

UNIVERSIDADE FEDERAL DO PAMPA

Cristiane Gonçalves Pinheiro

**Aplicativo para Recomendação de
Treinos e Agendamento de Aulas**

Alegrete
2019

Cristiane Gonçalves Pinheiro

**Aplicativo para Recomendação de
Treinos e Agendamento de Aulas**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Claudio Schepke

Alegrete
2019

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

P654a Pinheiro, Cristiane Gonçalves

Aplicativo para Recomendação de Treinos e Agendamento de
Aulas / Cristiane Gonçalves Pinheiro.

95 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade
Federal do Pampa, CIÊNCIA DA COMPUTAÇÃO, 2019.

"Orientação: Claudio Schepke".

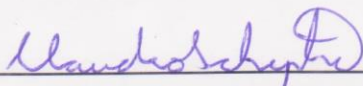
1. Aplicativo. 2. Academias. 3. Agendamento de Aulas. 4.
Recomendação de Treinos. 5. Controle de Desempenho. I. Título.

Cristiane Gonçalves Pinheiro

Aplicativo para Recomendação de Treinos e Agendamento de Aulas


Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 25 de novembro de 2019.
Banca examinadora:



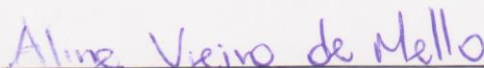
Prof. Dr. Claudio Schepke

Orientador
UNIPAMPA



Prof. Dr. Fábio Paulo Basso

UNIPAMPA



Profa. Dra. Aline Viera de Mello

UNIPAMPA

Este trabalho é dedicado aos meus pais,
por sempre acreditar no meu potencial
e contribuir com essa conquista.
Amo vocês mais que tudo.

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter iluminado o meu caminho. Aos meus pais, Ana e Alnei pelo apoio, incentivo e amor incondicional. Sem vocês a realização desse sonho não seria possível. Aos meus amigos, em especial o Nathan, obrigada pelos conselhos, frases de motivação e pelas horas de descontração que tornaram os trabalhos mais divertidos. Ao professor Claudio, pela orientação, apoio e confiança. E ao meu namorado, Bruno pelo apoio, carinho e incentivo.

“São as nossas escolhas,
mais do que as nossas capacidades,
que mostram quem realmente somos.”
(Alvo Dumbledore, Harry Potter e a Câmara Secreta).

RESUMO

As pessoas muitas vezes não possuem tempo para praticar exercícios físicos devido a uma rotina agitada. Por esse motivo, aplicativos de celular podem ser utilizados como suporte à prática de atividades físicas e no acompanhamento das mesmas. Nesse sentido, o objetivo deste trabalho é prover uma implementação de um aplicativo para agendamento de aulas e notificação de atividades. Além disso, a aplicação possibilita a visualização de treinos e desempenho do aluno. Para o desenvolvimento do trabalho foram inicialmente realizadas pesquisas bibliográficas com o propósito de analisar os aplicativos disponíveis ou encontrados na literatura científica, além de entrevistas para o levantamento de requisitos funcionais. Os conceitos utilizados para a implementação abordam temas como banco de dados, Linguagem Unificada de Modelagem (UML) e descrição das ferramentas utilizadas. A implementação proposta relaciona academias, *personais* e alunos, buscando flexibilizar o agendamento de aulas. Ao final, foi feita uma avaliação do software em duas academias, analisando usabilidade e se o mesmo está em conformidade com a proposta anunciada. Os resultados mostraram que embora poucos testes tenham sido realizados, o aplicativo cumpre com o objetivo.

Palavras-chave: Aplicativo. Academias. Notificação *Push*. Android. Agendamento. Controle de Desempenho.

ABSTRACT

People often do not have time to exercise due to a hectic routine. For this reason, mobile apps can be used to support and monitor physical activity. In this sense, the objective of this work is to provide an implementation of an application for class scheduling and activity notification. In addition, the application enables the visualization of training and student performance. For the development of the work were initially conducted bibliographical research with the purpose of analyzing the applications available or found in the scientific literature, as well as interviews for the survey of functional requirements. The concepts used for implementation address topics such as database, ac uml and description of the tools used. The proposed implementation relates academies, textit personais and students, seeking to make the scheduling of classes more flexible. At the end, an evaluation of the software was made in two gyms, analyzing usability and if it is in accordance with the announced proposal. The results showed that although few tests have been performed, the application fulfills the goal.

Key-words: App. Gym. Push Notification. Android. Scheduling. Performance Control.

LISTA DE FIGURAS

Figura 1 – Esquematização das categorias de funcionalidades encontradas nos trabalhos relacionados	34
Figura 2 – Caso de Uso Aluno	47
Figura 3 – Caso de Uso Personal	48
Figura 4 – Caso de Uso Academia	49
Figura 5 – Caso de Uso Aluno não Cadastrado	49
Figura 6 – Diagrama de Classes	50
Figura 7 – Tela Inicial do App	50
Figura 8 – Telas de <i>Logins</i>	51
Figura 9 – Telas de Menu	51
Figura 10 – Telas de Menu	52
Figura 11 – Funcionalidades da Academia	53
Figura 12 – Funcionalidades do Personal	54
Figura 13 – Funcionalidades do Personal ao Selecionar um Aluno	55
Figura 14 – Funcionalidades do Personal ao Selecionar um Aluno	56
Figura 15 – Funcionalidades do Aluno	57
Figura 16 – Funcionalidades do Aluno não Cadastrado e do Aluno	58
Figura 17 – Tela Inicial	59
Figura 18 – Login Aluno / Menu Aluno	60
Figura 19 – Visualizar Treinos / Exibir Exercícios	60
Figura 20 – Listar Avaliações / Exibir Desempenho	61
Figura 21 – Exibir Mapa	62
Figura 22 – Listar Aulas / Agendar Aula	62
Figura 23 – Login Personal / Menu Aluno	63
Figura 24 – Exibir Alunos	64
Figura 25 – Formulário de Cadastro para Alunos	64
Figura 26 – Pop-up	65
Figura 27 – Cadastrar Treino	65
Figura 28 – Formulário de Cadastro Treinos	66
Figura 29 – Cadastro de Exercícios	66
Figura 30 – Formulário de Cadastro Exercicios	67
Figura 31 – Cadastrar Avaliacao	67
Figura 32 – Formulário Cadastro Avaliacao	68
Figura 33 – Cadastrar Medidas	68
Figura 34 – Formulário Cadastro Medidas Corporais	69
Figura 35 – Cadastro Aulas / Formulario Aulas	69
Figura 36 – Cadastro Academias	70
Figura 37 – Login Academia	71

Figura 38 – Menu Academia	71
Figura 39 – Exibir Personal	72
Figura 40 – Cadastrar Personal	72
Figura 41 – Exibir Modalidades	73
Figura 42 – Formulário Cadastro Modalidades	73
Figura 43 – Menu	74
Figura 44 – Notificação	74
Figura 45 – Respostas da Pergunta 1	75
Figura 46 – Respostas da Pergunta 1	76
Figura 47 – Formulário Aplicado nas Entrevistas	88
Figura 48 – Formulário Aplicado na Validação	90
Figura 49 – Formulário Aplicado na Validação	91
Figura 50 – Formulário Aplicado na Validação	92
Figura 51 – Formulário Aplicado na Validação	93

LISTA DE TABELAS

Tabela 1 – Trabalhos Relacionados	32
Tabela 2 – Trabalhos Relacionados	33

LISTA DE ABREVIATURAS

App. Aplicativos Móveis

LISTA DE SIGLAS

API Interface de Programação de Aplicativos

BaaS Backend as a Service

CSS Cascading Style Sheets

FCM Firebase Cloud Messaging

HTML HyperText Markup Language

IDE Ambiente de Desenvolvimento Integrado

IMC Índice de Massa Corporal

JSP Java Server Page

NeoDatis ODB NeoDatis Object Data Base

PHP Hypertext Preprocessor

SaaS Software as a Service

SGBD Sistema de Gerenciamento de Banco de Dados

SGCA Sistema de Gerenciamento e Controle para Academias

UML Linguagem Unificada de Modelagem

XML *Extensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Justificativa	26
1.2	Organização deste trabalho	26
2	REFERENCIAL TEÓRICO	27
2.1	Dispositivos Móveis	27
2.2	Android	27
2.3	Aplicativos Móveis	28
2.4	Banco de Dados	28
2.5	UML	29
2.6	Notificação Push	29
2.6.1	Firebase Cloud Messaging	29
2.7	Geolocalização	30
2.8	Google Interface de Programação de Aplicativos (API) Maps	30
3	TRABALHOS RELACIONADOS E AVALIADOS	31
3.1	4GYM	35
3.2	Sistema para Academias	35
3.3	Sistema de Gerenciamento de Academias de Ginástica	35
3.4	Diário de um Marombeiro	36
3.5	FitDolphin	36
3.6	SGCA	36
3.7	SharpGym	37
3.8	SISACAD	37
3.9	FitGen	38
3.10	TecnoFit	38
3.11	DroidFitness	38
3.12	Sistema de Acompanhamento	39
3.13	Sistema de Gerenciamento para Academia	39
3.14	Gobody Fitness	39
3.15	SysAcademy	39
3.16	Sistema SCA	40
3.17	Gympass	40
3.18	Nossa Proposta	40
3.19	Análise dos Aplicativos Relacionados	41
4	METODOLOGIA	43
4.1	Elicitação de Requisitos e Proposta de Validação	43
4.2	Abordagem de Implementação	43

4.2.1	Ferramenta de Desenvolvimento - Android Studio	44
4.2.2	Ferramenta Prototipação - Justinmind Prototyper	44
4.2.3	Ferramenta Diagramação - Enterprise Architect	45
4.3	Firestore	45
4.4	Diagramas de Classes e Estudo de Caso	46
4.5	Protótipos das Telas	46
4.6	Considerações do Capítulo	47
5	RESULTADOS EXPERIMENTAIS	59
5.1	Implementação	59
5.2	Validação	75
6	CONSIDERAÇÕES FINAIS	79
	REFERÊNCIAS	81
	APÊNDICES	85
	APÊNDICE A – QUESTIONÁRIO APLICADO NAS ENTREVISTAS	87
	APÊNDICE B – QUESTIONÁRIO AVALIATIVO DO APLICATIVO	89
	Índice	95

1 INTRODUÇÃO

Aplicativos Móveis (App.)’s são softwares desenvolvidos para celulares, tablets, ou qualquer dispositivo móvel. Segundo Silva, Pires e Neto (2015) “os aplicativos móveis (App.’s) são produtos projetados e desenvolvidos para serem executados especificamente em dispositivos eletrônicos móveis”. Os aplicativos podem ser classificados em categorias, tais como: produtividade, lazer, saúde, entre outras. É possível utilizá-los para facilitar as tarefas cotidianas, como organizar e agendar compromissos, além do entretenimento. Os aplicativos vinculam as necessidades das pessoas em um ambiente de fácil interação, usual e móvel. Com isso, seu uso está cada vez mais disseminado e aperfeiçoado.

Atualmente as pessoas estão cada vez mais ocupadas com as tarefas do cotidiano, adquirindo uma má qualidade de vida.

A necessidade de descarregar o estresse adquirido pelo dia a dia movimentado, aliado a busca por uma vida mais saudável, faz muitas pessoas realizarem exercícios. As academias surgem com uma boa estratégia para esta finalidade, ou seja, quem deseja reduzir o estresse ou mesmo se exercitar na busca de um corpo mais saudável (BENTO et al., 1994, 2).

Com uma rotina agitada, muitas vezes as pessoas não conseguem dedicar um período de tempo à prática de exercícios físicos. Algumas matriculam-se em academias, mas não conseguem dar continuidade a rotina de exercícios, levando a desistência. Um dos motivos do abandono é o fato do usuário não poder acompanhar seus rendimentos, ocasionando um desânimo por parte do mesmo. Outro motivo é a conciliação entre os horários das atividades disponibilizadas pelas academias com seu tempo livre. “A associação das atividades em academias com o acompanhamento dos alunos, visando as informações das evoluções e metas relacionados aos objetivos iniciais, são de suma importância no desenvolvimento físico e mental do aluno” (BENTO et al., 1994).

Diante do exposto, buscou-se analisar e coletar dados com a finalidade de responder as seguintes questões de pesquisa: como um aplicativo pode auxiliar a encontrar um horário ideal para praticar exercícios? Como um aplicativo para controle de desempenho em academias pode incentivar alunos a realizar atividades físicas? Um aplicativo dinâmico, com notificação, capaz de adaptar-se as necessidades, além de sugerir e permitir agendar aulas, pode ser uma boa alternativa.

O presente trabalho tem como objetivo prover uma implementação de um aplicativo para recomendar aulas em academias. Para tanto, é proposto que o aplicativo possibilite a inclusão e o agendamento de atividades em academias. Também foi feita uma análise de como o aplicativo beneficia a qualidade dos serviços oferecidos por academias com base em estudos de aplicativos similares.

1.1 Justificativa

As pessoas em geral possuem algum tempo “livre” durante o dia que poderia ser melhor aproveitado através de um aplicativo que sugerisse aulas disponíveis em diferentes horários e academias por meio de lembretes, com o intuito de avisar o usuário que ele poderia estar praticando alguma atividade física. Isso pode ser feito através do desenvolvimento de um aplicativo para dispositivos móveis que possibilite notificações do tipo *push* de aulas em academias.

1.2 Organização deste trabalho

O trabalho encontra-se dividido em 6 capítulos. No Capítulo 2 são apresentados os conceitos necessários para o embasamento do trabalho. Os trabalhos relacionados são comentados no Capítulo 3. No Capítulo 4 é discutida a metodologia de desenvolvimento, os testes adotados, além da apresentação das ferramentas utilizadas no trabalho. No Capítulo 5 são mostrados os resultados obtidos através da avaliação da aplicação em academias. Já no Capítulo 6 são apresentadas as considerações finais e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo são descritos os referenciais teóricos usados no desenvolvimento deste trabalho.

2.1 Dispositivos Móveis

Segundo EduMobile (2017), dispositivos móveis são tecnologias que permitem um fácil e rápido deslocamento e acesso à internet. “Um dispositivo móvel é essencialmente um computador de bolso (ou portátil) geralmente equipado com um pequeno dispositivo de saída ou ecrã (*output*, em inglês) e um teclado de entrada de informações miniatura (*input*, em inglês)” (MOLINARI, 2018). Uma outra definição diz que um dispositivo móvel é “qualquer peça de equipamento eletrônico, como um celular ou um pequeno computador que você possa usar em lugares diferentes” (CAMBRIDGEDICTIONARY, 2018b).

Alguns exemplos de dispositivos são *smartphones*, *tablets* e *notebooks*. Estes dispositivos apresentam características físicas como: tela pequena, execução de tarefas a partir de toque na tela, etc. Essas características são usadas para a conectividade.

Todo dispositivo móvel possui um sistema operacional, para o qual são desenvolvidos aplicativos. Alguns exemplos de sistemas operacionais para dispositivos móveis são: Android, iOS, Windows Phone e Hongmeng.

2.2 Android

Segundo Gomes et al. (2012), Android é um sistema operacional baseado no kernel do Linux. Inicialmente ele foi desenvolvido para *smartphones* (BLÄSING et al., 2010). Atualmente seu uso expandiu-se para diversos produtos como *tablets*, relógios, etc. Android é uma máquina virtual Java rodando sobre o kernel do Linux, oferecendo suporte para o desenvolvimento de aplicações Java através de bibliotecas e serviços.

De acordo com Filho (2017), o Android surgiu em 2003, no Vale do Silício. Na época o nome era Android Inc. Em 2005 o Google comprou o Android Inc e em 2007 quando a Apple lançou o iPhone ocorreu a revolução na forma como as pessoas usavam os celulares.

Embora baseado no kernel Linux, Android possui propriedades exclusivas e limitações específicas devido a sua natureza móvel, o que dificulta a detecção e a reação a ataques de *malware*. De acordo com PEREIRA e SILVA (2009), isso se deve ao fato do sistema ser executado com o kernel Linux. Sempre que um aplicativo é instalado no sistema operacional Android, um novo usuário Linux é adicionado para o aplicativo em questão. Devido ao fato de os aplicativos ficarem isolados, qualquer tentativa de acesso as informações dos demais aplicativos é preciso uma autorização do usuário.

Segundo Burton e Felker (2014), Android tem uma ampla participação no mercado, visto que existem mais lojas de aplicativos, o que populariza a busca pelos mesmos. A API

de desenvolvimento para Android facilita a geração de aplicativos com inúmeros recursos em um curto prazo de tempo.

Outro motivo para o sucesso do sistema operacional é por ele ser uma plataforma aberta e possuir uma compatibilidade cruzada, ou seja, ele pode ser executado em dispositivos com tamanhos de telas variáveis e resoluções diferentes.

2.3 Aplicativos Móveis

Aplicativos Móveis são *softwares*. Um aplicativo móvel “é um programa de computador ou software desenvolvido para uma finalidade específica que você pode baixar em um celular ou outro dispositivo móvel” (CAMBRIDGEDICTIONARY, 2018a). “Para fornecer uma analogia, as aplicações são para os telefones móveis, programas são para computadores” (CUELLO; VITTONI, 2013).

Segundo Cuello e Vittone (2013), atualmente existem inúmeros aplicativos, cada um com suas características próprias. Porém no início os aplicativos eram desenvolvidos para melhorar a produtividade pessoal. De acordo com Lecheta (2016), atualmente mais de 3 bilhões de pessoas possuem um celular, e o mercado de App.'s rende bilhões de reais todos os anos.

2.4 Banco de Dados

Segundo Date (2004), banco de dados são sistemas para gerenciamento de registros ou informações. Um banco de dados é um conjunto de arquivos de dados armazenados “computacionalmente”, onde é possível realizar algumas tarefas como atualização, exclusão e inclusão de informações. “Um banco de dados é uma coleção de dados, tipicamente descrevendo o relacionamento entre uma ou mais entidades. ”(RAMAKRISHNAN; GEHRKE, 2000).

De acordo com Date (2004), os bancos de dados podem ser classificados em relacionais e não-relacionais. Os sistemas relacionais organizam os registros ou dados em forma de tabelas. Em contrapartida os sistemas não-relacionais manipulam outras estruturas de dados. Estruturas em forma de árvores representam, por exemplo, hierarquias.

Para usuários realizarem operações no banco de dados existe uma camada de *software*, denominada Sistema de Gerenciamento de Banco de Dados (SGBD). “SGBD é o *software* que trata de todo o acesso ao banco de dados.” (DATE, 2004). Alguns exemplos de SGBD's são Oracle, MySQL, SQL Server, PostgreSQL, DB2, SQLite, Firebase entre outros.

2.5 UML

Segundo Booch, Jacobson e Rumbaugh (2017), a UML, é uma linguagem gráfica para visualizar, especificar e documentar artefatos de *softwares*. UML proporciona um padrão para modelar arquiteturas de projetos de *softwares*, incluindo aspectos conceituais como funções do sistemas e itens concretos como classes, banco de dados, etc.

UML “é uma linguagem visual para modelar sistemas orientados a objetos” (BEZERRA, 2016). Estes representam os conceitos de paradigma da orientação a objetos. Essa representação ocorre no formato de diagramas. Esses diagramas são denominados artefatos que compõem a visão do sistema. Alguns exemplos de diagramas são: caso de uso, classes, sequência, etc. Segundo MELO (2010), os diagramas de caso de uso expressam as funcionalidades do sistemas; os diagramas de classe exibem entidades do mundo real e os diagramas de sequência exibem as interações correspondentes às mensagens trocadas entre objetos.

2.6 Notificação Push

Segundo AndroidDevelopers (2019), notificações são mensagens exibidas quando o aplicativo está fechado ou em segundo plano, com o objetivo de oferecer aos usuários lembretes, solicitações de mensagens ou informações importantes ou relevantes do aplicativo. A partir da notificação recebida é possível executar o aplicativo ou realizar ações. “As notificações permitem alertar o usuário que algum evento aconteceu mesmo que ele não esteja utilizando a aplicação ou até mesmo o aparelho no momento.” (GLAUBER, 2019). Ainda segundo Glauber (2019), a utilização de notificações de forma correta, resulta em diversos benefícios como o aumento da participação dos usuários.

As notificações podem ser do tipo *pull* ou do tipo *push*. De acordo com Santos (2016), notificações *pull* emitem uma solicitação ao servidor, que recebe a solicitação, processa e a entrega ao usuário. Sempre que uma nova solicitação for emitida o processo é repetido. As notificações *push* estabelecem ligações entre o usuário e o servidor. Esta ligação é efetuada apenas uma vez, definida durante a instalação do aplicativo e ficando permanente ativada.

2.6.1 Firebase Cloud Messaging

O Firebase Cloud Messaging (FCM) é um serviço integrado do *Firebase* que permite o envio de mensagens de notificações e envio de dados. O envio de mensagens do FCM aos usuários pode ser de dois tipos: mensagem de notificação e mensagem de dados (FIREBASE, 2019)

“O FCM é um serviço de plataforma cruzada que manipula o envio, o roteamento e a fila de mensagens entre aplicativos de servidor e aplicativos cliente móveis.” (MI-

CROSOFTDOCS, 2018). A partir do FCM é possível enviar notificações para um único usuário, para grupos de usuários ou enviar para diversos usuários sobre um mesmo tópico.

2.7 Geolocalização

“A geolocalização define as coordenadas geográficas de um telefone celular ou de um computador” (SPADARI, 2016). A localização pode ser obtida através de antenas, internet e o próprio GPS do dispositivo.

A partir da geolocalização é possível localizar lugares de interesse ou próximos do usuário, traçar rotas e indicar como está o trânsito. É possível também recomendar locais a partir da posição atual.

2.8 Google API Maps

Google oferece um conjunto amplo de API's para o desenvolvimento de programas. Cada API “é um conjunto de interfaces de programação de aplicativos desenvolvido pela Google, que permite a comunicação com as aplicações da empresa e sua integração a serviços como: Gmail, Google Maps” (DUARTE, 2019).

Segundo Douglas (2013), Google maps API é um serviço aberto onde é possível integrá-lo em qualquer aplicação. A API Google Maps permite a visualização de mapas reais, podendo estes serem personalizados com a utilização de marcação, além de exibir a localização atual do dispositivo no mapa.

O Google Maps possui também API's complementares envolvendo rotas, lugares e informações do trânsito, podendo-se, desta forma, aprimorar as aplicações. Para usar os serviços oferecidos por estas API's é necessário efetuar um pagamento.

3 TRABALHOS RELACIONADOS E AVALIADOS

Uma pesquisa bibliográfica de trabalhos relacionados foi realizada com o intuito de conhecer o estado da arte de aplicativos voltados para a gerência de academias. Segundo Marconi, Lakatos et al. (2002), a pesquisa bibliográfica compreende toda a bibliografia já publicada sobre um determinado assunto. Seu objetivo é aproximar o pesquisador com o tema abordado.

A consulta por trabalhos relacionados ocorreu de dois modos: (a) pesquisas em bibliotecas digitais amplamente difundidas como o Google Acadêmico, IEEE Explore e SCOPUS, de modo a constatar o que existe de aplicativos já publicados em artigos e monografias; (b) procura por app's publicados na Google Play. A Google Play foi escolhida por ser a loja oficial de aplicativos para Android e ter o maior número de downloads dentre lojas de aplicativos. As *strings* de buscas utilizadas foram: “aplicativo para controle de desempenho em academias de ginástica” e “aplicativo para agendamento de aulas em academias de ginástica”, tanto em português como em inglês.

Dos trabalhos encontrados, foram filtrados, após a revisão do texto ou descrição do aplicativo, somente aqueles que tem foco no treino e no desempenho dos alunos e abordam a questão do agendamento das aulas.

A Tabela 1 e Tabela 2 apresentam os principais trabalhos relacionados com a proposta a ser implementada. Nestas tabelas estão elencados os aplicativos e os seus respectivos: foco, custo, forma de interação, linguagem de programação e banco de dados utilizados, bem como as principais funcionalidades de software disponíveis. Para cada aplicativo foram encontrados um grande número de funcionalidades. Estas, por sua vez, foram agrupadas em algumas categorias. A Figura 1 elenca todas as funcionalidades encontrados nos trabalhos relacionados. As subcategorias foram agrupadas com base na descrição da funcionalidade.

As próximas seções apresentam cada um dos aplicativos encontrados nos trabalhos relacionados.

Tabela 1 – Trabalhos Relacionados

Aplicativo	Foco	Custo	Interação	Lingua- gem	B.D.	Categoria/ Funcionalidade
4GYM	Gerência/ Treino	0 ou 99	Web; App.	Não consta	Não consta	Financeiro, Grá- ficos, Treinamen- tos, Avaliações, Operacional e Controle.
Sistemas De Acade- mia	Treino/ Desempe- nho	0	App.	Java	SQLite	Cadastros e Vi- sualização.
Diário de um Ma- rombeiro	Treino/ Controle de carga	0	App.	Java	Não consta	Treinamento, Cadastros e Operacionais
Sistema de Geren- ciamento de Acade- mias de Ginástica	Gerência	0	<i>desktop</i>	Java	MySQL	Cadastros, Financeiro e Operacional.
<i>Fit Dolphin</i>	Treino	0	App.	Java	MySQL Object Box	Treinamento, Cadastro e Operacional.
SGCA	Gerência/ Controle de Infor- mações	Não consta	Web	Java (JSP e Java- beans), HMTL, CSS, Ja- vaScript	Postgre- SQL	Cadastros, Ope- racionais e Visu- alização.
SharpGym	Voltado para o personal trainer	Não consta	Web e App. Windows 8	C#	SQL Azure	Cadastro, Trei- namento, Con- trole, Avaliação e Gráficos
SISACAD	Gerência/ Desempe- nho	Não consta	Web	PHP, Ja- vaScript	MySQL	Cadastro, Trei- namento, Con- trole, Financeiro
FITGEN	Acompanha- mento de treinos/ Dietas	Não consta	App.	Java	SQLite	Treinamento, Cadastro e Visualização

Fonte – Autora

Tabela 2 – Trabalhos Relacionados

Aplicativo	Foco	Custo	Interação	Lingua- gem	B.D.	Categoria/ Funcionalidade
TecnoFit	Gestão	109 a 499	Web e App.	Não consta	Não consta	Financeiro, Gráfico, Operacional, Treinamento e Avaliação
Droid Fitness	Acompanhamento de atividades físicas	Não consta	App.	Java	NeoDatis ODB	Cadastro, Treinamento, Visualização e Avaliação
Sistema de Acompanhamento	Acompanhamento dos Exercícios	Não consta	App.	Java e PHP	MySQL	Cadastros e Gráficos
Sistema de Gerenciamento para Academia	Gerência/Treino	Não consta	desktop	Java	H2	Cadastro, Financeiro, Avaliação, Controle e Operacional
Gobody Fitness	Treino	0 a 99	Web e App.	Não consta	Não consta	Cadastro, Treinamento, Operacional e Avaliação
SysAcademy	Gerência	Não consta	Web	HTML, PHP,	MySQL	Cadastro, Avaliação, Visualização e Operacional
Sistema SCA	Gerência	Mensal: 94; Anual: 892	Web e App.	Não consta	Não consta	Financeiro, Controle, Avaliação, Treinamento e Operacional
Gympass	acesso à academias	Mensal: 119,90-599,90; Diária 20,00	App.	Não consta	Não consta	Fazer <i>check-in</i> no local escolhido e escolher a cidade e a academia
NOSSA PRO-POSTA	Treino/Agendamento de aulas	0	App.	Java	Firebase	Cadastro, Treinamento, Avaliação, Controle, Operacional, Visualização, Gráfico e "Leilão"

Fonte – Autora

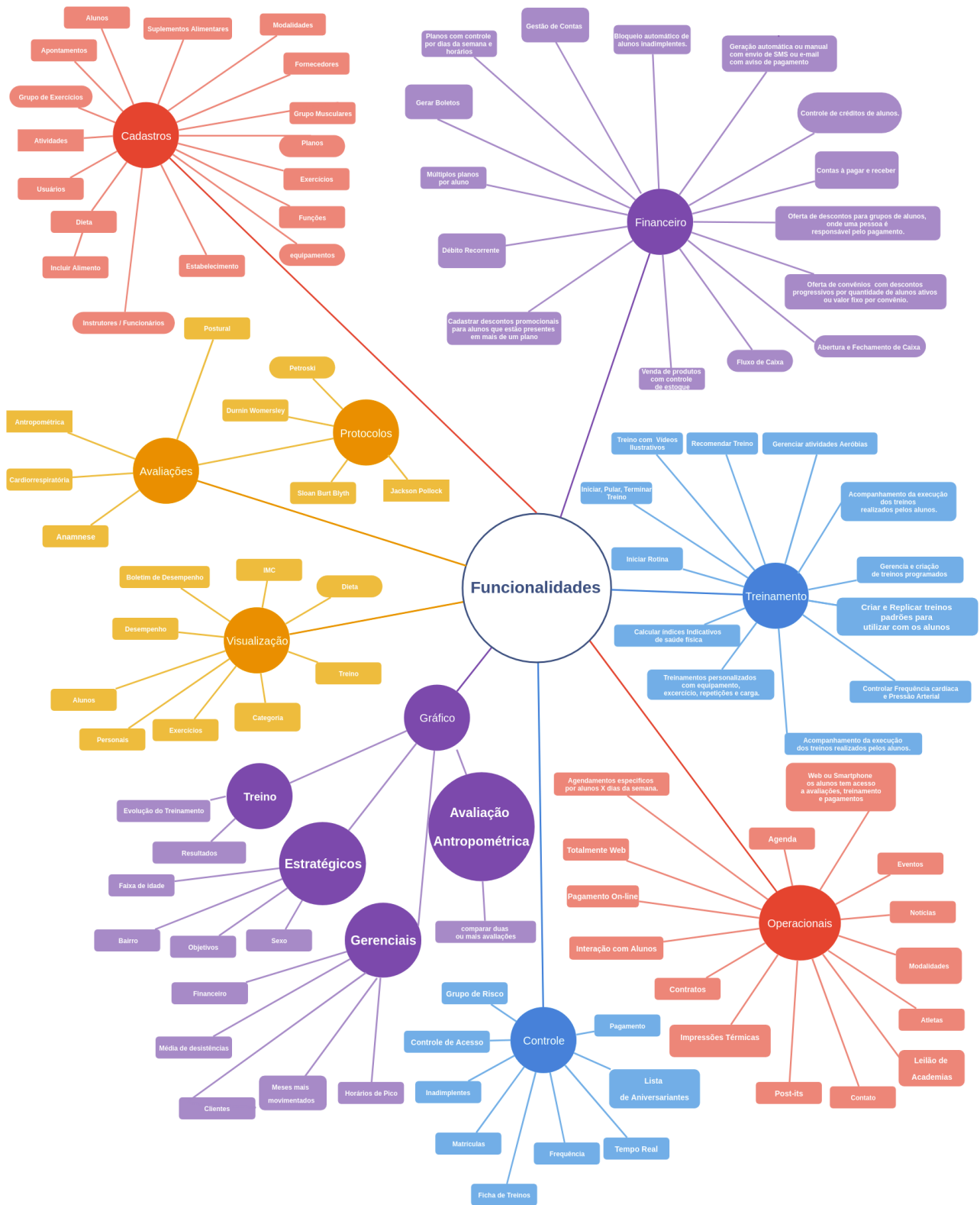


Figura 1 – Esquematização das categorias de funcionalidades encontradas nos trabalhos relacionados

Fonte – Autora

3.1 4GYM

O *software* 4GYM foi desenvolvido pela empresa Mobile Mind Soluções Tecnológicas (4GYM, 2019). O *software* possui as seguintes funcionalidades: controle de contas a pagar com previsão de saídas e entradas e um controle dos créditos dos alunos; Treinamentos; Mensalidades: gera o comprovante automático ou manual enviando um aviso de pagamento e bloqueando os inadimplentes; Planos/Modalidades; Promoções/Descontos; Venda de Produtos; Controle de Créditos; Fluxo de Caixa; Convênios; Grupos de Aluno; Débito Recorrente; Treinos Programados; Treino Padrão; Avaliações; Protocolos; Anamnese; Agenda; Controle de Acesso; *Semi-personal*; Grupo de Risco; Lista de Aniversariantes; Tempo Real: indica quais alunos estão na academia; Totalmente Web: acesso através de computadores, *tablets* e celulares; Pagamento online através do pag-seguro; Acesso para Alunos: avaliação, treinos e pagamentos; Interação com Alunos; Avaliação antropométrica - compara duas ou mais avaliações do aluno. Boletos; Gestão de contas; Abertura e fechamento de caixa; Contratos; *Post-its*; Impressões Térmicas; Elabora três tipos de gráficos: Estratégico - por idade, sexo, bairro; Gerenciais - desistências, meses mais movimentados, horários de pico.

O *software* está disponível em sites web e App. O mesmo possui uma versão gratuita e outra versão *premium* que custa R\$ 99,00. Conclui-se que o principal foco do *software* está na administração e gerencia de academias.

3.2 Sistema para Academias

O *software* Sistema para Academias foi desenvolvido por OLIVEIRA (2013). O *software* possui as seguintes funcionalidades: cadastro de usuários, atividades, grupo de exercícios e apontamentos; Consultar cadastros efetuados, Índice de Massa Corporal (IMC) e desempenho.

O *software* está disponível em versão App. de forma gratuita. O mesmo foi desenvolvido na linguagem Java, utilizado o SQLite como banco de dados. O foco do aplicativo está em treinamento e desempenho de alunos.

3.3 Sistema de Gerenciamento de Academias de Ginástica

O *software* foi desenvolvido por Jr, Granzoto e Pires (2013). Ele tem foco no gerenciamento de academias possuindo três tipos de usuários: administradores, professores e vendedores.

Os administradores tem as seguintes funcionalidades: cadastros, relatório de vendas, relatório de pagamentos e relatório de pedidos. Os professores possuem duas funcionalidades: cadastros de clientes e pagamento de mensalidade. Os vendedores podem cadastrar suplementos, consultar estoque e realizar pedidos aos fornecedores.

O *software* está disponível em versão *desktop* de forma gratuita. Ele foi desenvolvido na linguagem Java. O banco de dados usado foi o MySQL. O aplicativo é personalizável.

3.4 Diário de um Marombeiro

O *software* foi desenvolvido por Costa e Gonçalves (2015), com foco na análise de carga para praticantes de musculação. O App. possui as seguintes funcionalidades: cadastrar treinos, cadastrar exercícios e registrar controle de carga. O aplicativo retorna uma mensagem para cada configuração das informações cadastradas. Por exemplo, se houve uma falha concêntrica antes do intervalo de repetições a mensagem retornada é “É aconselhável reduzir a carga. Consulte seu instrutor!”. Se ocorrer a falha concêntrica no mínimo de repetições cadastradas a mensagem exibida é “É aconselhável manter a carga, porém com possibilidade de aumento. Consulte seu instrutor! ”. Se houver falha concêntrica no máximo de repetições cadastradas a mensagem exibida é “Possibilidade de aumentar a carga. Consulte seu instrutor!”. Por fim permite o envio do treino por *email*.

O *software* é disponível em versão App., de forma gratuita. Ele foi desenvolvido na linguagem Java, sendo que o banco de dados usado não foi especificado.

3.5 FitDolphin

O App. foi desenvolvido por Vidiri e Marques (2018), com foco no treinamento de alunos possui as seguintes funcionalidades: manter rotinas, exercícios, contas de usuário, e objetivo, iniciar rotina de exercícios, enviar *feedback*.

É disponível em versão App., gratuito, desenvolvido na linguagem Java. O banco de dados usado foi o MySQL para uso *online* e para uso *offline* foi utilizado *Object Box*.

3.6 SGCA

O Sistema de Gerenciamento e Controle para Academias (SGCA) foi desenvolvido por MONTEIRO, HARLEY et al. (2010), com foco no gerenciamento e controle de informações é dividido em duas áreas: **acesso público** que possui as seguintes informações:

- **eventos** que possui as seguintes funcionalidades: publicação de eventos de saúde que a academia está elaborando ou participando e divulgação de festas, bandas, fotos e outros patrocínios;
- **notícias** que possui as seguintes funcionalidades: Exibir as principais notícias de saúde no mundo e as novidades na região e no país onde fica localizada a empresa;

- **modalidades:** exibe informações sobre horários, preços, promoções; descrição detalhada das modalidades e visualizar o currículo e o perfil dos instrutores e das modalidades proposta pela academia;
- **atletas:** Divulgação das atividades em maior potencial da academia e o perfil de cada uma delas e exibe os atletas que obtiveram os melhores resultados nas atividades da academia avaliadas pelo professor ou instrutor;
- **contatos:** exibe informações sobre os possíveis contatos com a empresa como: telefone, endereço, *email*, fax e outros.

E **Acesso privado**, que possui as seguintes funcionalidades:

- cadastros de dados no sistema;
- visualização de dados do sistema;
- alterar e apagar os dados do sistema.

O SGCA está disponível em versão web. Possui custo mínimo mas não é informado o valor. O sistema foi desenvolvido nas linguagens Java com uso de Java Server Page (JSP) e javabeans, HyperText Markup Language (HTML), Javascript e Cascading Style Sheets (CSS). O banco de dados usado foi o PostgreSQL.

3.7 SharpGym

O *software* foi desenvolvido por Cavaleiro (2013), com foco no auxílio de profissionais de treinamentos e *personal trainer* que possuem academias. O sistema possui as seguintes funcionalidades: armazenar informações cadastrais, realizar o controle dos horários de entrada e saída dos treinos, registrar os tipos de atividades disponíveis na academia e o desempenho do aluno em atividades aeróbicas, registrar quantidade de carga, número de séries, repetições dos exercícios, informações de frequência cardíaca e pressão arterial, calcular os índices com IMC e percentual de gordura, demonstrar a evolução do treinamento, registrar a frequência dos alunos, gerar boletim de desempenho, contendo as avaliações físicas realizadas e evolução do treinamento.

É disponível em versão mobile para dispositivos com Windows 8 e Web. O aplicativo é gratuito, desenvolvido na linguagem C#, o banco de dados usado foi o SQL azure.

3.8 SISACAD

O *software* foi desenvolvido por Queiroga (2012), com foco na informatização de academias e filiais tanto na parte de gerencia como na disponibilização de resultados aos

frequentadores. O sistema possui as seguintes funcionalidades: efetuar *login*, cadastro de usuários, alunos, exercícios, séries, frequência, fornecedores e equipamentos, liberação de acesso a outros usuários, e geração de boletos.

É disponível em versão web, desenvolvido nas linguagens Hypertext Preprocessor (PHP) e JavaScript. O banco de dados usado foi o MySQL.

3.9 FitGen

O App. foi desenvolvido por MESSIAS (2015), com foco no apoio de atletas e pessoas em geral em suas práticas esportivas. O aplicativo ajuda no acompanhamento de treinos, dietas e especificamente quem pratica musculação no controle da carga de exercícios diários e na rotina de alimentação.

O sistema possui as seguintes funcionalidades: cadastrar treino, exercícios, dietas e alimentos, consultar treinos e dietas e recomendar treino, iniciar treino, passar treino, terminar treino. É disponível em versão aplicativo, desenvolvido nas linguagem Java. O banco de dados usado foi o SQLite.

3.10 TecnoFit

O sistema TecnoFit foi desenvolvido pela Tecnofit Tecnologia e Sistemas LTDA. “É um software de gerenciamento para academias ou qualquer outro negócio da área *fitness*, licenciado no modelo Software as a Service (SaaS) ou Software como Serviço, sem fidelidade ou qualquer surpresa” (TECNOFIT, 2018).

Possui funcionalidades para controle financeiro completo, *dashboards*, personalização de planos e contratos, facilidades para cobranças, relacionamento com cliente, controle de acesso de alunos, avaliação física e ficha de exercícios, relatórios, área do aluno, *crossFit*. É disponível na versão web mas possui um aplicativo para os alunos.

3.11 DroidFitness

O sistema *DroidFitness* foi desenvolvido por Fantinel e Parisoto (2012), com foco no acompanhamento das atividades físicas realizadas em academias de ginásticas.

Possui funcionalidades voltadas para cadastros, treinos e avaliações onde é possível cadastrar os estabelecimentos, instrutores, exercícios, séries, treinos, alunos, aparelhos e grupo musculares, visualizar o desempenho através de avaliações. É disponível em forma de aplicativo, desenvolvido na linguagem Java e o banco de dados utilizado foi NeoDatis Object Data Base (NeoDatis ODB).

3.12 Sistema de Acompanhamento

O sistema foi desenvolvido por Ribinski (2014), com foco no acompanhamento dos exercícios realizados por alunos em academias de ginásticas.

Possui funcionalidades voltadas para cadastros e geração de gráficos de resultados, onde é possível cadastrar as medidas, a data inicial e final dos treinos, treinos e exercícios. É disponível em forma de aplicativo, desenvolvido na linguagem Java o servidor foi desenvolvido na linguagem PHP, e o banco de dados utilizado foi MySQL.

3.13 Sistema de Gerenciamento para Academia

O sistema foi desenvolvido por SOUZA (2011), com foco no controle organizado dos registros de professores que atuam na academia e dos alunos que a frequentam. O objetivo é gerenciar os exercícios que serão aplicados nos alunos.

Este sistema possui as seguintes funcionalidades: *login* no sistema; cadastro de alunos, funcionários, funções, modalidades; controle das matrículas, das fichas de treinos, frequência, pagamentos; permite realizar pagamentos, matricular alunos, avaliação física e registro de entrada e saída da academia; emite relatórios dos alunos, dos aniversariantes, funcionários, modalidades, matrículas em dia e vencidos;

É disponível em forma de sistema *desktop*, desenvolvido na linguagem Java, faz uso da ferramenta *Hibernate* para fazer o relacionamento com o banco de dados, esse relacionamento é conhecido como mapeamento objeto/relacional, a ferramenta *iReport* foi utilizada para emitir os relatórios e o banco de dados utilizado foi o H2.

3.14 Gobody Fitness

O GoBody foi desenvolvido pela SIVIS, “acompanhe o desempenho de seus alunos com gráficos de evolução, crie metas, prescreva treinos de qualquer lugar e seu aluno irá receber na hora. Seja o melhor profissional, nossa ferramenta a tecnologia” (GOBODY, 2018).

O sistema possui as seguintes funcionalidades: cadastro de professores, cadastro de alunos, avaliação física, prescrição de treinos, *post* no Blog Fitness, lista de exercícios, envio de *email*, suporte e terminal de impressão.

É disponível em sites web e App., possui a versão gratuita e versão *premium* que custa R\$ 99,00. Com foco no treino dos alunos.

3.15 SysAcademy

O sistema *SysAcademy* foi desenvolvido por Dogado (2018), com foco no controle dos cadastros de alunos, das avaliações físicas efetuadas, dos treinos e controle de mensalidades.

Possui as seguintes funcionalidades: cadastro de alunos, *personais*, planos, exercícios, categoria e treino; avaliação física, consultar alunos, *personais*, exercícios e categorias, efetuar pagamento das mensalidades e emitir relatórios. É disponível em versão web, desenvolvido na linguagem HTML e PHP, com uso dos *frameworks bootstrap* e CSS e o banco de dados utilizado foi MySQL.

3.16 Sistema SCA

“O Sistema SCA é um software para controle e gerenciamento de academias, *studios* ou clubes, desenvolvido pela ProSistemas, com as seguintes funcionalidades: controle financeiro, controle de acesso, avaliação física, ficha de treinamento, reconhecimento facial, impressão digital, terminal da academia” (SISTEMASCA, 2019).

É disponível em sites web e App., possui a versão gratuita e versão *premium*, o plano mensal custa R\$ 94,00 e o plano anual que custa R\$ 892,00.

3.17 Gympass

“O Gympass é um aplicativo que dá acesso a academias no Brasil e no mundo por meio de uma assinatura mensal ou do pagamento de diárias” (PARREIRAS, 2018). O foco do App. é oferecer acesso a diversas academias por meio do pagamento de uma única mensalidade. Para utilizar o estabelecimento o cliente escolhe um plano que pode ser mensal ou anual. Após o pagamento do plano, basta escolher o local e fazer o *check-in* no aplicativo, é necessário estar conectado a internet para utilizar o aplicativo.

O Gympass possui planos mensais que variam de R\$ 119,90 até R\$ 599,90 mensais. É possível utilizar planos personalizados pagando a partir de R\$ 69,90. As diárias, custam em média R\$ 20, o pagamento é feito por cartão de crédito.

3.18 Nossa Proposta

Nossa proposta engloba dois objetivos. Primeiro, permite ao usuário ter mais autonomia para realizar suas atividades ao mesmo tempo em que este consegue executar um treino elaborado por um profissional e acompanhar seu rendimento. Segundo, é uma solução viável para auxiliar a encontrar o horário ideal para a prática de exercícios e motivar as pessoas a praticar atividades físicas.

Nossa proposta é desenvolver um aplicativo para auxiliar academias ou *personais trainers* a organizar treinos dos alunos e flexibilizar os horários para atender os usuários. A proposta é que os alunos escolham as aulas e os horários propícios para a prática de exercícios e até mesmo de menor custo. A ideia é o aplicativo exibir uma tela com as aulas ofertadas pelas diversas academias, permitindo ao aluno escolher quais deseja realizar.

Além desta funcionalidade, o software oferece outras aplicabilidades como cadastros de clientes, de treinos, de professores, de exercícios e de modalidades. O aluno, ao visualizar o treino, pode selecionar um exercício.

3.19 Análise dos Aplicativos Relacionados

Em relação aos aplicativos relacionados, muitos deles tem objetivos bastante específicos, como por exemplo o Diário de um Marombeiro, em que somente um controle de pesos é proposto. Já FITGEN possui uma parte de acompanhamento de treinos associado a dietas.

Em termos de custos, os aplicativos Sistemas De Academia, Diário de um Marombeiro, Sistema de Gerenciamento de Academias de Ginástica e *FitDolphin* são completamente gratuitos. Os aplicativos *4GYM*, *TecnoFit*, *Gobody Fitness*, Sistema SCA e *Gympass*. Os demais trabalhos relacionados não informam essa informação.

A maioria das ferramentas relatadas nos trabalhos relacionados tem foco em gerência e treinos. Em termos de gerência, os trabalhos citados apresentam fluxo de caixa, incluindo pagamentos, registros de contas e valores de mensalidades. Na versão inicial da nossa aplicação, o fluxo de caixa não será implementado. Um grande número de aplicativos oferece o cadastro de treinamentos e a visualização dos mesmos. Os cadastros podem ser feitos por treinadores e pelos próprios alunos. Deste conjunto, destaca-se o protótipo de aplicação “FitGen” que faz o controle de dieta em conjunto com a parte dos exercícios. Em nossa aplicação academias, treinadores e alunos podem se cadastrar em um único aplicativo.

Dois trabalhos relacionados fazem agendamento de aula, porém estas são somente as aulas ofertadas pela academia onde o aluno é usuário. Em nosso trabalho, o aluno poderá agendar suas aulas em mais de uma academia.

O aplicativo *Gympass* oferece aos usuários o recurso de efetivação da compra de planos fornecidos pelo aplicativo. Através do plano, o aplicativo concede acesso a diversas academias cadastradas no sistema. Em nossa aplicação, o recurso de notificações poderia ser disparado pela academia, a fim de divulgar anúncios de serviços e promoções.

SharpGym, um dos trabalhos relacionados, faz uso do banco de dados em nuvem SQL Azure, porém esse aplicativo é disponível apenas para dispositivos com *Windows 8*. Diferente dos demais trabalhos relacionados, optamos por usar o banco de dados *Firebase*. O banco de dados permite que as operações ocorram na Nuvem, sem a necessidade de um servidor físico. O mesmo possibilita até 100 conexões simultâneas na versão gratuita.

Sumarizando, nossa proposta é desenvolver um aplicativo onde é possível relacionar os dias e os horários das aulas disponibilizados por academias. Além disso, planejamos que o usuário possa agendar a atividade diretamente pelo aplicativo, proporcionando a flexibilização dos horários, com a intenção de atrair novos usuários para as academias.

Outra funcionalidade proposta é oferecer ao treinador uma lista de alunos ca-

dastrados e o mesmo poder realizar a edição dos treinos e avaliações de desempenho e visualizar a disponibilidade de horários. Com isso, os alunos poderão visualizar seu treino, e visualizar através de gráficos o seu desempenho como peso, medidas, etc.

4 METODOLOGIA

Neste Capítulo serão vistos os passos necessários para a elaboração de um aplicativo de controle de desempenho de alunos e leilão de horários em academias.

4.1 Elicitação de Requisitos e Proposta de Validação

A elicitação dos requisitos do aplicativo ocorreu através de entrevistas em quatro academias onde os entrevistados foram os *personais* e com um *personal*. Durante a entrevista, um formulário foi apresentado, conforme modelo disponível no Apêndice A. A partir das entrevistas foi possível analisar que as academias utilizam *softwares* do tipo *desktop*. Apenas uma academia utiliza aplicativo próprio para visualização de treinos.

A partir dos requisitos coletados, foi implementado um aplicativo que possibilite aos *personais trainers* cadastrar alunos e treinos e aos alunos, visualizar o treino, agendar aulas, além de acompanhar seu desempenho e a variação de suas medidas corporais. A partir disso, o aluno poderá escolher e efetuar o agendamento.

Após a conclusão da implementação e fase de testes, foi disponibilizada uma versão para a validação em algumas academias, a fim de analisar as funcionalidades do modelo proposto. Ao final, um teste com cinco alunos foi realizado, usando um questionário avaliativo sobre o aplicativo, com o intuito de levantar quais foram os pontos positivos e negativos identificados.

4.2 Abordagem de Implementação

O aplicativo foi desenvolvido para o sistema operacional Android. Este sistema está presente em 85% dos dispositivos móveis, conforme mensurado por Ciriaco (2017). Para a elaboração do aplicativo foi utilizada a IDE Android Studio, o qual facilita a modelagem/prototipação das telas e também a emulação em diferentes plataformas de hardware.

Java foi a linguagem de programação utilizada na implementação. Tradicionalmente a linguagem é adotada para o desenvolvimento de App. no sistema operacional Android. Java possui uma vasta documentação, além da mesma ter sido adotada como linguagem de programação na grande maioria dos trabalhos relacionados.

Para o desenvolvimento do aplicativo foi utilizada a IDE Android Studio. Através da IDE, o aplicativo foi implementado na linguagem de programação Java e os dados coletados foram armazenados no banco de dados *Firebase*. As próximas seções descrevem cada uma das ferramentas citadas acima, e mostram os protótipos das telas e os diagramas de classes e de caso de uso do aplicativo.

4.2.1 Ferramenta de Desenvolvimento - Android Studio

Segundo AndroidStudio (2018), Android Studio é o Ambiente de Desenvolvimento Integrado (IDE) oficial para o desenvolvimento de aplicativos Android. A IDE é baseada no IntelliJ IDEA.

Segundo Brito (2015), o Android Studio foi anunciado oficialmente em 2013. Porém, ao final de 2014, a versão 1.0 foi divulgada pela Google. O Android Studio possui algumas vantagens, dentre elas estão a interface agradável, possibilitando o uso da IDE de forma simplificada. Android Studio possui também um dos mais completos recursos de autocompletar das IDEs existentes para o desenvolvimento em Android. A sugestão de autocompletar é automática, não necessitando utilizar comandos para preencher. A IDE possui integração com o Git e outros sistemas de controle de versão. Android Studio oferece um ambiente de desenvolvimento multiplataforma, ou seja, pode ser instalado em diversos sistemas operacionais e é gratuito.

O desenvolvimento para Android pode ser feito em linguagem Java. Segundo Schildt (2013), a linguagem Java foi proposta por James Gosling, Patrick Naughton, Chris Warth, Ed Frank e Mike Sheridan na Sun Microsystems em 1991. Java foi desenvolvido por causa da necessidade de se ter uma linguagem independente e que pudesse ser utilizada no desenvolvimento de software para diversas arquiteturas e sistemas operacionais disponíveis.

Segundo Perry (2016), a linguagem Java possui estrutura, regras de sintaxe e paradigma de programação próprios, sendo derivada da linguagem C. De acordo com Java (2018), Java é uma linguagem rápida, segura e confiável.

4.2.2 Ferramenta Prototipação - Justinmind Prototyper

Para a prototipação de telas utilizou-se o software *Justinmind Prototyper*. Segundo Saldit (2017), o software foi desenvolvido para a criação de sites e aplicativos para a web e *mobile*. O programa oferece ferramentas para elaboração de protótipos e *wireframes* sem grandes complicações.

O objetivo da ferramenta *Justinmind Prototyper* é simplificar a etapa de desenvolvimento. Ela é indicada para *start-ups* e *freelancers* que estão trabalhando em projetos pequenos ou médios. Ainda, de acordo com Saldit (2017), o programa adapta as ferramentas oferecidas de acordo com o tipo de conteúdo que está sendo projetado, seja *mobile* ou web.

Segundo Saldit (2017), algumas vantagens de *Justinmind Prototyper* são: possibilitar a elaboração de múltiplas telas, usando *screenshots* e ícones de sistemas Android e iOS; permitir o ajuste das propriedades de cada elemento individualmente; possuir resoluções pré-definidas compatíveis com diversos tipos de aparelhos, como *iPhone*, *iPad*, *smartphones* e *tablets*; possuir *design* responsivo; permitir a adição de elementos e efeitos

interativos; e funcionar de forma colaborativa. Por outro lado, a ferramenta não possui a exportação das telas para o formato *Extensible Markup Language* (XML).

4.2.3 Ferramenta Diagramação - Enterprise Architect

Para a criação dos diagramas de classes e uso foi utilizado a ferramenta *Enterprise Architect*. Segundo Software.com (2018), “Enterprise Architect é uma plataforma de desenvolvimento colaborativa para modelagem, design e gerenciamento baseada em UML e padrões similares.” Enterprise Architect permite:

- Criar modelos complexos de informações, softwares e hardwares;
- Modelar, gerenciar e rastrear requisitos para soluções implantadas;
- Modelar as hierarquias de classe, a implantação, os componentes e os detalhes de implementação;
- Criar mapas mentais, modelos de processos de negócios e diagramas de fluxo de dados usando UML;

4.3 Firebase

Segundo Orlandi (2019), Firebase é um Backend as a Service (BaaS) para aplicações Web e *mobile*. Ele foi lançado em 2004, crescendo em popularidade e tornando-se uma ferramenta muito utilizada e recomendada devido à quantidade de serviços oferecidos por ele, além da facilidade de implementação.

Segundo Adriano (2018), *Firebase* é uma plataforma móvel do Google que auxilia a desenvolver App.'s com alta qualidade e curto espaço de tempo. Os serviços são divididos em dois grupos:

- Desenvolvimento;
 - Realtime Database: ou banco de dados em tempo real. “Permite o armazenamento e sincronismo dos dados entre usuários e dispositivos em tempo real com um banco de dados NoSQL hospedado na nuvem” (ORLANDI, 2019). Os dados seguem disponíveis mesmo *off-line*.
 - Auth: de acordo com (ORLANDI, 2019) o serviço, oferece diversos métodos de autenticação, como e-mail/senha, provedores de terceiros, entre outros métodos.
- Crescimento;
 - Storage: permite o armazenamento e o compartilhamento de imagens, áudio, vídeo ou outro conteúdo gerado pelos usuários.

- Cloud Messaging: permite enviar mensagens e notificações gratuitamente para usuários de várias plataformas. “As mensagens podem ser enviadas para dispositivos únicos, grupos de dispositivos, segmentos de usuários ou tópicos específicos” (ORLANDI, 2019).

Neste trabalho, o modo gratuito do Firebase é usado. Neste modo, é possível a conexão de até 100 usuários simultaneamente para uma dada aplicação. Já a largura de banda fica limitada a 10 Gbps e o espaço de armazenamento é de 1 GB.

4.4 Diagramas de Classes e Estudo de Caso

A seguir estão descritos os requisitos funcionais do sistema em forma de diagramas de estudo de caso. O sistema possui quatro atores, o aluno, o *personal*, a academia, e o usuário não cadastrado.

O ator aluno representa os alunos que possuem cadastro em academias, o ator *personal* representa os *personais* que trabalham em academias, o ator academia representa os estabelecimentos e o ator usuário não cadastrado representa as pessoas que não possuem cadastros mas que desejam praticar exercícios.

A Figura 2 apresenta as funcionalidades que o usuário aluno poderá efetuar no aplicativo. A Figura 3 demonstra as funcionalidades que serão realizadas pelo usuário *personal*. A Figura 4 expõe as funcionalidades a serem realizadas pelo usuário academia. Por fim, a Figura 5 apresenta as funcionalidades que o aluno não cadastrado poderá realizar.

A Figura 6 representa as classes que serão utilizadas no desenvolvimento do aplicativo e seus atributos.

4.5 Protótipos das Telas

A Figura 7 demonstra como é a tela inicial do aplicativo. O aplicativo solicita que o usuário identifique-se como aluno, *personal trainer*, academia ou aluno não cadastrado.

Ao identificar-se no sistema, é exibida uma tela de *login* como mostra a Figura 8 e ao efetuar o login uma tela com menu lateral é inicializada contendo as funcionalidades disponíveis, caso o usuário não possua cadastro ao identificar-se a tela com o menu lateral é inicializado, como mostra a Figura 9 e a Figura 10.

Caso o usuário identifique-se como academia ele pode realizar as funcionalidades descritas na Figura 11. Caso o usuário identifique-se como *personal* ele pode realizar as funcionalidades descritas na Figura 12.

Ao selecionar um aluno cadastrado o *personal* pode realizar as funcionalidades descritas na Figura 13.

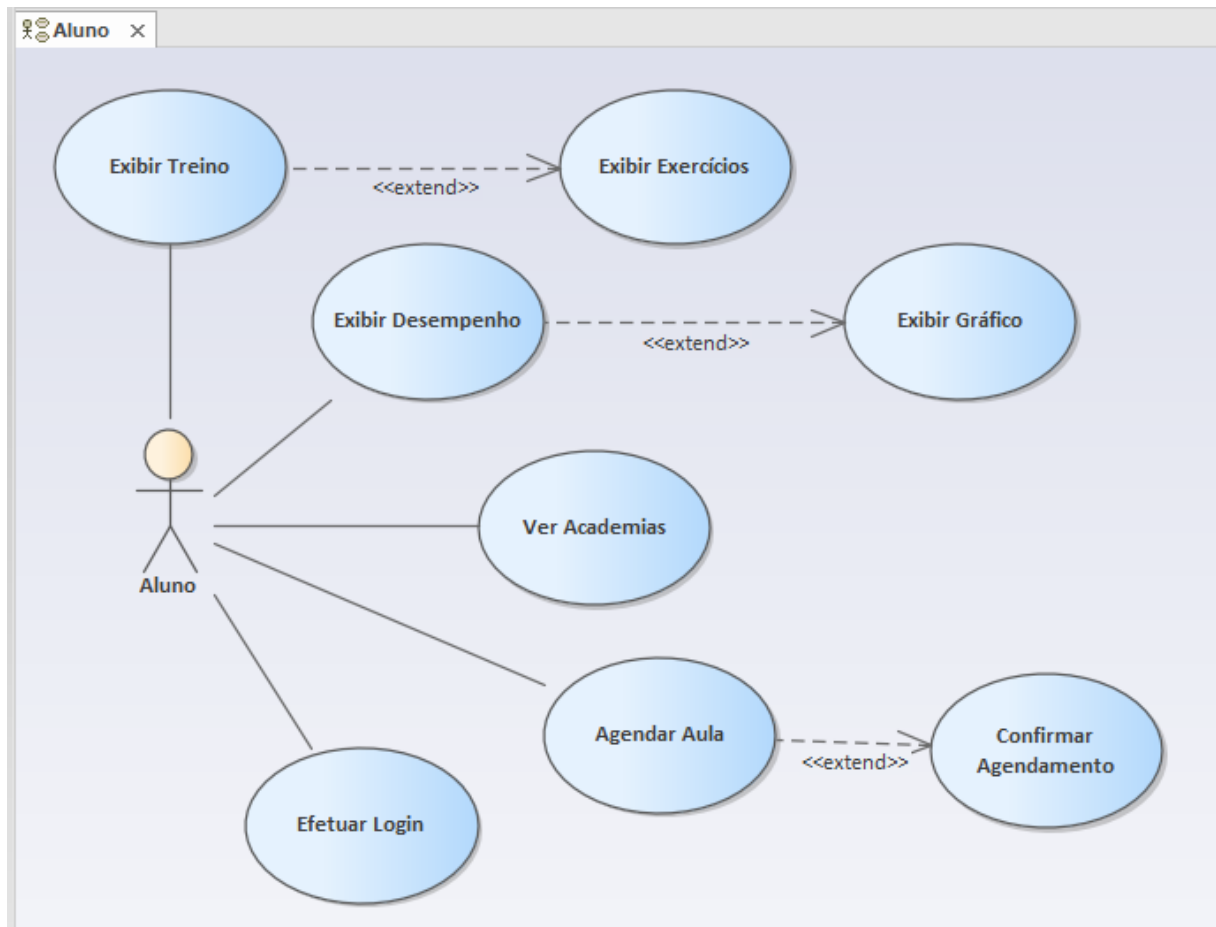


Figura 2 – Caso de Uso Aluno

Fonte – Autora

Ao selecionar um treino cadastrado ou uma avaliação cadastrada, o *personal* pode realizar as funcionalidades de cadastrar exercícios e medidas corporais respectivamente, como mostra a Figura 14.

Caso o usuário identifique-se como aluno ele pode realizar as funcionalidades descritas na Figura 15, tanto o aluno como o aluno não cadastrado possuem duas funcionalidades em comum, como mostra a Figura 16.

4.6 Considerações do Capítulo

Neste capítulo foram vistos os aspectos metodológicos utilizados para a elaboração do ferramenta de agendamento de treinos. Para tanto, detalhamos as ferramentas e recursos de *software* necessários para a prototipação e implementação do aplicativo. Por fim, apresentamos os diagramas de caso de uso, o diagrama de classes e o protótipo das telas.

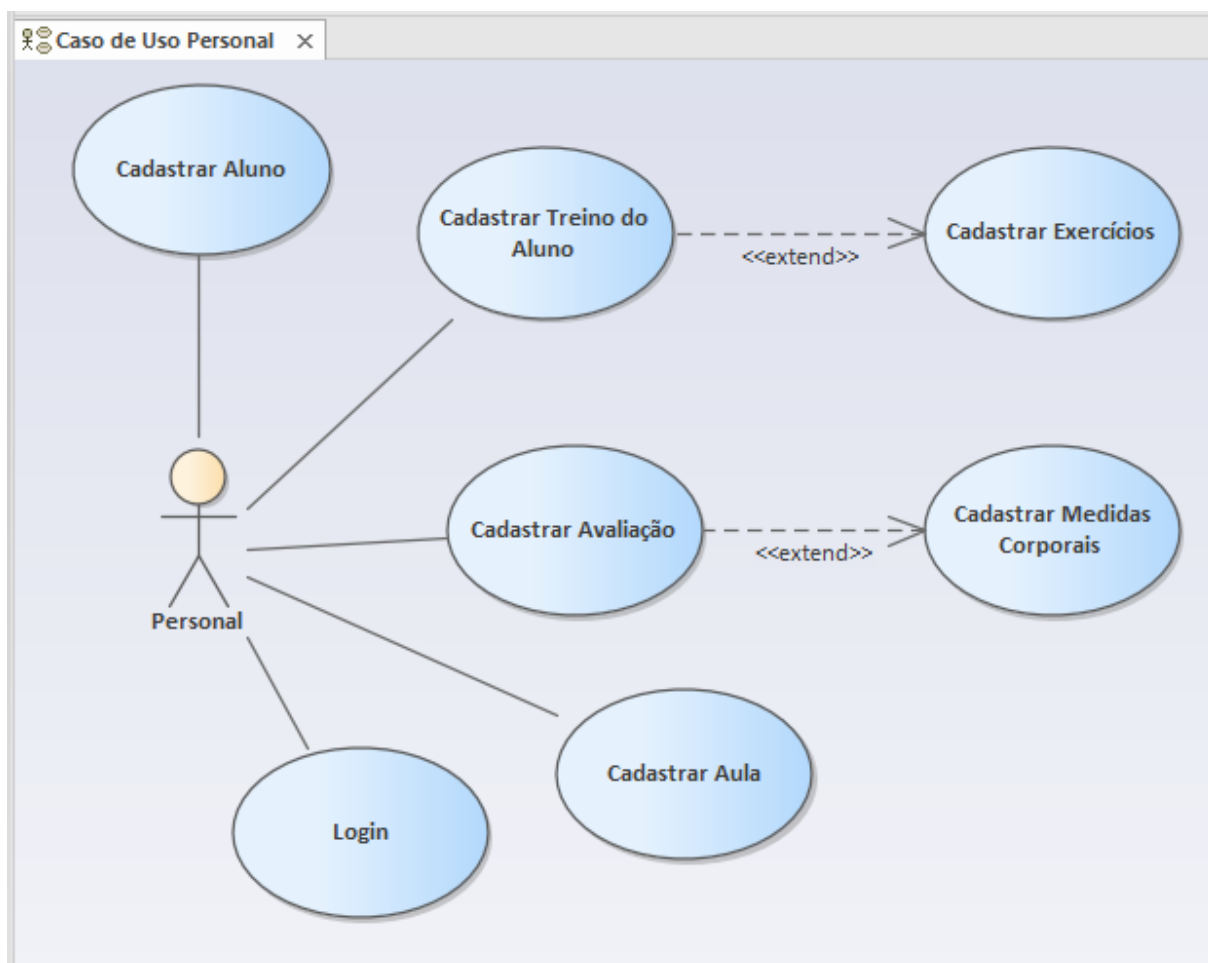


Figura 3 – Caso de Uso Personal

Fonte – Autora

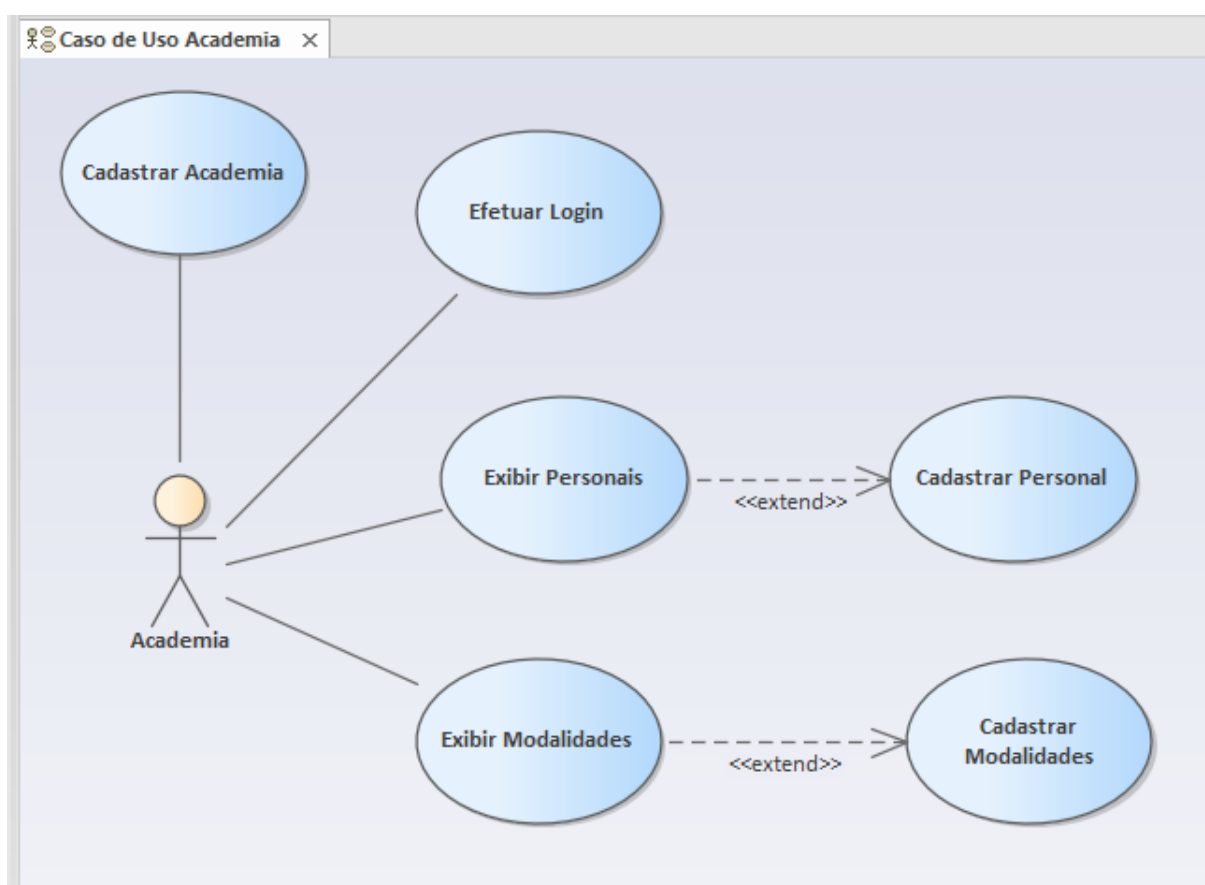


Figura 4 – Caso de Uso Academia

Fonte – Autora

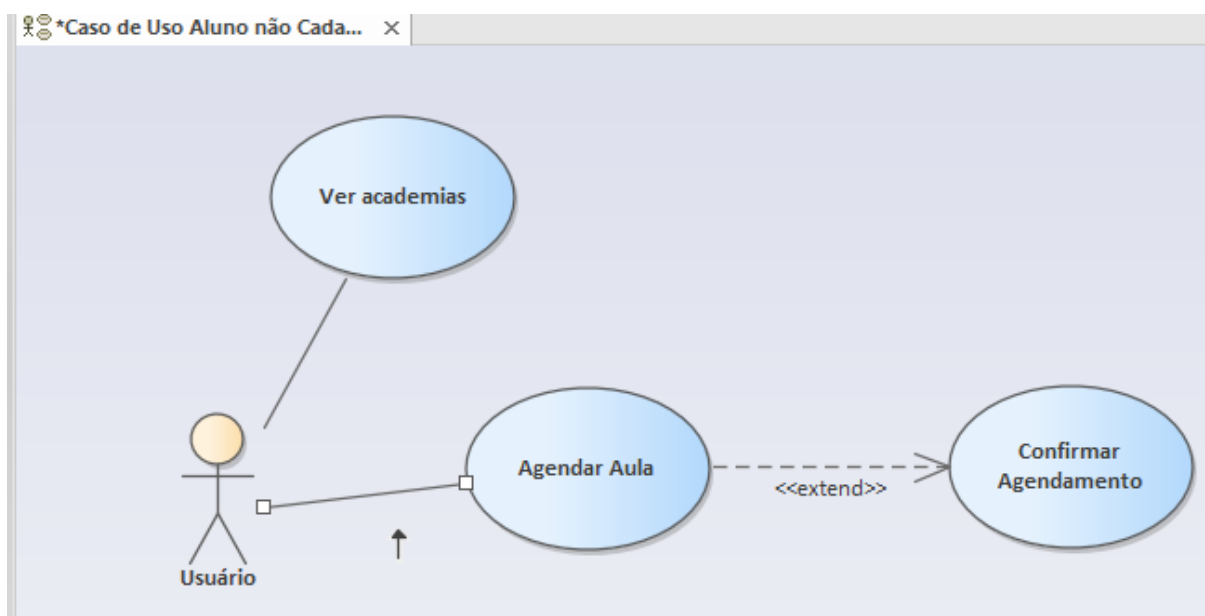


Figura 5 – Caso de Uso Aluno não Cadastrado

Fonte – Autora

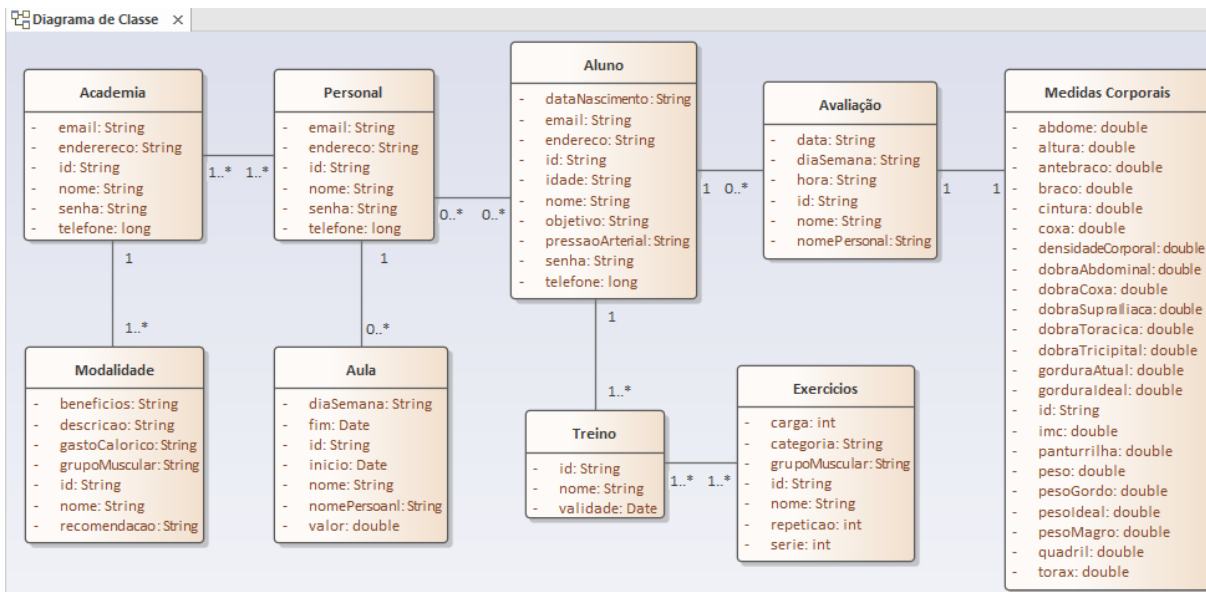


Figura 6 – Diagrama de Classes

Fonte – Autora

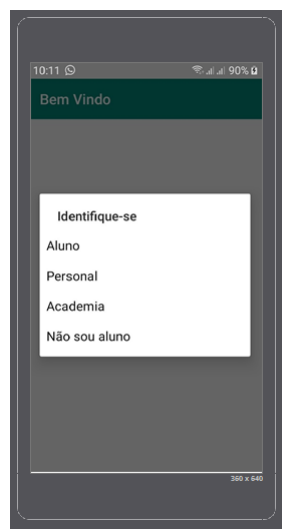


Figura 7 – Tela Inicial do App

Fonte – Autora

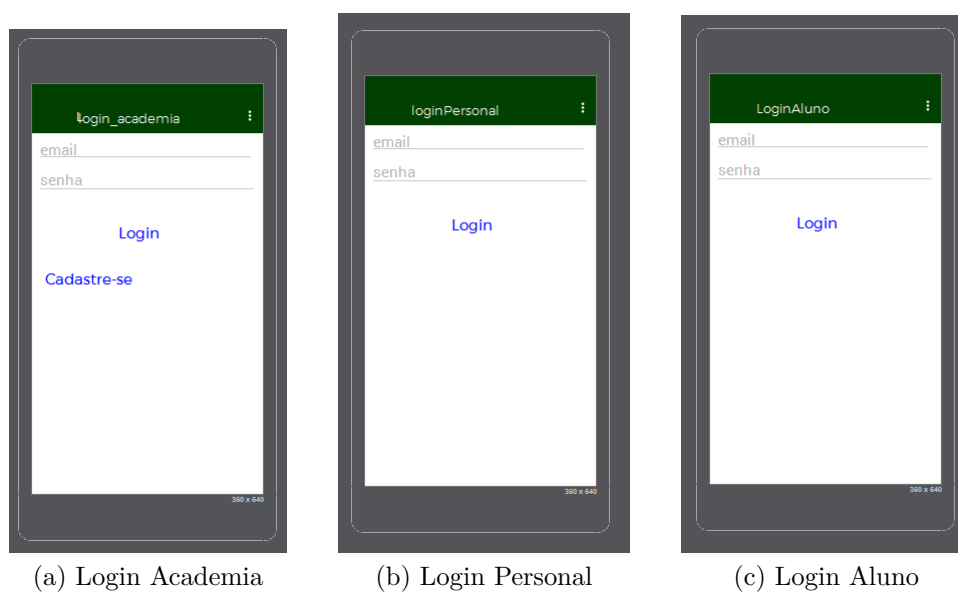
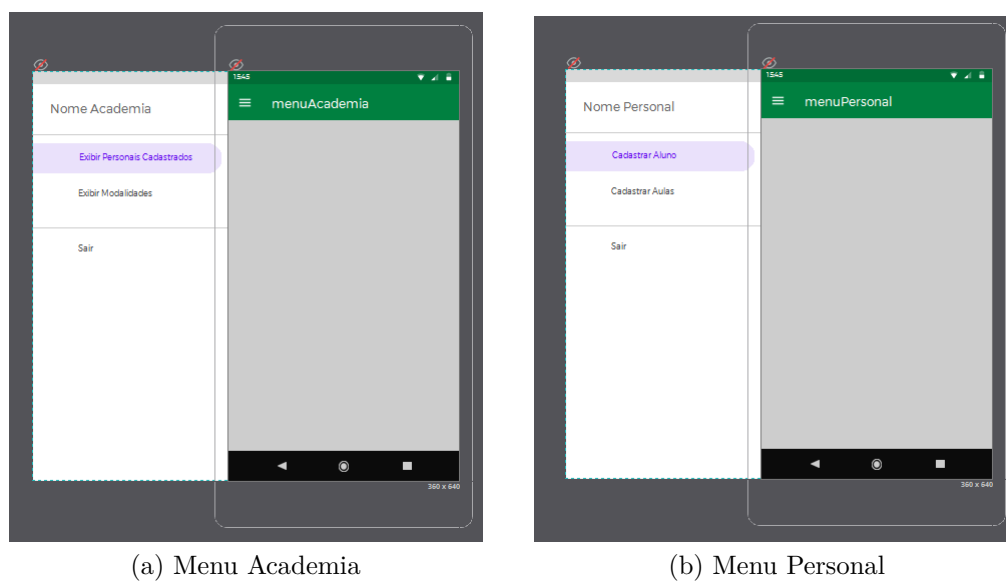
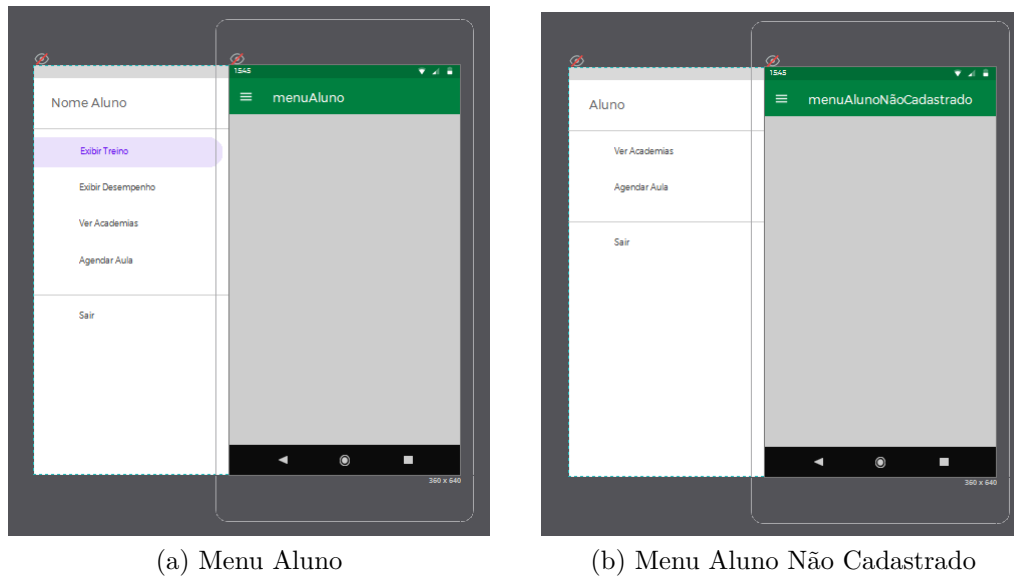
Figura 8 – Telas de *Logins*

Figura 9 – Telas de Menu



(a) Menu Aluno

(b) Menu Aluno Não Cadastrado

Figura 10 – Telas de Menu

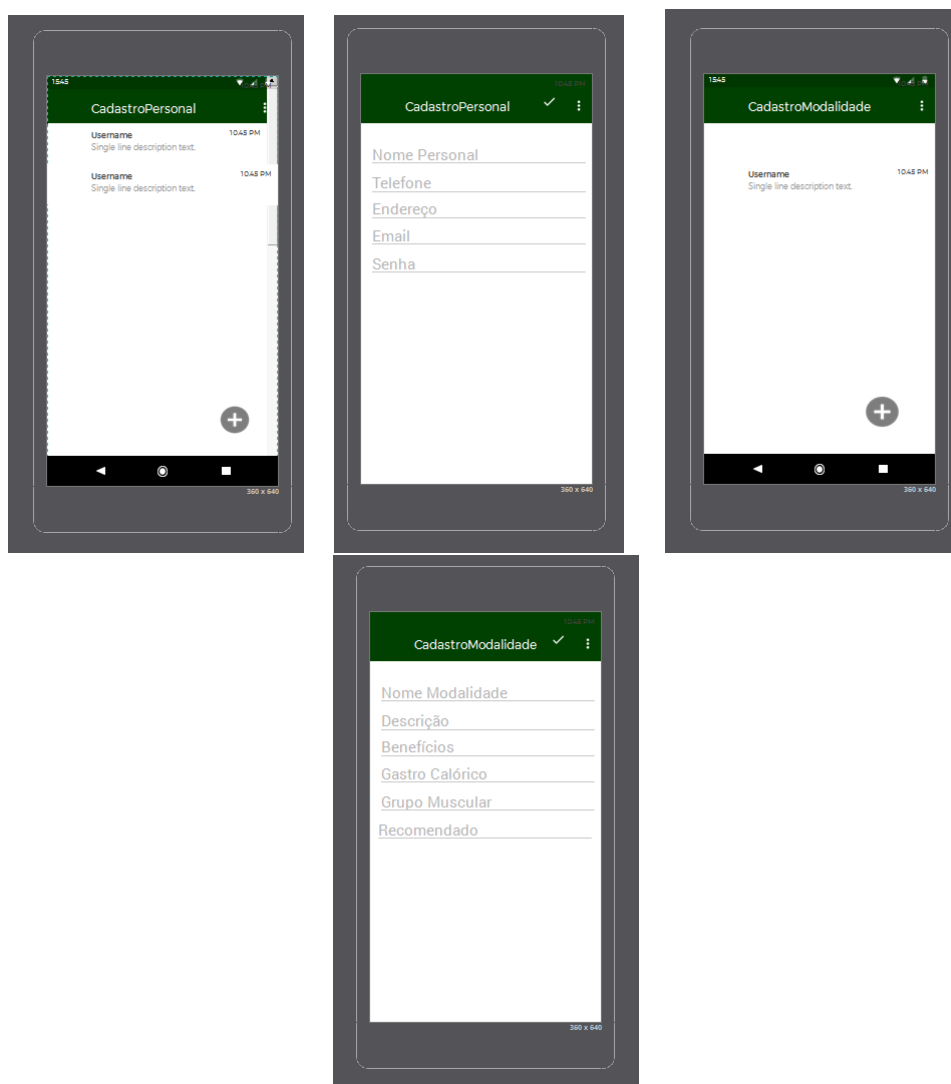


Figura 11 – Funcionalidades da Academia

Fonte – Autora

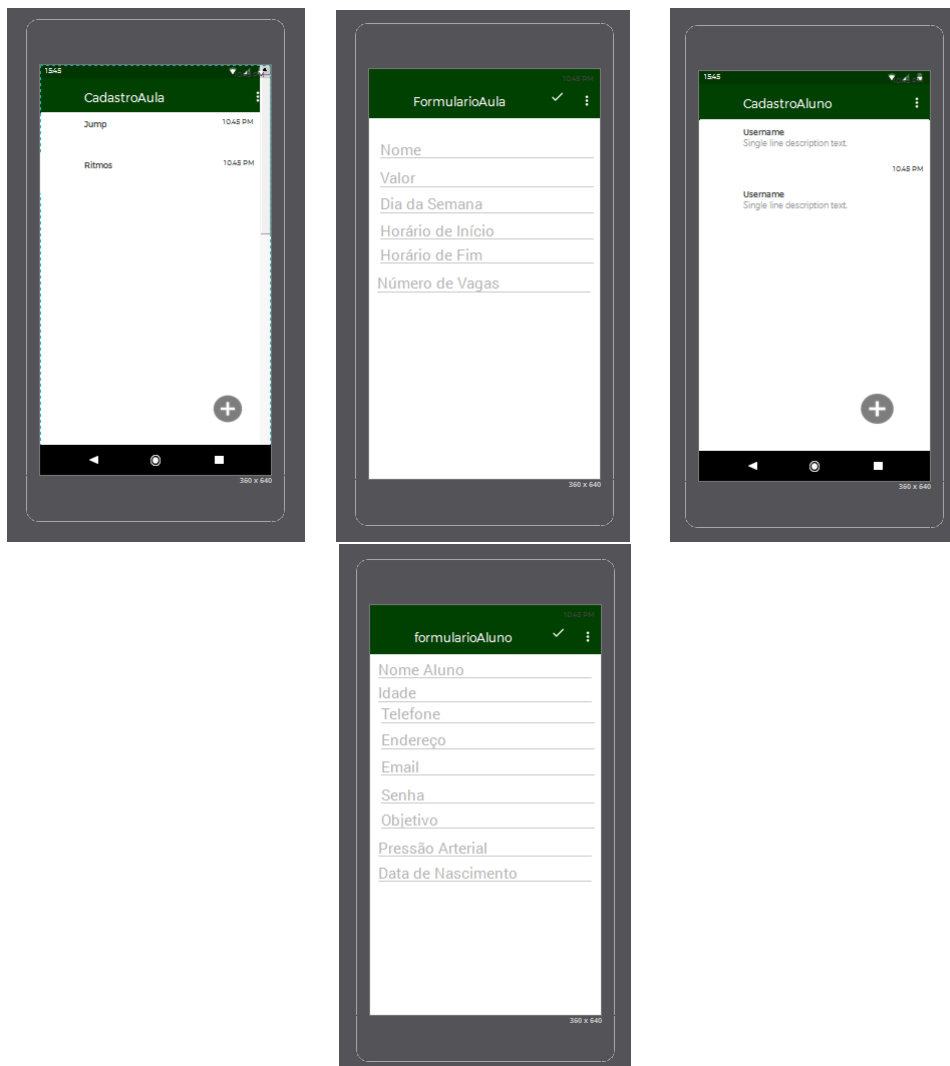


Figura 12 – Funcionalidades do Personal

Fonte – Autora

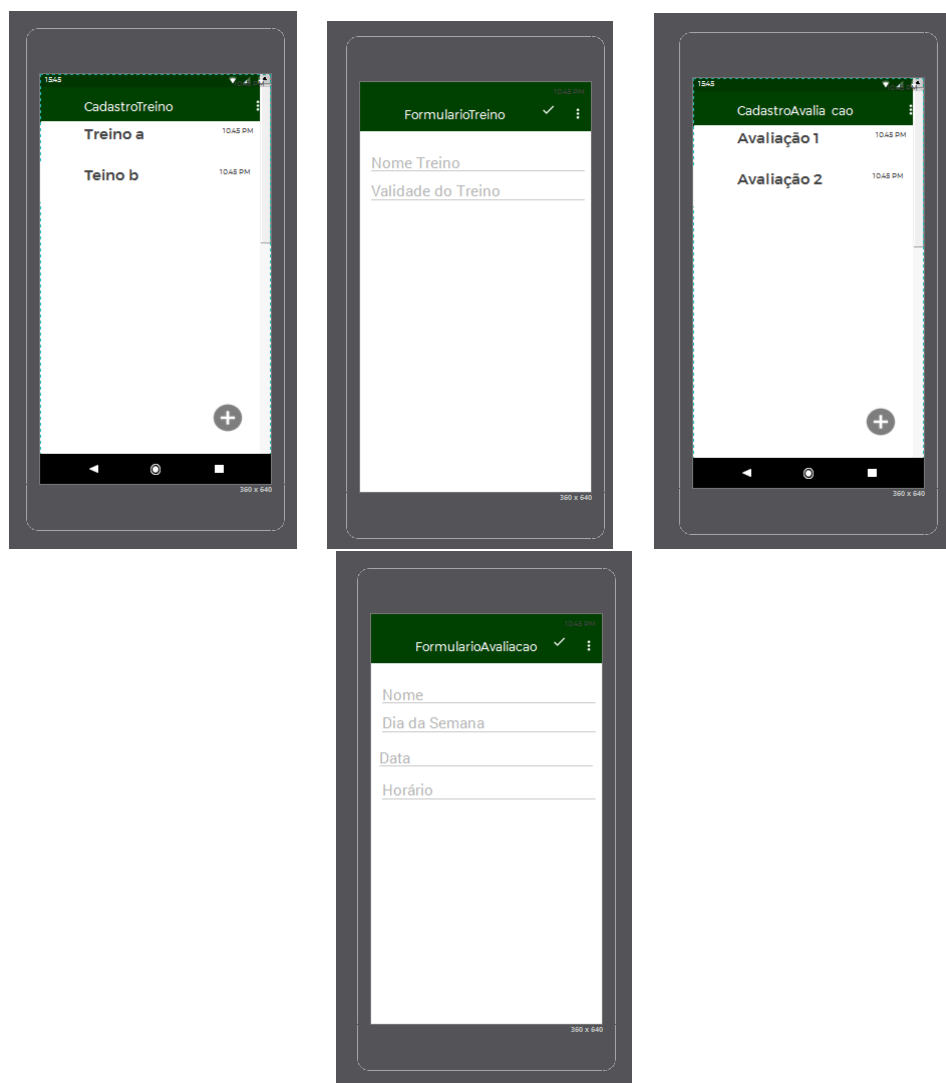


Figura 13 – Funcionalidades do Personal ao Selecionar um Aluno

Fonte – Autora

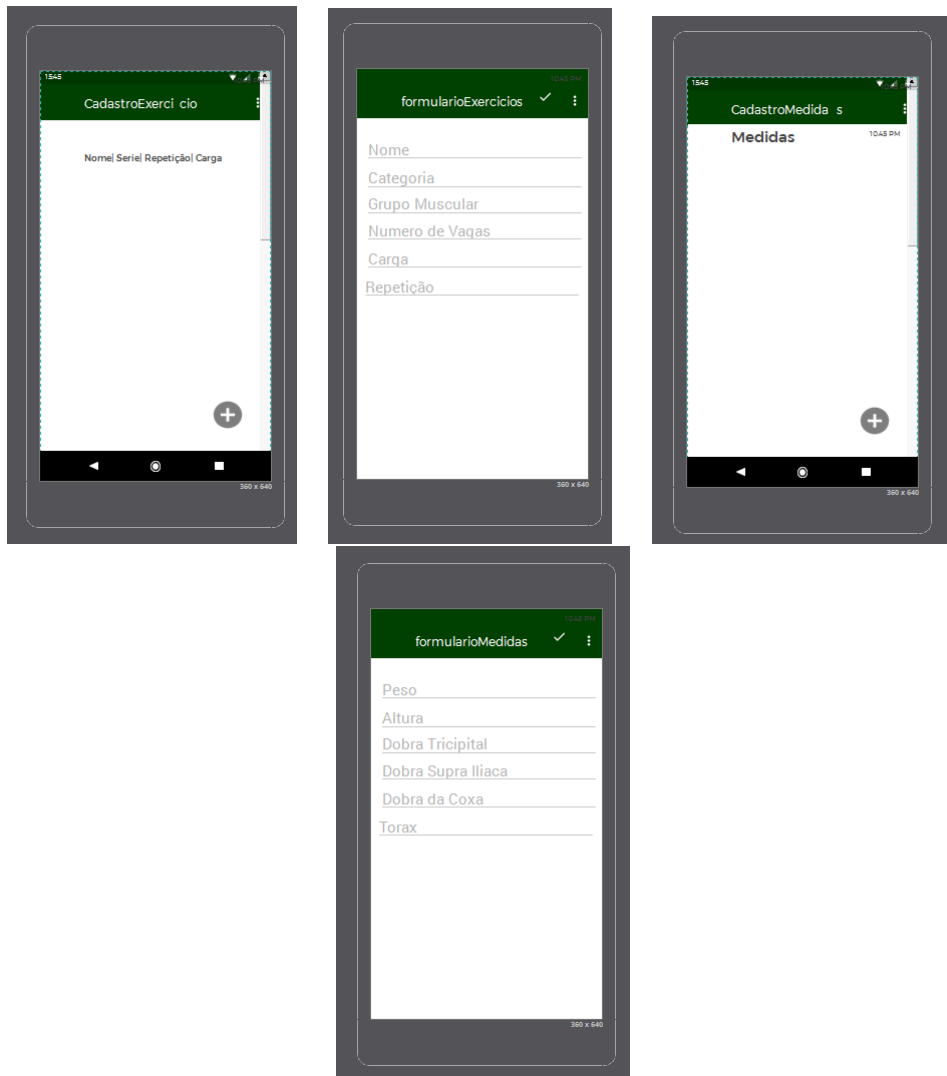


Figura 14 – Funcionalidades do Personal ao Selecionar um Aluno

Fonte – Autora

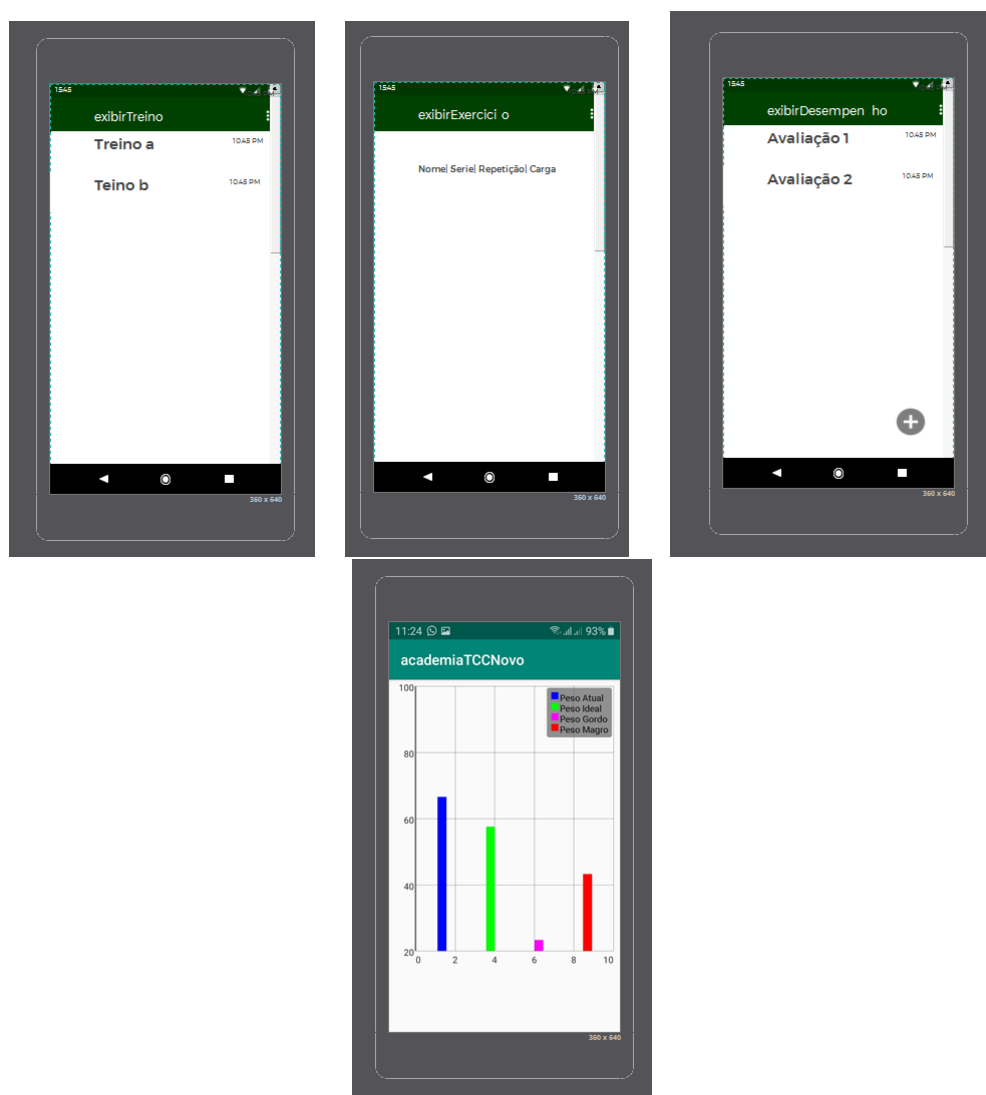


Figura 15 – Funcionalidades do Aluno

Fonte – Autora

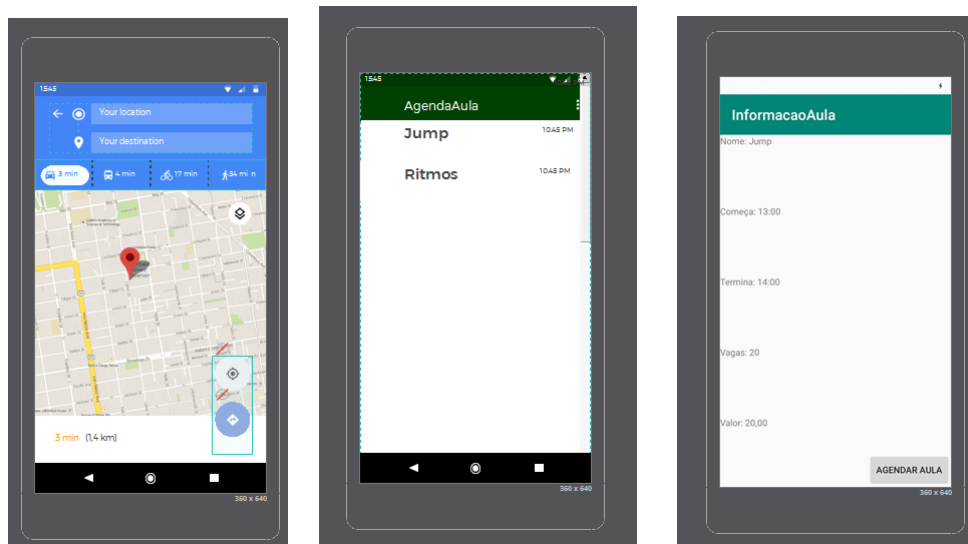


Figura 16 – Funcionalidades do Aluno não Cadastrado e do Aluno

Fonte – Autora

5 RESULTADOS EXPERIMENTAIS

Neste capítulo são descritas as etapas de implementação adotadas, seguindo a metodologia proposta. Após as implementações foi feita a validação do aplicativo.

5.1 Implementação

A tela inicial tem por objetivo identificar qual usuário está acessando o aplicativo. Isso é solicitado devido a cada usuário possuir um conjunto de funcionalidades próprias. A aplicação possui quatro tipos de usuários: alunos, *personais*, academias e usuários que não possuem vínculo com nenhuma academia, como mostra a Figura 17.

Ao identificar-se como aluno, o mesmo deve realizar seu login fornecido pelo *personal* disponível quando o aluno realiza sua matrícula em uma determinada academia. Após efetuar o login é exibido um menu lateral, onde o aluno poderá escolher entre exibir os treinos, visualizar seu desempenho, localizar academias através do mapa e agendar as aulas, como mostra a Figura 18.

Ao selecionar exibir treinos, é mostrado uma tela contendo todos os treinos do aluno. Ao selecionar um dos treinos é possível visualizar os exercícios daquele treino, como mostra a Figura 19.

Ao selecionar visualizar o desempenho é exibido uma tela contendo todas as avali-

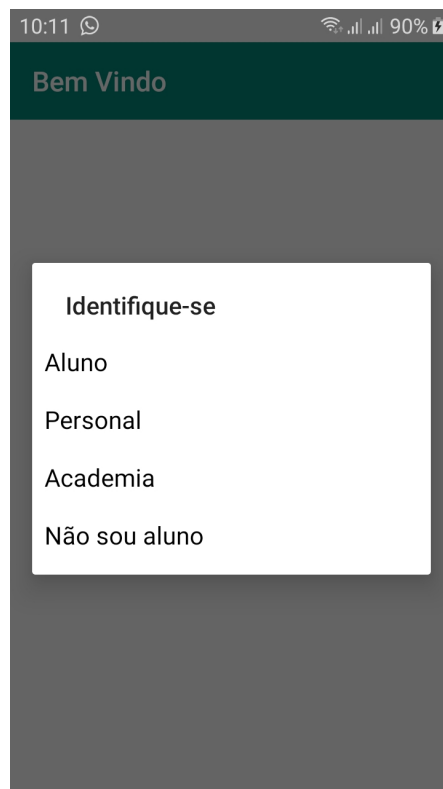


Figura 17 – Tela Inicial

Fonte – Autora

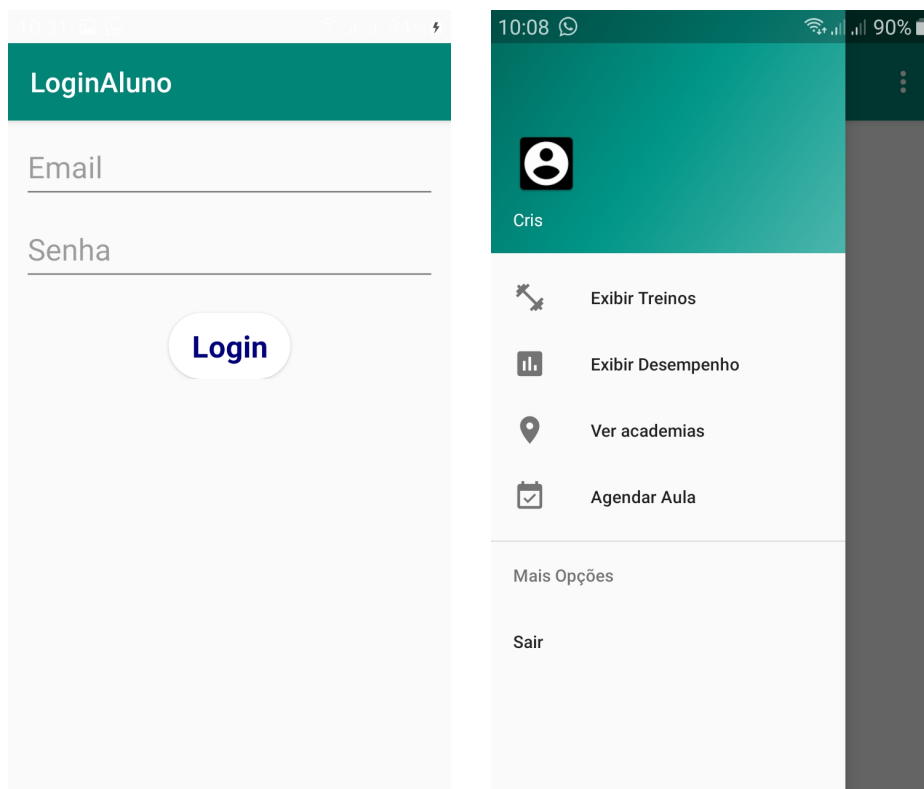


Figura 18 – Login Aluno / Menu Aluno

Fonte – Autora

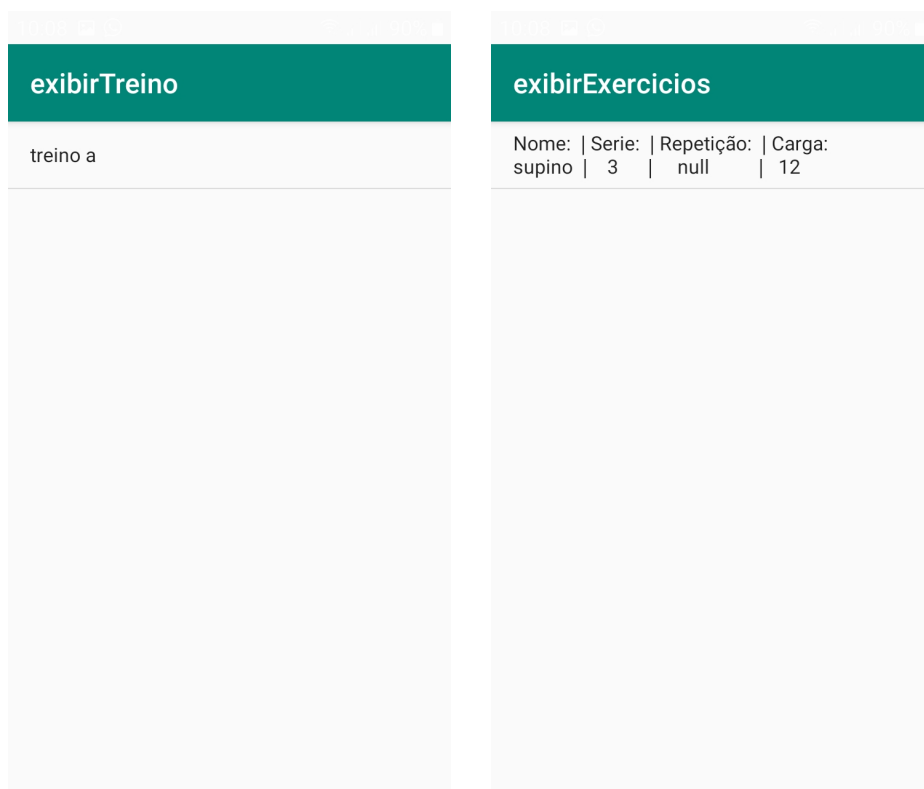


Figura 19 – Visualizar Treinos / Exibir Exercícios

Fonte – Autora

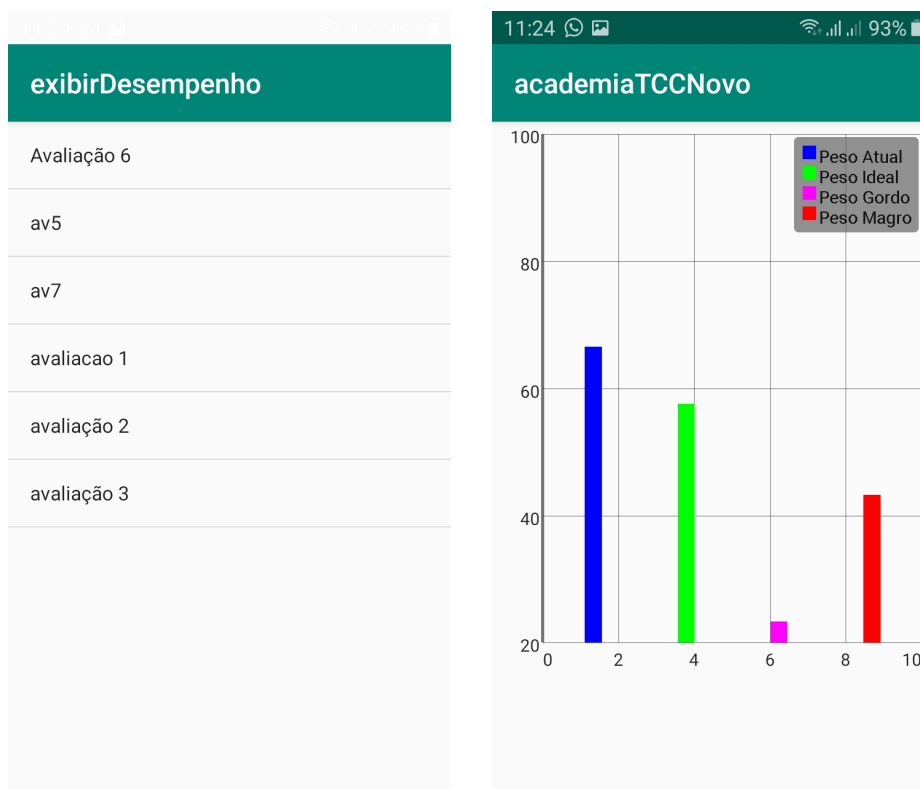


Figura 20 – Listar Avaliações / Exibir Desempenho

Fonte – Autora

ações do aluno e, ao selecionar uma das avaliações, é possível visualizar seu desempenho em um gráfico obtido através do protocolo de *Pollock* de três dobras. Esse protocolo é a soma das dobras cutâneas em relação a idade, como mostra a Figura 20.

Através da API Google Maps, é possível exibir um mapa onde consta a localização atual do usuário, como mostra a Figura 21. É possível agendar as aulas de qualquer academia que são exibidas por dia da semana e por hora, ou seja, a cada hora é alterada a exibição. Ao escolher uma aula é possível visualizar as informações como valor, quantidade de vagas e efetuar o agendamento, como mostra a Figura 22.

Ao identificar-se como *personal*, o mesmo deve realizar seu login fornecido pela academia no qual trabalha. Após efetuar o login, é exibido um menu lateral onde o *personal* poderá escolher entre cadastrar os alunos e cadastrar as aulas como mostra a Figura 23.

Ao selecionar cadastrar aluno, uma tela é inicializada contendo todos os alunos do profissional, como mostra a Figura 24. Ao clicar no botão de mais é exibido um formulário de cadastro, com o objetivo de levantar algumas informações importantes a respeito do aluno, como mostra a Figura 25. O *personal* pode cadastrar o treino e avaliação de um aluno, primeiro selecionando um aluno na lista, após um *pop-up* é exibido na tela perguntando qual ação deseja realizar, como mostra a Figura 26.

Ao selecionar cadastrar treino, é exibido uma lista com todos os treinos do aluno,



Figura 21 – Exibir Mapa

Fonte – Autora

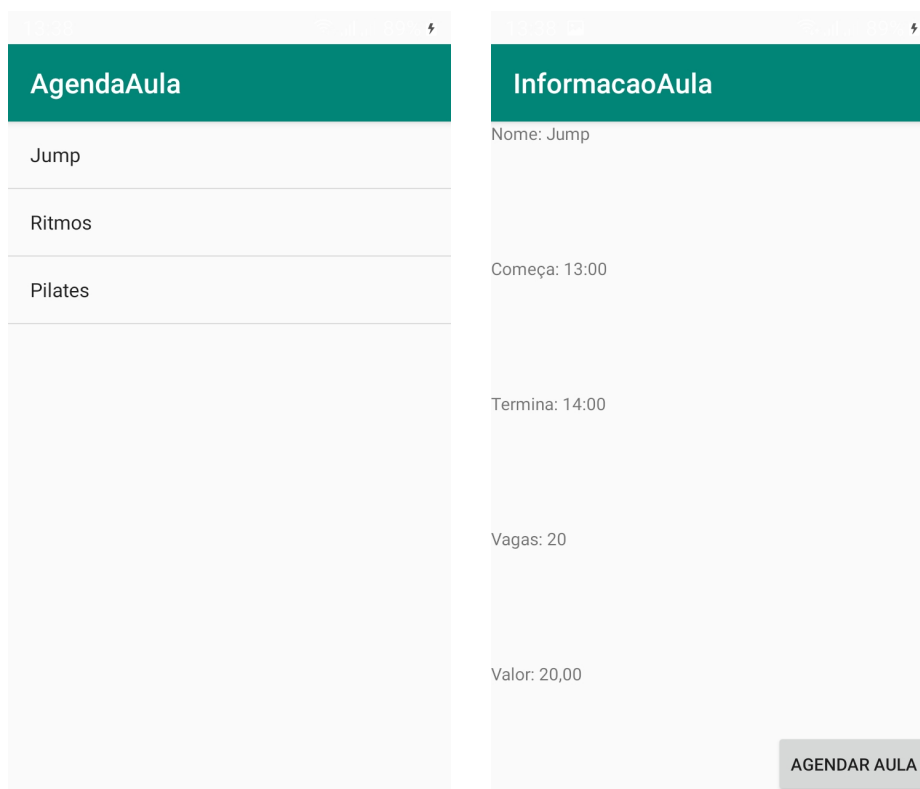


Figura 22 – Listar Aulas / Agendar Aula

Fonte – Autora

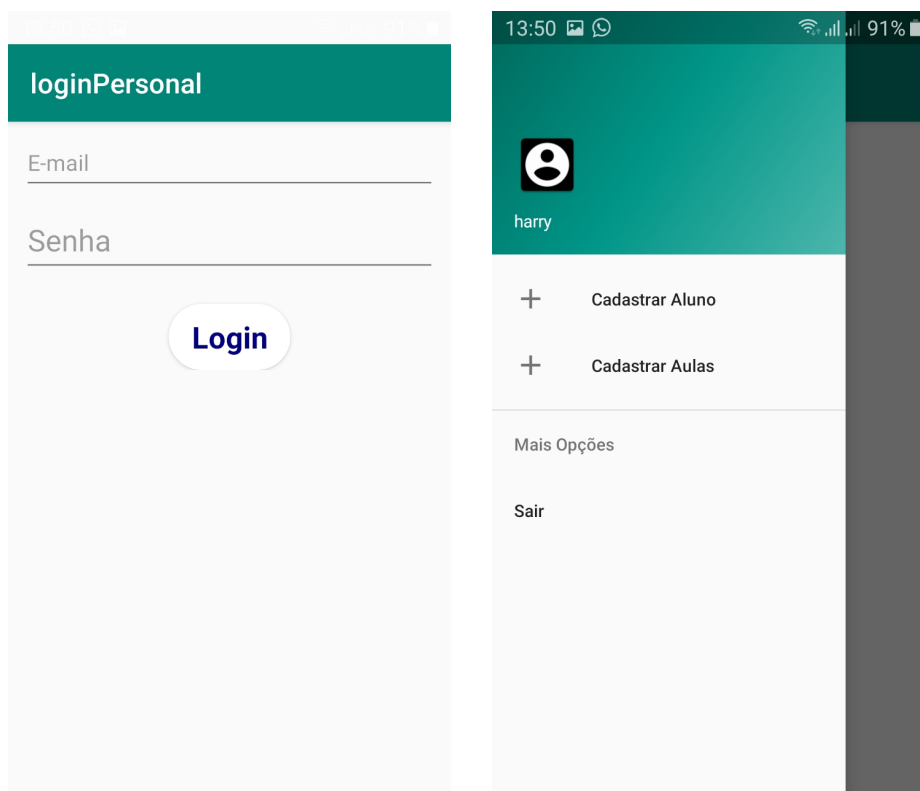


Figura 23 – Login Personal / Menu Aluno

Fonte – Autora

como mostrado na Figura 27. Ao clicar no botão de adicionar uma nova tela é inicializada contendo um formulário sobre o treino, conforme indicado na Figura 28. Ao selecionar um treino é possível cadastrar os exercícios deste treino, como mostra a Figura 29. Esse cadastro é preenchido no formulário do exercício, como mostra a Figura 30.

Ao selecionar cadastrar avaliação, é exibido uma lista com todas as avaliações do aluno, como mostrado na Figura 31. Ao clicar no botão de adicionar, uma nova tela é inicializada contendo um formulário sobre avaliação, conforme indicado na Figura 32. Ao selecionar uma avaliação, é possível cadastrar as medidas corporais do aluno como mostra a Figura 33. Esse cadastro é preenchido no formulário de medidas, como mostra a Figura 34.

Ao selecionar cadastrar aulas, é exibido uma lista com todas as aulas cadastradas pelo personal. Ao clicar no botão de adicionar uma nova tela é inicializada com um formulário solicitando informações sobre a aula, conforme indicado na Figura 35.

Ao identificar-se como academia, primeiro deve-se observar se a mesma possui cadastro. Caso não possua, é necessário clicar no botão cadastre-se e uma nova tela com um formulário é exibido como mostra a Figura 36. Após a efetivação do cadastro, é necessário realizar seu login, como mostra a Figura 37. Após efetuar o login, é exibido um menu lateral onde a academia poderá escolher entre cadastrar os *personais* e as modalidades, como mostra a Figura 38.

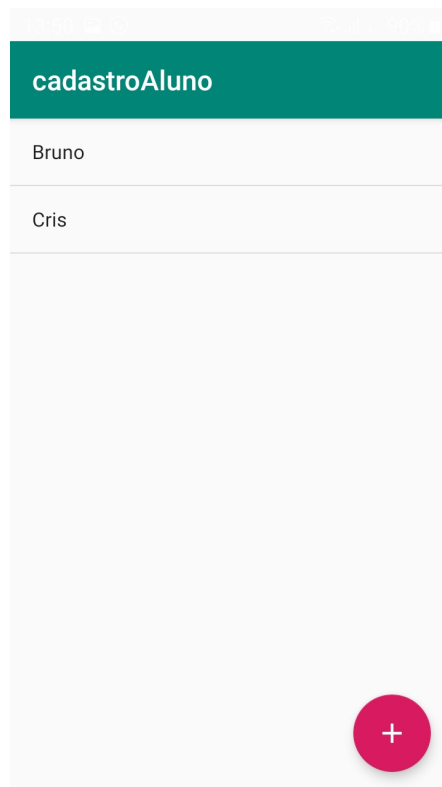


Figura 24 – Exibir Alunos

Fonte – Autora

A screenshot of a mobile application interface for a student registration form. The title bar is dark green with the text 'formularioAluno' and a white checkmark icon on the right. The form consists of several input fields, each with a light gray border and a label above it: 'Nome do Aluno', 'Idade do Aluno', 'Email', 'Objetivo', 'Endereco', 'Telefone', 'Pressão Arterial', 'Data de Nascimento', and 'Senha'. The fields are arranged vertically and are currently empty.

Figura 25 – Formulário de Cadastro para Alunos

Fonte – Autora

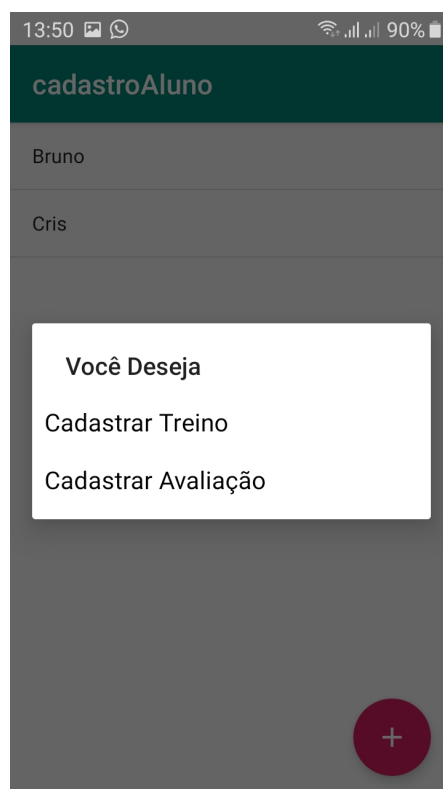


Figura 26 – Pop-up

Fonte – Autora

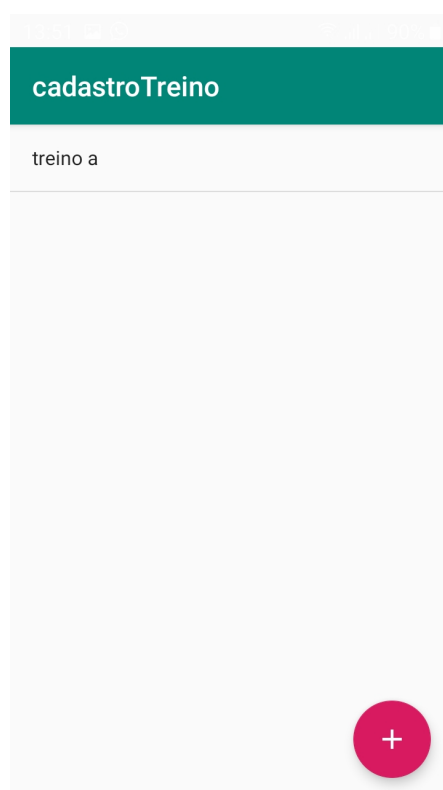


Figura 27 – Cadastrar Treino

Fonte – Autora



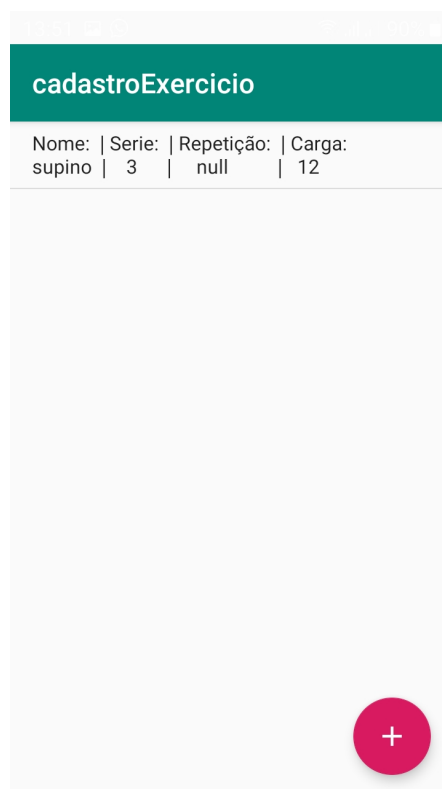
formularioTreino ✓

Nome do Treino

Validade do Treino

Figura 28 – Formulário de Cadastro Treinos

Fonte – Autora



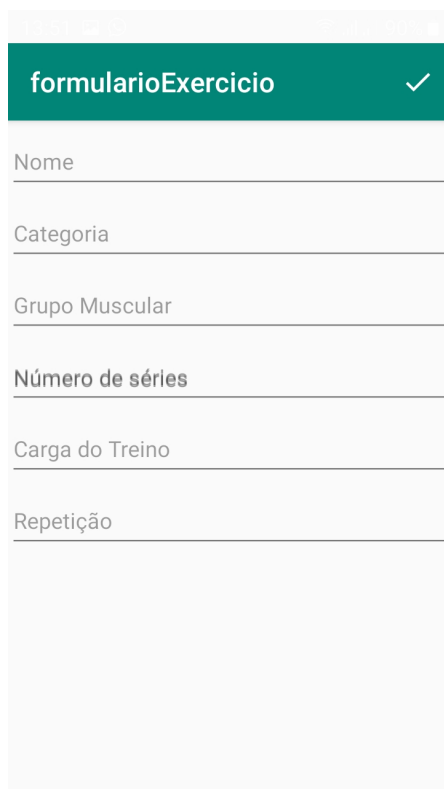
cadastroExercicio

Nome:	Serie:	Repetição:	Carga:
supino	3	null	12

+

Figura 29 – Cadastro de Exercícios

Fonte – Autora



formularioExercicio ✓

Nome

Categoria

Grupo Muscular

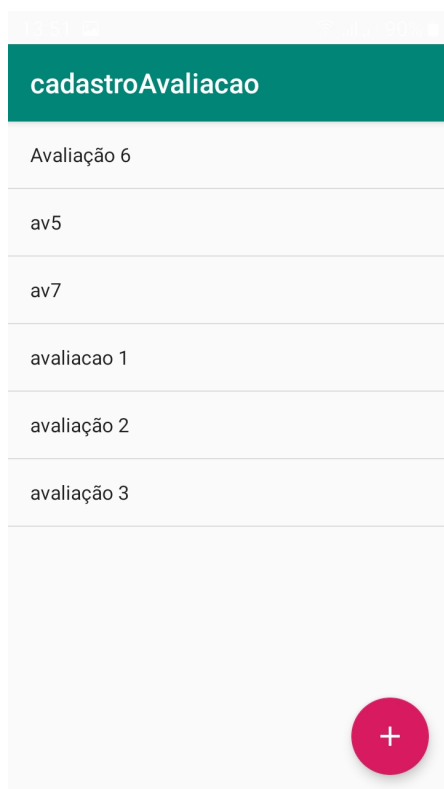
Número de séries

Carga do Treino

Repetição

Figura 30 – Formulário de Cadastro Exercicios

Fonte – Autora



cadastroAvaliacao

Avaliação 6

av5

av7

avaliacao 1

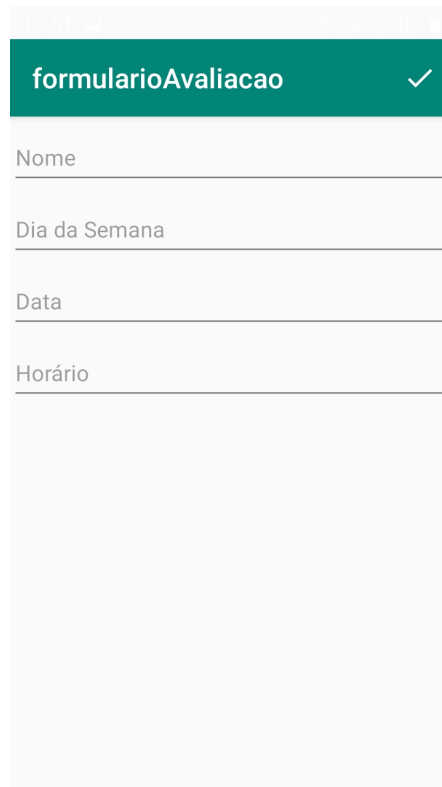
avaliação 2

avaliação 3

+

Figura 31 – Cadastrar Avaliacao

Fonte – Autora



formularioAvaliacao ✓

Nome

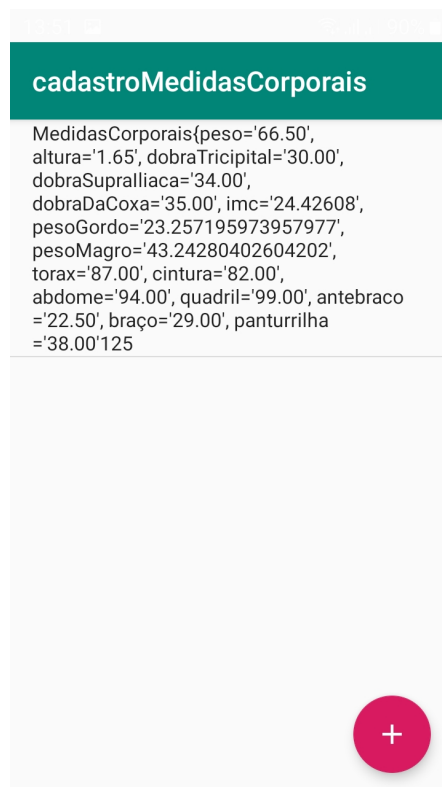
Dia da Semana

Data

Horário

Figura 32 – Formulário Cadastro Avaliacao

Fonte – Autora



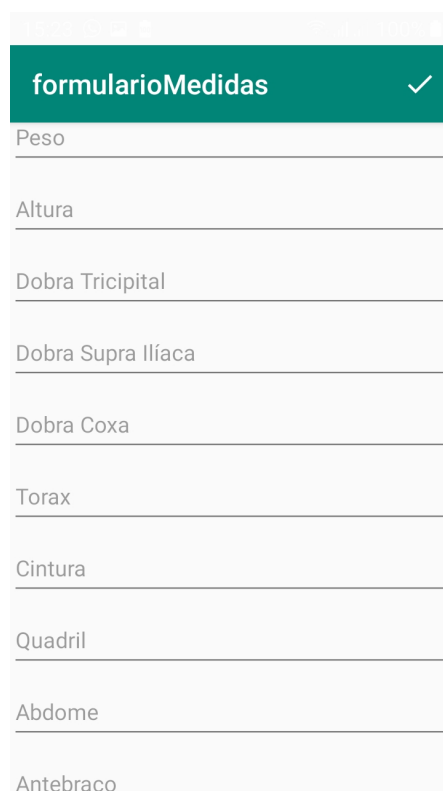
cadastroMedidasCorporais

MedidasCorporais(peso='66.50', altura='1.65', dobraTricipital='30.00', dobraSupralliacaa='34.00', dobraDaCoxa='35.00', imc='24.42608', pesoGordo='23.257195973957977', pesoMagro='43.24280402604202', torax='87.00', cintura='82.00', abdome='94.00', quadril='99.00', antebraço='22.50', braço='29.00', panturrilha='38.00')125

+

Figura 33 – Cadastrar Medidas

Fonte – Autora



formularioMedidas ✓

Peso

Altura

Dobra Tricipital

Dobra Supra Iliaca

Dobra Coxa

Torax

Cintura

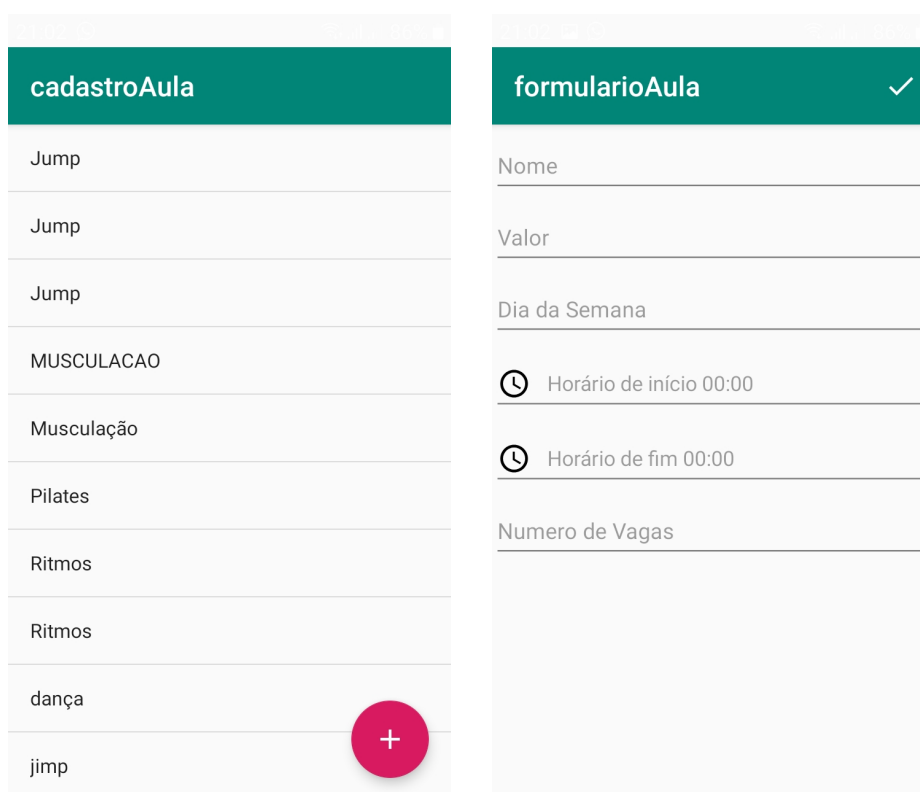
Quadril

Abdome

Antebraço

Figura 34 – Formulário Cadastro Medidas Corporais

Fonte – Autora



cadastroAula

Jump

Jump

Jump

MUSCULACAO

Musculação

Pilates

Ritmos

Ritmos

dança

jimp

+

formularioAula ✓

Nome

Valor

Dia da Semana

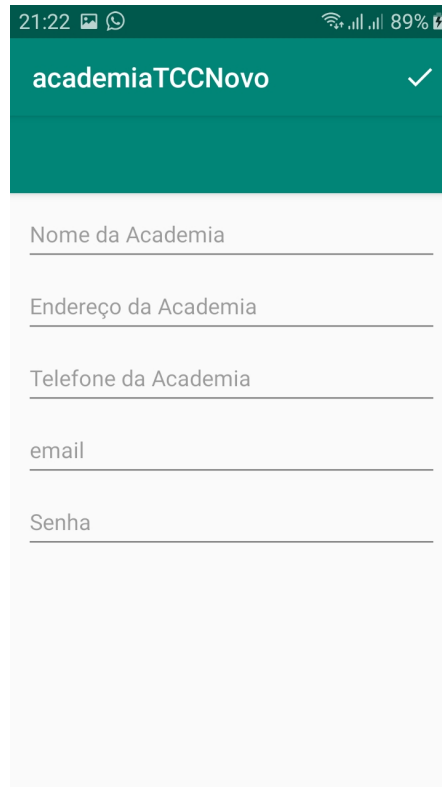
⌚ Horário de início 00:00

⌚ Horário de fim 00:00

Numero de Vagas

Figura 35 – Cadastro Aulas / Formulário Aulas

Fonte – Autora



21:22 89%

academiaTCCNovo ✓

Nome da Academia

Endereço da Academia

Telefone da Academia

email

Senha

Figura 36 – Cadastro Academias

Fonte – Autora

Ao selecionar exibir os *personais* cadastrados, uma nova tela é inicializada contendo uma lista com todos os *personais* da academia, como mostra a Figura 39, e ao clicar para adicionar um novo *personal*, um formulário é exibido como mostra a Figura 40.

Ao selecionar exibir modalidade, uma nova tela é inicializada contendo uma lista com todas as modalidades da academia, como mostra a Figura 41, e ao clicar para adicionar uma nova modalidade um formulário, é exibido como mostra a Figura 42.

Ao identificar-se como aluno não cadastrado, é exibido um menu lateral onde o usuário poderá escolher entre ver as academias e agendar aula Figura 43. Ao selecionar localizar academias através da API Google Maps, é possível exibir um mapa contendo a posição atual do usuário, como mostra a Figura 21. E por fim, é possível agendar as aulas de qualquer academia. as aulas são exibidas por dia da semana e por hora, ou seja, a cada hora é alterada a exibição, e ao escolher uma aula, é possível visualizar as informações como valores, quantidade de vagas e efetuar o agendamento, como mostra a Figura 22.

A diferença entre esse usuário e um aluno cadastrado é que o aluno cadastrado possui treinos e avaliações.

O aplicativo emite diariamente uma notificação, lembrando ao usuário de verificar se não há aulas disponíveis, como mostra a Figura 44.

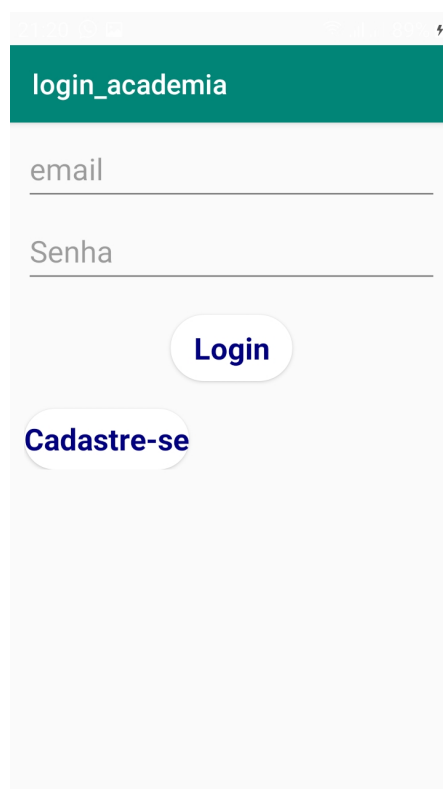


Figura 37 – Login Academia

Fonte – Autora

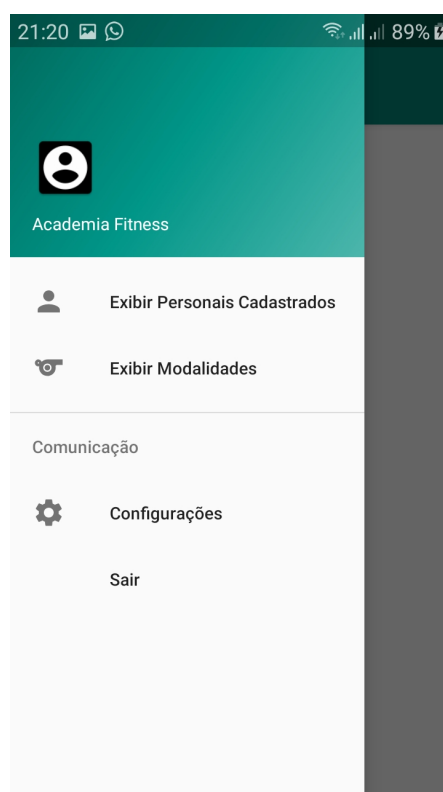


Figura 38 – Menu Academia

Fonte – Autora

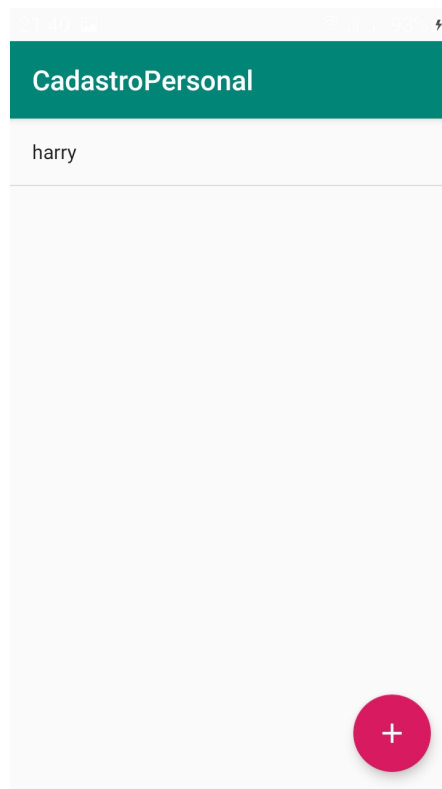


Figura 39 – Exibir Personal

Fonte – Autora

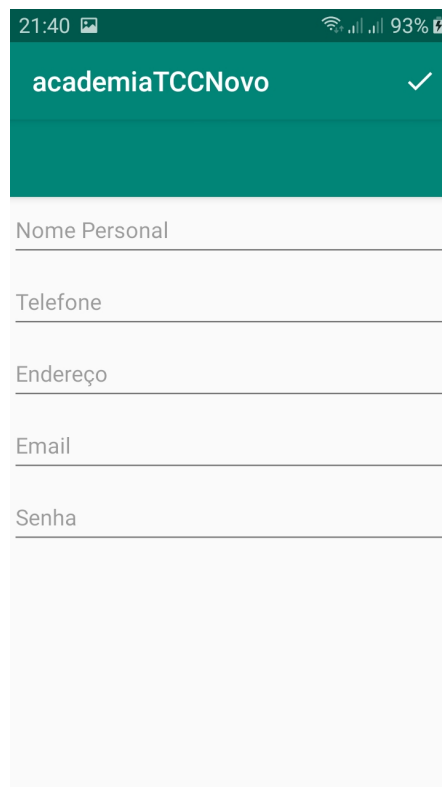


Figura 40 – Cadastrar Personal

Fonte – Autora

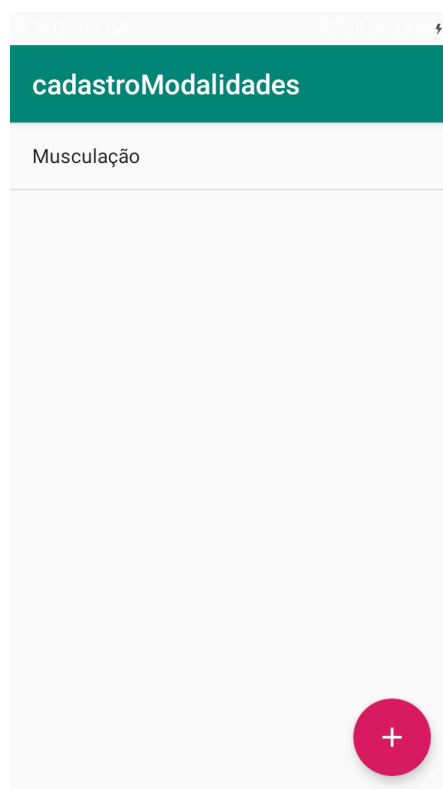


Figura 41 – Exibir Modalidades

Fonte – Autora

Nome da Modalidade
Descrição
Benefícios
Gasto Calórico
Trabalha quais grupos musculares
Recomendado para

Figura 42 – Formulário Cadastro Modalidades

Fonte – Autora

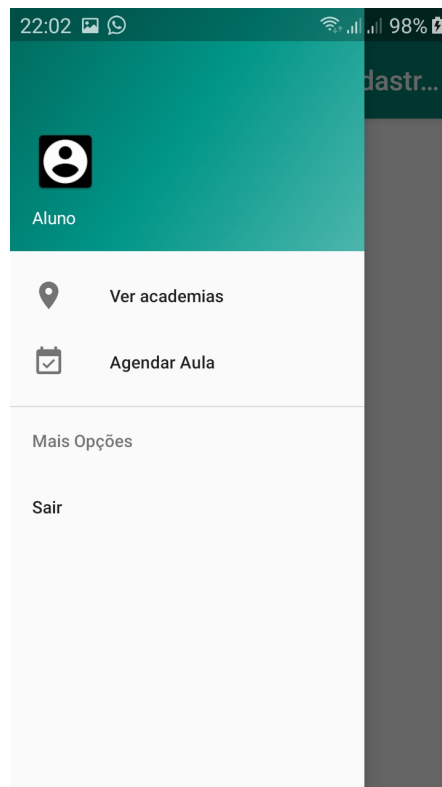


Figura 43 – Menu

Fonte – Autora

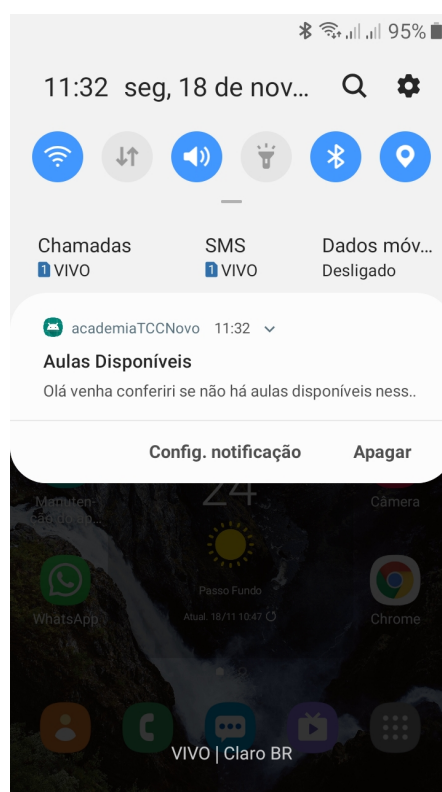


Figura 44 – Notificação

Fonte – Autora

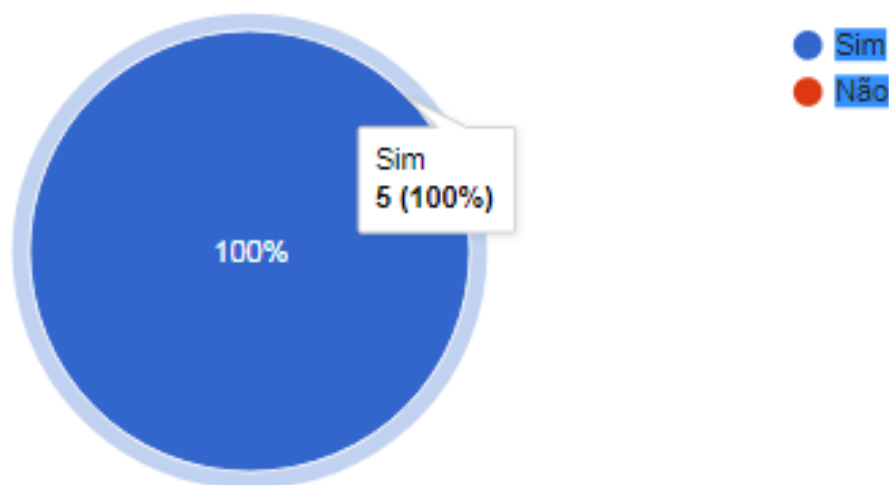


Figura 45 – Respostas da Pergunta 1

Fonte – Autora

5.2 Validação

Após a implementação foi disponibilizada uma versão do aplicativo e um formulário para avaliar a qualidade do mesmo, conforme modelo disponível no Apêndice B. O intuito da avaliação é verificar se o aplicativo atingiu o objetivo.

A avaliação contou com a participação de duas academias na cidade de Alegrete e o aplicativo foi utilizado por cinco usuários entre *personais* e alunos. Durante a execução eles puderam testar as funcionalidades livremente, os testes foram observados e quando houve dúvidas foram auxiliados. Ao final da execução foi solicitado que os usuários preenchessem um formulário. Com base nas questões, os resultados obtidos foram:

Para a pergunta “Você acha que o software estimula as pessoas à prática de exercícios físicos?” a resposta está exibida na Figura 45.

Para a pergunta “O que precisaria fazer para um software estimular a prática de exercícios?”, as respostas obtidas foram:

- “Assim como mostrar as aulas e horários disponíveis dar dicas sobre saúde e atividade física”;
- “A FACILIDADE COM QUE ELE TRAZ PARA SE TER ACESSO FÁCIL E COM AGILIDADES, ONDE POSSA IR DIRETO AOS PONTOS COM MUITA PRATICIDADE”;
- “Contar com notificações lembrando e estimulando seus usuários a fazerem as atividades propostas.”

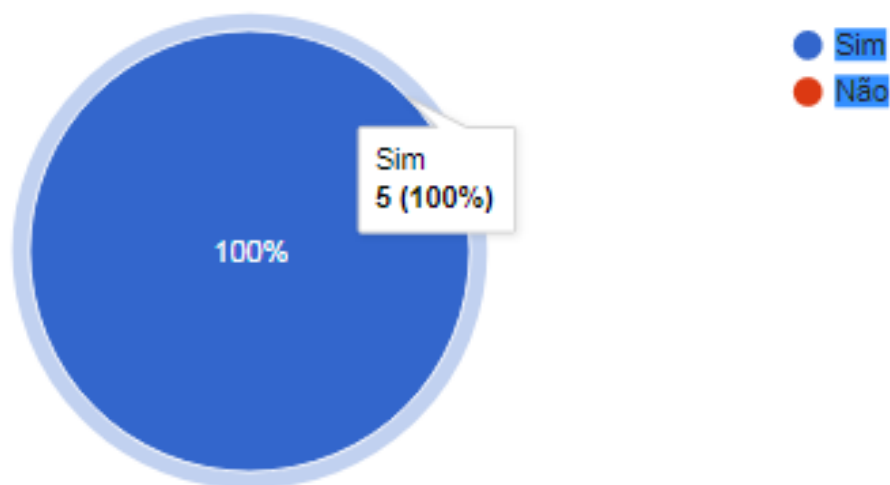


Figura 46 – Respostas da Pergunta 1

Fonte – Autora

Para as perguntas:

- "O Software atingiu os objetivos propostos?";
- "A usabilidade do software permite a fácil utilização do mesmo?";
- "Recomendaria o software para alguém?";
- "Usaria em sua academia ou com seus alunos?";
- "O aplicativo pode auxiliar a encontrar um horário ideal para praticar exercícios?"

os resultados das questões estão apresentadas na Figura 46.

Em sugestões de melhorias foram recomendadas:

- “Melhorar nas telas como é exibido as informações”;
- “FUTURAMENTE PODE SER MELHORADO A VISUALIZAÇÃO DA INTERFACE PARA FICAR MAIS CHAMATIVA. POR SER UM TRABALHO DE CONCLUSÃO DE CURSO ESTA MUITO BOM”
- “A possibilidade de inserção de imagens ou vídeos para os exercícios propostos, e talvez uma forma direta de interação personal/aluno, tipo um bate papo.”

Somente uma pessoa indicou que talvez o aplicativo não incentive a prática de exercícios.

Para a pergunta “Qual é a Perspectiva/Impacto do uso do software?”, as respostas obtidas foram:

- “Ter maior controle nas aulas e alunos que o professor e a academia possui, assim como saber o numero de procura pelas aulas. Para o aluno saber com mais rapidez as aulas e horários disponíveis.”;
- “Permite um controle dos alunos e das aulas ministradas, aperfeiçoando um trabalho muitas vezes manual.”;
- “MUITO BOA”;
- “Achei interessante, embora pra ficar mais completo seria interessante alguma forma de acrescentar imagens ou vídeos para os exercícios propostos.”;

Nenhuma defeito foi apontado. E o aplicativo obteve nota cinco pelos usuários.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento deste projeto possibilitou uma análise de aplicativos para o agendamento e gerenciamento de treinos de academias existentes. Foram avaliadas as ferramentas utilizadas no seu desenvolvimento. Além disso, foi feita uma pesquisa de campo para obter informações sobre como as academias da cidade do Alegrete utilizam *softwares* para auxiliarem em suas tarefas.

O desenvolvimento de um aplicativo para agendamento de aulas, torna-se útil devido o fato de muitas pessoas possuírem uma rotina agitada. Assim, o aplicativo pode auxiliar estas pessoas a escolherem aulas que adaptam-se a seus horários, organizando sua rotina de treinos, e incentivando a busca por uma vida mais saudável.

Após a validação foi possível observar que o aplicativo teve uma boa aceitação, onde as pessoas utilizariam em seu cotidiano, buscando conciliar seu dia a dia bastante agitado com uma rotina de exercícios saudáveis. Por esse motivo, o aplicativo atingiu seu objetivo.

Como trabalhos futuros a interface será reformulada, deixando ela mais prática de ser utilizada. Também serão geradas novas atualizações, permitindo a exibição de vídeos demonstrativos dos exercícios, efetivação do pagamento das aulas e exibição das academias próximas, tornando mais fácil o cotidiano e a viabilidade de uso do software para as pessoas. Será feita também a elaboração de um plano de negócio.

REFERÊNCIAS

- 4GYM. **4GYM Sistema para Academias e Studios**. 2019. Disponível em: <<https://4gym.com.br/funcionalidades>>. Acesso em: 17/05/2019. Citado na página 35.
- ADRIANO, T. S. **Introdução ao Firebase**. 2018. Disponível em: <<https://medium.com/@programadriano/introdu%C3%A7%C3%A3o-ao-firebase-bd59bfd03f29>>. Acesso em: 20/05/2019. Citado na página 45.
- ANDROIDDEVELOPERS. **Visão geral das notificações**. 2019. Disponível em: <<https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=pt-BR>>. Acesso em: 17/10/2019. Citado na página 29.
- ANDROIDSTUDIO. **Conheça o Android Studio**. 2018. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em: 31/05/2019. Citado na página 44.
- BENTO, A. R. et al. **Importância da Tecnologia no Acompanhamento das Atividades dos Alunos de Academia**. Berners-Lee, Tim et al. **Uniform Resource Locators (URL)**, 1994. Citado na página 25.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistema com UML**. Rio de Janeiro, RJ: Elsevier Editora Ltda., 2016. Citado na página 29.
- BLÄSING, T. et al. **An Android Application Sandbox System for Suspicious Software Detection**. In: IEEE. **2010 5th International Conference on Malicious and Unwanted Software**. [S.l.], 2010. p. 55–62. Citado na página 27.
- BOOCH, G.; JACOBSON, J.; RUMBAUGH, J. **Uml - Guia do Usuário, Tradução da Segunda Edição**. Rio de Janeiro: Elsevier Editora Ltda., 2017. Citado na página 29.
- BRITO, R. C. **Desenvolvimento Android utilizando a IDE Android Studio**. 2015. Disponível em: <<https://www.devmedia.com.br/desenvolvimento-android-utilizando-a-ide-android-studio/33872>>. Acesso em: 31/05/2019. Citado na página 44.
- BURTON, M.; FELKER, D. **Desenvolvimento de Aplicativos Android para Leigos**. Rio de Janeiro: Alta Books Editora, 2014. Citado na página 27.
- CAMBRIDGEDICTIONARY. **APP**. 2018. Disponível em: <<https://dictionary.cambridge.org/pt/dicionario/ingles/app>>. Acesso em: 21/05/2019. Citado na página 28.
- CAMBRIDGEDICTIONARY. **Mobile Device**. 2018. Disponível em: <<https://dictionary.cambridge.org/pt/dicionario/ingles/mobile-device>>. Acesso em: 21/05/2019. Citado na página 27.
- CAVALHEIRO, N. A. d. S. **SharpGym: Integração mobile x cloud, aplicada ao treinamento personalizado**. LUME Repositório Digital - UFRGS, 2013. Citado na página 37.

- CIRIACO, D. **Android cresce, iOS diminui e Windows Phone quase some, aponta relatório**. 2017. Disponível em: <<https://www.tecmundo.com.br/dispositivos-moveis/119411-android-ios-windows-phone-mercado.htm>>. Acesso em: 15/05/2019. Citado na página 43.
- COSTA, R. R. da; GONÇALVES, A. F. **Proposta de aplicativo para dispositivos móveis que auxilie os praticantes de musculação a controlarem a carga de seus exercícios**. III Seminário Empresarial e III Jornada de TI, 2015. Citado na página 36.
- CUELLO, J.; VITTONI, J. **Designing Mobile Apps**. [S.l.]: Smashing Media, 2013. Citado na página 28.
- DATE, C. **Introdução a Sistemas de Bancos de Dados**. Rio de Janeiro: Elsevier, 2004. Citado na página 28.
- DOGADO, A. K. S. **SYSACADEMY – Sistema de Gerenciamento de Academias**. Universidade Tecnológica Federal do Paraná, 2018. Citado na página 39.
- DOUGLAS, A. **Introdução à Google Maps API**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-a-google-maps-api/26967>>. Acesso em: 29/10/2019. Citado na página 30.
- DUARTE, C. **O que é Google APIs?**. 2019. Disponível em: <<https://maplink.global/blog/o-que-e-google-apis/>>. Acesso em: 31/10/2019. Citado na página 30.
- EDUMOBILE. **Módulo Dispositivos Móveis**. 2017. Disponível em: <http://www.nuted.ufrgs.br/oa/edumobile/m1_dm.html>. Acesso em: 22/04/2019. Citado na página 27.
- FANTINEL, L.; PARISOTO, W. **Sistema de Acompanhamento de Atividades Físicas–Droidfitness**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2012. Citado na página 38.
- FILHO, L. **Desenvolvendo seu primeiro aplicativo Android: Entre de cabeça no mundo dos aplicativos móveis, criando e publicando seu próprio programa para o sistema líder do mercado!** São Paulo: Novatec Editora, 2017. Citado na página 27.
- FIREBASE. **Sobre as mensagens do FCM**. 2019. Disponível em: <<https://firebase.google.com/docs/cloud-messaging/concept-options?hl=pt-PT>>. Acesso em: 22/10/2019. Citado na página 29.
- GLAUBER, N. **Dominando o Android com Kotlin**. NOVATEC, 2019. ISBN 9788575227268. Disponível em: <<https://books.google.com.br/books?id=JEeMDwAAQBAJ>>. Citado na página 29.
- GOBODY. **O mais completo Aplicativo Fitness disponível a qualquer hora e em qualquer lugar**. . 2018. Disponível em: <<https://www.gobody.com.br/#cont-3>>. Acesso em: 29/05/2019. Citado na página 39.

GOMES, R. C. et al. **Sistema Operacional Android**. Universidade Federal Fluminense, 2012. Citado na página 27.

JAVA. **O que é a Tecnologia Java e porque preciso dela?** . 2018. Disponível em: <https://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em: 26/04/2019. Citado na página 44.

JR, A. F. de M.; GRANZOTO, I. S.; PIRES, D. F. **SISTEMA DE GERENCIAMENTO DE ACADEMIAS DE GINÁSTICA**. *Revista Eletrônica de Sistemas de Informação e de Gestão Tecnológica*, v. 3, n. 1, 2013. Citado na página 35.

LECHETA, R. R. **Android Essencial: Edição resumida do livro Google Android**. São Paulo: Novatec Editora, 2016. Citado na página 28.

MARCONI, M. d. A.; LAKATOS, E. M. et al. **Técnicas de Pesquisa**. São Paulo: Atlas, 2002. v. 2. Citado na página 31.

MELO, A. **Desenvolvendo Aplicações com UML 2.2**. Rio de Janeiro, RJ: BRASPORT, 2010. Citado na página 29.

MESSIAS, V. H. D. S. **FITGEN: um Aplicativo Móvel de Apoio à manutenção de Treinos e Dietas**. **CONGRESSO DE COMPUTAÇÃO - UNIARA**, 2015. Citado na página 38.

MICROSOFTDOCS. **Mensagens na nuvem do firebase** . 2018. Disponível em: <<https://docs.microsoft.com/pt-br/xamarin/android/data-cloud/google-messaging/firebase-cloud-messaging>>. Acesso em: 30/10/2019. Citado na página 30.

MOLINARI, L. D. M. R. **Testes de Aplicações Mobile – Qualidade e Desenvolvimento em Aplicativos Móveis**. São Paulo, SP: Editora Saraiva, 2018. Citado na página 27.

MONTEIRO, M.; HARLEY, M. et al. **Sistema de gerenciamento e controle para Academias-SGCA**. *Sistemas de Informação & Gestão de Tecnologia.*, n. 1, 2010. Citado na página 36.

OLIVEIRA, A. L. M. D. **Sistema para Academias de Ginásticas em Android**. **Fundação Educacional do Município de Assis – FEMA – Assis**, 2013. Citado na página 35.

ORLANDI, C. **Firestore: serviços, vantagens, quando utilizar e integrações**. 2019. Disponível em: <<https://blog.rocketseat.com.br/firebase/>>. Acesso em: 23/04/2019. Citado 2 vezes nas páginas 45 e 46.

PARREIRAS, C. **O que é Gympass: saiba como funciona o serviço de academia**. 2018. Disponível em: <<https://www.techtudo.com.br/noticias/2018/08/o-que-e-gympass-saiba-como-funciona-o-servico-de-academia.ghtml>>. Acesso em: 05/06/2019. Citado na página 40.

PEREIRA, L.; SILVA, M. D. **Android para Desenvolvedores**. Rio de Janeiro, RJ: BRASPORT, 2009. Citado na página 27.

- PERRY, S. J. **Fundamentos da linguagem Java** . 2016. Disponível em: <<https://www.ibm.com/developerworks/br/java/tutorials/j-introjava1/index.html>>. Acesso em: 31/05/2019. Citado na página 44.
- QUEIROGA, E. M. **SISACAD Sistema Gerenciador de Academias Desportivas . Universidade Católica de Pelotas**, 2012. Citado na página 37.
- RAMAKRISHNAN, R.; GEHRKE, J. **Database Management Systems**. New York, NY: McGraw Hill, 2000. Citado na página 28.
- RIBINSKI, J. **Sistema para acompanhamento das avaliações referentes à prática de exercícios físicos**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2014. Citado na página 39.
- SALDIT. **Justinmind Prototyper!** 2017. Disponível em: <<http://licenciamentodesoftware.com.br/justinmind-prototyper/>>. Acesso em: 30/03/2019. Citado na página 44.
- SANTOS, J. M. M. M. d. **Push Notifications: A Influência das Notificações no Processo de Compra**. Tese (Doutorado) — Instituto Superior de Contabilidade e Administração do Porto - Instituto Politécnico do Porto, 2016. Citado na página 29.
- SCHILD, H. **Java para iniciantes**. Porto Alegre, RS: BOOKMAN EDITORA LTDA, 2013. Citado na página 44.
- SILVA, L. L. B.; PIRES, D. F.; NETO, S. C. **Desenvolvimento de Aplicações para Dispositivos Móveis: Tipos e Exemplo de Aplicação na plataforma iOS**. II Workshop de Iniciação Científica em Sistemas de Informação, Goiânia, Go, 2015. Citado na página 25.
- SISTEMASCA. **Sistema SCA**. 2019. Disponível em: <<https://www.sistemasca.com/>>. Acesso em: 27/05/2019. Citado na página 40.
- SOFTWARE.COM. **Enterprise Architect**. 2018. Disponível em: <<https://software.com.br/p/enterprise-architect>>. Acesso em: 02/04/2019. Citado na página 45.
- SOUZA, A. A. **SISTEMA DE GERENCIAMENTO PARA ACADEMIA. Fundação Educacional do Município de Assis – FEMA – Assis**, 2011. Citado na página 39.
- SPADARI, A. **Como se localizar com a tecnologia de Geolocalização** . 2016. Disponível em: <<https://br.ccm.net/faq/14926-como-se-localizar-com-a-tecnologia-de-geolocalizacao>>. Acesso em: 02/11/2019. Citado na página 30.
- TECNOFIT. **Tecnofit vai deixar a gestão da sua academia em boa forma !** 2018. Disponível em: <<https://www.tecnofit.com.br/>>. Acesso em: 02/05/2019. Citado na página 38.
- VIDIRI, J. P. S.; MARQUES, D. **FitDolphin: Aplicativo para controle de atividades físicas para plataforma Android**. Instituto Federal de São Paulo, 2018. Citado na página 36.

Apêndices

APÊNDICE A – QUESTIONÁRIO APLICADO NAS ENTREVISTAS

Levantamento de Requisitos

A academia possui sistema para cadastros de alunos, treinos, etc?

Sim

Não

A academia possui aplicativo próprio?

Sim

Não

Quais funcionalidades deveria ter um aplicativo?

Sua resposta

ENVIAR

Figura 47 – Formulário Aplicado nas Entrevistas

APÊNDICE B – QUESTIONÁRIO AVALIATIVO DO APLICATIVO

Avaliação do Aplicativo

Este formulário é para saber se o aplicativo desenvolvido atende as necessidades propostas

***Obrigatório**

Você acha que o software estimula as pessoas à prática de exercícios físicos? *

Sim

Não

O que precisaria fazer para um software estimular os exercícios?

Sua resposta

Software atingiu os objetivos propostos?

Sim

Não

Figura 48 – Formulário Aplicado na Validação

A usabilidade do software permite a fácil utilização do mesmo?

Sim

Não

Talvez

Qual é a Perspectiva/Impacto do uso do software ?

Sua resposta _____

Recomendaria o software para alguém?

Sim

Não

Talvez

Figura 49 – Formulário Aplicado na Validação

Usaria em sua academia ou com seus alunos?

Sim

Não

Talvez

Sugestão de melhorias.

Sua resposta _____

Apontamento de defeitos

Sua resposta _____

aplicativo pode auxiliar a encontrar um horário ideal para praticar exercícios?

Sim

Não

Talvez

Figura 50 – Formulário Aplicado na Validação

O aplicativo para controle de desempenho em academias pode incentivar alunos a realizar atividades físicas?

Sim

Não

Talvez

Que nota você daria ao aplicativo

1

2

3

4

5

Enviar

Figura 51 – Formulário Aplicado na Validação

ÍNDICE

API, 23, 27, 30, 61, 70

App., 25, 28, 32, 33, 35, 36, 38–40, 43, 45

BaaS, 45

CSS, 37, 40

FCM, 29, 30

HTML, 37, 40

IDE, 44

IMC, 35, 37

JSP, 37

NeoDatis ODB, 38

PHP, 38–40

SaaS, 38

SGBD, 28

SGCA, 36, 37

UML, 11, 29

XML, 45