrir o mundo da programação de aplicações móveis que certamente marcará para sempre a minha vida pessoal e profissional. Além disso, quero agradecer também pela disponibilidade, conselhos e apoio incansável em todas as fases da realização deste trabalho.

Ao meu irmão Diogo, pela contribuição notável neste trabalho enquanto interno de Medicina Geral e Familiar, nomeadamente na identificação das necessidades e requisitos da aplicação móvel para fins médicos.

Aos meus pais, que me apoiaram incessantemente em todo o meu percurso académico, que me proporcionaram as melhores condições para a realização desta dissertação e que colaboraram na teste e desenvolvimento da aplicação ao longo do tempo.

Aos meus amigos e colegas, que me ajudaram em todo o meu percurso universitário e que, apesar da situação pandémica, estiveram sempre disponíveis e compreensíveis em todos os momentos.

Ao Instituto de Telecomunicações pelos recursos e condições disponibilizadas durante a realização deste trabalho.

Por fim, agradeço a todos os meus professores e ao Instituto Superior Técnico pelos conhecimentos que me transmitiram e por toda a experiência positiva que tive ao longo do meu percurso académico.

Abstract

Nowadays, physical exercise is considered an effective way to prevent numerous health problems, like cardiovascular diseases, strokes, hypertension or diabetes. The benefits resulting from physical exercise make the prescription in a medical context a vital method for health control and improvement of the population's quality of life.

This thesis will address the development and the publication process of a mobile application on Android that tracks physical activity using GNSS. This application is designed especially for a medical context, where health professionals have the possibility of effectively prescribe a physical exercise plan that appropriately addresses the needs and limitations of the patient. This application also allows the analysis of the performance and the consequent adjustment of the plan according to the evolution of the user. This data can be shared between devices by using a QR Code based communication system.

This mobile application was developed in Android Studio using Java and was based on an MVP (Model View Presenter) architecture pattern.

The results revealed the practicability of the technology used for the recording of physical activity like walking or running. However, in an urban environment, there are some vulnerabilities in the data acquisition but they do not jeopardize the main goal of this application.

Keywords

Physical Activity; Health; GNSS; Mobile Application; Android; QR Code

Resumo

Atualmente, a prática da atividade física é considerada como um meio eficaz para prevenir e combater inúmeros problemas relacionados com a saúde das pessoas, desde doenças cardiovasculares, AVCs, hipertensão ou diabetes. Os benefícios decorrentes da atividade física tornam a prescrição de exercício físico em contexto médico um instrumento vital para o controlo da saúde e melhoria da qualidade de vida da população.

Nesta dissertação será abordado o desenvolvimento e publicação de uma aplicação móvel em Android que realiza o registo e monitorização de atividade física usando GNSS. Esta aplicação foi desenhada especialmente para o contexto médico, onde o profissional de saúde tem a possibilidade de prescrever eficazmente um plano de exercício físico adequado às necessidades e limitações do paciente. No seguimento do acompanhamento do utente, esta aplicação permite ao profissional de saúde analisar os dados estatísticos de desempenho e ajustá-los de acordo com a evolução do utente. O acesso aos dados é efetuado por partilha utilizando a tecnologia de QR Code.

Esta aplicação móvel foi desenvolvida em Android Studio utilizando Java e teve como base um padrão de arquitetura MVP (Model View Presenter).

Os resultados obtidos revelaram a viabilidade da tecnologia de localização utilizada para a correta gravação de dados de uma atividade física como andar ou correr. Porém, em ambiente urbano, existem algumas vulnerabilidades na recolha de dados que não põem em causa o principal objetivo da aplicação.

Palavras Chave

Atividade Física; Saúde; GNSS; Aplicação Móvel; Android; QR Code;

Conteúdo

1	Introdução	1
1.1	Motivação	3
1.2	Objetivo	3
1.3	Estrutura do Documento	4
2	Estado de Arte	7
2.1	Aplicações de Monitorização de Atividade Física	9
2.2	Sistemas de Navegação para Dispositivos Móveis	11
3	Tecnologias	13
3.1	Sistemas GNSS	15
3.1.1	Definição do Sistema	15
3.1.2	Segmentos do Sistema	15
3.1.3	Sistemas de Coordenadas e de Tempo	16
3.2	Sistema de Coordenadas de Referência	17
3.2.1	Figura da Terra	17
3.2.2	Sistemas de Coordenadas	18
3.2.3	GNSS e o Sistema de Coordenadas	19
3.2.3.A	Conversão para Coordenadas Cartesianas	19
3.2.3.B	Conversão para a Altitude Geóide	19
3.3	QR Code	21
3.4	Desenvolvimento em Android	22
3.4.1	Linguagem e Ambiente de Desenvolvimento	22
3.4.2	Elementos de Desenvolvimento	22
3.4.3	Padrão de Arquitectura	24
4	Implementação Lógica	27
4.1	Aquisição e Tratamento dos Dados de Localização	29
4.2	Definição de Objetivos	31
4.3	Comunicação de Dados	31

4.3.1	Partilha de Dados	32
4.3.2	Receção de Dados	35
5	Aplicação	37
5.1	Arquitetura Geral	39
5.2	Interface	41
5.2.1	Página Inicial	41
5.2.2	Atividade	43
5.2.3	Histórico	46
5.2.4	Perfil	47
5.2.5	Partilha de Dados	48
5.2.6	Scan de Dados	49
5.2.7	Definições	50
6	Testes ao Sistema	53
6.1	Teste de Intervalo de Aquisição de Dados	55
6.1.1	Teste em Ritmo de Caminhada	56
6.1.2	Teste em Ambiente de Corrida	58
6.1.3	Conclusão	60
6.2	Teste de Precisão	61
6.2.1	Testes em Ambiente Rural e Urbano	61
6.2.2	Conclusão	63
7	Publicação da Aplicação	65
8	Conclusão	73
	Bibliografia	77

Lista de Figuras

2.1	Interface de pré-gravação atividade da aplicação Strava	9
2.2	Interface de gravação de dados da aplicação Strava	9
2.3	Interface de gravação de atividade da aplicação Running App	10
2.4	Interface de gravação de atividade da aplicação Map My Run	10
3.1	Representação da constelação de satélites GPS em órbita da Terra [1]	16
3.2	Representação gráfica do modelo elipsoide em 2D	17
3.3	Modelo geoide da figura da Terra [2]	18
3.4	Grelha de interpolação bilinear - adaptado de [3]	20
3.5	Representação da altitude em relação ao elipsoide, altitude geoide e altitude do geoide em relação ao elipsoide - adaptado de [4]	21
3.6	Exemplo de um QR Code com link para o website do Instituto Superior Técnico (IST)	21
3.7	Diagrama representativo da interação entre uma Atividade e um Fragmento	23
3.8	Diagrama da arquitetura MVP	25
4.1	Código utilizado para efetuar a recolha periódica dos dados de localização	29
4.2	Processo de junção de atividades	33
4.3	Diagrama de Venn	33
4.4	Processo de codificação do QR Code	34
4.5	Código utilizado para efetuar o processamento e codificação do QR Code	35
4.6	Processo de descodificação do QR Code	35
4.7	Código utilizado para efetuar a leitura do QR Code	36
4.8	Código utilizado para efetuar o processo de descodificação do QR Code	36
5.1	Interface da Splash Activity	40
5.2	Interface genérica da Main Activity	40
5.3	Menu de navegação lateral de uma interface genérica da Main Activity	40
5.4	Interface superior da Página Inicial	41

5.5	Interface inferior da Página Inicial	41
5.6	Interface do Progresso Semanal	42
5.7	Interface do Progresso Mensal	43
5.8	Janela da legenda do código de cores	43
5.9	Interface inicial da Atividade	44
5.10	Interface com uma atividade a decorrer	44
5.11	Interface do mapa da atividade	45
5.12	Interface superior do Histórico	46
5.13	Interface inferior do Histórico	46
5.14	Interface de Armazenamento	47
5.15	Interface do Perfil	48
5.16	Interface da Partilha de dados	49
5.17	Interface do Scan de Dados - scan do QR Code	50
5.18	Interface do Scan de Dados - relatório de dados	50
5.19	Interface superior das Definições	51
5.20	Interface inferior das Definições	51
6.1	Fotografia do ambiente de teste na zona da Batalha	55
6.2	Percurso teórico percorrido	55
6.3	Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 1 segundo	56
6.4	Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 2 segundos	56
6.5	Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 4 segundos	57
6.6	Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 8 segundos	57
6.7	Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 1 segundo	58
6.8	Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 2 segundos	58
6.9	Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados 4 segundos	59
6.10	Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 8 segundos	59
6.11	Fotografia do ambiente de teste no centro da cidade de Leiria	61
6.12	Histograma de distribuição de dados de erro no percurso em ambiente rural	62

6.13	Histograma de distribuição de dados de erro no percurso em ambiente urbano	62
7.1	Interface inicial de criação da aplicação	67
7.2	Interface de configuração da aplicação	68
7.3	Interface com detalhes do relatório de pré-lançamento - estabilidade	69
7.4	Interface com detalhes do relatório de pré-lançamento - desempenho	69
7.5	Interface com detalhes do relatório de pré-lançamento - acessibilidade	70
7.6	Interface com detalhes do relatório de pré-lançamento - segurança e fidedignidade	70
7.7	Ficha de Loja da aplicação TocáMexer	71

Lista de Tabelas

6.1	Informação adicional sobre os testes efetuados em ritmo de caminhada	57
6.2	Informação adicional sobre os testes efetuados em ritmo de corrida	59

Acrónimos

API	Application Programming Interface
AVC	Acidente Vascular Cerebral
DGS	Direção Geral de Saúde
EGM	Earth Gravity Model
FOC	Full Operational Capability
IST	Instituto Superior Técnico
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
MVP	Model View Presenter
NFC	Near Field Communication
OMS	Organização Mundial da Saúde
QR	Quick Response
SDK	Software Development Kit
WGS84	World Geodetic System 1984
XML	Extensible Markup Language

1

Introdução

Conteúdo

1.1	Motivação	3
1.2	Objetivo	3
1.3	Estrutura do Documento	4

1.1 Motivação

Segundo a Organização Mundial da Saúde (OMS) [5], a prática regular de atividade física oferece benefícios significativos para a saúde do ser humano. Atividade física pode ser definida como qualquer movimento corporal produzido pelos músculos associado a um consumo de energia. Andar e correr são algumas das formas mais comuns de prática da atividade física que podem oferecer múltiplos benefícios para a saúde. A atividade física pode atuar como fator de proteção para doenças cardiovasculares, Acidentes Vasculares Cerebrais (AVCs), hipertensão, diabetes e alguns tipos de cancro. Além disso, esta também está associada a uma melhoria da saúde mental, atraso do desenvolvimento de demência e melhoria da qualidade de vida e bem estar.

Segundo as recomendações da Direção Geral de Saúde (DGS) [6] e do U.S. Preventive Services Task Force [7], cada adulto deve procurar acumular por semana 150 minutos de atividade física moderada ou 75 minutos de atividade vigorosa.

Todos estes benefícios mencionados tornam a prescrição de exercício físico em ambiente clínico um instrumento importante para o controlo da saúde e melhoria da qualidade de vida. No entanto, existem alguns obstáculos que impedem o sucesso da mesma. Por um lado, existem dificuldades tanto na transmissão da informação sobre a prescrição pretendida como na verificação do cumprimento da mesma. Por outro lado, este tipo de prescrições diferencia-se em muito de uma prescrição clássica de um medicamento. Ao passo que uma prescrição de um medicamento é muitas vezes uma indicação médica mais simples, rápida e com prazo de execução finito, uma prescrição de um tipo de atividade física como andar ou correr está geralmente associada a uma tarefa que consome um período significativo de tempo e que deve ser realizado de forma regular e a longo prazo.

A identificação destes problemas levanta a necessidade de criar uma solução que facilite a comunicação médico-paciente e ajude a garantir de forma consensual uma prescrição e monitorização eficaz da prática de atividade física.

1.2 Objetivo

O principal objectivo desta tese é o desenvolvimento de uma aplicação móvel em Android que efetue o registo e permita a monitorização de atividade física (andar ou correr). O registo da atividade física pretende-se que seja efetuado através da recolha de dados de localização que, após internamente processados, gera múltiplas estatísticas gerais de atividade como a distância percorrida, a duração da atividade e a rapidez média.

Outro objectivo deste projecto é integrar na aplicação funcionalidades que ajudem a monitorizar e a personalizar a atividade a realizar pelo utilizador consoante o seu perfil e os objetivos estabelecidos. Assim sendo, pretende-se criar uma interface onde seja possível personalizar o tipo de objectivo pre-

tendido (espacial ou temporal), a distribuição semanal dos mesmos e a intensidade esperada na sua atividade.

Pretende-se também criar um mecanismo que permita a partilha de dados de atividade. Esta funcionalidade tem a intenção de responder à necessidade de existir um processo simples e eficaz da partilha de dados de desempenho com os profissionais de saúde.

Ademais, pretende-se desenvolver uma aplicação com uma interface acessível e intuitiva. Em primeiro lugar, este requisito é motivado pela necessidade de desenvolver uma aplicação inclusiva e de fácil utilização para qualquer utilizador que possua um smartphone. Por outro lado, existe também a necessidade de criar uma interface que facilite a interpretação de dados por parte dos profissionais de saúde.

Estabeleceu-se como objectivo final a disponibilização pública da aplicação na loja oficial de aplicações da Android (Google Play Store).

Este projecto conta com o apoio e colaboração de um médico interno da especialidade de Medicina Geral e Familiar. Esta parceria tem como principal propósito a partilha de conhecimentos necessários de modo a criar um produto final adequado às necessidades do contexto de utilização clínica. Além de serem discutidos os objetivos e requisitos iniciais da aplicação, pretende-se que haja uma constante comunicação ao longo de todas as fases do trabalho.

1.3 Estrutura do Documento

Esta dissertação é composta por 8 capítulos, em primeiro lugar o capítulo introdutório em que se faz uma abordagem sobre a motivação, os objetivos e a estrutura da tese.

No capítulo 2 é efetuado o estudo sobre algumas das aplicações de monitorização de atividade física mais populares. Neste estudo, são relatadas as características principais e alguns dos aspetos em falta, tendo as necessidades sido identificadas na secção anterior de Motivação. É também feito um estudo geral sobre os atuais sistemas de navegação e a respetiva importância no funcionamento da sociedade dos dias de hoje.

No capítulo 3 é elaborada uma breve análise sobre alguns dos aspetos teóricos utilizados no desenvolvimento do projecto. Em primeiro lugar, é efetuado um estudo sobre as características do sistema Global Navigation Satellite Systems (GNSS) e dos respetivos sistemas de referência de coordenadas. Além disso, é feita uma abordagem sobre o princípio de funcionamento de um QR Code que será utilizado como mecanismo de partilha de dados na aplicação. Por fim, é elaborada uma análise sobre os aspetos básicos da programação em Android e os elementos de arquitetura geralmente utilizados para a construção de uma aplicação estável, estruturada e com bom desempenho.

O capítulo 4 aborda os principais aspetos relacionados com o funcionamento geral da aplicação,

nomeadamente o processo de aquisição de dados de localização, os mecanismos de armazenamento de dados e o sistema de comunicação.

O capítulo 5 contém uma descrição sobre a organização arquitetónica da aplicação, além de uma apresentação sobre a interface do utilizador e as respetivas funcionalidades associadas.

No capítulo 6 é elaborada uma análise sobre o desempenho da aplicação no que toca à aquisição de dados de localização. Esta análise tem como principal objectivo testar a aplicação em ambiente quotidiano e identificar pontos fracos e pontos fortes.

No capítulo 7 contém uma breve abordagem sobre o processo habitual de submissão da aplicação na Google Play Store.

No capítulo 8 estão reunidas as conclusões sobre o trabalho efetuado e algumas considerações finais sobre um possível trabalho futuro que pode ser elaborado nesta temática.

2

Estado de Arte

Conteúdo

2.1	Aplicações de Monitorização de Atividade Física	9
2.2	Sistemas de Navegação para Dispositivos Móveis	11

2.1 Aplicações de Monitorização de Atividade Física

Existem inúmeras aplicações móveis de monitorização de atividade física. Um dos exemplos mais populares a nível mundial é o “Strava”, uma aplicação que regista vários tipos de atividade física, como corridas, caminhadas ou percursos de bicicleta.

Esta aplicação está bem estabelecida no mercado das aplicações muito por conta do variado número de funcionalidades. Nas figuras 2.1 e 2.2 é possível verificar algumas partes da interface desta aplicação. A primeira e principal funcionalidade é o registo de uma atividade física usando o recetor GNSS, onde é possível durante a atividade obter, por exemplo, estatísticas como o tempo, distância e ritmo médio. Em simultâneo, é possível acompanhar em tempo real o progresso da atividade num mapa. No final, a atividade pode ser gravada e posteriormente revista numa interface de histórico.

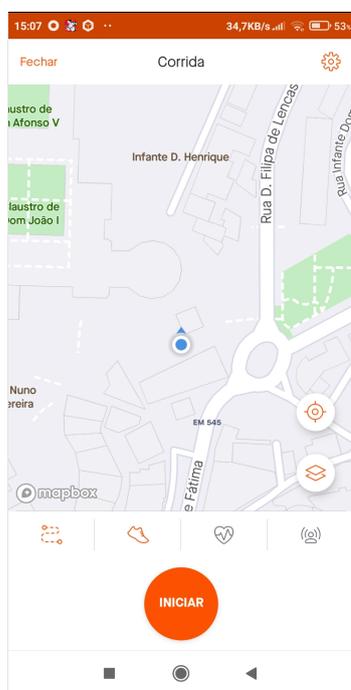


Figura 2.1: Interface de pré-gravação atividade da aplicação Strava

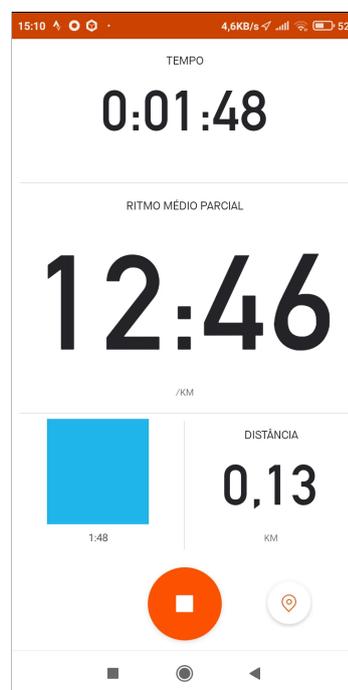


Figura 2.2: Interface de gravação de dados da aplicação Strava

Outra das funcionalidades em destaque nesta aplicação são os desafios pessoais. Esta função permite participar em desafios criados mensalmente e têm como objectivo incentivar a atividade do utilizador. Estes desafios são criados por terceiros e apenas é possível criar desafios pessoais caso tenha acesso à versão paga da aplicação.

Além disso, existe uma componente social onde é possível partilhar a atividade pessoal com outras pessoas e desafiá-las para competições amigáveis.

Para além desta aplicação, existem outras que partilham algumas das mesmas funcionalidades.

A aplicação “Running App” (figura 2.3) faz o registo de atividade usando o recetor GNSS, permite escolher objetivos de actividade e apresenta algumas estatísticas semanais e mensais do utilizador. A aplicação “Map My Run” (figura 2.4) efetua também o registo da atividade usando GPS, mostra algumas estatísticas gerais de atividade e tem uma componente social onde é possível partilhar a atividade pessoal com terceiros.

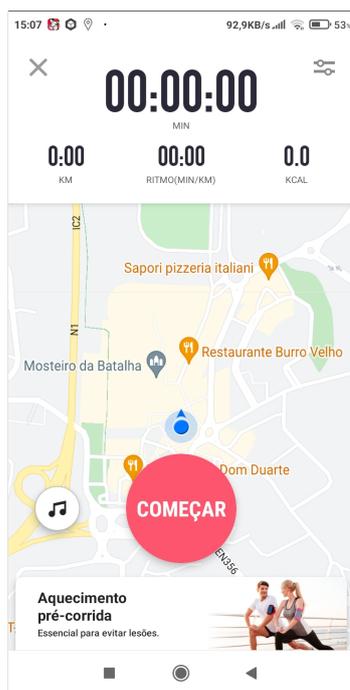


Figura 2.3: Interface de gravação de atividade da aplicação Running App



Figura 2.4: Interface de gravação de atividade da aplicação Map My Run

Porém, existem alguns aspetos que tornam estas aplicações menos adequadas para o contexto e motivações apresentadas no capítulo anterior.

Em primeiro lugar o “Strava” e o “Map My Run” requerem o preenchimento obrigatório de um cadastro para utilizar a aplicação. Para além disso, as três aplicações mencionadas têm interfaces e um grande número de funcionalidades complexas. Estes aspetos podem representar um obstáculo para utilizadores menos experientes ou com o único objectivo de efetuar a gravação e registo da sua atividade física.

Além disso, apesar de o “Strava” e o “Map My Run” terem uma função onde é possível partilhar os seus dados de atividade com terceiros, estas interfaces não estão especialmente desenvolvidas para serem utilizadas num contexto clínico. Por um lado, estas funcionalidades estão desenhadas para um contexto mais social e relaxado, onde existe uma relação bidirecional na partilha de dados. Por outro, o formato de apresentação de dados é pouco focado nas informações essenciais que são necessárias serem partilhadas num contexto médico.

2.2 Sistemas de Navegação para Dispositivos Móveis

O Global Navigation Satellite Systems (GNSS) é uma denominação geral para os vários sistemas de navegação por satélite que permite aos utilizadores com recetores compatíveis determinar a sua posição e tempo através do processamento do sinal enviado por satélites da rede. Estes sinais têm proveniência num variado número de constelações de diversos países. Entre os principais incluem-se o GPS, Galileo, GLONASS e Beidou [8].

O Global Positioning System (GPS) é um sistema desenvolvido pelos Estados Unidos da América, tendo o seu primeiro satélite sido lançado na década de 70. Atualmente este sistema conta com 30 satélites no total, dos quais 24 satélites estão ativos. Desde 1995 que este sistema está em Full Operational Capability (FOC), ou seja, com capacidade operacional total [9].

Galileo é um sistema construído pela União Europeia e conta atualmente com 26 satélites em órbita, dos quais 24 satélites estão ativos. Os primeiros satélites foram lançados em 2011 e prevê-se que sejam atingidos 30 satélites no total até à FOC [10].

O GLONASS é um sistema operado pela Rússia e é constituído por 24 satélites ativos. À semelhança do GPS, este sistema existe desde os anos 70 e em 2011 atingiu o estatuto de FOC [11].

Por fim, o Beidou foi desenvolvido pela China e tem capacidade operacional total desde 2020. Este sistema é atualmente constituído por 44 satélites, sendo esta a constelação mais numerosa dentro de todos os sistemas mencionados [12].

Um recetor de navegação suportado em GNSS, nomeadamente um smartphone, pode não estar limitado à utilização de um sistema em específico, tanto que pode ter acesso a um conjunto variado de serviços GNSS. Este aspeto contribui para uma melhor disponibilidade e qualidade do serviço, o que conseqüentemente leva a uma melhor exatidão no cálculo da posição do dispositivo [8].

Além do GNSS, é ainda possível obter informação da localização usando Wi-Fi. Apesar de menos preciso, esta fonte pode ser utilizada como complemento ao sistema de posicionamento por satélite. O processamento destes sinais em Android é efetuado com recurso a uma Application Programming Interface (API) de localização da Google (Fused Location Provider API). Esta API efetua a receção do sinal de uma ou mais fontes e faz uma estimativa da localização do dispositivo. [13]

Num mundo cada vez mais digital, os sistemas de navegação tornaram-se num pilar fundamental para o normal funcionamento da sociedade dos dias de hoje. Em 2019, 6.4 mil milhões de dispositivos usavam GNSS e prevê-se que este número cresça para 9.6 mil milhões em 2029. Além disso, em 2019 os smartphones representavam aproximadamente 90% da quota dos dispositivos com recetor GNSS. Um dos tipos de utilizador mais usual é o cidadão comum, onde este sistema é utilizado maioritariamente como sistema de navegação e de localização. Além disso, este sistema é também utilizado como ferramenta em variados setores da sociedade mundial, como a agricultura, construção, transportes, exploração mineira, logística, militar, entre outros [8] [14].

3

Tecnologias

Conteúdo

3.1	Sistemas GNSS	15
3.2	Sistema de Coordenadas de Referência	17
3.3	QR Code	21
3.4	Desenvolvimento em Android	22

Este capítulo tem como objectivo inicial apresentar os conceitos básicos de funcionamento dos sistemas GNSS. Ademais, é efetuado um estudo sobre os modelos utilizados de representação da Terra, os sistemas de referência mais utilizados e respetiva relação com os sistemas GNSS. Além disso, é efetuada uma breve análise sobre algumas das características gerais do QR Code. Por fim, é feita uma análise geral sobre alguns dos aspetos mais importantes no desenvolvimento de aplicações em Android.

3.1 Sistemas GNSS

3.1.1 Definição do Sistema

Cada sistema Global Navigation Satellite Systems (GNSS) é composto por um conjunto de satélites que têm como função principal efetuar a transmissão de ondas de rádio para recetores que posteriormente possibilita a estimação da sua posição. Cada um destes sinais envia informação sobre a posição do satélite na rede e o respetivo instante de envio. Com esta informação, é calculado o tempo de propagação do envio da mensagem que conseqüentemente possibilita o cálculo da distância entre os dois pontos de comunicação. A informação conjunta de vários sinais permite a estimação da posição do recetor por trilateração, tal como a longitude, latitude e altitude em relação ao elipsoide com uma exatidão na ordem das unidades de metro. Além disso, este sistema fornece informação sobre o tempo com uma exatidão na ordem dos nanossegundos [15] [16] [17].

Um dos fatores que afeta diretamente a qualidade na obtenção de dados de localização através de satélite é o ambiente do recetor. Por um lado, locais a céu aberto sem edifícios nas proximidades são ideais para a receção do sinal GNSS. Porém, em ambientes urbanos com edifícios de elevada altura existem dois grandes constrangimentos: o bloqueio do campo de visão que leva à conseqüente redução de satélites visíveis e a interferência por multi-caminho. Em condições normais, o sinal enviado por um satélite propaga-se aproximadamente numa linha reta até ao recetor, sendo esta informação utilizada como assunção para a estimativa da posição do recetor em relação ao satélite. Porém, quando ocorre interferência por multi-caminho, ou seja, o sinal é refletido até chegar ao recetor, o tempo de propagação aumenta alterando conseqüentemente a estimativa da distância do recetor em relação ao satélite emissor, que por sua vez levará a um maior erro na estimação da sua posição. [16] [17].

3.1.2 Segmentos do Sistema

Estes sistemas estão tipicamente estruturados em três segmentos distintos: o do espaço, o de controlo e o do utilizador.

O segmento do espaço consiste no conjunto de satélites em atividade que transmitem sinais de

rádio para os dispositivos recetores que utilizam o sistema. Cada sistema GNSS tem associado o seu segmento espacial composto pela sua constelação. Entre eles incluem-se os sistemas GPS, GLO-NASS, BeiDou e Galileo. Estes satélites orbitam na zona da órbita terrestre média e estão distribuídos de forma uniforme em volta da Terra de forma a efetuar uma cobertura completa da mesma. Na figura 3.1 é possível visualizar uma representação esquemática da constelação de satélites GPS em órbita da Terra [16] [18].

O segmento de controlo consiste numa rede global de infraestruturas terrestres que tem como objetivo monitorizar e corrigir a atividade dos satélites da rede. Além de estimarem a informação atualizada sobre localização e sobre a integridade dos satélites, estas estações têm também a função de enviar e corrigir informações dos satélites como o acerto do relógio interno. [16] [18].

Por fim, o segmento do utilizador diz respeito a todo o universo de utilizadores possuem recetores do sinal de GNSS transmitido pelos satélites do sistema. Entre os utilizadores mais comuns enumeram-se o dispositivos móveis, automóveis e dispositivos militares [16] [18].

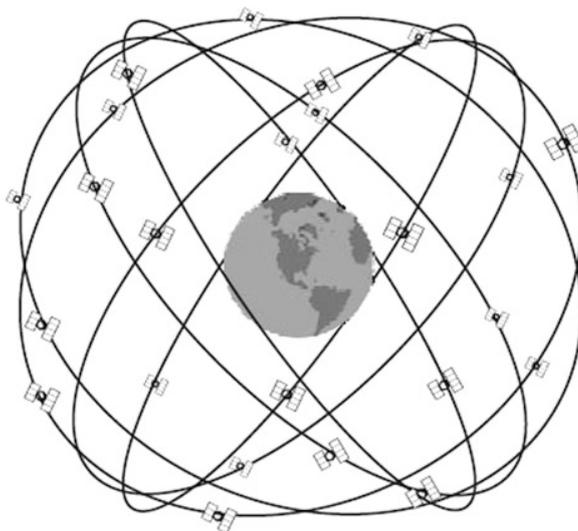


Figura 3.1: Representação da constelação de satélites GPS em órbita da Terra [1]

3.1.3 Sistemas de Coordenadas e de Tempo

O sistema GPS e a API de localização da Google utilizam como sistema de coordenadas de referência terrestre o World Geodetic System 1984 (WGS84). O centro da origem deste sistema está localizado no centro de massa da Terra e tem como base dois parâmetros de referência do modelo elipsoide da Terra: o semieixo maior ou raio equatorial (a) e o coeficiente de achatamento do elipsoide (f) [19] [20].

Além disso, o GPS tem um sistema dedicado de tempo (GPST) baseado em relógios atômicos situados nas estações de monitorização em Terra [19].

3.2 Sistema de Coordenadas de Referência

Um sistema de coordenadas de referência é um modelo que faz a caracterização, de forma local ou global, de um determinado ponto geográfico. Consoante a utilidade existem vários sistemas utilizados de forma a efetuar esta caracterização. Para entender cada um destes sistemas é necessário em primeiro lugar entender o complexo e irregular formato da Terra.

3.2.1 Figura da Terra

A geóide é uma superfície que define de forma aproximada o formato irregular da Terra. Este modelo tem como base um modelo mais simples de representação do formato terrestre chamado elipsoide terrestre. O elipsoide tem como base dois parâmetros de referência do modelo elipsoide da Terra: o semieixo maior ou raio equatorial (a) e o semieixo menor (b). Na figura 3.2 é possível observar uma representação esquemática deste modelo [19] [20].

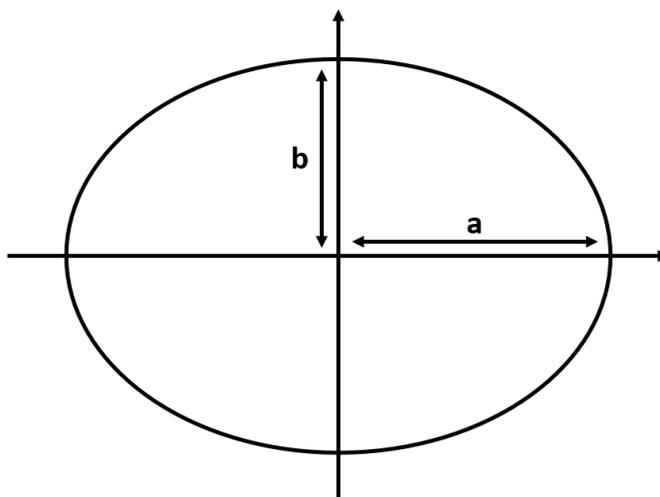


Figura 3.2: Representação gráfica do modelo elipsoide em 2D

Porém, a forma padrão de caracterização de um elipsoide assenta na utilização dos valores do semieixo maior e do coeficiente de achatamento do elipsoide (f). O coeficiente de achatamento pode ser calculado usando a seguinte expressão [19] [20]:

$$f = \frac{a - b}{a} \quad (3.1)$$

Ao longo da história foram propostos vários modelos elipsoides com diferentes combinações de

valores de a e f . Uma dessas versões do elipsoide é o modelo WGS84, determinado em 1984 e ainda hoje utilizado como modelo referência pelo sistema GPS. Para este modelo em específico são utilizados os seguintes valores de referência [19] [20]:

$$a = 6378137.0m \quad f = \frac{1}{298.257223563}$$

Todavia, o formato irregular da forma da Terra justifica a utilização de um modelo mais preciso como a geóide. O modelo geóide é uma versão corrigida ponto a ponto do modelo elipsoidal da Terra, ou seja, é descrita pela altitude dos seus pontos relativa ao elipsoide da Terra, formando assim a chamada ondulação do geóide. Na figura 3.3 é possível visualizar o formato hiperbolizado da superfície da Terra com base no modelo geóide [20].

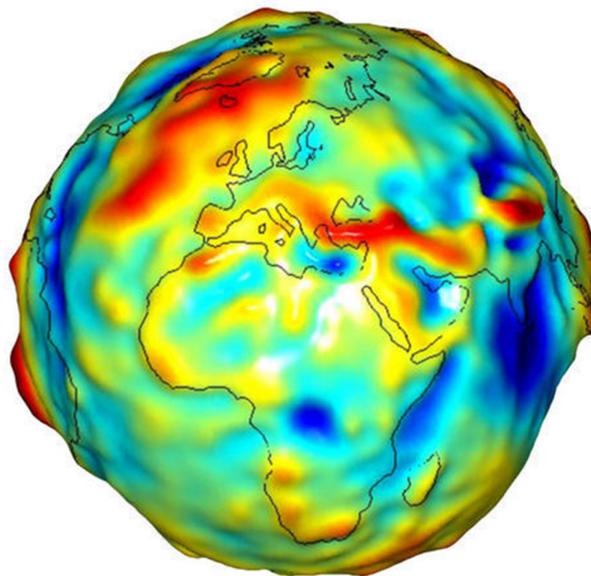


Figura 3.3: Modelo geóide da figura da Terra [2]

3.2.2 Sistemas de Coordenadas

Os sistemas de coordenadas são usados para descrever a posição de um determinado ponto no espaço. Dois dos sistemas importantes para caracterizar um determinado ponto na superfície terrestre são o sistema de coordenadas geodésicas e o sistema cartesiano [19] [20].

O sistema de coordenadas geodésicas caracteriza um determinado ponto através de três parâmetros: latitude geodésica (ϕ), longitude geodésica (λ) e a altitude geodésica (h). O sistema referencial elipsoidal é um dos sistemas que utiliza este sistema de coordenadas [19] [20].

Por outro lado, o sistema cartesiano de coordenadas terrestre tem como base um espaço tridimen-

sional cujo a origem situa-se no centro da Terra e utiliza as coordenadas x , y e z para caracterizar um ponto no espaço. Com este sistema é possível calcular de forma mais facilitada distâncias entre pontos através do método de distância tridimensional euclidiana [19].

3.2.3 GNSS e o Sistema de Coordenadas

Em smartphones Android, a estimação da localização do dispositivo é efetuada com recurso à API de localização da Google (Fused Location Provider API). Esta API efetua a receção e processamento das várias fontes de informação como os satélites da rede GNSS ou Wi-Fi e calcula a posição atualizada do dispositivo sob o referencial WGS84 com base em coordenadas geodésicas. No entanto, este sistema de coordenadas nem sempre se apresenta o mais adequado para todos os contextos, como por exemplo para o cálculo da distância entre dois pontos no espaço. Alternativamente, a utilização de um sistema alternativo como o cartesiano facilita o cálculo deste parâmetro. Adicionalmente, é igualmente necessária a conversão dos dados assentes no formato elipsoide do formato WGS84 para o modelo geoide [19] [20].

3.2.3.A Conversão para Coordenadas Cartesianas

O cálculo das coordenadas cartesianas (x , y e z) usando coordenadas geodésicas (ϕ , λ e h) é dado pelas expressões em 3.2. Como referido anteriormente, o f é o valor tabelado do coeficiente de achatamento do elipsoide segundo o modelo WGS84 e o R_N é o raio da curvatura no ponto que pode ser calculado tal como indicado na expressão 3.3 [19] [20].

$$\begin{aligned} x &= (R_N + h) \cos \phi \cos \lambda \\ y &= (R_N + h) \cos \phi \sin \lambda \\ z &= ((1 - f)^2 R_N + h) \sin \phi \end{aligned} \tag{3.2}$$

$$R_N = \frac{a}{\sqrt{1 - f(2 - f)(\sin \phi)^2}}. \tag{3.3}$$

3.2.3.B Conversão para a Altitude Geoide

A conversão de coordenadas com base no modelo elipsoide para o geoide é feito recorrendo a um modelo de aproximação designado de Earth Gravity Model (EGM). Este modelo reúne uma compilação de altitude em relação ao geoide em relação ao elipsoide WGS84 (G) de pontos distribuídos uniformemente numa grelha retangular que caracteriza a superfície terrestre tal como representado na figura 3.4.

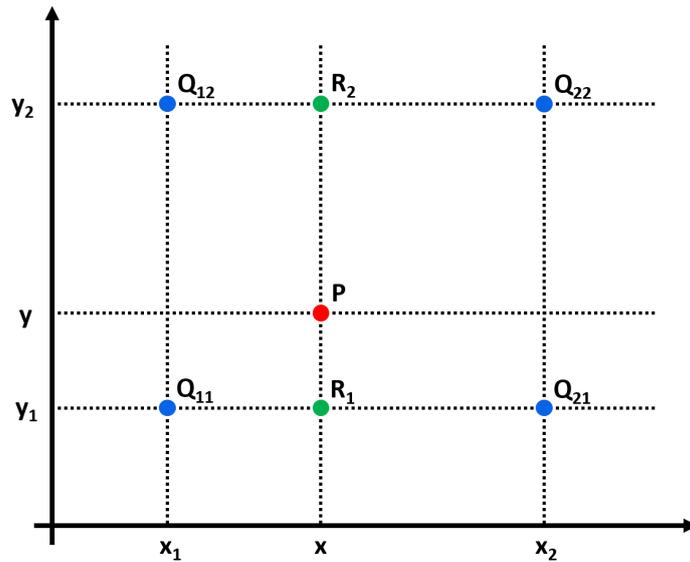


Figura 3.4: Grade de interpolação bilinear - adaptado de [3]

Este modelo está disponível em várias versões com grades de densidade de pontos distintas (varia entre 5', 2,5' e 1' para a versão EGM2008 do modelo). O G para qualquer ponto do globo é calculado usando interpolação bilinear na grade retangular de pontos. Escolhendo os quatro pontos mais próximos na grade Q_{11} , Q_{12} , Q_{21} e Q_{22} e os respectivos valores de G $f(Q_{11})$, $f(Q_{12})$, $f(Q_{21})$ e $f(Q_{22})$, utilizam-se as seguintes expressões para calcular o valor da $G = f(x, y)$ do ponto P [3]:

$$\begin{aligned}
 f(x, y_1) &= \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \\
 f(x, y_2) &= \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \\
 G = f(x, y) &= \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)
 \end{aligned} \tag{3.4}$$

Após este processo, é necessário efetuar a conversão da altitude relativa à superfície do elipsoide para a altitude em relação ao geóide. Na figura 3.5 é possível visualizar uma representação das três grandezas envolvidas neste cálculo: a altitude em relação ao elipsoide (h), a altitude em relação ao geóide (H) e a altitude do geóide em relação ao elipsoide (G). H pode ser assim determinado, de forma aproximada, pela seguinte expressão [19] [20]:

$$H = h - G \tag{3.5}$$

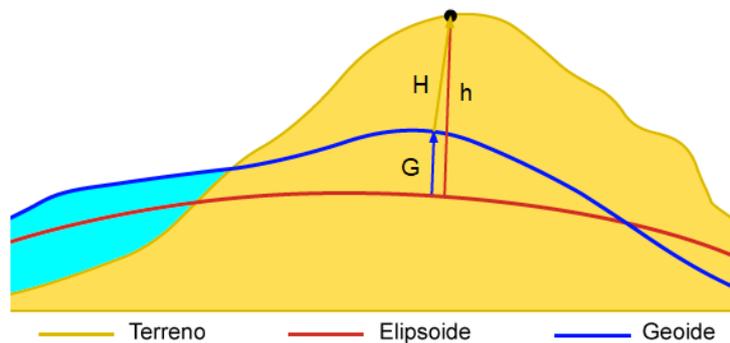


Figura 3.5: Representação da altitude em relação ao elipsoide, altitude geóide e altitude do geóide em relação ao elipsoide - adaptado de [4]

3.3 QR Code

Um Quick Response (QR) Code é um tipo de código de barras formado por uma matriz bidimensional. Esta tecnologia é normalmente utilizada com recurso a uma câmara de um dispositivo móvel e tem como intuito geral codificar e transmitir informação que pode ser posteriormente lida por outros dispositivos. Este código pode ser utilizado para codificar texto simples, URLs, contactos pessoais, entre outros. Além do uso mais simples e quotidiano, esta tecnologia é utilizada em múltiplos sectores da economia como a indústria, logística, transporte, saúde e publicidade [21] [22].

A capacidade de armazenamento deste código depende diretamente de dois fatores importantes: o grau de redundância e a densidade de informação máxima. Na figura 3.6 é possível observar um exemplo de um QR Code com o link para o website do Instituto Superior Técnico (IST).



Figura 3.6: Exemplo de um QR Code com link para o website do IST

Por variadas vezes a leitura da imagem é dificultada pela baixa qualidade da câmara, o ângulo e a luminosidade desadequada existente no momento da captura. Com isto, torna-se importante a existência deste mecanismo de redundância para facilitar a leitura deste código. Dependendo do que for preten-

dido, o grau de redundância varia normalmente entre 7 e 30%, sendo a capacidade de armazenamento consequentemente afetada [21] [22].

Além disso, a legibilidade do código é afetada pela densidade de informação do código. A partir de uma determinada densidade, o código torna-se pouco legível devido a fatores como a resolução do ecrã que transmite o código e a resolução de captura da câmara que faz a leitura do código [21] [22].

Consequentemente, existe uma capacidade máxima de armazenamento para o tamanho máximo do código que pode variar entre os 1276 e os 2956 bytes dependendo do nível de redundância pretendido [21] [22].

3.4 Desenvolvimento em Android

3.4.1 Linguagem e Ambiente de Desenvolvimento

O Android é um sistema operativo baseado em Linux desenvolvido para dispositivos móveis como telemóveis, tablets, smartwatches, televisões, entre outros dispositivos. Atualmente este sistema operativo detém 72,44 e 45,33% da quota de mercado em telemóveis e tablets, respetivamente, tornando-o assim num dos softwares mais importantes da sociedade atual [23].

À semelhança com outros sistemas operativos, o Android utiliza aplicações de forma a estender as funcionalidades gerais do dispositivo. Estas aplicações são primariamente desenvolvidas em Java ou em Kotlin e têm como base um sistema de desenvolvimento de software desenhado especificamente para este contexto denominado de Software Development Kit (SDK). Este ambiente de desenvolvimento contém ferramentas como um debugger, bibliotecas e simuladores de dispositivos android. Embora as ferramentas deste ambiente de desenvolvimento possam ser usadas através da linha de comando, uma das soluções mais populares é a utilização de um ambiente de desenvolvimento integrado como o Android Studio. Além de integrar as ferramentas do SDK, este software fornece uma interface gráfica que permite aos programadores construir as suas aplicações de forma mais rápida e eficiente [24].

3.4.2 Elementos de Desenvolvimento

À semelhança de outros tipos de software, uma aplicação em Android é constituída por vários componentes que em conjunto fazem parte de uma arquitetura geral da aplicação. Alguns dos elementos mais comuns são a Atividade, o Fragmento e o Serviço. Estes elementos arquitetónicos são importantes não só para uma correta estruturação da aplicação como para um melhor desempenho e gestão de recursos [24].

Uma Atividade é um componente obrigatório numa aplicação móvel que tem como função principal representar visualmente as interfaces da aplicação com a qual o utilizador pode interagir. Cada ativi-

dade faz a ponte entre a componente lógica da aplicação e o componente de layout que está associada a um ficheiro de layout Extensible Markup Language (XML) [24].

Um Fragmento é um componente reutilizável da Interface de Utilizador de uma aplicação que pode estar diretamente conectada a uma Atividade. Este elemento pode ocupar uma parte ou todo o ecrã da aplicação, sendo possível também cada atividade estar associada a mais do que um Fragmento. Esta associação é possível através da utilização de um elemento de Layout chamado de FrameLayout. Na figura 3.7 está presente um diagrama que representa a interação entre uma Atividade, um Fragmento e um FrameLayout [24].

Neste caso, a Atividade tem associada um FrameLayout no seu layout que é responsável por efetuar o processamento dos elementos de layout do Fragmento. Sempre que a atividade pretender utilizar outro Fragmento, esta pode associá-la ao FrameLayout, substituindo conseqüentemente o Fragmento anterior [24].

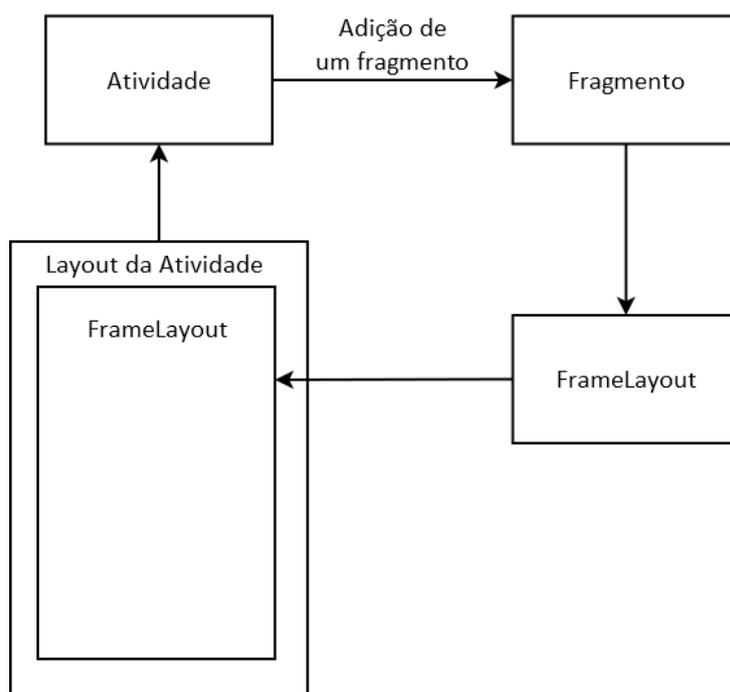


Figura 3.7: Diagrama representativo da interação entre uma Atividade e um Fragmento

Um Serviço é um componente que tem como função principal realizar operações de longa duração que não interagem diretamente com a interface de utilizador da aplicação. Além disso, uma característica importante deste componente é que este não está obrigatoriamente associado a uma atividade ou um Fragmento. Conseqüentemente, a aplicação pode alternar entre Atividades e Fragmentos sem o Serviço ser interrompido [24].

Existem dois tipos Serviços indispensáveis para o desenvolvimento de a maior parte das aplicações

em Android: o Serviço em primeiro plano e o Serviço em segundo plano. Um serviço em primeiro plano executa operações que são diretamente perceptíveis ao utilizador, como por exemplo um leitor de vídeo. Por outro lado, um serviço em segundo plano executa ações que não são diretamente perceptíveis ao utilizador, como por exemplo um serviço de mensagens que está constantemente em alerta sobre uma possível nova mensagem [24].

Além destes três elementos, é importante realçar que grande parte das aplicações necessitam de utilizar mecanismos de armazenamento de dados.

Um dos mecanismos nativos do ambiente Android é o SharedPreferences. Este componente é uma interface que permite aceder e gravar dados de preferência do utilizador. Este sistema de armazenamento de dados está estruturado sob o formato de chave-valor e garante a subsistência da informação mesmo quando a aplicação é encerrada [24].

Porém, para armazenar dados mais complexos são utilizadas base de dados. Um tipo de base de dados muito utilizado neste contexto é o MySQL. MySQL é um sistema de gestão de base de dados que utiliza SQL como interface. Este sistema caracteriza-se por ter um excelente desempenho, estabilidade e facilidade de manutenção [25].

3.4.3 Padrão de Arquitectura

No desenvolvimento de uma aplicação em Android, especialmente em projetos de grande complexidade, é recomendada a implementação de padrão de arquitetura de software adequada às necessidades da mesma. Um padrão de arquitetura é uma técnica de organização estrutural para software que efetua a separação de responsabilidades dos vários componentes do programa. Uma das vantagens na utilização desta técnica é a modularidade aos ficheiros do projecto. Esta técnica permite uma melhor manutenção e legibilidade das várias partes do código [24] [26].

Uma das arquiteturas existentes para o desenvolvimento móvel em Android é a arquitetura Model View Presenter (MVP). Esta arquitetura é composta por três componentes essenciais: o Model, o View e o Presenter. Na figura 3.8 apresenta um esquema com os vários componentes mencionados e as respetivas interações principais entre eles [26].

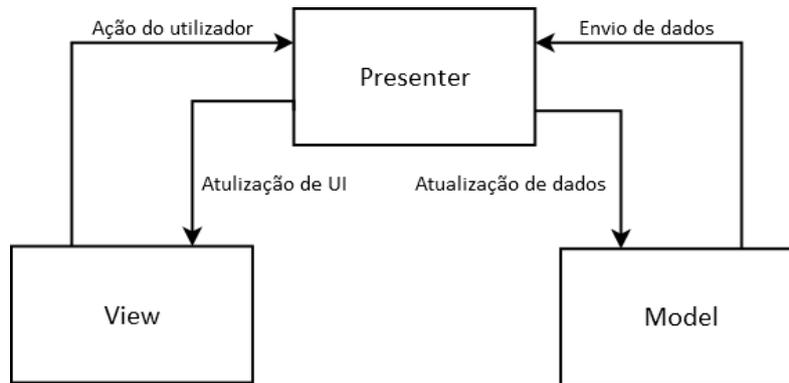


Figura 3.8: Diagrama da arquitetura MVP

O Model é o módulo responsável pelo armazenamento de dados. Este elemento faz a leitura e escrita dos dados, sejam estes localizados numa base de dados local ou numa API externa [26].

A View é o componente responsável por interagir com as componentes da interface do utilizador. Além de providenciar a visualização de dados, este elemento tem também a função de ler as ações do utilizador no dispositivo de modo a serem interpretadas pelo Presenter [26].

Por fim, o Presenter tem como papel principal efetuar a ponte entre o Model e o View. Quando necessário, o Presenter retira dados do Model que depois são enviados para o View para serem exibidos no ecrã do utilizador. O Presenter tem ainda a função de efetuar, quando necessário, todo o processamento lógico mais pesado da aplicação [26].

4

Implementação Lógica

Conteúdo

4.1 Aquisição e Tratamento dos Dados de Localização	29
4.2 Definição de Objetivos	31
4.3 Comunicação de Dados	31

Este capítulo tem como objectivo abordar alguns dos principais aspetos da implementação lógica da aplicação. Entre eles incluem-se o processo de aquisição e de interpretação de dados de localização, o sistema de armazenamento de dados e o processo de partilha e de receção de dados entre dois dispositivos.

4.1 Aquisição e Tratamento dos Dados de Localização

Uma das principais funcionalidades que se pretende implementar na aplicação é a gravação e o cálculo de estatísticas de uma atividade física, como a distância percorrida, a rapidez média, altitude e tempo. Uma das soluções possíveis passa pela aquisição periódica de dados de localização GNSS. Cada atualização contém um conjunto de dados referentes à localização do dispositivo, como a latitude, longitude, altitude, Timestamp, entre outras informações, obtidos com base na API de localização da Google - Fused Location Provider API. Na figura 4.1 está presente uma parte do código utilizado para efetuar configuração e receção dos dados obtidos com base nesta API. Em primeiro lugar, é efetuada a configuração inicial do pedido de obtenção de dados de localização. Numa fase seguinte é definido o *listener* de forma a efetuar a receção e processamento dos dados obtidos.

```
LocationRequest locationRequest = LocationRequest.create()
    .setInterval(LOCATION_REQUEST_INTERVAL)
    .setFastestInterval(LOCATION_REQUEST_FASTEST_INTERVAL)
    .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
    .setWaitForAccurateLocation(true);

fusedLocationProviderClient.requestLocationUpdates(locationRequest, new LocationCallback() {
    @Override
    public void onLocationResult(@NonNull LocationResult locationResult) {
        processLocationResult(locationResult)
    }
})
```

Figura 4.1: Código utilizado para efetuar a recolha periódica dos dados de localização

Para cada uma destas atualizações de localização é necessário efetuar algum processamento lógico antes de ser possível efetuar o cálculo das estatísticas pretendidas, nomeadamente a distância percorrida ou a altura em relação ao nível médio do mar (altitude em relação à geóide).

A determinação da distância percorrida numa atividade é efetuada calculando sucessivamente a distância entre os pontos obtidos pelo recetor GNSS. Esse cálculo é efetuado utilizando o método euclidiano para a distância entre pontos com coordenadas cartesianas. Deste modo, é necessário efetuar a conversão dos pontos fornecidos sobre o formato de coordenadas geodésicas para o formato cartesiano. Este processo é efetuado com base nas expressões 3.2 e 3.3.

Por outro lado, uma das informações obtidas pelo recetor GNSS é a altitude em relação ao elipsoide do modelo WGS84 no ponto do recetor. Por este motivo, é necessário efetuar a conversão da altitude

relativa ao elipsoide para a altitude em relação ao geoide. Esta conversão é feita com base na leitura do ficheiro EGM2008-5 que contém uma compilação de altitudes geoide em relação ao elipsoide. Os pontos da grelha estão espaçados de 5' em latitude e longitude, sendo que é necessário efetuar a interpolação bilinear para calcular o valor para um determinado ponto no globo. O cálculo da altitude geoide é obtido subtraindo à altitude elipsoide (fornecida pela pelo sinal GNSS [27]) a altura do geoide em relação ao elipsoide obtida pela interpolação bilinear.

Outro aspeto importante nesta parte da aplicação é garantir que a recolha de dados de localização é efetuada de forma consistente e sem interrupções. Numa aplicação deste tipo onde é normalmente feita a gravação de dados de localização durante períodos relativamente longos de tempo (na ordem da dezena de minutos), é vital que todo este processamento continue a ser feito mesmo quando a aplicação é colocada em segundo plano, ou seja, quando outra aplicação é aberta ou quando o utilizador bloqueia o dispositivo móvel.

Uma das soluções possíveis para resolver este problema seria colocar o código desta funcionalidade num Serviço com permissões para executar em segundo plano. Porém, solicitar a permissão de acesso à localização em segundo plano é um processo burocraticamente complexo no caso de se pretender publicar esta aplicação na Google Play Store. Estas práticas sucedem-se por forma a melhorar a privacidade e evitar o abuso de acesso à localização em segundo plano em casos onde a sua utilização não é justificada.

De forma a contornar este problema, uma das soluções alternativas e sugeridas pela Android é a utilização de um Serviço em primeiro plano com notificações ativas. Esta abordagem garante que a aplicação, durante a gravação de uma atividade, nunca seja colocada em segundo plano. Isto acontece em virtude da existência de uma notificação permanente na barra de notificações do dispositivo. Esta notificação obriga a aplicação a permanecer ativa e em primeiro plano. Por outro lado, esta notificação serve também como lembrete para o utilizador saber que a aplicação está em funcionamento e a recolher dados de localização.

Quando a gravação de uma atividade física é concluída, é necessário proceder ao cálculo e armazenamento de todas as estatísticas gerais na memória interna do dispositivo, para que, noutras partes da aplicação, estas possam ser utilizadas. A solução encontrada para este problema foi a criação de uma base de dados em MySQL. Para este contexto em específico, foi criada uma tabela onde são armazenadas, para cada atividade efetuada, os seguintes dados:

- Distância percorrida;
- Tempo da atividade;
- Timestamp;
- Ganho de altitude positiva;
- Rapidez média;

4.2 Definição de Objetivos

Outra funcionalidade que se pretende implementar na aplicação é a criação de um mecanismo onde sejam estabelecidos objetivos para a atividade do utilizador. Esta função serve como ferramenta de auxílio para a monitorização da atividade física, sendo que são estabelecidos objetivos (de distância ou de tempo) que podem ser posteriormente utilizados para analisar o desempenho da atividade realizada pelo utilizador.

O armazenamento desta informação é feito usando uma base de dados em MySQL. Cada entrada da tabela guarda informação sobre o Timestamp, o tipo de objectivo estabelecido, as intensidades mínima e máxima e os valores do objectivo para cada um dos dias da semana. A alteração de qualquer um destes parâmetros faz gerar uma nova entrada na tabela. O conjunto de todas entradas da tabela forma o histórico de alterações dos valores de objectivo.

4.3 Comunicação de Dados

Outro objectivo importante no desenvolvimento desta aplicação é a criação de um sistema partilha de dados de atividade entre utilizadores. Esta funcionalidade tem como intuito principal ser utilizada em contexto médico onde o paciente tem a possibilidade de partilhar e apresentar de forma voluntária o seu histórico de atividade física. Com esta informação, o dispositivo do médico tem a possibilidade de calcular estatísticas gerais e apresentar os dados de forma clara e descomplicada, permitindo assim uma análise mais rápida e eficaz da atividade física do seu paciente. Esta funcionalidade pode também ser aplicada no contexto do fitness, onde o personal trainer pode consultar e monitorizar os dados de desempenho do seu cliente.

Esta funcionalidade foi implementada com base na transmissão de informação sob a partilha de um QR Code. Uma das principais vantagens deste método de comunicação é que esta não depende de uma conexão à Internet para funcionar. Este fator revela-se fundamental principalmente em contexto médico, por vários motivos. Segundo um estudo da Marktest, 84,4% dos residentes em Portugal Continental com mais de 15 anos usava um Smartphone como telemóvel principal [28]. Além disso, a Anacom conclui que, em 2020, 78% em Portugal tinham acesso a Internet móvel [29]. Por outro lado, nem sempre é possível garantir existência de uma conexão Wi-Fi em todos os consultórios dos vários estabelecimentos de saúde. A conjunção destes factos permite concluir que nem sempre é possível garantir uma conexão via Internet para todos os utilizadores de smartphones, nomeadamente em ambiente médico.

Alternativamente ao QR Code, existem outros métodos que poderiam ser utilizados de forma a alcançar o objetivo pretendido. Entre eles incluem-se a tecnologia Bluetooth, Near Field Communication (NFC) ou a transmissão de dados por internet.

Em primeiro lugar, a tecnologia Bluetooth está presente em todos os smartphones atuais e apresenta-se como um método eficaz para a transmissão de pequenas quantidades de dados entre dispositivos separados por uma curta distância. Por outro lado, o NFC é uma tecnologia cada vez mais presente nos smartphones atuais e, à semelhança do Bluetooth, apresenta-se como uma forma competente para a transmissão de dados entre dispositivos a curtas distâncias. Por fim, a transmissão de dados por internet apresenta-se como o formato mais comum para a transmissão fiável de dados sem fios para longas distâncias. Além disso, todos estes métodos apresentam a vantagem conjunta de permitirem a transmissão de uma quantidade superior de dados em relação ao QR Code. Porém, todos estes requerem algum conhecimento tecnológico básico para a sua utilização efetiva, aspeto este que poderia representar uma barreira para utilizadores de smartphones menos experientes.

Tendo em consideração o objectivo de criar uma solução inclusiva e de fácil utilização, a tecnologia de partilha de dados por QR Code tornou-se uma solução alternativa viável para o contexto desta aplicação.

4.3.1 Partilha de Dados

Uma das fases do processo de comunicação de dados de atividade é a partilha de dados através da criação e codificação de um QR Code que é composto por vários passos. Para perceber cada um deles é necessário em primeiro lugar entender o tipo de informação necessária a ser transferida.

A solução mais simples seria codificar num QR Code com toda a informação armazenada na base de dados das atividades e na base de dados dos objetivos definidos. Porém, a capacidade de armazenamento finita de um QR Code obriga a que seja efetuado um pré-processamento destes dados evitando assim o envio de informação desnecessária ou redundante.

Uma forma de condensar estes dados começa pelo agrupamento de todas as estatísticas das atividades guardadas em estatísticas diárias. Pegando num exemplo em que num único dia é efetuado o registo de duas atividades diferentes: uma onde foram percorridos 2 quilómetros em 20 minutos e outra onde foram percorridos 3 quilómetros em 40 minutos. No processo de partilha de dados, em vez de enviar a informação de cada uma destas atividades, é feito o agrupamento das duas atividades numa estatística diária (neste caso, a informação a enviar deste dia é que foram percorridos 5 quilómetros em 60 minutos). Na figura 4.2 é possível observar o esquema do processo anteriormente explicado.

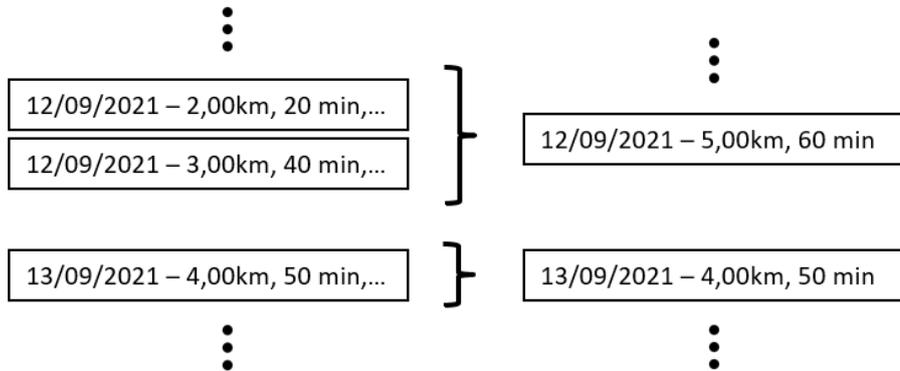


Figura 4.2: Processo de junção de atividades

Além disso, é necessário garantir o envio da informação de todos os dias relevantes. Por um lado, é necessário garantir que é enviada a informação dos dias em que foi efetuada uma atividade, incluindo dias em que não existia um objetivo diário definido. Por outro lado, é igualmente necessário enviar a informação dos dias em que foram estabelecidos objetivos diários, incluindo dias em que não foram efetuados registos de alguma atividade do utilizador. Em conjunto, estes dois grupos contemplam a informação suficiente para caracterizar todo o histórico da atividade do utilizador, tal como esquematizado no diagrama de Venn da figura 4.3.



Figura 4.3: Diagrama de Venn

Toda estes dados são agrupados num único conjunto de dados diário. A cada conjunto de dados diário dá-se o nome de pacote diário. Cada um destes pacotes terá assim os dados estritamente

necessários para caracterizar cada dia de atividade. Cada pacote é composto pelos seguintes dados:

- Distância (*short*) - 2 bytes;
- Tempo (*int*) - 4 bytes;
- Timestamp - 4 bytes;
- Intensidade mínima (*short*) - 2 bytes;
- Tipo/valor do objectivo (*short*) - 2 bytes;

Deste conjunto de dados, importa referir a lógica de implementação por detrás do tipo/valor do objectivo. Caso o objectivo seja do tipo “distância”, o valor inserido nesse parâmetro é o valor do objectivo, mais 1. Caso o objectivo seja do tipo “tempo”, o valor inserido nesse parâmetro é igual ao simétrico do valor do objectivo, menos 1. Desta forma é possível utilizar todo o alcance do tipo de dado *short* (-2^{15} a $2^{15} - 1$) e evitar o uso de um tipo de dado extra (*boolean* neste caso) que outrora ocuparia 1 byte extra. Cada pacote diário perfaz assim um tamanho total de 14 bytes.

É a partir deste pacote diário que é projetada a mensagem a ser codificada no QR Code. Na figura 4.4 é possível visualizar esquematicamente todo o processo até à criação de QR Code.

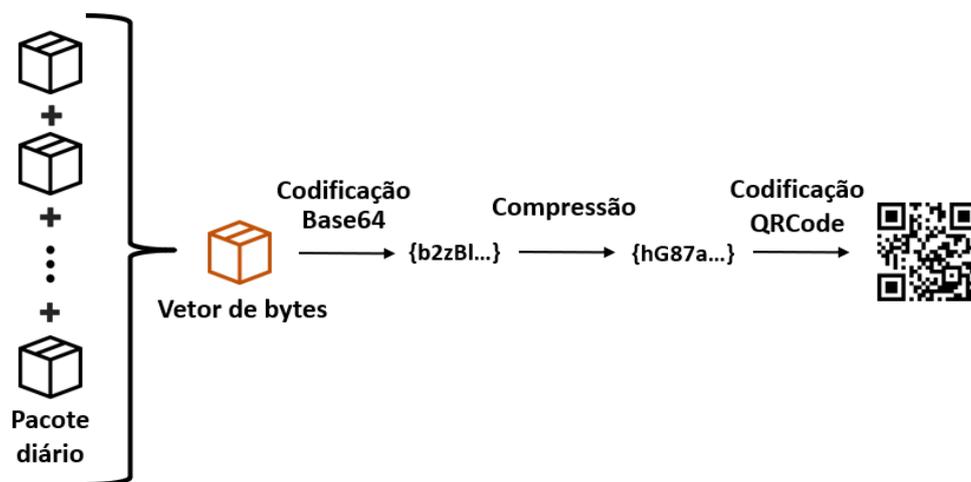


Figura 4.4: Processo de codificação do QR Code

Numa primeira fase, cada um dos pacotes diários é unido de forma a originar um vetor de bytes. Após este passo, é utilizado o método de codificação Base64 que converte a informação contida no vetor de bytes numa *string*. Seguidamente é utilizada uma função de compressão de dados. A utilização deste procedimento é adequada neste contexto por dois motivos: em primeiro lugar reduz-se significativamente o espaço ocupado pela mensagem; por outro lado, apesar de existir um processamento extra na compressão e a respetiva descompressão destes dados, a ordem de grandeza dos mesmos é pequena o suficiente para que esse tempo extra seja impercetível. Por fim, é feita a codificação

do QR Code, que seguidamente é exibido no dispositivo do utilizador. Na figura 4.5 é apresentado o código utilizado para a realização do processo atrás referido, em que a variável “baos” representa a *ByteArrayOutputStream* com a informação que se pretende que seja transmitida. Cada uma das linhas de código efetua a codificação em Base64, a compressão dos dados e a codificação do QR Code em formato *Bitmap*, respetivamente.

```
String encodedData = Base64.encodeToString(baos.toByteArray(), Base64.DEFAULT);
String encodedDataCompressed = CompressionUtil.compressAndReturnB64(encodedData);
Bitmap bitmapData = QRCode.from(encodedCompressed).withSize(2000,2000).bitmap();
```

Figura 4.5: Código utilizado para efetuar o processamento e codificação do QR Code

4.3.2 Receção de Dados

O processo de descodificação do QR Code acontece de forma semelhante à codificação do código, embora em ordem inversa. Na figura 4.6 está representado um esquema com todo o processo de leitura e interpretação do QR Code.

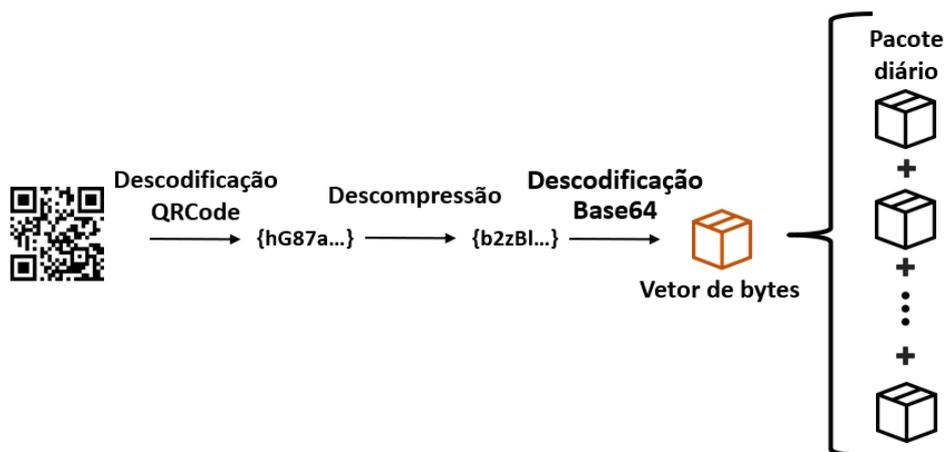


Figura 4.6: Processo de descodificação do QR Code

Em primeiro lugar, o QR Code é descodificado de forma a obter a *string* correspondente. Numa segunda fase, é efetuada a descompressão dos dados. Seguidamente é utilizada uma função de descodificação Base64 que descodifica a *string* num vetor de bytes. A descodificação termina com a leitura iterativa dos dados deste vetor. Após a descodificação da informação, esta é possível ser processada e apresentada de forma personalizada no dispositivo do recetor. Nas figuras 4.7 e 4.8 são apresentados excertos do código utilizado no processo anteriormente referenciado. O código da primeira figura efetua a leitura do resultado da leitura do QR, ao passo que a segunda figura mostra o

código utilizado para a extração, descompressão e descodificação de Base64 para um vetor de bytes.

```
mCodeScanner = new CodeScanner(requireActivity(), binding.scannerView);  
mCodeScanner.setDecodeCallback(result -> requireActivity().runOnUiThread(() ->  
| presenter.processQrCode(result)));
```

Figura 4.7: Código utilizado para efetuar a leitura do QR Code

```
String encodedDataCompressed = result.getText();  
String encodedData = CompressionUtil.decompressB64(encoded);  
byte[] decoded = Base64.decode(encodedDecompressed, Base64.DEFAULT);
```

Figura 4.8: Código utilizado para efetuar o processo de descodificação do QR Code

5

Aplicação

Conteúdo

5.1	Arquitetura Geral	39
5.2	Interface	41

Este capítulo tem como objetivo descrever a arquitetura geral da aplicação, assim como detalhar os vários componentes de interface e a respetiva relação com a componente lógica da aplicação.

A aplicação móvel “TocáMixer” foi desenvolvida em Android Studio com recurso a Java e tem como principal funcionalidade efetuar o registo e monitorização de atividade física com recurso a GNSS. Cada atividade gravada tem informação sobre a distância percorrida, o tempo, a rapidez média e elevação positiva. Além disso é possível estabelecer objetivos diários de forma a comparar o progresso do utilizador com o progresso esperado. Toda essa informação é utilizada para gerar relatórios informativos sobre o desempenho do utilizador que podem facultativamente ser partilhados com outros dispositivos por QR Code.

De forma a tornar a linguagem mais familiar e de fácil entendimento para o público em geral, na interface da aplicação utilizei o termo “velocidade média” em vez de “rapidez média”. Desta forma, em todo o texto deste capítulo fez mais sentido usar o termo “velocidade média”.

5.1 Arquitetura Geral

Esta aplicação está organizada em duas Atividades distintas: a Splash Activity e a Main Activity.

A Splash Activity é uma Atividade onde é apresentado o título da aplicação durante um curto período de tempo, tal como é possível visualizar na figura 5.1. Além de apresentar a aplicação ao utilizador, a utilização desta Atividade tem como intuito principal efetuar o carregamento dos componentes iniciais da aplicação. Entre estes incluem-se o carregamento de dados armazenados em base de dados ou o carregamento de componentes de interface da aplicação.

A Main Activity é a Atividade mais complexa da aplicação onde todos os Fragmentos da aplicação estão sediados. Na figura 5.2 é possível visualizar os vários componentes da interface geral da Main Activity.

O componente de layout FrameLayout é responsável por gerir alternadamente os ciclos de vida de cada uma destes Fragmentos consoante o comportamento do utilizador durante a utilização da aplicação. Entre estes incluem-se a Página Inicial, a Atividade, o Histórico, o Perfil, a Partilha de Dados, o Scan de Dados e as Definições. Todos estes Fragmentos mencionados (com exceção das Definições) estão estruturados sob o padrão de arquitetura MVP.

Por outro lado, a Main Activity contém ainda um menu de navegação inferior visível na figura 5.2 e um menu de navegação lateral visível na figura 5.3. A principal função destes elementos é servir como meio de navegação entre as várias funcionalidades da aplicação.



Figura 5.1: Interface da Splash Activity

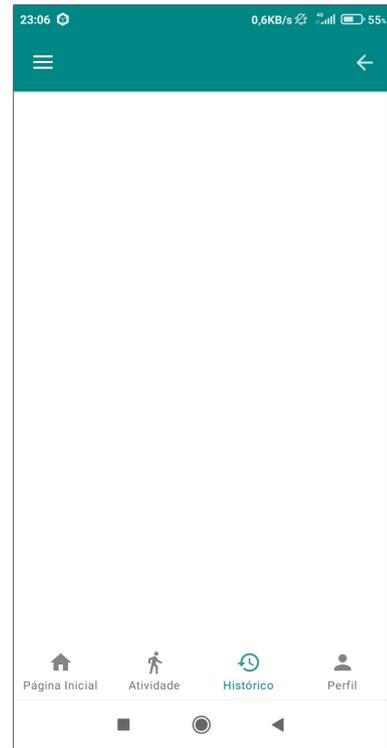


Figura 5.2: Interface genérica da Main Activity

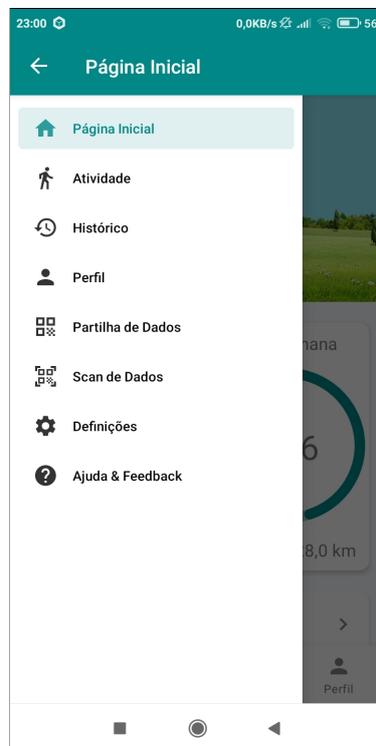


Figura 5.3: Menu de navegação lateral de uma interface genérica da Main Activity

5.2 Interface

5.2.1 Página Inicial

Após a exibição da Splash Activity, a Main Activity é iniciada com o carregamento do Fragmento da Página Inicial, tal como apresentado nas figuras 5.4 e 5.5. Nesta parte da aplicação são mostradas várias estatísticas gerais sobre a histórico de atividade do utilizador assim como o acesso a algumas funcionalidades gerais da aplicação.



Figura 5.4: Interface superior da Página Inicial

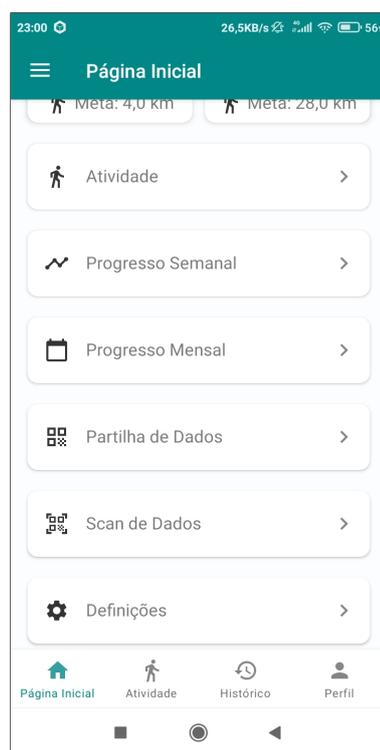


Figura 5.5: Interface inferior da Página Inicial

Na parte superior do ecrã é apresentada uma imagem alusiva ao tema da aplicação. Em baixo deste componente, são apresentados dois Cardviews com informações gerais acerca do progresso da atividade do utilizador. Consoante a escolha do tipo de objetivo é apresentado um gráfico circular referente ao progresso atual em comparação com o objetivo pré-determinado. O Cardview do lado esquerdo apresenta a informação sobre o próximo objetivo a ser realizado e o respetivo progresso, ao passo que o Cardview do lado direito apresenta a informação do progresso semanal em comparação com o respetivo objetivo esperado.

Abaixo deste elemento é possível encontrar um botão de atalho de acesso para o Fragmento de Atividade. Mais abaixo encontram-se dois botões retangulares que permitem o acesso a informação mais detalhada sobre com o progresso da atividade física do utilizador. Ao pressionar o botão “Progresso

Semanal” é apresentado um gráfico misto de pontos e barras (figura 5.6) onde é possível comparar o progresso semanal com a meta diária previamente estabelecida.

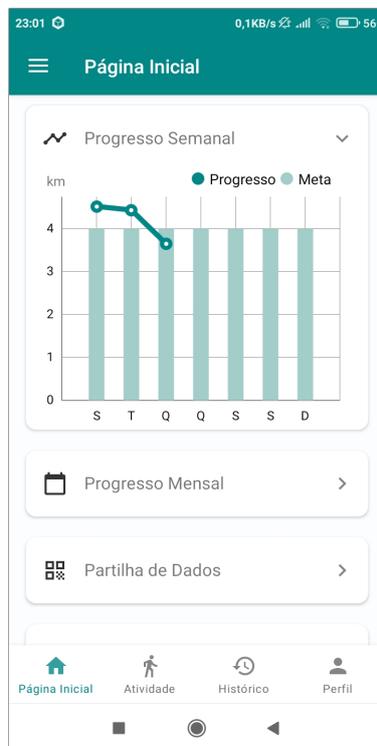


Figura 5.6: Interface do Progresso Semanal

Ao pressionar o botão “Progresso Mensal” é apresentado um calendário mensal (figura 5.7) com um código de cores onde é possível ter uma perspectiva mais geral sobre o progresso comparado com os objetivos traçados pelo utilizador ao longo de um mês.

Nesta vista de calendário é apresentado um código de cores que avalia o desempenho de cada dia comparado com os objetivos estabelecidos. Neste caso, cada ponto verde identifica um dia em que a meta diária e a meta de intensidade mínima foram ambas alcançadas. Por outro lado, a laranja são identificados os dias em que a meta diária foi alcançada embora sem cumprir com a intensidade mínima estabelecida. A vermelho são identificados os dias em que a meta diária não foi alcançada. Por fim, a roxo são identificados os dias em que uma atividade extra foi efetuada num dia sem meta diária estabelecida. Ao clicar no ícone de informações ao lado direito do texto “Progresso Semanal” é aberto uma janela com a legenda do código de cores (figura 5.8)).

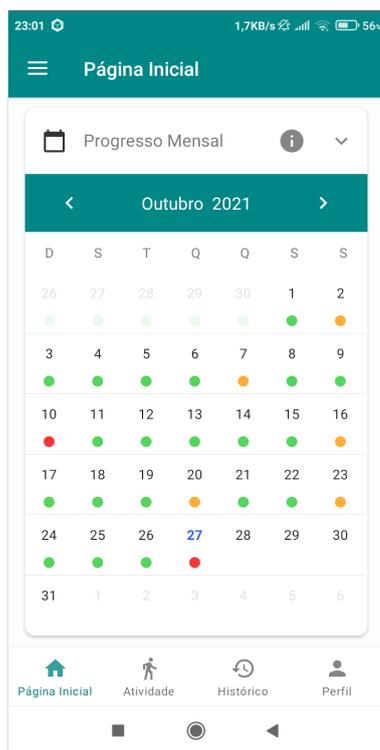


Figura 5.7: Interface do Progresso Mensal

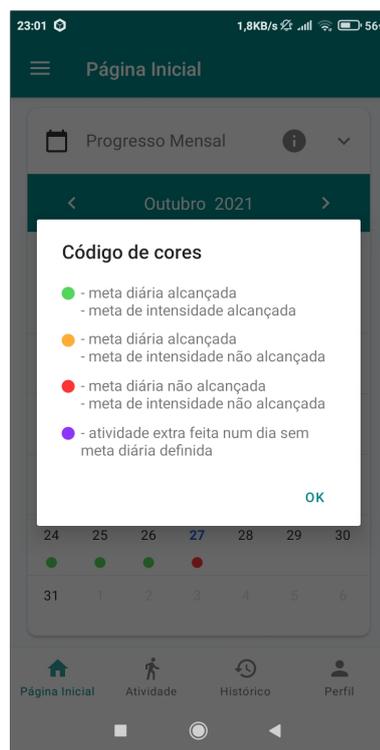


Figura 5.8: Janela da legenda do código de cores

Por fim, os botões “Partilha de Dados”, “Scan de Dados” e “Definições” visíveis na figura 5.5 são botões de atalho para os respetivos Fragmentos presentes na Main Activity.

5.2.2 Atividade

A principal funcionalidade deste Fragmento é a gravação de uma atividade através da recolha e interpretação periódica de dados de localização.

A interface inicial desta parte da aplicação está representada na figura 5.9. Pressionando o botão verde de “Iniciar Atividade”, a interface atualiza de modo a mostrar os botões de “Pausa” e de “Concluir”, tal como está representado na figura 5.10. Além disso, esta ação desencadeia o início da aquisição de dados de localização obtidos por GNSS. É com essa informação que são sucessivamente calculados e atualizados parâmetros como a distância percorrida ao longo do tempo, velocidade e a altura em relação ao nível médio do mar. Em conjunto com o tempo decorrido desde o início da atividade, todos estes parâmetros são continuamente atualizados na parte superior do ecrã deste Fragmento.

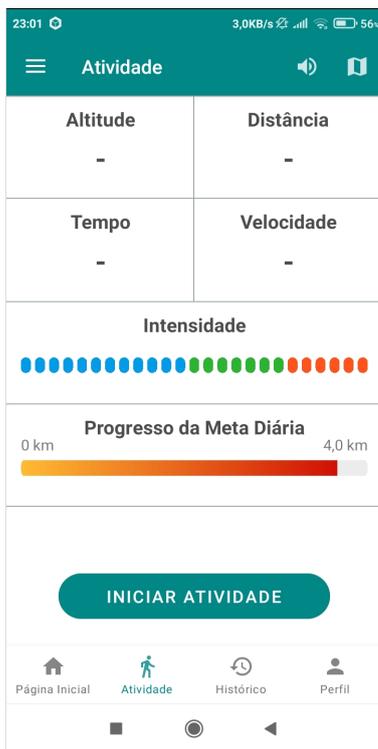


Figura 5.9: Interface inicial da Atividade



Figura 5.10: Interface com uma atividade a decorrer

Abaixo destes valores podem encontrar-se ainda duas animações visuais com informações sobre o estado atual da atividade.

A primeira animação com o nome de “Intensidade” dá a conhecer ao utilizador se está a executar a sua atividade física dentro da intensidade pré-determinada nas definições do Fragmento do Perfil. Este gráfico apresenta três zonas coloridas: a azul representa o conjunto de velocidades abaixo do ritmo mínimo definido; a vermelho reúne o conjunto de velocidades acima do ritmo máximo estabelecido; a verde compreende o conjunto de velocidades acima do ritmo mínimo e abaixo do ritmo máximo pré-determinado. A velocidade atual é representada pela maior barra vertical do gráfico.

A segunda e última representação gráfica com o nome de “Progresso da meta diária” dá informação sobre o progresso do utilizador em função da meta estabelecida à priori para o dia da atividade. Esta informação vai atualizando ao longo da atividade e pode representar um de dois tipos de meta diária: distância em quilómetros (km) ou tempo em minutos (min). Caso o utilizador não estabeleça nenhum objetivo, este gráfico não é visível.

Por fim, é possível aceder à representação espacial do trajeto efectuado ao clicar no botão com o ícone de um mapa no canto superior direito da aplicação. Com esta ação a aplicação muda de interface e passa a mostrar em tempo real o trajeto efetuado ao longo do percurso utilizando um mapa do Google Maps (figura 5.11). A interface anterior da figura 5.10 pode voltar a ser mostrada ao clicar no ícone de

retroceder no canto superior direito da aplicação.



Figura 5.11: Interface do mapa da atividade

A atividade é encerrada pressionando o botão vermelho “Concluir”. Com esta ação, todos os componentes de interface anteriormente mencionados (à exceção do gráfico do progresso da meta) voltam ao estado inicial da figura 5.9. Alternativamente, e se ativadas essas funcionalidades nas Definições da aplicação, a atividade pode ser terminada automaticamente em duas circunstâncias diferentes:

- A primeira circunstância possível é na situação em que o utilizador permaneça parado durante um período significativo de tempo (1 minuto). Esta situação pode ocorrer quando o utilizador, após terminar a sua atividade física, se esquece de terminar manualmente a atividade no botão de “Concluir”.
- A segunda e última situação é quando a aplicação regista uma velocidade superior a uma velocidade pré-estabelecida. Esta funcionalidade tem como intuito evitar que, inusitadamente, os dados de uma “não atividade” continuem a ser adquiridos. Este tipo de situações pode ocorrer, por exemplo, quando o utilizador se esquece de encerrar a contagem de uma atividade e de seguida utiliza um veículo de transporte. A velocidade para a qual este sistema acionado pode ser personalizado de acordo com as preferências do utilizador.

5.2.3 Histórico

À semelhança da Página Principal, neste Fragmento são apresentados dois conteúdos gráficos: um gráfico onde é possível comparar o progresso semanal com a meta diária previamente estabelecida (figura 5.12); e um calendário com uma vista mensal com informação sobre o desempenho do utilizador (figura 5.13). Estes elementos comportam-se de forma idêntica aos gráficos descritos na secção da Página Principal.



Figura 5.12: Interface superior do Histórico

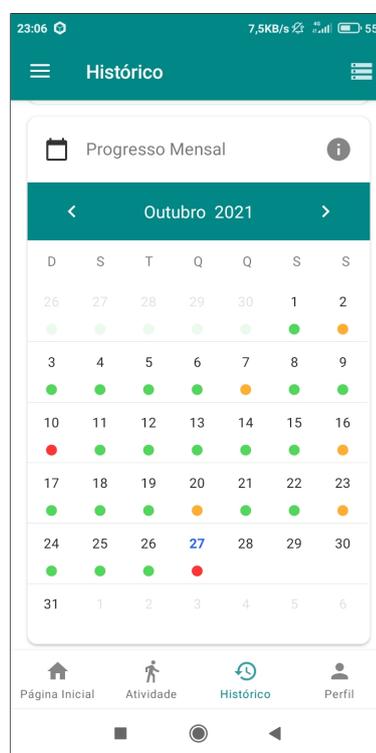


Figura 5.13: Interface inferior do Histórico

Ao pressionar no ícone do canto superior direito é possível aceder a uma nova interface de Armazenamento onde é apresentado o histórico de atividades efetuadas por ordem decrescente cronologicamente, tal como é possível visualizar na figura 5.14. Para cada atividade é apresentada informação sobre o Timestamp de início de atividade, o tempo, a distância, a variação de altitude positiva e a velocidade média. Cada uma destas atividades pode ser eliminada arrastando o Cardview horizontalmente (ou para a direita ou para a esquerda).



Figura 5.14: Interface de Armazenamento

5.2.4 Perfil

Neste Fragmento é possível personalizar vários aspetos sobre a monitorização de objetivos do utilizador. Na figura 5.15 é possível visualizar a interface deste Fragmento.

Na parte superior do ecrã encontra-se o elemento “Tipo de Meta” onde é possível selecionar ou o tempo ou a distância como referência para a monitorização da atividade do utilizador. Abaixo deste elemento encontra-se a “Intensidade”, onde é possível personalizar as velocidades mínima e máxima espectáveis para o exercício do utilizador. Por fim, encontra-se o elemento de “Distribuição Semanal”, onde é possível escolher o valor do objetivo para cada um dos dias da semana. Note-se que o tipo de objetivo está intimamente relacionado com o tipo de meta escolhido na primeira opção do ecrã. Todos estes dados são armazenados numa tabela de uma base de dados em MySQL.



Figura 5.15: Interface do Perfil

5.2.5 Partilha de Dados

Este Fragmento tem como objetivo efetuar a partilha dos dados do utilizador com outros dispositivos que tenham também a mesma aplicação. Os dados partilhados dizem respeito ao histórico mais recente de atividade física do utilizador. Para partilhar estes dados, é gerado e apresentado um QR Code que codifica toda a informação que é necessária transmitir para o dispositivo destino. Na figura 5.16 é possível visualizar a interface do Fragmento.



Figura 5.16: Interface da Partilha de dados

5.2.6 Scan de Dados

Este Fragmento serve como complemento do Fragmento anterior de partilha de dados. Esta funcionalidade é utilizada no dispositivo em que o utilizador pretenda obter os dados de outro utilizador. Para ler os dados codificados em QR Code, este Fragmento utiliza a câmara fotográfica do dispositivo (figura 5.17). A leitura é efetuada apontando a câmara fotográfica a outro dispositivo que esteja com a funcionalidade de Partilha de Dados aberta.

Após a descodificação da informação, o ecrã muda para uma interface de apresentação de dados estatísticos semelhante ao que é encontrada no Fragmento de Página Inicial, tal como é possível verificar na figura 5.18.



Figura 5.17: Interface do Scan de Dados - scan do QR Code

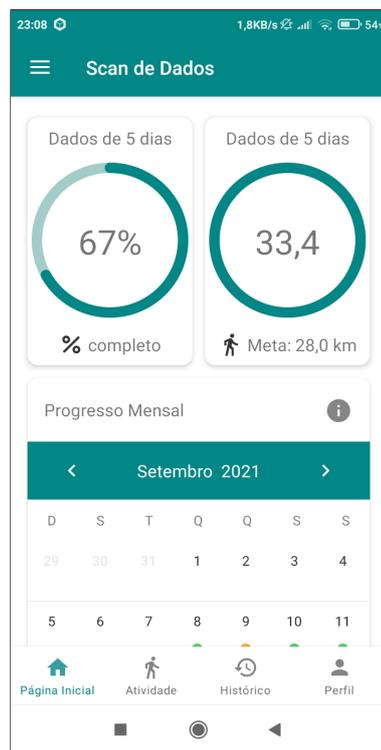


Figura 5.18: Interface do Scan de Dados - relatório de dados

Na figura 5.18, na parte superior do ecrã é possível visualizar dois Cardviews com as informações gerais sobre o progresso geral do utilizador à qual foi lido o QR Code. Abaixo destes elementos encontra-se um calendário mensal onde é possível obter uma perspetiva geral sobre o progresso diário do utilizador a analisar. O código de cores utilizado é em todo semelhante ao referenciado no elemento “Progresso Mensal” na secção 5.2.1 do Fragmento Página Inicial.

5.2.7 Definições

Neste Fragmento é possível personalizar várias definições sobre o funcionamento geral da aplicação. Nas figuras 5.19 e 5.20 é possível visualizar a interface deste Fragmento.

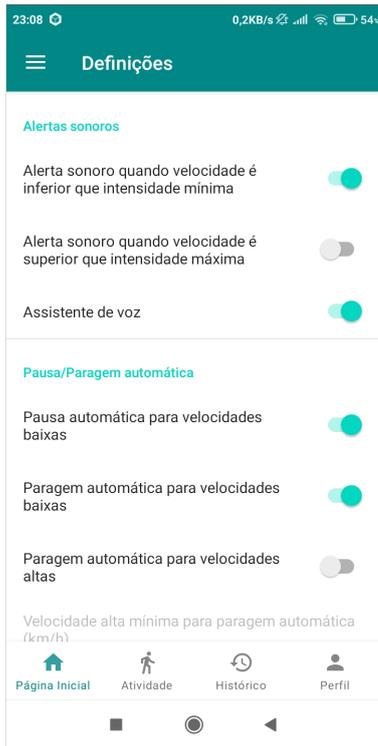


Figura 5.19: Interface superior das Definições



Figura 5.20: Interface inferior das Definições

Na parte superior do ecrã é possível encontrar a secção de “Alertas sonoros” onde é possível gerir os vários tipos de alertas sonoros disponíveis na aplicação. Os dois primeiros funcionam com base na comparação da velocidade do utilizador com as velocidades mínima e máxima estabelecidas do Fragmento do Perfil. Caso, por exemplo, o primeiro alerta esteja ativo e o utilizador percorra alguma parte da sua atividade abaixo da intensidade mínima definida, um sinal sonoro será reproduzido a informar sobre o ocorrido. O segundo alerta de intensidade máxima funciona de maneira análoga quando a velocidade do utilizador ultrapassa a intensidade máxima estabelecida. Por fim, é possível encontrar o “Assistente de Voz”. Esta funcionalidade utiliza um mecanismo de “Text-to-Speech” que vai interagindo com o utilizador em vários momentos sua da atividade, como indicar quando o objetivo diário é cumprido, por exemplo.

Mais abaixo no ecrã é possível encontrar a secção da “Pausa automática”. Nesta parte do ecrã é possível ativar/desativar funcionalidades como a pausa automática para velocidades baixas, a paragem automática para velocidades baixas e a paragem automática para velocidades altas.

Como o próprio nome indica, a ativação da primeira permite à aplicação detetar quando o utilizador para de forma temporária em alguma parte da sua atividade e pausar o registo da mesma. A atividade volta a ser contabilizada assim que o utilizador retome o seu movimento. Esta funcionalidade pode ser útil caso se pretenda apenas registar a atividade física útil, evitando o registo de momentos em que o utilizador está momentaneamente parado.

A paragem automática funciona de maneira semelhante à pausa automática, com a particularidade de encerrar de forma definitiva a atividade quando é detetado que o utilizador está um longo período de tempo sem realizar qualquer movimento.

Ademais, a ativação da paragem automática para velocidades altas permite ao dispositivo encerrar de forma definida uma atividade quando é registado uma velocidade superior à esperada.

Por fim, está disponível uma opção para repor os dados da aplicação, retornando assim a aplicação ao seu estado inicial.

6

Testes ao Sistema

Conteúdo

6.1	Teste de Intervalo de Aquisição de Dados	55
6.2	Teste de Precisão	61

Este capítulo tem como objetivo principal descrever e explicar dois testes realizados no desenvolvimento da aplicação. Em primeiro lugar, foi efetuado um teste com variados intervalos de atualização da aquisição de dados de localização. Além disso, foi realizado um teste à precisão da aquisição de dados de localização em ambientes urbano e rural.

6.1 Teste de Intervalo de Aquisição de Dados

Este primeiro teste tem como finalidade principal validar a qualidade de representação de um dado percurso. Para isso, percorreu-se várias vezes o trajeto recolhendo dados de localização para diferentes intervalos de tempo de aquisição. Os ensaios foram efetuados num circuito com aproximadamente 511 metros na zona da Batalha (distância medida no Google Earth). Este local situa-se num ambiente rural perto de habitações de baixa altura, tal como é possível observar na figura 6.1. Além disso, tal como é possível verificar na figura 6.2, o circuito escolhido caracteriza-se por conter curvas e retas, ambiente este que se aproxima da maioria dos contextos em que esta aplicação será utilizada.



Figura 6.1: Fotografia do ambiente de teste na zona da Batalha

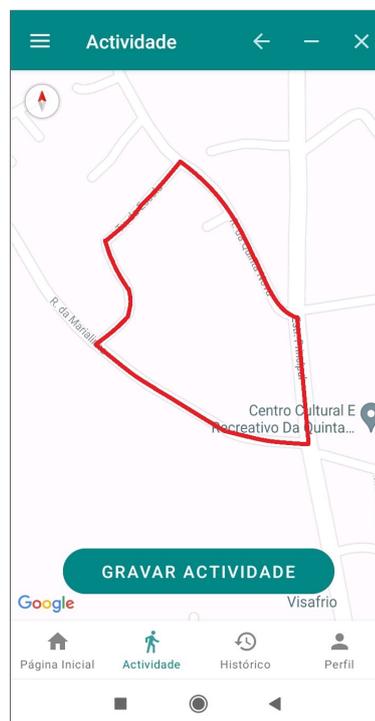


Figura 6.2: Percurso teórico percorrido

Para este percurso foram efetuados ensaios com características diferentes: um em ritmo de caminhada e outro com ritmo de corrida. A realização destes dois ensaios ajudará a analisar vários intervalos de aquisição de dados em duas das circunstâncias mais comuns do uso desta aplicação.

6.1.1 Teste em Ritmo de Caminhada

Nas figuras 6.3, 6.4, 6.5 e 6.6 encontram-se representados as trajetórias obtidas em ritmo de caminhada em quatro testes com intervalos de aquisição de dados diferentes: 1, 2, 4 e 8 segundos, respetivamente. Além disso, na tabela 6.1 é possível verificar alguns resultados associados aos quatros testes efetuados neste ambiente.

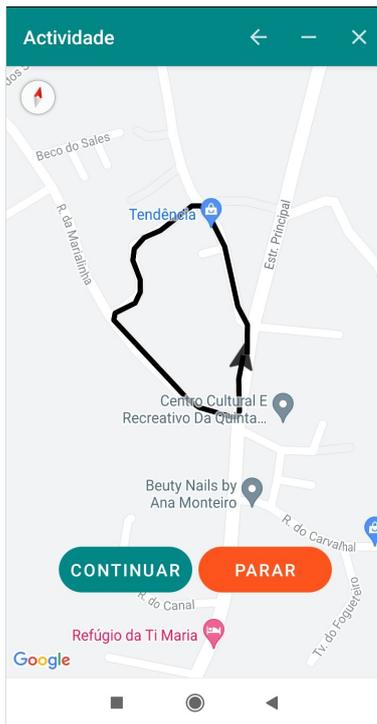


Figura 6.3: Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 1 segundo

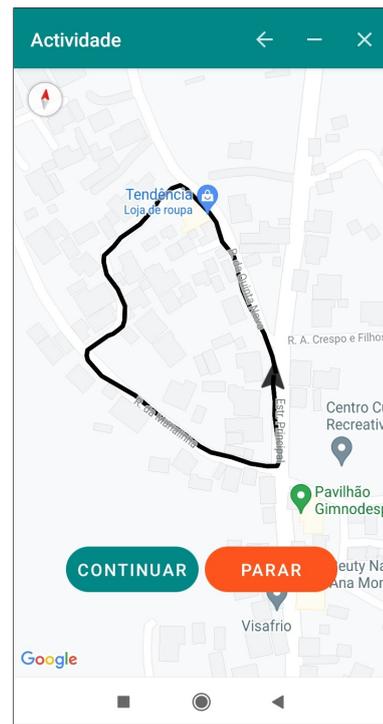


Figura 6.4: Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 2 segundos

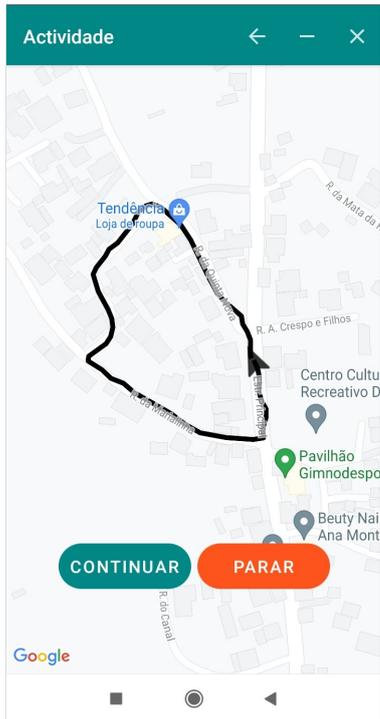


Figura 6.5: Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 4 segundos

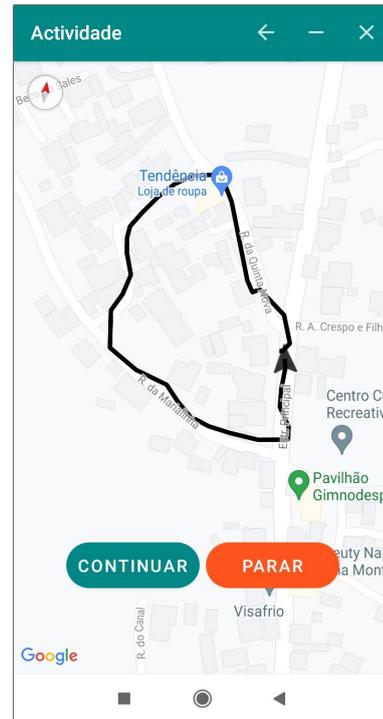


Figura 6.6: Percurso gravado em ritmo de caminhada com um intervalo de aquisição de dados de 8 segundos

Tabela 6.1: Informação adicional sobre os testes efetuados em ritmo de caminhada

Intervalo de tempo (s)	1	2	4	8
Rapidez média (km/h)	6,0	6,0	5,6	5,7
Distância medida (m)	512,1	507,4	506,8	481,0
Erro relativo da distância medida (%)	1,9	2,8	2,9	7,9

Em primeiro lugar, é possível verificar que as trajetórias obtidas com $\Delta t = 1$ s e $\Delta t = 2$ s representam de forma muito exata a trajetória esperada para este percurso. Em relação a $\Delta t = 4$ s, apesar de existirem alguns pontos do percurso desajustados em relação ao percurso real, este teste acabou igualmente por retratar com bastante exatidão o percurso esperado. O teste com o intervalo de recolha de dados de 8 s acabou por descrever de forma pouco rigorosa o percurso efetuado.

Em relação à distância calculada, o percurso gravado com 1, 2 e 4 segundos destacam-se positivamente com erros entre 1,9% e 2,9%. Em contrapartida, para $\Delta t = 8$ s obteve-se um erro de 7,9% em relação à distância esperada.

6.1.2 Teste em Ambiente de Corrida

Nas figuras 6.7, 6.8, 6.9 e 6.10 encontram-se representadas as trajetórias obtidas em ritmo de corrida, em quatro testes com intervalos de aquisição de dados diferentes: 1, 2, 4 e 8 segundos, respetivamente. Além disso, na tabela 6.2 é possível verificar alguns resultados associados aos quatros testes efetuados neste ambiente.

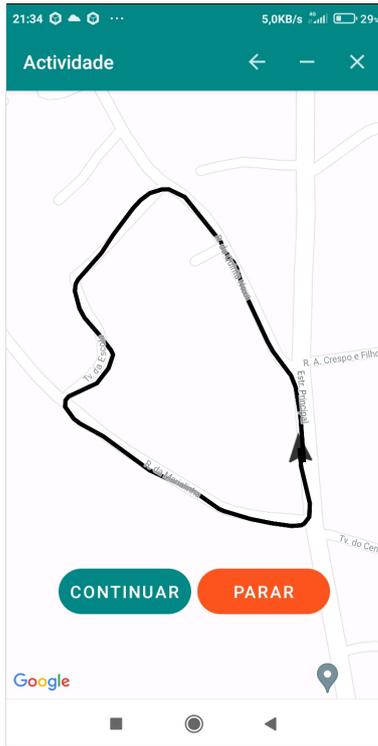


Figura 6.7: Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 1 segundo

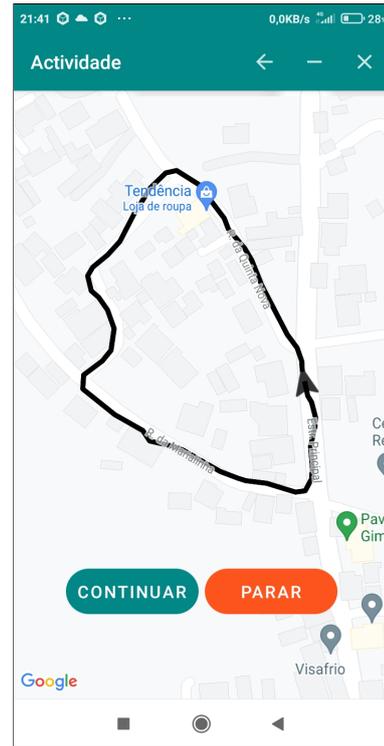


Figura 6.8: Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 2 segundos

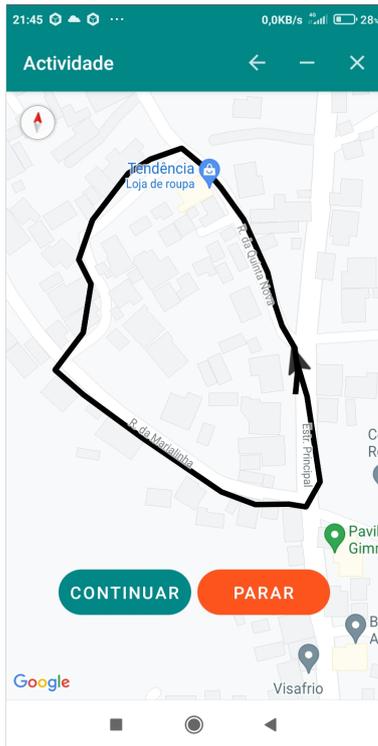


Figura 6.9: Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados 4 segundos

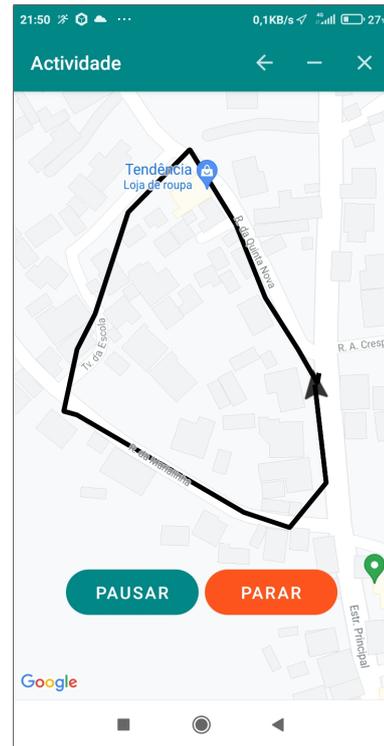


Figura 6.10: Percurso gravado em ritmo de corrida com um intervalo de aquisição de dados de 8 segundos

Tabela 6.2: Informação adicional sobre os testes efetuados em ritmo de corrida

Intervalo de tempo (s)	1	2	4	8
Rapidez média (km/h)	10,5	10,0	11,2	10,9
Distância medida (m)	524,6	507,6	498,3	463,7
Erro relativo da distância medida (%)	0,5	2,8	4,6	11,2

Em primeiro lugar, é possível constatar que, apesar de algumas imprecisões, os trajetos obtidos com $\Delta t = 1$ s e $\Delta t = 2$ s representam adequadamente a trajetória prevista para este circuito. Em relação ao teste com o intervalo de recolha de dados de 4 segundos, o percurso obtido contém um maior número de imprecisões em relação aos anteriores, especialmente em curvas acentuadas. Por fim, o teste obtido com $\Delta t = 8$ s apresentou um elevado número de erros em comparação com o trajeto efetuado, em particular em curvas ligeiras e acentuadas.

Relativamente às distâncias obtidas, o percurso registado com $\Delta t = 8$ s salientou-se novamente por se distanciar significativamente do valor real com um erro relativo em relação à distância medida de 11,2%. Contudo, as distâncias obtidas com intervalos de 1, 2 e 4 segundos aproximaram-se razoavelmente do valor esperado, com erros de 0,5%, 2,8% e 4,6%.

6.1.3 Conclusão

A realização deste teste permite concluir que, tendencialmente, quanto maior o intervalo de aquisição de dados de localização, pior é a qualidade de representação de um dado trajeto no mapa. Em comparação com os outros intervalos testados, para $\Delta t = 8$ s obteve-se um elevado número de pontos desencontrados com a trajetória esperada, em particular em curvas.

É também possível concluir que, tendencialmente, a escolha do intervalo de aquisição afeta a exatidão do cálculo da distância percorrida. A aquisição efetuada com um intervalo de 1 segundo foi o ensaio onde se obteve menor erro no cálculo das distância (entre 0,5% e 1,9 %), ao passo que para $\Delta t = 8$ s foi obtido um erro maior (entre 7,9% e 11,2 %).

Além disso, é possível concluir que o ritmo da atividade interfere diretamente com a qualidade da captura do percurso efetuado. Por um lado, os percursos efetuados em ritmo de caminhada e com um intervalo baixo de aquisição de dados apresentaram tendencialmente os melhores resultados. Por outro lado, os percursos realizados em ritmo mais intenso e com um intervalo de tempo maior entre aquisição de dados verificaram-se tendencialmente mais afastados do resultado esperado (511 metros).

No contexto prático desta aplicação, estes resultados sugerem que o intervalo mais adequado para a aquisição de dados é de 1 segundo. Porém, a escolha do intervalo de aquisição poderia suscitar alguma reflexão caso a implementação da aplicação tivesse como base o envio em tempo real destes dados por internet. O consumo de energia numa aplicação com estas características iria aumentar com o aumento da frequência de envio de dados. Neste caso, o intervalo de tempo mais adequado para minimizar o consumo de energia sem por em causa a qualidade necessária na obtenção de dados poderá rondar entre os 2 e os 4 segundos.

6.2 Teste de Precisão

Este teste tem como objectivo averiguar e comparar o erro na obtenção de dados de localização em ambiente rural e urbano. Para o ambiente rural foi escolhido o mesmo local do teste anterior. Por outro lado, o ambiente urbano seleccionado está localizado no centro cidade de Leiria (figura 6.11). O caminho percorrido é caracterizado por conter edifícios de média altura e ruas mais estreitas, comparado com o percurso em ambiente rural.



Figura 6.11: Fotografia do ambiente de teste no centro da cidade de Leiria

Para cada um dos locais foram efetuados ensaios de forma a obter 500 aquisições de dados de localização. A realização destes dois ensaios tem como objectivo comparar a qualidade da recepção de dados utilizando o parâmetro de erro calculado em cada um dos pacotes de dados de localização. Este parâmetro é obtido na API de localização da Google através de uma estimativa do erro radial da localização em metros com um intervalo de confiança de 68% [30].

6.2.1 Testes em Ambiente Rural e Urbano

Nas figuras 6.12 e 6.13 estão representados dois histogramas de distribuição dos valores de erro em metros obtidos durante o teste em ambiente rural e urbano, respetivamente. Cada gráfico é composto por 500 aquisições.

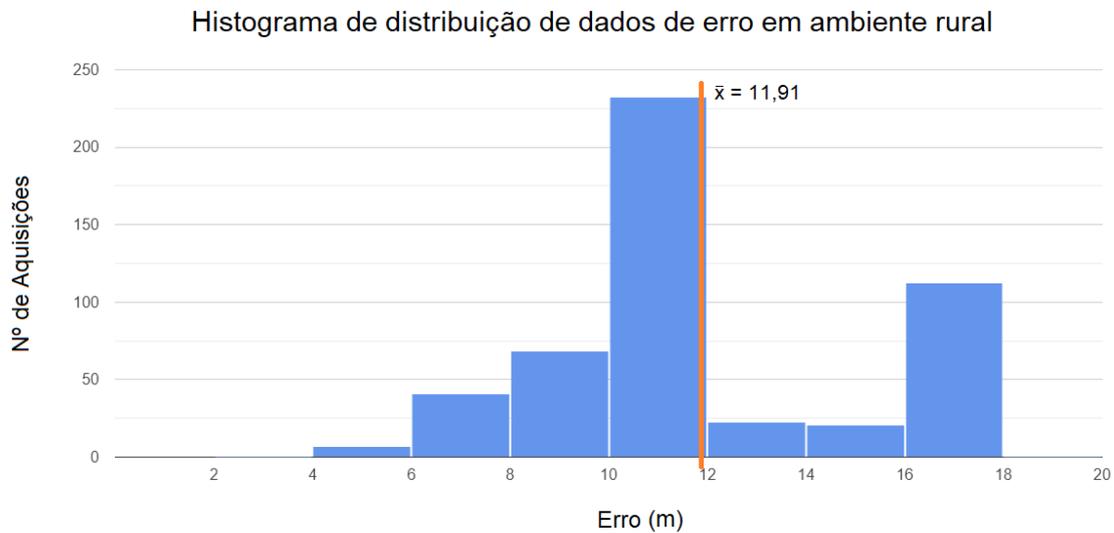


Figura 6.12: Histograma de distribuição de dados de erro no percurso em ambiente rural

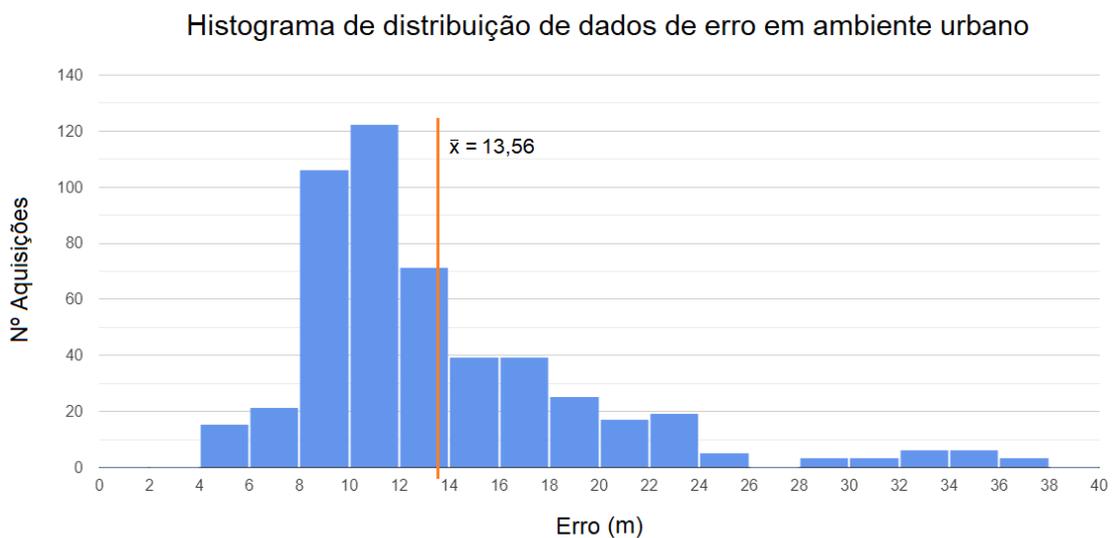


Figura 6.13: Histograma de distribuição de dados de erro no percurso em ambiente urbano

Em primeiro lugar, é possível verificar que a média obtida em ambiente rural (11,91m) é menor em relação à obtida em ambiente urbano (13,56m). Apesar desta ligeira diferença, é possível reparar que em ambiente urbano a distribuição dos valores é significativamente mais ampla (entre 4 e 38 metros), em comparação com a obtida em ambiente rural (entre 4 e 18 metros).

6.2.2 Conclusão

Os resultados obtidos permitem concluir que existe efetivamente uma diferença no erro dos dados obtidas em ambiente rural em comparação com um ambiente urbano.

Em média, o erro calculado em ambiente rural teve um valor de 11,91 metros, ao passo que para o ambiente urbano obteve-se uma média de 13,56 metros. Apesar da diferença entre as médias ser reduzida, em alguns dos pontos obtidos em ambiente urbano obteve-se um erro mais elevado (até 38 metros), situação esta que não ocorreu em ambiente rural. Esta diferença poder-se-á explicar pela morfologia estreita das ruas e a presença de edifícios altos.

7

Publicação da Aplicação

Uma das formas mais habituais de disponibilizar publicamente uma aplicação móvel passa pela publicação da mesma numa loja online de aplicações. Uma das lojas mais populares para a publicação de aplicações em Android é a Google Play Store. Esta loja online possui mais de 2 milhões e 700 mil aplicações disponíveis para download para qualquer utilizador que disponha de um smartphone com Android. Apesar deste número elevado, todas as aplicações desta loja passam por um processo de escrutínio e de controlo de modo a organizar e filtrar possíveis aplicações que não cumpram diretrizes exigíveis para um produto deste tipo.

O processo para a publicação desta aplicação na Google Play Store requiere a criação de uma conta de desenvolvedor Google. A taxa de pagamento é única (25 dólares), o que permite ao desenvolvedor publicar mais do que uma aplicação por conta.

O procedimento de publicação de uma aplicação “TocáMexer” iniciou-se a 2 de Setembro com o preenchimento de um formulário inicial de registo. Neste formulário foram fornecidas informações como o nome da aplicação, a língua oficial e o tipo de aplicação (figura 7.1). Após o preenchimento, foi facultado um acesso a um painel de controlo onde é possível gerir múltiplas configurações relacionadas com o teste e lançamento da aplicação.

The screenshot shows the 'Criar aplicação' (Create application) form in the Google Play Console. The form is titled 'Criar aplicação' and contains several fields: 'Nome da aplicação' (Application name) with the value 'TocáMexer' and a character count of 9/50; 'Idioma predefinido' (Default language) set to 'Português (Portugal) - pt-PT'; 'App ou jogo' (App or game) with radio buttons for 'Aplicação' (selected) and 'Jogo'; and 'Gratuita ou paga' (Free or paid) with radio buttons for 'Gratuito' (selected) and 'Pagas'. A warning message at the bottom states: 'Pode editar estas informações até publicar a app. Após a publicação, não é possível alterar uma app gratuita para paga.' The form has 'Cancelar' and 'Criar aplicação' buttons at the bottom right.

Figura 7.1: Interface inicial de criação da aplicação

Numa fase seguinte, procedeu-se para o início da distribuição da aplicação sob a forma de teste interno. Este tipo de teste caracteriza-se pela distribuição online da aplicação para um grupo de até 100 testadores. Esta versão da aplicação não necessitou de revisão por parte da Google Play Store, pelo que o prazo de disponibilização foi imediato. Este tipo de teste teve como principal intuito efetuar de

forma simples uma rápida distribuição de uma determinada versão inicial da aplicação para um grupo restrito de pessoas. A primeira distribuição desta versão ocorreu dia 4 de Setembro e foram distribuídas 5 versões diferentes ao longo de um período de 6 dias.

Posteriormente, realizou-se o lançamento de uma versão de teste fechado para ensaiar a aplicação num conjunto mais alargado de pessoas. Este processo teve começo no dia 11 de Setembro onde foram lançadas variadas versões durante 24 dias. Ao contrário da versão em teste interno, esta versão requereu o preenchimento de informação adicional sobre a aplicação e a configuração da Ficha da loja, tal como é possível verificar na figura 7.2.

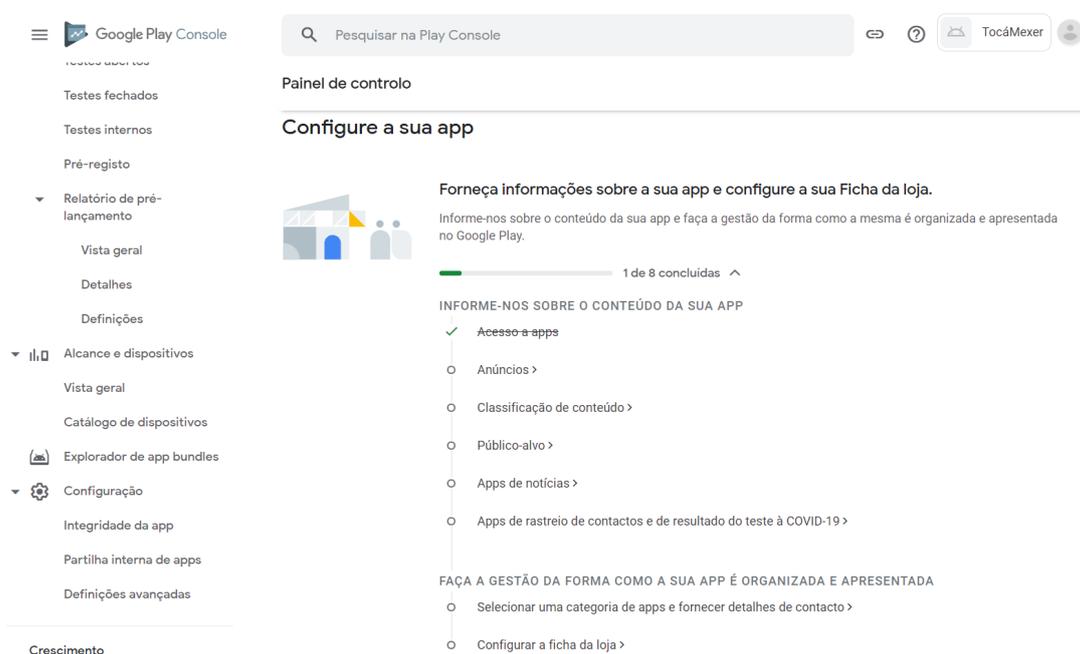


Figura 7.2: Interface de configuração da aplicação

Estas informações foram fornecidas de forma a garantir que a aplicação é segura para os utilizadores previstos. Em primeiro lugar, foi necessário submeter um documento de política de privacidade para partilhar a forma como são tratados os dados confidenciais do utilizador e se a aplicação contém ou não anúncios. Por outro lado, foram fornecidos detalhes sobre o tipo de aplicação e o público-alvo. Além disso foi necessário descrever como são utilizadas quaisquer autorizações confidenciais no âmbito da aplicação. Por fim, configurou-se a Ficha de Loja apresentada na Google Play Store.

Após a submissão desta informação, a aplicação submetida foi sujeita a um processo de revisão da Google Play Store. Além da revisão das informações referidas anteriormente, foram efetuados quatro tipos de testes robotizados:

- Teste de estabilidade - verifica por exemplo se a aplicação falha ou deixa de responder durante a execução da aplicação (figura 7.3);

Figura 7.3: Interface com detalhes do relatório de pré-lançamento - estabilidade

- Teste de desempenho - identifica problemas com o tempo de arranque e com renderização lenta (figura 7.4);

Dispositivo	Média da CPU	Média da rede enviada	Média da rede recebida	Média da memória	Tempo de início a frio
⚠ Dispositivos de teste com avisos					
Xiaomi Redmi 6A	11,21%	302 B	4.96 KB	156 MB	11 mil ms
Nokia Nokia 1	12,70%	116 B	1.77 KB	86.8 MB	19,3 mil ms
✅ Teste dispositivos sem problemas					
Samsung Galaxy S20 5G	1,85%	0 B	0 B	242 MB	1,62 mil ms
Google Pixel 3	2,40%	4 B	2 B	236 MB	-
Google Pixel 5	2,86%	0 B	0 B	327 MB	2,26 mil ms
Samsung Galaxy S9	2,22%	134 B	21.2 KB	214 MB	2,73 mil ms

Figura 7.4: Interface com detalhes do relatório de pré-lançamento - desempenho

- Teste de acessibilidade - averigua se a facilidade de utilização da aplicação em termos de interface de utilizador (figura 7.5);

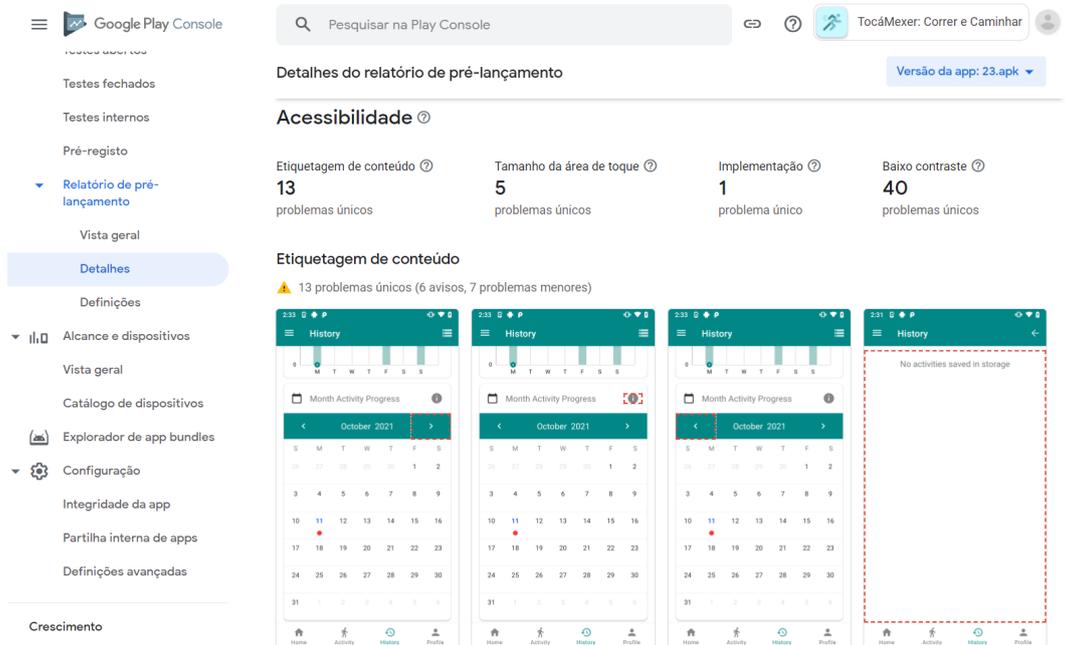


Figura 7.5: Interface com detalhes do relatório de pré-lançamento - acessibilidade

- Teste de segurança e fidedignidade - permite identificar vulnerabilidades de segurança que possam representar um risco para os utilizadores e para o desenvolvedor (figura 7.6);

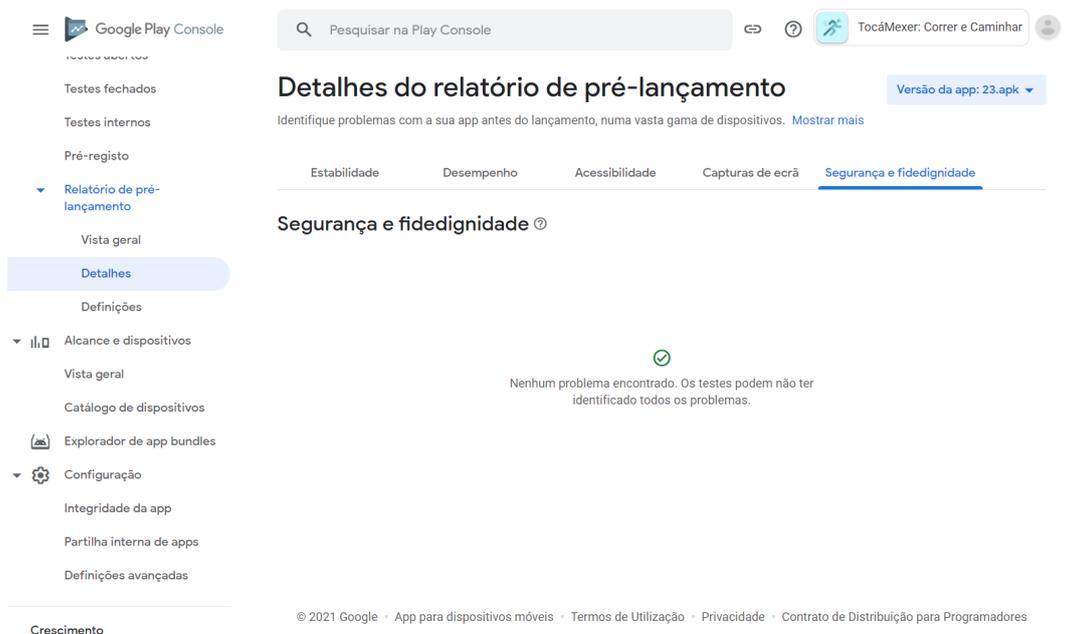


Figura 7.6: Interface com detalhes do relatório de pré-lançamento - segurança e fidedignidade

Numa fase posterior, no dia 7 de Outubro foi lançada uma versão de teste aberto de forma a disponibilizar publicamente a aplicação na Google Play Store. Nesta fase do processo, qualquer pessoa pôde participar neste programa de teste e experimentar a aplicação. Na figura 7.7 é possível observar a Ficha de Loja na Google Play Store desta versão da aplicação.

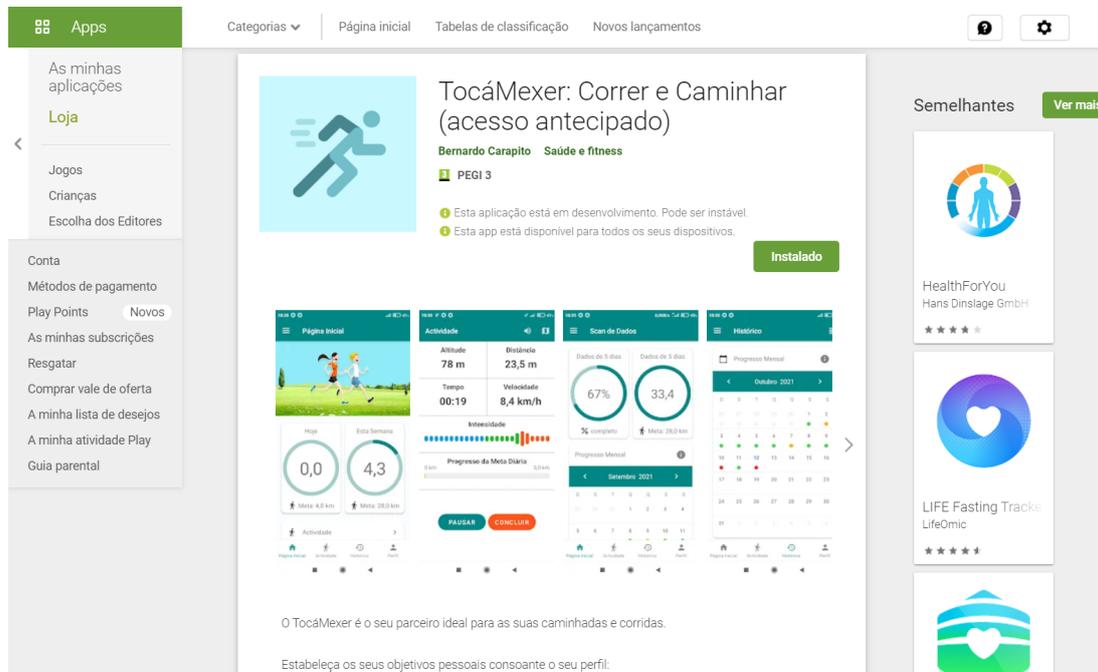


Figura 7.7: Ficha de Loja da aplicação TocáMixer

8

Conclusão

Este trabalho teve como principal objetivo o desenvolvimento e publicação de uma aplicação em Android que monitoriza a atividade física usando GNSS.

Numa fase inicial, foram definidas as especificações pertinentes em colaboração com um médico interno da especialidade de Medicina Geral e Familiar. Esta colaboração revelou-se muito importante já que permitiu estabelecer de forma clara e pertinente os requisitos essenciais para a criação de um produto adequado às necessidades do contexto de utilização médica.

Outro aspeto crucial no desenvolvimento da aplicação foi o constante contacto com outras pessoas próximas que, ao longo do tempo, foram testando a aplicação. Esta interação permitiu recolher feedback relacionado com as várias funcionalidades implementadas ao longo do tempo, permitindo assim corrigir problemas como erros de interface, bugs e defeitos de implementação. Esta interação permitiu ainda identificar e corrigir erros particulares que foram surgindo para vários modelos de telemóveis, como por exemplo erros de layouts (devido à variedade de resolução das telas de telemóvel) ou erros relacionados com as diferenças de versão de sistema operativo.

Os resultados obtidos no capítulo 6 foram importantes para demonstrar a viabilidade da utilização desta aplicação, principalmente num meio não muito urbanizado. Porém, esta solução apresentou algumas falhas no que toca à obtenção de dados de localização em ambiente acentuadamente urbano. Uma das explicações possíveis é a existência de edifícios de elevada altura que contribuem para o bloqueio do campo de visão para os satélites da rede e para a ocorrência de interferência por multicaminho.

Apesar destas vulnerabilidades encontradas, é possível concluir que todos os objetivos inicialmente estabelecidos foram cumpridos com sucesso. Além do desenvolvimento completo de todas as funcionalidades pretendidas, foi também atingido o objetivo de publicar a aplicação na Google Play Store.

Para trabalho futuro, existem inúmeros aspetos a serem revistos. Uma das limitações da aplicação é o limite do tamanho de partilha de informação utilizando um QR Code. Uma solução possível seria o desenvolvimento de um sistema onde seriam enviados e digitalizados consecutivamente vários QR Codes até toda a informação necessária ser transmitida.

Outro aspeto a ser melhorado seria a implementação de uma funcionalidade de comunicação de dados bidirecional. Além da já implementada partilha de dados de desempenho por QR Code, esta funcionalidade poderia ser expandida por forma a possibilitar o médico transmitir a prescrição ao seu paciente sem este necessitar de inserir os objetivos manualmente.

Uma forma de resolver o problema de recolha de dados em zonas muito urbanizadas poderia passar pela criação de um sistema de tratamento de dados de localização mais complexo. Sabendo que as ruas e estradas são um dos principais ambientes de utilização desta aplicação, uma das formas de tornar o cálculo da trajetória mais suave seria utilizar a Roads API da Google, nomeadamente a função de "Snap to Roads". Este serviço retorna a melhor correspondência da geometria de uma trajetória

em estrada para um determinado conjunto de coordenadas enviadas pelo dispositivo. Porém, existem algumas contrariedades nesta solução. Em primeiro lugar, em ambiente urbano, onde por vezes o erro é elevado, a correspondência do trajeto poderá não ser precisa devido à elevada densidade de ruas neste tipo de ambientes. Por outro lado, existem custos associados à utilização desta API da Google.

Por fim, a realização deste projecto foi um grande desafio pessoal. Além de aprofundar os meus conhecimentos sobre GNSS e telecomunicações em geral, este trabalho permitiu-me ter a minha primeira experiência tanto no desenvolvimento de uma aplicação móvel de grande dimensão como no respetivo processo de publicação numa loja de aplicações. Apesar das muitas dificuldades que tive ao longo do desenvolvimento deste projeto, sinto que esta aplicação pode ser uma mais-valia em utilização em contexto médico.

Bibliografia

- [1] European Union Agency for the Space Programme. Constellation Arrangement. Accessed: 12-Set-2021. [Online]. Available: <https://www.gps.gov/multimedia/images/constellation.jpg>
- [2] NASA/University of Texas Center for Space Research. Gravity model figure of Europe and Africa. Accessed: 12-Set-2021. [Online]. Available: <https://appel.nasa.gov/wp-content/uploads/2008/03/1a3-300x290.jpg>
- [3] M. R. Khosravi and S. Samadi, "A blind scalable edge-guided reconstruction filter for smart environmental monitoring through green iomt-uav network," *IEEE Transactions on Green Communications and Networking*, vol. 5, pp. 726 – 736, June 2021.
- [4] Direção Geral do Território. Modelo do Geoide. Accessed: 10-Set-2021. [Online]. Available: <https://www.dgterritorio.gov.pt/geodesia/modelo-geoide>
- [5] World Health Organization. Physical activity. Accessed: 28-Set-2021. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/physical-activity>
- [6] Direção Geral de Saúde. Relatório Sobre Atividade Física 2020, SNS e DGS. Accessed: 08-Out-2021. [Online]. Available: <https://noticias.ecosaude.pt/2021/04/27/relatorio-sobre-atividade-fisica-2020-sns-e-dgs/>
- [7] U.S. Preventive Services Task Force. Healthy Diet and Physical Activity for Cardiovascular Disease Prevention in Adults With Cardiovascular Risk Factors: Behavioral Counseling Interventions. Accessed: 08-Out-2021. [Online]. Available: <https://www.uspreventiveservicestaskforce.org/uspstf/recommendation/healthy-diet-and-physical-activity-counseling-adults-with-high-risk-of-cvd>
- [8] European Global Navigation Satellite Systems Agency. GNSS Market Report Issue 6, October 2019. Accessed: 23-Out-2021. [Online]. Available: <https://www.euspa.europa.eu/gnss-market-report-issue-6-october-2019>
- [9] United States Government. The Global Positioning System. Accessed: 15-Set-2021. [Online]. Available: <https://www.gps.gov>

- [10] European Union Agency for the Space Programme. European GNSS Service Center. Accessed: 23-Out-2021. [Online]. Available: <https://www.gsc-europa.eu>
- [11] GLONASS. Applied Consumer Center GLONASS. Accessed: 23-Out-2021. [Online]. Available: <https://www.glonass--iac-ru>
- [12] Beidou. BeiDou Navigation Satellite System. Accessed: 23-Out-2021. [Online]. Available: <https://www.beidou.gov.cn>
- [13] Google. Fused Location Provider API. Accessed: 24-Out-2021. [Online]. Available: <https://developers.google.com/location-context/fused-location-provider>
- [14] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*. Artech House, 2006.
- [15] European Union Agency for the Space Programme. What is GNSS? Accessed: 12-Set-2021. [Online]. Available: <https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss>
- [16] G. P. Petropoulos and P. K. Srivastava, *GPS and GNSS Technology in Geosciences*. Elsevier Science Publishing Co Inc, 2021.
- [17] H. L. B. Hofmann-Wellenhof and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more*. Wien: Springer, 2008.
- [18] Navstar Global Positioning System. Navstar GPS Space Segment/Navigation User Segment Interfaces. [Online]. Available: <https://www.gps.gov/technical/icwg/IS-GPS-200D.pdf>
- [19] Z. Lu, Y. Qu, and S. Qiao, *Geodesy - Introduction to Geodetic Datum and Geodetic Systems*. Springer-Verlag Berlin Heidelberg, 2014.
- [20] S. T. Kouzeleas, "Conversion of gps data to cartesian coordinates via an application development adapted to a cad modelling system," July 2007. [Online]. Available: <http://papers.cumincad.org/data/works/att/a38d.content.03652.pdf>
- [21] D. C. Hiroko Kato, Keng T. Tan, *Barcodes for Mobile Devices*. Cambridge University Press, 2010.
- [22] D. ADC, *QR Code Essentials*, Denso Automatic Data Capture, 2012.
- [23] Statcounter. Mobile Operating System Market Share Worldwide. Accessed: 14-Set-2021. [Online]. Available: <https://gs.statcounter.com/os-market-share>
- [24] Google. Developer Guides. Accessed: 17-Set-2021. [Online]. Available: <https://developer.android.com/guide>

- [25] Oracle. MySQL Documentation. Accessed: 17-Set-2021. [Online]. Available: <https://dev.mysql.com/doc/>
- [26] M. L. Murphy, *Android's Architecture Components*. CommonsWare, 2019.
- [27] Google. Location - getAltitude(). [Online]. Available: [https://developer.android.com/reference/android/location/Location#getAltitude\(\)](https://developer.android.com/reference/android/location/Location#getAltitude())
- [28] Marktest. 7,2 milhões de portugueses usam smartphone. Accessed: 13-Out-2021. [Online]. Available: <https://www.marktest.com/wap/a/n/id~2700.aspx>
- [29] Anacom. Tráfego de voz e de Internet móvel cresce em 2020. Accessed: 13-Out-2021. [Online]. Available: <https://anacom.pt/render.jsp?contentId=1603502>
- [30] Google. Location - getAccuracy(). [Online]. Available: [https://developer.android.com/reference/android/location/Location#getAccuracy\(\)](https://developer.android.com/reference/android/location/Location#getAccuracy())

