
**Fit Measure: Desenvolvimento de um
aplicativo para avaliação física utilizando
Flutter e MongoDB Atlas**

Rafael Tadeu da Silva

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 20/11/2023

Assinatura: _____

Fit Measure: Desenvolvimento de um aplicativo para avaliação física utilizando Flutter e MongoDB Atlas

Rafael Tadeu da Silva

Márcio Lopes Júnior

Monografia apresentada ao Centro Universitário de Formiga UNIFOR/MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof^o. Márcio Lopes Júnior

Unifor-MG - Formiga

20/11/2023

*Dedico este trabalho a Deus,
minha mulher e filho, meu
pai, minha mãe e meus ir-
mãos, sem vocês não conse-
guiria*

Agradecimentos

Este momento tão aguardado enfim chegou, a conclusão deste trabalho e do curso, apesar de todos os imprevistos e dificuldades que não somente eu, mas o mundo como um todo enfrentou. Neste momento me recordo de cada um que me ajudou nesta fase, sem sua ajuda não teria conseguido.

Em primeiro lugar, agradeço a Deus, pela saúde, força e oportunidades que me deu ao longo dos anos.

Agradeço especialmente ao meu pai Sebastião que tanto apoio me deu, me ensinou a trabalhar e a lutar pelos meus objetivos e a minha mãe Sônia que sempre cuidou de mim e se preocupa com meu bem estar. Meus irmãos José e Jéssica, os quais estimo muito e contribuíram cada qual a sua maneira para me incentivar.

A minha mulher Elisângela e meu filho João, pelo amor, carinho, paciência e ajuda durante mais essa jornada e juntos vamos nos preparar para a próxima.

Ao meu orientador Márcio, que apesar de suas grandes responsabilidades e compromissos, sempre esteve disposto a me ensinar e guiar durante estes anos da faculdade. Agradeço ainda aos muitos professores e demais profissionais do corpo docente que tornaram possível esta conquista.

Agradeço também aos meus colegas de curso e de trabalho com os quais enfrentamos e superamos juntos os desafios diários e com isso nos tornamos melhores profissionais.

A todos que direta ou indiretamente contribuíram com minha formação, muito obrigado!

“Toda tecnologia suficientemente avançada é indistinguível de mágica.”

Arthur C. Clarke

Sumário

Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos do Trabalho	2
1.3 Justificativa	2
1.4 Estrutura da Monografia	2
2 Referencial Teórico	3
2.1 Antropometria: definição e importância	3
2.2 Ferramentas para medições antropométricas e como usá-las	4
2.2.1 Massa corporal	4
2.2.2 Estatura	5
2.2.3 IMC - Índice de massa corporal	5
2.2.4 Perímetro	6
2.2.5 Índice de conicidade	7
2.2.6 Dobras cutâneas	7
2.3 Programação de dispositivos móveis	11
2.3.1 Linguagem de programação - Dart	12
2.3.2 Framework Frontend - Flutter	12
2.3.3 Escolha do IDE	14
2.3.4 Banco de dados	14
2.3.5 API Rest	15
2.3.6 Padrão de Arquitetura MVC	16
3 Estudo da Arte	17
3.1 PhisiCalc	17

3.2	Fitnezz.io - Avaliação Física	18
3.3	Avaliação Física PRO	18
4	Metodologia e Desenvolvimento	20
4.1	Levantamento de Requisitos	20
4.2	Diagrama de caso de uso	21
4.3	Desenvolvimento da aplicação	22
4.3.1	Criação do Banco de dados	22
4.3.2	Interface	23
4.3.3	Conexão entre banco de dados e aplicação	23
4.3.4	Autenticação do usuário	25
4.3.5	Recuperação de senha	25
4.3.6	Perfil de usuário e aluno	26
4.3.7	Calculadora de composição corporal	26
4.3.8	Visualização de relatórios	27
4.3.9	Compartilhamento de relatórios	27
5	Funcionamento e Resultados	28
5.1	Apresentação do sistema	28
5.2	Telas de carregamento e login	28
5.3	Tela de cadastro de usuário	28
5.4	Telas de recuperação de senha	30
5.5	Tela menu	30
5.6	Tela perfil do usuário	30
5.7	Telas de lista e perfil de aluno	32
5.8	Tela calculadora de medidas	32
5.9	Tela de relatórios	33
5.10	Compartilhamento de relatórios	35
5.11	Publicação na <i>PlayStore</i>	36
6	Conclusões	37
6.1	Considerações Finais	37
6.2	Trabalhos Futuros	38
6.2.1	Acesso dos alunos	38
6.2.2	Login com o Firebase e redes sociais	38
6.2.3	Novas formas de cálculo de medidas corporais	38
6.2.4	Criar versões para outras lojas de aplicativos	38

Lista de Figuras

2.1	Balança	4
2.2	Medidas dos perímetros de cintura(A) e quadris(B)	6
2.3	Perfil morfológico associado a um duplo cone e a um cilindro direcionado à descrição da distribuição da gordura corporal.	7
2.4	Figura Compassos específicos para medida de espessura de dobras cutâneas. A. Lange, B. Harpeden, C. Sanny e D. Cescorf.	8
2.5	A a C. Medidas de espessura de dobras cutâneas.	8
2.6	Ciclo de vida de um <i>Widget</i>	13
2.7	Padrão MVC	16
3.1	Telas do aplicativo PhisiCalc	17
3.2	Telas do aplicativo Fitnezz.io	18
3.3	Telas do aplicativo Avaliação Física PRO	19
4.1	Diagrama de casos de uso	21
4.2	Exemplo de documentos do banco de dados	22
4.3	Conexão com o banco de dados através do mongo_dart	24
4.4	Método main - tentativa de conexão	25
4.5	Método para resetar senha do usuário	26
4.6	Método para criar e compartilhar os relatórios em PDF	27
5.1	Telas iniciais do Fit Measure	29
5.2	Tela de Cadastro de Usuário	29
5.3	Telas de Recuperação de Senha	30
5.4	Telas de Menu e Diálogo Deslogar	31
5.5	Tela de Perfil do Usuário	31
5.6	Telas de Menu Alunos	32
5.7	Exemplo de Cálculos de Medidas	33
5.8	Telas de Relatório	34

5.9	Tela de Relatório de Porcentagem de Gordura	34
5.10	Exportando relatório	35
5.11	Página download aplicativo Fit Measure na <i>PlayStore</i>	36

Lista de Tabelas

2.1	Índice de massa corporal	5
2.2	Tabela referência Índice Cintura/Quadril	6
2.3	Tabela Mulheres - Referência porcentagem gordura corporal.	9
2.4	Tabela Homens - Referência porcentagem gordura corporal.	10

Lista de Siglas

ACID - *Atomicidade, Consistência, Isolamento e Durabilidade*

AOT - *Ahead-of-time - Antes do tempo*

API - *Application Programming Interface - Interface de Programação de Aplicação*

CRUD - *Create, Read, Update, Delete - Criar, Consultar, Atualizar, Deletar*

CSS - *Cascading Style Sheets - Folhas de Estilo em Cascata*

HTML - *HyperText Markup Language - Linguagem de Marcação de Hipertexto*

HTTP - *Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto*

IDE - *Integrated Development Environment - Ambiente de Desenvolvimento Integrado*

IMC - *Índice de Massa Corporal*

JIT - *Just-in-time - No momento certo*

JSON - *JavaScript Object Notation - Notação de Objeto JavaScript*

MVC - *Model-View-Controller - Modelo-Visão-Controlador*

NoSQL - *Not only SQL - Não apenas SQL*

REST - *Representational State Transfer - Transferência de Estado Representacional*

SDK - *Software Development Kit - Kit de Desenvolvimento de Software*

SQL - *Structured Query Language - Linguagem de Consulta Estruturada*

UI - *User Interface - Interface de Usuário*

UML - *Unified Modeling Language - Linguagem de Modelagem Unificada*

Resumo

SILVA, R. T. *Desenvolvimento de um aplicativo para avaliação física utilizando Flutter e MongoDB Atlas*. Monografia (Graduação) — Centro universitário de Formiga – Unifor-mg – Formiga, 2023.

A preocupação com as medidas antropomórficas cresce entre médicos e pesquisadores, pois afetam diretamente a saúde de seus pacientes. Existem diversas formas e ferramentas para fazer estas avaliações. Com o avanço tecnológico e alta aceitação dos celulares com Android, estes começaram a ser utilizados para as mais diversas finalidades, substituindo os computadores pessoais por conta de sua grande mobilidade. Apesar disso, criar aplicativos para Android pode levar muito tempo principalmente utilizando a linguagem padrão para desenvolvimento nesta plataforma, a linguagem Java, por isso a Google criou um *framework* chamado Flutter, que permite criar aplicativos para Android e outras plataformas com uma única base de código agilizando o processo. Além disso, é necessário guardar as informações fornecidas sobre os usuários, para isso podemos utilizar o banco de dados NoSQL *MongoDB Atlas*, o qual não precisa de gerenciamento e conta com alta disponibilidade. Este estudo visa demonstrar se é fácil criar aplicativos com as ferramentas descritas e se este aplicativo pode facilitar a vida dos profissionais que fazem a avaliação física

Palavras-chave: Avaliação Antropomórfica, Desenvolvimento de Aplicativos, Android, MongoDB Atlas, Dart, Flutter

Abstract

SILVA, R. T. *Development of an application for physical assessment using Flutter and MongoDB Atlas*. Monografia (Graduação) — Centro universitário de Formiga – Unifor-mg – Formiga-MG, 2023.

Concern about anthropomorphic measurements is growing among doctors and researchers, as they directly affect the health of their patients. There are several ways and tools to carry out these assessments. With technological advancement and high acceptance of Android cell phones, they began to be used for the most diverse purposes, replacing personal computers due to their great mobility. Despite this, creating applications for Android can take a long time, especially using the standard language for development on this platform, the Java language, which is why Google created a framework called Flutter, which allows you to create applications for Android and other platforms with a single code base. speeding up the process. Furthermore, it is necessary to store the information provided about users, for this we can use the NoSQL MongoDB Atlas database, which does not require management and has high availability. This study aims to verify how easy it is to create applications with the tools described and that this application can make the lives of professionals who carry out physical assessments easier.

Keywords: Anthropomorphic Assessment, Application Development, Android, MongoDB Atlas, Dart, Flutter

Introdução

1.1 Considerações Iniciais

A obesidade é um dos mais graves problemas de saúde que a humanidade deve enfrentar, é o que afirma a Organização Mundial da Saúde. Estima-se que ao redor do mundo em 2025, serão 2,3 bilhões de adultos acima do peso e destes 700 milhões estarão com obesidade. (METABÓLICA, 2023)

De acordo com a Pesquisa de Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico (Vigitel) de 2019, houve um aumento de 72% de pessoas com sobrepeso ou obesidade quando comparada com uma outra pesquisa feita em 2006, onde 11,8% tinham este problema de saúde, chegando a 20,3% dos entrevistados em 2019. (METABÓLICA, 2023)

A antropologia, ciência que estuda o ser humano em sua totalidade, possui uma área que estuda as dimensões e proporções do corpo humano, a antropometria. Esta área é responsável por estudar as diversas dimensões do corpo, como estatura e peso corporal. O peso do corpo pode ser dividido em gordura e massa magra e existem diferentes maneiras de descobrir estes valores. Para que sejam criados planos de tratamento eficazes é necessário ter um conhecimento das ferramentas de avaliação antropométricas.(ANDRADE et al., 2018)

Os procedimentos laboratoriais são a primeira opção para a análise da composição corporal já que oferecem estimativas bastante precisas sobre a quantidade de gordura e massa magra no corpo de um indivíduo, mas sua utilização é limitada por conta dos altos custos dos equipamentos, da sofisticação metodológica e das dificuldades de se preparar os avaliados para a realização destes procedimentos.(RIBEIRO; MELO; TIRAPÉGUI, 2018) Por isso, na grande maioria das avaliações são usadas técnicas antropométricas que utilizam equipamentos mais baratos e fáceis de usar, como a fita métrica e o compasso. (ANDRADE et al., 2018)

Além dos equipamentos utilizados para descoberta das medidas antropométricas, podemos

utilizar outros recursos tecnológicos para ajudar a calcular estas medidas para os profissionais interessados como pesquisadores e médicos, como computadores e celulares.

Dentre os celulares, as vendas de dispositivos Android e os downloads de aplicativos estão crescendo exponencialmente. A primeira geração de telefones Android foi lançada em outubro de 2008. Em 2015, já existiam mais de 1 bilhão de smartphones e tablets Android em uso, e mais de 1,5 milhão de aparelhos Android sendo ativados diariamente. (DEITEL; DEITEL, 2015)

Além de ajudar a fazer os cálculos, os aplicativos de smartphones podem armazenar as informações inseridas em bancos de dados para que possam recuperar estas informações novamente em outro momento. Esses bancos de dados podem ser locais, sendo acessíveis apenas do aparelho onde foram armazenados, ou em servidores remoto. (SIMAS et al., 2019)

1.2 Objetivos do Trabalho

O objetivo geral deste projeto é criar um aplicativo que realize os diferentes cálculos antropométricos e apresente os resultados de forma simples através de gráficos. Para alcançar este objetivo, foram projetados os seguintes objetivos específicos:

- Criação de um banco de dados que vai armazenar os dados dos usuários, tanto as medidas de entrada como os resultados separados por datas para que possam ser acessados e comparados futuramente.
- Desenvolvimento de um aplicativo que vai receber as informações digitadas pelo usuário realizar os diversos cálculos, armazenar no banco de dados e mostrar os resultados utilizando gráficos e outros recursos.
- Os resultados e gráficos são capazes de mostrar as mudanças quando comparadas duas medições realizadas em períodos distintos.

1.3 Justificativa

Para que as pessoas possam prevenir esta condição ou ainda direcionar o tratamento de pacientes acima do peso ou obesas, deve-se acompanhar seu peso e reconhecer tal problema. (METABÓLICA, 2016) A presente proposta visa aproveitar os avanços tecnológicos para facilitar, agilizar e automatizar a aplicação do conhecimento antropomórfico agregado até então, a fim de auxiliar os profissionais da saúde a diagnosticarem problemas antropomórficos ou avaliadores físicos a decidir sobre quais ajustes fazer nos treinos para obter os melhores resultados.

1.4 Estrutura da Monografia

Para concluir o objetivo acima proposto, vamos apresentar estudos relacionados a cada uma das ferramentas e tecnologias utilizadas, dividindo em: seção 2.1 definição e importância de antropometria, 2.2 ferramentas de medida antropométricas e como utilizá-las, seção 2.3 programação de dispositivos móveis. Na seção 3 veremos exemplos de programas similares, e na seção 4 as etapas e metodologias para a construção do aplicativo. Na seção 5 explicamos o funcionamento do sistema.

Referencial Teórico

2.1 Antropometria: definição e importância

A antropometria pode ser definida como “uma área da antropologia que estuda o tamanho, as dimensões e as proporções do corpo humano”.(ANDRADE et al., 2018)

A origem do termo antropometria vem da junção de dois termos gregos ANTHROPO que significa “homem” e METRY que se refere a “medida”.(VELHO et al., 1993 apud PETROSKI, 2007)

Segundo Andrade et al. (2018), “a estrutura corporal é formada pelo peso do corpo, pela estatura, pelas circunferências e pelos diâmetros, sendo que quando são encontradas alterações em relação à média da população normalmente estão relacionadas com diversas disfunções corporais”.

Petroski (2007) explica que as medidas de estatura, comprimentos e alturas obtidas pelas técnicas de medição antropométricas podem ser utilizadas para determinar a média da estatura, comprimento e altura de uma população, o que pode ser utilizado para melhorar a criação dos diversos equipamentos e utensílios usados pela humanidade, visando adaptar as medidas destes para atender não apenas a maioria das pessoas que os utilizam hoje, mas também dos futuros usuários.

Andrade et al. (2018) explana sobre como a avaliação antropométrica permite obter informações sobre a saúde do paciente, as quais podem determinar quadros de sobrepeso, obesidade ou desnutrição e a necessidade de encaminhamento para outros médicos, como por exemplo um paciente com sobrepeso, deve ser acompanhado por um nutricionista.

2.2 Ferramentas para medições antropométricas e como usá-las

Para Ribeiro, Melo e Tirapegui (2018), é possível obter de forma precisa as medidas da composição corporal utilizando equipamentos e procedimentos laboratoriais, mas a complexidade destes métodos de avaliação e o alto custo dos equipamentos fazem que sejam usados apenas em uma pequena parcela dos avaliados. Essas dificuldades abrem espaço para a utilização das técnicas antropométricas, as quais exigem equipamentos relativamente mais acessíveis e de simples utilização.

De acordo com Andrade et al. (2018), a avaliação antropométrica pode ser realizada de forma segura e rápida, utilizando equipamentos que ocupam pouco espaço e possuem baixo custo. Alguns dos equipamentos mais utilizados são a fita métrica, o compasso, o adipômetro e a balança. Cada ferramenta permite aferir diferentes medidas corporais e o profissional deve saber utilizá-las de forma precisa para conseguir obter os dados que devem ser anotados para serem utilizados em determinados cálculos que fazem estimativas de percentual de gordura corporal.

2.2.1 Massa corporal

Petroski (2007) esclarece que a massa corporal expressa o volume corporal de um indivíduo, sendo igual a soma das massas orgânicas e inorgânicas de seu corpo. Essa medida antropométrica pode ser usada para descobrir como está o processo de crescimento de uma pessoa e servir como indicador de seu estado nutricional, mas esses fatores devem ser avaliados levando em conta outras variáveis como idade, sexo e estatura.

Segundo Petroski (2007), a massa corporal pode ser encontrada utilizando uma balança (Figura 2.1) com precisão de 100 gramas e verificada pelo INMETRO regularmente, onde o avaliado deve subir, de preferência com roupas leves e sem calçados. Normalmente esta avaliação é feita apenas uma vez, tendo seu valor registrado.

Figura 2.1: Balança



Fonte: Imagem de jannoon028 (n.d.) no Freepik

2.2.2 Estatura

Petroski (2007) explica que é possível obter diversas medidas de altura de um indivíduo, pois cada parte do corpo pode ser medida em relação à distância do solo utilizando diversas ferramentas, mas uma das mais importantes para descobrir o desenvolvimento corporal e estado nutricional é a estatura de um indivíduo que pode ser aferida utilizando um estadiômetro ou uma fita métrica metálica adaptada com hastes, ambos com variação de 1mm. O avaliado deve estar de pé e descalço, com os pés unidos. Para maior precisão, devem ser feitas três medições, calculando sua média para utilização nos cálculos. A cada medição a pessoa deve sair e retornar ao local da avaliação.

2.2.3 IMC - Índice de massa corporal

Ribeiro, Melo e Tirapegui (2018) explica que é possível utilizar os valores da massa corporal e estatura para construir índices que podem ser usados para analisar a composição corporal de uma pessoa. Os cálculos para construir estes índices envolvem dividir o peso corporal por alguma potência da estatura, sendo o índice de massa corporal (IMC) o mais utilizado, sua fórmula é:

$$\text{IMC}\left(\frac{kg}{m^2}\right) = \frac{\text{Peso corporal}(kg)}{\text{Estatura}(m^2)}$$

Oliveira, Tavares e Bosco (2015 apud ANDRADE et al., 2018) classifica os resultados obtidos no cálculo do IMC através da tabela 2.1:

Tabela 2.1: Índice de massa corporal

Classificação	IMC (kg/m ²) Pontos principais
Magreza grave	< 16,00
Magreza moderada	16,00 - 16,99
Magreza leve	17,00 - 18,49
Normal	18,50 - 24,99
Sobrepeso	25,00 - 29,99
Obesidade grau I	30,00 - 34,99
Obesidade grau II	35,00 - 39,99
Obesidade grau III	> 40

Fonte: Oliveira, Tavares e Bosco (2015 apud ANDRADE et al., 2018)

Ribeiro, Melo e Tirapegui (2018) explica que a adoção pelo cálculo do IMC se dá pela fácil obtenção das medidas envolvidas no cálculo, peso corporal e estatura, e pela direta proporcionalidade dos riscos para a saúde com o aumento dos valores obtidos pelo IMC.

Embora, no âmbito epidemiológico, os valores do IMC sejam utilizados como importantes indicadores da composição corporal, sua interpretação, no contexto individual, deve ser feita com cautela. (RIBEIRO; MELO; TIRAPEGUI, 2018)

Andrade et al. (2018) reforça a importância da compreensão e avaliação dos resultados obtidos pelo IMC e suas limitações. Este índice é adequado apenas para pessoas sedentárias,

tendo em vista que ele se baseia no peso total, somando o peso dos músculos, gordura, água, entre outros, mas não leva em conta variáveis como quantidade de exercícios praticados, o que altera a quantidade de massa magra e gordura no corpo alterando sua proporção. Desse modo é comum que duas pessoas com portes atléticos totalmente diferentes, mas com mesma altura e peso, obtenham o mesmo IMC.

2.2.4 Perímetro

Andrade et al. (2018) é possível utilizar a razão cintura-quadril para ajudar na interpretação dos dados do IMC. A figura 2.2 mostra como utilizar a fita métrica para obter os valores em centímetros dos perímetros da cintura e do quadril, em seguida é calculado seu índice com a seguinte fórmula:

$$\text{Índice cintura/quadril} = \frac{\text{Perímetro da cintura}(cm)}{\text{Perímetro do quadril}(cm)}$$

Figura 2.2: Medidas dos perímetros de cintura(A) e quadris(B)



Fonte: (RIBEIRO; MELO; TIRAPEGUI, 2018)

Costa (2001 apud ANDRADE et al., 2018) afirma que além destes cálculos, pode-se averiguar o perímetro da região abdominal que indica risco de doenças crônico-degenerativas nos valores maiores que 102 cm para homens e 88 cm para mulheres.

Oliveira, Tavares e Bosco (2015 apud ANDRADE et al., 2018) classifica os resultados obtidos no cálculo do índice cintura/quadril através da tabela 2.2:

Tabela 2.2: Tabela referência Índice Cintura/Quadril

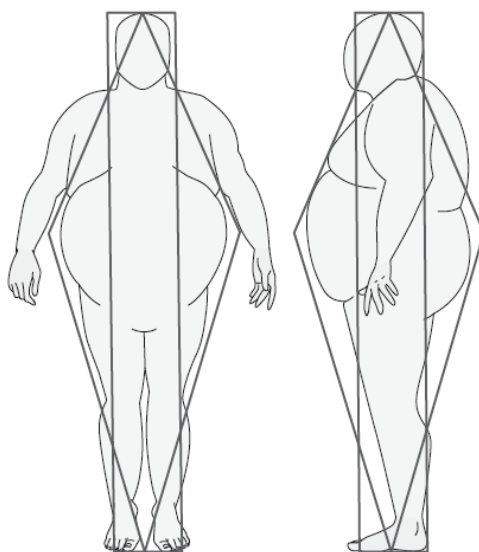
Gênero	Idade(anos)	Baixa	Moderada	Alta	Muito alta
Homens	20 - 29	< 0,82	0,83 a 0,88	0,89 a 0,94	> 0,95
	30 - 39	≤ 0,83	0,84 a 0,91	0,92 a 0,95	≥ 0,97
	40 - 49	≤ 0,87	0,88 a 0,95	0,96 a 1,00	≥ 1,01
	50 - 59	≤ 0,89	0,90 a 0,96	0,97 a 1,02	≥ 1,03
	60 - 69	≤ 0,90	0,91 a 0,98	0,99 a 1,03	≥ 1,04
Mulheres	20 - 29	< 0,70	0,71 a 0,77	0,78 a 0,82	> 0,83
	30 - 39	≤ 0,71	0,72 a 0,78	0,79 a 0,84	≥ 0,85
	40 - 49	≤ 0,72	0,73 a 0,79	0,80 a 0,87	≥ 0,88
	50 - 59	≤ 0,73	0,74 a 0,81	0,82 a 0,88	≥ 0,89
	60 - 69	≤ 0,75	0,76 a 0,83	0,84 a 0,90	≥ 0,91

Fonte: (OLIVEIRA; TAVARES; BOSCO, 2015 apud ANDRADE et al., 2018)

2.2.5 Índice de conicidade

Segundo Ribeiro, Melo e Tirapegui (2018), o índice de conicidade é mais uma opção para realizar os cálculos antropométricos e fornece informações sobre a distribuição da gordura corporal e não apenas o valor total aproximado. Conforme mostrado na figura 2.3, esse índice se baseia no pressuposto de que quando existe um acúmulo de gordura na região central do corpo, apresenta um formato parecido com um cone duplo, por outro lado, quando a pessoa apresenta menor gordura na mesma região, apresenta um formato que se assemelha a um cilindro.

Figura 2.3: Perfil morfológico associado a um duplo cone e a um cilindro direcionado à descrição da distribuição da gordura corporal.



Fonte: (RIBEIRO; MELO; TIRAPEGUI, 2018)

Segundo Ribeiro, Melo e Tirapegui (2018) a fórmula para medir o índice de conicidade é a seguinte:

$$\text{Índice de conicidade} = \frac{\text{Perímetro da cintura}(m)}{0,109 \sqrt{\frac{\text{Peso Corporal}(kg)}{\text{Estatura}(m)}}$$

Ribeiro, Melo e Tirapegui (2018) explica que este índice parece se associar a predisposição de riscos às doenças cardiovasculares e metabólicas, sendo indicativo que baixos riscos estão ligados a valores próximos a 1 (um formato cilíndrico), por outro lado valores próximos a 1,73 (um formato similar a um duplo cone) sugerem um elevado risco de aparecimento destas doenças.

2.2.6 Dobras cutâneas

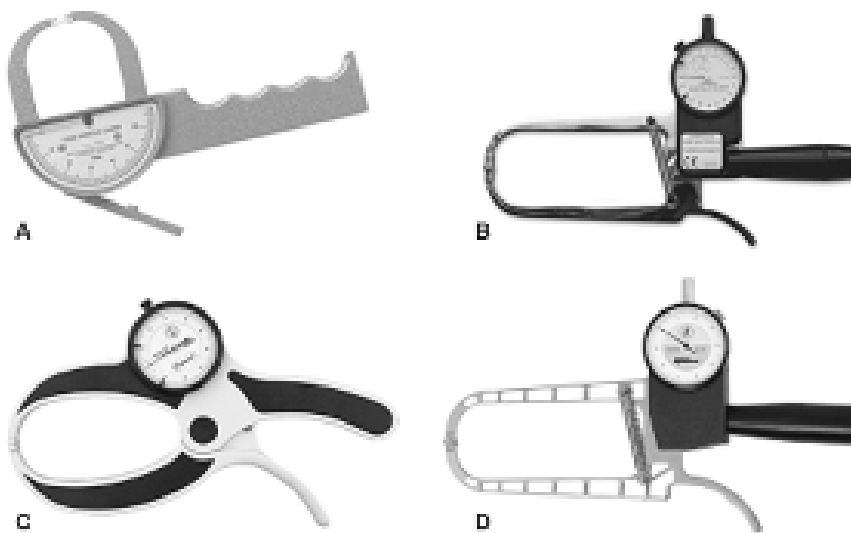
Ribeiro, Melo e Tirapegui (2018) afirma que grande parte da gordura corporal se encontra no tecido subcutâneo e é possível fazer uma análise da composição corporal a partir da mensuração das espessuras de regiões do corpo chamadas dobras cutâneas. Foram feitos estudos em cadáveres usando compassos para medir as espessuras das dobras cutâneas e em seguida foram feitas medidas diretas do tecido adiposo com uma incisão na área analisada. Essa análise

permitiu descobrir uma relação estatística entre essas medidas e que a disposição da gordura não é uniforme por todo o corpo.

Segundo Petroski (2007), as dobras cutâneas também são conhecidas como pregas cutâneas e podem ser utilizadas como uma forma indireta de estimar a composição corporal através de equações, pois existe uma relação linear entre os pontos anatômicos pinçados e a adiposidade corporal.

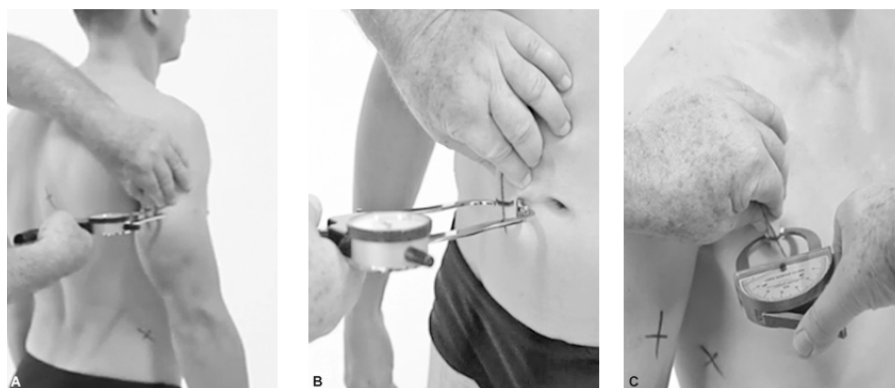
Andrade et al. (2018), chama a atenção para o fato de que existem diversas equações que levam em conta variáveis como gênero, idade e dobras analisadas, sendo importante usar a correta para encontrar um resultado correto. Sugere a utilização do adipômetro (Figura 2.4) para realizar as mensurações e que as seguintes áreas são as mais usadas nos cálculos: tríceps, subescapular, bíceps, axilar, torácica ou peitoral, supraíliaca, supraespinhal, coxa e panturrilha.

Figura 2.4: Figura Compassos específicos para medida de espessura de dobras cutâneas. A. Lange, B. Harpeden, C. Sanny e D. Cescorf.



Fonte: (RIBEIRO; MELO; TIRAPEGUI, 2018)

Figura 2.5: A a C. Medidas de espessura de dobras cutâneas.



Fonte: (RIBEIRO; MELO; TIRAPEGUI, 2018)

Jackson e Pollock (1985 apud ANDRADE et al., 2018) criaram um método para realizar um cálculo através das dobras cutâneas do tríceps, abdominal e suprailíaca (Figura 2.5). Nesse método, cada área deve ser medida duas vezes e as médias de cada área são então somadas e comparadas por uma tabela de referência. Foram criadas duas tabelas, uma para cada gênero. As tabelas 2.3 e 2.4 possuem várias linhas que indicam a soma das pregas e várias colunas que separam os valores em porcentagem de referência por idade.

Alvarenga (2007 apud ANDRADE et al., 2018) salienta que os valores de referência da gordura corporal adequados devem ser entre 8 e 20% para homens e de 13 a 28% entre mulheres.

Tabela 2.3: Tabela Mulheres - Referência porcentagem gordura corporal.

Soma das Pregas cutâneas (mm)	< 22	23-27	28-32	33-37	38-42	43-47	48-52	53-57	> 57
8-10	1,3	1,8	2,3	2,9	3,4	3,9	4,5	5,0	5,5
11-13	2,2	2,8	3,3	3,9	4,4	4,9	5,5	6,0	6,5
14-16	3,2	3,8	4,3	4,8	5,4	5,9	6,4	7,0	7,5
17-19	4,2	4,7	5,3	5,8	6,3	6,9	7,4	8,0	8,5
20-22	5,1	5,7	6,2	6,7	7,3	7,9	8,4	8,9	9,5
23-25	6,1	6,6	7,2	7,7	8,3	8,8	9,4	9,9	10,5
26-28	7,0	7,6	8,1	8,6	9,2	9,8	10,3	10,9	11,4
29-31	8,0	8,5	9,1	9,5	10,2	10,7	11,3	11,8	12,4
32-34	8,9	9,4	10,0	10,5	11,1	11,6	12,2	12,8	13,3
35-37	9,8	10,4	10,9	11,5	12,0	12,6	13,1	13,7	14,3
38-40	10,7	11,3	11,8	12,4	12,9	13,5	14,1	14,6	15,2
41-43	11,6	12,2	12,7	13,3	13,8	14,4	15,0	15,5	16,1
44-46	12,5	13,1	13,6	14,2	14,7	15,3	15,9	16,4	17,0
47-49	13,4	13,9	14,5	15,1	15,6	16,2	16,8	17,3	17,9
50-52	14,3	14,8	15,4	15,9	16,5	17,1	17,6	18,2	18,8
53-55	15,1	15,7	16,2	16,8	17,4	17,9	18,5	19,1	19,7
56-58	16,0	16,5	17,1	17,7	18,2	18,8	19,4	20,0	20,5
59-61	16,9	17,4	17,9	18,5	19,1	19,7	20,2	20,8	21,4
62-64	17,6	18,2	18,8	19,4	19,9	20,5	21,1	21,7	22,2
65-67	18,5	19,0	19,6	20,2	20,8	21,3	21,9	22,5	23,1
68-70	19,3	19,9	20,4	21,0	21,6	22,2	22,7	23,3	23,9
71-73	20,1	20,7	21,2	21,8	22,4	23,0	23,6	24,1	24,7
74-76	20,9	21,5	22,0	22,6	23,2	23,8	24,4	25,0	25,5
77-79	21,7	22,2	22,8	23,4	24,0	24,6	25,2	25,8	26,3
80-82	22,4	23,0	23,6	24,2	24,8	25,4	25,9	26,5	27,1
83-85	23,2	23,8	24,4	25,0	25,5	26,1	26,7	27,3	27,9
86-88	24,0	24,5	25,1	25,7	26,3	26,9	27,5	28,1	28,7

Continua na próxima página

89-91	24,7	25,3	25,9	26,5	27,1	27,6	28,2	28,8	29,4
92-94	25,4	26,0	26,6	27,2	27,8	28,4	29,0	29,6	30,2
95-97	26,1	26,7	27,3	27,9	28,5	29,1	29,7	30,3	30,9
98-100	26,9	27,4	28,0	28,6	29,2	29,8	30,4	31,0	31,6
101-103	27,5	28,1	28,7	29,3	29,9	30,5	31,1	31,7	32,3
104-106	28,2	28,8	29,4	30,0	30,6	31,2	31,8	32,4	33,0
107-109	28,9	29,5	30,1	30,7	31,3	31,9	32,5	33,1	33,7
110-112	29,6	30,2	30,8	31,4	32,0	32,6	33,2	33,8	34,4
113-115	30,2	30,8	31,4	32,0	32,6	33,2	33,8	34,5	35,1
116-118	30,9	31,5	32,1	32,7	33,3	33,9	34,5	35,1	35,7
119-121	31,5	32,1	32,7	33,3	33,9	34,5	35,1	35,7	36,4
122-124	32,1	32,7	33,3	33,9	34,5	35,1	35,8	36,4	37,0
125-127	32,7	33,3	33,9	34,5	35,1	35,8	36,4	37,0	37,6

Fonte: (JACKSON; POLLOCK, 1985 apud ANDRADE et al., 2018)

Tabela 2.4: Tabela Homens - Referência porcentagem gordura corporal.

Soma das Pregas cutâneas (mm)	< 22	23-27	28-32	33-37	38-42	43-47	48-52	53-57	> 57
23-25	9,7	9,9	10,2	10,4	10,7	10,9	11,2	11,4	11,7
26-28	11,0	11,2	11,5	11,7	12,0	12,3	12,5	12,7	13,0
29-31	12,3	12,5	12,8	13,0	13,3	13,5	13,8	14,0	14,3
32-34	13,6	13,8	14,0	14,3	14,5	14,8	15,0	15,3	15,5
35-37	14,8	15,0	15,3	15,5	15,8	16,0	16,3	16,5	16,8
38-40	16,0	16,3	16,5	16,7	17,0	17,2	17,5	17,7	18,0
41-43	17,2	17,4	17,7	17,9	18,2	18,4	18,7	18,9	19,2
44-46	18,3	18,6	18,8	19,1	19,3	19,6	19,8	20,1	20,3
47-49	19,5	19,7	20,0	20,2	20,5	20,7	21,0	21,2	21,5
50-52	20,6	20,8	21,1	21,3	21,6	21,8	22,1	22,3	22,6
53-55	21,7	21,9	22,1	22,4	22,6	22,9	23,1	23,4	23,6
56-58	22,7	23,0	23,2	23,4	23,7	23,9	24,2	24,4	24,7
59-61	23,7	24,0	24,2	24,5	24,7	25,0	25,2	25,5	25,7
62-64	24,7	25,0	25,2	25,5	25,7	26,0	26,7	26,4	26,7
65-67	25,7	25,9	26,2	26,4	26,7	26,9	27,2	27,4	27,7
68-70	26,6	26,9	27,1	27,4	27,6	27,9	28,1	28,4	28,6
71-73	27,5	27,8	28,0	28,3	28,5	28,8	29,0	29,3	29,5
74-76	28,4	28,7	28,9	29,2	29,4	29,7	29,9	30,2	30,4

Continua na próxima página

77-79	29,3	29,5	29,8	30,0	30,3	30,5	30,8	31,0	31,3
80-82	30,1	30,4	30,6	30,9	31,1	31,4	31,6	31,9	32,1
83-85	30,9	31,2	31,4	31,7	31,9	32,2	32,4	32,7	32,9
86-88	31,7	32,0	32,2	32,5	32,7	32,9	33,2	33,4	33,7
89-91	32,5	32,7	33,0	33,2	33,5	33,7	33,9	34,2	34,4
92-94	33,2	33,4	33,7	33,9	34,2	34,4	34,7	34,9	35,2
95-97	33,9	34,1	34,4	34,6	34,9	35,1	35,4	35,6	35,9
98-100	34,6	34,8	35,1	35,3	35,5	35,8	35,0	36,3	36,5
101-103	35,3	35,4	35,7	35,9	36,2	36,4	36,7	36,9	37,2
104-106	35,8	36,1	36,3	36,6	36,8	37,1	37,3	37,5	37,8
107-109	36,4	36,7	36,9	37,1	37,4	37,6	37,9	38,1	38,4
110-112	37,0	37,2	37,5	37,7	38,0	38,2	38,5	38,7	38,9
113-115	37,5	37,8	38,0	38,2	38,5	38,7	38,0	39,2	39,5
116-118	38,0	38,3	38,5	38,8	39,0	39,3	39,5	39,7	40,0
119-121	38,5	38,7	39,0	39,2	39,5	39,7	40,0	40,2	40,5
122-124	39,0	39,2	39,4	39,7	39,9	40,2	40,4	40,7	40,9
125-127	39,4	39,6	39,9	40,1	40,4	40,6	40,9	41,1	41,4
128-130	39,8	40,0	40,3	40,5	40,8	41,0	41,3	41,5	41,8

Fonte: (JACKSON; POLLOCK, 1985 apud ANDRADE et al., 2018)

2.3 Programação de dispositivos móveis

Os dispositivos móveis fazem parte da vida cotidiana da maioria das pessoas, sobretudo os smartphones. Ele mudou e evoluiu muito desde de o primeiro aparelho lançado em 1993 pela IBM até os dias atuais. Já em meados de 2006, a maioria dos telefone celulares podia se conectar à internet, mas o verdadeiro sucesso dos telefones inteligentes veio em 2007 com o lançamento o iPhone da Apple. A grande inovação foi a adição de tela sensível ao toque, além as outras funções já existentes nos concorrentes. Não demorou muito para surgir um concorrente direto, em 2008 a T-Mobile lançou o *High-Tech Computer* HTC G1, o primeiro smartphone executando o Android, um sistema operacional lançado pela Google.(SIMAS et al., 2019)

Payne (2019) explica que existem diversas maneiras que uma pessoa pode ter acesso a um serviço digital, seja ele um site da Web ou uma aplicação para smartphones como Android ou iOS. Cada uma destas plataformas exige que seus programas sejam escritos em linguagens diferentes, entre elas JavaScript para Web, Swift para iOS e Kotlin para Android. Dificilmente um programador conhece com maestria todas elas, sendo então necessário vários desenvolvedores com diferentes conhecimentos para entregar o mesmo serviço nestas plataformas. Isso eleva o custo de produção e principalmente o custo de manutenção.

Programas que rodam no Android são chamados *applications* ou *apps* para encurtar.(GOOKIN, 2020) Os aplicativos nativos são os criados na linguagem padrão de cada sistema. A segmentação das linguagens torna incompatíveis aplicativos criados para a plataforma concorrente,

sendo este um grande problema, para desenvolver software para vários sistemas seria necessário uma equipe com desenvolvedores que conhecem as várias linguagens ou equipes diferentes para cada plataforma. (SIMAS et al., 2019)

Segundo Simas et al. (2019), uma possível solução para este problema são os aplicativos híbridos, que são uma fusão entre aplicações nativas e sistemas Web. Nestes são usados *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript ou linguagens alternativas que simulem este comportamento, para gerar uma aplicação Web que é compilada em executáveis para cada sistema e publicados nas respectivas lojas. A grande vantagem é a criação e manutenção de uma única base de código para todos os sistemas reduzindo em muito o trabalho e os custos. Em contrapartida, os aplicativos híbridos tem limitações e normalmente não tem acesso a todos os recursos nativos e são mais lentos que aplicativos nativos. Mesmo com as limitações as aplicações híbridas foram cada vez mais adotadas por conta de suas vantagens.

Com o passar do tempo, foram criados *frameworks* capazes de desenvolver aplicativos nativos usando as técnicas e ferramentas de desenvolvimento Web. Alguns exemplos são o *Ionic Framework* baseado no angular (Web) e o *React Native* baseado nas bibliotecas *React*. (SIMAS et al., 2019) Como outra alternativa pode-se citar o Flutter baseado na linguagem *Dart*. (PAYNE, 2019)

2.3.1 Linguagem de programação - Dart

Segundo (BARRO, 2023), Dart foi criada pela Google em 2011 e é uma linguagem multiparadigma, orientada a objetos e de código aberto, sendo uma de suas principais vantagens a alta velocidade quando comparada ao JavaScript por exemplo. Ela pode funcionar de duas formas diferentes, *just-in-time* (JIT) na qual um trecho do código é convertido quando é chamado ou pode ser compilada para código nativo num processo chamado *ahead-of-time* (AOT).

O objetivo inicial da Google ao criar o Dart era substituir o JavaScript nos navegadores como a linguagem padrão. A linguagem foi recebida com diversas críticas entre os programadores, sendo acusada de ser como uma tentativa da Google de fragmentar os padrões da internet. Outra crítica, esta ligada à fama da Google de criar e abandonar projetos e serviços de forma repentina. Tudo isso mudou com o sucesso do *framework* Flutter, também criado pela Google. (BARRO, 2023)

2.3.2 Framework Frontend - Flutter

Napoli (2020) explica que o Flutter é um *framework* portátil de interface de usuário (UI na sigla em inglês) capaz de criar aplicações modernas, reativas e com código nativo com suporte para iOS, Android, Web, Windows, MacOS e Linux. Utiliza uma linguagem moderna, orientada a objetos chamada Dart, que é utilizada para escrever a UI e a lógica de negócio.

Payne (2019) afirma que a missão da Google com o Flutter, lançado em 2018, é construir uma melhor forma de desenvolver para dispositivos móveis e que seu objetivo é ser capaz de criar aplicações nativas para diversos sistemas utilizando uma única base de código.

Segundo Payne (2019), existem outras soluções concorrentes do Flutter como o Xamarin lançado em 2011 pela Microsoft e o *React Native* lançado em 2015 pelo Facebook que também

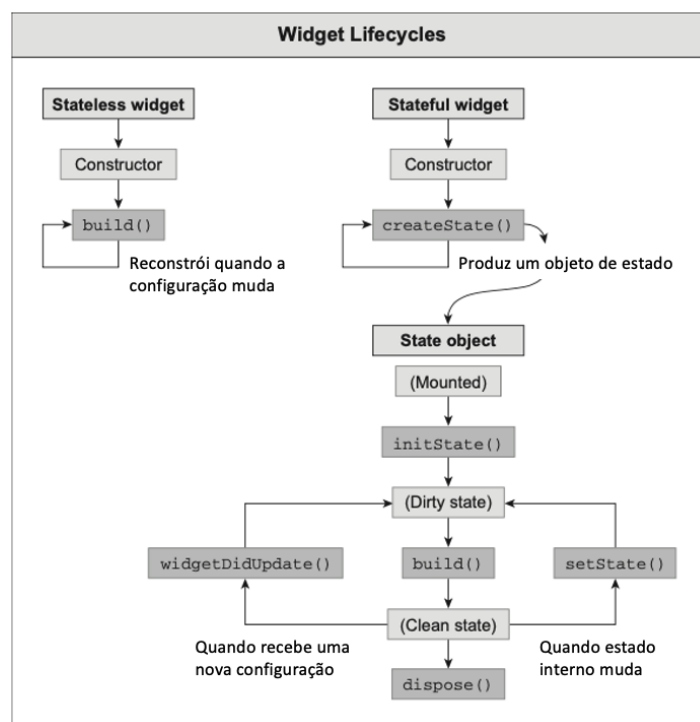
são capazes de criar aplicativos nativos Android e iOS, mas somente o Flutter é capaz de criar aplicativos Web com a mesma base de código. Além disso, sendo a solução mais recente, a Google pôde aproveitar o que deu certo e corrigir o que deu errado nas demais soluções anteriores, além de implementar inovações durante a criação de seu *framework*.

Para Napoli (2020), o Flutter é rápido, ótimo para rodar animações e fornece ferramentas para criar aplicações bonitas e profissionais. Outro recurso que ajuda muito no desenvolvimento é o *Hot Reload* que torna possível recarregar o estado do aplicativo para ver alterações feitas no código, sem que as informações atuais sejam perdidas.

No Flutter praticamente tudo é um *widgets*, um *widget* pode ser considerado um pedaço da interface do usuário, parte do aplicativo onde são feitas as interações, não apenas a parte visível da tela como por exemplo botões, textos e imagens, mas também existem *widgets* não visíveis que modificam a interação do demais como o posicionamento e espaçamento na tela. Os *widgets* são escritos em código Dart e podem conter outros *widgets*. São criados usando composição, isso significa que eles podem ter um filho ou mais se necessário. Estes *widgets* formam uma grande hierarquia chamada *widget tree*. (ZAMMETTI, 2019)

A grande maioria dos *widgets* pode ser classificada em duas categorias: *stateless* ou *stateful*. Os *StatelessWidgets* não guardam informações dentro de si, recebendo os dados que precisa de partes externas como seus *widgets* pais ou de outra classe externa e eles não decidem quando serão construídos ou destruídos. Em contraste os *StatefulWidgets* são associados a objetos *State* que podem armazenar as informações do *widget* e que possuem um método *setState* que indica ao Flutter quando o *widget* deve ser reconstruído. (WIDMIL, 2020) A figura 2.6 mostra as diferenças citadas entre estes dois tipos de *widget*.

Figura 2.6: Ciclo de vida de um *Widget*



Fonte: (WIDMIL, 2020), tradução própria

Payne (2019) explica que o Flutter SDK (software development kit) é a única ferramenta indispensável, pode ser instalada seguindo instruções no site¹ oficial do Flutter e afirma que embora seja possível escrever aplicativos Flutter sem um IDE, usando editores de texto e executando pelo terminal com os comandos do Flutter SDK, praticamente ninguém faria isso, pois existem vários IDEs que suportam o Flutter.

2.3.3 Escolha do IDE

Segundo a Amazon (2023), um ambiente de desenvolvimento integrado (IDE da sigla em inglês) é um software utilizado por programadores para criar outros programas e aplicações. Desenvolvedores utilizam os IDEs para facilitar seu trabalho.

Johnson (2019), explica que existem vários IDEs, mas a escolha sobre qual será usado depende de muitos fatores, dentre eles as funcionalidades oferecidas, atalhos de teclado, códigos prontos, cores e quaisquer outras funções que aumentem a produtividade. Uma vez familiarizado com um IDE, dificilmente um programador irá trocar por outro, pois levará muito tempo até aprender a usar a nova ferramenta e consequentemente ocorrerá perda de produtividade.

Sole (2019) afirma que o *Visual Studio Code* (VSCode) lançado pela Microsoft não é apenas outra destas ferramentas, mas sim um IDE muito poderoso criado especialmente para facilitar a criação de aplicações Web e móvel usando as principais linguagens de programação disponíveis, com recursos úteis já integrados como controle de versão *Git* e ferramentas de depuração. Além disso é totalmente gratuito e de código aberto.

Por estar disponível nos três sistemas operacionais mais usados (Windows, Mac e Linux), ser rápido e leve, o VSCode ganhou popularidade rapidamente, principalmente entre os desenvolvedores Web. Além destes pontos ele possui outras qualidades que o separam das outros IDEs, como suporte ao IntelliSense, ferramentas de refatoração e depuração e várias extensões criadas por outros usuários.(JOHNSON, 2019)

2.3.4 Banco de dados

Kroenke e Auer (2007) definem dados como registros de fatos e números, descrevem bancos de dados como estruturas usadas para armazenar tais dados e afirmam que grande parte das aplicações dependem de algum tipo de banco de dados. Estima-se que existam mais de 10 milhões de bancos de dados ativos no mundo atualmente.

Segundo Kroenke e Auer (2007), existem vários tipos de sistemas de bancos de dados sendo o mais comum e utilizado o sistema de banco de dados relacional. Neste tipo de sistema os dados são convertidos em registros e guardados em estruturas chamadas de tabelas. As tabelas são predefinidas, normalmente representando entidades e relacionamentos, de modo que os dados a ser inseridos devem respeitar estas definições para garantir atomicidade, consistência, isolamento e durabilidade dos dados (ACID). Para realizar operações neste tipo de banco de dados, como inserir ou recuperar os dados, é utilizada uma linguagem chamada SQL (*Structured Query Language*).

¹<https://flutter.dev/docs/get-started/install>

Giamas (2022) explica que a linguagem SQL existe desde 1970, mas começou a ganhar popularidade com o crescimento da internet, quando os sites deixaram de conter apenas conteúdos estáticos e passaram a exibir dados armazenados em bancos de dados. Por outro lado, o termo *Not only SQL* (NoSQL, não apenas SQL) surgiu em 1998, para representar um banco de dados que não seguia os padrões SQL, mas ainda era relacional.

Segundo Giamas (2022), somente em 2009 o termo NoSQL foi utilizado para referenciar bancos de dados não relacionais que estavam sendo criados pela Google e pela Amazon. Estes novos bancos utilizavam um sistema de armazenamento baseado em chave-valor focando em alta performance, escalabilidade, flexibilidade e menor complexidade, com uma menor preocupação nos conceitos de ACID.

Giamas (2022) apresenta o *MongoDB*, um banco de dados NoSQL orientado a documentos desenvolvido e publicado em 2009 pela empresa 10gen, atualmente chamada de Mongo Inc. Inicialmente o *MongoDB* não possuía recursos básicos de segurança e garantias de ACID, mas compensava em termos de performance e flexibilidade. Com o passar do tempo, novos recursos foram sendo desenvolvidos, até chegar na versão 6 que é a atual e consegue lidar com cargas de trabalho bem diferentes que vão de pequenos aplicativos até aplicações de grandes empresas com centenas de servidores.

Como relatado por Giamas (2022), existem diferentes formas para utilizar o *MongoDB*, como executar nos servidores locais ou remotos como instâncias EC2 da Amazon, ou ainda o *MongoDB Atlas* que é uma solução hospedada pela própria MongoDB Inc.

MongoDB Atlas é um serviço de banco de dados das mesmas pessoas que construíram o *MongoDB*. Ele simplifica a implantação e manutenção dos seus bancos de dados enquanto oferece a versatilidade que você precisa para construir aplicações globais resilientes e performáticas nos provedores da nuvem da sua escolha. (MONGO, 2023)

No site da MongoDB é possível escolher entre três tipos de planos de serviço no *MongoDB Atlas*, o *Serverless* (sem servidor), *Dedicated* (dedicado) e *Shared* (compartilhado). No *Serverless* o cliente é cobrado por quantidade de leituras e escritas, no *Dedicated* uma máquina fica exclusiva para o cliente e existem diversos subplanos com valores e quantidade de armazenamento e RAM e por último no *Shared* uma máquina é dividida entre vários clientes, sendo disponibilizada uma versão gratuita. (MONGO, 2023)

2.3.5 API Rest

Interface de Programação de Aplicações (API) é um termo usado para designar diversas coisas em situações diferentes, mas ficou conhecido como um serviço de internet onde os programas fazem chamadas HTTP com a intenção de ler ou escrever dados, como em um banco de dados remoto. Existem muitas APIs disponíveis e é possível um programador criar a própria. (PAYNE, 2019)

A Transferência de Estado Representacional (REST) é um estilo de software arquitetural para uma API que usa menor largura de banda para a transferência de dados. Uma API permite que duas aplicações conversem uma com a outra. (TYAGI, 2022)

Os métodos mais usados por desenvolvedores para as chamadas HTTP são: *GET* para ler

registros, *DELETE* para deletar registros, *POST* para inserir registros, *PUT* para substituir registros e *PATCH* para atualizar registros.

O padrão mais adotado para comunicações com API é o formato JSON. (PAYNE, 2019) A Notação de Objeto JavaScript (JSON) é um formato de dados intercambiáveis, pode ser utilizado em várias linguagens de programação e é baseado em texto. JSON usa o formato chave/valor para armazenar informações e é legível para humanos.

2.3.6 Padrão de Arquitetura MVC

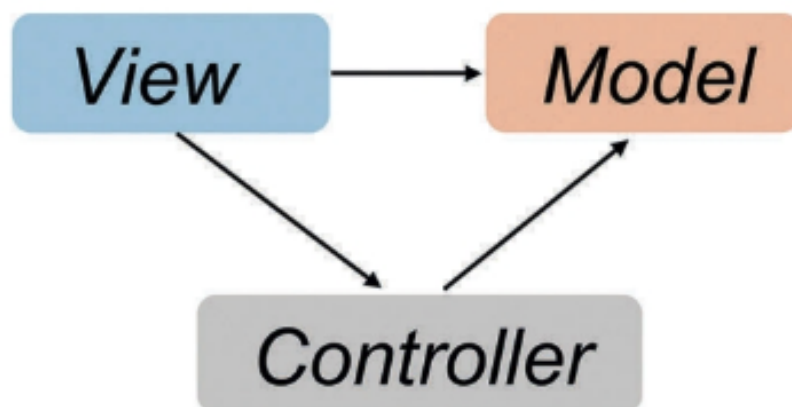
A arquitetura de sistemas é uma subárea da engenharia de software que se preocupa com os processos de desenvolvimento de um software, abrangendo desde a concepção até o desenvolvimento do produto. Na arquitetura de software são definidos os componentes do sistema e como eles vão interagir. Sua principal função é servir de visão geral do produto para a equipe de desenvolvedores. (ZENKER et al., 2019)

Pressman e Maxim (2021) define “modularidade” como uma separação de interesses, ou seja, classificar e separar trechos de um projeto complexo em partes mais simples de serem resolvidas. Aplicando isso ao projeto de software temos componentes chamados “módulos” que se comunicam para atender aos requisitos do sistema.

Segundo Zenker et al. (2019), existem diversos tipos de arquitetura de sistemas, sendo muito importante escolher a correta para o projeto, pois ela definirá como são detalhados os componentes, sua reutilização e manutenibilidade.

A arquitetura Modelo-Visão-Controlador (MVC) é um exemplo desses tipos de arquitetura. No “modelo” ficam o conteúdo e a lógica de processamento. A figura 2.7 mostra a interação entre as partes do MVC. A ‘visão’ cuida da interface de usuário, apresentando os dados contidos no modelo quando necessário. O “controlador” faz a comunicação entre o modelo e a visão. (PRESSMAN; MAXIM, 2021)

Figura 2.7: Padrão MVC



Fonte: (ZENKER et al., 2019)

O padrão arquitetural MVC tem como principais vantagens a criação de programas mais testáveis e manuteníveis quando comparado ao modelo monolítico. (ANDERSON, 2023)

Estudo da Arte

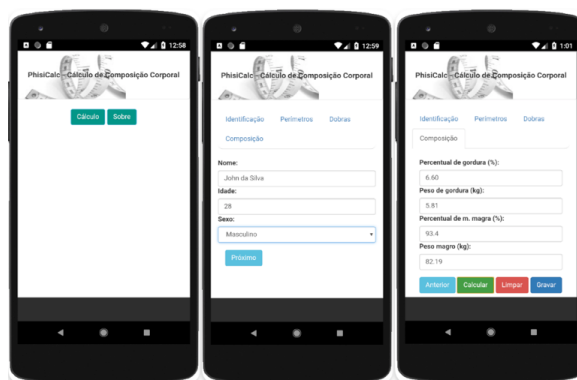
3.1 PhisiCalc

PhisiCalc é um aplicativo muito simples que permite realizar cálculos sobre a composição corporal, foi criado para ser uma ferramenta para academias e profissionais de educação física, com estes valores podem ser criados programas e exercícios individuais para os alunos. Possui mais de 10 mil downloads e nota de 4,4, mas apenas 90 avaliações. (GOOGLE-PLAY, 2023d)

O aplicativo possui poucas telas, iniciando com 2 opções Cálculo e Sobre. Ao acessar a página de cálculo o usuário pode informar os dados básicos do aluno, suas medidas de perímetro e dobras e na última tela consegue ver o resultado dos percentuais de gordura e massa magra, bem como seus respectivos valores em kg.

O aplicativo é fácil de usar e bem útil, mas poderia ter muito mais recursos, como um cadastro de alunos e gráficos para representar visualmente o resultado. A figura 3.1 mostra algumas telas do aplicativo.

Figura 3.1: Telas do aplicativo PhisiCalc



Fonte: (GOOGLE-PLAY, 2023d)

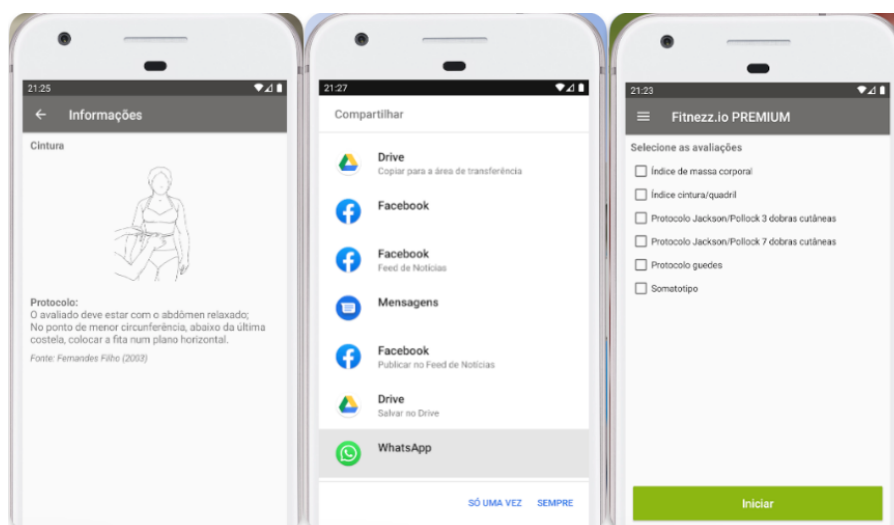
3.2 Fitnezz.io - Avaliação Física

O aplicativo Fitnezz.io realiza diversos cálculos muito utilizados pelos profissionais da área de educação física para cálculos da composição corporal, possui interface intuitiva e imagens que demonstram como devem ser feitas as aferições das medidas. Permite registrar, monitorar e compartilhar os resultados das avaliações. Com mais de 100 mil downloads e avaliação de 4,3 na Play Store.(GOOGLE-PLAY, 2023c)

A figura 3.2 mostra que é um aplicativo que visa a simplicidade, mas não deixa de fora algumas funções muito úteis. Ao iniciar é possível escolher quais avaliações serão feitas, como IMC, cálculo de dobras cutâneas de Pollock, entre outras. Em seguida são coletadas informações básicas como nome e medidas do aluno. Ao final são apresentados os resultados das contas solicitadas. Estas informações podem ser compartilhadas para os alunos.

O aplicativo possui grande potencial, com destaque para a exibição das imagens de como fazer as medições. Em contrapartida, possui algumas falhas severas, como exigir dados que poderiam ser opcionais em algum tipos de cálculos e o pior problema é a falta de atualização, tendo sido atualizado pela última vez em 2019, por conta disso é impossível instalar em aparelhos mais modernos, por conta de incompatibilidade.

Figura 3.2: Telas do aplicativo Fitnezz.io



Fonte: (GOOGLE-PLAY, 2023c)

3.3 Avaliação Física PRO

O Avaliação Física PRO é um aplicativo com foco nos profissionais de academia, permitindo que estes façam avaliações físicas dos alunos e prescrevam treinos. Os alunos podem se cadastrar, mas precisam ser adicionados a um professor, que pode acompanhar seu desenvolvimento. Até a presente data, o aplicativo já foi baixado mais de 100 mil vezes na Play Store e possui nota de 4,8 com mais de mil avaliações. (GOOGLE-PLAY, 2023a)

O aplicativo inicia numa tela onde o usuário pode ver seus dados, treinos e avaliações feitas, no caso do professor também pode ver quantos alunos possui cadastrados. Entre as

muitas funcionalidades, destaca-se a área de treinos onde podem ser disponibilizados vídeos de como devem ser feitos os exercícios e resultados da avaliação, onde mostra a situação atual da distribuição de gordura corporal e outros dados, além de diversos gráficos como mostrado na figura 3.3, para facilitar o entendimento e a evolução ao longo do tempo.

O aplicativo atende bem aos usuários, evidenciado pelas avaliações. A maior desvantagem é possuir um plano gratuito limitado a apenas um aluno, isso permite conhecer e experimentar suas funções, mas será necessário pagar uma mensalidade para ter acesso a todas as funções e cadastrar mais alunos.

Figura 3.3: Telas do aplicativo Avaliação Física PRO



Fonte: (GOOGLE-PLAY, 2023a)

Os pontos fortes e fracos dos aplicativos estudados nesse capítulo foram levados em consideração ao criar o aplicativo Fit Measure, como por exemplo:

- Foi utilizado um design simples como o PhisiCalc, mas com mais funções que este.
- Foi incluída a função de compartilhamento de resultados como visto no Fitnezz.io, mas garantindo compatibilidade com os aparelhos mais novos.
- Os resultados podem ser visualizados em gráficos como no Avaliação Física PRO, mas podem ser incluídos vários alunos sem custo.

Metodologia e Desenvolvimento

4.1 Levantamento de Requisitos

A primeira coisa a definir quando inicia-se um projeto de software é o que deve ser construído. Apesar de ser uma tarefa clara, sua execução pode ser complexa, levando um longo tempo e exigindo planejamento e experiência para conseguir bons resultados. (REINEHR, 2020)

Os requisitos de um software definem suas características e restrições, demonstrando como devem ser atendidas as necessidades do usuário do sistema. Em geral, estes requisitos não dependem das tecnologias envolvidas na construção do software. Em um levantamento de requisitos bem feito pode ser encontrado cedo algum erro no projeto que se fosse percebido em fases mais avançadas de desenvolvimento poderia custar entre 50 a 200 vezes mais. Desta maneira é possível dizer que a base para o sucesso de um projeto de software é uma análise bem feita dos requisitos, onde estes são bem identificados, definidos e classificados.(MACHADO, 2015)

Através das pesquisas realizadas para a criação deste software e conversas com profissionais de academia, foi possível identificar os seguintes requisitos:

- Cadastro do usuário: pode ser feito pelo próprio aplicativo informando nome, e-mail, uma senha forte e opcionalmente uma foto.
- Autenticação do usuário: para acessar o sistema é necessário digitar o e-mail e a senha;
- Cadastro dos alunos: o usuário deve informar os dados do aluno como nome, sexo, data de nascimento, e-mail e opcionalmente uma foto;

- Cálculo da composição corporal: devem ser informadas as medidas das dobras cutâneas para realização do cálculo usando o método das 7 dobras de Pollock, estes devem ser vinculados ao histórico do aluno;
- Visualização do relatório: ao escolher um aluno o sistema deve mostrar os valores da composição corporal realizada na medição atual, bem como gráficos comparando esta com a medição anterior, caso exista;
- Compartilhamento do relatório: o sistema deve ser capaz de exportar os resultados do relatório para envio ao aluno.

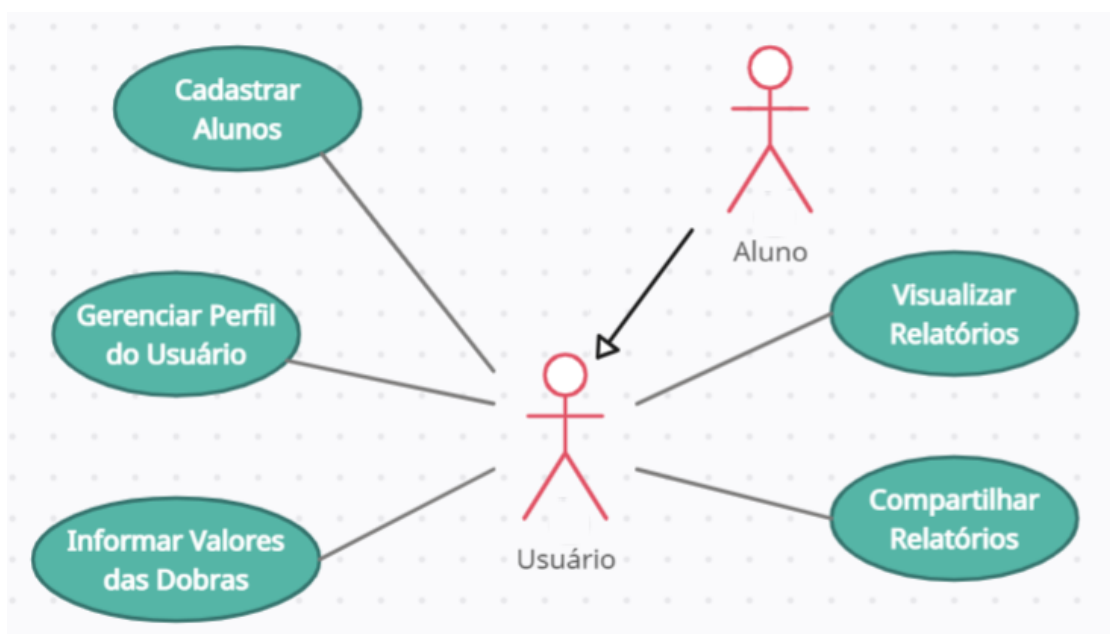
Pode-se definir como prioridades da aplicação as operações de cadastro e leitura dos dados, bem como a realização dos cálculos para que estes sejam confiáveis e garantam um resultado correto, tendo em vista que estes podem possivelmente ser usados para criação de treinos e diagnósticos caso seja necessário. O compartilhamento dos relatórios com o aluno é uma função interessante, mas tem prioridade mais baixa, pois este pode não compreender seu conteúdo sem uma explicação de um profissional da área.

4.2 Diagrama de caso de uso

O diagrama de casos de uso é um dos vários diagramas propostos pela UML e é utilizado para apresentar graficamente os requisitos funcionais do projeto, demonstrando as funções e serviços oferecidos a cada usuário. (REINEHR, 2020)

O diagrama de casos de uso do aplicativo Fit Measure esta representado na figura 4.1 e demonstra as principais funções disponíveis para o usuário, como cadastro de alunos, calculadora de composição corporal, visualização e exportação dos resultados.

Figura 4.1: Diagrama de casos de uso



Fonte: Autoria Própria

4.3 Desenvolvimento da aplicação

O desenvolvimento do aplicativo teve início após o levantamento e a análise dos requisitos e criação do caso de uso.

4.3.1 Criação do Banco de dados

Nos bancos de dados NoSQL orientados a documentos não existe uma estrutura fixa que defina como os dados serão armazenados e nem como vão se relacionar, como ocorre nos bancos de dados relacionais. Eles trabalham com conjuntos de documentos chamados de coleções. Um documento pode ser definido como um conjunto de campos que trabalham com uma estrutura chave-valor, ele também possui um identificador único. Os campos de um documento podem armazenar diversos tipos de valores, como *strings*, números ou até mesmo outros documentos. Por conta de sua natureza não estruturada, é possível realizar alterações em um documento incluindo ou excluindo campos sem causar problemas ao banco de dados. (LÓSCIO; OLIVEIRA; PONTES, 2016)

Na figura 4.2 estão ilustrados exemplos que representam alguns documentos das coleções usadas no aplicativo.

Figura 4.2: Exemplo de documentos do banco de dados



Fonte: Autoria Própria

É possível observar que todos os documentos possuem o campo “_id” que é gerado automaticamente pelo *MongoDB Atlas* e serve como identificador único no banco de dados. Nos documentos da coleção Usuário são registrados os dados pessoais e os campos “email” e “senha” que serão utilizados para acesso ao sistema. Caso não exista um usuário já cadastrado com o e-mail informado, este já ficará disponível para acessar o sistema pela tela de *Login*.

A coleção Aluno armazena os dados pessoais de cada um dos alunos além de fazer o vínculo com o usuário que o cadastrou utilizando o campo “idUsuario”. O campo “urlFoto” dos documentos de usuário e aluno é opcional e é usado para mostrar uma foto no aplicativo. Caso não seja encontrado, uma imagem padrão é usada.

Na coleção Medida são registradas as medições da altura, peso e das dobras cutâneas realizadas numa determinada data e são vinculadas ao aluno pelo campo “idAluno”. O aplicativo utiliza estes dados para realizar o cálculo da composição corporal, apresentar os resultados e montar os gráficos.

4.3.2 Interface

A interface do aplicativo foi desenvolvida utilizando um tema padronizado em todas suas telas. Ao utilizar o Flutter essa tarefa é simplificada, pois ele permite criar um arquivo que define as cores, fontes e formatos de diversos *widgets*. Essa configuração foi feita no arquivo “tema.dart” e é importada e utilizada no arquivo “main_app.dart”.

O mesmo ocorre com a navegação entre telas, para tanto foi criado um arquivo chamado “rotas.dart” onde define-se uma variável do tipo mapa com o formato de chave-valor, sendo a chave uma *string* que define o nome de determinada rota e o valor a classe que define o que será exibido na tela.

Além disto, foram criados alguns *widgets* customizados que foram usados em mais de um arquivo, como por exemplo o *CircleImage*, que é utilizado para mostrar a foto do aluno ou do usuário. Isto foi feito para facilitar a manutenção, ao alterar o arquivo do *widget* a alteração acontece em todas as telas, mantendo o padrão.

4.3.3 Conexão entre banco de dados e aplicação

Segundo Tiwari (2011), o termo NoSQL vem da união da palavra No, não em inglês, e da sigla SQL, representando tecnologias de banco de dados que não utilizam apenas SQL. NoSQL é uma nova e crescente tecnologia que está sendo endossada por grandes empresas de tecnologia como a Google e a Amazon. Esta sendo adotada por muitos desenvolvedores e já existem muitos gerenciadores NoSQL, como por exemplo MongoDB e Firebase.

O *MongoDB Atlas* foi o banco de dados escolhido para o Fit Measure, pois se trata de uma das tecnologias mais novas da área e possui uma versão gratuita que será ideal para atender a demanda inicial do aplicativo. Uma das primeiras etapas do projeto foi criar o banco de dados, para isso foi necessário acessar o site da MongoDB, criar uma conta de usuário e criar o banco. Após fazer isso, foi gerada uma *string* de conexão, a qual usamos para conectar através do servidor de API Rest criado em Dart. Para interagir com o banco de dados é utilizado um *plugin* chamado *mongo_dart*.

Mongo_dart é uma livreria desenvolvida em Dart puro, criada para ser utilizada no lado do servidor. Ela permite acessar várias versões do *MongoDB*, para tanto ela possui vários diferentes métodos que podem ser usados para interagir com versões mais antigas. A versão 6.0 e 7.0 também são acessíveis, mas por conta das diversas mudanças, algumas funções ainda não foram completamente implementadas. As operações de CRUD e outras mais avançadas podem ser feitas com este *plugin*. (PUB-DEV, 2023)

Para começar a fazer consultas, inserções ou outras operações no banco, primeiro é criada uma variável chamada “db” que será utilizada como ponto de acesso, então cria-se um método “init” que é responsável por passar os parâmetros, em seguida é solicitada a abertura da conexão. Este método é assíncrono, pois a conexão pode levar alguns segundos e não deve impedir os demais processos executados no servidor. Este método retorna verdadeiro em caso de sucesso ou falso em caso de algum erro ou falha, como demonstrado no código da figura 4.3.

Figura 4.3: Conexão com o banco de dados através do mongo_dart

```
> core > data > mongo_datasource.dart > ...
class MongoDatasource {
  late final Db db;

  Future<bool> init() async {
    try {
      const user = 'fit_root';
      const password = 'krujNBPTF4k6esdU';
      const host = 'cluster0.dkyodtd.mongodb.net';
      const database = 'fitmeasure';

      db = await Db.create('mongodb+srv://$user:$password@$host/$database');
      await db.open();
      return true;
    } catch (e) {
      return false;
    }
  }
}
```

Fonte: Autoria Própria

A figura 4.4 mostra o código executado ao iniciar o programa no servidor. Após instanciar o objeto “datasource” é feita a chamada de seu método “init”, onde caso retorne falso podem ser feitas até três novas tentativas. Em caso de sucesso a conexão já pode ser usadas para atender as chamadas a API Rest, em caso de falha total, as chamadas da API vão retornar o erro da conexão com o banco.

Figura 4.4: Método main - tentativa de conexão

```
main.dart > ...
void main() async {
  MongoDatasource datasource = MongoDatasource();

  for (var i = 0; i < 4; i++) {
    final conexaoBemSucedida = await datasource.init();
    if (conexaoBemSucedida) break;
  }

  runApp(const FitMeasureApp());
}
```

Fonte: Autoria Própria

Todas as operações do CRUD (criação de documentos, recuperação, alteração e exclusão) são feitas através de requisições HTTP que o aplicativo *front-end* faz para a API do servidor remoto Dart. O *back-end* por sua vez utiliza o *plugin* *mongo_dart* para realizar as operações no banco de dados *MongoDB Atlas*, onde os dados são persistidos. Estas operações podem retornar três resultados:

- 200: Operação realizada com sucesso.
- 404: Recurso não encontrado.
- 500: Falha no servidor.

Em caso de sucesso, o servidor retorna a informação utilizando uma mensagem com texto estruturado no formato JSON, que o aplicativo transforma em um objeto como Usuário, Medida ou Aluno, para ser utilizado internamente.

4.3.4 Autenticação do usuário

Ao abrir o aplicativo no smartphone Android, o usuário deve informar seu e-mail e sua senha para validar seu acesso, estes dados são verificados no banco de dados *MongoDB Atlas*.

Caso seja um novo usuário que ainda não possua conta, ele deve clicar no botão “Novo Usuário” e deve cadastrar seu nome, um e-mail válido e uma senha.

Após a autenticação, o aplicativo armazena o usuário atual, não sendo necessário fazer um novo login ao abrir e fechar o aplicativo. Caso não queira que seus dados fiquem salvos no aparelho, o usuário deve clicar no botão “Deslogar”, dessa forma ao abrir o aplicativo novamente será solicitado os dados de acesso.

4.3.5 Recuperação de senha

Se o usuário não lembrar sua senha, pode clicar no botão “Esqueci a senha” e informar seu e-mail. Se este estiver cadastrado, um código de validação será gerado pelo servidor (Figura 4.5) e em seguida será enviado para o e-mail informado, em uma nova tela o usuário deve digitar o código de validação e uma nova senha.

Figura 4.5: Método para resetar senha do usuário

```
Future<Response> resetPassword(Request request) async {
  try {
    final body = await request.readAsString();
    final json = jsonDecode(body) as Map<String, dynamic>;
    final email = json['email'];
    final result = await _datasource.findByEmail(email);

    if (result == null) {
      | return Response.notFound('Usuário não encontrado');
    }

    final resetCode = Random().nextInt(999999).toString().padLeft(6, '0');
    resetCodes.add((email, resetCode));
    sendEmail(email, resetCode);

    return Response.ok(jsonEncode(result));
  } catch (e) {
    | return Response.notFound('Usuário ou senha inválidos');
  }
}
```

Fonte: Autoria Própria

4.3.6 Perfil de usuário e aluno

Após autenticar seu acesso, o usuário pode alterar seu perfil e colocar uma foto ou trocar seu nome e senha. Não é permitido alterar o e-mail, pois este é seu identificador único. Os dados são salvos na coleção “Usuarios”.

Para cadastrar um perfil do aluno, o usuário deve acessar a tela de alunos e informar os dados básicos como nome, e-mail, sexo e data de nascimento, opcionalmente pode cadastrar uma foto. As informações são gravadas na coleção “Alunos”.

4.3.7 Calculadora de composição corporal

Para realizar os cálculos de IMC e dobras cutâneas o usuário deve abrir a tela de Calculadora de Medidas. Nesta tela deve-se escolher um aluno, para isso deve clicar na “lupa”, que mostrará os alunos cadastrados por este usuário. Após a escolha os dados do aluno serão carregados e os demais campos de altura, peso e das sete dobras serão habilitados para digitação. Estes campos aceitam apenas valores numéricos com até duas casas decimais. Caso o aluno escolhido já tenha feito uma medição anterior, os últimos dados de altura e peso serão preenchidos, permitindo atualização para os valores atuais. Ao informar a altura e o peso, o IMC será exibido, e ao informar os valores das dobras, serão exibidos o percentual de gordura, o peso magro e o peso da gordura corporal. O sistema permitirá salvar a medição após o preenchimento de todos os campos. Estes valores salvos serão usados para a visualização dos relatórios.

4.3.8 Visualização de relatórios

Para visualizar os relatórios o usuário deve escolher um aluno através da “lupa” assim como é feito na calculadora de medidas. Ao selecionar o aluno o sistema mostrará gráficos com as duas últimas medições caso existam. Também é possível compartilhar este relatório com o aluno exibindo até as últimas dez medições, caso existam, para uma visão da evolução ao longo do tempo. Os valores dos gráficos são separados pelas datas de aferição e exibem uma linha que liga de uma medição até a próxima, mostrando o aumento ou a redução dos resultados. São gerados gráficos de linha de Peso Corporal, IMC, Massa Magra e Porcentagem da Gordura.

4.3.9 Compartilhamento de relatórios

Para compartilhar o relatório em PDF, o usuário deve clicar no botão compartilhar. Essa ação executa o método “_renderPDF” o qual gera cria uma versão dos gráficos em imagem, acrescenta dados do aluno e cria uma versão em PDF. Após este processo, o aplicativo faz uma chamada para o sistema Android compartilhar o arquivo gerado.

Figura 4.6: Método para criar e compartilhar os relatórios em PDF

```
Future<void> _renderPDF() async {
    final PdfDocument document = PdfDocument();
    double top = 60;

    //gera imagem gráficos
    final bitmapPDF = await _createPdfBitmap(_chartPDF);

    //configurações pdf
    document.pageSettings.size = PdfPageSize.a4;
    final PdfPage page = document.pages.add();

    //dados a serem exibidos
    final AlunoModel(:nome, :idade, :sexo) = alunoAtual.value;
    final dadosAluno = 'Nome: $nome\nIdade: $idade\nSexo: $sexo';
    final date = dateToStringBR(DateTime.now());

    //desenha o dados do aluno
    page.graphics.drawString("Relatório Gerado em: $date \n$dadosAluno",
        PdfStandardFont(PdfFontFamily.helvetica, 12),
        brush: PdfSolidBrush(PdfColor(0, 0, 0)),
        bounds: Rect.fromLTWH(0, 0, page.getClientSize().width, 20));

    //desenha os gráficos
    page.graphics.drawImage(bitmapPDF,
        Rect.fromLTWH(150, top, page.getClientSize().width - 250, 200));

    //grava PDF num local temporário
    final List<int> bytes = document.saveSync();
    final Directory directory = await getApplicationDocumentsDirectory();
    File file = File('${directory.path}/Output.pdf');
    await file.writeAsBytes(bytes, flush: true);

    //Exporta o arquivo
    await Share.shareXFiles([XFile(file.path)]);
}
```

Fonte: Autoria Própria

Funcionamento e Resultados

5.1 Apresentação do sistema

Após o desenvolvimento do sistema Fit Measure, foi feita uma série de testes a fim de verificar seu funcionamento e conformidade das funcionalidades e cálculos realizados. Foram feitos testes unitários automatizados que verificam as funções e seus retornos e em seguida testes manuais que verificam a interface do usuário e sua integração com as funcionalidades. Esta fase é realizada várias vezes para encontrar falhas e pontos de melhoria que devem ser corrigidos no sistema.

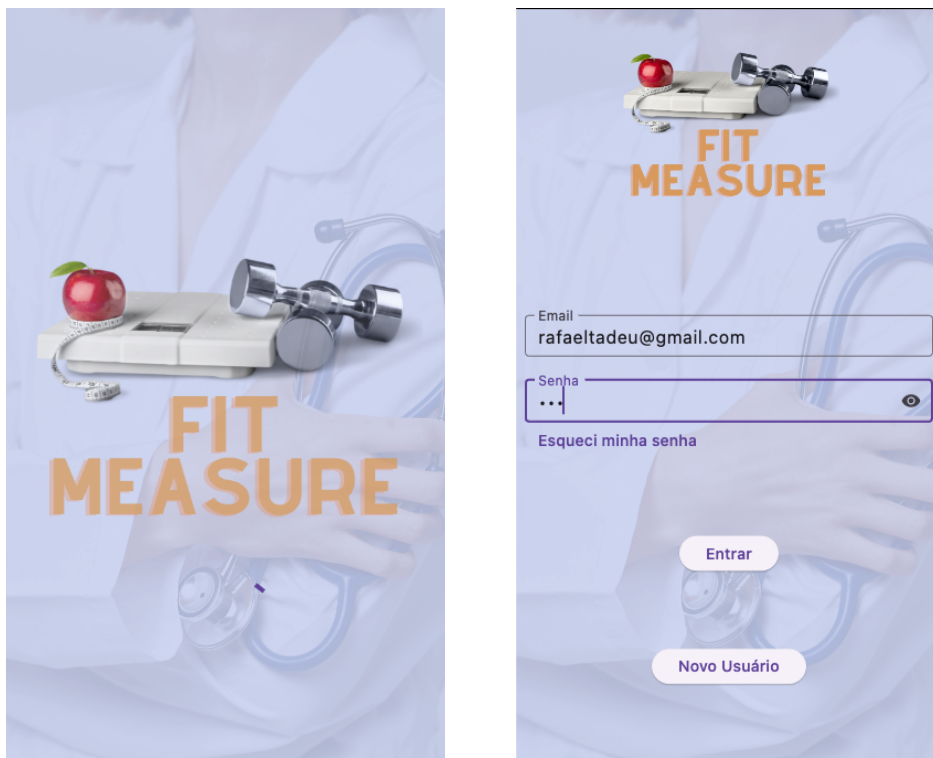
5.2 Telas de carregamento e login

A figura 5.1a mostra a primeira tela que o usuário tem contato ao abrir o sistema, é a tela de carregamento, nela é mostrada a logo do aplicativo. A imagem começa bem grande e reduz para o tamanho normal. Enquanto isso são feitas as conexões com o banco de dados e uma verificação do usuário. Após alguns segundos a tela de carregamento fecha e redireciona para a tela de *login* (Figura 5.1b) caso não seja feito o *login* automático. Nesta tela o usuário deve informar seu e-mail e senha e clicar em “Entrar”.

5.3 Tela de cadastro de usuário

Se for a primeira vez que o usuário acessa o aplicativo, ele deve clicar em “Novo Usuário” na tela de login. Ele será redirecionado para a tela de cadastro de usuário onde deve preencher seu nome, um e-mail válido, uma senha e a confirmação desta senha, conforme a figura 5.2. Caso as senhas informadas não sejam iguais ou o e-mail já tenha sido cadastrado, será exibida uma mensagem de erro, e os dados devem ser alterados pelo usuário.

Figura 5.1: Telas iniciais do Fit Measure



(a) Splash do Fit Measure

(b) Tela de Login

Fonte: Autoria Própria

Figura 5.2: Tela de Cadastro de Usuário

The image shows the 'Novo Usuário' (New User) registration screen. At the top, there is a back arrow and the title 'Novo Usuário'. Below the title are four input fields: 'Nome' (Name) with the value 'Jenifer Foster', 'Email' with the value 'jenfoster@email.com', 'Senha' (Password) with masked characters and a visibility toggle, and 'Confirme a Senha' (Confirm Password) with masked characters and a visibility toggle. At the bottom of the form is a 'Salvar' (Save) button.

Fonte: Autoria Própria

5.4 Telas de recuperação de senha

Para recuperar a senha em caso de esquecimento, o usuário deve clicar em “Esqueci minha senha” na tela de *Login*, será redirecionado para uma tela que deve informar seu e-mail (Figura 5.3a) e um código será enviado para o e-mail informado caso seja de um usuário válido. A figura 5.3b mostra que na próxima tela o usuário deve informar o código recebido e digitar uma nova senha, confirmar a senha novamente e clicar no botão “Salvar”.

Figura 5.3: Telas de Recuperação de Senha

The figure consists of two side-by-side screenshots of a mobile application interface for password recovery.
Screenshot (a) is titled 'Resetar Senha'. It features a back arrow on the top left and the title 'Resetar Senha'. Below the title is the instruction 'Informe seu email para receber o link de reset de senha'. There is an input field labeled 'Email' containing the text 'rafaeltadeu@email.com'. Below the input field is a rounded button labeled 'Enviar'.
Screenshot (b) is titled 'Nova Senha'. It features a back arrow on the top left and the title 'Nova Senha'. Below the title is an input field labeled 'Email' containing 'rafaeltadeu@email.com'. Below that is the instruction 'Informe o código recebido por email'. There is an input field labeled 'Código' containing '886345'. Below that is the instruction 'Informe sua nova senha'. There are two input fields for the password: 'Senha' and 'Confirmação de Senha'. Both fields contain masked characters (dots) and have a toggle icon on the right. Below the password fields is a rounded button labeled 'Resetar Senha'.

(a) Tela de Resetar Senha

(b) Tela de Nova Senha

Fonte: Autoria Própria

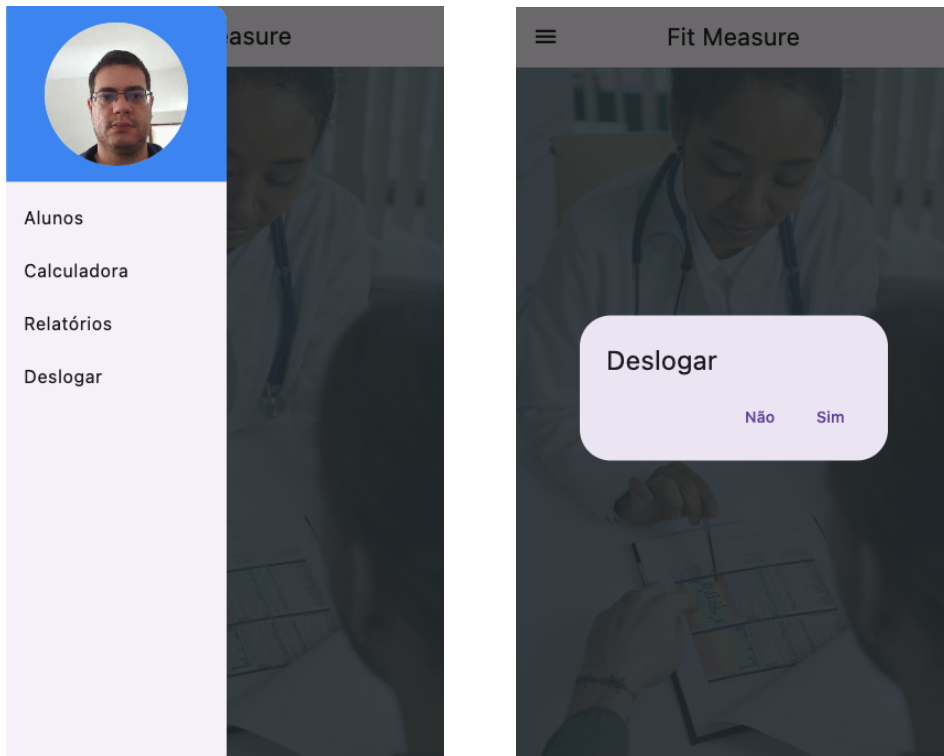
5.5 Tela menu

Após a autenticação, será exibida a tela de menu (Figura 5.4a), que é composta por um menu do tipo *Drawer* que se expande da lateral esquerda quando clicado. Neste menu aparecem a foto do usuário, o botão Alunos, o botão Calculadora, o botão Relatórios e por fim o botão Deslogar, que exibe um diálogo (Figura 5.4b).

5.6 Tela perfil do usuário

Ao clicar sobre a foto do usuário no menu, o sistema abre a tela de perfil do usuário (Figura 5.5), que permite trocar a foto e alterar o nome e a senha. Não é possível mudar o e-mail, por ser usado como identificador.

Figura 5.4: Telas de Menu e Diálogo Deslogar

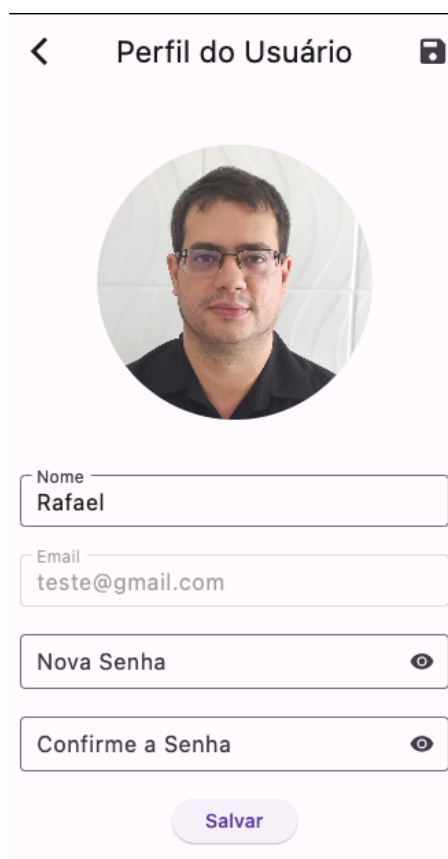


(a) Tela de Menu

(b) Diálogo de Deslogar

Fonte: Autoria Própria

Figura 5.5: Tela de Perfil do Usuário

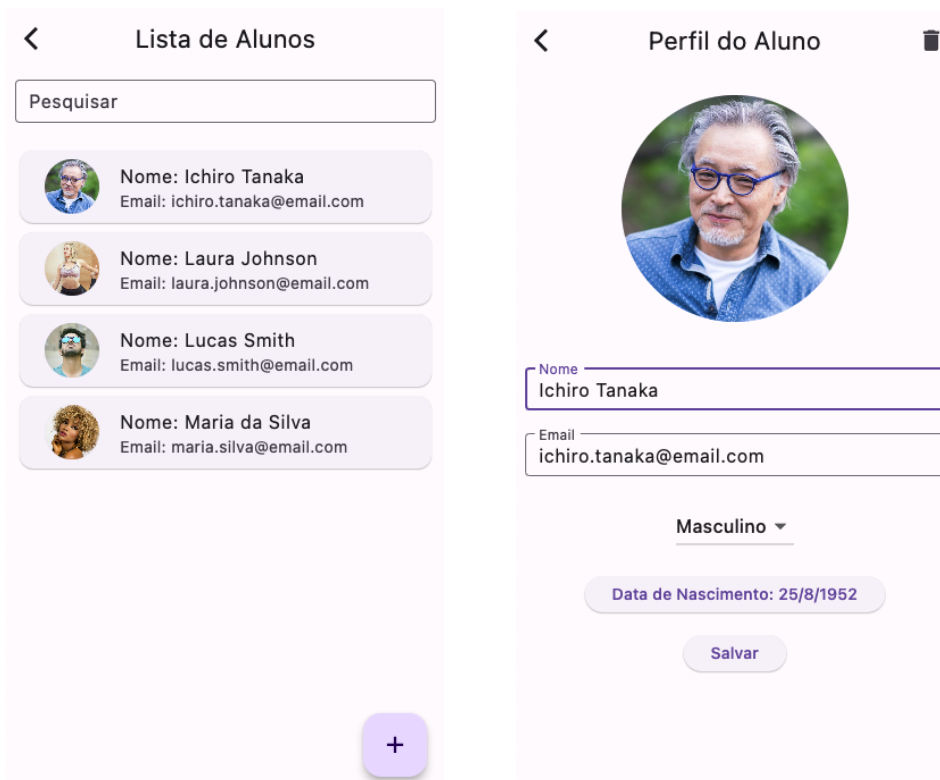


Fonte: Autoria Própria

5.7 Telas de lista e perfil de aluno

Clicando sobre o botão Alunos no menu, abre a tela de Lista de Alunos (Figura 5.6a), que mostra todos os alunos do usuário e permite pesquisar por nome. Ao clicar em um aluno, pode-se editar suas informações pessoais como nome, e-mail, sexo e data de nascimento, além da foto, como mostrado na figura 5.6b.

Figura 5.6: Telas de Menu Alunos



(a) Tela de Lista de Alunos

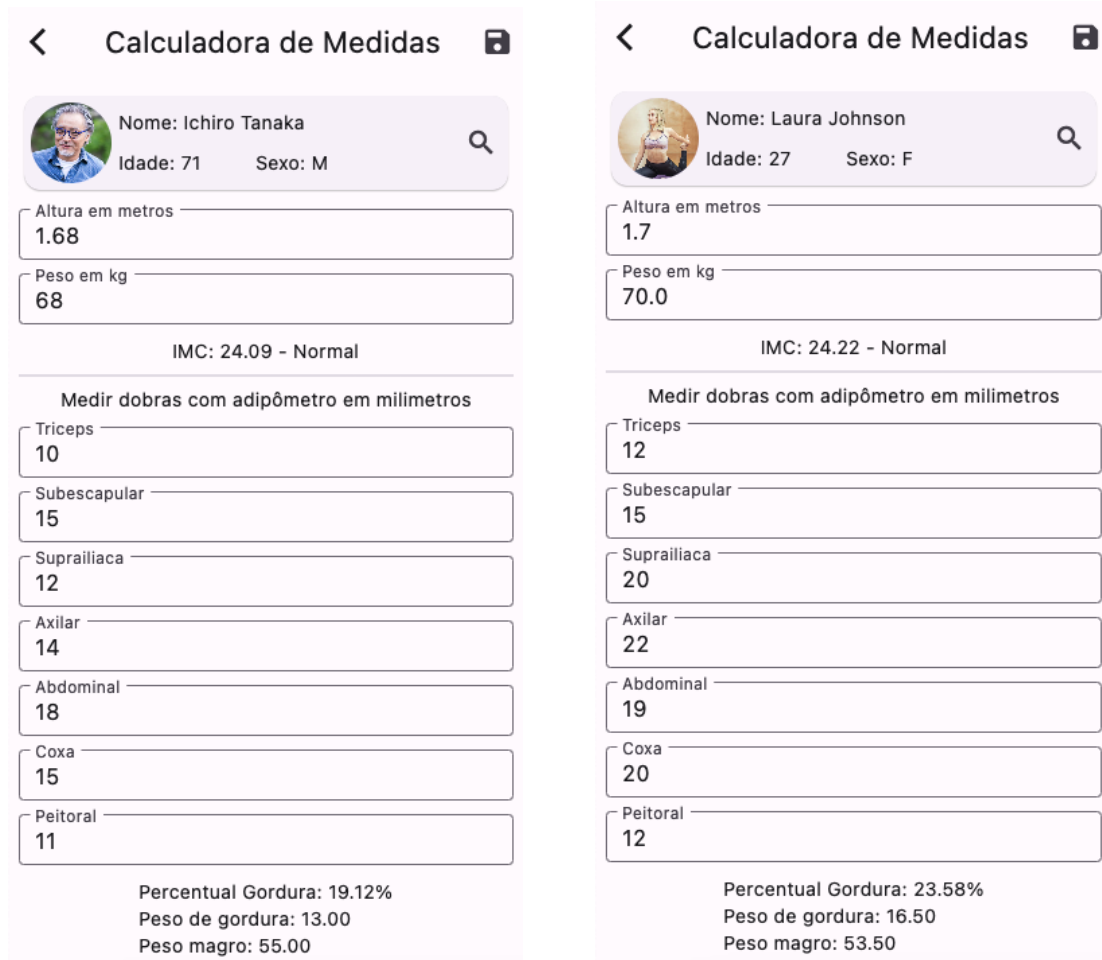
(b) Tela Perfil do Aluno

Fonte: Autoria Própria

5.8 Tela calculadora de medidas

Clicando sobre o botão Calculadora no menu, o sistema abre a tela de calculadora de medidas, como mostrado nas figuras 5.7a e 5.7b. Inicialmente os campos das medidas estão desabilitados e a única ação possível é buscar um aluno clicando na “lupa”. Isso exhibe a lista de alunos na qual deve ser selecionado um aluno, ao fazer isso os campos de medida são liberados. Ao informar os dados é exibido o IMC e a Massa Magra, a Gordura Corporal e Percentual de Gordura. Após informar os dados corretos é possível salvar os dados, para isso o usuário deve clicar no desenho de um disquete na parte superior direita.

Figura 5.7: Exemplo de Cálculos de Medidas



(a) Exemplo 1

(b) Exemplo 2

Fonte: Autoria Própria

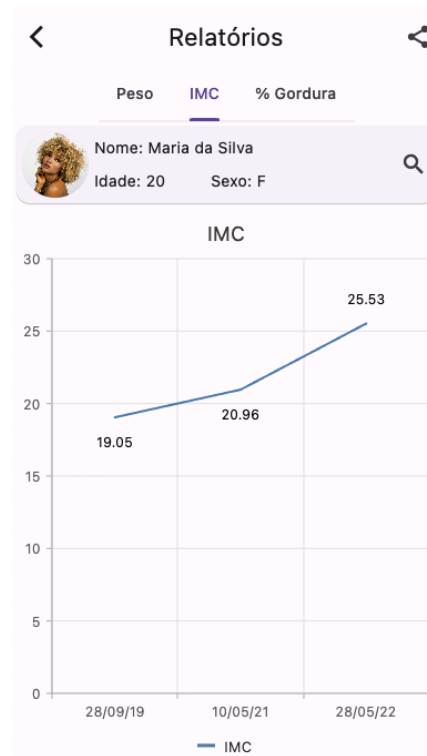
5.9 Tela de relatórios

Nesta tela de relatórios, deve ser escolhido um aluno clicando na “lupa”, da mesma forma que na tela de calculadora de medidas. Ao fazer isso, são exibidos gráficos que mostram o histórico das últimas medições feitas pelo aluno. Utilizando a barra acima dos dados do aluno, pode ser exibido o gráfico do Peso, IMC, Percentual de Gordura, como mostrado nas figuras 5.8a, 5.8b e 5.9.

Figura 5.8: Telas de Relatório



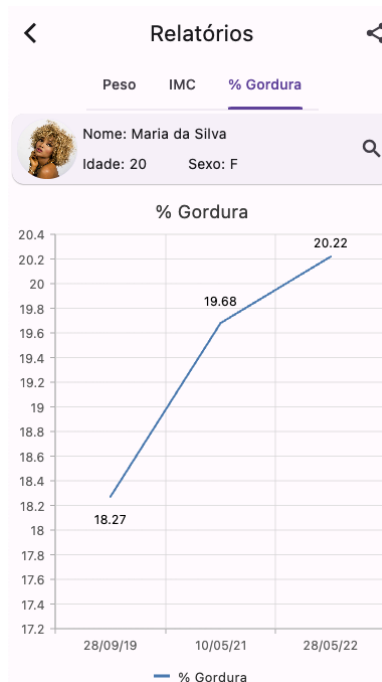
(a) Tela de Relatório Peso



(b) Tela de Relatório de IMC

Fonte: Autoria Própria

Figura 5.9: Tela de Relatório de Porcentagem de Gordura



Fonte: Autoria Própria

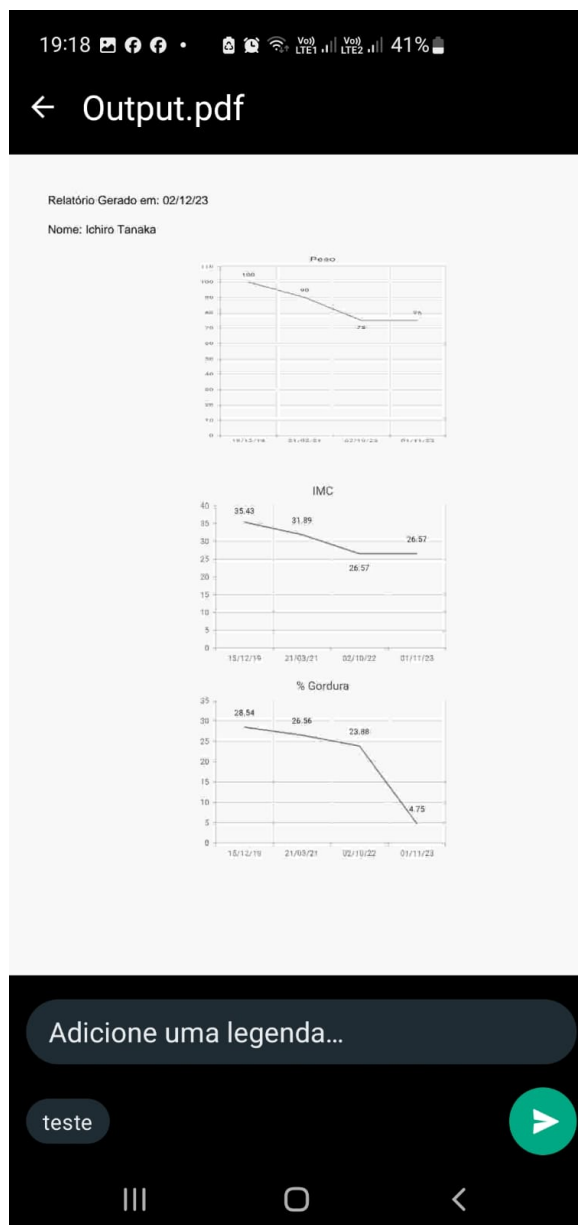
5.10 Compartilhamento de relatórios

Para compartilhar o relatório gerado na tela de relatórios, é necessário clicar no ícone de “Compartilhamento”. O processo de exportação (Figura 5.10a) será executado, criando um arquivo PDF que será enviado ao sistema Android com a intenção de compartilhar, isso vai permitir enviar o arquivo por e-mail ou WhatsApp. O arquivo PDF (Figura 5.10b) conterá o nome do aluno, a data de geração e os gráficos que podem mostrar até as últimas dez medições feita pelo usuário.

Figura 5.10: Exportando relatório



(a) Tela de Relatório Peso



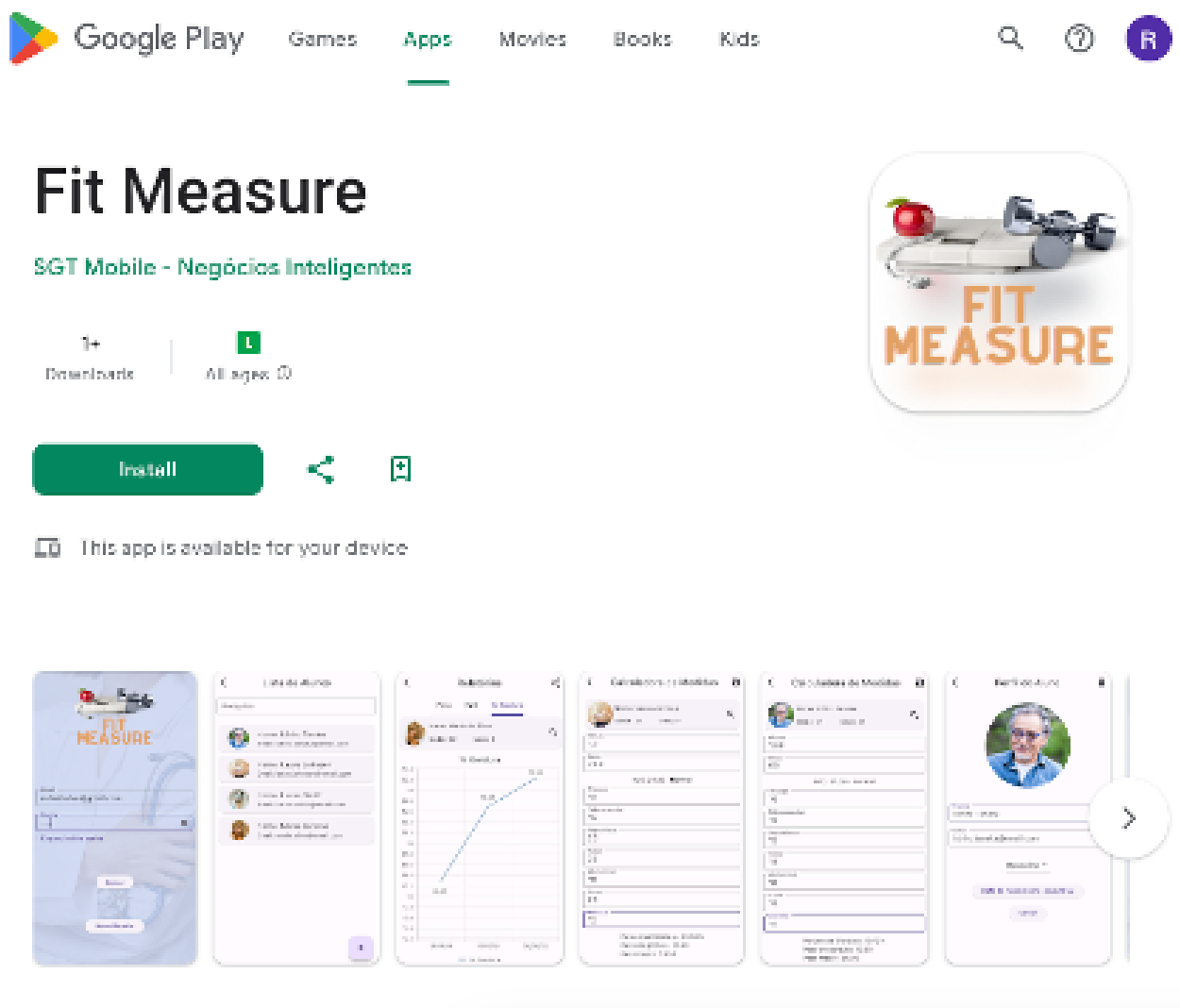
(b) Compartilhando o arquivo PDF

Fonte: Autoria Própria

5.11 Publicação na PlayStore

Assim que o aplicativo ficou pronto, ele foi enviado para a *PlayStore*, onde foi revisado pela equipe da Google e foi aprovado para publicação. Segundo ??), esta revisão analisa uma série de fatores como conteúdo do aplicativo e obediência as leis locais, violação de propriedade intelectual e privacidade do usuário e se o programa não contém *malwares*. Esta análise e publicação atestam pela qualidade do aplicativo Fit Measure, pois caso violasse alguma destas políticas seria barrado. A figura 5.11 mostra a página onde os usuários de smartphones Android podem obter o sistema.

Figura 5.11: Página download aplicativo Fit Measure na *PlayStore*



Fonte: (GOOGLE-PLAY, 2023b)

Conclusões

6.1 Considerações Finais

Ao término deste projeto, percebe-se que o sistema pode atender ao objetivo proposto de facilitar a realização dos cálculos antropométricos realizados por profissionais de academia e estética, além de permitir o compartilhamento dos resultados com seus alunos. O sistema é rápido e simples de usar, além de acessível, por estar disponível para a grande maioria de celulares que rodam o sistema Android que atualmente domina o mercado.

Para a criação do sistema foram utilizadas linguagens e *frameworks* atuais como é o caso do Dart e do Flutter, pois além de fornecerem estabilidade e facilidade da criação de telas bonitas e animadas, ainda permitem a publicação facilitada do sistema em outras plataformas, como Web, Windows e iPhone se houver mercado para estas.

O aplicativo atendeu as funcionalidades inicialmente previstas no projeto, mas ao longo de seu desenvolvimento notou-se que muitas funções ainda poderiam ser acrescentadas ou realizadas de forma mais simples. Um exemplo é a implantação da autenticação de usuários através do *Firebase Authentication*, que permite realizar o login usando contas de redes sociais, como Google, Facebook e Github, caso fosse utilizado poderia substituir as funcionalidades do capítulo 4.3.4.

Um grande desafio que ocorreu no desenvolvimento do sistema, foi a mudança da forma de acesso ao banco de dados *MongoDB Atlas*. Inicialmente o acesso era feito diretamente do aplicativo, mas isso se provou inviável por dois motivos insegurança e problema de conectividade. Inseguro, pois o aplicativo poderia ser decompilado e a senha e o endereço do banco de dados poderiam ser encontrados. O problema de conectividade acontecia quando ocorria troca ou falta de conexão com a internet isso também causava a quebra da conexão com o banco, sendo necessário reabrir o aplicativo. A solução para esses problemas foi a montagem do servidor remoto de API em Dart apresentado no capítulo 4.3.3.

O aplicativo foi recentemente lançado na *Play Store*¹, onde será utilizado por usuários que podem mandar suas sugestões de melhorias, dessa forma o projeto deve ser sempre atualizado para manter sua relevância. Por fim, o desenvolvedor deve refatorar o código de modo a facilitar a manutenção do sistema e seu futuro crescimento.

6.2 Trabalhos Futuros

Após implantar os requisitos levantados inicialmente e fazer testes com o sistema, novas ideias de funcionalidades, correções e pontos de melhoria surgiram para tornar o aplicativo mais completo e fácil de usar.

6.2.1 Acesso dos alunos

Disponibilizar uma área de acesso para os alunos, onde eles poderiam trocar dados pessoais e foto, além de ter acesso a suas medições.

Outra ideia é criar um espaço para comunicação do professor com os alunos, onde poderia passar informações de treino personalizado para cada aluno.

6.2.2 Login com o Firebase e redes sociais

O sistema permite login com e-mail e senha do usuário, mas existem ferramentas que tornam isto mais fácil, como o login com um clique que é possível ao integrar as ferramentas do *Firebase*. Este recurso é gratuito e amplamente usado entre os mais variados aplicativos.

6.2.3 Novas formas de cálculo de medidas corporais

Existem muitos tipos de cálculos que ainda podem ser incluídos no sistema como as 3 dobras de Pollock, índice de conicidade e índice de cintura/quadril. Também é possível fazer uma pré-avaliação sobre os resultados encontrados em cada cálculo, informando se está na média esperada ou fora dela.

Além disso, podem ser incluídas imagens ou vídeos de como realizar cada medição.

6.2.4 Criar versões para outras lojas de aplicativos

O Flutter possui a capacidade de gerar aplicativos com a mesma base de código para os principais Sistemas Operacionais *desktop* como Windows, MacOS, Linux, além do principal concorrente do Android, o iPhone. É possível criar versões destes sistemas e disponibilizar para mais usuários.

¹https://play.google.com/store/apps/details?id=br.com.sgtmobile.app_fit_measure

Referências Bibliográficas

ALVARENGA, L. L. *Classificação do estado nutricional e da composição corporal de praticantes de atividade física em academia. Revista Brasileira de Nutrição Esportiva*. [S.l.: s.n.], 2007.

AMAZON. *O que é IDE (Ambiente de desenvolvimento integrado)?* 2023. Disponível em: <<https://aws.amazon.com/pt/what-is/ide/>>. Acesso em: 12 Jan. 2023.

ANDERSON, R. *Parte 2, adicionar um controlador a um aplicativo ASP.NET Core MVC*. 2023. Disponível em: <<https://learn.microsoft.com/pt-br/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-7.0>>. Acesso em: 14 Oct. 2023.

ANDRADE, G. et al. *Métodos e técnicas de avaliação estética*. E-Book: Grupo A, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788595023192>>. Acesso em: 20 mai. 2023.

BARRO, B. B. *Conheça a Linguagem Dart e Entenda as Polêmicas que a Envolvem*. 2023. Disponível em: <<https://www.hostinger.com.br/tutoriais/linguagem-dart>>. Acesso em: 08 Oct. 2023.

COSTA, R. F. da. *Composição corporal: teoria e prática da avaliação*. [S.l.]: Manole, 2001.

DEITEL, P.; DEITEL, A. *Android - Como Programar*. Grupo A, 2015. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788582603482>>. Acesso em: 21 jun. 2023.

GIAMAS, A. *Mastering MongoDB 6.x: Expert's techniques to run high-volume and fault-tolerant database solutions using MongoDB 6.x*. [S.l.]: Packt, 2022.

GOOGLE-PLAY. *Avaliação Física PRO*. 2023. Disponível em: <<https://play.google.com/store/apps/details?id=com.codeapp.avaliacaofisica>>. Acesso em: 19 Set. 2023.

GOOGLE-PLAY. *Fit Measure*. 2023. Disponível em: <https://play.google.com/store/apps/details?id=br.com.sgtmobile.app_fit_measure>. Acesso em: 20 Nov. 2023.

GOOGLE-PLAY. *Fitnezz.io - Avaliação Física*. 2023. Disponível em: <<https://play.google.com/store/apps/details?id=br.com.ava>>. Acesso em: 18 Set. 2023.

GOOGLE-PLAY. *PhisiCalc*. 2023. Disponível em: <<https://play.google.com/store/apps/details?id=pconradjunior.phisicalc>>. Acesso em: 19 Set. 2023.

- GOOKIN, D. *Android For Dummies*. [S.l.]: John Wiley & Sons, Inc, 2020.
- JACKSON, A. S.; POLLOCK, M. L. *Practical assessment of body composition. The Physician and Sportsmedicine*. [S.l.: s.n.], 1985.
- JANNOON028. *Foto grátis feche acima da escada velha no assoalho de madeira*. n.d. Disponível em: <https://br.freepik.com/fotos-gratis/feche-acima-da-escada-velha-no-assoalho-de-madeira_1129142.htm>. Acesso em: 10 out. 2023.
- JOHNSON, B. *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. [S.l.]: John Wiley & Sons, 2019.
- KROENKE, D.; AUER, D. *Database Concepts*. [S.l.]: Prentice, 2007.
- LÓSCIO, B. F.; OLIVEIRA, H. R.; PONTES, J. C. S. *NoSql no desenvolvimento de aplicações Web colaborativas*. 2016. Disponível em: <https://www.researchgate.net/profile/Bernadette-Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas-links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf>. Acesso em: 23 Set. 2023.
- MACHADO, F. N. R. *Análise e gestão de requisitos de software: Onde nascem os sistemas*. [S.l.]: Érica, 2015.
- METABÓLICA, A. B. para o Estudo da Obesidade e da S. *Diretrizes Brasileiras de Obesidade 2016*. 2016. Disponível em: <<https://abeso.org.br/wp-content/uploads/2019/12/Diretrizes-Download-Diretrizes-Brasileiras-de-Obesidade-2016.pdf>>. Acesso em: 07 Oct. 2023.
- METABÓLICA, A. B. para o Estudo da Obesidade e da S. *Mapa da Obesidade*. 2023. Disponível em: <<https://abeso.org.br/obesidade-e-sindrome-metabolica/mapa-da-obesidade/>>. Acesso em: 07 Oct. 2023.
- MONGO. *MongoDB Pricing*. 2023. Disponível em: <<https://www.mongodb.com/pt-br/pricing>>. Acesso em: 20 Set. 2023.
- NAPOLI, M. L. *Beginning Flutter: A Hands On Guide To App Development*. [S.l.]: John Wiley & Sons, Inc, 2020.
- OLIVEIRA, A. M. de; TAVARES, A. M. V.; BOSCO, S. M. D. *Nutrição e Atividade Física: do Adulto Saudável às Doenças Crônicas*. [S.l.]: Atheneu, 2015.
- PAYNE, R. *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*. [S.l.]: Apress, 2019.
- PETROSKI, E. L. *Antropometria: Técnicas e padronizações*. [S.l.]: Nova Letra, 2007.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software*. Grupo A, 2021. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9786558040118>>. Acesso em: 14 Oct. 2023.
- PUB-DEV. *Mongo-dart - MongoDB driver for Dart programming language*. 2023. Disponível em: <https://pub.dev/packages/mongo_dart>. Acesso em: 25 Set. 2023.
- REINEHR, S. *Engenharia de requisitos*. Grupo A, 2020. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9786556900674>>. Acesso em: 22 out. 2023.

- RIBEIRO, S. M. L.; MELO, C. M. D.; TIRAPEGUI, J. *Avaliação Nutricional - Teoria e Prática*. Grupo GEN, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788527733694>>. Acesso em: 21 abr. 2023.
- SIMAS, V. L. et al. *Desenvolvimento para dispositivos móveis*. [S.l.]: Grupo A, 2019.
- SOLE, A. D. *Visual Studio Code Distilled Evolved Code Editing for Windows, macOS, and Linux*. [S.l.]: Apress, 2019.
- TIWARI, S. *Professional NoSQL*. [S.l.]: John Wiley & Sons Inc, 2011.
- TYAGI, P. *Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web, & Desktop*. [S.l.]: CRC Press, 2022.
- VELHO, N. M. et al. *Antropometria: Uma Revisão Histórica do Período Antigo ao Contemporâneo. Comunicação, Movimento e Mídia na Educação Física*. [S.l.: s.n.], 1993.
- WIDMIL, E. *Flutter in Action*. [S.l.]: Manning Publications Co, 2020.
- ZAMMETTI, F. *Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK*. [S.l.]: Apress, 2019.
- ZENKER, A. M. et al. *Arquitetura de Sistemas*. Grupo A, 2019. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9788595029767>>. Acesso em: 13 Oct. 2023.