

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JEAN FELIPE PEITER

**APLICATIVO MOBILE PARA INCENTIVO À PRÁTICA DE ATIVIDADES
FÍSICAS**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2021

JEAN FELIPE PEITER

**APLICATIVO MOBILE PARA INCENTIVO À PRÁTICA DE ATIVIDADES
FÍSICAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Vinicius Pegorini

PATO BRANCO
2021



TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

Aplicativo Mobile para Incentivo à Prática de Atividades Físicas

POR

Jean Felipe Peiter

Este trabalho de conclusão de curso foi apresentado no dia **14 de maio de 2021**, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof. MSc Vinicius Pegorini

Professor orientador

Profa. MSc Andreia Scariot Beulke

Professora convidada

Prof. Dr. Robison Cris Brito

Professor convidado

Assinam também:

Prof. Dr. Edilson Pontarolo

Coordenador do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Profa. Dra. Mariza Miola Dosciatti

Responsável pela Atividade de Trabalho de Conclusão de Curso



Documento assinado eletronicamente por (Document electronically signed by) **ANDREIA SCARIOT BEULKE, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em (at) 17/05/2021, às 12:37, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **ROBISON CRIS BRITO, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em (at) 17/05/2021, às 18:55, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **VINICIUS PEGORINI, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em (at) 17/05/2021, às 21:12, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **MARIZA MIOLA DOSCIATTI, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em (at) 17/05/2021, às 23:12, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por (Document electronically signed by) **EDILSON PONTAROLO, COORDENADOR(A) DE CURSO/PROGRAMA**, em (at) 18/05/2021, às 09:00, conforme horário oficial de Brasília (according to official Brasilia-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site (The authenticity of this document can be checked on the website) https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador (informing the verification code) **2036730** e o código CRC (and the CRC code) **1D375781**.

RESUMO

O sedentarismo vem sendo considerado o mal deste século e a atividade física como aliada direta ao tratamento farmacológico e ao combate e à prevenção da obesidade. São muitos os dados que sustentam essas afirmações e, da mesma forma, são muitas as iniciativas para motivar as pessoas na realização de atividades físicas. Além das tradicionais atividades em academias, há as atividades realizadas ao ar livre como os esportes coletivos, as caminhadas, as corridas e andar de bicicleta, para citar algumas. Apesar da quantidade e da diversidade de opções de atividades físicas que são oferecidas na maioria das cidades, realizá-las ainda não é opção para todas as pessoas que deveriam ou poderiam fazê-lo. Aplicativos computacionais, especialmente para dispositivos móveis como, *smartphones*, *smartwatches* e pulseiras eletrônicas, têm sido aliados na prática da realização de atividades físicas e para auxiliar a criar e manter o hábito de realizar esse tipo de exercício. Esses aplicativos podem ser utilizados como um auxílio no desafio de cumprir metas, como andar uma quantidade de quilômetros definidos pela pessoa. Considerando a possibilidade de uso de tecnologias como auxiliares na prática de atividade física, por meio deste trabalho é reportado o desenvolvimento de um aplicativo para dispositivos móveis Android para registrar dados de corridas e caminhadas realizadas pelos seus usuários. No aplicativo os dados coletados durante a realização do percurso da caminhada ou corrida, visando que o usuário possa avaliar o seu gasto energético e ter acesso às informações que possam auxiliá-lo no controle calórico e possam ser importantes para a sua saúde e como motivação para a prática de atividade física. O aplicativo utiliza mecânicas de jogos para incentivar corridas e caminhadas a serem praticadas, dando pontos de experiência após cada percurso finalizado, e duplicando essa pontuação se a atividade for realizada simultaneamente por dois usuários. Os dispositivos conectam-se e compartilham dados das atividades durante as mesmas por meio do banco de dados em tempo real em nuvem, o Firebase Realtime Database.

Palavras-chave: Aplicativo Android. Aplicativo para acompanhamento de prática de atividade física. Aplicativo móvel.

ABSTRACT

The sedentary lifestyle has been considered the century's great evil and physical activity considered the direct ally to pharmacological treatment, to prevent and fight against obesity. There is a lot of data to confirm this and a lot of initiatives motivating people to practice physical exercise. In addition to traditional physical activities, there are plenty of outdoors exercises, such as team sports, walking, running, and cycling. Despite the number and diversity of physical activities offered in most cities, exercising still is not the choice to all the people who should or could do it. Computer applications, especially those for mobile devices like smartphones, smartwatches, and smart bands, have been allies in performing physical activities and helping to create habits to do this kind of activity. Those apps can be used to challenge personal goals achievements, like walking a certain number of miles. Considering the usage of technologies to help on physical activity practice, this essay reports the development of an Android mobile app that records data from walks and runs done by its users. Data will be collected during the course, aiming the user to measure his body energy expenditure, and having access to meaningful information that can help him on calories control, and can be relevant for his health, like the motivation to exercise. The app has games mechanics to encourage walks and runs being done, giving experience points after each finished course, and doubling it if the activity is done by two users, simultaneously. Both devices connect and share data of the activities during it, through the Firebase Realtime Database, on the cloud.

Key words: Android app. Mobile app to collect data from walkings and running. Mobile app.

LISTA DE FIGURAS

Figura 1 – Pilha de software do Android.....	20
Figura 2 – Diagrama de casos de uso	36
Figura 3 – Diagrama de entidades e relacionamentos do banco de dados	39
Figura 4 – Protótipos Lo-Fi. a) Cadastro de dados; b) Atividade; c) Níveis de Progresso.	42
Figura 5 – Protótipos Hi-Fi. a) Cadastro de dados; b) Atividade; c) Níveis de Progresso.....	43
Figura 6 – Tela de Cadastro de Dados. a) Formulário Completo; b) Seleção de Data	45
Figura 7 – Tela Principal	46
Figura 8 – Busca de Usuários Para Atividade em Dupla	48
Figura 9 – Contagem Regressiva para Atividade	49
Figura 10 – Tela de Atividade. a) Estado em Andamento; b) Estado Pausado.....	50
Figura 11 – Detalhes da Atividade Realizada. a) Idioma em português; b) Idioma em inglês.	51
Figura 12 – Níveis de Progresso.....	52
Figura 13 – Histórico de Atividades.....	53
Figura 14 – Hierarquia de Pacotes.....	54

LISTAGEM DE CÓDIGOS

Listagem 1 – Monitoramento de solicitações para Atividade.....	56
Listagem 2 – Controle de Publicação e Inscrição de Atividade em Dupla	58
Listagem 3 – Monitoramento da Atividade do Companheiro	59
Listagem 4 – Exibição da Rota no Mapa e Afastamento de Câmera	60
Listagem 5 – Cálculo de gasto calórico.....	63
Listagem 6 – Cálculo de gasto calórico.....	64
Listagem 7 – Criação do banco de dados	65
Listagem 8 – Interface do Banco de Dados para Níveis de Progresso	66

LISTA DE QUADROS

Quadro 1 - Lista de ferramentas e tecnologias	29
Quadro 2 – Requisitos funcionais.....	34
Quadro 3 – Requisitos não funcionais.....	35
Quadro 4 – Caso de uso cadastrar usuário.....	36
Quadro 5 – Caso de uso iniciar atividade	37
Quadro 6 – Caso de uso pausar e retomar atividade.....	37
Quadro 7 – Caso de uso finalizar atividade.....	38
Quadro 8 – Caso de uso visualizar estatísticas	39
Quadro 9 – Campos da tabela usuario_nivel.....	40
Quadro 10 – Campos da tabela atividade	40
Quadro 11 – Campos da tabela atividade_posicao.....	41

LISTA DE TABELAS

Tabela 1 – Uso e gosto por jogos por faixa etária	25
Tabela 2 – Tempo semanal dedicado à prática de atividade física por faixa etária	26
Tabela 3 – Prática caminhada e corrida por faixa etária.....	26
Tabela 4 – Atividade física praticada e gostar e utilizar jogos de celular	27
Tabela 5 – Relação entre tempo livre e tempo utilizado para a prática de atividade física.....	27
Tabela 6 – Relação entre tipo de sistema operacional e praticar corrida ou caminhada	28

LISTA DE GRÁFICOS

Gráfico 1 – Faixas etárias	22
Gráfico 2 – Uso de jogos no celular	23
Gráfico 3 – Prática de atividade física.....	24
Gráfico 4 – Tempo livre diário	24
Gráfico 5 – Realização de atividade física sozinho ou acompanhado.....	25
Gráfico 6 – Uso de aplicativo para medir desempenho de atividade física.....	27
Gráfico 7 – Uso de Sistema Operacional Móvel	28

LISTA DE SIGLAS

MET	<i>Metabolic Equivalent of Task</i>
API	<i>Applications Programming Interface</i>
DAO	<i>Data Access Object</i>
DVM	<i>Dalvik Virtual Machine</i>
GPS	<i>Global Positioning System</i>
GUI	<i>Graphical User Interfaces</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
JNI	<i>Java Native Interface</i>
OMS	Organização Mundial de Saúde
SDK	<i>Software Development Kit</i>
SO	Sistema Operacional

SUMÁRIO

1 INTRODUÇÃO	13
1.1 CONSIDERAÇÕES INICIAIS	13
1.2 OBJETIVOS	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos	14
1.3 JUSTIFICATIVA	15
1.4 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 ATIVIDADE FÍSICA E SAÚDE.....	17
2.1 DISPOSITIVOS MÓVEIS	18
2.2.1 ANDROID.....	19
3 DADOS DE PESQUISA SOBRE REALIZAÇÃO DE ATIVIDADE FÍSICA.....	22
4 MATERIAIS E MÉTODO	29
4.1 MATERIAIS	29
4.2 MÉTODO	30
5 RESULTADO	33
5.1 ESCOPO DO SISTEMA	33
5.2 MODELAGEM DO SISTEMA	34
5.3 INTERFACE GRÁFICA DO SISTEMA.....	41
5.3.1 Protótipos Lo-Fi.....	41
5.3.2 Protótipos Hi-Fi	42
5.3.3 Interface gráfica no aplicativo	43
5.4 HIERARQUIA DE PASTAS DO PROJETO	53
5.5 IMPLEMENTAÇÃO DO SISTEMA.....	55
5.5.1 Monitoramento de solicitações de atividade em dupla.....	55
5.5.2 Comunicação entre usuários durante um percurso	57
5.5.3 Interação com o mapa da atividade	60
5.5.4 Cálculo de gasto calórico.....	63
5.5.5 Cálculo de pontuação.....	64
5.5.6 Acesso ao Banco de Dados Local	65
6 CONCLUSÃO.....	68

REFERÊNCIAS	70
Apêndice A – Questionário	72

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais que definem o contexto no qual se insere o aplicativo desenvolvido como resultado da realização deste trabalho. Neste capítulo também são apresentados os objetivos e a justificativa do trabalho. E por fim está a organização do texto por meio da apresentação dos seus capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

O sedentarismo está sendo considerado o mal do século e a atividade física a principal solução na prevenção das doenças associadas ao sedentarismo e à obesidade e um ótimo aliado ao tratamento farmacológico (PISA, 2017). A obesidade está sendo considerada uma epidemia global. Dados de 2011 da Cruz Vermelha, citados por Presse (2011), indicavam que no mundo era maior o número de pessoas obesas do que as que passavam fome. Eram 925 milhões de desnutridos e 1,5 bilhão de pessoas obesas. Desconsiderados os problemas de saúde, é certo que o sofrimento dos que passam fome é incomparável ao das pessoas obesas, mas, seja como for, a população mundial está com sobrepeso.

Wannmacher (2016) destaca que obesidade é problema de caráter pandêmico, multiétnico, que ocorre em países de alta, média e baixa renda (particularmente em áreas urbanas), em homens e mulheres e em todas as faixas etárias. O sobrepeso ocorre pelo desequilíbrio entre o consumo e o gasto de calorias. As pessoas têm consumido alimentos mais calóricos (e também menos saudáveis) paralelamente ao aumento do sedentarismo.

O sedentarismo é apontado como a principal causa da obesidade. E a obesidade agrava o sedentarismo porque para pessoas com sobrepeso se torna mais difícil realizar atividade física. Esses dois fatores estão relacionados ao desenvolvimento de doenças como hipertensão arterial, colesterol, triglicérides, arteriosclerose, cirrose não alcoólica, osteoporose, resistência à insulina e diabetes (PISA, 2017).

Os meios urbanos de transporte individuais e coletivos disponíveis, a relativa facilidade de aquisição de veículos, a falta de vias urbanas adequadas para pedestres e ciclistas podem ser indicados como alguns dos fatores que têm contribuído para que as pessoas andem menos. E a realização de atividades físicas em períodos de descanso do trabalho ou da escola tem sido substituída por entretenimento eletrônico. Essas não são as únicas causas da obesidade e do sedentarismo, mas é evidente que elas existem em grande medida. Até porque a obesidade é considerada doença multifatorial, ocorrendo pela interação de fatores genéticos e condições do

ambiente. Muitos dos mecanismos fisiopatológicos que levam a obesidade são ainda desconhecidos. Contudo, entre os fatores ambientais que devem ser corrigidos, destacam-se hábitos alimentares incorretos e a falta de atividade física (WANNMACHER, 2016).

A Organização Mundial de Saúde (OMS, 2005) preconiza que uma pessoa para ser considerada fisicamente ativa deve realizar pelo menos 150 minutos de atividade física por semana, podendo ser divididos em pelo menos 3 dias na semana e com sessões mínimas de 30 minutos (PISA, 2017).

Entre os aplicativos para dispositivos móveis com a finalidade de incentivar o usuário a praticar atividades físicas cita-se o *Zombies, Run!* (ZOMBIES, RUN!, 2019) que possui uma temática apocalíptica/pós-apocalíptica e conforme os percursos são realizados, a pessoa ouve a narrativa da história e coleta suprimentos para sua vila. Esses suprimentos são necessários para que a vila cresça, obtenha recursos melhores e mais moradores, entre outros.

Considerando esse contexto de necessidade de realização de prática de atividade física e a disponibilidade que as pessoas em geral possuem de dispositivos móveis, por meio deste trabalho foi desenvolvido um aplicativo que visa incentivar a prática da realização de atividade física nas modalidades corrida e caminhada, fazendo com que o usuário ganhe pontos de experiência por cada atividade realizada, e possibilitando dois usuários a realizarem um percurso ao mesmo tempo, conectados pelo aplicativo, e, assim, duplicando a pontuação que receberiam caso a atividade fosse a praticada individualmente.

1.2 OBJETIVOS

A seguir são apresentados os objetivos do aplicativo proposto neste trabalho, sendo que o objetivo geral está relacionado com o resultado principal e os objetivos específicos ilustram funcionalidades e complementam o objetivo geral.

1.2.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis Android para registrar dados de corridas e caminhadas realizadas pelos seus usuários.

1.2.2 Objetivos Específicos

- Criar um sistema de pontuação para as pessoas que participam de corridas e caminhadas quando uma atividade é finalizada;
- Possibilitar que dois usuários colem mais pontos pela realização conjunta de caminhadas e corridas;
- Registrar dados durante a realização do percurso da caminhada ou corrida que auxiliem o usuário a avaliar o seu gasto energético e ter acesso a informações que possam auxiliá-lo no controle calórico e possam ser importantes para a sua saúde e como motivação para a prática de atividade física.

1.3 JUSTIFICATIVA

Em 2018 o Brasil possuía, oficialmente, mais *smartphones* ativos do que pessoas. Dados da Fundação Getúlio Vargas, da sua pesquisa anual sobre uso de tecnologia, revelou que em abril de 2018 havia 220 milhões de celulares em funcionamento no país que contava com uma população de 207,6 milhões de habitantes, de acordo com os dados do Instituto Brasileiro de Geografia e Estatística (IBGE) (DEMARTINI, 2018). O Brasil registrou 231,8 milhões de linhas móveis no mês de novembro de 2018, com forte incremento dos chips 4G, de acordo com dados na base das prestadoras de telefonia (AVELAR; DUARTE, 2019).

Os dados de quantidade de dispositivos móveis existentes no Brasil aliados aos dados de obesidade apresentados na introdução justificam o uso dessa tecnologia como auxiliar na prática de atividade física, visando, assim, uma vida mais saudável, com redução de sobrepeso e de doenças relacionadas à obesidade e ao sedentarismo.

Visando incentivar e estimular a prática de atividade física, o aplicativo desenvolvido como resultado deste trabalho será como um jogo, desafiando o usuário a realizar caminhadas e corridas na busca de recompensas. Essas recompensas são incrementadas se a atividade é realizada em dupla. O aplicativo permite a sincronização dos dispositivos móveis. Essa forma de recompensa visa que o aplicativo incentive outras pessoas a realizarem atividades físicas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos. Este é o primeiro e apresenta as considerações iniciais com o contexto do aplicativo desenvolvido, os objetivos e a justificativa. O Capítulo 2 apresenta sobre atividade física e saúde e sobre dispositivos móveis e Android. No Capítulo 3 são apresentados dados obtidos por meio de uma pesquisa realizada para auxiliar

na coleta de requisitos do aplicativo. No Capítulo 4 estão as ferramentas e as tecnologias utilizadas na modelagem e que serão utilizadas na implementação subsequente do aplicativo. No Capítulo 5 é apresentado o resultado da realização do trabalho que é a definição dos requisitos do aplicativo e a sua modelagem na forma de casos de uso e diagramas de entidades e relacionamentos do banco de dados, interface do sistema, hierarquia de pastas e implementação do sistema. Por fim, estão as referências utilizadas no texto.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos e dados sobre saúde e tecnologia e como as mesmas se conectam com a aplicação desenvolvida.

2.1 ATIVIDADE FÍSICA E SAÚDE

A atividade física pode ser definida como qualquer movimento corporal produzido que requer gasto de energia acima dos níveis de repouso (FREIRE *et al.*, 2014). A prática de atividade física tem sido considerada como um meio de preservar e melhorar a saúde e a qualidade de vida da pessoa em qualquer idade.

A obesidade na infância e na adolescência acarreta altos custos para os sistemas de cuidado à saúde. A obesidade durante a infância e a adolescência tem consequências adversas na mortalidade precoce e morbidade física na idade adulta a curto e a longo prazo (BARROS; SILVA, 2017).

Dados da OMS indicam que 20% dos adultos e 80% dos adolescentes não praticam exercícios físicos com a frequência e a intensidade adequadas para a sua faixa etária. Sedentarismo pode agravar riscos de doenças crônicas não transmissíveis (OMS, 2018).

As doenças crônicas não transmissíveis figuram entre os principais problemas de saúde pública no mundo (WORLD..., 2011). No Brasil, essas doenças se tornaram a principal prioridade na área de saúde pública. Isso porque elas são responsáveis por 72% das mortes em 2007 (SCHMIDT *et al.*, 2011). Esse percentual é 20% superior à média mundial (WORLD..., 2011).

A prática de atividade física em níveis recomendados tem se mostrado capaz de auxiliar na prevenção de doenças ou de atenuar os seus efeitos em pessoas que já apresentam essas doenças (CRUZ; BERNAL; CLARO, 2018). O Ministério da Saúde adota para a população brasileira a recomendação proferida pela OMS, de prática de atividade física, sugerindo o mínimo de 150 minutos por semana de atividade de intensidade moderada ou equivalente (75 minutos de atividade de intensidade vigorosa).

Para obter-se a quantidade de calorias queimadas durante uma atividade física, é utilizada a constante MET, ou equivalente metabólico de tarefa (do inglês *metabolic equivalent of task*), em que, quanto maior a intensidade da atividade, maior o valor do MET. Os valores padrão dos METs foram estabelecidos no Compêndio de Atividades Físicas (originalmente, *The Compendium of Physical Activities*, em inglês), sem levar em consideração idade, massa

corpórea e sexo (The Compendium of Physical Activities, 2020). Uma unidade MET é medida pelo volume de oxigênio por minuto, sobre o peso da pessoa por minuto, que é equivalente à uma caloria por quilograma por hora. Portanto, uma pessoa de 80kg que percorreu 6km no tempo de uma hora, teve um gasto energético aproximado de 504kcal.

A prática de atividades físicas pode ser auxiliada por recursos tecnológicos. Além dos aplicativos e sistemas para os profissionais, os dispositivos móveis podem ser utilizados pelas pessoas que realizam atividades físicas com diversas finalidades, de motivação a monitoramento de sinais vitais.

Os recursos dos dispositivos móveis tais como *smartphones* os tornam parte essencial da vida das pessoas. Esses dispositivos podem prover muitos serviços úteis para gerenciar a saúde do usuário (TAWALBEH; BAKHADER, 2016).

2.2 DISPOSITIVOS MÓVEIS

Os dispositivos sem fio (*wireless*), tais como os telefones celulares, mantêm os seus usuários conectados ao mundo exterior em qualquer tempo e lugar que possua esse tipo de serviço. Desenvolvimento de aplicações para dispositivos *wireless* será uma grande demanda para os próximos anos. Aplicações para dispositivos móveis proveem um novo conjunto de desafios para os seus projetistas. *Smartphones* têm sido usados para aplicações pessoais e por grandes corporações (GADHIYA; WANDRA; VAGHELA, 2012).

Os dispositivos móveis têm rapidamente ganhado aceitação como plataforma multimídia. Os desenvolvedores desses aplicativos podem trabalhar com áudio, vídeo, mensagens multimídia e outros para criar aplicações móveis com conteúdo rico (GAVALAS; ECONOMOU, 2011).

Computação móvel é um campo de comunicação móvel que descreve como os dispositivos móveis podem comunicar-se entre si. Como dispositivos móveis, eles podem acessar a Internet em qualquer lugar e horário (GADHIYA; WANDRA; VAGHELA, 2012).

Embora a escolha da plataforma de desenvolvimento é amplamente conduzida pelo mercado, ela também depende das características das plataformas disponíveis e dos requisitos particulares da aplicação (GAVALAS; ECONOMOU, 2011). Esses autores citam como principais, as seguintes plataformas para dispositivos móveis: Java Micro Edition, .Net Compact Framework, Flash Lite e Android.

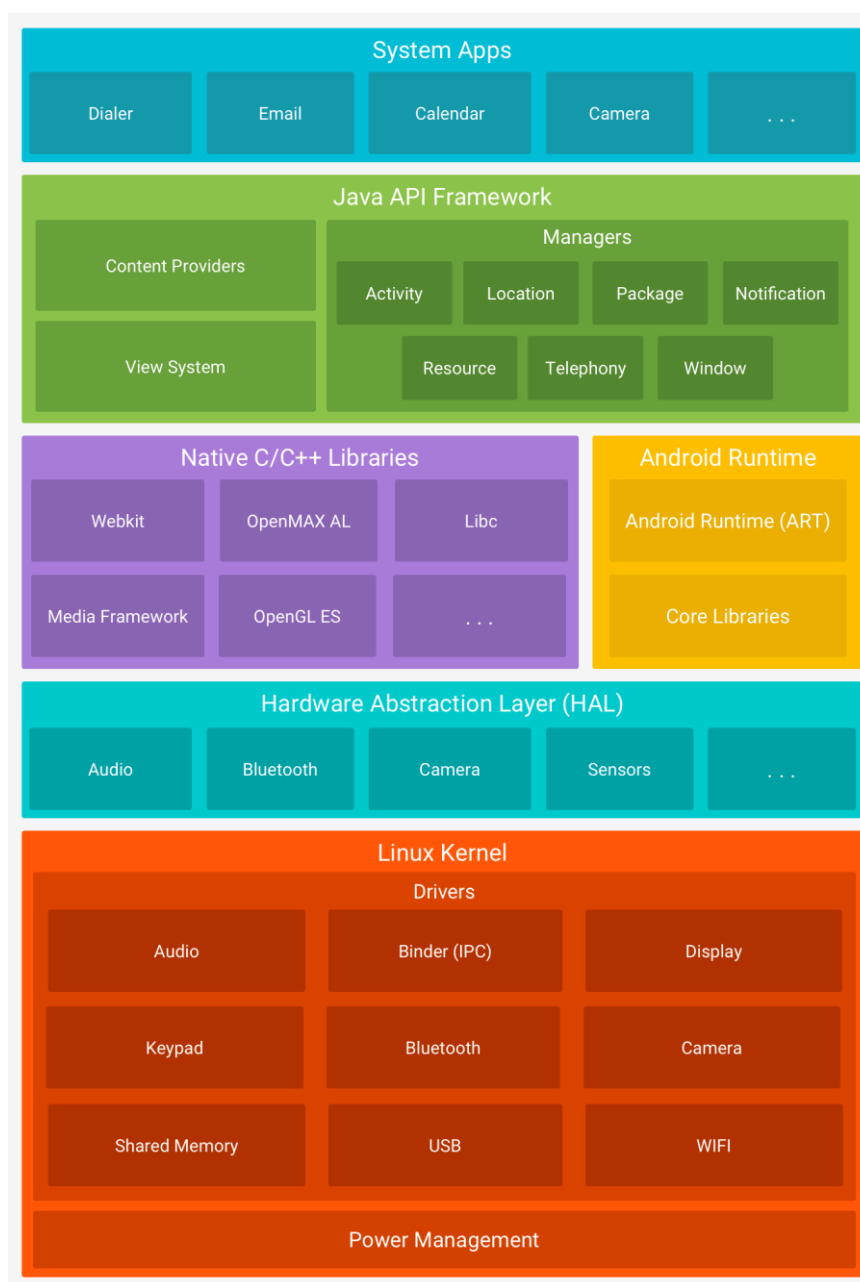
2.2.1 ANDROID

A Google lançou o Android em 2007 para promover padrões abertos para dispositivos móveis. Android é uma plataforma de software livre Apache com uma licença *open source* para dispositivos móveis baseada em Linux (GAVALAS; ECONOMOU, 2011).

A plataforma Android é um conjunto de serviços de *software* para dispositivos móveis (GUANA et al., 2012). Esses serviços são agrupados e organizados em camadas, na ideia de pilha, incluindo: sistema operacional (SO), *middleware*, interface com o usuário e aplicação (BRAY, 2014). Li e Wang (2004) destacam que os aplicativos Android refletem a ideia de hierarquia no desenvolvimento de aplicações usando a linguagem Java: a camada de mais baixo nível usa C e C++ e sobre ela está código Java que utiliza as bibliotecas C e C++ por meio de *Java Native Interface* (JNI).

A Figura 1 apresenta a estrutura ou arquitetura da plataforma Android (ANDROID ARCHITECTURE, 2019), conhecida também como pilha de software.

Figura 1 – Pilha de software do Android



Fonte: Android Developers (2019, p.s.n.).

A arquitetura da plataforma Android apresentada esquematicamente na Figura 2 é dividida em cinco camadas, com funcionalidades e comportamentos específicos. As camadas são: aplicações, *frameworks*, bibliotecas, *runtime* e *kernel* do Linux (GUANA et al., 2012, PRIMORAC; RUSSO, 2012):

a) Camada de aplicação – contém as aplicações para o usuário. Exemplos: clientes de *e-mail*, navegadores, utilitários e jogos. Essas aplicações são as básicas que vem com o SO e as que são desenvolvidas para propósitos específicos que estendem ou utilizam funcionalidades do SO.

b) Camada de *framework* da aplicação – é o conjunto extensível de componentes de software usados pelas aplicações que executam no dispositivo móvel.

c) A camada de biblioteca – é o conjunto de pacotes C/C++ usados pelo *framework* da aplicação para gerenciar funcionalidades como as de renderização da tela, segurança de dispositivos, persistência de aplicações, entre outras.

d) A camada de *runtime* – é composta pela máquina virtual Dalvik e as bibliotecas básicas (*core libraries*) que especificam o ambiente de execução da aplicação no sistema operacional. A *Dalvik Virtual Machine* (DVM) é utilizada para iniciar as bibliotecas básicas que são escritas em Java.

e) Camada de *kernel* do Linux – é uma versão do Linux customizada que provê as funcionalidades de baixo nível do sistema operacional como gerenciamento de memória e de processos, agendamento de tarefas e segurança.

As aplicações Android são principalmente escritas em Java e compiladas em um formato executável Dalvik (dex), que é um código de *bytes* customizado. Cada aplicação executa em seu próprio processo, com sua instância da máquina virtual Dalvik que executa arquivos “dex” que são convertidos em tempo de compilação a partir de classe padrão e arquivos no formato “jar” (GAVALAS; ECONOMOU, 2011).

Os desenvolvedores Android têm acesso completo aos *frameworks* e às *Applications Programming Interface* (API) que as aplicações principais (*core*) usam e as bibliotecas de desenvolvimento de software da Google. O *Software Development Kit* (SDK) Android fornece suporte para o desenvolvimento de aplicativos com funcionalidades ricas, como manipulação de telas *touch-screen*, acelerômetros, gráficos 3D e *Global Positioning System* (GPS) e a colaboração entre aplicações como *e-mail*, mensagens, calendário, redes sociais e serviços baseados em localização (GAVALAS; ECONOMOU, 2011).

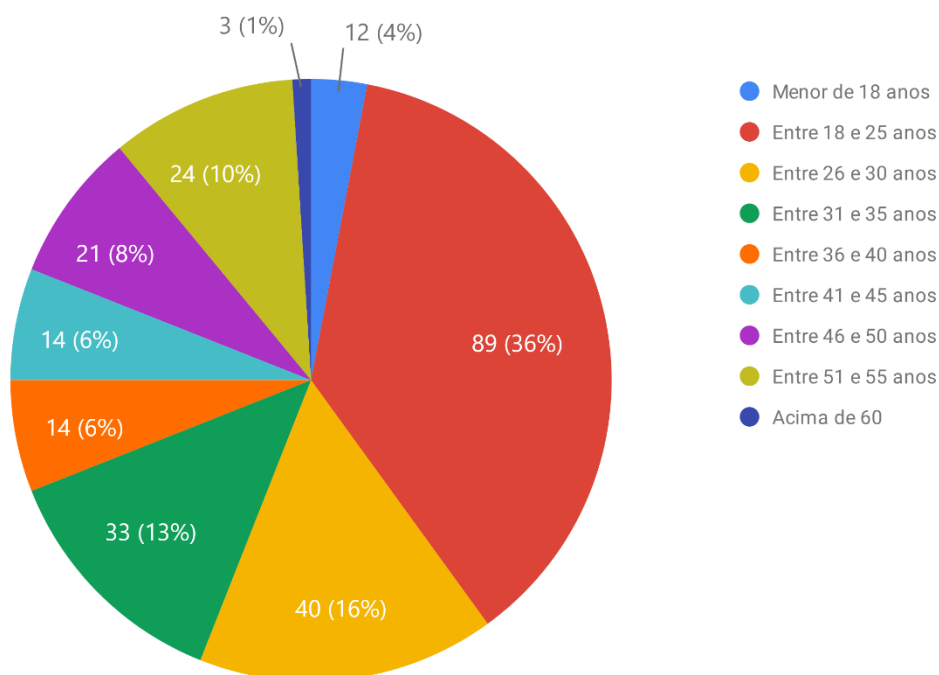
3 DADOS DE PESQUISA SOBRE REALIZAÇÃO DE ATIVIDADE FÍSICA

Uma pesquisa informal foi realizada visando coletar dados que pudessem auxiliar na definição dos requisitos do aplicativo objeto deste trabalho, que é na forma de jogo para dispositivos móveis e voltado para motivação na realização de corridas e caminhadas. Essa pesquisa foi realizada por meio de um formulário (Google Forms que consta no APÊNDICE A) e respondida por 250 pessoas, sendo 53% do sexo masculino e 47% do sexo feminino. A seguir são apresentados dados dessa pesquisa. Na pergunta sobre sexo havia a opção “Prefiro não dizer”, mas essa não foi escolhida por nenhum dos respondentes.

No rótulo dos dados nos gráficos em setores a seguir é apresentada a quantidade e o respectivo percentual, entre parênteses. Nas tabelas a seguir os dados são apresentados como quantidade, exceto quando explicitados que o dado está apresentado em percentual.

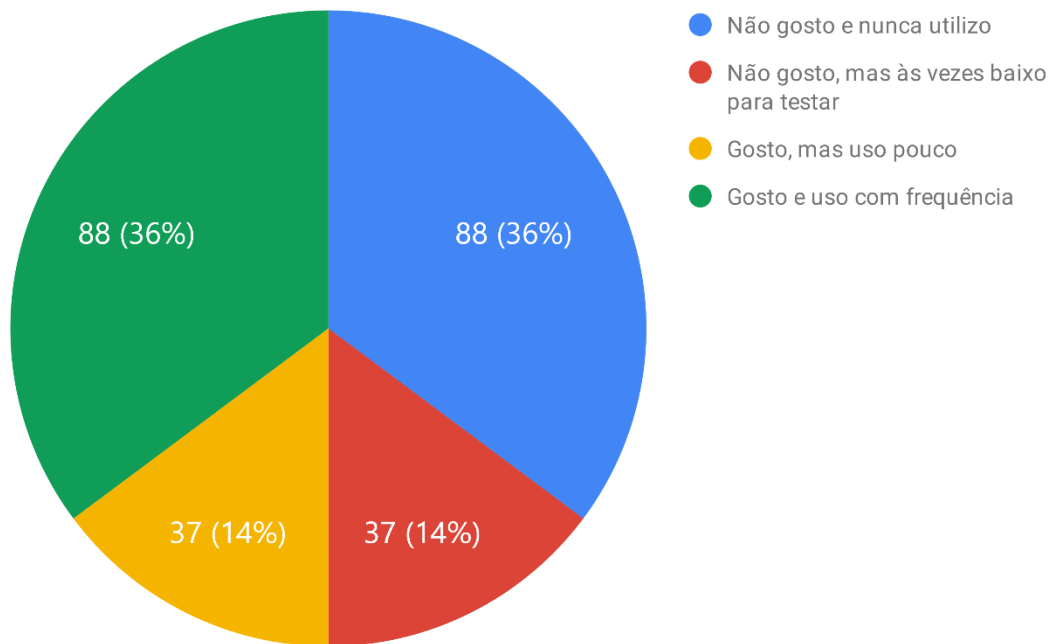
O Gráfico 1 apresenta os dados por faixa etária dos que responderam à pesquisa. O destaque para a quantidade foram as pessoas entre 18 e 25 anos, com 89 participantes.

Gráfico 1 – Faixas etárias



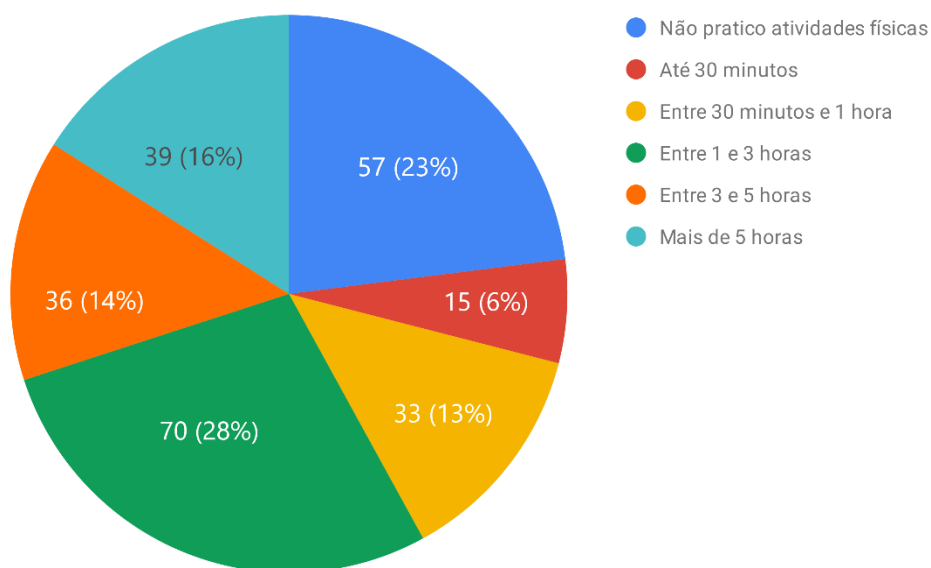
Fonte: Autoria própria.

Dados referentes ao uso de jogos no celular são apresentados no Gráfico 2. O percentual de pessoas que gostam e usam pouco é o mesmo de pessoas que não gostam e nunca utilizam.

Gráfico 2 – Uso de jogos no celular

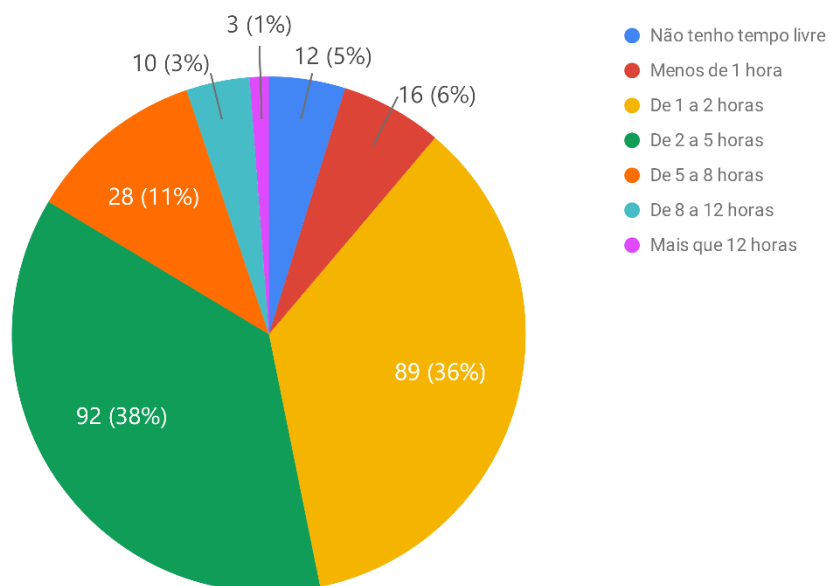
Fonte: Autoria própria.

Dados do tempo semanal dedicado à prática de atividade física são apresentados no Gráfico 3. Apenas 23% das pessoas que responderam a essa pergunta não praticam atividade física, considerando que a faixa de idade dos participantes é bastante diversificada e que 30% os participantes trabalham e estudam, 45% somente trabalham e 21% somente estudam. Restando 7% de pessoas que não trabalham ou estudam.

Gráfico 3 – Prática de atividade física

Fonte: Autoria própria.

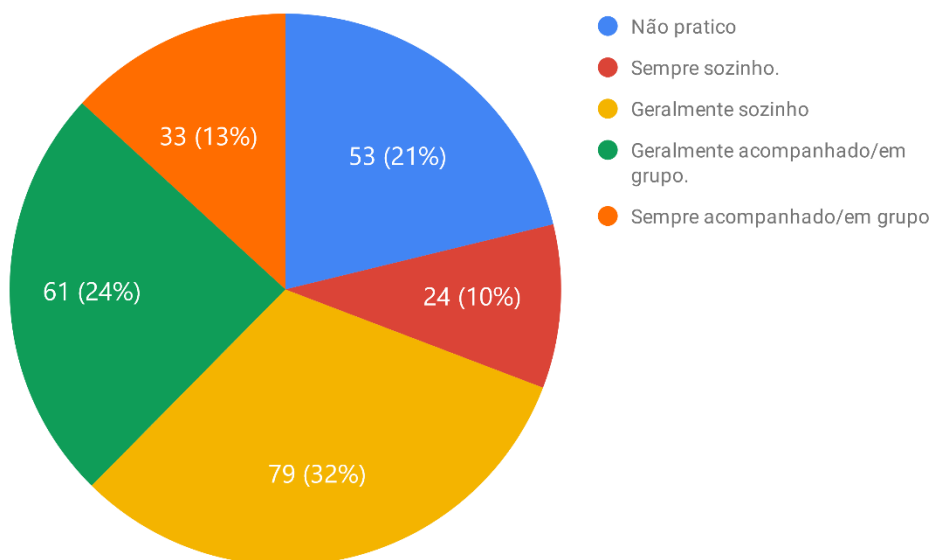
Dados do tempo livre diário são apresentados no Gráfico 4. O percentual de pessoas que diz ter de 1 a 5 horas de tempo livre por semana é de 74%.

Gráfico 4 – Tempo livre diário

Fonte: Autoria própria.

Dados se a atividade física é realizada com a pessoa sozinha ou acompanhada são apresentados no Gráfico 5.

Gráfico 5 – Realização de atividade física sozinho ou acompanhado



Fonte: Autoria própria.

A Tabela 1 apresenta o resultado do cruzamento dos dados entre a faixa etária e o gosto e uso de celulares.

Tabela 1 – Uso e gosto por jogos por faixa etária

Gosta e utiliza jogos Idade	Gosto e uso com frequência	Gosto, mas uso pouco	Não gosto e nunca utilizo	Não gosto, mas às vezes baixo para testar
Menor de 18 anos	2	5	1	0
Entre 18 e 25 anos	21	37	19	14
Entre 26 e 30 anos	6	17	8	9
Entre 31 e 35 anos	4	15	11	3
Entre 36 e 40 anos	1	4	6	3
Entre 41 e 45 anos	3	1	9	1
Entre 46 e 50 anos	3	5	12	1
Entre 51 e 55 anos	1	4	19	1
Entre 56 e 60 anos	0	0	2	1

Fonte: Autoria própria.

Na Tabela 2 estão os dados da frequência semanal da prática de atividade física por faixa etária.

Tabela 2 – Tempo semanal dedicado à prática de atividade física por faixa etária

Idade	Tempo semanal atividade física					
	Até 30 minutos	Entre 1 e 3 horas	Entre 3 e 5 horas	Entre 30 minutos e 1 hora	Mais de 5 horas	Não pratico atividades físicas
Menor de 18 anos	2	3	1	0	0	2
Entre 18 e 25 anos	4	23	12	11	13	28
Entre 26 e 30 anos	2	8	6	7	7	11
Entre 31 e 35 anos	4	10	6	4	6	3
Entre 36 e 40 anos	0	8	1	1	2	2
Entre 41 e 45 anos	1	1	4	1	3	3
Entre 46 e 50 anos	1	6	2	5	3	4
Entre 51 e 55 anos	0	8	4	6	4	3
Entre 56 e 60 anos	0	1	0	0	0	1
Acima de 60 anos	0	0	0	0	1	0

Fonte: Autoria própria.

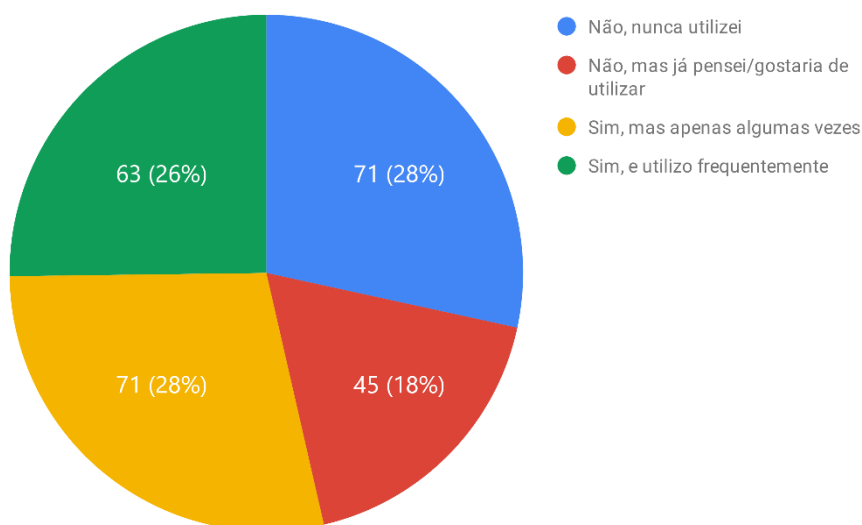
A relação entre faixa etária e número de pessoas que praticam corrida e/ou caminhada como atividade física é apresentada na Tabela 3.

Tabela 3 – Prática caminhada e corrida por faixa etária

Idade	Pessoas que praticam corrida/caminhada	Total pessoas da faixa etária
Menor de 18 anos	5	8
Entre 18 e 25 anos	46	89
Entre 26 e 30 anos	23	40
Entre 31 e 35 anos	24	33
Entre 36 e 40 anos	7	14
Entre 41 e 50 anos	6	14
Entre 46 e 50 anos	15	21
Entre 51 e 55 anos	13	25
Acima de 60 anos	1	1

Fonte: Autoria própria.

Uso de aplicativo para medir o desempenho enquanto realiza corrida, caminhada ou passeio (a pé ou bicicleta) é apresentado no Gráfico 6.

Gráfico 6 – Uso de aplicativo para medir desempenho de atividade física

Fonte: Autoria própria.

A Tabela 4 apresenta a relação entre o tipo de atividade física praticada regularmente e o gosto e o uso de jogos de celular.

Tabela 4 – Atividade física praticada e gostar e utilizar jogos de celular

Atividade física praticada regularmente	Gosta e utiliza jogos				
	Gosto, mas uso pouco	Gosto e uso com frequência	Não gosto e nunca utilizo	Não gosto, mas às vezes baixo para testar	Não gosto e nunca utilizo
Ciclismo/Andar de bicicleta	6	4	10	4	3
Corrida/Caminhada	48	22	22	17	27
Futebol	6	7	9	4	1
Vôlei	4	1	1	0	0
Musculação	27	6	17	12	6
Não pratico	19	10	16	7	0

Fonte: Autoria própria.

A relação entre a quantidade de tempo livre diário que a pessoa possui e a quantidade de tempo destinada à prática de atividade física é apresentada na Tabela 5.

Tabela 5 – Relação entre tempo livre e tempo utilizado para a prática de atividade física

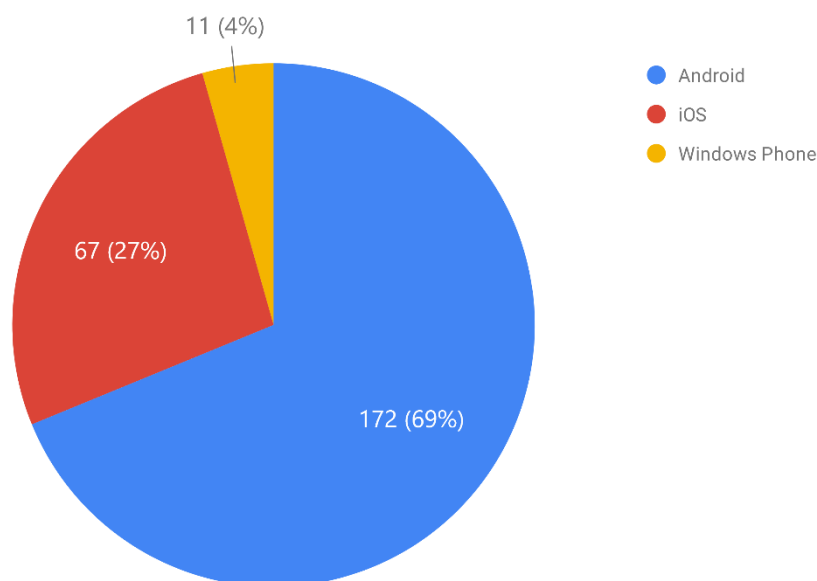
Tempo de prática de atividade física	Tempo livre diário						
	Nenhum	Menos de 1 hora	De 1 a 2 horas	De 2 a 5 horas	De 5 a 8 horas	De 8 a 12 horas	Mais que 12 horas
Não pratico atividades físicas	5	3	19	21	6	3	0
Até 30 minutos	0	3	6	3	1	0	0

Entre 30 minutos e 1 hora	2	3	14	11	2	1	0
Entre 1 e 3 horas	3	5	26	25	8	2	1
Entre 3 e 5 horas	0	1	11	13	7	2	2
Mais de 5 horas	0	1	13	19	4	1	1

Fonte: Autoria própria.

O Gráfico 7 apresenta o sistema operacional utilizado pelos participantes da pesquisa.

Gráfico 7 – Uso de Sistema Operacional Móvel



Fonte: Autoria própria.

A relação entre o SO do celular do usuário e a prática de corridas ou caminhadas é apresentada na Tabela 6.

Tabela 6 – Relação entre tipo de sistema operacional e praticar corrida ou caminhada

Sistema operacional	Corrida/caminhada	Total pessoas	Percentual
Android	99	172	57,56%
iOS	44	67	65,67%
Windows Phone	5	11	45,45%

Fonte: Autoria própria.

4 MATERIAIS E MÉTODO

A seguir estão os materiais e o método utilizados para a modelagem e a implementação do aplicativo a ser obtido como resultado da realização deste trabalho.

4.1 MATERIAIS

O Quadro 1 apresenta a lista de ferramentas e tecnologias que foram utilizadas para o desenvolvimento deste trabalho.

Quadro 1 - Lista de ferramentas e tecnologias

Ferramenta / Tecnologia	Versão	Finalidade
Adobe XD	CC	Prototipação de alta fidelidade de telas
Android Studio	4.2	Ambiente de desenvolvimento integrado
Android SDK	29	Plataforma de desenvolvimento
Balsamiq	3.5.17	Prototipação de baixa fidelidade de telas
Java	11.0.10	Linguagem de programação
Lucid Chart	Web	Modelagem do aplicativo
Room	2.2.5	Banco de Dados
Firebase Auth	19.4.0	Autenticação com redes sociais
Firebase Realtime Database	19.5.0	Banco de dados em nuvem com comunicação em tempo real

Fonte: Autoria própria.

Das ferramentas utilizadas, destacam-se as de prototipação, Balsamiq e Adobe XD, foram utilizadas para criar o fluxo básico do aplicativo, e definir a aparência final e interações de componentes e tela, respectivamente, processo comum no desenvolvimento de aplicativos. Também são relevantes as ferramentas para persistência de dados e comunicação em tempo real, foi utilizado o Room Database, para persistência no dispositivo, o qual é uma abstração do SQLite, para armazenar os dados das atividades realizadas e dos níveis de progresso do usuário. O banco de dados em nuvem Realtime Database foi utilizado para salvar as informações pessoais do usuário, e para realizar a comunicação dos dispositivos durante as atividades.

4.2 MÉTODO

O método consiste nas atividades de levantamento de requisitos, modelagem desses requisitos e na implementação do aplicativo. A seguir é descrita sucintamente a realização dessas etapas.

Levantamento de requisitos

A ideia da implementação do aplicativo surgiu a partir do conhecimento de outros aplicativos existentes nas lojas de aplicativos, para a prática de atividades físicas. Porém, para o que se discorre neste trabalho, foi verificada a possibilidade de implementar funcionalidades ainda não existentes nos aplicativos analisados, como a de sincronização dos aparelhos quando duas pessoas realizam uma atividade de forma conjunta, por exemplo, uma caminhada, e, além disso, esse é um trabalho acadêmico que visa a aplicação de tecnologias e de conceitos de engenharia de software no desenvolvimento de aplicativos computacionais.

No desenvolvimento dessa aplicação verificou-se a utilidade de uma ferramenta, em enredo de jogo, capaz de registrar dados obtidos por sensores durante caminhadas e corridas realizadas pelos seus usuários, calculá-los com base nas medidas corporais informadas pelo usuário e apresentá-los visual e informativamente após a atividade ser praticada. Além da obtenção, do cálculo e do registro de dados, são peças fundamentais do projeto a oferta de pontuações ao final de cada trajeto percorrido e a possibilidade de praticar atividades físicas em dupla, conectado via dados móveis com outro dispositivo Android.

Para o levantamento de requisitos complementares e visando um viés prático para os usuários, foi realizada uma pesquisa que obteve um total de 250 respostas, a fim de traçar perfis de usuários, bem como a prática de atividade física, suas experiências com dispositivos móveis e expectativas com o possível uso do aplicativo oriundo deste trabalho. Foram feitas perguntas como idade, sexo, se a pessoa trabalha, estuda, ambos ou nenhum, se pratica atividades físicas e quanto tempo dedica por semana, quanto tempo ocioso tem, em média, diariamente, e perguntas sobre dispositivos móveis e familiaridade com aplicativos de celular.

Baseado nos resultados da pesquisa, a plataforma Android é a que possui mais usuários, com aproximadamente 7 a cada 10 pessoas. Além disso, em média, cerca de 60% dos usuários de cada SO praticam caminhadas ou corridas como atividade física.

De acordo com as respostas mostradas no Gráfico 5, 112 das 197 pessoas que praticam exercícios fazem isso geralmente ou sempre sozinhas. Esses números podem ser reduzidos com a possibilidade e incentivo de praticar atividades em dupla com o uso do aplicativo, visto que, se praticada acompanhada, a atividade renderá o dobro de pontos para recompensa.

Segundo o Gráfico 6, 179 das 250 pessoas utilizam, já utilizaram ou têm interesse em utilizar um aplicativo para medir/acompanhar desempenho durante atividades físicas.

Segundo os dados da Tabela 5, 223 pessoas têm no máximo cinco horas livres por dia. Dessas, 96 praticam até 3 horas de atividades por semana e 69 não praticam nenhum tipo de exercício físico, totalizando 165 pessoas. Lembrando que, de acordo com a OMS (2005), o tempo recomendado à prática de atividades físicas semanal é de 150 minutos, ou 2,5 horas. Então, estas 165 pessoas estão abaixo ou perto do tempo ideal para ter uma vida saudável e fisicamente ativa.

Outra motivação e circunstância para definição dos requisitos foi o incentivo à prática de atividades físicas, em combate ao sobrepeso à obesidade e ao sedentarismo, independentemente de faixa etária.

Análise e projeto

A análise e o projeto consistiram na modelagem dos requisitos.

Os dados são armazenados localmente, no banco de dados do dispositivo móvel e não há necessidade de controle de permissões de acesso ou de restrição a funcionalidades, pois haverá apenas um perfil de acesso.

O banco de dados consiste em três tabelas para persistência: das informações sobre as atividades que o usuário pratica; dos dados do trajeto de cada atividade; e do nível de progresso do usuário no jogo. Os dados pessoais do usuário são salvos tanto em nuvem quanto localmente, utilizando *Shared Preferences*.

Prototipação

A prototipação das telas do aplicativo teve a finalidade de definir como seria o leiaute da *Graphical User Interface* (GUI), interface gráficas de usuário ou utilizador antes de iniciar a codificação das mesmas. Essa etapa foi dividida em duas partes:

1) Protótipos de baixa fidelidade.

Os protótipos *lo-fi* (baixa fidelidade) serviram de ideia base para a criação das telas, posicionando provisoriamente os elementos, sem atribuição definitiva de cores e funções. Desta etapa até o produto final houve alterações significativas.

2) Protótipos de alta fidelidade.

Os protótipos *hi-fi* (alta fidelidade) são ilustrações e *designs* de telas com aspecto muito próximo às do produto final, nas quais cores, posições e interações com outras telas foram muito semelhantes ao produto final. Os *hi-fi* não passaram por alterações drásticas até chegar ao *design* do final.

Implementação

A implementação da aplicação foi dividida em cinco partes:

1) Internacionalização, cadastro de usuário e interface inicial.

Na fase inicial de implementação foram criados os arquivos de recursos, que terão as finalidades, principalmente, de evitar a prática de *hardcode* e facilitar o uso dos recursos criados e, ainda, de deixar o aplicativo utilizável tanto em português quanto em inglês. Nesta fase também houve o desenvolvimento da interface de login por meio de redes sociais, da tela de boas-vindas e cadastro dos dados do usuário, e da interface de menus.

2) Desenvolvimento da interface da atividade, captura de dados de GPS, cálculos e persistência.

O segundo estágio de desenvolvimento teve como foco as rotinas para obtenção de dados dos sensores e realização dos cálculos necessários para registrar os dados da tabela atividade. Alguns dos dados calculados e formatados foram exibidos em tela enquanto a atividade está em andamento. Foram criadas abstrações e implementações do gerenciador banco de dados Room para persistir estes dados no *smartphone*.

3) Descoberta e comunicação com dispositivos conectados.

A quarta fase foi desenvolvido o fluxo de conexão para permitir a comunicação entre dois celulares que possuem o aplicativo, realizado por meio do banco de dados com atualizações em tempo real, o Firebase Realtime Database.

4) Listagem de atividades realizadas.

No quinto estágio foi criada a tela com o histórico de atividades, com informações básicas e resumidas de cada percurso, e a tela de detalhes, que é exibida quando pressionado um dos itens da lista.

5) Codificação do sistema de pontuação e níveis de progresso.

Na sexta parte foram elaborados os níveis de progresso e algoritmos para o sistema de pontuação. Os pontos são acumulativos, para que o usuário passe de um nível para outro. Foi realizada uma tela para visualização do histórico de progresso.

5 RESULTADOS

Este capítulo apresenta o resultado da realização deste trabalho que é a modelagem e codificação de um aplicativo para dispositivos móveis Android para registrar dados de corridas e caminhadas realizadas pelos seus usuários.

5.1 ESCOPO DO SISTEMA

O aplicativo foi desenvolvido para dispositivos móveis de sistema operacional Android e atende ao contexto de um jogo, no qual é possível iniciar e finalizar corridas e caminhadas. O usuário pode monitorar, em tempo real, sua velocidade, distância percorrida, ritmo médio e calorias gastas durante o percurso. Esses dados são obtidos por meio do uso de sinais de GPS. Os dados são armazenados no aparelho celular utilizando o Room Database, uma camada de abstração sobre o SQLite (ANDROID DEVELOPERS – Room, 2021), provido pela API nativa do Android,

É disponibilizada a realização dos percursos (corrida ou caminhada) em dupla, estando os celulares de dois usuários conectados à internet. Assim, duas pessoas podem cooperar no progresso de níveis do aplicativo e incentivarem-se mutuamente ou unilateralmente para a prática de atividade física.

Na primeira utilização do aplicativo, o usuário deve autenticar-se com uma conta do Google ou Facebook, para que seja possível armazenar os dados pessoais em nuvem. Os dados pessoais básicos são: sexo, data de nascimento, massa corporal e altura, os quais são utilizados nos cálculos de gasto de calorias.

Após fornecer os dados básicos, o usuário é redirecionado para a tela principal na qual é exibida a data e hora atual, um mapa interativo exibindo a posição atual, e botões para iniciar uma atividade em dupla ou individual. Pressionando o para iniciar a atividades individuais, é iniciada uma contagem regressiva para o início do exercício. Durante o percurso, são exibidas as informações de velocidade, distância, tempo, ritmo médio e calorias gastas até o momento.

Na condição de início de atividade em dupla, o usuário é redirecionado para a tela de busca de usuários, na qual são exibidos outros usuários conectados, igualmente buscando por um companheiro para praticar exercícios. Selecionando um dos usuários é enviada uma mensagem de solicitação, para iniciar uma atividade, para o mesmo. Caso o convite seja aceito, é inicializada a contagem regressiva para ambos os jogadores. Sempre que conectado a um parceiro, em cada momento que são recebidas as informações do GPS, é enviado um resumo

da atividade ao dispositivo conectado. No momento que um dos participantes pressiona o botão de pausa, ambos têm suas atividades pausadas, e assim que um deles pressiona para retomar, é retomada para ambos. Se um decide finalizar, é finalizada apenas para o mesmo, não interferindo seu parceiro caso desejar continuar a se movimentar.

Ao final de cada trajeto realizado, o usuário é recompensado com pontos de experiência (XP), calculados a partir do tempo em movimento e gasto de energia. No caso de caminho em dupla, a pontuação é duplicada, por estar sendo incentivada a participação de outra pessoa na realização de uma atividade física. Foram criados 56 níveis de progresso, cada um com pontuações máximas distintas em ordem crescente, isto é, a cada nível atingido, o usuário precisará fazer mais pontos do que foi necessário para o nível anterior. Os intervalos de pontuações iniciam em trezentos, e seguem gradualmente até 4100 pontos. O último nível possui a pontuação final de cem mil pontos. Os nomes dos níveis de progresso são dados a partir de cidades ou capitais de países.

Na tela principal, há um menu inferior com opções para as outras telas. Uma delas é a de histórico, onde são exibidas as atividades realizadas anteriormente, apresentadas em *cards*, cada um com informações básicas. Pressionando em um *card*, é aberta a tela na qual é mostrado detalhadamente cada exercício, bem como o mapa com seu percurso desenhado.

O menu possui ainda uma terceira opção, que ao ser selecionada é apresentado na parte superior da tela o número do nível que o usuário está atualmente, circundado por uma barra de progresso circular, e à sua direita o nome do nível, e o score atual e mínimo para o próximo nível. Abaixo da pontuação atual, é apresentado a data e horário da conquista do nível atual. Abaixo desta primeira seção, há os níveis pelos quais o usuário já passou, em forma de lista, com cada item exibindo seu número, nome, pontuação máxima, e momento em que foi atingido.

5.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a listagem dos requisitos funcionais (RF) identificados para o aplicativo.

Quadro 2 – Requisitos funcionais

Identificação	Nome	Descrição
RF01	Cadastrar usuário.	Os dados pessoais do usuário são utilizados para calcular o gasto calórico durante a caminhada ou a corrida e para imersão no aplicativo.
RF02	Iniciar atividade física.	Momento em que a captura e processamento de dados é iniciado.

RF03	Finalizar atividade física.	Momento em que a captura e processamento de dados é encerrado.
RF04	Obter informações de localização.	A posição geográfica do dispositivo é obtida para realizar cálculos de distância, velocidade, ritmo médio, entre outros.
RF05	Calcular gasto calórico.	Baseado na velocidade da atividade e peso do usuário será estimada a quantidade de calorias gastas até o momento.
RF06	Processar os dados durante a atividade.	Os cálculos para monitorar os dados e informações da atividade são realizados enquanto o usuário faz a atividade física, para que sejam atualizados em tempo real na tela.
RF07	Sincronizar com outro dispositivo Android.	A sincronização é responsável por trocar informações com a pessoa enquanto a atividade física é realizada.
RF08	Oferecer recompensas por percurso finalizado.	O aplicativo dá ao usuário, a cada final de trajeto, uma recompensa em XP, que são somados para passar de nível.
RF09	Acumular bônus como níveis de progresso.	Em complemento ao RF08, os bônus que o usuário recebe no final de cada trajeto são acumulados gradativamente e fazem parte de níveis de progresso. Quanto mais XP o usuário tem, maior seu nível, e a cada nível alcançado, mais XP devem ser feitos para chegar ao próximo nível.
RF11	Apresentar histórico dos trajetos realizadas previamente.	O usuário pode visualizar detalhadamente os dados das atividades que praticou, sendo apresentados os dados que foram coletados e processados durante o exercício.

Fonte: Autoria própria.

A listagem do Quadro 3 apresenta os requisitos não funcionais (RNF) identificados para o aplicativo como um todo, também denominados de requisitos suplementares.

Quadro 3 – Requisitos não funcionais

Identificação	Nome	Descrição
RNF01	Compartilhar e receber dados da atividade pela internet	Na opção de percurso em dupla os dados de cada usuário são constantemente compartilhados entre os dispositivos pareados.
RNF02	Internacionalização	O aplicativo tem disponibilidade em língua portuguesa e inglesa.
RNF03	Tema com contraste entre cores	Os elementos visuais têm cores que são bem visualizadas quando o aplicativo é utilizado ao ar livre em condições de sol.
RNF04	Tempo para iniciar a atividade	Após o cronômetro ser acionado, o usuário tem, inicialmente, 5 segundos para iniciar atividade, porém podem ser adicionados mais cinco segundos, com tempo máximo de sessenta segundos após a última vez de quando pressionado o botão.

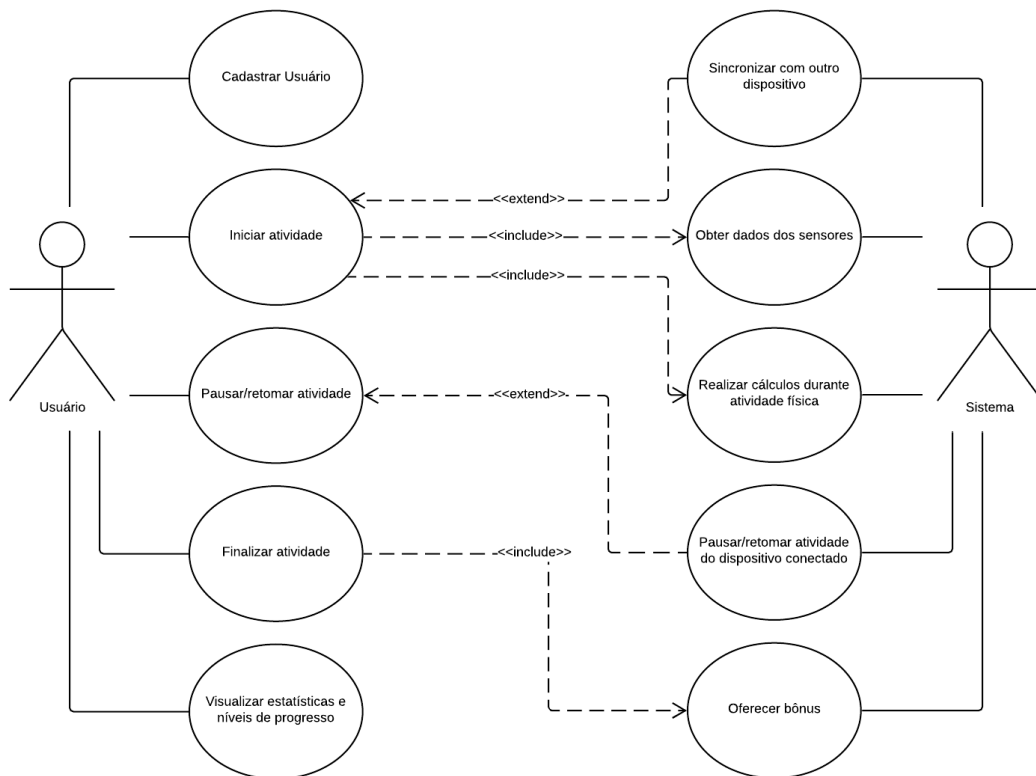
Fonte: Autoria própria.

O diagrama de casos de uso apresentado na Figura 2 contém as funcionalidades essenciais do aplicativo realizadas pelos seus atores que são: sistema e usuário.

O ator sistema é responsável por obter dados dos sensores, realizar cálculos e sincronizar com outro aparelho durante a corrida ou caminhada e ao final de cada exercício, conceder bônus de pontuação. Sincronizar com outro dispositivo é a parte na qual o aplicativo inicia a troca de informações com outro dispositivo, conectado via internet.

O ator usuário é quem realiza a atividade física, sendo responsável por informar seus dados e medidas na primeira utilização do aplicativo. Iniciar a atividade é a ação tomada logo antes da atividade acontecer e finalizar ocorre ao seu término. Após o término da atividade, o usuário tem a possibilidade de visualizar as estatísticas do itinerário percorrido e os níveis de progresso conquistados.

Figura 2 – Diagrama de casos de uso



Fonte: Autoria própria.

A seguir estão as expansões dos casos de uso. No Quadro 4 está a descrição do caso de uso cadastrar usuário.

Quadro 4 – Caso de uso cadastrar usuário

Caso de uso:

Cadastrar usuário.

Descrição:

Na primeira utilização do aplicativo, o usuário deve fornecer seus dados pessoais e medidas biométricas, sendo obrigatórios: sexo, data de nascimento, peso e altura. O aplicativo armazena esses dados, e são posteriormente utilizados nos cálculos realizados a partir dos dados obtidos das caminhadas e corridas realizadas.

Evento Iniciador:

Inicialização primária do aplicativo.

Atores:

Usuário. Sequência de Eventos: <ol style="list-style-type: none"> 1. Usuário abre o aplicativo. 2. Sistema apresenta a tela para preencher os campos com as informações necessárias. 3. Usuário preenche os campos e confirma para salvar. 4. Sistema apresenta a tela principal. Pós-Condição: Dados do usuário registrados nas preferências compartilhadas do app, e em nuvem.	
Nome do fluxo alternativo (extensão)	Descrição
Linha 3: Dados não são válidos.	3.1. No momento de salvar, o sistema faz a verificação e constata que há dados inválidos ou não informados. É apresentada mensagem.
	6.2 Retorna para a tela de preenchimento de informações – evento 2.

Fonte: Autoria própria.

O caso de uso iniciar atividade é apresentado no Quadro 5.

Quadro 5 – Caso de uso iniciar atividade

Caso de uso: Iniciar atividade Descrição: Após iniciar a atividade, o aplicativo obtém dados por meio de GPS e calcula informações como velocidade e ritmo. Evento Iniciador: O usuário deve iniciar a atividade física no aplicativo. Atores: Usuário e sistema. Pré-condição: O usuário deve preencher seus dados e permitir o uso dos sensores. Sequência de Eventos: <ol style="list-style-type: none"> 1. Usuário dá início à atividade. 2. Sistema verifica se os sensores estão disponíveis/ativos. 3. Sistema inicia um <i>listener</i> para obtenção dos dados de GPS. 4. Sistema processa os dados obtidos, a fim de obter informações úteis. 5. Sistema apresenta na tela, em tempo real, os dados recebidos, processados e tratados. Pós-Condição: Usuário finaliza a atividade. Sistema grava em banco de dados.	
Nome do fluxo alternativo (extensão)	Descrição
Linha 2: Usuário seleciona para iniciar atividade em dupla.	2.1 Quando é selecionado para atividade em dupla, o Sistema apresentará uma tela para buscar outros dispositivos, e conectar com outro dispositivo.
	2.2 Segue para o evento 3.

Fonte: Autoria própria.

O caso de uso pausar e retomar atividade é apresentado no Quadro 6.

Quadro 6 – Caso de uso pausar e retomar atividade

Caso de uso: Iniciar atividade Descrição:

<p>Enquanto o exercício está em andamento, o usuário tem a possibilidade de pausar a atividade, e, caso deseje continuar, retomá-la.</p> <p>Evento Iniciador: Usuário deve pressionar de pausa ou retomar atividade.</p> <p>Atores: Usuário e sistema.</p> <p>Pré-condição: A atividade já deve ter sido iniciada, e ainda não finalizada.</p> <p>Sequência de Eventos: 1. Usuário pausa a atividade. 2. Usuário retoma a atividade.</p>	
Nome do fluxo alternativo (extensão)	Descrição
Linha 1: Pausar atividade em dupla.	1.1 Quando a atividade é em dupla, quando o usuário faz a suspensão em seu próprio dispositivo, sistema faz com que seu parceiro também tenha a atividade suspensa.
Linha 2: Retomar atividade em dupla.	1.1 Quando a atividade é em dupla, quando o usuário faz a retomada em seu próprio dispositivo, o sistema faz com que seu parceiro também tenha a atividade continuada.

Fonte: Autoria própria.

O caso de uso finalizar atividade é apresentado no Quadro 7.

Quadro 7 – Caso de uso finalizar atividade

<p>Caso de uso: Finalizar atividade.</p> <p>Descrição: Quando o usuário finaliza uma atividade, o sistema dá uma recompensa em pontos.</p> <p>Evento Iniciador: Usuário deve pressionar botão para finalizar a atividade.</p> <p>Atores: Usuário e sistema</p> <p>Pré-condição: Usuário deve ter iniciado o exercício.</p> <p>Sequência de Eventos: 1. Usuário finaliza atividade. 2. Sistema calcula quantidade de pontos, conforme o desempenho do usuário durante a atividade e soma aos pontos obtidos anteriormente, estabelecendo o progresso do usuário no jogo.</p> <p>Pós-Condição: Informações são registradas no SQLite. Tela de detalhes da atividade é apresentada.</p>

Fonte: Autoria própria.

A descrição do caso de uso visualizar estatísticas é apresentada no Quadro 8.

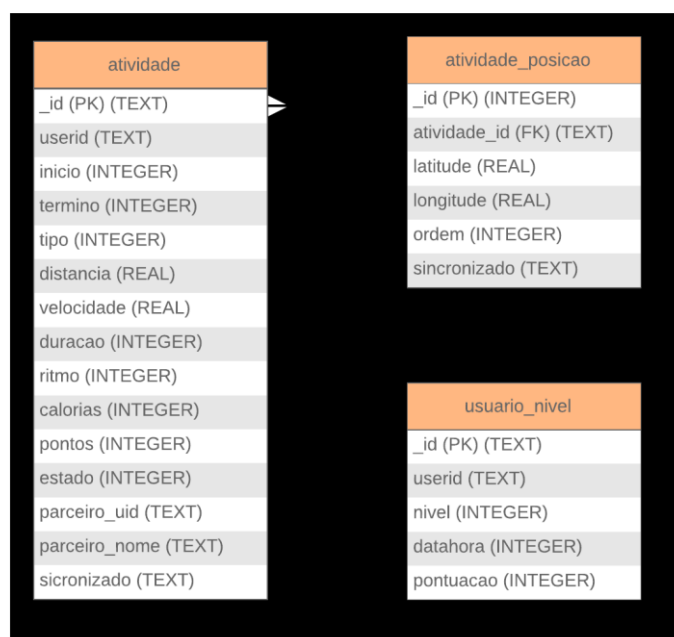
Quadro 8 – Caso de uso visualizar estatísticas

Caso de uso:	Visualizar estatísticas.
Descrição:	O usuário pode visualizar o histórico de suas caminhadas/corridas e suas respectivas informações, detalhadamente.
Evento Iniciador:	Abrir tela de histórico de atividades.
Atores:	Usuário.
Pré-condição:	Usuário deve ter completado pelo menos uma caminhada/corrida.
Sequência de Eventos:	<ol style="list-style-type: none"> 1. Usuário visualiza <i>cards</i> de cada atividade praticada na tela de histórico. 2. Usuário pressiona sobre um <i>card</i> para ver as informações detalhadas.
Pós-Condição:	Tela com as informações da atividade escolhida pelo usuário é apresentada.

Fonte: Autoria própria.

Na Figura 3 está o diagrama de entidades e relacionamentos que representam o banco de dados da aplicação. Para as chaves primárias das tabelas *atividade* e *usuario_nivel* foi decidido utilizar texto, no formato UUID, para serem criados identificadores, de fato, únicos. Já na tabela *atividade_posicao*, foram utilizados identificadores de tipo inteiro, gerados sequencialmente pelo *Room*, para a chave primária, pois durante uma única atividade podem ser registradas centenas, ou até mesmo milhares, de posições, e considerando o custo alto, em relação à performance em um dispositivo móvel, é mais viável utilizar inteiros seriados do que identificadores únicos universais.

Figura 3 – Diagrama de entidades e relacionamentos do banco de dados



Fonte: Autoria própria.

Uma pessoa terá um nível de progresso e uma pontuação dentro daquele nível, bem como o momento em que este foi atingido. O Quadro 9 apresenta os campos da tabela que registra o progresso do usuário.

Quadro 9 – Campos da tabela usuario_nivel

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
_id	Texto	Não	Sim	Não	
userid	Texto	Não	Não	Não	Referência do id do usuário salvo nas <i>Shared Preferences</i>
nivel	Numérico	Não	Não	Não	Referência do enum NivelProgresso
datahora	Numérico	Não	Não	Não	
pontuacao	Numérico	Não	Não	Não	

Fonte: Autoria própria.

Os campos da tabela atividade estão apresentados no Quadro 10. Uma pessoa irá praticar várias atividades. Essa tabela possui a referência do id do usuário salvo nas *Shared Preferences*.

Quadro 10 – Campos da tabela atividade

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
_id	Texto	Não	Sim	Não	
userid	Texto	Não	Não	Não	Referência do id do usuário salvo nas <i>Shared Preferences</i>
inicio	Numérico	Não	Não	Não	
termino	Numérico	Não	Não	Não	
tipo	Texto	Não	Não	Não	Referência do enum AtividadeTipo
distancia	Numérico	Não	Não	Não	
velocidade	Numérico	Não	Não	Não	
duracao	Numérico	Não	Não	Não	
ritmo	Numérico	Não	Não	Não	
calorias	Numérico	Não	Não	Não	
pontos	Numérico	Não	Não	Não	
estado	Numérico	Não	Não	Não	Referência do enum AtividadeEstado
parceiro_uid	Texto	Sim	Não	Não	Referência do id do usuário do dispositivo conectado
parceiro_nome	Texto	Sim	Não	Não	Nome do usuário do dispositivo

					conectado
sincronizado	Texto	Não	Não	Não	

Fonte: Autoria própria.

No Quadro 11 estão apresentados os campos da tabela atividade_posicao. Nesta tabela são registrados os dados de latitude e longitude, de cada instante em que é obtida a coordenada geográfica pelo GPS, assim como a ordem de cada posição, a fim de não haver inconsistências ao calcular distâncias e projetar o percurso no mapa. Cada atividade possui vários registros de posições relacionadas a ela.

Quadro 11 – Campos da tabela atividade_posicao

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
_id	Numérico	Não	Sim	Não	
atividade_id	Texto	Não	Não	Sim	Referência da tabela atividade
latitude	Numérico	Não	Não	Não	
longitude	Numérico	Não	Não	Não	
ordem	Numérico	Não	Não	Não	
sincronizado	Texto	Não	Não	Não	

Fonte: Autoria própria.

5.3 INTERFACE GRÁFICA DO SISTEMA

Nesta seção será apresentada a interface gráfica do sistema. Iniciando com os protótipos Lo-Fi, de baixa fidelidade, os quais não exibem necessariamente o mesmo design da versão final.

5.3.1 Protótipos Lo-Fi

As figuras 4a, 4b e 4c são protótipos feitos com a ferramenta Balsamiq Mockups, e representam as telas de cadastro de dados, atividade e níveis de progresso, respectivamente.

Figura 4 – Protótipos Lo-Fi. A) Cadastro de dados; b) Atividade; c) Níveis de Progresso

(a) (b) (c)

(a) Cadastro de dados: Tela de boas-vindas para Jean. Instrução: "Para realizar cálculos mais precisos de calorias queimadas, informe um pouco sobre você." Campos de formulário: Sexo (Masculino), Data de nascimento (//), Altura (1,70m), Peso (80kg). Botão "Confirmar".

(b) Atividade: Tela de atividade com distância de 8888.88 km. Métricas: 00:00:00 (Duração), 000' 00" (Pace (min/km)), 8888888 (Calorias). Botão de pausa.

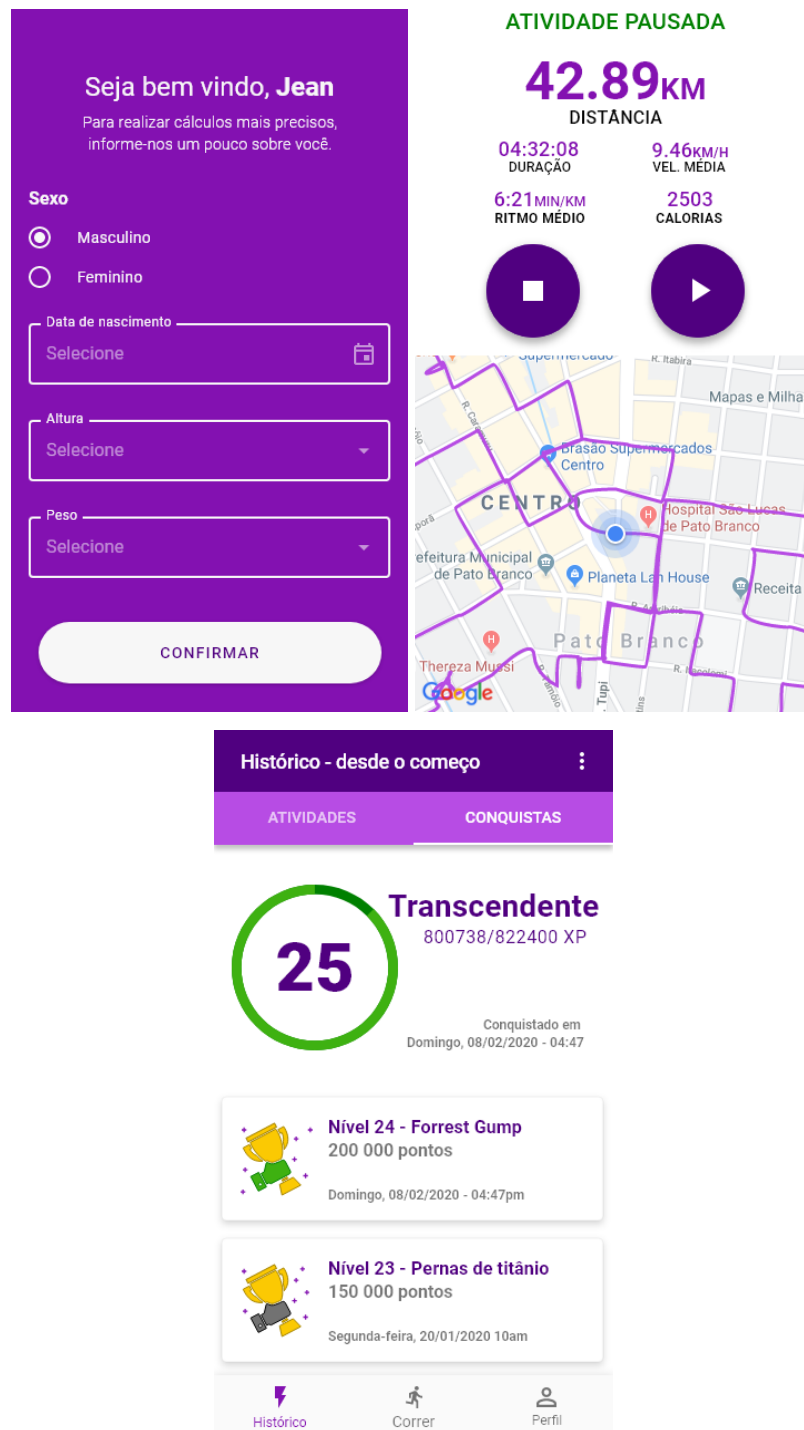
(c) Níveis de Progresso: Tela de atividades com nível 25 "Super-humano" (8 888 888 pontos para Transcendente). Atividades listadas: De bruços (1km, 100 pontos), Engatinhando (5km, 500 pontos), Primeiros passos (10km). Menu inferior: Histórico, Atividade, Perfil.

Fonte: Autoria própria.

5.3.2 Protótipos Hi-Fi

As Figuras 5a, 5b e 5c são protótipos feitos com a ferramenta Adobe XD, representando as telas de cadastro de dados, atividade e níveis de progresso, respectivamente, bem como as telas identificadas no tópico 5.3.1, porém, desta vez com cores já definidas e elementos gráficos mais detalhados, assemelhando-se muito às telas que foram feitas no aplicativo.

Figura 5 – Protótipos Hi-Fi. a) Cadastro de dados; b) Atividade; c) Níveis de Progresso



Fonte: Aatoria própria.

5.3.3 Interface gráfica no aplicativo

Nesta subseção serão apresentadas as capturas de tela durante o uso do aplicativo, exibindo as telas que foram mostradas anteriormente, para comparação aos protótipos, e algumas das outras desenvolvidas, a fim de apresentar o sistema.

Na Figura 6a é exibida a tela de cadastro de dados, onde no topo é mostrada a mensagem de boas-vindas, seguida do nome do usuário recuperado da sua conta Google ou Facebook, utilizada na autenticação. O primeiro componente utilizado para coletar dados do usuário é um *RadioButton*, no qual é informado o sexo. Logo após, são exibidos três campos de texto, *EditText*, que quando pressionados individualmente, são abertos *dialogs* para informar data de nascimento, altura e peso, respectivamente, mostrados nas figuras 6a, 6b, 6c e 6d. Na região inferior deste leiaute há o botão de confirmação, que, ao ser pressionado registra os dados do usuário localmente, nas *SharedPreferences* e em nuvem, no *Firebase Realtime Database*.

Figura 6 – Tela de Cadastro de Dados. a) Formulário Completo; b) Seleção de Data de Nascimento; c) Seleção de Altura; Seleção de Peso;
(a) (b)

(a) Formulário Completo:

Seja bem-vindo(a), Jacinta

Para realizar cálculos mais precisos, informe-nos um pouco sobre você.

Sexo

Feminino

Masculino

Data de nascimento: 07 setembro 1967

Altura: 1.59m

Peso: 53.3kg

CONFIRMAR

(b) Seleção de Data de Nascimento:

Seja bem-vindo(a), Jacinta

Para realizar cálculos mais precisos, informe-nos um pouco sobre você.

1967

Qui, 7 de set

setembro de 1967

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

CANCELAR OK

CONFIRMAR

(c) Seleção de Altura:

Seja bem-vindo(a), Jacinta

Para realizar cálculos mais precisos, informe-nos um pouco sobre você.

Sexo

Altura

58

1 . 59 m

2 60

CANCELAR OK

Peso

CONFIRMAR

(d) Seleção de Peso:

Seja bem-vindo(a), Jacinta

Para realizar cálculos mais precisos, informe-nos um pouco sobre você.

Sexo

Peso

52 2

53 . 3 kg

54 4

CANCELAR OK

Peso

CONFIRMAR

Fonte: Autoria própria.

Após o cadastro ser efetuado o usuário é redirecionado para a tela principal da aplicação, que conta com um menu inferior, o qual possui os itens de Histórico, Progresso e Correr, respectivamente. Por padrão, é selecionado a opção de correr, que em seu leiaute possui dois componentes do tipo *TextClock* para apresentar data e horário atuais, nesta ordem. Na sequência há um componente de mapa interativo, onde mostra a posição atual do dispositivo. No canto inferior direito do mapa são dispostos dois botões redondos de cor roxo claro com ícones brancos, um logo acima do outro, justapostos ao mapa. O botão superior possui ícone de conexão em par, simbolizando a atividade em dupla, e ao ser pressionado o usuário é levado à tela de busca de outros usuários. O botão inferior, com ícone de perfil, representa a atividade só, e quando pressionado dá início à contagem regressiva para iniciar a atividade. Esta tela é representada na Figura 7.

Figura 7 – Tela Principal



Fonte: Autoria própria.

Os usuários disponíveis para realização de atividade em dupla são exibidos no formato de lista, com seu nome e status *online* ao lado. Ao ser selecionado por um usuário é exibido um botão que possui como texto o número de solicitações recebidas. Ao pressioná-lo é aberta uma caixa de diálogo mostrando os usuários que enviaram os convites. Confirmando o convite do

parceiro, é iniciada a contagem regressiva para ambos. A busca de usuários pode ser observada na Figura 8.

Figura 8 – Busca de Usuários Para Atividade em Dupla

Fonte: Autoria própria.

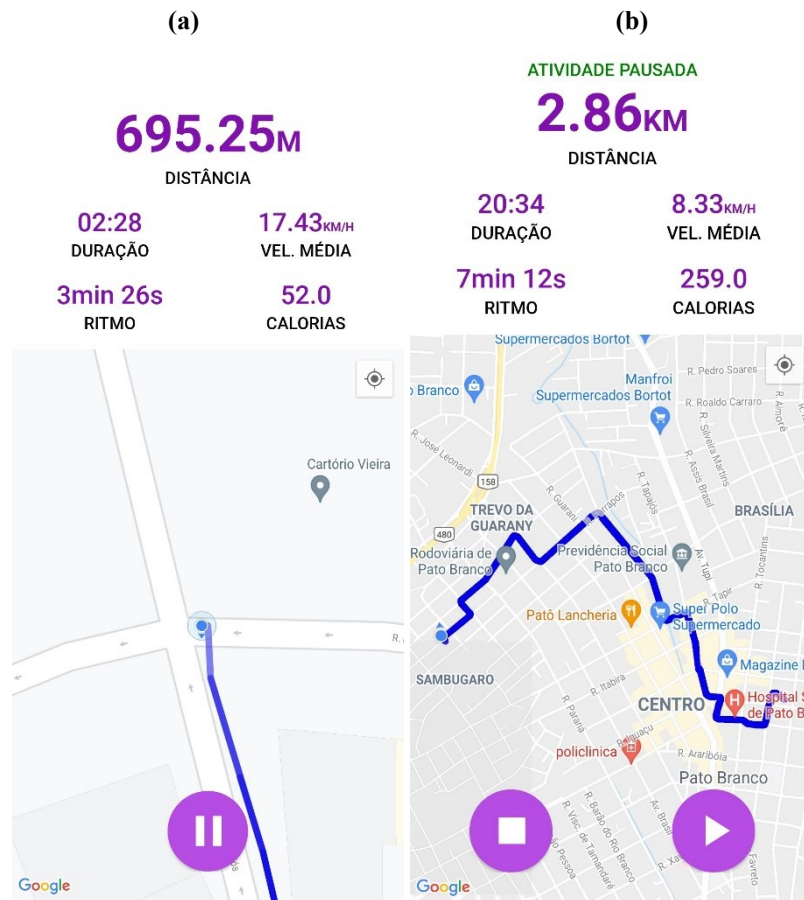
A contagem regressiva é de, originalmente, cinco segundos, podendo ser cancelada para não ser realizada a atividade, adiada de cinco em cinco segundos, até um minuto, ou interrompida, para a atividade ser iniciada mais cedo. Estas três ações são efetuadas por meio de botões em tela. A Figura 9 identifica o leiaute descrito para a contagem regressiva.

Figura 9 – Contagem Regressiva para Atividade

Fonte: Autoria própria.

Após a contagem regressiva ser completada, é iniciada a atividade, essa tela pode ser observada na Figura 10a. Na parte superior da tela é mostrada distância percorrida, logo abaixo a duração da atividade e velocidade média, uma ao lado da outra, e abaixo destas, informações de ritmo e gasto calórico, dispostas identicamente. Estes dados estão em cor roxo escuro, logo acima de seus respectivos rótulos, estes em fonte preta. Abaixo, há um mapa interativo no qual é mostrado a posição atual do dispositivo. Na parte inferior do mapa, sobrepondo-o, há um botão circular para pausar a atividade, que, sendo pressionado, o mesmo é ocultado, e são revelados os botões para parar e retomar, e com que o mapa afaste a visualização da posição atual para que seja possível visualizar o trajeto completo. Também é mostrado intermitentemente um texto de cor verde no topo da tela, acima de onde é mostrada a distância, indicando que a atividade está pausada. As figuras 10a e 10b representam a tela de atividade em seus estados em andamento e pausado.

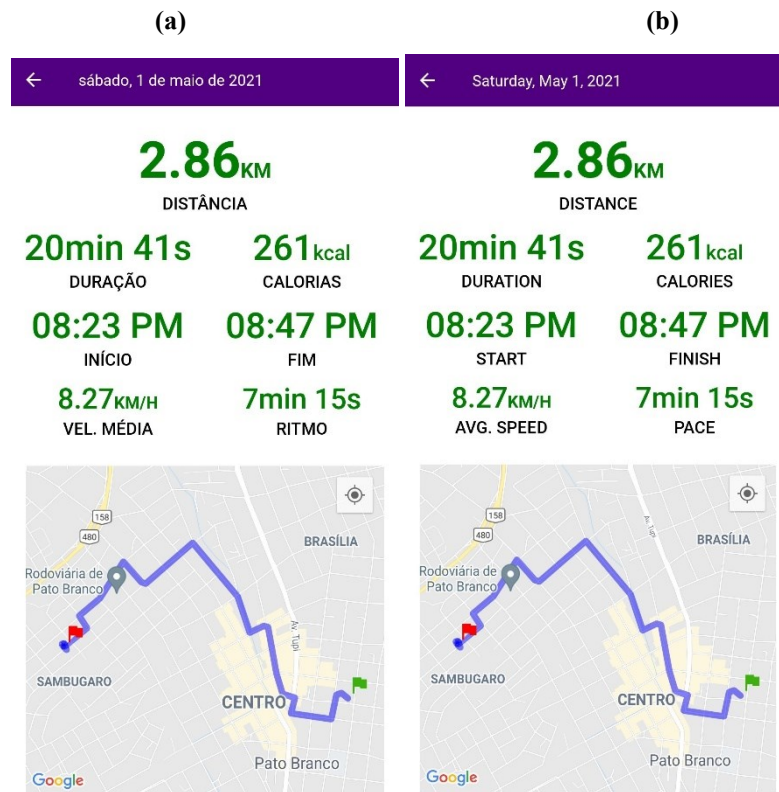
Figura 10 – Tela de Atividade. a) Estado em Andamento; b) Estado Pausado



Fonte: Autoria própria.

Posteriormente à conclusão da atividade o utilizador é encaminhado para a tela de detalhamento da atividade, na qual, na barra superior é mostrada a data em que o percurso foi realizado, e na parte principal da tela são dispostas as informações que eram mostradas enquanto o trajeto estava sendo realizado, adicionadas dos horários em que foi iniciado e terminado. Os detalhes contam, também, com o mapa interativo da rota tendo o caminho traçado em roxo, sendo posicionados um ícone de bandeira verde no ponto inicial, e um de bandeira vermelha ao ponto final, conforme ilustrado na Figura 11. A Figura 11a representa a tela do aplicativo enquanto é utilizado o idioma original do dispositivo, e a Figura 11b exibe a mesma tela, porém quando o *smartphone* teve seu idioma alterado para inglês. A tradução das telas foi possível por meio dos recursos de internacionalização da plataforma de desenvolvimento Android.

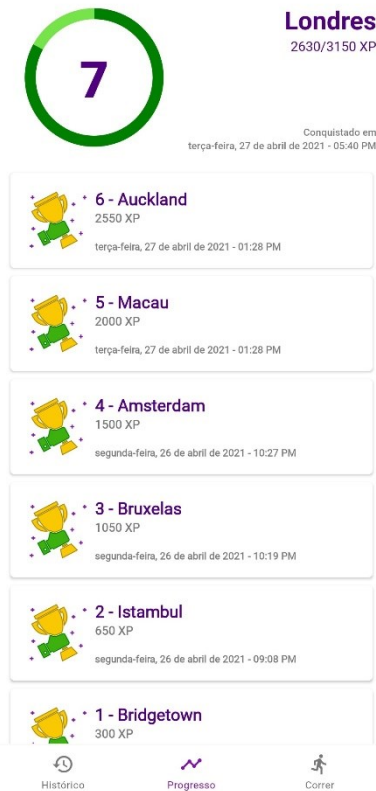
Figura 11 – Detalhes da Atividade Realizada. a) Idioma em português; b) Idioma em inglês



Fonte: Autoria própria.

Os níveis de progresso do usuário são mostrados assim que o item central no menu inferior da tela principal da aplicação é pressionado. Na parte superior deste leiaute há as informações do nível atual, e logo abaixo, uma *RecyclerView* com cada nível já conquistado, conforme exibido na Figura 12.

Figura 12 – Níveis de Progresso



Fonte: Autoria própria.

O histórico de corridas é mostrado a partir da interação com o primeiro item do menu inferior. Nesta tela é trazido um resumo de todas as atividades praticadas, com informações de distância e tempo totais, quantidade de percursos realizados na companhia de um parceiro, quantidade total, e datas do primeiro e último trajeto realizado. Abaixo, é apresentada uma lista de *cards* com breves informações de cada atividade, como data, quantidade de pontos conquistados, distância e duração, que ao pressionar em um *card*, é aberta a tela de detalhes, retratada anteriormente na Figura 11. A Figura 13 representa a tela de histórico.

Figura 13 – Histórico de Atividades



Fonte: Autoria própria.

5.4 HIERARQUIA DE PASTAS DO PROJETO

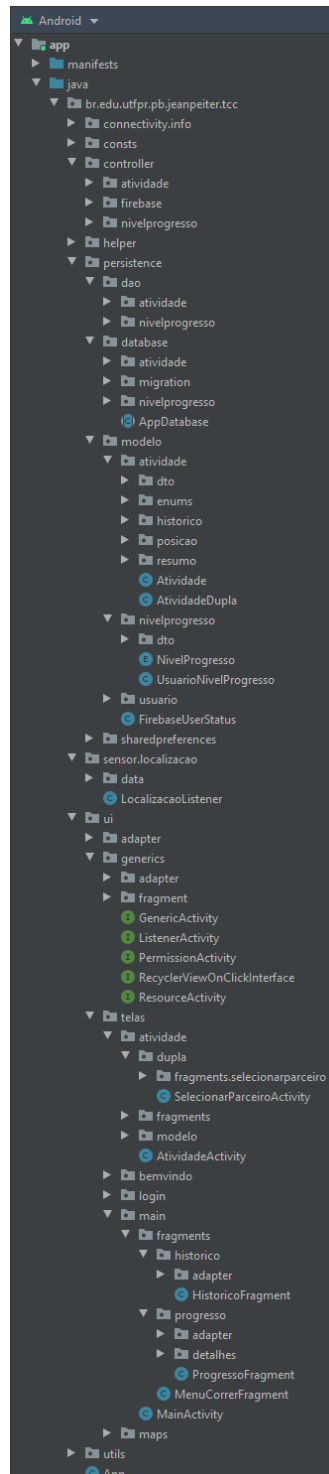
O projeto foi dividido em oito pacotes principais e suas subdivisões. Os principais, em ordem alfabética, são:

- *connectivity*: classes para verificações sobre conexões de GPS e rede;
- *consts*: classes com valores imutáveis, utilizadas na lógica do aplicativo;
- *controller e helper*: classes e pacotes de classes que realizam as regras de negócio do sistema;
- *persistence*: pacote que concentra classes para as operações que envolvem escritas e lidas ao banco de dados e preferências compartilhadas;
- *sensor*: detém classes que realizam obtenção e dos dados de GPS, e transmissão dos mesmos para outras classes;
- *ui*: dispõe de pacotes de classes de *Activities* e classes que fazem com que os dados processados sejam apresentados em tela. Nesta pasta também estão localizadas classes abstratas e interfaces que são estendidas e implementadas por outras, a fim de facilitar o desenvolvimento;

- *utils*: concentra classes utilitárias que manipulam dados e/ou executam ações, porém sem alterar nem manter estado de variáveis.

Os pacotes do projeto como estão na IDE, são exibidos na Figura 14.

Figura 14 – Hierarquia de Pacotes



Fonte: Autoria própria.

5.5 IMPLEMENTAÇÃO DO SISTEMA

A IDE utilizada no desenvolvimento do sistema foi o Android Studio, com a API de nível 29 da SDK Android, restringindo a instalação e funcionamento do aplicativo para dispositivos que possuam o Android 10 ou posterior como versão do SO. A API Java utilizada foi a versão de número 11, possibilitando utilizar expressões *Lambda* e referências de métodos, fazendo com que o código seja menos verboso, portanto, de melhor leitura.

Foram utilizadas bibliotecas e serviços de terceiros para facilitar o desenvolvimento, desde açúcar sintático – com o *Lombok*, para geração automática de códigos – quanto a respeito de arquitetura, com o *EventBus*, por exemplo, para realizar comunicação entre *Activities* e *Fragments*.

O serviço de autenticação, *Authentication*, do Firebase foi utilizado para registrar uma referência do usuário, em nuvem, no aplicativo, assim sendo possível que este possa abrir o *app* em outro dispositivo, ou no mesmo, caso haja desinstalação do aplicativo, sem que seus dados pessoais sejam extraviados. Ainda da mesma plataforma, o serviço de banco de dados em tempo real, o *Realtime Database*, foi fundamental para que fosse feito o cadastro de dados do usuário, para acontecer a busca de usuários para atividade em dupla, e posteriormente, para comunicação entre ambos os praticantes durante o percurso.

Uma das bibliotecas consumidas de maior relevância, é a *UnitOf* (disponível em: <https://github.com/Digidemic/UnitOf>), responsável, neste projeto, por fazer conversões de unidades de tempo, distância, velocidade, e volume, em virtude de que os dados coletados pelo sistema, em muitas vezes, não são armazenados no mesmo formato e/ou unidade de medida em que são apresentados ao usuário. Esta ferramenta foi a única em que foi necessário ser adicionada por meio de arquivo *.jar* – o compilado de um projeto Java – pois não há uma versão disponível para Gradle, a ferramenta de compilação e gerenciamento de pacotes do Android.

Perante o exposto, é possível detalhar trechos do desenvolvimento do sistema. Foram selecionadas rotinas relevantes da aplicação, tais como o processo comunicação entre usuários antes e durante um percurso, respectivamente, obtenção de dados do GPS, interação com o mapa da atividade, cálculo de gasto calórico, de pontuação por atividade e leitura e escrita no banco de dados do dispositivo.

5.5.1 Monitoramento de solicitações de atividade em dupla

Este tópico refere-se ao monitoramento de uma coleção específica do *Realtime Database* para verificar convites para atividade em dupla. Trecho de código apresentado na Listagem 1.

Listagem 1 – Monitoramento de solicitações para Atividade

```
private void monitorarPendentes() {
    solicitacoes = new HashMap<>();
    FirebaseAtividadeDuplaController.getInstance().monitorarPendentes(new
    ChildEventListener() {

        private void usuarioConectado(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {
            if (dataSnapshot.getValue(String.class)
                .equals(FirebaseAtividadeDuplaController.VALOR_CONFIRMACAO))
            {
                iniciarAtividadeDupla(dataSnapshot.getKey());
            } else {
                solicitacoes.put(dataSnapshot.getKey(),
dataSnapshot.getValue(String.class));
                atualizarUi();
            }
        }

        @Override
        public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
            usuarioConectado(dataSnapshot, s);
        }

        @Override
        public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
            usuarioConectado(dataSnapshot, s);
        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {
            solicitacoes.remove(dataSnapshot.getKey());
            atualizarUi();
        }

        @Override
        public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
    });
};
```

```
}
```

Fonte: Autoria própria.

Assim que essa tela é aberta, a rota de solicitações, definida como “info/connected/{userId}/pendentes”, em que “userId” entre chaves é substituído, em tempo de execução, pelo identificador do usuário, é monitorada para descobrir usuários que enviaram o pedido de corrida em dupla. O código acima identifica esta rotina de monitoramento. Na classe em que é iniciada a observação por alterações na rota de convites, chamada *SelecionarParceiroActivity*, há uma variável do tipo *HashMap* chamada *solicitacoes*, com chaves e valores de tipo *String*, sendo referentes aos identificadores dos usuários que enviam solicitações, e seu nome, para ser exibido em tela.

No início do método *monitorarPendentes*, a variável *solicitacoes* é reiniciada, para não haver solicitações pendentes. Logo em seguida, é chamado o método de mesma assinatura, desta vez da classe *FirestoreAtividadeDuplaController*, em que é passado por parâmetro um listener que monitora as solicitações na rota de pendências. Este listener conta com três métodos próprios, de um total de cinco, que foram utilizados para atualizar a interface gráfica com a quantidade de pendências, sendo estes: *onChildAdded*, *onChildChanged* e *onChildRemoved*. Os dois primeiros métodos citados utilizam do método *novaSolicitacao*, no qual, caso o valor recebido seja igual ao valor esperado para iniciar uma atividade, esta é iniciada. Caso contrário, é adicionado à variável *solicitacoes* o conjunto de código e nome do usuário que realizou o convite, e atualizada a UI com os valores novos. Já no terceiro método, *onChildRemoved*, é removida a referência do usuário que havia solicitado uma atividade, e, como nos métodos anteriores, renovada a UI.

5.5.2 Comunicação entre usuários durante um percurso

Nas listagens 2 e 3 encontram-se partes importantes do código-fonte criado para fazer com que dois usuários troquem informações relativas às suas respectivas atividades, em tempo real, enquanto estão realizando as atividades.

Listagem 2 – Controle de Publicação e Inscrição de Atividade em Dupla

```
private String pathAtividadeDupla() {
    if (this.pathAtividadeDupla == null) {
        List<String> uids = Arrays.asList(this.userId, this.parceiroUid);
        uids.sort(Comparator.naturalOrder());
        this.pathAtividadeDupla = PATH.concat("/")
            .concat(uids.get(0))
            .concat("_")
            .concat(uids.get(1));
    }
    return this.pathAtividadeDupla;
}

public void atualizar(Atividade atividade) {
    String pathAtividadeUser =
this.pathAtividadeDupla().concat("/").concat(this.userId);
    FirebaseController.setValue(pathAtividadeUser, atividade.toDto());
}

public void monitorarParceiro(ValueEventListener
listenerMonitorarParceiro) {
    setListenerMonitorarParceiro(listenerMonitorarParceiro);
    String pathAtividadeParceiro =
this.pathAtividadeDupla().concat("/").concat(this.parceiroUid);
    FirebaseController.getDatabase(pathAtividadeParceiro)
        .addValueEventListener(getListenerMonitorarParceiro());
}
}
```

Fonte: Autoria própria.

A Listagem 2 traz alguns métodos da classe `FirestoreAtividadeDuplaController`. O método `pathAtividadeDupla` é responsável por criar o caminho da coleção do *Realtime Database* onde constantemente são salvas as atividades de ambos os usuários, enquanto estão sendo realizadas. Este método verifica se já exista um caminho criado na classe em questão. Caso não haja, é criado um uma lista de `Strings` contendo os códigos do usuário do dispositivo atual e de seu parceiro conectado, e ordenada alfabeticamente. A partir da constante `PATH`, de valor “atividade_dupla”, concatenada por uma barra, o primeiro item da lista, um traço baixo e, por fim, o segundo item da lista, é criada a rota para a atividade entre os dois usuários, de forma que ambos se conectem por meio dela, e finalmente retornado este caminho.

No método `atualizar` é criado o caminho da atividade do usuário atual por meio do retorno do método `pathAtividadeDupla` seguido de uma barra e o identificador do usuário de que se trata, e utilizado este caminho para salvar o objeto do tipo `Atividade`, recebido por parâmetro, em sua forma de `DTO` (*Data Transfer Object*, no inglês), para que seja economizado

espaço de armazenamento em nuvem. Este método é chamado sempre que é obtida uma nova localização pelo GPS, ou quando o usuário pressiona os botões de pausa, retomada e finalização da atividade.

Por fim, o método `monitorarParceiro` recebe um *listener* como seu único argumento, que é atribuído à variável `listenerMonitorarParceiro` da mesma classe via *setter*. É criado o caminho para monitorar as publicações de atividades do parceiro, também por meio do retorno do método `pathAtividadeDupla` seguido de uma barra e o identificador do usuário conectado. Por último, é adicionado o *listener* para monitorar o parceiro, via *getter*.

Listagem 3 – Monitoramento da Atividade do Companheiro

```
private void monitorarAtividadeParceiro() {
    FirebaseAtividadeDuplaController.getInstance().monitorarParceiro(new
    ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            AtividadeDTO dto = dataSnapshot.getValue(AtividadeDTO.class);
            if (dto != null) {
                Atividade atividade = new Atividade().parse(dto);
                switch (atividade.getEstado()) {
                    case EM_ANDAMENTO:
                        if
(!AtividadeEstadoSingleton.getInstance().isEmAndamento()) {
                            AtividadeActivity.this.updateFragment(new
                            AtividadeFragmentBundle(RETOMAR));
                        }
                        break;
                    case PAUSADA:
                        if
(!AtividadeEstadoSingleton.getInstance().isPausada()) {
                            AtividadeActivity.this.updateFragment(new
                            AtividadeFragmentBundle(PAUSAR));
                        }
                        case FINALIZADA:
                            break;
                }
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {}
    });
}
```

Fonte: Autoria própria.

A Listagem 3 mostra o método `monitorarAtividadeParceiro` da classe `AtividadeActivity`, classe onde é feita inscrição nas publicações do parceiro conectado. Este método invoca o método `monitorarParceiro` citado na Listagem 2, passando por parâmetro um `ValueEventListener` que verifica as alterações do companheiro. A lógica é realizada no método `onDataChange` do *listener*, sendo validado que os dados recebidos não são nulos, e assim, transformados em um objeto do tipo `Atividade`, nomeado `atividade`. É utilizada a estrutura condicional `switch/case` com a propriedade “tipo” do objeto `atividade`, caso o tipo seja “em andamento”, e a atividade do usuário atual estiver não estiver em andamento, ela é pausada. Caso a atividade do parceiro seja do tipo “pausada” e a atividade do usuário atual estiver não estiver suspensa, é dado sequência.

O fluxo de dados apresentados nas listagens 2 e 3 resulta em que, quando dois usuários estão realizando uma atividade em dupla, no momento que um deles pressiona o botão de pausa, a atividade do seu parceiro também é pausada, e quando é pressionado o botão para continuar com a atividade pausada, em um dos dispositivos, ambos os praticantes têm suas atividades retomadas.

5.5.3 Interação com o mapa da atividade

A Listagem 4 mostra como é feito para traçar a rota no mapa enquanto uma atividade está em andamento e o afastamento da câmera ao pausar o exercício.

Listagem 4 – Exibição da Rota no Mapa e Afastamento de Câmera

```
// Subscribe no LocalizacaoListener
@Subscribe(threadMode = ThreadMode.MAIN)
public void subscribe(LocationObservedData data) {
    if (data != null && data.getLocation() != null) {
        LatLng latLng = locationUtils.toLatLng(data.getLocation());
        ultimaPosicao = latLng;
        if (AtividadeEstadoSingleton.getInstance().isEmAndamento()) {
            moverCamera(latLng);
        }
    }
}

// Subscribe na atividade enquanto está em andamento
@Subscribe(threadMode = ThreadMode.MAIN)
public void subscribe(AtividadePosicao posicao) {
    if (posicao != null) {
        LatLng latLng = locationUtils.toLatLng(posicao);

        latLngBounds = latLngBounds == null ?
```

```

        new LatLngBounds.Builder()
            .include(latLng)
            .build() :
        latLngBounds.including(latLng);
        addPolyLine(latLng);
    }
}

// Subscribe nas ações de Pausar e Retomar
@Subscribe(threadMode = ThreadMode.MAIN)
public void update(AtividadeActivityBundle bundle) {
    switch (bundle.getMetodo()) {
        case PAUSAR:
            zoomOut();
            break;
        case RETOMAR:
            isZoomedOut = false;
            moverCamera(ultimaPosicao);
            break;
        default:
            break;
    }
}

private void moverCamera(LatLng latLng) {
    if (!this.isZoomedOut && this.isMapReady() && latLng != null) {
        this.gmap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,
this.zoomCamera));
    }
}

private void zoomOut() {
    if (this.isMapReady() && this.latLngBounds != null) {
        isZoomedOut = true;

this.gmap.animateCamera(CameraUpdateFactory.newLatLngBounds(this.latLngBounds,
this.paddingMap));
    }
}

private void addPolyLine(LatLng latLng) {
    if (isMapReady()) {
        this.polylineOptions.add(latLng);
        this.gmap.addPolyline(this.polylineOptions);
    }
}
}

```

Fonte: Autoria própria.

No trecho acima são demonstrados métodos da classe `MapaFragment`, na qual é exibido o mapa durante uma atividade. Os três métodos públicos, de retorno `void`, assinatura

subscribe e anotados com @Subscribe, possuem essa anotação pois são disparados por meio do EventBus.

O primeiro método subscribe da listagem, com parâmetro do tipo LocationObservedData, é invocado toda vez que o GPS recebe uma informação da posição atual, e tem a função de receber um objeto que possui dados de latitude e longitude, convertê-lo em um objeto do tipo LatLng e atribuí-lo à variável ultimaPosicao da classe, e caso a atividade estiver em andamento, passa-o para o método moverCamera, que, por sua vez, posiciona a câmera do mapa na coordenada desejada.

O segundo método subscribe, que possui como argumento um objeto de tipagem AtividadePosicao, é chamado sempre que é publicado, ao EventBus, um objeto do tipo em questão, neste caso, na tela de atividade ao serem recebidos do GPS e a atividade estar em andamento. Neste método a posição recebida, caso não seja nula, é convertida para um objeto do tipo LatLng. Posteriormente a isso, caso a variável latLngBounds – do tipo LatLngBounds, que representa um retângulo em coordenadas geográficas, para delimitar áreas – tenha seu valor como nulo, é atribuída a ela uma nova instância, com os dados de coordenadas recebidos. Caso contrário, a nova posição é adicionada à esta variável, para que possa ser delimitada a área mostrada ao afastar a câmera. Por fim, é disparado o método addPolyLine, que tem a responsabilidade de adicionar a coordenada recebida por parâmetro à variável polylineOptions – do tipo PolylineOptions, classe que tem como função desenhar linhas no mapa – e finalmente, esta é adicionada ao mapa, resultando no traçado do percurso que está sendo realizado.

O terceiro e último método com anotação @Subscribe, observa por mudanças efetuadas no EventBus originadas quando os botões visíveis na tela de atividade são pressionados, e dentro do método que se retrata, é feita a checagem por qual ação foi realizada. No caso de a ação ser do tipo “Pausa”, é executado o método zoomOut, que valida se o mapa e os limites de coordenadas são diferentes de nulo, e em condição verdadeira, marca a flag isZoomedOut como verdadeiro e faz a animação de afastamento da câmera, para ser mostrado o percurso inteiro, semelhante à uma vista aérea. Se a ação for do tipo “Retomar”, a flag isZoomedOut é marcada como falso e é invocado o método moverCamera com a variável de classe ultimaPosicao, para que, no mapa, a câmera foque na última posição que foi recebida.

5.5.4 Cálculo de gasto calórico

Na Listagem 5 é retratada a regra de negócio em que é calculada a quantidade de calorias queimadas pelo praticante.

Listagem 5 – Cálculo de gasto calórico

```
public Long calorias(Double pesoKg, Double distanciaMetros, Long
duracaoMilis) {
    Double velocidadeKmh = velocidadeEmKmh(distanciaMetros, duracaoMilis);
    double minutos =
    conversorTempo.fromMilliseconds(duracaoMilis).toMinutes();
    Double mets = unidadeMetabolica.getMet(velocidadeKmh);
    return calorias(mets, pesoKg, minutos);
}

private Long calorias(double mets, double pesoKg, double minutos) {
    double mililitroPorMinuto = mets * UnidadeMetabolica.V_O2 * pesoKg;
    double litroPorMinuto =
    conversorVolume.fromMilliliters(mililitroPorMinuto).toLiters();
    double caloriasPorMinuto = litroPorMinuto *
    UnidadeMetabolica.CAL_1L_O2;
    double gastoTotal = caloriasPorMinuto * minutos;
    return Double.valueOf(gastoTotal).longValue();
}
```

Fonte: Autoria própria.

Na seção de código acima, o primeiro método `calorias`, que possui os argumentos `pesoKg`, `distanciaMetros` e `duracaoMilis`, dos tipos `Double`, `Double` e `Long`, respectivamente, tem o dever de converter estes valores para outras unidades de medidas, e repassá-los ao outro método de mesma assinatura, porém, com sobrecarga. Os valores de distância em metros e duração da atividade em milissegundos, são convertidos para quilômetros por hora, e atribuído na variável `velocidadeKmh`. A duração em milissegundos é convertida desta unidade de medida, para minutos, por meio da biblioteca `UnitOf`, e valor, este convertido, dado à variável `minutos`. Com a velocidade em quilômetros por hora é possível obter o valor da variável `mets`, que representa a quantidade de unidades metabólicas equivalentes àquela velocidade. Por fim, os valores de `velocidadeKmh`, `minutos` e `mets` são repassados ao segundo método `calorias`.

O segundo método de assinatura `calorias` é responsável por efetuar o cálculo estimado de gasto calórico. Este cálculo leva em consideração o peso da pessoa, tempo de atividade e quantidade de METs, considerados a partir da velocidade.

Para obter a quantidade de calorias queimadas uma atividade, são necessários dois valores finais: volume de oxigênio por minuto, atribuído na variável `litroPorMinuto` e a quantidade de calorias gastas por minuto, salva na variável `caloriasPorMinuto`.

A fim de obter-se o volume em litros por minuto, é necessário, primeiramente medir os mililitros por minuto, que são obtidos pela multiplicação da quantidade de unidades metabólicas, pela constante de volume de oxigênio por minuto (3,5 mililitros) – gravado na constante `UnidadeMetabolica.V_O2` – e pelo peso, em quilogramas, do praticante. Este valor é registrado na variável `mililitroPorMinuto`, que, por sua vez, é transformada em litros por minuto por meio do conversor de volume da `UnitOf`.

Para conseguir a quantidade de calorias gastas em um minuto, basta multiplicar o volume de oxigênio por minuto pela constante de energia gasta por cada litro de oxigênio (5 calorias) – definida na constante `UnidadeMetabolica.CAL_1L_O2`. O resultado desta multiplicação é armazenado na variável `caloriasPorMinuto`.

Por último, o valor total de calorias queimadas no tempo total de atividade é o produto das calorias por minuto com o tempo em minutos, gravado na variável `gastoTotal`. O retorno do método que envolve este cálculo é a parte inteira de `gastoTotal`.

Para comparação de resultados, foi utilizado o site <https://caloriesburnedhq.com/calories-burned-running/>, que possui uma calculadora de calorias baseada no peso, distância e tempo. Os resultados colhidos utilizando as fórmulas desenvolvidas neste trabalho foram muito próximos aos que são gerados pela calculadora do site, desde que informados os mesmos parâmetros.

5.5.5 Cálculo de pontuação

O algoritmo para computar a pontuação que o usuário conquista após uma atividade é dada pelo trecho apresentado na Listagem 6.

Listagem 6 – Cálculo de gasto calórico

```
public Long calcular(Long duracaoSegundos, Double distanciaMetros, Long
calorias, AtividadeTipo tipo) {
    double duracaoMinutos = new
UnitOf.Time().fromSeconds(duracaoSegundos).toMinutes();
    double distanciaKm = new
UnitOf.Length().fromMeters(distanciaMetros).toKilometers();

    double distanciaTempo = duracaoMinutos * distanciaKm;
    long pontuacao = (long) (distanciaTempo + calorias);
}
```

```

    long multiplicadorTipo = AtividadeTipo.DUPLA.equals(tipo) &&
duracaoMinutos >= 15 ? 2L : 1L;
    final int FATOR = 5;

    return pontuacao * multiplicadorTipo * FATOR;
}

```

Fonte: Autoria própria.

O método calcular retratado na Listagem 6 recebe quatro argumentos, sendo eles: `duracaoSegundos`, do tipo `Long`, indicando o tempo de atividade em segundos; `distanciaMetros`, de tipagem `Double`, o qual se refere à distância total percorrida, em metros; `calorias`, do tipo `Long`, relativo à quantidade de calorias gastas na atividade; e, por fim, `tipo`, de tipagem `AtividadeTipo`, referente ao modo da atividade (sozinho ou dupla).

Os valores de duração em segundos e distância em metros são convertidos em duração em minutos, e distância em quilômetros, respectivamente, e salvos nas variáveis `duracaoMinutos` e `distanciaKm`, nesta ordem. Estas duas variáveis são, então, multiplicadas uma pela outra, gerando o resultado da variável `distanciaTempo`, que, por sua vez é somada à quantidade de calorias, e extraída a parte inteira, gerando a quantidade de pontos, na variável `pontuacao`.

Há, ainda, um multiplicador, que duplica os pontos caso a atividade seja em dupla e tenha duração de, no mínimo, 15 minutos. O resultado dessa expressão é guardado na variável `multiplicadorTipo`.

O retorno deste método é o produto dos pontos pelo multiplicador e pelo número 5 (cinco), fator definido para passagem de níveis de progresso ser mais acessível.

5.5.6 Acesso ao Banco de Dados Local

A comunicação com o banco de dados do dispositivo é realizada por meio da API do *Room Database*, com classes que implementam esta API. As listagens 7 e 8 e mostram, como é realizado o acesso ao banco de dados.

Listagem 7 – Criação do banco de dados

```

@Database (
    exportSchema = false,
    entities = {
        AtividadeDTO.class,
        AtividadePosicaoDTO.class,

```

```

        UsuarioNivelProgressoDTO.class,
    }, version = 2)
public abstract class AppDatabase extends RoomDatabase {

    private final static String DB_NAME = "tcc_ads_jeanpeiter";

    public abstract AtividadeDao atividadeDao();
    public abstract UsusarioNivelProgressoDao ususarioNivelProgressoDao();

    private static AppDatabase instance;
    public static AppDatabase getInstance(Context context) {
        if (instance == null) {
            instance = Room
                .databaseBuilder(context, AppDatabase.class,
AppDatabase.DB_NAME)
                .build();
        }
        return instance;
    }
}
}

```

Fonte: Autoria própria.

A classe abstrata `AppDatabase`, mostrada acima, é a responsável por realizar a criação do banco de dados, efetuar o acesso direto a ele, e informar quais são as entidades gravadas. Esta classe é anotada com `@Database`, anotação oriunda do *Room*, com as propriedades `exportSchema`, significando se o banco de dados da aplicação deve ser exportado para uma pasta no armazenamento interno do dispositivo; `entities`, indicando as classes que serão persistidas; e `version`, para a versão do banco de dados. Esta classe possui os métodos abstratos `atividadeDao`, do tipo `AtividadeDao`, e `ususarioNivelProgressoDao`, tipado com `UsusarioNivelProgressoDao`, que são interfaces separadas para ler e escrever no banco de dados as informações das atividades realizadas e do nível de progresso do usuário, respectivamente.

A classe `AppDatabase` segue o *design pattern Singleton*, no qual existe apenas uma instância da mesma classe na aplicação inteira. O método `getInstance` deste *Singleton* faz a verificação se a instância é nula, atribui-a um valor utilizando a API do *Room*, com o padrão *Builder*, sendo passado por parâmetros o contexto da chamada, a própria classe, e o nome do banco de dados, sendo essa instância o retorno do método `getInstance`.

Listagem 8 – Interface do Banco de Dados para Níveis de Progresso

```

@Dao
public interface UsusarioNivelProgressoDao {

```

```

@Insert
void create(List<UsuarioNivelProgressoDTO> niveisProgresso);

@Update
void update(UsuarioNivelProgressoDTO nivelProgresso);

@Query(UsusarioNivelProgressoQueries.NIVEL_ATUAL)
UsuarioNivelProgressoDTO nivelAtual(String uid);

@Query(UsusarioNivelProgressoQueries.HISTORICO)
List<UsuarioNivelProgressoDTO> historico(String uid);
}

```

Fonte: Autoria própria.

A interface `UsusarioNivelProgressoDao` possui a anotação `@Dao`, indicando ao sistema que é responsável por ler e gravar ao banco de dados. Esta possui quatro métodos:

- `create`: anotado com `@Insert` e possuindo o parâmetro `niveisProgresso` do tipo lista de `UsuarioNivelProgressoDTO`, o *Room Database* compreende que este método é responsável por fazer a persistência de vários objetos do tipo `UsuarioNivelProgressoDTO`;
- `update`: com a anotação `@Update` e argumento um objeto do tipo `UsuarioNivelProgressoDTO`, da mesma forma como o método anterior, o *Room* determina que será atualizado um registro de nível de progresso;
- `nivelAtual` e `historico`: estes métodos possuem a anotação `@Query` – que, por sua vez, recebe uma `String` por parâmetro, tendo o valor de uma consulta ao banco de dados. Cada método recebe uma `String` por parâmetro, que é utilizada na consulta, em tempo de execução.

6 CONCLUSÃO

O intuito principal do aplicativo é incentivar a prática de atividades físicas, utilizando da mecânica de um jogo, para estimular o usuário a chegar ao último nível, sendo necessário realizar sempre mais exercícios, e possibilitando que as corridas e caminhadas sejam realizadas em companhia de outra pessoa, para que o avanço nos níveis de progresso seja mais rápido.

A decisão pelo uso dos serviços e ferramentas utilizadas justifica-se na facilidade de desenvolvimento, pois com as bibliotecas de terceiros algumas rotinas foram simplificadas, em relação a códigos-fonte. Caso não tivessem sido adicionadas ao projeto, as classes e métodos seriam muito mais verbosos. Além disso, a escolha em usar estas ferramentas explica-se em oferecer, também, a melhor experiência ao usuário, visto que as APIs de terceiros já possuem padrões e resultados bem consolidados, com melhor performance comparado a codificar tudo de forma própria, e sendo utilizadas em aplicativos com incontáveis usuários, do mundo todo.

Algumas partes, citadas na primeira versão deste trabalho, não foram realizadas, ou tiveram de ser implementadas diferentemente de como havia sido citado. A primeira delas, é a parte de oferecimento de dicas e indicações de alimentos ao usuário, conforme as calorias queimadas durante o percurso. Este requisito foi retirado do escopo, devido a algumas variáveis estarem envolvidas, por exemplo a preferência de alimentos do usuário ser diferente dos alimentos disponíveis para sugestão e pelo fato de que seria apenas mostrado em tela logo após finalizada a atividade, o que seria uma ação demasiada simples para o usuário, pelo tempo que é visualizada, porém para o desenvolvedor seria de mesma complexidade que o processo de níveis de progresso, por exemplo. A segunda parte é a comunicação entre dispositivos, por meio de *Bluetooth*. Inicialmente foram criados processo de comunicação via *Bluetooth*, tanto manualmente quanto utilizando bibliotecas de terceiros, porém foram encontradas limitações a nível de código-fonte, que impossibilitaram o seguimento do uso deste protocolo. Além disso, analisando o avanço de tecnologias e *gadgets* sem fio, como relógios e pulseiras inteligentes, ou até mesmo fones de ouvido, visto que mesmo praticando uma atividade ao lado de outra pessoa, o usuário pode utilizar outro dispositivo pessoal conectado ao seu *smartphone*. Considerando estas duas circunstâncias, foi decidido substituir a troca de dados via *Bluetooth*, pela comunicação por meio do *Firebase Realtime Database*, em que ambos os dispositivos se conectam via internet e têm seus dados da atividade física compartilhados em tempo real.

Um ponto negativo de proporcionar a comunicação entre dois dispositivos via internet, é de que, em algum momento, um dos membros da dupla perca sinal de rede, fazendo com que

seu parceiro não receba mais atualizações naquele instante, impactando até mesmo caso um deles desejar pausar a atividade, o que não será replicado ao seu companheiro.

As vantagens são, como citado anteriormente, o serviço de *Bluetooth* do dispositivo fica livre para o usuário utilizar como bem entender, sendo ouvido músicas ou utilizando outro dispositivo pessoal, ligado ao celular. Além disso, o usuário não fica refém do raio de alcance do *Bluetooth* do seu celular, podendo assim, realizar uma atividade com alguém que não está no mesmo local, apenas precisando, por exemplo, combinar data e horário com um amigo que mora em outro estado, até mesmo país, para iniciarem uma corrida juntos.

Para trabalhos futuros no aplicativo, podem ser implementadas atividades em grupo, para três ou mais pessoas poderem compartilhar seus dados, sendo possível realizar competições com inúmeros participantes.

Por fim, pode-se constatar que o sistema desenvolvido atende os propósitos caracterizados, e o objetivo concluído. A aplicação, em seu estado final, é capaz de realizar, etapa por etapa, os processos citados na Seção 5.1 (Escopo do Sistema) e segue a estrutura definida na Modelagem do Sistema, na Seção 5.2.

REFERÊNCIAS

- Ainsworth BE, Haskell WL, Whitt MC, Irwin ML, Swartz AM, Strath SJ, et al. **Compendium of physical activities: an update of activity codes and MET intensities.** *Med Sci Sports Exerc.* 2000;32(9 Suppl):S498-504.
- ANDROID ARCHITRECTURE. Disponível em: http://www.tutorialspoint.com/android/android_architecture.htm. Acesso em: 11 abr. 2019.
- ANDROID DEVELOPERS. Disponível em: <https://developer.android.com/guide/platform>. Acesso em: 10 abr. 2019.
- ANDROID DEVELOPERS - Room. Disponível em: <https://developer.android.com/reference/android/arch/persistence/room/package-summary>. Acesso em: 10 abr. 2021.
- AVELAR E DUARTE. **Telefonia móvel 2018 (estatísticas).** 2019. Disponível em: <http://www.avellareduarte.com.br/fases-projetos/conceituacao/demandas-do-publico/pesquisas-de-usuarios-atividades-2/dados-sobre-o-publico-alvo/dispositivos-moveis-2018-estatisticas>. Acesso em: 20 mar. 2019.
- BARROS, Ricardo do R.; SILVA, Luciana Rodrigues. **Promoção da atividade física na infância e adolescência: manual de orientação.** Grupo de Trabalho em Atividade Física. Sociedade Brasileira de Pediatria, 2017, p. 1-14.
- BRAY, Tim. **What Android is.** 2010. Disponível em: <http://www.tbray.org/ongoing/When/201x/2010/11/14/What-Android-Is>. Acesso em: 11 abr. 2019.
- CRUZ, Michele Santos da; BERNAL, Regina Tomie Ivata; CLARO, Rafael Moreira. Tendência da prática de atividade física no lazer entre adultos no Brasil (2006-2016). **Cad. Saúde Pública**, 2018; v. 34, n. 10, e00114817, p. 1-14.
- DEMARTINI, Felipe. **Brasil já tem mais de um smartphone ativo por habitante.** 2018. Disponível em: <https://canaltech.com.br/produtos/brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-112294>. Acesso em: 20 mar. 2019.
- FREIRE, Rafael Silveira. Prática regular de atividade física: estudo de base populacional no Norte de Minas Gerais, Brasil. **Rev Bras Med Esporte**, v. 20, n.5, Set/Out, 2014, p. 345-349.
- GADHIYA, Sohil A.; WANDRA, Kalpesh H.; VAGHELA, Vimalkumar B. **Role of mobile augmentation in mobile application development.** In: IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA), 2012, p. 1-5.
- GAVALAS, Damianos; ECONOMOU, Daphne. Development platforms for mobile applications: status and trends. January/February 2011, **IEEE Software**. 2011, p. 77-86.
- GUANA, Victor; ROCHA, Fabio; HINDLE, Abram; STROUL, Eleni. **Do the stars align? Multidimensional analysis of Android's layered architecture.** MSR 2012, p. 124-127.

LI, Yang; WANG, Xinning. **Design of adaptive media transmission based on android platform**. In: 2014 IEEE International Conference on Consumer Electronics, China, 2014, p. 1-4.

OMS - Organização Mundial da Saúde. **OMS: 80% dos adolescentes no mundo não praticam atividades físicas suficientes**. 2018. Disponível em: <https://nacoesunidas.org/oms-80-dos-adolescentes-no-mundo-nao-praticam-atividades-fisicas-suficientes>. Acesso em: 02 abr. 2019.

PISA, Marcel. **Saiba mais sobre a importância da atividade física na prevenção de doenças**. 2017. Disponível em: <https://claretiano.edu.br/batatais/noticias/88557/saiba-mais-sobre-a-importancia-da-atividade-fisica-na-prevencao-de-doencas>. Acesso em: 18 mar. 2019.

PRESSE, da France. **Número de obesos no mundo supera o de famintos, diz Cruz Vermelha**. 2011. Disponível em: <http://g1.globo.com/ciencia-e-saude/noticia/2011/09/numero-de-obesos-no-mundo-supera-o-de-famintos-diz-cruz-vermelha.html>. Acesso em: 20 mar. 2019.

PRIMORAC, Sanja; RUSSO, Mladen. **Android application for sending SMS messages with speech recognition interface**. In: 5th International Convention MIPRO, 2012, p. 1763-1767.

SCHMIDT, Maria Inês; DUNCAN, Bruce Bartholow; SILVA, Gulnar Azevedo e; MENEZES, Ana Maria; MONTEIRO, Carlos Augusto; SANDHI, Maria Barreto. **Doenças crônicas não transmissíveis no Brasil: carga e desafios atuais**. The Lancet Series, 2011, p. 61-74. Disponível em: <http://download.thelancet.com/flatcontentassets/pdfs/brazil/brazilpor4.pdf>. Acesso em: 05 abr. 2019.

SQLITE. **Features of SQLite**. Disponível em: <<http://www.sqlite.org/features.html>>. Acesso em: 12 abr. 2019.

TAWALBEH, Lo'ai A.; BAKHADER, Waseem. **A mobile cloud system for different useful applications**. In: 4th International Conference on Future Internet of Things and Cloud Workshops, IEEE, 2016, p. 295-298.

The Compendium of Physical Activities (Compendium). Disponível em: <https://sites.google.com/site/compendiumofphysicalactivities/corrected-mets>. Acesso em: 25 feb. 2021.

WANNMACHER, Lenita. **Obesidade como fator de risco para morbidade e mortalidade: evidências sobre o manejo com medidas não medicamentosas**. OPAS/OMS – Representação Brasil. ISBN: 978-85-7967-108-1 Vol. 1, Nº 7, 2016, p. 1-10.

WORLD HEALTH ORGANIZATION. **Global health and aging**. National Institute on Aging. National Institutes of Health. U.S. Department of Health and Human Services, 2011. Disponível em: https://www.who.int/ageing/publications/global_health.pdf. Acesso em: 02 abr. 2019.

ZOMBIES, RUN!. Disponível em: <https://zombiesrungame.com>. Acesso em: 02 abr. 2019.

APÊNDICE A – QUESTIONÁRIO

Pesquisa Para Desenvolvimento de Aplicativo de Corridas/Caminhadas

* Required

Sobre Você (1/3)

Qual sua idade? *

- Menor de 18 anos
- Entre 18 e 25 anos
- Entre 26 e 30 anos
- Entre 31 e 35 anos
- Entre 36 e 40 anos
- Entre 41 e 45 anos
- Entre 46 e 50 anos
- Entre 51 e 55 anos
- Entre 56 e 60 anos
- Acima de 60 anos

Qual seu sexo? *

Choose

Feminino

Masculino

Prefiro não dizer

Rotinas diárias: *

- Apenas trabalho
- Apenas estudo
- Trabalho e estudo
- Não trabalho e não estudo
- Other: _____

Em média, quanto tempo livre você tem por dia?

- Não tenho tempo livre
- Menos de 1 hora
- De 1 a 2 horas
- De 2 a 5 horas
- De 5 a 8 horas
- De 8 a 12 horas
- Mais que 12 horas

O que gosta de fazer no tempo livre?

Your answer

BACK

NEXT

Pesquisa Para Desenvolvimento de Aplicativo de Corridas/Caminhadas

* Required

Sobre Atividades Físicas (2/3)

Quanto tempo você dedica por semana para a prática de atividades físicas? *

- Não pratico atividades físicas
- Até 30 minutos
- Entre 30 minutos e 1 hora
- Entre 1 e 3 horas
- Entre 3 e 5 horas
- Mais de 5 horas

Você costuma se deslocar a pé para ir trabalhar, estudar, fazer compras, etc? *

- Sim, sempre a pé
- Sim, mas às vezes com outros meios de transporte
- Não vou a pé, mas vou de bike/skate/outro meio de transporte limpo
- Não, vou apenas de carro/moto/ônibus

Que tipo de atividades físicas você pratica normalmente? *

- Não pratico
- Musculação
- Corrida/Caminhada
- Ciclismo/Andar de bicicleta
- Futebol
- Basquete
- Vôlei
- Natação
- Other: _____

Você já utilizou algum aplicativo para medir seu desempenho enquanto fazia uma corrida, caminhada ou passeio (a pé ou bicicleta)? *

- Sim, e utilizo frequentemente
- Sim, mas apenas algumas vezes
- Não, mas já pensei/gostaria de utilizar
- Não, nunca utilizei

BACK

NEXT

Pesquisa Para Desenvolvimento de Aplicativo de Corridas/Caminhadas

* Required

Sobre Sua Experiência Com Aplicativos (3/3)

Qual o sistema operacional do seu celular? *

- Android
- iOS
- Windows Phone
- Other: _____

Quanto espaço livre tem no armazenamento do seu celular? *

- Muito, tenho espaço de sobra
- O suficiente, porém tenho que realizar limpezas
- Pouco, mesmo fazendo limpezas
- Nenhum, meus apps sempre lotam a memória

Você tem algum desses dispositivos móveis além do seu celular? *

Não tenho

Tablet

Smartwatch

Pulseira eletrônica

Notebook

Other: _____

Você gosta e utiliza jogos de celular? *

Gosto e uso com frequência

Gosto, mas uso pouco

Não gosto, mas às vezes baixo para testar

Não gosto e nunca utilizo

Escreva o que mais prende sua atenção em um app: *

Your answer _____

BACK

NEXT