

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUCIANO ZANCAN MAZZUTTI

**Planfit: Solução mHealth voltada para
personal trainers**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Valter Roesler

Porto Alegre
2019

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. André Inácio Reis

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“A maior parte dos homens é como a pedra do íman. Tem um lado que atrai e outro que repele.”

— VOLTAIRE

AGRADECIMENTOS

Gostaria de iniciar os agradecimentos pela minhas mães, Neuza e Neiva, ao meu pai Ignácio, que mesmo estando longe, sempre me apoiaram antes, durante, e com certeza apoiarão depois de minha graduação. Ainda, agradeço à nona Marcelina, ao meu irmão Géris, cunhada Milene, e minha sobrinha Laura.

Gostaria de agradecer, também, ao professor Valter Roesler e à educadora física Malu Kroeff Barbosa, pela oportunidade de participar do projeto Emagreca@Saudavel, que com certeza me proporcionou uma grande evolução, e me abriu portas para o mercado de trabalho.

Agradeço aos meus colegas e amigos do "Pagode", onde sempre há uma partida de Uno e boas risadas. Ammes, Elisandra, Tibu, Adão, Cabelo, e Susan, espero levá-los para toda a vida. Também, um agradecimento especial ao Matheus, que sempre esteve ao meu lado nas horas mais difíceis, e que espero que sempre esteja.

Finalmente, gostaria de agradecer a todos os demais, professores, colegas de graduação, colegas de trabalho, familiares, pelo carinho, aprendizado e companheirismo.

RESUMO

Este trabalho teve por objetivo mostrar a criação de um sistema voltado para certos tipos de profissionais e seus alunos/clientes. A validação do trabalho é proposta na área da saúde e qualidade de vida, especificamente na profissão de “personal trainer”. Foram utilizadas as melhores técnicas para a construção de aplicativos móveis para esta tarefa, e a concepção e validação do sistema foi realizada através de entrevistas com um profissional da área. Para utilização, o profissional poderá utilizar um sistema "*backend*"*web* para definir o conteúdo que seus clientes (usuários finais da aplicação móvel) irão visualizar.

Palavras-chave: Android. Aplicativo. *Personal Trainer*.

Planfit: mHealth Solution for Personal Trainers

ABSTRACT

This paper aimed to show the creation of a system aimed at certain types of professionals and their students/clients. Work validation is proposed in the area of health and quality of life, specifically in the profession of "personal trainer". The best techniques were used to build mobile applications for this task, and the design and validation of the system was performed through interviews with a professional in the area. For use, the professional may use a back-end web system (which is not the focus of this paper) to define the content that their clients (mobile application end users) will see.

Keywords: Android. Application. Personal Trainer.

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
REST	<i>Representational State Transfer</i>
JSON	<i>Javascript Object Notation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IMC	Índice de massa corporal
ICQ	Índice de cintura e quadris

LISTA DE FIGURAS

Figura 3.1	Tela de menu do aplicativo do Marco Aurélio Soares.	16
Figura 3.2	Imagens referentes ao aplicativo criado pelo Marco Aurélio Soares.	16
Figura 3.3	Imagens referentes ao aplicativo Trei.no.	17
Figura 3.4	Imagens referentes ao aplicativo Emagreca@Saudavel 1.	18
Figura 3.5	Imagens referentes ao aplicativo Emagreca@Saudavel 2.	19
Figura 4.1	Representação de um modelo cliente-servidor utilizando <i>smartphones</i>	23
Figura 4.2	Exemplo de requisição GET para obter as notícias do servidor	25
Figura 4.3	Representação da arquitetura do aplicativo	26
Figura 5.1	Tela de <i>login</i>	33
Figura 5.2	Tela principal do servidor.	34
Figura 5.3	Tela de detalhes do usuário.	34
Figura 5.4	Exibição dos gráficos de evolução do usuário no servidor.	35
Figura 5.5	Exibição das tarefas concluídas pelo usuário selecionado.	35
Figura 5.6	Lista de tarefas cadastradas.	36
Figura 5.7	Lista de notícias cadastradas.	36
Figura 5.8	Lista categorias de educação cadastradas.	37
Figura 5.9	Lista de subcategorias de educação cadastradas.	37
Figura 5.10	Lista treinos de usuários.	38
Figura 5.11	Atribuição de treino semanal a um usuário.	38
Figura 5.12	Lista de entidades do tipo "Gerais/Outros".	39
Figura 5.13	Lista de <i>feedbacks</i> de tarefas.	39
Figura 5.14	Lista de <i>feedbacks</i> de treinos.	39
Figura 5.15	Seção de envio de mensagens aos usuários.	40
Figura 5.16	Imagens referentes às telas de Cadastro de Usuários.	41
Figura 5.17	Tela inicial do cadastro e atualização de dados de perfil.	41
Figura 5.18	Imagens referentes às telas de preenchimento de dados e cálculo de IMC e ICQ.	42
Figura 5.19	Imagens referentes à apresentação do questionário ao usuário.	43
Figura 5.20	Imagens referentes à tela de tarefas.	44
Figura 5.21	Tarefa contendo imagem.	44
Figura 5.22	Imagens referentes à tela de subcategorias de educação.	45
Figura 5.23	Subcategoria de educação contendo vídeo.	45
Figura 5.24	Imagens referentes à tela de notícias.	46
Figura 5.25	Tela de evolução da aplicação móvel, com botões de adição de medidas...	47
Figura 5.26	Imagens referentes ao menu e perfil.	48
Figura 5.27	Tela de leitura de mensagens.	49
Figura 5.28	Imagens referentes às notificações.	49
Figura 5.29	Tela de configurações.	49
Figura 5.30	Tela de perguntas frequentes.	50

LISTA DE TABELAS

Tabela 3.1	Módulos do sistema - Informações de perfil ou pessoais	21
Tabela 3.2	Módulos do sistema - Abas do aplicativo.....	22
Tabela 3.3	Módulos do sistema - Funcionalidades gerais do menu	22
Tabela 4.1	Web Services do tipo POST (envio de informações do aplicativo móvel ao servidor)	28
Tabela 4.2	Web Services do tipo GET para todos os usuário(recebimento de infor- mações do servidor no aplicativo móvel).....	29
Tabela 4.3	Web Services do tipo GET por usuário (recebimento de informações do servidor no aplicativo móvel).....	31
Tabela 4.4	Web Services do tipo PUT (envio de informações do aplicativo ao servidor)	32
Tabela 6.1	Avaliação de Requisitos do Sistema	51

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo	11
1.2 Organização do Texto	12
2 CONCEITOS BÁSICOS	13
2.1 Diferença entre eHealth e mHealth	13
2.2 Banco de dados para dispositivos móveis	13
2.3 Comunicação entre dispositivos móveis e servidor web	13
2.4 APIs de imagens para dispositivos móveis	14
3 LEVANTAMENTO DE REQUISITOS	15
3.1 Trabalhos Relacionados	15
3.1.1 Aplicativo Móvel para Academia: Estudo de Tecnologias e Desenvolvimento	15
3.1.2 Trei.no	16
3.1.3 Emagreca@Saudavel	17
3.2 Levantamento de Requisitos Através de Entrevistas	19
4 PLANFIT: O SISTEMA PROPOSTO	23
4.1 Arquitetura do Servidor <i>web</i>	24
4.2 Arquitetura do Cliente (Dispositivo Móvel)	24
4.3 Detalhamento da comunicação	26
5 DETALHAMENTO DA IMPLEMENTAÇÃO	33
5.1 Implementação do servidor	33
5.2 Implementação do aplicativo móvel	40
6 VALIDAÇÃO DO SISTEMA	51
6.1 Avaliação de Requisitos	51
6.2 Entrevista de validação	51
7 CONCLUSÕES	53
REFERÊNCIAS	54

1 INTRODUÇÃO

Há uma importante preocupação para pacientes, médicos e formuladores de políticas sobre ser possível controlar os custos, mantendo a qualidade dos serviços de saúde (OWENS et al., 2011). Segundo (BERGMO, 2015), os custos com cuidados de saúde continuam a aumentar. O uso de tecnologias de informação e comunicação nos cuidados de saúde (*eHealth*) têm sido uma proposta para o aumento de eficiência e qualidade dos cuidados de saúde, além de melhorar o acesso a esses (EYSENBACH, 2001; BASHSHUR et al., 2013)

Existem profissões em que o trabalho consiste, em grande parte, na criação de planos e metas e sua adequação aos clientes de forma que a personalização se torna quase automática. Pode-se exemplificar com alguns profissionais da área da saúde e de qualidade de vida, como nutricionistas e treinadores (*personal trainers*). Nesses casos, um cliente, aqui definido como o usuário de serviços prestados pelo profissional, recebe um plano de ação (dieta, sequência de exercícios, etc) a ser cumprido.

De acordo com (CAR; SHEIKH, 2004), (KATZ et al., 2003) e (JENNETT et al., 2003), *eHealth* tem potencial de tornar os profissionais da área de saúde mais eficientes e produzir benefícios sistêmicos e *spin-offs* tecnológicos. Por exemplo, consultas e monitoramento online de questões de saúde não urgentes têm potencial de reduzir visitas em consultórios e encontros com os profissionais, consequentemente reduzindo tempo gasto por ambas as partes, e tornando os profissionais mais eficientes e com possibilidade de atender mais clientes.

1.1 Objetivo

Este trabalho tem por objetivo fornecer uma tecnologia que permita facilitar a comunicação entre profissional de saúde de certas áreas e seus clientes. A área de foco é nas profissões com rotinas de trabalho repetitivas, como por exemplo nutricionistas, que criam planos de dieta para cada paciente, ou *personal trainers*, que criam listas de exercícios para cada aluno.

Pode-se citar os seguintes benefícios dessa interação remota entre profissional e cliente:

a) Otimização do seu tempo e do seu cliente, visto que o número de encontros presenciais é minimizado;

- b) Acompanhamento em tempo real das atividades efetuadas pelos clientes;
- c) Possibilidade de aumento do número de clientes, visto que o tempo foi otimizado;
- d) Possibilidade de diminuição de custo para o cliente, pois o custo do profissional é diluído em mais pessoas;
- e) Possibilidade de aumento de renda, pelo aumento no número de clientes;
- f) Possibilidade de envio de informações de parceiros (propagandas), possibilitando uma renda extra ao profissional e maior venda a seus parceiros.

Para atingir esse objetivo, foi desenvolvido um sistema chamado Planfit, composto por uma aplicação para smartphones e uma aplicação web, e foi feita a validação na área de personal trainers. Um grande diferencial deste sistema é que o profissional altera o conteúdo do aplicativo móvel através da aplicação web, não necessitando maiores conhecimentos em informática.

1.2 Organização do Texto

Este capítulo 1 (Introdução) tem o objetivo de relacionar dados de uma visão geral deste trabalho de conclusão. O restante do texto está organizado da seguinte forma:

O capítulo 2, chamado de Conceitos básicos, é um capítulo auxiliar, que traz alguns conceitos que facilitam o entendimento do trabalho.

No capítulo 3, chamado Levantamento de Requisitos, estão expostos alguns trabalhos relacionadas a este, que serviram para a criação de ideias iniciais do sistema. Também, ao final, estão contidas entrevistas realizadas com profissionais da área de *personal trainer*, trazendo uma definição para o sistema desenvolvido em prol deste trabalho.

No capítulo 4 (Planfit: O sistema proposto) está descrita a arquitetura do sistema desenvolvido, tanto na parte *web*, quanto na parte do aplicativo móvel, além de um detalhamento sobre a comunicação entre ambos para a integração do sistema como um todo.

No capítulo 5 (Detalhamento da implementação) está detalhada a parte visual do sistema, com mostra de suas interfaces, e explicação sobre o fluxo de utilização do sistema (tanto na parte do servidor quanto do aplicativo).

Para finalizar, no capítulo 6 (Validação do sistema), está a validação realizada junto com profissional da área de *personal trainer*, para quem é destinado o sistema, e no capítulo 7 estão as conclusões. Ao final, referências.

2 CONCEITOS BÁSICOS

Este capítulo mostra alguns conceitos que facilitam o entendimento deste trabalho de graduação.

2.1 Diferença entre eHealth e mHealth

De acordo com a Organização Mundial da Saúde (OMS), também conhecida como *World Health Organization* (WHO) ¹, *mobile-Health* ou *m-Health* ² conceitua-se pelo uso de aplicações móveis, como *smartphones* para auxiliar o monitoramento de pacientes no cuidado com a saúde, enquanto *eHealth* ³ são práticas relacionadas ao cuidado com a saúde auxiliadas por meios eletrônicos. Sendo assim, o conceito de *mHealth* está incluído dentro de *eHealth*.

2.2 Banco de dados para dispositivos móveis

Dispositivos móveis, assim como aplicações *web*, também possuem banco de dados para persistência de informações no aparelho. Por padrão, o Android utiliza o banco de dados SQLite (BHOSALE; PATIL; PATIL, 2015), porém existem alternativas, como o Realm ⁴, que é um banco de dados *no-SQL* (*Structured Query Language*). Como o aplicativo não necessitou de um banco de dados muito robusto, e também pela facilidade de desenvolvimento, foi utilizado o banco de dados Realm na aplicação móvel.

2.3 Comunicação entre dispositivos móveis e servidor web

Para a melhor comunicação entre servidor e aplicativo, sem que haja perda de dados, e que possua um controle de envio e recebimento de dados com fácil implementação, foi utilizado a biblioteca Retrofit (VOGEL; SCHOLZ; WEISER, 2017). O Retrofit é um cliente REST (*Representational State Transfer*) que permite *upload* e *download* fácil de tipos de dados através de *webservices*. Por padrão são utilizados JSONs (*Javascript*

¹WHO (Acesso em novembro de 2019). <https://www.who.int/>.

²mHealth (Acesso em novembro de 2019). <https://en.wikipedia.org/wiki/MHealth>.

³mHealth (Acesso em novembro de 2019). <https://en.wikipedia.org/wiki/EHealth>.

⁴Realm (Acesso em novembro de 2019). <https://realm.io/products/realm-database/>.

Object Notation).

2.4 APIs de imagens para dispositivos móveis

A biblioteca Retrofit é utilizada para envio e recebimento de dados, em geral JSONs. Porém para o controle de imagem, com cacheamento e melhor desempenho de rede e espaço físico do dispositivo, foi utilizado a biblioteca Picasso. Picasso⁵ é uma biblioteca para carregamento de imagens, que provê uma melhor manipulação no *download* de imagens com utilização mínima de memória.

Além disso, qualquer meio que utilize o protocolo HTTP (*Hypertext Transfer Protocol*) para comunicação, como o Retrofit e o Picasso, terá como apoio a biblioteca OkHttp⁶, que é um Cliente HTTP e provê carregamentos de dados mais rápidos contribui para a diminuição do consumo de banda.

⁵Picasso (Acesso em novembro de 2019). <http://square.github.io/picasso/>.

⁶OkHttp (Acesso em novembro de 2019). <https://square.github.io/okhttp/>.

3 LEVANTAMENTO DE REQUISITOS

Neste capítulo estão são expostas algumas soluções já existentes voltadas a *personal trainers*, que nortearam parte do desenvolvimento do sistema, além de entrevistas para uma completa definição do Planfit.

3.1 Trabalhos Relacionados

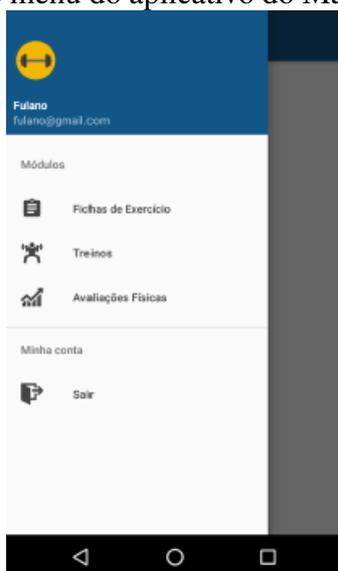
Nesta seção serão apresentados alguns trabalhos relacionados a este, que serviram de norte à concepção do sistema Planfit. O critério de escolha foi por criação de aplicativo Mobile/Android, e por área de aplicação (*academias/fitness*).

3.1.1 Aplicativo Móvel para Academia: Estudo de Tecnologias e Desenvolvimento

No trabalho de conclusão de curso do Marco Aurélio Soares (SOARES, 2016), o autor buscou uma maneira de automatizar o gerenciamento de fichas de exercícios e avaliações físicas nas academias, motivado pelo fato de muitas academias realizarem a função manualmente devido à falta de aplicações ou até mesmo da falta de implementação de algumas tecnologias já existentes. O autor realizou um estudo sobre as tecnologias na área de aplicativos móveis, e com isso desenvolveu um aplicativo contendo funções de gerenciamento de fichas de exercício, treinos e avaliações físicas, gerando um produto para auxiliar as atividades nas academias.

A aplicação desenvolvida no referido TCC possui funcionalidade de criar, editar e exibir treinos ao usuário, criação de fichas de avaliação física, e envio dessas informações para armazenamento em um servidor, porém apenas para persistência de dados. A tela de menu, com todas as funcionalidades do aplicativo encontra-se na Figura 3.1, a tela de treinos está na Figura 3.2(a), onde é possível visualizar os treinos para o usuário, e a tela de criação de avaliações físicas está na Figura 3.2(b). Uma diferença deste aplicativo para o Planfit é a possibilidade de avaliação física, porém pouco escalável por necessitar de cadastro no próprio aparelho. Já no Planfit, como será visto nos próximos capítulos, o profissional pode cadastrar o treino para cada usuário através da interface do servidor *web*, facilitando seu trabalho.

Figura 3.1: Tela de menu do aplicativo do Marco Aurélio Soares.



Fonte: (SOARES, 2016)

Figura 3.2: Imagens referentes ao aplicativo criado pelo Marco Aurélio Soares.



(a) Tela de treinos.



(b) Tela de criação de avaliação física.

Fonte: (SOARES, 2016)

3.1.2 Trei.no

O aplicativo Trei.no¹ foi idealizado para *personal trainers*. O aplicativo possui *feed* com notícias, novidades, divulgação de promoções e eventos ao aluno, além de trei-

¹trei.no (Acesso em novembro de 2019). <http://trei.no>.

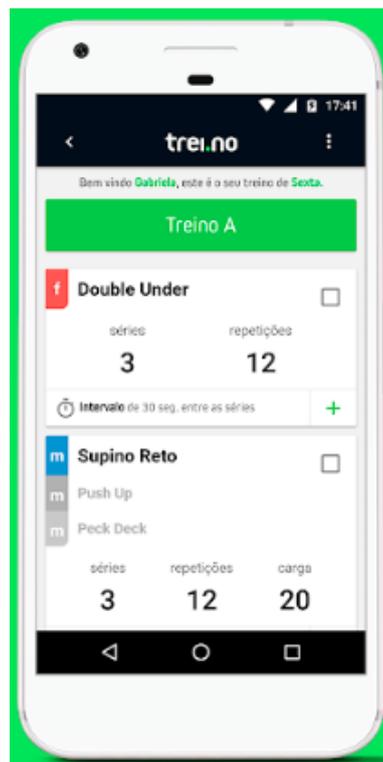
nos para os alunos com contador regressivo, alerta de vibração, e descanso entre séries. A Figura 3.3(b) mostra a tela de notícias e eventos, e a Figura 3.3(b) mostra a tela de treinos.

Em relação ao Planfit, o aplicativo Trei.no possui funcionalidades semelhantes como lista de treinos, contador regressivo e tela de notícias. Entretanto, o Planfit possui uma gama maior de funcionalidades como abas de evolução, educação, exibição de perfil e mensagens, como será exibido adiante.

Figura 3.3: Imagens referentes ao aplicativo Trei.no.



(a) Tela de notícias e eventos do aplicativo Trei.no.



(b) Tela de treino do aplicativo Trei.no

Fonte: Site <https://trei.no/>

3.1.3 Emagreça@Saudavel

O aplicativo Emagreça@Saudável (Lúcia Kroeff Barbosa et al., 2018) é um sistema de apoio à educação em saúde com foco na mudança de comportamento para hábitos mais saudáveis e melhoria da qualidade de vida. O aplicativo possui mensagens motivacionais, contador de passos, aulas e treinos diversos, informações e receitas de alimentação, histórias de superação e acompanhamento de evolução através de medidas.

O Emagreça@Saudavel serviu de base para a criação do Planfit, por isso possui layout e algumas funcionalidades muito parecidas (tela de evolução com gráficos por

exemplo), como será exibido nos próximos capítulos.

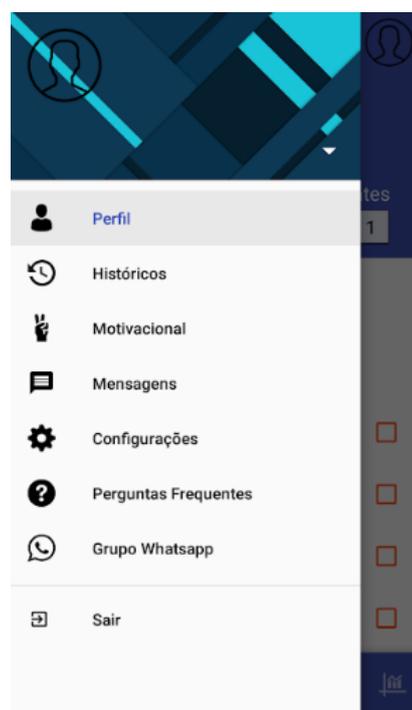
A Figura 3.4(a) mostra a tela de tarefas do aplicativo. Como poderá ser visto nos próximos capítulos, a diferença para o Planfit é que este possui várias listas, uma por dia da semana, enquanto o Emagreca@Saudável possui uma única lista de tarefas. A Figura 3.4(b) mostra as possibilidades do menu do Emagreca@Saudavel. A Figura 3.5(a) apresenta uma lista de exercícios ao usuário, assim como a quantidade de calorias consumidas por hora de prática. Além disso, o aplicativo contém um *ranking* de usuários para incentivar a prática de exercícios (tela da Figura 3.5(b)).

O Planfit foi desenvolvido pensando em *personal trainers* e suas necessidades, possuindo menor abrangência que o Emagreca@Saudavel, pois este trata muito da questão motivacional, buscando estimular os usuários através de *rankings* e mensagens motivacionais. Além disso, o Emagreca@Saudavel engloba o público geral, onde qualquer usuário pode desfrutar de suas funcionalidades. Já o Planfit engloba apenas os usuários que são alunos do profissional, pois estes deverão receber treinos personalizados.

Figura 3.4: Imagens referentes ao aplicativo Emagreca@Saudavel 1.



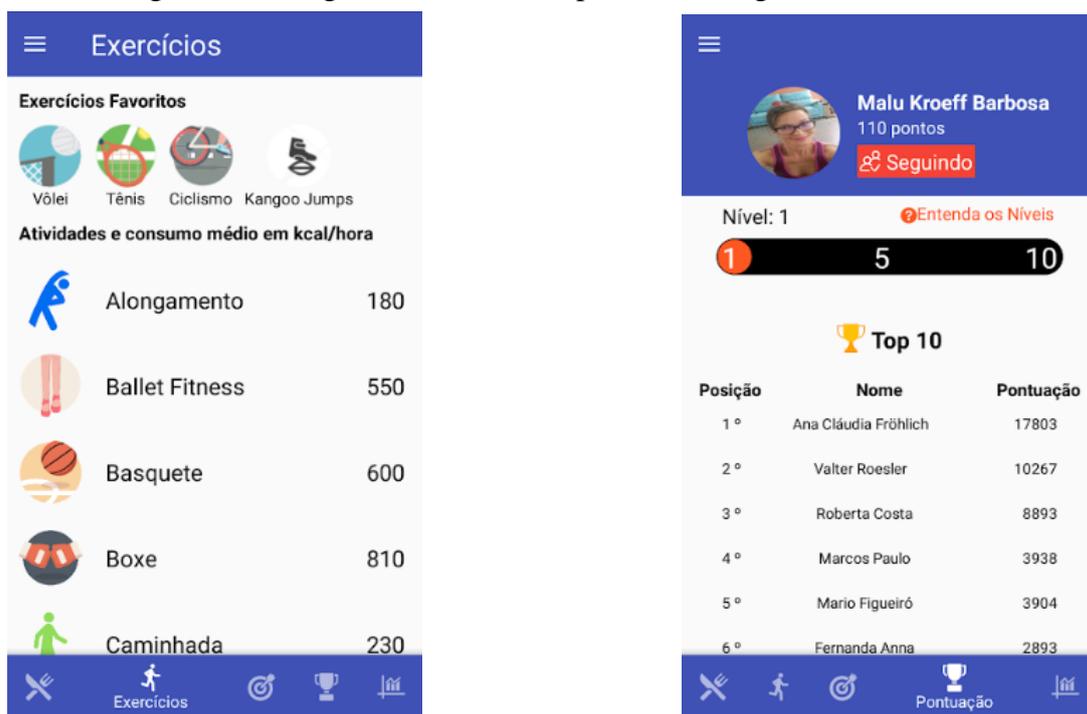
(a) Tela de tarefas do aplicativo Emagreca@Saudavel.



(b) Tela de menu do aplicativo Emagreca@Saudavel.

Fonte: Site <https://play.google.com/store/apps/details?id=br.prav.emagrecasaudavel>

Figura 3.5: Imagens referentes ao aplicativo Emagreca@Saudavel 2.



(a) Tela de exercícios do aplicativo Emagreca@Saudavel.

(b) Tela de ranking do aplicativo Emagreca@Saudavel.

Fonte: Site <https://play.google.com/store/apps/details?id=br.prav.emagrecasaudavel>

3.2 Levantamento de Requisitos Através de Entrevistas

A aplicação desenvolvida possuiu acompanhamento e validação de profissionais da área. Inicialmente foram realizadas entrevistas com dois *personal trainers*, um possuindo centro de treinamento fixo, denominado "profissional 1", e outro cujo atendimento é realizado online e presencialmente, porém sem local fixo, revezando entre academias na cidade, este denominado "profissional 2".

O profissional 1, que possuía centro de treinamento próprio, dispunha de uma estrutura de organização de aulas e alunos utilizando televisores, *tablets* e *smartphones*. O segundo, que não possuía local fixo, se organizava normalmente através de planilhas. Entretanto, o profissional 2 aparentou, através de seus relatos, ter maiores possibilidades de adquirir e utilizar produtos voltados a *personal trainers*. Considerando o estilo de trabalho dinâmico relatado pelo entrevistado e o fato de já realizar consultorias online, seria de maior praticidade, para esse caso, o uso de aplicativos para automatização de treinos e dicas de treinamento personalizadas para seus alunos.

Através das necessidades do profissional 2, foi possível gerar um sistema consti-

tuído de um aplicativo e um servidor *web* capaz de suprir grande parte de suas necessidades, possibilitando o controle e gerenciamento de mais alunos, ou emprego de menos tempo por aluno, conseqüentemente possibilitando um maior rendimento.

O sistema abstraído da entrevista com o profissional 2 possui os módulos da Tabela 3.1 (Informações de perfil ou pessoais), da Tabela 3.2 (Abas do aplicativo), e da Tabela 3.3 (Funcionalidades gerais do menu).

A Tabela 3.1 apresenta algumas informações que o profissional julgou importantes de extrair do usuário. Primeiramente, é necessário ter um cadastro com informações pessoais para permitir ao *personal trainer* a visualização do perfil de cada aluno. Em outro momento, foram definidos os cálculos que o aplicativo deveria exibir ao usuário. Além desses, foi proposto um cadastro de questionários personalizáveis pelo profissional para que seja possível obter informações específicas do usuário do aplicativo.

A Tabela 3.2 mostra os módulos principais do aplicativo. O módulo de tarefas é o principal, pois comporta os exercícios do aluno, de acordo com o dia da semana. Também permite ao usuário registrar o momento de início e fim do treino, além de feedback para o *personal trainer*. Através destes *feedbacks* é possível informar ao *personal trainer* quaisquer sintomas durante a realização do treino (ex. dores musculares, desconforto, etc.). O módulo educação permite ao profissional indicar exercícios, planos de alimentação, massagens, etc. Como o profissional relatou utilizar parcerias com nutricionistas e lojas de suplementos, o módulo de notícias permitirá cadastrar promoções e notícias que possam ser pertinentes ao usuário/cliente. Por último, como o entrevistado sugeriu uma maneira do usuário visualizar sua evolução, a aba evolução permitirá visualização de gráficos de medidas como peso, cintura, e quadril, além de cálculos como IMC (índice de massa corporal) e ICQ (índice de cintura e quadris).

A Tabela 3.3 mostra funcionalidades do menu do aplicativo. O primeiro item do menu possibilita ao usuário visualizar e editar suas informações de perfil. O segundo trata da tela de leitura de mensagens que podem ser enviadas pelo profissional através do servidor *web*. O terceiro item se refere à tela de configurações. O quarto permite ao profissional cadastrar algumas informações de dúvidas sobre a utilização do aplicativo.

Tabela 3.1: Módulos do sistema - Informações de perfil ou pessoais

Módulo	Descrição
Cadastro de usuários	Permite <i>login</i> e cadastro de senha. Também é possível entrar através da rede social Facebook.
Dados do perfil	O usuário poderá informar idade, sexo, peso, altura, escolaridade, nível de consumo alcoólico, se é fumante e se faz uso de medicamentos.
IMC/ICQ	Cálculo automático do IMC e ICQ a ser exibido no cadastro e exibição de perfil.
Questionário	Um teste que poderá ser mostrado ao usuário através de um <i>link</i> da <i>web</i> possibilitando obtenção de informações específicas. Exemplo: Teste WHOQOL (<i>World Health Organization Quality of Life Assessment</i>) (POWER; KUYKEN, 1998)

Tabela 3.2: Módulos do sistema - Abas do aplicativo

Módulo	Descrição
Tarefas	Seção do aplicativo onde são exibidos os treinos aos usuários, de acordo com o dia corrente, com possibilidade de marcação de tarefas concluídas via <i>checkbox</i> , cálculo de tempo de treino, e envio de feedback ao profissional.
Educação	Aba onde será apresentado ao usuário as dicas cadastradas pelo profissional no servidor <i>web</i>
Notícias	Seção onde o profissional poderá cadastrar notícias ou artigos para exibir ao usuário. Também poderá cadastrar <i>links</i> para redirecionar a promoções e patrocínios.
Evolução	seção onde é apresentado ao usuário gráficos de evolução de peso, IMC, cintura, quadril, ICQ.

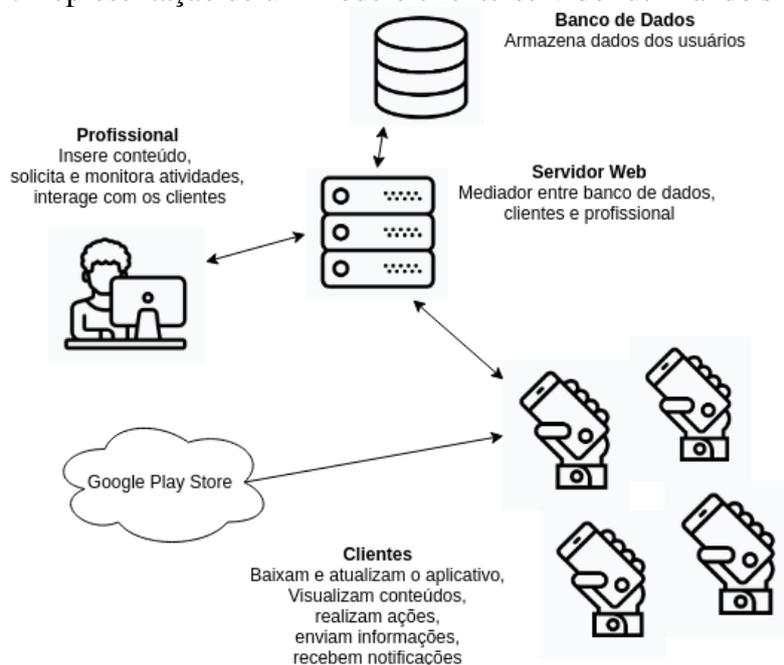
Tabela 3.3: Módulos do sistema - Funcionalidades gerais do menu

Módulo	Descrição
Dados e edição de perfil	Item responsável por permitir ao usuário visualizar e alterar suas informações cadastrais.
Leitura de mensagens	Item responsável por exibir notificações e mensagens ao usuário.
Configurações	Seção do aplicativo onde será possível definir configurações gerais do aplicativo.
F.A.Q	Seção de perguntas frequentes, onde o profissional poderá cadastrar respostas sobre dúvidas de utilização do aplicativo.

4 PLANFIT: O SISTEMA PROPOSTO

O sistema apresentado neste trabalho foi construído com uma arquitetura baseada no modelo cliente-servidor (OLUWATOSIN, 2014), permitindo a troca de informações entre as partes. Para tanto, foi desenvolvido um aplicativo de *smartphone*, para o papel de cliente e um servidor *web* para armazenamento e edição de informações referentes ao aplicativo. Esse servidor é centralizado, tem acesso ao seu banco de dados e está no meio do caminho entre os inúmeros usuários-cliente e o banco de dados global do sistema. A Figura 4.1 mostra uma visão geral da arquitetura do sistema.

Figura 4.1: Representação de um modelo cliente-servidor utilizando *smartphones*



Do ponto de vista do cliente da aplicação móvel, é necessário somente realizar o *download* do aplicativo na plataforma *Google Play Store*, efetuar o cadastro e realizar as atividades atribuídas pelo profissional. Quanto à utilização pelo profissional, este deverá cadastrar e editar as informações que serão transmitidas à aplicação móvel.

A funcionalidade está dividida em duas vias; (i) Sempre que houver alguma alteração no banco de dados, esta alteração será enviada do servidor ao aplicativo; (ii) Sempre que o usuário interagir com a aplicação, esta fica encarregada de repassar as informações ao servidor. Mais detalhes sobre a arquitetura do servidor e do aplicativo estão nas seções 4.1 e 4.2.

4.1 Arquitetura do Servidor *web*

Para armazenamento global de informações e acompanhamento por parte do profissional, foi utilizado um servidor *web*. Sua interface e comunicação foram construídas de forma que o profissional operador não necessite conhecimentos de programação para o gerenciamento. O servidor foi construído complementarmente para a integração com o aplicativo e torná-lo funcional.

Um código para servidor já utilizado no sistema Emagreca@Saudavel (Lúcia Kroeff Barbosa et al., 2018) foi utilizado para a criação do servidor *backend* devido às similaridades das aplicações.

Para o desenvolvimento da aplicação *server-side* (no servidor), foi utilizada a linguagem de programação JAVA, empregando a arquitetura de software MVC (Model View Controller) fazendo uso do framework Java Server Faces (JSF), tecnologia que viabiliza a implementação desta arquitetura e permite uma maior modularidade do projeto.

O servidor Oracle GlassFish Server foi utilizado para hospedagem. Para o gerenciamento do banco de dados foi utilizado o PostgreSQL¹. O servidor utiliza REST para criar uma interface de comunicação que permite a integração com o aplicativo móvel. Os detalhes do protocolo de comunicação estão na seção 4.3.

4.2 Arquitetura do Cliente (Dispositivo Móvel)

Para este trabalho foi escolhido o sistema operacional Android, sistema operacional móvel mais popular do mundo (GILSKI; STEFANSKI, 2015), que possui aproximadamente 76% do *market share* de *smartphones* segundo o site Statcounter²

Outras opções de soluções híbridas como Ionic³ e Xamarin⁴ foram descartadas devido à curva de aprendizado extra e pela dificuldade de compilação de código para IOS, visto a necessidade do ambiente de programação Xcode⁵. Para a programação Android, foi utilizada sua IDE (*Integrated Development Environment*) padrão, o Android Studio⁶.

O aplicativo Planfit se comunica com o servidor através do cliente HTTP Retrofit.

¹PostgreSQL (Acesso em novembro de 2019). <https://www.postgresql.org/>.

²StatCounter (Acesso em novembro de 2019). <https://gs.statcounter.com/os-market-share/mobile/worldwide>

³Ionic (Acesso em novembro de 2019). [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)).

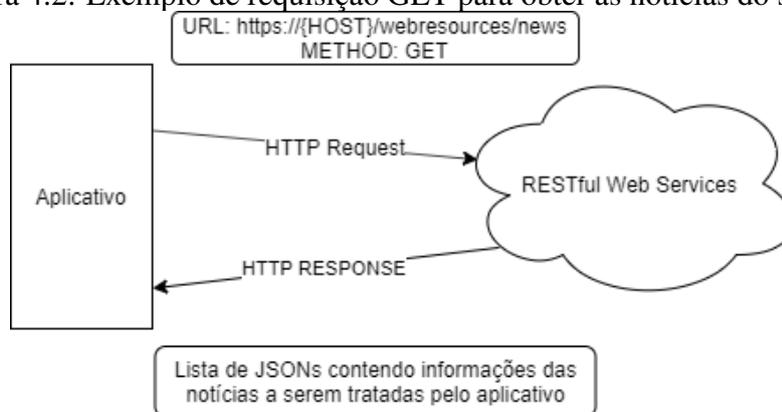
⁴Xamarin (Acesso em novembro de 2019). <https://en.wikipedia.org/wiki/Xamarin>.

⁵Xcode (Acesso em novembro de 2019). <https://pt.wikipedia.org/wiki/Xcode>.

⁶Android Studio (Acesso em novembro de 2019). <https://developer.android.com/studio/>.

Sempre que uma requisição POST, PUT, GET é enviada ao servidor, um arquivo JSON pode ser enviado em anexo. O servidor processa a requisição e retorna outro arquivo JSON de resposta. Um exemplo de uma requisição GET está na Figura 4.2

Figura 4.2: Exemplo de requisição GET para obter as notícias do servidor



Fonte: O autor

Para o banco de dados do aplicativo móvel foi utilizado o Realm, evitando a necessidade de baixar todos os dados do servidor novamente sempre que a aplicação é aberta. Foi criada uma estratégia que permite que os dados sejam buscados novamente no servidor apenas quando há alguma alteração.

Auxiliariamente ao Retrofit, foram utilizadas as bibliotecas OkHttp e Picasso para menor consumo de banda de rede.

O banco de dados Realm, possui *listeners* já implementados para mudanças em seus dados, dispensando o uso de *observers*⁷ para monitoramento de mudanças nos dados. Foi utilizado o padrão de criação Singleton⁸ para recuperar uma instância do banco de dados sempre que necessário, fechando tal instância quando esta não fosse mais utilizada.

Para evitar complicações no desempenho do aplicativo, as requisições HTTP e os acessos ao banco de dados são realizados em plano de fundo, em outras *threads*, permitindo fluidez na interface de usuário. Além disso, é realizado um controle de conexão ativa com o servidor, de modo a evitar perda de informações em caso de desconexão.

Para evitar manipulações de funções de várias camadas (banco de dados, conexão, etc) no código principal de uma *activity*⁹, ou um *fragment*¹⁰, foi utilizado o padrão estrutural Façade¹¹, onde com apenas uma chamada de função, como por exemplo “*downloadNews()*”, é possível baixar todas as notícias do servidor e salvá-las no banco de dados

⁷Observer Pattern (Acesso em dezembro de 2019). https://en.wikipedia.org/wiki/Observer_pattern

⁸Singleton (Acesso em novembro de 2019). https://en.wikipedia.org/wiki/Singleton_pattern.

⁹Activity (Acesso em dezembro de 2019). <https://developer.android.com/reference/android/app/Activity>

¹⁰Fragment (Acesso em dezembro de 2019). <https://developer.android.com/guide/components/fragments>

¹¹Façade (Acesso em novembro de 2019). <https://pt.wikipedia.org/wiki/Façade>.

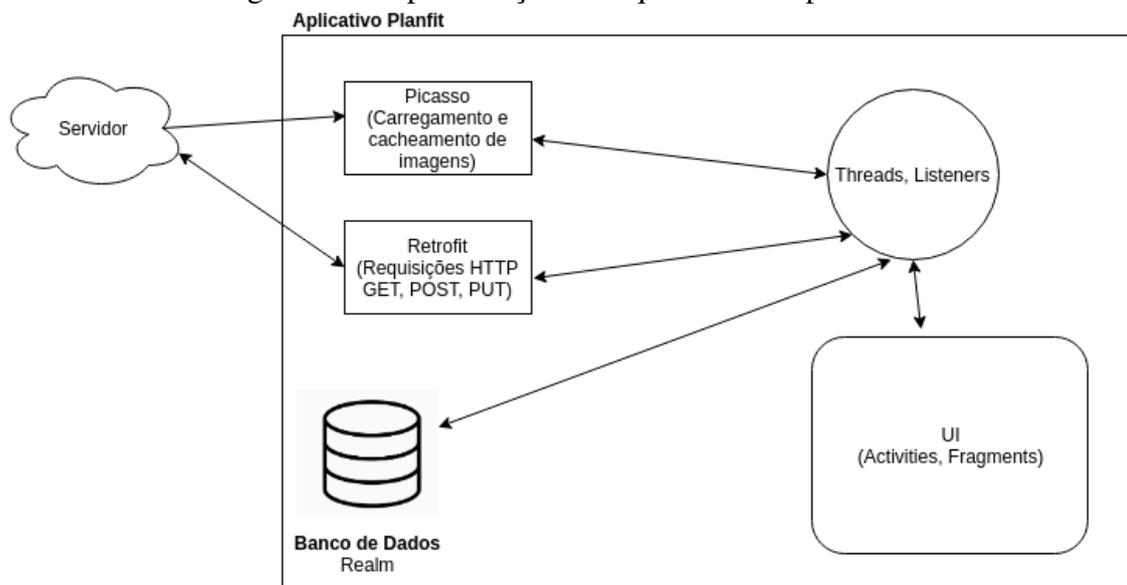
Realm, dando mais modularidade e menos acoplamento ao código.

Outro framework auxiliar utilizado foi o Dagger ¹², que realiza injeção de dependências. Assim, facilita-se a obtenção de objetos necessitados pelas classes, sendo uma alternativa a padrões como o Factory ¹³.

Um meio de rastrear os *crashes* (erros) do aplicativo é através do Crashlytics¹⁴, que é uma biblioteca para tal fim, permitindo assim uma melhor qualidade da aplicação.

Na Figura 4.3 está ilustrado o funcionamento do lado do aplicativo móvel: O servidor envia e recebe dados através da API Retrofit, e também envia imagens através da API Picasso (com auxílio do OkHttp). Ambas APIs se comunicam com *threads* do aparelho para realizar CRUD (*create, read, update, delete*) de informações no banco de dados. A interface de usuário, por fim, exibe os dados e interage com o usuário.

Figura 4.3: Representação da arquitetura do aplicativo



Fonte: O autor

4.3 Detalhamento da comunicação

No início deste capítulo foi descrito que os *web services* utilizados neste trabalho são baseados em REST. Esta seção exibe um detalhamento do protocolo utilizado. A Tabela 4.1 apresenta os Web Services do tipo POST (onde o aplicativo envia informações ao servidor, criando alguma linha nas tabelas do banco de dados). A Tabelas 4.2 e 4.3

¹²Dagger (Acesso em junho de 2019). <https://github.com/google/dagger>.

¹³Factory (Acesso em junho de 2019). https://pt.wikipedia.org/wiki/Factory_Method.

¹⁴Crashlytics (Acesso em novembro de 2019). <https://fabric.io/kits/android/crashlytics/>.

apresentam os *web services* do tipo GET (onde o aplicativo solicita e baixa informações do servidor, lendo alguma informação das tabelas do banco de dados). A Tabela 4.4 apresenta os *web services* do tipo PUT (onde o aplicativo móvel solicita a troca do status de alguma entrada já existente em uma tabela do banco de dados).

Alguns parâmetros são padrão para todos os *web services*, e serão explicados aqui para evitar redundância nas explicações das tabelas. São eles:

- String Date: Data que foi feita a transação no formato dd/mm/aaaa.
- int UserId: Identificador de usuário. Cada usuário existente no sistema possui um UserId único como chave primária.
- String entityType: Identifica o tipo da tabela relacionada à consulta. Por exemplo, pode ser “Tarefa”, “Notícia”, “Educação”, etc.
- int entityId: Identifica a linha da tabela definida em “entityType”. Por exemplo, se o campo entityType = "Tarefa", então o entityId vai indicar especificamente qual tarefa está sendo requisitada. Supondo que existam 100 tarefas, esse identificador explicita qual delas está sendo requisitada.
- int id: Identificador único de uma tabela do banco de dados, sendo utilizado como chave primária no banco de dados.

Procurando deixar mais claro, por exemplo, uma requisição do tipo GET na tabela notícias com o id 30, teria um retorno possível no formato: {"entityId":1, "userId":52, "rating":5, "id": 30, "entityType":"task", "text":"Excelente!", "date":"05/11/2019"}. Isso quer dizer que na data de 05 de novembro de 2019 (campo “date”) o usuário 52 (campo “userId”) enviou um *feedback* na tarefa (campo “entityType”) número 1 (campo “entityId”) com o comentário “Excelente!” (campo “text”) e com avaliação 5 estrelas (campo “rating”). Em outro exemplo, com o id 41, pode-se receber o seguinte: {"id":41, "userId":75, "entityType":"news", "entityId":67, "date":"06/11/2019"}. Isso quer dizer que na data de 06 de novembro de 2019 (campo “date”) o usuário 75 (campo “userId”) completou a Tarefa (campo “entityType=task”) número 67 (campo “entityId”). Essa é a tarefa 41 da tabela de tarefas (campo “id”).

Por questões de negócios futuros, esta seção tem o objetivo não de ser uma documentação completa de uso dos *web services* do sistema, mas sim de mostrá-los de forma geral, e explicar como eles se relacionam.

A Tabela 4.1 abaixo possui os *web services* do tipo POST, que são utilizados para enviar informações do aplicativo para o servidor. Essas informações referem-se ao cadastro (criação de usuário), conclusão de tarefas, inserção de medidas, envio de feedbacks,

entre outras.

Tabela 4.1: Web Services do tipo POST (envio de informações do aplicativo móvel ao servidor)

Tipo	Estrutura
<p>Avaliação de atividades: Envia feedback de uma atividade realizada pelo usuário, que pode ser uma nota de 1 a 5 e/ou um comentário em texto.</p>	<p>O comentário dado pelo usuário vem através do campo “text”, e a nota através do campo “rating”.</p> <p>{int id; String entityType; int entityId; int userId; String text; int rating; String date }</p>
<p>Inserção de medidas: Envia ao servidor uma nova medida cadastrada pelo usuário</p>	<p>Os parâmetros específicos são o peso, o IMC, a cintura, o quadril, o ICQ e a data da medição.</p> <p>{int id; int userId; float weight; float imc; float waist; float hips; float icq; String date }</p>
<p>Tarefa concluída: Envia ao servidor a informação de uma tarefa que foi concluída</p>	<p>entityType é enviado como padrão "task", indicando que é uma tarefa.</p> <p>{int id; int userId; String entityType; int entityId }</p>
<p>Cadastro de novo usuário: Quando o usuário é novo ele é cadastrado no servidor para posteriormente realizar <i>login</i></p>	<p>A maioria dos parâmetros é opcional, como medicamentos ou telefone.</p> <p>{int id; String email; String password; String name; String birthDate; String gender; String scholarship; String medicines; String alcoholConsumption; boolean smokes; float height; float weight; float hips; float waist; float imc; float icq; String phone }</p>
<p>Cadastro de imagem de usuário: Se o usuário não possui imagem e inseri-la, esta será cadastrada.</p>	<p>Será enviado um id para a imagem, o id do usuário referente a aquela imagem, e a imagem decodificada em String.</p> <p>{int id; int userId; String image }</p>

Fonte: O autor

A tabela 4.2 descreve *web services* do tipo GET com conteúdo criado pelo administrador do sistema, como as categorias de educação, ou notícias.

Tabela 4.2: Web Services do tipo GET para todos os usuário(recebimento de informações do servidor no aplicativo móvel)

Tipo	Estrutura
Categorias de educação: Recebe a lista de categorias de educação.	Cada categoria de educação tem uma imagem própria e um texto descrevendo a mesma. {int id; String name; String image; String description;}
Subcategorias de educação: Recebe a lista de subcategorias de educação.	As subcategorias de educação estão relacionadas à uma categoria de educação. Possuem texto de descrição e imagem ou <i>link</i> de vídeo. <i>Category</i> indica o id da categoria vinculada. {int id; String name; String image; String video; int category; String description }
Gerais/Outros: Recebe a lista de Gerais/Outros. Esta lista é utilizada para configurações adicionais, como o link do questionário, ou o conteúdo do FAQ.	A lista de Gerais/Outros vem com um título indicando a que se refere, e uma descrição como conteúdo. Esta lista é tratada no aplicativo. {int id; String name; String description }
Notícias: Recebe a lista de Notícias.	A lista de notícias vem com um título, um ícone, uma imagem, uma descrição, e um <i>link</i> caso a notícia possua link externo. {int id; String icon; String name; String image; String description, String link }
Mensagens: Recebe a lista de mensagens não lidas	O campo <i>createdAt</i> se refere à data de criação da mensagem em <i>timestamp</i> . {int id; String message; long createdAt }

Fonte: O autor

A tabela 4.3 descreve *web services* do tipo GET exclusivos por usuário (acessados

através de seu identificador).

Tabela 4.3: Web Services do tipo GET por usuário (recebimento de informações do servidor no aplicativo móvel)

Tipo	Estrutura
<p>Treino: Recebe a informação da configuração do treino do usuário.</p>	<p>Cada treino possui um conjunto diferente de versões de tarefas por dia da semana, o que permite o controle para baixar as tarefas apenas se a versão mudar.</p> <pre>{ int id; String title; String description; int stage; int domVersion; int segVersion; int terVersion; int quaVersion; int quiVersion; int sexVersion; int sabVersion; int timeDom; int timeSeg; int timeTer; int timeQua; int timeQui; int timeSex; int timeSab; }</pre>
<p>Tarefas do treino: Recebe as tarefas relacionadas ao treino do usuário por dia da semana.</p>	<p>Cada tarefa possui título, descrição, imagem, ícone, link de vídeo, e informação de dias da semana que ela está associada, além do tipo (se imagem, vídeo ou outro).</p> <pre>{ int id; String icon; String image; String type; String video; String title; String description; String linkEntity; String weekDay; }</pre>
<p>Tarefas concluídas: Recebe as informações de quais tarefas foram concluídas pelo usuário, assim como o dia da semana.</p>	<p>Com as tarefas concluídas é possível recuperar os dados do treino já realizado caso o usuário apague os dados do aplicativo.</p> <pre>{ int id; int userId; String weekDay; int entityId }</pre>
<p>Download de perfil: Recebe o perfil do usuário, juntamente com seus dados. É utilizado no login.</p>	<p>Traz todas as informações cadastrais do usuário.</p> <pre>{ int id; String email; String password; String name; String birthDate; String gender; String scholarship; String medicines; String alcoholConsumption; boolean smokes; float height; float weight; float hips; float waist; float imc; float icq; String phone }</pre>
<p>Download de imagem: Baixa a imagem do usuário no servidor.</p>	<p>Traz uma String contendo a imagem decodificada, assim como o identificador do usuário relativo à imagem.</p> <pre>{ int id; int userId; String image }</pre>
<p>Medidas: Recebe a lista de medidas já cadastradas pelo usuário</p>	<p>Os parâmetros são o peso, o IMC, a cintura, o quadril, o ICQ e a data de cada medida.</p> <pre>{ int id; int userId; float weight; float imc; float waist; float hips; float icq; String date }</pre>

Fonte: O autor

A tabela 4.4 abaixo descreve *web services* do tipo PUT, que são requisições de atualizações de dados já existentes no servidor. Alguns exemplos são a atualização de informações de perfil, da sua própria imagem, ou de medidas.

Tabela 4.4: Web Services do tipo PUT (envio de informações do aplicativo ao servidor)

Tipo	Estrutura
<p>Envio de medidas: Envia as medidas cadastradas pelo usuário no aplicativo para o servidor, atualizando-as e plotando o gráfico.</p>	<p>Os parâmetros específicos são o peso, o IMC, a cintura, o quadril, o ICQ e a data. {int id; int userId; float weight; float imc; float waist; float hips; float icq; String date }</p>
<p>Atualização do perfil do usuário: Envia o perfil atualizado do usuário para o servidor, caso algo seja alterado.</p>	<p>Os parâmetros são os mesmos enviados no cadastro e opcionais. {int id; String email; String password; String name; String birthDate; String gender; String scholarship; String medicines; String alcoholConsumption; boolean smokes; float height; float weight; float hips; float waist; float imc; float icq; String phone }</p>
<p>Atualização da imagem do usuário no servidor: Caso o usuário altere sua imagem, envia a nova ao servidor.</p>	<p>Envia a imagem decodificada em String, além do identificador do usuário. {int id; int userId; String image }</p>
<p>Leitura de mensagem: Assim que o usuário ler uma mensagem, a informação de lida será atualizada no servidor.</p>	<p>Identifica a mensagem que deve ter o <i>status</i> de lida alterado. O parâmetro <i>createdAt</i> se refere à data de criação da mensagem. {int id; String message; long createdAt }</p>

Fonte: O autor

5 DETALHAMENTO DA IMPLEMENTAÇÃO

Este capítulo contém informações mais detalhadas sobre a implementação e seu resultado, tanto para o aplicativo quanto para o servidor.

5.1 Implementação do servidor

Para cadastrar informações ou visualizar o conteúdo cadastrado no servidor é necessário inicialmente realizar *login* (Figura 5.1)

Figura 5.1: Tela de *login*.



PlanFit

Efetue o login para acessar o sistema:

Usuário: planfitadmin

Senha:

Entrar

Fonte: O autor

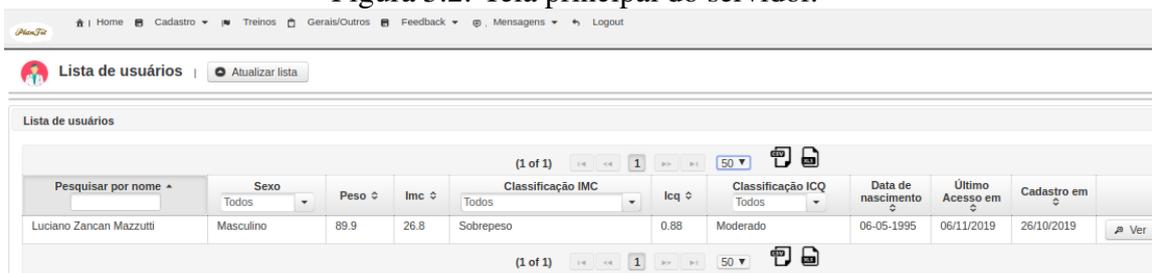
O servidor foi idealizado para permitir que alguém sem conhecimento de programação possa fazer alterações no conteúdo a ser mostrado no aplicativo móvel. Possui várias seções que, dependendo do conteúdo cadastrado, "moldam" o aplicativo. Logo após realizar o *login*, a tela da Figura 5.2 é exibida ao profissional, onde é possível visualizar uma lista de usuários, com um resumo e opção de visualizar mais detalhes separadamente ao clicar no ícone "ver", à direita de cada usuário.

Algumas abas na parte superior possibilitam navegar para as abas referentes ao cadastro de outras informações, são elas:

Cadastro, onde é possível cadastrar, editar e visualizar lista de tarefas, notícias, educação e subcategorias de educação;

- Treinos, onde é possível criar e editar treinos semanais dos usuários;
- Gerais/Outros, onde é possível cadastrar informações para a seção de perguntas frequentes e *link* para questionário;
- Feedback, onde é possível visualizar feedbacks de tarefas e treinos;
- Mensagens, onde é possível enviar mensagens aos usuários.

Figura 5.2: Tela principal do servidor.



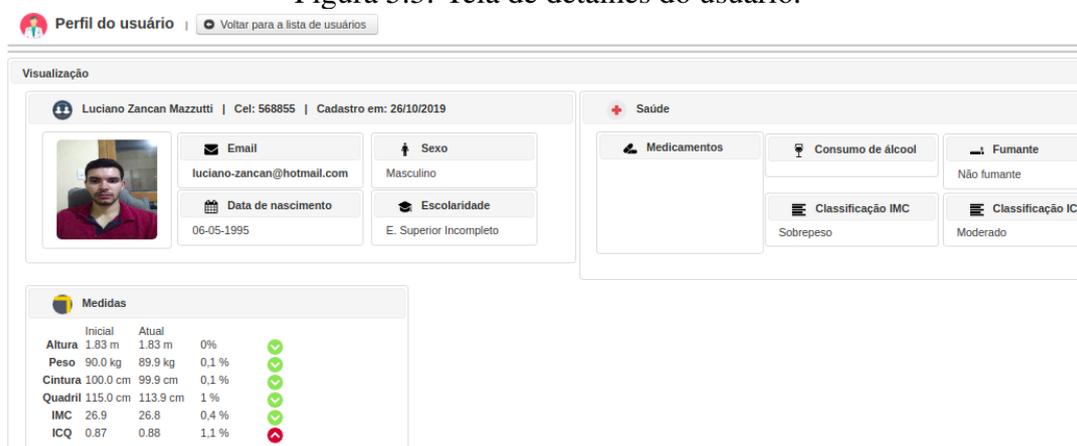
The screenshot shows a web application interface with a navigation menu at the top: Home, Cadastro, Treinos, Gerais/Outros, Feedback, Mensagens, and Logout. Below the menu is a header for 'Lista de usuários' with an 'Atualizar lista' button. The main content area is titled 'Lista de usuários' and contains a table with search filters and a list of users. The table has columns for 'Pesquisar por nome', 'Sexo', 'Peso', 'Imc', 'Classificação IMC', 'Icq', 'Classificação ICQ', 'Data de nascimento', 'Último Acesso em', and 'Cadastro em'. A single user, Luciano Zancan Mazzutti, is listed with the following details: Masculino, 89.9, 26.8, Sobrepeso, 0.88, Moderado, 06-05-1995, 06/11/2019, and 26/10/2019. A 'Ver' button is next to the user's name.

Pesquisar por nome	Sexo	Peso	Imc	Classificação IMC	Icq	Classificação ICQ	Data de nascimento	Último Acesso em	Cadastro em	
Luciano Zancan Mazzutti	Masculino	89.9	26.8	Sobrepeso	0.88	Moderado	06-05-1995	06/11/2019	26/10/2019	Ver

Fonte: O autor

O profissional pode visualizar mais detalhes sobre um usuário clicando no botão "Ver". Ao clicar neste botão, a tela da Figura 5.3 é aberta, mostrando as informações do usuário como medidas, informações de contato, informações de saúde, gráficos de evolução (Figura 5.4), e tarefas concluídas (Figura 5.5).

Figura 5.3: Tela de detalhes do usuário.

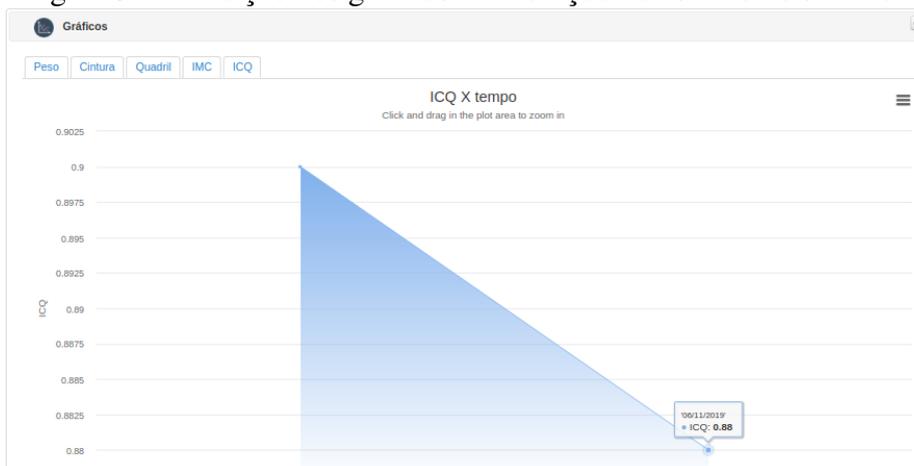


The screenshot shows the 'Perfil do usuário' page for Luciano Zancan Mazzutti. It includes a profile picture, contact information (Email: luciano-zancan@hotmail.com, Cel: 568855, Cadastro em: 26/10/2019), and a 'Saúde' section with sub-sections for Medicamentos, Consumo de álcool, Fumante, Classificação IMC (Sobrepeso), and Classificação ICQ (Moderado). A 'Medidas' section shows a comparison of initial and current values for various metrics, with green checkmarks indicating stability or improvement and a red X indicating a decrease.

	Inicial	Atual		
Altura	1.83 m	1.83 m	0%	✓
Peso	90.0 kg	89.9 kg	0,1 %	✓
Cintura	100.0 cm	99.9 cm	0,1 %	✓
Quadril	115.0 cm	113.9 cm	1 %	✓
IMC	26.9	26.8	0,4 %	✓
ICQ	0.87	0.88	1,1 %	✗

Fonte: O autor

Figura 5.4: Exibição dos gráficos de evolução do usuário no servidor.



Fonte: O autor

Figura 5.5: Exibição das tarefas concluídas pelo usuário selecionado.

Extratos

Tarefas

Titulo	Data
Agachamento 15x3	'06/11/2019'
Treino de pernas	'06/11/2019'
Bicicleta 20 minutos	'06/11/2019'
Esteira 10 minutos - 8km/h	'06/11/2019'
Agachamento 15x3	'06/11/2019'
Treino de costas	'05/11/2019'
Esteira 10 minutos - 8km/h	'05/11/2019'

Fonte: O autor

Todas as abas de cadastro possuem uma lista referente ao seu contexto, além de possuir opções de visualização, edição, exclusão, ou criação de uma nova entidade. A aba de cadastro de tarefas está na Figura 5.6, cadastro de notícias na Figura 5.7, cadastro de categorias de educação na Figura 5.8, e por último a de cadastro de subcategorias de educação na Figura 5.9.

Figura 5.6: Lista de tarefas cadastradas.

Lista de Tarefas | Cadastrar nova tarefa

Lista

(1 of 1) 1 50

Pesquisar tarefa	Tipo de Midia	Rating	
 Agachamento 15x3	image	0.0	Visualizar Editar Excluir
 Bicicleta 20 minutos	image	0.0	Visualizar Editar Excluir
 Esteira 10 minutos - 8km/h	image	0.0	Visualizar Editar Excluir
 Treino de braços	image	5.0	Visualizar Editar Excluir
 Treino de costas	video	0.0	Visualizar Editar Excluir
 Treino de peitoral	image	0.0	Visualizar Editar Excluir
 Treino de pernas	image	0.0	Visualizar Editar Excluir

(1 of 1) 1 50

Fonte: O autor

Figura 5.7: Lista de notícias cadastradas.

Lista de Noticias | Cadastrar nova Noticia

Lista

(1 of 1) 1 50

Pesquisar noticia	
 Aulão Outubro Rosa	Visualizar Editar Excluir
 Camisetas para academia - Netshoes	Visualizar Editar Excluir
 Circuito Poa Day Run - 2019	Visualizar Editar Excluir
 Comprar marmitas fit	Visualizar Editar Excluir
 Cupom de desconto suplementos	Visualizar Editar Excluir
 Curta VidaNova Academia e ganhe brinde	Visualizar Editar Excluir
 Nova unidade Bodytech	Visualizar Editar Excluir
 Novo grupo whatsapp	Visualizar Editar Excluir
 Novos shakes sabor baunilha e morango	Visualizar Editar Excluir

(1 of 1) 1 50

Fonte: O autor

Figura 5.8: Lista categorias de educação cadastradas.

The screenshot shows a web interface for 'Categorias de Educação'. At the top, there is a search bar and a button 'Cadastrar nova Categoria'. Below this is a table with the following structure:

Pesquisar categoria -	Descrição	Opções
<p>Dicas de alimentação</p> 	Esta categoria contém dicas de alimentação	Visualizar Editar Excluir
<p>Dicas de exercícios</p> 	Esta categoria contém dicas de exercícios	Visualizar Editar Excluir

Fonte: O autor

Figura 5.9: Lista de subcategorias de educação cadastradas.

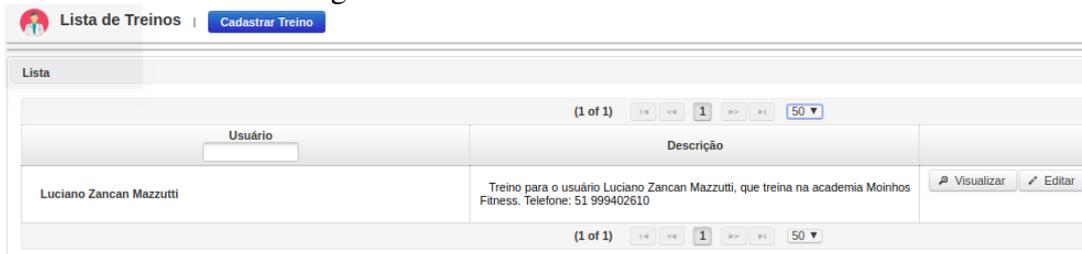
The screenshot shows a web interface for 'Subcategorias de Educação'. At the top, there is a search bar and a button 'Cadastrar nova Subcategoria'. Below this is a table with the following structure:

Pesquisar subcategorias -	Categoria	Opções
7 Dicas para iniciar na musculação	Dicas de exercícios	Visualizar Editar Excluir
Dica de treino na esteira	Dicas de exercícios	Visualizar Editar Excluir
Dicas de treino de pernas	Dicas de exercícios	Visualizar Editar Excluir
Dicas - O que comer?	Dicas de alimentação	Visualizar Editar Excluir
Massagem antiestresse	Dicas de massagem	Visualizar Editar Excluir
Maximizando bíceps e tríceps	Dicas de exercícios	Visualizar Editar Excluir
Melhore seu supino	Dicas de exercícios	Visualizar Editar Excluir
O que comer após o treino?	Dicas de alimentação	Visualizar Editar Excluir
Tutorial Reiki	Dicas de massagem	Visualizar Editar Excluir

Fonte: O autor

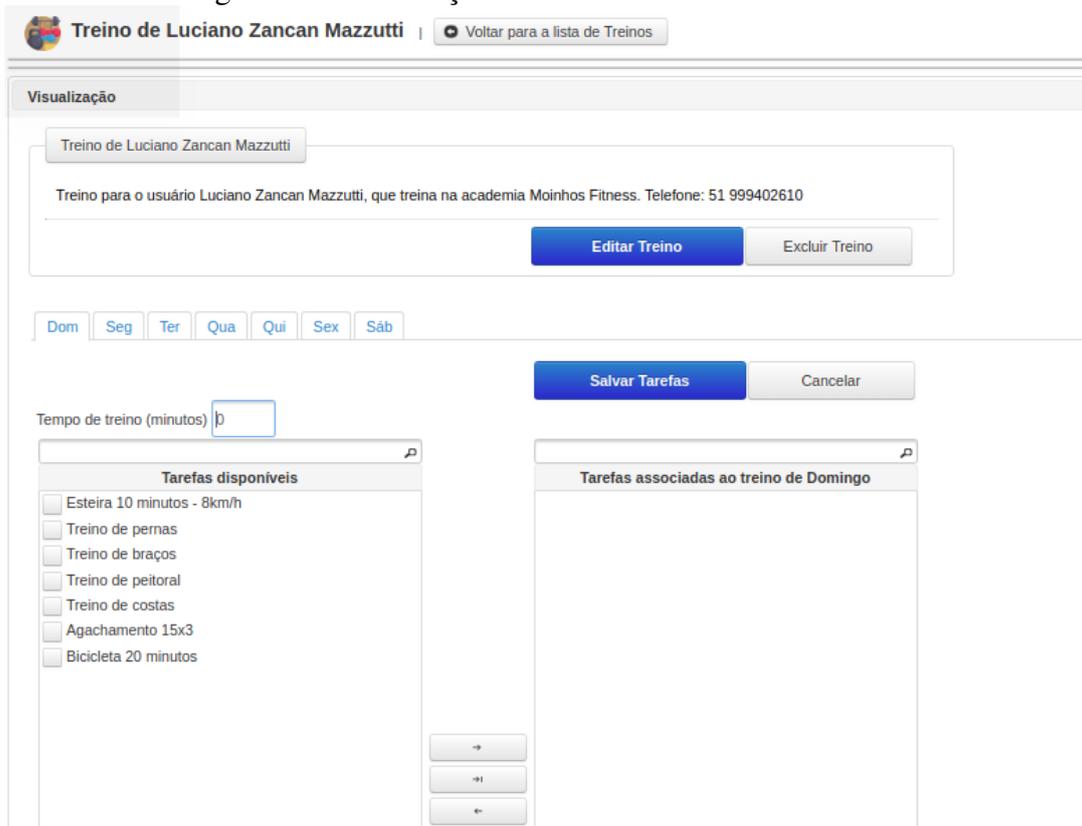
Ao lado do botão de cadastro está o botão de treinos. Ao clicar neste botão o profissional entra em uma tela onde é exibida uma lista de treinos já cadastrados (Figura 5.10, além da opção de cadastrar um novo treino, e atribuir tarefas e minutos ao treino (Figura 5.11).

Figura 5.10: Lista treinos de usuários.



Fonte: O autor

Figura 5.11: Atribuição de treino semanal a um usuário.



Fonte: O autor

Na Figura 5.12 está a tela de cadastro, visualização e edição de entidades do tipo "Gerais/Outros". Atualmente existem duas opções de entidades ("faq", onde é possível cadastrar um texto em formato HTML ¹, e "questionario", onde é possível cadastrar um *link* para o questionário). O aplicativo trata de traduzir as informações contidas nessas entidades, e exibi-las ao usuário.

¹HTML (Acesso em novembro de 2019). <https://pt.wikipedia.org/wiki/HTML>.

Figura 5.12: Lista de entidades do tipo "Gerais/Outros".

Id do Geral/Outro	Name	Description
3	questionario	<p>https://forms.gle/iZsenimxD93jm4ez7</p>
4	faq	<p>1. Como realizo meus treinos no aplicativo?</p><p>Primeiramente é necessário clicar em iniciar no treino referente ao dia desejado. Assim que iniciado, você pode realizar as atividades e marcá-las. Quando acabar, clique em parar, e envie um feedback indicando como foi o treino, se há algo para mudar, como você se sentiu, etc.</p>2.<strong style='color: rgb(0, 0, 0);><p>Posso fazer um treino de outro dia antes?</p><p>Sim, se você preferir realizar o treino de sexta na quarta, e o de quarta na sexta (por exemplo), é possível. Basta iniciar o treino correspondente ao dia desejado. Porém todos os treinos renovam a cada semana. É possível apenas realizar o mesmo treino uma vez por semana.</p></p>

Fonte: O autor

A seção de *feedback* mostra ao *personal trainer* as avaliações de tarefas e treinos enviados pelo usuário do aplicativo móvel. A partir desta tela é possível verificar se há necessidade de intervenção junto ao usuário de acordo com seu *feedback*. A Figura 5.13 mostra a lista de *feedbacks* relativos às tarefas, enquanto a Figura 5.14 mostra a lista referente aos treinos.

Figura 5.13: Lista de *feedbacks* de tarefas.

Informações	Comentário	Ranking	Data
Luciano Zancan Mazzutti comentou na Tarefa: Treino de braços Visualizar Tarefa	Luciano Zancan Mazzutti senti pegar bem o braço 06/11/2019	5	06/11/2019 Excluir

Fonte: O autor

Figura 5.14: Lista de *feedbacks* de treinos.

Comentário	Data
Luciano Zancan Mazzutti Foi bom, cansei bastante. Um pouco puxado, talvez na próxima poderia ser um pouco mais leve. Só tive um probleminha, meu pé estava com dor	06/11/2019 Excluir

Fonte: O autor

Por último, há a seção de mensagens (Figura 5.15), onde é possível selecionar um usuário, múltiplos usuários, ou todos os usuários através de *checkboxes*. Também é possível visualizar quais usuários leram cada mensagem.

Figura 5.15: Seção de envio de mensagens aos usuários.

Enviar nova mensagem | Ir para a lista de mensagens

Cadastro

B I U S

Confira uma promoção nova na aba Notícias!!!

Enviar mensagem

Selecionar todos usuários

Pesquisar por nome ^

Luciano Zancan Mazzutti

Fonte: O autor

5.2 Implementação do aplicativo móvel

Assim que o usuário instalar e abrir o aplicativo pela primeira vez, é necessário realizar um cadastro. A Figura 5.16(b) exibe a tela de cadastro de usuário. Também é possível cadastrar pelo Facebook na tela de *login* (Figura 5.16(a)).

Figura 5.16: Imagens referentes às telas de Cadastro de Usuários.

(a) Tela de Login do aplicativo. Entrar com o Facebook também cadastra o usuário automaticamente.

(b) Tela de Cadastro do aplicativo. Para habilitar o cadastro o usuário deverá concordar com os termos de uso.

Fonte: O autor

Assim que realizar o cadastro ou *login* com a rede social Facebook, o usuário deverá responder algumas perguntas para ser possível acompanhar sua evolução através dos dados coletados. Essas telas exibem perguntas como peso, idade, informação de escolaridade, se fuma, se utiliza remédios, etc.

Inicialmente, como apresentado na Figura 5.17, são requeridos ao usuário dados como sexo e data de nascimento, que posteriormente serão utilizados para cálculos como ICQ.

Figura 5.17: Tela inicial do cadastro e atualização de dados de perfil.

Fonte: O autor

Para os cálculos de IMC e ICQ, existem duas telas. A primeira requisita peso e altura e apresenta o cálculo de IMC em tempo real ao usuário (Figura 5.18(a)). A segunda requisita os dados de cintura e quadril, para cálculo e exibição em tempo real de seu ICQ (Figura 5.18(b)).

Figura 5.18: Imagens referentes às telas de preenchimento de dados e cálculo de IMC e ICQ.

Selecione seu Peso e Altura

70 9
71 , 0 Quilos
72 1

0 70
1 , 71 Metros
2 72

IMC = 24.3
Peso Normal

VOLTAR AVANÇAR

(a) Tela de preenchimento dos dados do perfil. O cálculo do IMC automático é exibido ao usuário.

Selecione sua Cintura e Quadril

94 9
95 , 0 Cintura (cm)
96 1

104 9
105 , 0 Quadril (cm)
106 1

ICQ = 0.90
Alto

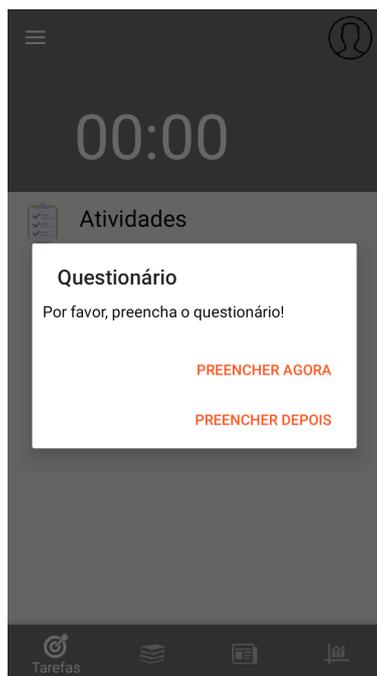
VOLTAR AVANÇAR

(b) Tela de preenchimento dos dados do perfil. O cálculo do ICQ automático é exibido ao usuário.

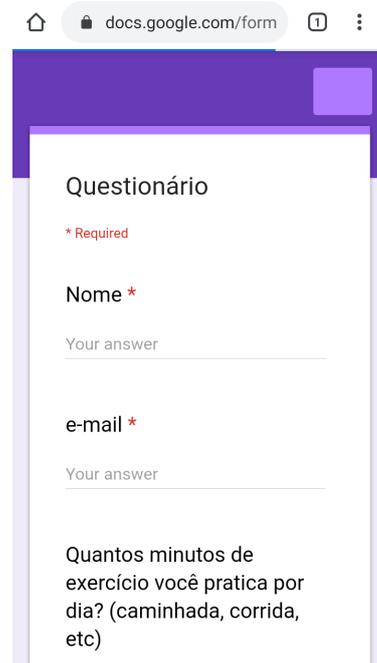
Fonte: O autor

As informações de perfil são informações quantitativas e generalistas. Para a obtenção de informações específicas e qualitativas, o profissional pode cadastrar um questionário externo. Neste caso é exibido ao usuário a pergunta de preenchimento de questionário (Figura 5.19(a)) que pode redirecioná-lo ao *link* do questionário através do navegador (Figura 5.19(b)).

Figura 5.19: Imagens referentes à apresentação do questionário ao usuário.



(a) Apresentação do questionário ao usuário novo.



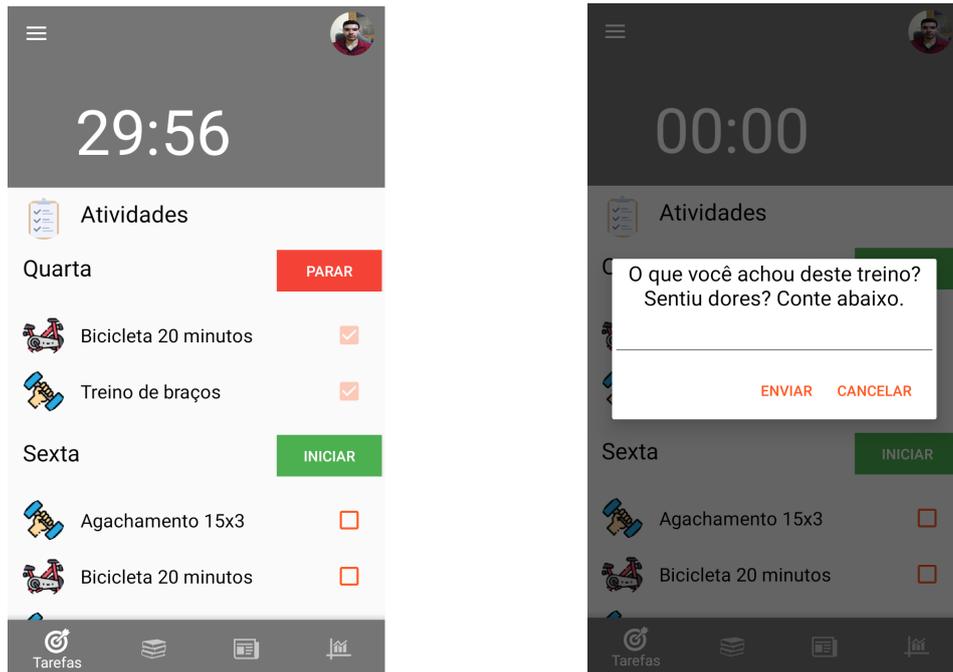
(b) Questionário aberto em navegador.

Fonte: O autor

Sempre que o usuário abrir o aplicativo será exibida a tela "tarefas" contendo os treinos atribuídos pelo *personal trainer* no servidor. Esta é a seção principal do aplicativo móvel, pois é onde o usuário interage com uma lista circular de tarefas de acordo com o dia em questão. Cada tarefa pode ser aberta em outra tela para exibir detalhes sobre o que deve ser realizado. Assim que o treino estiver iniciado (clcando no botão iniciar) é possível marcar tarefas como concluídas (clcando no *checkbox*). O tempo de treino transcorrido desde o início fica exibido em um contador na parte superior da tela.

Assim que o usuário terminar o treino, este pode clicar em parar para finalizar o contador e abrir a tela de *feedback*. Assim, é possível retornar ao profissional a percepção de esforço após o treino, além de informações como desconfortos durante a realização do treino. As Figuras 5.20(a) e 5.20(b) mostram respectivamente a tela de tarefas e a exibição do *feedback* do treino para preenchimento.

Figura 5.20: Imagens referentes à tela de tarefas.



(a) Aba tarefas com lista circular de atividades por dia da semana.

(b) Exibição do preenchimento do *feedback* ao usuário.

Fonte: O autor

Cada tarefa contém um título, texto, e imagem ou vídeo. Na Figura 5.21 é possível visualizar um exemplo de tarefa com imagem. Também observa-se as estrelas, um sistema de notas onde o usuário pode enviar um feedback indicando se gostou da tarefa, se é muito desgastante, etc.

Figura 5.21: Tarefa contendo imagem.



Fonte: O autor

Outra aba do aplicativo refere-se à seção de educação. Esta seção exibe conteúdo

cadastrado pelo profissional no servidor *web*. Na aba educação, poderão ser exibidos imagens, textos ou vídeos ao usuário, e qualquer alteração realizada no servidor será alterada no aplicativo. A Figura 5.22(a) exibe a lista de categorias de educação, e a Figura 5.22(b) exibe a lista de subcategorias de educação.

Figura 5.22: Imagens referentes à tela de subcategorias de educação.



(a) Lista de categorias de educação



(b) Lista de subcategorias de educação.

Fonte: O autor

Cada categoria de educação exibe uma imagem, texto, e uma lista de subcategoria de educação. Cada subcategoria pode exibir imagem, vídeo e texto. Um exemplo contendo vídeo está na Figura 5.23.

Figura 5.23: Subcategoria de educação contendo vídeo.



Massagem antiestresse

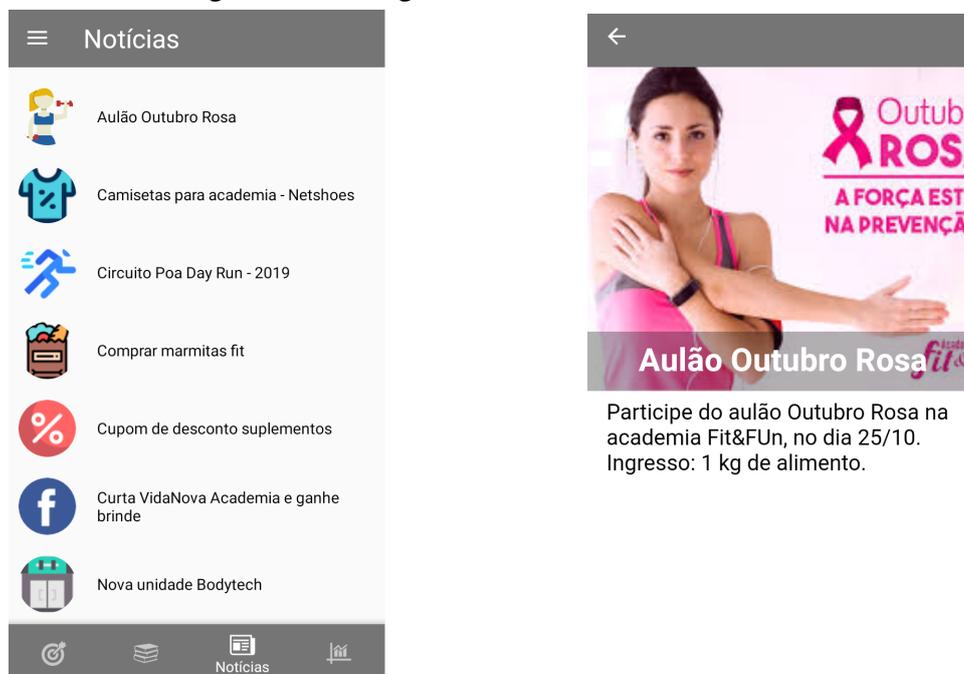
Veja no vídeo como realizar uma massagem antiestresse

Fonte: O autor

A seção de notícias é abrangente e flexível. Caso o profissional realize o cadastro

manual das notícias, estas serão exibidas de maneira semelhante à aba educação. Caso cadastre apenas um *link* de notícia, esta será aberta no navegador do *smartphone*. Na Figura 5.24(a) está a exibição da tela de notícias, e na Figura 5.24(b) está a tela de exibição de uma notícia cadastrada manualmente no servidor (com título, texto e imagem).

Figura 5.24: Imagens referentes à tela de notícias.



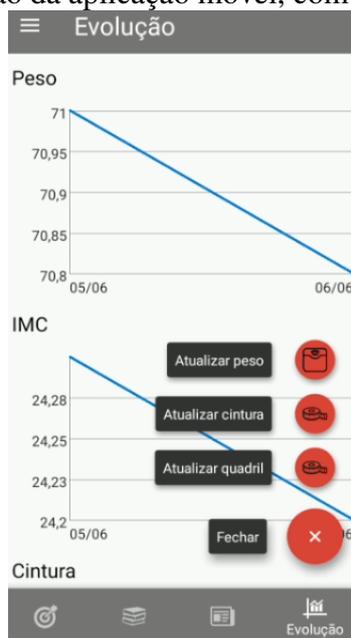
(a) Lista de notícias

(b) Notícia sem *link*.

Fonte: O autor

A última aba da esquerda para a direita é a de evolução. Esta seção é destinada à mostra de gráficos de evolução ao usuário da aplicação móvel. Nesta aba o usuário poderá inserir novas medidas diariamente ou semanalmente para acompanhamento do profissional. Sempre que uma nova medida é inserida, esta é atualizada também no gráfico do servidor. A Figura 5.25 mostra a tela de evolução.

Figura 5.25: Tela de evolução da aplicação móvel, com botões de adição de medidas

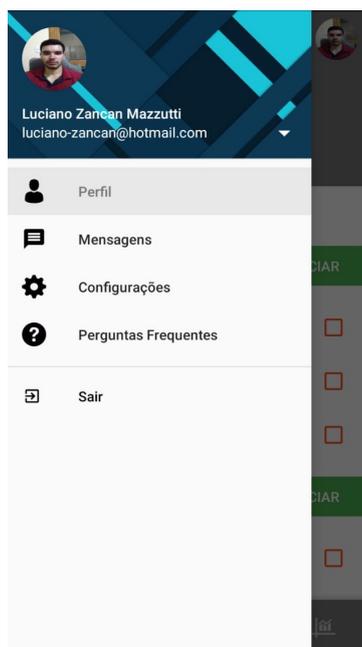


Fonte: O autor

Em todas as telas principais do aplicativo está disponível o acesso ao menu. Caso o usuário toque no ícone com três traços no topo à esquerda, é aberto o menu ao usuário. A Figura 5.26(a) mostra todas as opções do menu: Acesso à tela de perfil, de mensagens, de configurações, de perguntas frequentes, além da opção de sair do aplicativo.

A Figura 5.26(b) mostra a tela aberta ao tocar em "Perfil". Nesta tela, o usuário consegue visualizar suas medidas iniciais e atuais. Também, ao tocar no ícone do lápis no fundo à direita, é possível alterar a imagem do perfil, assim como abrir as mesmas telas exibidas no cadastro de usuário para edição de informações cadastrais.

Figura 5.26: Imagens referentes ao menu e perfil.



(a) Exibição do menu.

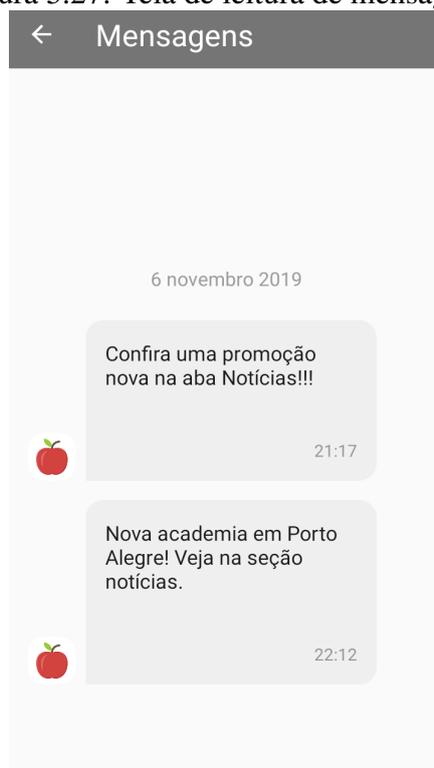


(b) Tela que possibilita a visualização do perfil do usuário.

Fonte: O autor

Caso o profissional envie uma mensagem através do servidor, o aplicativo exibirá uma notificação ao usuário, como na Figura 5.28(a). Ao tocar na notificação, ou ao selecionar a opção "Mensagens" no menu, a tela de mensagens (Figura 5.27) é aberta. Além da notificação de mensagem, existe o lembrete para inserção de medidas (Figura 5.28(b)), que caso tocado abrirá a tela de evolução.

Figura 5.27: Tela de leitura de mensagens.

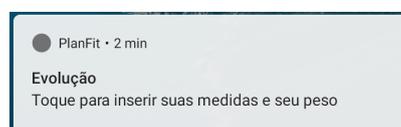


Fonte: O autor

Figura 5.28: Imagens referentes às notificações.



(a) Notificação de nova mensagem.

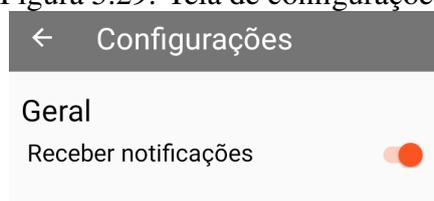


(b) Notificação semanal de inserção de medidas

Fonte: O autor

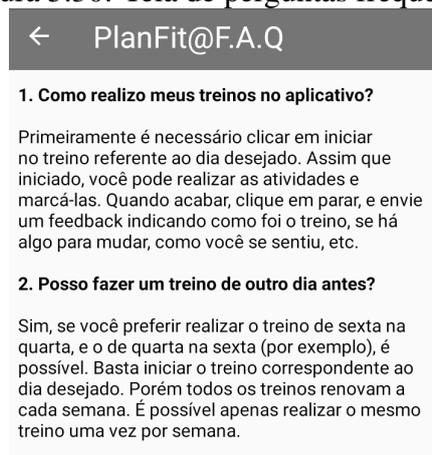
O item configurações do menu (Figura 5.29) permite ao usuário definir preferência sobre recebimento de notificações, e o item de perguntas frequentes (FAQ-*Frequent Asked Questions*) (Figura 5.30) contém perguntas cadastradas pelo profissional no servidor na seção "gerais/outros", a fim de esclarecer algumas dúvidas recorrentes sobre o funcionamento do aplicativo.

Figura 5.29: Tela de configurações.



Fonte: O autor

Figura 5.30: Tela de perguntas frequentes.



Fonte: O autor

6 VALIDAÇÃO DO SISTEMA

Este capítulo trata da validação do sistema desenvolvido neste trabalho. Nas seções a seguir são expostos uma avaliação de quais requisitos foram implementados e uma entrevista de validação.

6.1 Avaliação de Requisitos

A Tabela 6.1 mostra os itens implementados através do levantamento de requisitos do capítulo 3.

Tabela 6.1: Avaliação de Requisitos do Sistema

Módulo	Indicativo
Cadastro de usuários	Implementado
Dados do perfil	Implementado
IMC/ICQ	Implementado
Questionário	Implementado
Tarefas	Implementado
Educação	Implementado
Notícias	Implementado
Evolução	Implementado
Dados e edição de perfil	Implementado
Leitura de mensagens	Implementado
Configurações	Implementado
F.A.Q	Implementado

6.2 Entrevista de validação

Após a conclusão do sistema, com o servidor *web* e o aplicativo móvel finalizados e integrados, foi realizada uma mostra do funcionamento ao profissional (*personal trai-*

ner) da entrevista de definição do sistema. Este profissional foi o responsável por gerar a impressão inicial das necessidades do sistema (o mesmo "profissional 2" do capítulo 3).

Como o sistema foi pensado nos relatos do *personal trainer* entrevistado, houve uma grande aceitação por parte do profissional, demonstrando entusiasmo para iniciar a utilização com seus alunos. "Isso tem um potencial enorme, vai ajudar muito meu trabalho, posso ter mais alunos, e mesmo com os atuais já vai dar um 'up' no meu trabalho", afirmou o entrevistado.

Sobre as informações já captadas pelo aplicativo (informações de perfil), o profissional pontuou a importância de informações como escolaridade e medicamentos. De acordo com ele, "qualquer informação é boa pois ajuda a saber o perfil do aluno". Também mencionou sobre o IMC: "O que seria interessante é encaixar um questionário chamado PARQ, que funciona para grandes populações, assim como o IMC que já tem e é para grandes populações também, é melhor ter alguma informação do que nenhuma".

Sobre as configurações do aplicativo, que atualmente só possui a opção de exibição de notificações, o entrevistado mencionou: "Configurações o aluno não vai ter muito o que configurar, talvez o tamanho da letra". Além disso, o profissional relatou a necessidade de uma aba de parceiros no menu, separada da aba de notícias: "É interessante ter uma aba separada de parceiros, mostrando os parceiros".

O profissional prezou a tela de tarefas, porém sugeriu adaptações para sua rotina de trabalho: "Os dias da semana deveriam se trocados por treinos A, B, C... Ou aeróbico, neuromuscular, funcional... Senão o aluno se obriga sempre a treinar, por exemplo, peito na segunda-feira". Também mencionou que seria interessante possuir, por exemplo, informações como "na sexta-feira o aluno fez o treino C, e não obrigá-lo a realizar o treino C toda sexta-feira".

No geral, o *personal trainer* aprovou o sistema, solicitando que, se possível, fosse contatado após a disponibilização do sistema.

Quanto ao trabalho, as dicas sugeridas pelo *personal trainer* foram de grande valia, visto que se adaptam muito mais às suas necessidades, e provavelmente a necessidades de outros profissionais da área, objetivo principal deste trabalho.

7 CONCLUSÕES

Este trabalho visou a criação de um sistema chamado Planfit, que é voltado a *personal trainers*. Esse sistema é composto por uma aplicação para *smartphones* e uma aplicação *web*, que deverão ser utilizados para diminuir o tempo "*face-to-face*" entre profissionais da área de *personal trainers* e clientes. Foi objetivo deste trabalho, também, melhorar o acompanhamento de cada aluno pelo *personal trainer*, devido à disponibilidade online de cada atividade realizada, *feedbacks* de treinos, e de gráficos de evolução, entre outras coisas. Com a diminuição do tempo dispendido por cada atendimento, espera-se que haja um aumento no número de atendimentos e conseqüente maior rentabilidade.

O aplicativo fruto deste trabalho foi construído para *smartphones* com o sistema operacional Android a ser utilizado por alunos de um profissional de *personal trainer*, e o servidor *web* possui uma interface que possibilita ao profissional inserir conteúdo, como dicas de exercícios, notícias, treinos em forma de lista de tarefas. Além disso, o *personal* poderá acompanhar a evolução dos alunos, além de alterar os conteúdos para cada aluno, permitindo um treino personalizado por usuário.

De acordo com a validação, foi possível perceber que o sistema oferece recursos necessários ao profissional, visto que o entrevistado aprovou a primeira versão apresentada.

Perspectivas envolvem alterações sugeridas pelo profissional entrevistado, além da disponibilização do aplicativo na loja *Play Store* do *Google*, e hospedagem do servidor para disponibilização *online*, visto que todos os testes foram realizados em rede local por falta de infraestrutura para hospedagem.

REFERÊNCIAS

BASHSHUR, R. L. et al. Sustaining and Realizing the Promise of Telemedicine. **Telemedicine and e-Health**, 2013. ISSN 1530-5627.

BERGMO, T. S. **How to measure costs and benefits of ehealth interventions: An overview of methods and frameworks**. 2015.

BHOSALE, S. T.; PATIL, M. T.; PATIL, M. P. SQLite: Light Database System. **International Journal of Computer Science and Mobile Computing**, 2015.

CAR, J.; SHEIKH, A. Email consultations in health care: 2—acceptability and safe application. **BMJ**, 2004. ISSN 14685833.

EYSENBACH, G. What is e-health? **Journal of Medical Internet Research**, 2001. ISSN 14388871.

GILSKI, P.; STEFANSKI, J. Android OS: A Review. **TEM Journal**, 2015.

JENNETT, P. A. et al. **The socio-economic impact of telehealth: A systematic review**. 2003.

KATZ, S. J. et al. Effect of a triage-based E-mail system on clinic resource use and patient and physician satisfaction in primary care: a randomized controlled trial. **Journal of general internal medicine**, 2003. ISSN 0884-8734.

OLUWATOSIN, H. S. Client-Server Model. **IOSR Journal of Computer Engineering**, 2014. ISSN 22788727.

OWENS, D. K. et al. High-value, cost-conscious health care: Concepts for clinicians to evaluate the benefits, harms, and costs of medical interventions. **Annals of Internal Medicine**, 2011. ISSN 15393704.

POWER, M.; KUYKEN, W. World Health Organization Quality of Life Assessment (WHOQOL): Development and general psychometric properties. **Social Science and Medicine**, 1998. ISSN 02779536.

SOARES, M. A. Aplicativo móvel para academia [manuscrito] : estudo de tecnologias e desenvolvimento. **Biblioteca IFMG Campus Sabará**, 2016.

VOGEL, L.; SCHOLZ, S.; WEISER, D. **Using Retrofit 2.x as REST client**. 2017.

Lúcia Kroeff Barbosa, M. et al. Uma solução m-Health para apoio à educação em saúde com foco na mudança de comportamento para hábitos saudáveis. **RENOTE**, 2018. ISSN 1679-1916.