



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Matheus Ferreira Eziquiel**

**MEU TREINO:**

**APLICATIVO PARA GERENCIAMENTO DE ATIVIDADES FÍSICAS**

**CAMPINA GRANDE - PB**

**2022**

**Matheus Ferreira Eziquiel**

**MEUTREINO:**

**APLICATIVO PARA GERENCIAMENTO DE ATIVIDADES FÍSICAS**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**Orientador : José Antônio Beltrão Moura**

**CAMPINA GRANDE - PB**

**2022**

**Matheus Ferreira Eziquiel**

**MEUTREINO:**

**APLICATIVO PARA GERENCIAMENTO DE ATIVIDADES FÍSICAS**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**BANCA EXAMINADORA:**

**José Antão Beltrão Moura**

**Orientador – UASC/CEEI/UFCG**

**Marcus Salerno de Aquino**

**Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro**

**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 02 de Setembro de 2022.**

**CAMPINA GRANDE - PB**

## RESUMO

Com o crescimento do mercado *fitness*, a crescente conscientização sobre hábitos saudáveis torna-se cada vez mais comum o uso das tecnologias da informação para auxiliar o gerenciamento deste modo de vida. Entretanto, dificilmente os aplicativos de gerenciamento de treino oferecem a possibilidade de um profissional de educação física se conectar com seu aluno para que o treino seja criado e acompanhado, em sua maioria, essa responsabilidade é do próprio estudante, que na maioria dos casos, não possui o conhecimento necessário. Perante o exposto, surgiu o tema principal para este trabalho, o desenvolvimento de uma aplicação que auxilia pessoas adeptas deste modo de vida saudável a realizar o gerenciamento de suas atividades físicas e que permita a conexão de *personal-trainers* com seus alunos. O aplicativo disponibilizado para Android, foi desenvolvido em Flutter, um *framework open source*, e para armazenamento de dados, um servidor remoto utilizando Firebase. O objetivo do aplicativo é fornecer uma aplicação mobile na qual o usuário poderá estabelecer o contato com seu profissional de educação física para facilitar o gerenciamento de suas atividades físicas, de modo que aborda os processos de desenvolvimento de software, desde a fase de levantamento e análise de requisitos, projeto, implementação e por fim implantação.

# MeuTreino: Aplicativo para gerenciamento de atividades físicas

Trabalho de Conclusão de Curso

Matheus Ferreira Eziquiel  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil  
matheus.eziquiel@ccc.ufcg.edu.br

José Antão Beltrão Moura  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil  
antao@computacao.ufcg.edu.br

## Resumo

Com o crescimento do mercado *fitness*, a crescente conscientização sobre hábitos saudáveis torna-se cada vez mais comum o uso das tecnologias da informação para auxiliar o gerenciamento deste modo de vida. Entretanto, dificilmente os aplicativos de gerenciamento de treino oferecem a possibilidade de um profissional de educação física se conectar com seu aluno para que o treino seja criado e acompanhado, em sua maioria, essa responsabilidade é do próprio estudante, que na maioria dos casos, não possui o conhecimento necessário. Perante o exposto, surgiu o tema principal para este trabalho, o desenvolvimento de uma aplicação que auxilia pessoas adeptas deste modo de vida saudável a realizar o gerenciamento de suas atividades físicas e que permita a conexão de *personal-trainers* com seus alunos. O aplicativo disponibilizado para Android, foi desenvolvido em Flutter, um *framework open source*, e para armazenamento de dados, um servidor remoto utilizando Firebase. O objetivo do aplicativo é fornecer uma aplicação mobile na qual o usuário poderá estabelecer o contato com seu profissional de educação física para facilitar o gerenciamento de suas atividades físicas, de modo que aborda os processos de desenvolvimento de software, desde a fase de levantamento e análise de requisitos, projeto, implementação e por fim implantação.

## Palavras Chave

Gerenciador de treinos, exercícios físicos, *personal trainer*, aluno, aplicativo mobile.

## 1. INTRODUÇÃO

Atualmente, tecnologias da informação estão cada vez mais presentes na vida das pessoas. De acordo com um levantamento realizado pela Fundação Getúlio Vargas (FGV), o Brasil possui mais de um smartphone por habitante [1], e, segundo pesquisa promovida pelo Comitê Gestor da Internet do Brasil, 81% da população com mais de 10 anos têm internet em casa [2]. Diante disto, aplicativos móveis para realizar o controle financeiro, que auxiliam na saúde, educação, que atuam na área de entretenimento e esportes, tornam-se ainda mais práticos, principalmente quando essas ferramentas promovem uma solução mais otimizada.

Com a praticidade fornecida por essas tecnologias, e a facilidade em consumir esses serviços através em smartphones, em qualquer ambiente, acaba sendo cada vez mais comum a utilização desses dispositivos em academias, durante atividades físicas. Embora seja comum, ainda existem funcionalidades que são definidas de forma manual, que acabam consumindo tempo para serem executadas, como a definição dos treinos, a análise do progresso dos alunos por exemplo. Além disso, a maioria dos aplicativos que auxiliam neste gerenciamento, passam a responsabilidade de criar o plano de treinos para o aluno, que por não possuir o conhecimento necessário acabam definindo atividades físicas erradas, podendo ocasionar problemas de saúde [3].

Apresentado tal cenário, foi desenvolvido o MeuTreino, um aplicativo de gerenciamento de treinos que permite que *personal trainers* se conectem com seus alunos virtualmente, possibilitando o gerenciamento e acompanhamento de suas atividades físicas. Além disso, o aplicativo busca valorizar estes profissionais, que, acabam não exercendo seu trabalho pois existem outras aplicações que geram planos de treino sem a análise de um profissional de educação física.

## 2. SOLUÇÃO

Nesta seção será apresentado a solução para o problema anteriormente apresentado, como arquitetura, modelagem de dados e funcionalidades desenvolvidas no sistema.

### 2.1 Visão geral

O MeuTreino é um aplicativo para Android que permite que um *personal trainer* crie e gerencie as atividades físicas de seu aluno ou de qualquer outro usuário, de forma simples através de seu smartphone. O software promove vantagens tanto para o profissional, centralizando as informações de todos seus alunos e gerenciando o plano de treino de cada, quanto para o aluno, que sempre terá disponível seu treino e o acompanhamento de seu instrutor.

### 2.2 Funcionalidades

1. Interface distinta para um instrutor: Para possuir o controle de todos os seus alunos, os instrutores possuem uma interface intuitiva, que disponibilizará todos os seus usuários associados, permitindo selecionar um de seus alunos, podendo assim verificar os treinos atribuídos (Figura 1).

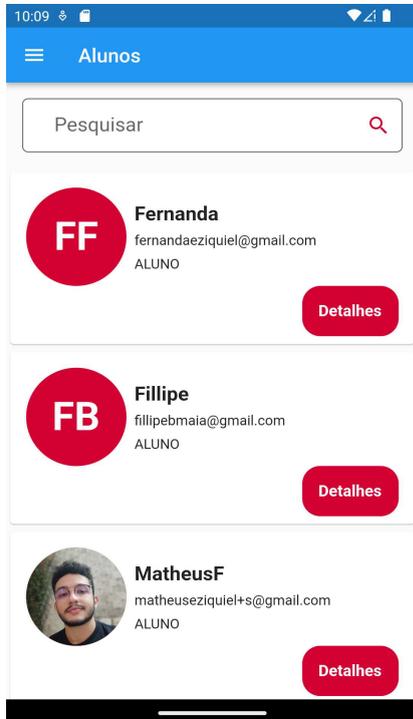


Figura 1 - Tela inicial do instrutor

2. Sistema para a criação do plano de treino: Para esta funcionalidade apenas o usuário do tipo “instrutor” poderá realizar esse tipo de operação.

Cada plano de treino possui 7 informações básicas que devem ser preenchidas no formulário de criação:

- a. O tipo de treino, que representa o objetivo deste treino, se é de hipertrofia, resistência, força muscular por exemplo;
- b. O nome do treino;
- c. A região que será trabalhada nesse treino, superior, inferior ou ambos;
- d. O músculo principal exercitado, bíceps, costas, quadríceps, por exemplo;
- e. A quantidade de séries;
- f. A quantidade de repetições;
- g. A cadência de execução do exercício em segundos, que consiste no ritmo que o treino é executado;
- h. O tempo de descanso entre séries em segundos;
- i. O dia da semana em que o treino será executado;

Clicando no botão no canto inferior direito, o *personal trainer* poderá adicionar mais

exercício ao treino, e para finalizar, basta clicar no canto superior direito em “Salvar” (Figura 2).

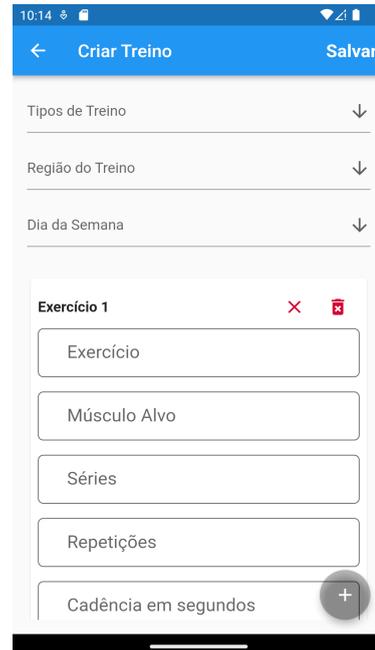


Figura 2 - Cadastro do treino

3. Interface distinta para um aluno: Este possuirá uma tela contendo todos os instrutores disponíveis para que seja possível se cadastrar a um *personal trainer* (Figura 3).

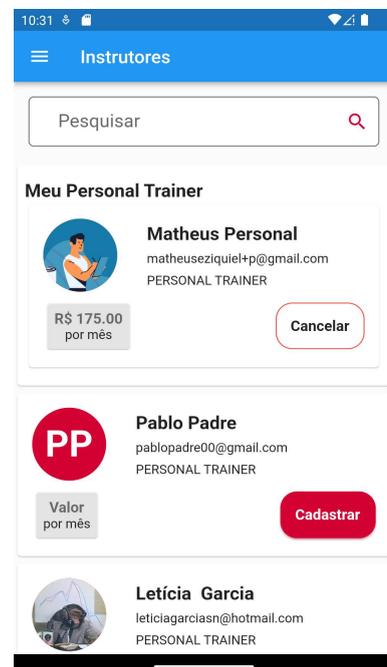


Figura 3 - Tela do aluno para listagem de instrutores

4. Listagem detalhada dos treinos: Nesta, ambos usuários poderão ver os treinos que estão associados para aquele específico aluno, trazendo as informações gerais para cada treino

da semana (Figura 4). Clicando no treino, o usuário terá acesso a uma visualização mais detalhada do plano de treino, visualizando assim, todos os exercícios (Figura 5).



Figura 4 - Detalhes do aluno



Figura 5 - Detalhes do treino

## 2.2 Arquitetura

Visando a persistência de dados por se tratar de uma aplicação móvel, o software utiliza a arquitetura *Serverless* [4] no back-end, que consiste em um modelo de desenvolvimento nativo em nuvem para o criação e

execução de aplicações sem realizar o gerenciamento de servidores [5]. Nele o *Google Cloud Platform* (GCP) é responsável pela manutenção, hospedagem e escala da infraestrutura do servidor, retirando essa responsabilidade do desenvolvedor, além de executar as funções da aplicação (Figura 6). O front-end será responsável por realizar a interação entre os dois tipos de usuário, *personal-trainer* e *aluno*.

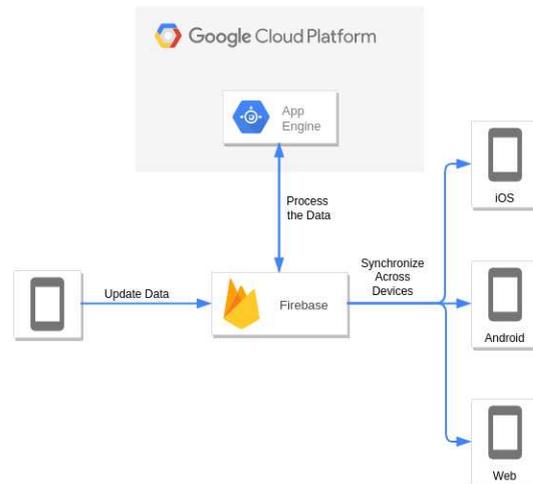


Figura 6 - Diagrama simplificado da arquitetura de uma aplicação com firebase

Fonte: Site Visual Paradigm<sup>1</sup>

### 2.2.1 Tecnologia front-end

Por se tratar de um aplicativo móvel, foi escolhida a tecnologia Flutter Framework [6] criada pela Google, devido a sua estratégia de renderização dos componentes na tela, a possibilidade de implementação tanto para Android quanto para iOS, a facilidade de implementação utilizando a linguagem *Dart*. Além de fornecer diversos componentes (*widgets*) prontos com uma ampla documentação com casos fáceis de compreender o uso, agilizando ainda mais o processo de desenvolvimento.

### 2.2.2 Autenticação de usuários

Para realizar a criação e autenticação dos usuários, foi utilizado o Firebase Authentication [7]. Este serviço fornece uma solução simples de autenticação que lida com os fluxos de UI (*user interface*) para realizar o login ou cadastro utilizando endereços de e-mail, e outros provedores de identidade, como o login do Google, com o FirebaseUI. Além da facilidade na implementação, fornece uma ótima segurança, pois este sistema do Firebase aplica o conhecimento interno do Google, que gerencia um dos maiores bancos de dados de contas do mundo.

### 2.2.3 Tecnologia back-end

A tecnologia definida para o back-end foi o Firebase [8]. O Firebase, assim como o Flutter, é uma plataforma

<sup>1</sup> Disponível em:

<https://online.visual-paradigm.com/pt/diagrams/template/s/google-cloud-platform-diagram/firebase-and-google-app-engine/>. Acesso em: 3 set. 2022.

desenvolvida pela Google. Ela foi escolhida pela facilidade na criação de um back-end destinado para aplicativos móveis. Essa plataforma fornece a sincronização de dados em tempo real, além de fornecer serviços de autenticação, execução de eventos no banco de dados, armazenamento de arquivos, monitoramento e hospedagem da aplicação, possibilitando um desenvolvimento mais rápido. Os serviços fornecidos por essa tecnologia utilizados foram: Firestore Database [9], Cloud Storage [10] e o Firebase Authentication, mencionado anteriormente.

O Firestore Database é um banco de dados NoSQL hospedado na nuvem e mantém seus dados em forma de documentos organizados em coleções. Esse modelo de organização de dados suporta uma ampla variedade de tipos de dados, além de promover uma facilidade para a realização de consultas, pois não é preciso realizar a relação entre várias entidades para obter um resultado.

O Cloud Storage é um serviço de armazenamento de objetos, que permite a realização de *uploads* e *downloads* de arquivos de forma mais confiável, pois são realizados independente da qualidade da rede, além de estar integrado com o Firebase Authentication, trazendo ainda mais segurança para a aplicação.

### 2.2.4 Estrutura de Dados

Por se tratar de um banco de dados não-relacional o Firebase não precisa de regras de negócio para sua implementação. Porém, foi desenvolvido o seguinte esquema (Figura 7) para a representação dos documentos, coleções e subcoleções presentes na aplicação.

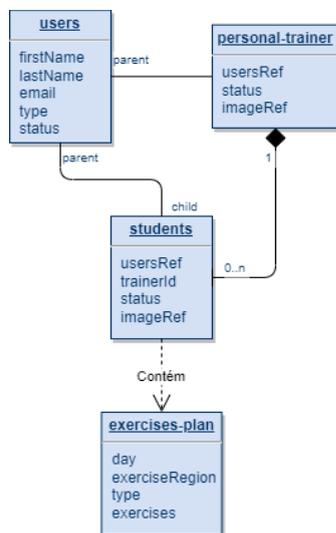


Figura 7 - Esquema de estrutura de dados.

Fonte: Elaborada pelo autor

Por ser um banco de dados NoSQL, a criação do exercício gera uma subcoleção em *students*, que permite o armazenamento de vários treinos, sem a necessidade de outra coleção, tornando as consultas mais rápidas e fáceis.

## 3. METODOLOGIA

Posteriormente, em contato com profissionais de educação física foi realizada a elicitación dos requisitos da aplicação, onde foi possível definir as principais funcionalidades, futuras melhorias que a aplicação pode fornecer e a definição do MVP (*Minimum Viable Product*).

### 3.1 Processo de Desenvolvimento

Para a etapa do desenvolvimento foi utilizado a metodologia ágil [11], seguindo alguns princípios do *Scrum* como: iterações de duas semanas, *product backlog*, e entrega de um MVP (*Minimum Viable Product*) ao final de cada iteração. Este processo foi dividido nas seguintes partes.

#### 3.1.1 Levantamento de requisitos

Para esta etapa, profissionais de educação física, que participaram como clientes desse projeto, e alguns de seus alunos elucidaram as necessidades existentes e diante disso foram escolhidas as duas principais funcionalidades do sistema: estabelecer a relação *personal trainer* e aluno e permitir que o instrutor gerencie os treinos de seus alunos.

Para colher essas informações, foi disponibilizado um formulário online aos usuários contendo as seguintes perguntas:

1. Nome do usuário;
2. Tipo do usuário (Instrutor/Aluno)
3. Utiliza/Utilizou algum aplicativo semelhante?
4. Caso positivo para a pergunta 3, qual a principal diferença do aplicativo proposto?
5. Qual a vantagem do aplicativo proposto?
6. Qual a desvantagem do aplicativo proposto?

O template do formulário pode ser obtido através do seguinte link

([Formulário - MeuTreino - Formulários Google.pdf](#))

Os resultados obtidos podem ser visualizados através desta planilha

(<https://docs.google.com/spreadsheets/d/1XUchtSmaTxbdw7EbquiCwDsGam4kcf-z2V9VgducoaY/edit?usp=sharing>).

#### 3.1.2 Análise de Requisitos

Após a coleta dos resultados, foi possível identificar as principais funcionalidades e realizar a definição do MVP para cada uma, de modo a tornar a próxima etapa, de implementação, mais organizada.

#### 3.1.3 Implementação

Primeiramente, foi discutido as tecnologias que seriam utilizadas no projeto. Essa decisão levou em consideração alguns fatores como a experiência em outros trabalhos com as ferramentas, a alta no mercado, a ampla documentação oferecida e que reduziriam o tempo de desenvolvimento, além de gerar um bom produto final.

Esta etapa ocorreu com iterações de duas semanas, para cada funcionalidade apresentada abaixo:

1. Modelagem dos dados. Mesmo utilizando um sistema não-relacional e documentos para o armazenamento de dados, foi definida a sua estrutura para manter a organização do projeto e facilitar nas próximas etapas.
2. Implementação das classes de serviço, para realizar as requisições da aplicação com o Firebase.
3. Implementação da interface do usuário instrutor.
4. Implementação da interface do usuário aluno.
5. Implementação de *streams* [12] no front-end. Esta funcionalidade foi crucial para o sistema pois permite que o usuário receba as informações alteradas em tempo real.
6. Realizar melhorias, corrigir bugs encontrados pelos usuários.

Após a finalização de cada iteração, era disponibilizada uma nova versão do aplicativo para os usuários.

### 3.1.4 Repasse do Código-Fonte

O aplicativo está disponível em um repositório<sup>2</sup> público no GitHub, que, para facilitar sua manutenção e implementação de novas funcionalidades, em resumo, permitir a escalabilidade da aplicação, foi desenvolvido seguindo uma estrutura de diretórios. Essa estrutura se baseia na organização dos arquivos por seu tipo ou sua funcionalidade [13]. A figura 7 representa a organização destes arquivos na aplicação.

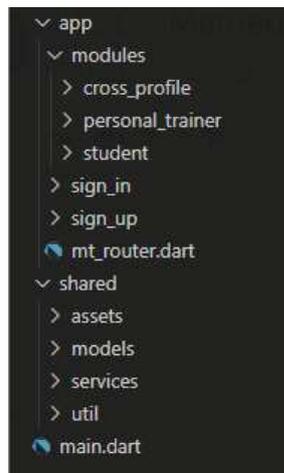


Figura 7 - Estrutura de diretórios da aplicação

Os *modules* consistem nas telas da aplicação e são divididos em 3 partes: *student* e *personal\_trainer* representam as interfaces exclusivas para os alunos e instrutores respectivamente, já *cross\_profile* é destinado para as interfaces que ambos usuários podem utilizar.

Em *shared* está localizado todas as funcionalidades que os *modules* possuem que são reutilizadas entre si, como por exemplo os *models*. Neste, possui as entidades existentes na aplicação, contendo toda sua estrutura de dados a ser armazenada no banco de dados, que, no Flutter, possui uma documentação ampla para a sua

serialização<sup>3</sup>. Além disso, contém o diretório *services*, responsável por toda a lógica de requisição ao Firebase para cada *model* existente na aplicação.

## 4. Avaliação e Experiência

### 4.1 Avaliação dos usuários

Para avaliar o grau de satisfação dos usuários com o MeuTreino, foi utilizado o modelo de questionário PSSUQ (*Post-Study System Usability Questionnaire*) [14]. Esse questionário foi disponibilizado para 2 profissionais de educação física e 10 usuários normais. O uso do aplicativo foi realizado por cerca de uma semana onde foi solicitado seu uso rotineiramente, de modo que todos utilizassem todas as funcionalidades, cadastrando novos planos de treinos, associando usuários à *personal trainers*, por exemplo.

O PSSUQ é composto por 15 questões com 7 opções para escolher com a escala de usabilidade entre “Concordo Totalmente” e “Discordo Totalmente”. Para a obtenção do resultado, foi calculado a média aritmética das respostas em cada item do questionário. O resultado de satisfação é representado na imagem abaixo (Figura 8).

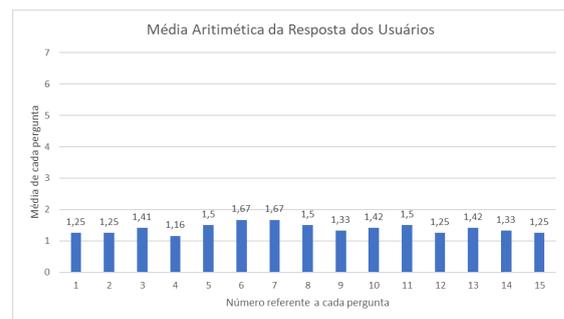


Figura 8 - Gráfico de barras com média aritmética das respostas dos usuários.

A partir dos resultados obtidos, foi observado que a variação das respostas ocorreram no intervalo de 1,25 e 1,67, permitindo que seja analisado, a partir do modelo PSSUQ, que os usuários concordam com o que foi questionado. Assim, é possível constatar que os usuários estavam satisfeitos com a aplicação para o que foi originalmente proposto.

Porém, mesmo diante do nível de satisfação apresentado, foi verificado que ainda existe um grau de insatisfação quando questionado sobre como o sistema se comporta diante de erros do usuário, representado pelas perguntas 6 e 7, que possuem as maiores médias, 1,67 para ambas. Para solucionar estes problemas e melhorar a usabilidade da aplicação, foi implementado componentes visuais para exibição de erros, além de aumentar a cobertura dos erros apresentados no aplicativo, como por exemplo campos inválidos, erros de conexão com o servidor.

<sup>3</sup> JSON and serialization. [S. l.], [202-]. Disponível em: <https://docs.flutter.dev/development/data-and-backend/json>. Acesso em: 3 set. 2022.

<sup>2</sup> Disponível em: [https://github.com/Ferreiramfe/mt\\_app](https://github.com/Ferreiramfe/mt_app)

Além disso, ao analisar os resultados obtidos, o questionamento se o sistema tem todas as funções e capacidades que o usuário esperava ter, pergunta 14, foi verificado que alguns usuários não concordavam totalmente. Estes por sua vez foram reportados pelos usuários para que fosse possível utilizar essas informações para realizar melhorias na aplicação.

O template do formulário pode ser obtido através do seguinte link

([MeuTreino - Questionário - Formulários Google.pdf](#))

Os resultados obtidos podem ser visualizados através desta planilha

(<https://docs.google.com/spreadsheets/d/1JpROi9IIQaMyRVchGORaFF113GHoScRCRO84cMPkvaM/edit?usp=sharing>).

## 4.2 Experiência: Desafios e Limitações

No processo de desenvolvimento, o principal desafio encontrado foi criar uma interface simples, intuitiva e amigável para o usuário. A cada entrega do MVP foi possível coletar o *feedback* do usuário e a partir disso, foram realizadas melhorias para tornar a experiência do usuário mais confortável, como por exemplo a construção do formulário para criar o plano de treino, para tornar ainda mais simples essa operação por parte do instrutor. Para implementar a interface, realizar a padronização dos componentes da tela, foi estudado os princípios de *Material Design* [15] e sobre *User Interface (UI)* [16], o que contribuiu bastante com o uso do *framework* Flutter, por possuir diversos componentes prontos que são facilmente editados e com ampla comunidade de suporte.

Além disso, a modelagem da estrutura dos dados era algo essencial para o desenvolvimento da aplicação, pois permite manter a consistência e organização dos dados tramitados no sistema. Porém, a escolha do Firestore Database, que utiliza um sistema de coleções e documentos, gerou esse desafio, por não necessitar de um sistema de tipagem de dados em sua estrutura. Dessa forma, a responsabilidade para garantir a integridade dos dados foi realizada fora da base de dados, utilizando a abordagem de *Domain Models* [16] que permite descrever entidades, as informações que são tramitadas, além de definir a estrutura dessas informações. Essa abordagem trouxe diversos benefícios para a aplicação, fornecendo facilidade na visualização dos dados, maior tolerância a erros, além de tornar mais simples uma possível refatoração do código.

Quanto às limitações, houve um impedimento em desenvolver a aplicação para iOS, pois para isso, é necessário possuir um ambiente de desenvolvimento configurado para esse sistema.

## 5. Conclusões e Trabalhos Futuros

### 5.1 Conclusões

Este trabalho apresentou o desenvolvimento de uma aplicação *mobile*, seguindo a metodologia ágil com alguns princípios do *Scrum*, desde a etapa de levantamento de requisitos, escolha de tecnologias, implementação da aplicação, até a disponibilização de uma versão utilizável do sistema, que permitiu a aplicação do questionário apresentado para avaliar o grau de satisfação do usuário final.

Portanto, concluímos que, após os resultados coletados com o formulário de satisfação disponibilizado e com a resposta média do grau de satisfação com o sistema representado abaixo pela figura 9, onde em 1 o usuário concorda totalmente com a pergunta, o objetivo deste trabalho, que consiste em permitir que um profissional de educação física realize o gerenciamento de atividades físicas de seus alunos e que o aluno consiga se conectar com seu instrutor, através de uma aplicação com uma interface simples e de fácil entendimento, com sincronismo dos dados entre usuários foi alcançado.

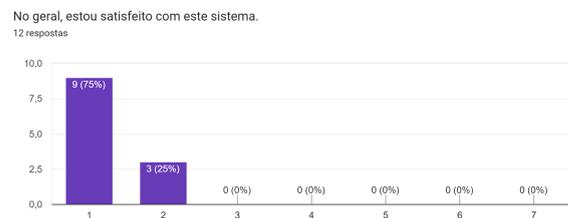


Figura 9 - Mdia de satisfao dos usurios com o sistema.

### 5.2 Trabalhos Futuros

Para trabalhos futuros, pretendemos adicionar a aplicao as funcionalidades que possuiam menor prioridade e assim no foram implementadas.

Essas permitem que os alunos consigam solicitar ao seu instrutor um novo treino ou a edio do mesmo. Outro ponto  fornecer aos usurios a possibilidade de adicionarem mais informaes, como suas medidas, para que seja verificado o seu progresso com o decorrer do tempo.

Foi verificado com a anlise de requisitos, a funcionalidade para trazer ao MeuTreino uma possvel nova camada de usurios, possibilitando o gerenciamento de planos alimentares. Dessa forma, buscamos aumentar a busca por esse servio e incorporar nutricionistas como usurios da aplicao, sendo responsveis por gerenciar a dieta de seus clientes.

Alm de melhorias visuais, tambm pretendemos usar um ambiente de desenvolvimento para iOS, para que seja possvel disponibilizar o aplicativo para esse sistema.

## 6. Referncias

[1] BRASIL tem mais smartphones que habitantes, aponta FGV. CNN Brasil, [S. l.], p. <https://www.cnnbrasil.com.br/business/brasil-tem-mais-smartphones-que-habitantes-aponta-fgv/#:~:text=O%20Brasil%20tem%20atualmente%20mais,de%20acordo%20com%20o%20IBGE,26 maio 2022>.

- [2] BRASIL tem 152 milhões de pessoas com acesso à internet. Agência Brasil, [S. l.], p. <https://agenciabrasil.ebc.com.br/geral/noticia/2021-08/brasil-tem-152-milhoes-de-pessoas-com-acesso-internet>, 23 ago. 2021.
- [3] ORTOPEDISTA alerta sobre prática errada de exercícios físicos. Jornal O Celeiro, [S. l.], p. <http://jornalceleiro.com.br/2018/08/ortopedista-alerta-sobre-pratica-errada-de-exercicios-fisicos/>, 11 ago. 2018.
- [4] MARTINFOWLER. *In*: Serverless Architectures. [S. l.], 22 maio 2018. Disponível em: <https://martinfowler.com/articles/serverless.html>. Acesso em: 3 ago. 2022.
- [5] REDHAT. *In*: O que é serverless?. [S. l.], 31 out. 2017. Disponível em: <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-serverless>. Acesso em: 3 ago. 2022.
- [6] Flutter - Framework para desenvolvimento mobile. Disponível em: [https://flutter.dev/?gclid=Cj0KCQjwgO2XBhCaARIsANrW2X3W-ETNSH0DYgI5Dr98Ev\\_7GVqgxwIN77VMnhw-bjOWQ1tArCOKYvUaAmxkEALw\\_wcB&gclid=awds](https://flutter.dev/?gclid=Cj0KCQjwgO2XBhCaARIsANrW2X3W-ETNSH0DYgI5Dr98Ev_7GVqgxwIN77VMnhw-bjOWQ1tArCOKYvUaAmxkEALw_wcB&gclid=awds). Acesso em: 28 jul. 2022.
- [7] FIREBASE Authentication. [S. l.], 2022. Disponível em: <https://firebase.google.com/docs/auth?hl=pt-br>. Acesso em: 3 ago. 2022.
- [8] FIREBASE. *In*: Make your app the best it can be. [S. l.], 2022. Disponível em: <https://firebase.google.com/>. Acesso em: 3 ago. 2022.
- [9] FIREBASE. *In*: Escolher um banco de dados: Cloud Firestore ou Realtime Database. [S. l.], 2022. Disponível em: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore>. Acesso em: 3 ago. 2022.
- [10] FIREBASE. *In*: Cloud Storage para Firebase. [S. l.], 2022. Disponível em: <https://firebase.google.com/docs/storage>. Acesso em: 3 ago. 2022.
- [11] FIA BUSINESS SCHOOL. *In*: Scrum: o que é e como aplicar a metodologia ágil para gestão?. [S. l.], 21 fev. 2020. Disponível em: <https://fia.com.br/blog/scrum/>. Acesso em: 2 ago. 2022.
- [12] MACORATTI, José Carlos. Apresentando o conceito de Stream. [S. l.], [201-]. Disponível em: [https://macoratti.net/19/12/flut\\_stream1.htm](https://macoratti.net/19/12/flut_stream1.htm). Acesso em: 2 set. 2022.
- [13] SCALABLE Flutter Folder Structure for Big Apps. [S. l.], 19 jun. 2022. Disponível em: <https://levelup.gitconnected.com/scalable-flutter-folder-structure-for-big-apps-480a9b266793>. Acesso em: 3 set. 2022.
- [14] WILL T. *In*: PSSUQ (Post-Study System Usability Questionnaire). [S. l.]. Disponível em: <https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire/>. Acesso em: 4 ago. 2022.
- [15] INTRODUCTION - Material Design. [S. l.]. Disponível em: <https://material.io/design/introduction>. Acesso em: 5 ago. 2022.
- [16] USER Interface Design Basics. [S. l.]. Disponível em: <https://www.usability.gov/what-and-why/user-interface-design.html>. Acesso em: 5 ago. 2022.
- [17] DOMAIN Modeling. [S. l.], 2021. Disponível em: <https://www.scaledagileframework.com/domain-modeling/>. Acesso em: 2 set. 2022.