

# DESENVOLVIMENTO DE UMA API RESTFUL PARA APLICATIVOS DA SAÚDE NO ESPORTE

Luiz Felipe Marcondes de Lima<sup>1</sup>, Giovane Galvão<sup>2</sup>

<sup>1</sup>Discente do Centro Universitário Campo Real

<sup>2</sup>Docente do Centro Universitário Campo Real

Rua Comendador Norberto, 1299 - Santa Cruz - Guarapuava - PR - Brasil

{[engs-luizlima@camporeal.edu.br](mailto:engs-luizlima@camporeal.edu.br), [prof\\_giovanegalvao@camporeal.edu.br](mailto:prof_giovanegalvao@camporeal.edu.br)}

## Resumo

Com base na fragmentação dos aplicativos voltados à saúde esportiva, que dificulta a integração e a entrega de serviços na área, este trabalho desenvolveu uma *Application Programming Interface* (API) baseada na arquitetura *Representational State Transfer* (REST) para unificar a gestão de treinos, dietas e rotinas na medicina esportiva. Utilizando metodologias ágeis, como *Design Thinking* e *Scrumban*, e fundamentado em uma pesquisa de mercado que identificou a escassez de soluções colaborativas, o projeto teve como objetivo implementar um ambiente centralizador para a interação entre profissionais (treinadores e nutricionistas) e seus alunos e/ou pacientes. Os resultados demonstram que a solução proposta preenche a lacuna de plataformas unificadas, oferecendo rotas flexíveis que aprimoram a experiência do usuário e apresentam tempo médio de resposta de 577 milissegundos (ms), contribuindo para uma comunicação ágil e eficiente entre os componentes da API.

**Palavras-chave:** Saúde esportiva. API. REST. Metodologias ágeis. Plataforma unificada. Experiência do usuário

## Abstract

Based on the fragmentation of applications focused on sports health, which makes integration and delivery of services in the area difficult, this work developed an *Application Programming Interface* (API) based on the *Representational State Transfer* (REST) architecture to unify the management of training, diets and routines in sports medicine. Using

agile methodologies, such as Design Thinking and Scrumban, and based on market research that identified the scarcity of collaborative solutions, the project aimed to implement a centralized environment for interaction between professionals (trainers and nutritionists) and their students and/or patients. The results demonstrate that the proposed solution fills the gap of unified platforms, offering flexible routes that improve the user experience and have an average response time of 577 milliseconds (ms), contributing to agile and efficient communication between API components.

**Keywords:** Sports health. API. REST. Agile methodologies. Unified platform. User experience

## 1 INTRODUÇÃO

A crescente digitalização da área da saúde tem impulsionado o desenvolvimento de plataformas voltadas ao bem-estar físico. De acordo com o relatório *Brazil MHealth Apps Market Size & Outlook, 2024-2030*, o mercado brasileiro de aplicativos móveis de saúde (*mHealth*) movimentou aproximadamente US\$ 713,5 milhões em 2023, com previsão de alcançar US\$ 2,019 bilhões até 2030 (GRAND VIEW RESEARCH, 2024). No cenário global, observa-se um crescimento expressivo no número de aplicativos digitais de saúde voltados ao consumidor. Segundo o relatório *Digital Health Trends 2024* do Instituto IQVIA, o total de aplicativos de saúde disponíveis em lojas virtuais passou de 66.713 em 2013 para 337.522 em 2024. Embora o volume total de aplicativos tenha atingido números recordes, o estudo aponta que mais da metade dos aplicativos desenvolvidos desde 2008 já não estão mais disponíveis comercialmente, refletindo um processo de consolidação e amadurecimento do mercado global (IQVIA INSTITUTE, 2024).

Entretanto, apesar da popularização dos aplicativos de saúde, observa-se uma grande fragmentação no ecossistema digital voltado ao cuidado com o corpo e a qualidade de vida. Estudos apontam que a proliferação excessiva de aplicativos de saúde gera um fenômeno conhecido como *app overload*, no qual profissionais e usuários têm dificuldade para encontrar e utilizar as ferramentas mais adequadas devido à fragmentação das informações e funcionalidades distribuídas em múltiplas plataformas (Van Velsen et al., 2013)

Em ambientes corporativos a adesão de várias soluções diferentes gera constante alternância entre aplicativos, que não apenas consome tempo, cerca de quatro horas semanais por funcionário, mas também aumenta o estresse cognitivo e fragmenta a atenção (Murty, Dadlani e Das, 2022). Além disso, gera custos mais altos, onde o usuário paga por assinaturas

acumuladas, incluindo aplicativos que oferecem funções sobrepostas, ou seja, semelhantes, mas mesmo assim, os usuários continuam pagando por ela (Kovalskiy, 2025).

Diante desse cenário, torna-se essencial o desenvolvimento de soluções que reduzam essa sobrecarga, promovendo integração, centralização e interoperabilidade entre diferentes serviços e perfis de usuários. Um meio de mitigar esse cenário é consolidar as aplicações utilizadas através de integrações inteligentes, conforme recomendado por especialistas em tecnologia no local de trabalho (CXAPP, 2025). Deste modo, este projeto propõe o desenvolvimento de uma API *RESTful* unificada, que funcione como um *gateway* de integração entre profissionais da saúde esportiva, como nutricionistas e treinadores e o aluno/paciente, oferecendo um ambiente único para o gerenciamento de treinos, dietas e rotinas. Essa abordagem busca não apenas mitigar o problema identificado, mas também contribuir para um ecossistema mais coeso, acessível e eficiente no contexto da saúde esportiva digital.

## 1.1 OBJETIVOS

A fragmentação dos softwares na saúde do esporte abriu oportunidades para melhorar a qualidade de vida do usuário final. Sendo assim, o presente trabalho estabelece o objetivo de desenvolver uma plataforma digital que promova uma colaboração entre nutricionistas, treinadores e alunos, centralizando o gerenciamento de serviços e atuando como porta de entrada para a unificação do ecossistema da saúde digital. Para alcançar o objetivo geral do projeto, foram definidos objetivos específicos que abrangem tanto metas técnicas quanto a criação de artefatos essenciais para validar, garantir a segurança e melhorar a usabilidade da API. Entre esses objetivos, inclui-se assegurar a implementação da arquitetura REST, disponibilizar documentação completa que facilite o uso da API e disponibilizar a API para uso público. Além disso, prevê-se a implementação de um mecanismo de autenticação segura utilizando JSON Web Token (JWT), fortalecendo a proteção dos dados e o controle de acesso.

## 1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está estruturado em seis capítulos, com o objetivo de apresentar de forma clara e objetiva todas as etapas envolvidas no projeto. A seguir, descreve-se brevemente o conteúdo de cada capítulo:

- Capítulo 1: Apresenta o contexto do problema, a motivação para o desenvolvimento da solução proposta, os objetivos geral e específicos, a relevância do projeto e a organização do trabalho.

- Capítulo 2: Reúne os principais conceitos relacionados ao tema, incluindo definições de API, arquitetura seguindo a metodologia REST e o panorama atual dos aplicativos voltados à saúde no esporte, ferramentas utilizadas no desenvolvimento e metodologias.
- Capítulo 3: Apresenta uma análise do mercado e da literatura acadêmica sobre plataformas de saúde esportiva. Evidencia a fragmentação existente tanto nas funcionalidades oferecidas (treino, nutrição e rotina) quanto na colaboração entre profissionais, justificando a necessidade da solução proposta.
- Capítulo 4: Descreve como as metodologias de *Design Thinking* e *Scrumban* foram implementadas e suas importâncias no desenvolvimento da aplicação.
- Capítulo 5: Discorre sobre o desenvolvimento e a performance da API *RESTful*, validando seu funcionamento por meio de testes e métricas. Analisa os resultados, evidenciando as contribuições da solução para resolver o problema da fragmentação no mercado de saúde esportiva.
- Capítulo 6: Conclui o trabalho, resumindo suas contribuições e o alcance dos objetivos. Discute as limitações do projeto e sugere oportunidades para pesquisas e desenvolvimentos futuros.

A estrutura adotada visa garantir uma abordagem progressiva e fundamentada para o entendimento do problema, desenvolvimento da solução e análise de sua aplicabilidade no contexto da saúde esportiva digital.

## **2 REFERENCIAL TEÓRICO**

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento do projeto proposto, abordando conceitos relacionados à utilização de aplicativos na área de saúde voltada ao esporte, APIs e ao modelo REST. A seguir, são discutidos os principais conceitos que sustentam a solução tecnológica deste trabalho.

### **2.1 APLICATIVOS DE SAÚDE NO ESPORTE**

A saúde no esporte tem ganhado destaque devido ao aumento da preocupação com o bem-estar físico e a qualidade de vida. Segundo a Organização Mundial de Saúde (OMS), em 2022, 2,5 bilhões de adultos de 18 anos ou mais estavam acima do peso, incluindo 890 milhões, que estavam com obesidade, correspondendo a cerca de 43% dos adultos (WHO, 2025).

Os aplicativos de saúde no esporte surgiram como ferramentas para promover a atividade física e o acompanhamento nutricional. Eles oferecem funcionalidades como planos de treino personalizados, monitoramento de dieta, rastreamento de atividades físicas e integração com *wearables* (dispositivos vestíveis, como *smartwatches* e *fitness bands*). Alguns exemplos populares são citados no Quadro 1.

Quadro 1 - Aplicativos Tendência no Mercado MHealth

Aplicativo	Descrição
Growth Dieta e Treino	É um aplicativo que oferece planos alimentares e de exercícios baseados em metas individuais, permitindo o acompanhamento de dieta, treinos e hidratação (GROWTH SUPPLEMENTS, 2025).
MyFitnessPal	Uma aplicação que auxilia o usuário no acompanhamento alimentar através de registros de refeições, receitas e dicas e suporte da comunidade (MYFITNESSPAL, 2025).
Strava	Focado em corridas e ciclismo, Strava ajuda os usuários a entenderem seu progresso e desempenho além de auxiliar no treino através de dicas personalizadas por inteligência artificial (STRAVA, 2025).
Freeletics	Oferece dietas e planos de treinos personalizados para cada usuário, acompanhamento do progresso e um treinador digital que motiva e oferece conselhos aos usuários. (FREELETICS, 2025).

Fonte: Elaborado pelo autor, 2025.

Esses aplicativos têm contribuído significativamente para a motivação dos usuários, oferecendo *feedback* em tempo real, gamificação (recompensas por metas alcançadas) e comunidades online para compartilhamento de progresso.

## 2.2 API

Uma API é um conjunto de ferramentas, definições e protocolos responsáveis por permitir a conexão entre um ou mais sistemas (RED HAT, 2023).

O uso de APIs justifica-se pelos diversos benefícios que proporcionam, especialmente no contexto da integração de sistemas. APIs permitem tanto integrações internas entre os

próprios departamentos ou sistemas de uma empresa quanto integrações externas conectando parceiros, fornecedores, clientes e serviços de terceiros. Dessa forma, contribuem significativamente para a automação de processos, a escalabilidade de soluções e a ampliação de ecossistemas digitais. Além disso, uma API bem projetada pode ser utilizada não apenas por seus desenvolvedores originais, mas também por usuários externos, ampliando seu alcance e utilidade (Biehl, 2015).

No contexto deste trabalho, a API será o núcleo da arquitetura proposta, servindo como o ponto central de possíveis integrações entre aplicativos que conectam os profissionais de saúde esportiva (nutricionistas, treinadores) e os alunos/pacientes.

Com isso, torna-se mais rápida e eficiente a disponibilidade de funcionalidades como recomendações personalizadas de treino, plano alimentar e acompanhamento da rotina, graças à agilidade proporcionada pela integração via API.

### 2.3 MODELO REST

O modelo REST é um estilo arquitetural que combina elementos de diferentes estilos de rede, como cliente-servidor e sistemas em camadas, adicionando restrições que definem uma interface uniforme entre os componentes. As APIs que seguem os princípios REST são amplamente adotadas em sistemas distribuídos devido às suas características de simplicidade, escalabilidade e compatibilidade, herdadas da arquitetura proposta por Fielding, especialmente em aplicações *web* e móveis (Fielding, 2000).

O REST foi projetado para utilizar de forma ideal a infraestrutura do protocolo HTTP, tendo como conceito central o recurso (*resource*), que estabelece uma interface uniforme entre cliente e servidor (Biehl, 2015).

As principais restrições do estilo REST incluem:

- Uso extensivo das capacidades do protocolo HTTP;
- Definição de recursos em vez de métodos;
- Interface uniforme baseada em métodos HTTP;
- Comunicação sem estado (*stateless*);
- Baixo acoplamento e independência das requisições;
- Utilização de códigos de status HTTP adequados.

### 2.4 FERRAMENTAS E TECNOLOGIAS

O desenvolvimento da API *RESTful* proposta neste trabalho faz uso de um conjunto de ferramentas e tecnologias modernas, amplamente adotadas no mercado de *software*, com foco

em escalabilidade, segurança e facilidade de integração. A seguir, no Quadro 2 são descritas as principais tecnologias envolvidas.

Quadro 2 - Ferramentas e Tecnologias Utilizadas no Desenvolvimento da API

Tecnologia	Descrição
Laravel	<i>Framework</i> utilizado como base para o desenvolvimento <i>backend</i> da API. O Laravel permite a construção de aplicações <i>web</i> utilizando a linguagem PHP, oferecendo diversos recursos integrados, como autenticação, rotas e <i>Object-Relational Mapper</i> (ORM) (LARAVEL, 2025). O framework se destaca pela eficiência do código, estrutura de roteamento robusta e aderência ao padrão arquitetural <i>Model, View e Controller</i> (MVC), oferecendo vantagens significativas em organização, manutenibilidade e desenvolvimento ágil de APIs <i>RESTful</i> (Ariyanto et al., 2024).
PostgreSQL	Banco de dados relacional adotado para o gerenciamento dos dados da aplicação pela sua gratuidade e a familiaridade que o desenvolvedor possui (POSTGRESQL, 2025).
Postman	Ferramenta utilizada para realizar testes de integração e desempenho da API. Permite simular requisições e validar o comportamento da aplicação, além disso, é utilizada para a geração automática da documentação da API com base nas requisições dentro da plataforma (POSTMAN, 2025).
JWT	Mecanismo de autenticação e autorização utilizado para a segurança das rotas. JWTs são uma forma compacta de transmitir informações com segurança entre partes como um objeto <i>JavaScript Object Notation</i> (JSON), sendo amplamente adotado em cenários de autorização e troca de informações em APIs <i>RESTful</i> (JSON WEB TOKEN, 2025).

Fonte: Elaborado pelo autor, 2025.

A escolha dessas ferramentas visa garantir uma solução robusta, segura e com potencial de integração com outros sistemas do ecossistema digital de saúde.

## 2.5 METODOLOGIAS DE GESTÃO E DESENVOLVIMENTO

O desenvolvimento de uma aplicação complexa, como uma API, depende da adoção de metodologias de gestão e desenvolvimento que garantam eficiência, adaptabilidade e alinhamento com as necessidades do usuário final. Para este projeto, optou-se por uma abordagem híbrida e pragmática, combinando princípios do *Design Thinking e Scrum*.

### 2.5.1 Design Thinking

O *Design Thinking* é uma abordagem centrada no ser humano para resolver problemas complexos de forma criativa e inovadora. Desenvolvida por David Kelley e Tim Brown, esta metodologia é um processo iterativo que busca entender profundamente as necessidades, experiências e pontos de dor dos usuários, olhando as situações pelo ponto de vista do cliente (ALT, 2017; BROWN, 2008). A metodologia consiste de 5 passos: Imersão e Empatia, que corresponde à compreensão do contexto do problema através de entrevistas, observação e análises; Definição, que é a síntese das informações coletadas para definir com clareza o problema central a ser resolvido; Ideação, que envolve a geração de um grande volume de ideias e possíveis soluções, sem julgamentos, através de técnicas como brainstorming; Prototipagem, que se refere à criação de representações, como protótipos; e Teste, que consiste na validação dos protótipos com usuários reais, coletando feedback para refinamento e iteração contínua da solução. Sua utilização é importante na inovação para não criar soluções desconexas dos reais problemas do usuário.

### 2.5.2 Scrum

O Scrum é uma metodologia que torna o processo de desenvolvimento mais visível, permitindo ao usuário, ter uma melhor percepção do que está sendo desenvolvido dentro do projeto e também que ele faça os ajustes necessários para que o projeto continue sendo desenvolvido de forma pragmática. O progresso do Scrum baseia-se em etapas chamadas de *sprints*, com duração média e recomendada de 2 a 4 semanas, onde cada *sprint* é precedida de uma reunião de planejamento, onde o time de desenvolvimento, juntamente com o *Product Owner* (PO), estimam e definem as prioridades de cada tarefas a serem realizadas dentro de cada *sprint* (Pereira et al., 2007).

A utilização da metodologia Scrum define três funções essenciais no desenvolvimento do projeto, como representado no Quadro 3.

Quadro 3 - Cargos e Funções da Metodologia Scrum

Cargo	Descrição
<i>Scrum Master</i>	Responsável por garantir que os valores, práticas e regras do Scrum sejam cumpridos.
<i>Product Owner</i>	Mediador entre a equipe e as partes interessadas, o <i>Product Owner</i> gerencia o <i>Product Backlog</i> e decide quando o produto deve ser entregue, sempre com predisposição para entregas mais frequentes.
Time de Desenvolvimento	A equipe é responsável por determinar quanto trabalho pode completar durante cada <i>sprint</i> e se comprometem com a entrega de cada etapa.

Fonte: Elaborado pelo autor, 2025.

Além dos papéis, o *Scrum* utiliza artefatos como *Product Backlog* (lista priorizada de funcionalidades desejadas), *Sprint Backlog* (tarefas selecionadas para o *sprint* atual) e gráficos de *Burndown* (visualização do progresso), bem como cerimônias como *Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint Retrospective*, que proporcionam estrutura e transparência ao processo de desenvolvimento (Sachdeva, 2016).

### 2.5.3 Kanban

O Kanban é uma metodologia criada no Japão, praticada na Toyota por iniciativa do engenheiro Taiichi Ohno. Segundo especialistas, o Kanban é um meio de gerir as mudanças de um projeto com ênfase na visualização do trabalho em andamento, limitando a demanda para cada etapa do projeto, medir as consequências das decisões tomadas em relação ao fluxo de trabalho e por fim, identificar melhorias no projeto a fim de manter uma cultura onde o projeto é continuamente melhorado (Boeg, 2010).

O método Kanban, utiliza de quadros *kanban* a fim de facilitar a visualização das informações. O quadro *kanban* não tem um padrão a ser seguido, permitindo a customização para a adesão em qualquer projeto, se limitando apenas ao espaço físico ou a aplicação usada.

### 2.5.4 Scrumban

O *Scrumban*, metodologia proposta inicialmente por Corey Ladas, combina elementos do Scrum e do Kanban (Ladas, 2009). Como fusão de dois métodos ágeis, visa preencher possíveis lacunas de ambos os métodos de forma integrada. Entre as vantagens dessa

metodologia híbrida, pode-se destacar a flexibilidade para adaptar-se a mudanças repentinas nos projetos sem causar sobrecarga significativa na equipe de desenvolvimento (Reis; Augusto Albuquerque, 2021).

As principais qualidades dessa metodologia híbrida são:

- O *Scrumban* oferece flexibilidade permitindo a inclusão de alterações durante o processo de desenvolvimento do projeto, tornando a utilização de *sprints* opcional se o fluxo contínuo for suficiente.
- Não é necessário a conclusão de uma *sprint* para entregar resultados, os recursos podem ser entregues conforme são finalizados.
- Limitação de trabalho em progresso para se adaptar a carga de trabalho que a equipe consegue suportar.
- Maior colaboração devido a disponibilização dos cartões em um quadro *kanban*.

O *Scrumban* dispensa alguns papéis tradicionais do Scrum e os membros da equipe gerenciam suas próprias *sprints* (ATLASSIAN, 2025).

## 2.6 HEURÍSTICA VADER PARA TESTE DE API

A heurística VADER é uma técnica proposta por Stuart Ashman para orientar a realização de testes em APIs *RESTful* sem que seja necessário o envolvimento de usuários finais (Alves, 2020). O termo VADER representa cinco pilares fundamentais na avaliação da funcionalidade, segurança e desempenho de uma API, conforme descritas no Quadro 4.

Quadro 4 - Pilares da Heurística VADER

Pilar	Descrição
<i>Verbs</i> (Verbos)	avalia o uso correto dos métodos HTTP, como GET, POST, PUT e DELETE, verificando se cada verbo está associado à operação adequada conforme os princípios REST.
<i>Authorization</i> (Autorização)	examina os mecanismos de autenticação e autorização empregados, como <i>Basic</i> e <i>Bearer tokens</i> , garantindo o controle de acesso seguro aos recursos.
<i>Data</i> (Dados)	analisa os dados trafegados na API, considerando tipagem, formatação (geralmente em JSON), paginação e tamanho das

	respostas, assegurando consistência e eficiência na comunicação.
<i>Errors</i> (Erros)	verifica a padronização dos códigos e mensagens de erro HTTP, como 400, 403, 404, 500 e 503, avaliando se corresponde corretamente às situações encontradas durante a execução.
<i>Responsiveness</i> (Capacidade de resposta)	mede o desempenho e estabilidade da API, considerando tempo de resposta, uso de recursos, capacidade de concorrência e a aplicação do princípio <i>fail fast</i> <sup>1</sup> .

A aplicação dessa heurística fornece uma visão sistemática do comportamento da API, permitindo identificar falhas e oportunidades de melhoria de forma estruturada e replicável.

### 3. ESTADO DA ARTE

O acesso à área da saúde por meio de plataformas digitais já se consolidou como uma prática comum entre os consumidores. De acordo com a pesquisa *The New Era of Consumer Engagement: Insights from Rock Health's Ninth Annual Consumer Adoption Survey*, conduzida pela Rock Health com 8.014 adultos, 76% dos entrevistados relataram já ter utilizado algum tipo de atendimento virtual (ROCK HEALTH, 2024).

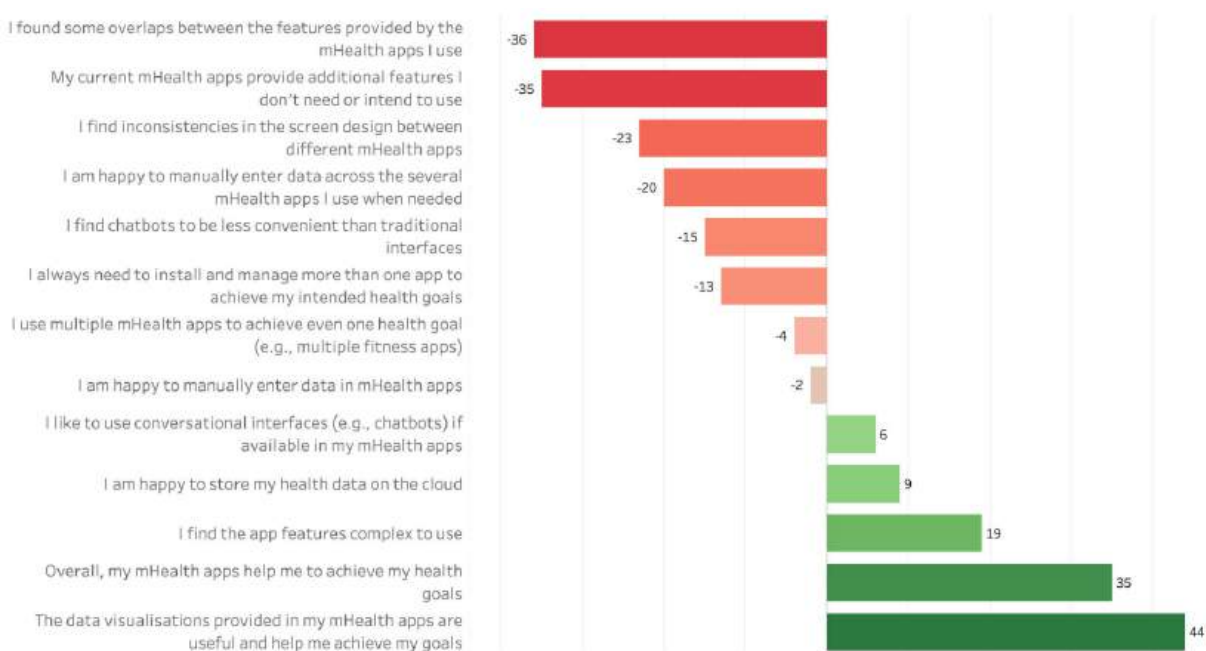
A mesma pesquisa conclui que os usuários estão considerando diversos critérios na hora de escolher aplicativos de cuidados com a saúde, não se limitando mais a um único canal. Os consumidores priorizam fatores como conveniência, facilidade de uso e segurança dos dados. Diante disso, destaca-se a recomendação de que os inovadores do setor invistam no aprimoramento de ofertas de atendimento omnicanal (ROCK HEALTH, 2024).

A fragmentação do ecossistema de aplicativos de saúde esportiva é corroborada por pesquisas recentes, nas quais a maioria dos respondentes (78,6%) prefere uma plataforma unificada em detrimento de múltiplos aplicativos especializados (Philip et al., 2023). Como ilustrado na Figura 1, os usuários enfrentam desafios significativos, como sobreposição de funcionalidades, necessidade de instalar múltiplos aplicativos e entrada manual de dados em diferentes plataformas.

---

<sup>1</sup>Fail-fast refere-se à prática de projetar sistemas que detectam e relatam falhas o mais cedo possível, interrompendo a execução ao identificar condições inválidas ou erros, evitando que problemas se propaguem e causem falhas mais complexas no sistema.

Figura 1 - Principais desafios relatados por usuários de aplicativos de saúde móvel (mHealth)



Fonte: Philip et al. (2023, p. 19885)

Para investigar de forma estruturada as lacunas de integração e colaboração neste ecossistema, foi conduzido um Mapeamento Sistemático da Literatura (MS), cujo protocolo detalhado encontra-se no Apêndice A. A análise dos resultados desse mapeamento, detalhada a seguir, revela com clareza a magnitude da fragmentação tanto nas funcionalidades oferecidas quanto na integração entre os atores do ecossistema.

### 3.1 TRABALHOS RELACIONADOS

A análise dos artigos referenciados no Quadro 11.A revelou uma ausência significativa de interoperabilidade entre as aplicações atualmente disponíveis no mercado, evidenciando uma oportunidade para o desenvolvimento de soluções mais integradas e eficientes. Algumas aplicações, no entanto, se destacam por promover uma maior colaboração entre os usuários.

Dentre elas, merecem destaque as seguintes: o aplicativo “Fitness&Health”, por exemplo, surge como uma solução diferenciada ao integrar de forma eficiente os serviços de nutrição e exercícios, conectando o usuário final a personal trainers de diferentes regiões do Brasil.

Essa integração visa preencher a lacuna existente em aplicativos que, comumente, abordam esses aspectos de maneira isolada. Já o “Diet Engine” propõe uma abordagem inovadora, indo além dos tradicionais registros manuais. Ele incorpora tecnologias como

visão computacional para classificar alimentos por meio de imagens, além de aplicar o processamento de linguagem natural (PLN) para realizar análises nutricionais dinâmicas.

Outro destaque é seu assistente inteligente, um *chatbot* que fornece recomendações personalizadas de dietas e exercícios, considerando restrições alimentares, metas de saúde e histórico do usuário.

Outra aplicação relevante é o “Aplicativo Fitness”, cujo principal objetivo é incentivar uma rotina mais saudável. Ele oferece funcionalidades como o rastreamento de saúde, que permite capturar e exibir dados como batimentos cardíacos, monitoramento de sono e massa corporal. Também permite o monitoramento de refeições, com cálculo da massa corporal ao fim do dia, além de sugestões de dieta baseadas nas escolhas alimentares registradas pelo usuário.

Por fim, o “SugarControl” se apresenta como uma ferramenta robusta voltada para pessoas com diabetes tipo 2, focado em combater os efeitos do sedentarismo e das más escolhas alimentares.

Entre suas funcionalidades, destacam-se o monitoramento em tempo real de sinais vitais, como níveis de glicose no sangue, pressão arterial, frequência cardíaca e saturação de oxigênio, utilizando dispositivos médicos digitais.

O aplicativo também realiza o acompanhamento da dieta e nutrição com categorias visuais, atribuindo pontuações a escolhas saudáveis e não saudáveis, além de registrar atividades físicas, definir metas personalizadas e emitir lembretes. Adicionalmente, o “SugarControl” oferece recursos voltados ao bem-estar mental, como meditação guiada, *yoga* terapêutica e mantras espirituais, com relatórios de estado emocional antes e depois das práticas.

Um painel analítico personalizado permite o acompanhamento de progresso em diversas áreas, enquanto mecanismos de *feedback* e gamificação incentivam o engajamento dos usuários. Por fim, o aplicativo coleta dados demográficos como idade, gênero e histórico de diabetes para personalizar ainda mais suas recomendações.

### 3.2 RESULTADOS E DISCUSSÃO DO MAPEAMENTO

O número total de artigos analisados inicialmente foi de 166. Após a aplicação dos Critérios de inclusão (CI) e Critérios de exclusão (CE) citados no Apêndice A, 35 artigos foram identificados como potencialmente relevantes, dos quais 11 foram selecionados para compor a versão final deste trabalho, por serem considerados adequados para responder às questões de pesquisa.

A distribuição desses artigos selecionados, bem como os resultados da triagem nas diferentes bases consultadas, está apresentada no Quadro 5, que detalha o número de artigos analisados, os identificados como possivelmente relacionados e os efetivamente incluídos, de acordo com cada serviço de busca utilizado.

Quadro 5 - Resultados da Triagem de Artigos por Base de Dados

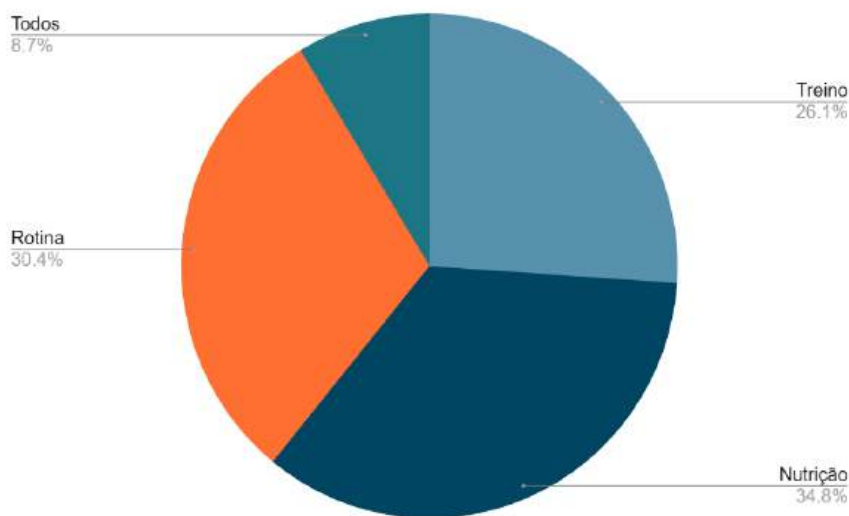
Serviço de busca	Analisados	Possivelmente relacionados	Incluídos
ACM Digital Library	57	6	4
Google Acadêmico	27	10	6
ScienceDirect	31	12	1
Portal de Periódicos da CAPES	51	7	0

Fonte: Elaborado pelo autor, 2025.

As Figuras 2 e 3 foram elaboradas a partir da análise dos 11 artigos selecionados para fornecer uma compreensão quantitativa das lacunas existentes nas soluções atuais do mercado, diretamente relacionadas às Questões de Pesquisa presentes no Apêndice A.

Através da análise dessas métricas, torna-se possível identificar com clareza quais funcionalidades são frequentemente implementadas e quais perfis de usuários profissionais são priorizados nas aplicações, como treinadores e nutricionistas.

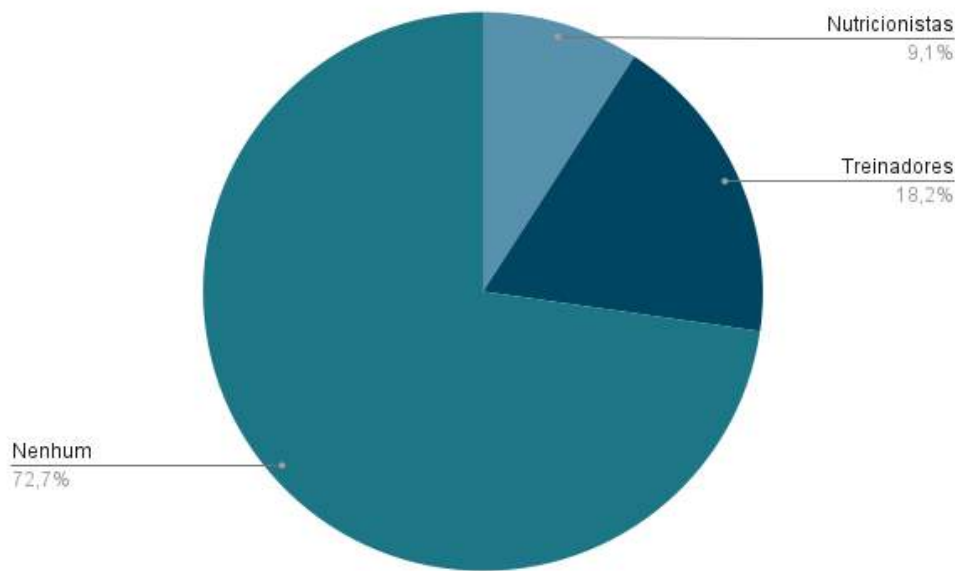
Figura 2 - Distribuição de funcionalidades em aplicativos de saúde esportiva



Fonte: Elaborado pelo autor, 2025.

A Figura 2 demonstra que 26,1% dos aplicativos analisados integram funcionalidades de treino, 34,8% integram funcionalidades de nutrição e 30,4% integram funcionalidades de gerenciamento de rotina. Apenas 8,7% dos aplicativos integram todas as funcionalidades (treino, nutrição e rotina), evidenciando uma fragmentação do mercado.

Figura 3 - Proporção de aplicativos por categoria profissional integrada



Fonte: Elaborado pelo autor, 2025.

Complementando esta análise, a Figura 3 revela a distribuição dos profissionais integrados nestas aplicações, mostrando que 18,2% dos aplicativos incluem treinadores, 9,1% envolvem nutricionistas, 72,7% não oferecem integração nenhuma entre os atuantes, e nenhum oferece um ambiente colaborativo completo que une ambos os profissionais com os alunos e/ou pacientes.

Esta análise conjunta demonstra que a fragmentação é dupla: tanto nas funcionalidades oferecidas quanto na integração entre os diferentes atores do ecossistema de saúde esportiva. A disparidade na inclusão de profissionais, com predominância de treinadores em detrimento de nutricionistas, reforça a segmentação do mercado, onde as soluções tendem a focar em apenas uma categoria profissional.

Os gráficos evidenciam a magnitude do problema, respondendo diretamente às QPs 1 e 2: a integração total é rara (QP1) e a colaboração plena entre profissionais é inexistente (QP2). Essa dupla fragmentação responde também à QP3, identificando como principal limitação das soluções atuais a incapacidade de fornecer um ecossistema coeso e verdadeiramente colaborativo. Diante desse cenário, a QP4 pode ser respondida pela

necessidade crítica de se desenvolver plataformas unificadas que não apenas agreguem todas as funcionalidades (treino, nutrição e rotina), mas que também integrem, em um único ambiente, a atuação colaborativa de diferentes profissionais (como treinadores e nutricionistas) com os alunos, superando a dependência de múltiplas aplicações externas e a entrada manual e fragmentada de dados.

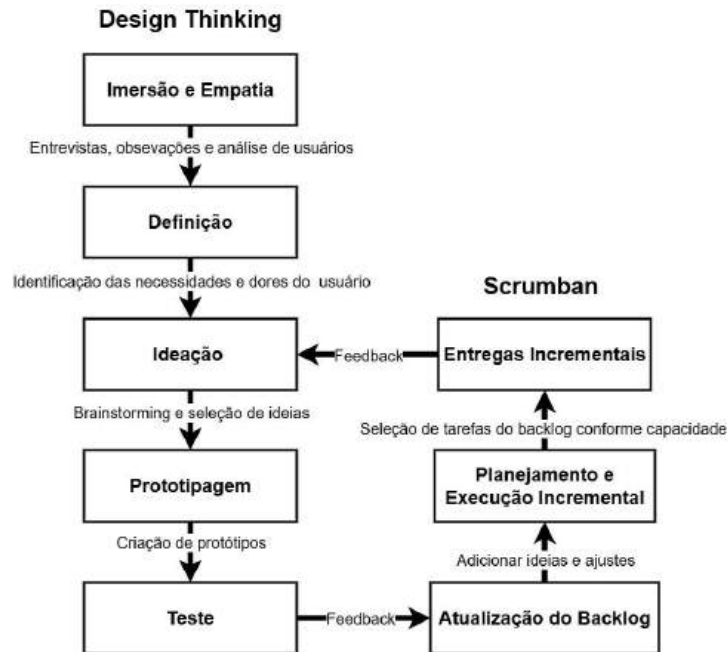
#### **4. METODOLOGIA**

Este capítulo descreve a metodologia utilizada para o desenvolvimento da API *RESTful* proposta, apresentando de forma integrada as abordagens e *frameworks* selecionados para garantir uma construção ágil, centrada no usuário e alinhada aos objetivos do projeto. A metodologia adotada combina princípios do *Design Thinking* e *Scrumban*, criando um fluxo coerente que prioriza a validação contínua e a entrega de valor desde as fases iniciais do projeto, como apresentado na Figura 4.

O fluxo ilustrado demonstra a integração entre as etapas criativas do *Design Thinking* e a execução iterativa do *Scrumban*. As fases de Imersão e Empatia, Definição e Ideação permitem compreender as necessidades dos usuários e propor soluções adequadas. Em seguida, as etapas de Prototipagem e Teste viabilizam a validação das hipóteses e o refinamento contínuo das funcionalidades.

No *Scrumban*, essas informações são incorporadas em entregas incrementais, organizadas por meio de um *backlog* priorizado, garantindo adaptação constante e evolução colaborativa do produto.

Figura 4 - Fluxo da integração das metodologias Design Thinking e Scrumban



Fonte: Elaborado pelo autor, 2025.

#### 4.1 DESIGN THINKING

Para garantir que a API atendesse efetivamente às demandas reais dos usuários, adotou-se a abordagem do *Design Thinking*.

A aplicação do *Design Thinking* neste projeto justifica-se pela necessidade de assegurar que a API possua utilidade real na interação entre nutricionistas, treinadores e alunos de academias. O processo foi conduzido em etapas conforme descrito no Quadro 6.

Quadro 6 - Etapas do Design Thinking Aplicadas no Projeto

Etapa	Descrição
Imersão e Empatia	Realizou-se a compreensão do contexto do problema por meio de informações extraídas de artigos acadêmicos, além da análise das soluções disponíveis no mercado. Essa etapa permitiu identificar as principais dores, expectativas e oportunidades de melhoria.
Definição	As informações coletadas foram sintetizadas para definir claramente o problema central a ser solucionado. A partir disso, foram identificados requisitos funcionais, como a integração entre planos de treino e dieta

	e a autenticação segura, e requisitos não funcionais, como desempenho e escalabilidade da aplicação.
Ideação	Foram geradas diversas ideias e alternativas de solução para atender às necessidades levantadas, utilizando técnicas como <i>brainstorming</i> e análise de viabilidade. Essa etapa possibilitou explorar diferentes formas de estruturar o ambiente para integrar as funcionalidades.
Prototipagem	Desenvolveu-se um protótipo no Figma, representando visualmente as interações esperadas entre os usuários e os serviços oferecidos pela API, possibilitando antecipar ajustes antes da implementação, como apresentado no Apêndice B.
Teste	Foram realizados testes de integração no Postman, com o objetivo de validar os fluxos e refinar as rotas e respostas da API.

Fonte: Elaborado pelo autor, 2025.

Essa abordagem garantiu que o desenvolvimento da API fosse orientado pela real necessidade dos usuários, reduzindo o risco de criar uma solução desconectada das reais necessidades do mercado.

## 4.2 SCRUMBAN

O *Scrumban* foi selecionado com o objetivo de preencher as lacunas deixadas em aberto pelo Scrum e Kanban ao unir funcionalidades dos dois métodos, flexibilidade a alterações dos requisitos de produção e a visualização do fluxo de trabalho do Kanban (Reis; Augusto Albuquerque, 2021). A metodologia foi aplicada da seguinte forma:

- Fluxo Contínuo com Incrementos: O projeto adotou um fluxo contínuo inspirado no Kanban, onde as tarefas eram puxadas do *backlog* conforme a capacidade da equipe e priorização. O *backlog* foi constantemente refinado, e cada entrega parcial da API permitiu validação antecipada das funcionalidades implementadas.
- Quadro Kanban: Foi utilizado um quadro digital na ferramenta Trello, que possibilitou a visualização clara do progresso das tarefas em colunas e o ajuste dinâmico das demandas conforme surgiam mudanças ou novas prioridades, conforme apresentado na Figura 11.C.

## 5 RESULTADOS E DISCUSSÕES

Esta seção apresenta e analisa os resultados obtidos no desenvolvimento da API, estabelecendo relações diretas com os objetivos propostos. Os dados coletados durante a implementação e testes são organizados para demonstrar o alcance dos objetivos de interoperabilidade, segurança, desempenho e integração que fundamentaram este trabalho. A análise destes resultados não apenas valida a abordagem técnica adotada, mas também fornece percepções mensuráveis sobre a viabilidade de soluções integradas para superar a lacuna encontrada no mercado das aplicações de saúde no esporte.

### 5.1 CRIAÇÃO DA API RESTFUL

A apresentação dos resultados organiza-se em três dimensões principais: quantificação do desenvolvimento da API, análise de desempenho e caracterização da arquitetura modular implementada. Esta estrutura permite uma compreensão abrangente do escopo, complexidade e eficiência da solução desenvolvida, estabelecendo bases sólidas para a discussão subsequente sobre o significado e implicações destes resultados no contexto dos desafios identificados na revisão da literatura.

Além da implementação das 101 rotas da API, foram desenvolvidas interfaces intuitivas e específicas para cada perfil de usuário (treinador, nutricionista e aluno), as quais estão detalhadas no Apêndice D. Essas interfaces refletem a arquitetura modular da aplicação e permitem a interação unificada entre os profissionais e seus alunos/pacientes.

#### 5.1.1 Quantificação do Desenvolvimento da API

A quantificação do desenvolvimento revela a escala e complexidade da API implementada, fornecendo métricas objetivas e aderência aos princípios REST. O Quadro 7 sintetiza a distribuição das 101 rotas desenvolvidas, categorizadas por método HTTP e respectivos tempos médios de resposta, oferecendo uma visão geral da arquitetura do sistema.

Quadro 7 - Distribuição de Rotas por Método HTTP e Tempo Médio de Resposta

Método HTTP	Quantidade	Tempo médio de resposta
GET	41	495,39
POST	33	950,42

DELETE	14	466,5
PUT	13	525,07

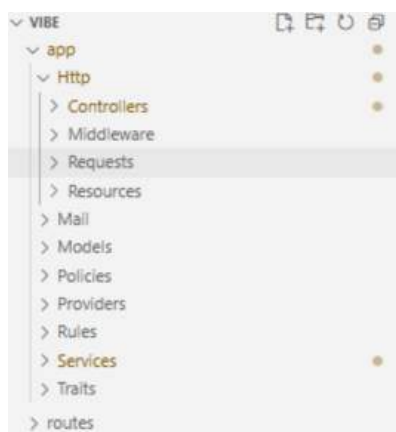
Fonte: Elaborado pelo autor, 2025.

A arquitetura implementada supera a lacuna de integração identificada no estado da arte, onde apenas 8,7% das aplicações analisadas integravam completamente as três funcionalidades principais (treino, nutrição e rotina). Além disso, a API implementa um modelo de colaboração, permitindo que treinadores e nutricionistas atuem no mesmo ambiente.

### 5.1.2 Arquitetura e Implementação Técnica

A aplicação foi estruturada baseada no padrão *Model-View-Controller* (MVC), garantindo separação de responsabilidades, manutenibilidade e escalabilidade. A camada de modelos representa as entidades do domínio, como *User*, *Training* e *Diet*, permitindo validações e relacionamentos com outras tabelas representadas no Apêndice E. A camada de controladores gerencia a lógica da aplicação, processa requisições HTTP, orquestra operações entre modelos e retorna respostas, seguindo princípios REST. Para uniformizar as respostas, foram utilizados recursos e coleções, incluindo metadados de paginação. A Figura 5 mostra a organização modular da aplicação, com separação clara entre modelos, controladores e recursos.

Figura 5 - Estrutura de Pastas da Aplicação



Fonte: Elaborado pelo autor, 2025.

Os métodos correspondentes aos verbos HTTP são acionados dentro dos controladores para executar tarefas específicas conforme a demanda do usuário. Para requisições POST, a

função *store* é responsável por inserir novas informações na base de dados. Em chamadas GET, utilizam-se as funções *index* e *show*, destinadas, respectivamente, à listagem completa dos dados de um recurso e à obtenção de informações detalhadas. Para o método PUT, a função *update* realiza a atualização dos dados armazenados. A utilização desses quatro métodos exemplifica a implementação do padrão RESTful, que se baseia no uso consistente e semântico dos verbos HTTP.

Esse padrão é aplicado de forma uniforme em toda a API, sendo adaptado para os diferentes modelos de domínio, como alimentos, receitas, refeições e dietas (no contexto do nutricionista), além de exercícios, séries, treinos e programas de treino (no contexto do treinador).

Para garantir a consistência das respostas, foram utilizados *resources* e *collections* específicos para cada modelo, retornando, juntamente com os dados solicitados, informações de paginação, como quantidade total de itens, número de páginas e links de navegação. Quanto à consistência dos dados, sempre que uma informação é criada, atualizada ou excluída por meio de controladores que utilizam cache, um método de invalidação é acionado para remover o cache correspondente. Essa prática assegura que as respostas fornecidas ao usuário reflitam o estado atual do banco de dados, evitando o retorno de informações desatualizadas armazenadas em cache.

Além disso, a autenticação e o controle de acesso foram implementados por meio de um middleware personalizado que valida tokens JWT em todas as requisições que exigem autenticação, conforme ilustrado na Figura 6.

Figura 6 - Middleware Gerenciador de Autenticação

```
class JwtAuthenticate
{
  0 references | 0 overrides
  public function handle(Request $request, Closure $next): Response
  {
    try {
      $user = JWTAuth::parseToken()->authenticate();
      if (!$user) {
        return response()->json(data: [
          'success' => false,
          'error' => 'User not found.'
        ], status: 404);
      }
    } catch (\Exception $e) {
      return response()->json(data: [
        'success' => false,
        'error' => 'Not authorized'
      ], status: 401);
    }

    return $next($request);
  }
}
```

Fonte: Elaborado pelo autor, 2025.

### 5.1.3 Análise de Desempenho e Eficiência

Os testes de desempenho realizados com o Postman revelaram um tempo médio de resposta de 577 ms para o conjunto total de rotas, que é considerado bom, pois está dentro da margem de 0.1 a 1 segundo de tempo médio de resposta, que não prejudica a interação do usuário (Ben, 2024). A análise detalhada por categoria de operação demonstra padrões consistentes com a complexidade inerente a cada tipo de requisição:

- Os métodos PUT, DELETE e GET apresentaram melhor desempenho.
- As operações POST mostraram maior latência nas rotas de autenticação, refletindo o tempo de resposta inerente a utilização do serviço gratuito de *Simple Mail Transfer Protocol* (SMTP) em ambiente de testes.

A Figura 18.F apresenta um gráfico de dispersão, que permite identificar o tempo médio de resposta das requisições, separado por método HTTP, também, O Quadro 12.F demonstra as três rotas com maior tempo de resposta na aplicação e seus respectivos valores. O gráfico em formato de pizza, apresentado na Figura 16.G, ilustra a distribuição percentual dos métodos HTTP, confirmando a presença de todas as operações *Create, Read, Update e Delete* (CRUD) entre as funcionalidades da aplicação.

#### 5.1.4 Aderência aos Princípios REST

A API desenvolvida segue os princípios REST, validados por meio da heurística VADER, que considera cinco dimensões: *Verbs*, *Authorization*, *Data*, *Errors* e *Responsiveness*, a seguir, como cada pilar foi seguido:

Quadro 8 - Aderência da API à Heurística VADER

Pilar	Como foi seguido
<i>Verbs</i> (Verbos)	Os recursos da API utilizam corretamente os verbos HTTP. O mapeamento CRUD foi aplicado nas rotas, como demonstrado na documentação da API (Apêndice H).
<i>Authorization</i> (Autorização)	O controle de acesso é realizado através de <i>Bearer Token</i> no <i>header</i> da requisição, via <i>JSON Web Tokens</i> (JWT), com validação a cada requisição, como representado no Apêndice I. Foram implementadas 7 rotas de autenticação, tratando adequadamente códigos 401 e 403, assegurando segurança e escalabilidade.
<i>Data</i> (Dados)	Os dados trafegados são validados quanto à tipagem, formatados em JSON, paginados em listagens, incluindo quantidade total de dados, página anterior e posterior, e otimizados quanto ao tamanho das respostas, assegurando comunicação eficiente e previsível entre cliente e servidor.
<i>Errors</i> (Erros)	A API retorna códigos HTTP padronizados, como 200, 201, 400, 401, 403, 404 e 500, para indicar o resultado das operações. A documentação interativa detalha respostas de sucesso e erro para todas as 101 rotas, assegurando tratamento semântico de exceções (Apêndice H).
<i>Responsiveness</i> (Capacidade de Resposta)	O tempo médio de resposta é de 577 ms, com redução de 42% em rotas críticas após implementação de <i>cache</i> (Apêndice J). A arquitetura <i>stateless</i> e em camadas garante escalabilidade e manutenção do desempenho sob diferentes cargas.

Fonte: Elaborado pelo autor, 2025.

A aplicação da heurística VADER comprova que a API atende aos princípios REST, apresentando baixo acoplamento, segurança, escalabilidade e simplicidade. A documentação

interativa, demonstrada no Apêndice H, organiza todas as rotas, métodos, parâmetros e exemplos de resposta, facilitando a adoção da API no ecossistema de saúde esportiva.

## 5.2 CONTRIBUIÇÕES DO PROJETO

Esta seção apresenta uma análise dos resultados obtidos no desenvolvimento da API, estabelecendo relações diretas entre as evidências empíricas coletadas e os objetivos propostos inicialmente. A análise não apenas valida o cumprimento dos requisitos, mas também identifica *trade-offs*<sup>2</sup>, limitações e oportunidades de otimização, oferecendo uma visão abrangente sobre a eficácia da solução implementada no contexto da saúde esportiva digital.

Como sintetizado no Quadro 9, os resultados demonstram como as metas técnicas e funcionais se materializaram em artefatos concretos, validando a abordagem escolhida e oferecendo informações práticas sobre o desempenho da solução.

Quadro 9 - Relação entre Objetivos Específicos, Resultados Obtidos e Evidências

Objetivo	Resultado	Evidência
Garantir a utilização da API em ambientes externos	Implementação de rotas padronizadas com documentação pública interativa	Documentação automática gerada via Postman (Apêndice H), contemplando todos os <i>endpoints</i> com exemplos de requisição e resposta
Assegurar a implementação da arquitetura REST	API desenvolvida seguindo integralmente os princípios REST	A aderência aos princípios REST foi validada pela heurística VADER, conforme detalhado na seção 5.1.4
Disponibilizar documentação completa para facilitar o uso da API	Documentação interativa pública com 101 <i>endpoints</i> detalhados	Coleções organizadas no Postman com exemplos de requisições, parâmetros, <i>headers</i> e respostas para todos os métodos (Apêndice H)

---

<sup>2</sup> *Trade-off* é um termo em inglês utilizado para descrever uma situação em que é necessário abrir mão de algo em troca de outro benefício, representando um equilíbrio entre vantagens e desvantagens em uma decisão.

Implementar autenticação segura utilizando JSON <i>Web Token</i> (JWT)	Sistema de autenticação robusto com 7 rotas dedicadas	Implementação de <i>login</i> , registro, recuperação de senha e <i>refresh tokens</i> .
--	---	--

Fonte: Elaborado pelo autor, 2025.

Os resultados evidenciam que todos os objetivos específicos foram plenamente atendidos, apresentando métricas e comprovações que atestam o cumprimento de cada meta estabelecida

Em comparação com as limitações identificadas no mapeamento sistemático, observam-se avanços significativos:

- **Integração completa:** Todos os módulos interconectados pela aplicação e 8,7% dos módulos conectados nas aplicações mapeadas no estado da arte;
- **Colaboração efetiva:** Ambiente unificado para treinadores, nutricionistas e alunos e nenhuma solução similar identificada nos sistemas mapeados na seção do estado da arte;
- **Interoperabilidade:** Arquitetura baseada em padrões abertos que permite integração com outros sistemas.

A solução desenvolvida aborda diretamente o fenômeno do *app overload* ao consolidar em uma única API funcionalidades que normalmente exigiriam múltiplos aplicativos. Enquanto o mercado atual apresenta apenas 8,7% de integração completa entre módulos de treino, nutrição e rotina, este trabalho integra todas essas funcionalidades em um ambiente digital e unificado, gerando benefícios significativos para o ecossistema de aplicativos da saúde no esporte. Entre eles, destacam-se a eliminação da alternância entre aplicativos, que reduz a sobrecarga cognitiva e temporal, a mitigação da necessidade de assinaturas múltiplas com funcionalidades sobrepostas e a garantia de um fluxo contínuo de informações entre treino, dieta e rotina. Além disso, a arquitetura da API serve como base única para diferentes *front-ends*, ampliando seu potencial de substituir aplicativos fragmentados por um ecossistema integrado e eficiente.

O modelo de colaboração implementado também representa uma inovação significativa no ecossistema de saúde esportiva digital, diferenciando-se das soluções existentes que focam predominantemente em usuários individuais. Esta API permite que treinadores criem e atribuam programas de treino específicos, enquanto nutricionistas podem

desenvolver e atribuir planos alimentares personalizados. Por sua vez, os alunos têm a capacidade de acompanhar de forma centralizada todo o seu progresso. A principal inovação, no entanto, reside na facilitação da troca de informações e na genuína colaboração entre os diferentes profissionais envolvidos no cuidado de um mesmo aluno ou paciente, criando assim uma abordagem integrada e sinérgica para o seu desenvolvimento.

Os resultados apresentados na seção evidenciam que a API desenvolvida cumpriu integralmente os objetivos específicos propostos, demonstrando eficácia técnica, aderência aos princípios REST e capacidade de integração entre múltiplos módulos do ecossistema de saúde esportiva. A consolidação de funcionalidades de treino, nutrição e acompanhamento de rotina em uma única plataforma não apenas supera as limitações observadas em soluções existentes, mas também promove inovação ao facilitar a colaboração entre profissionais e alunos, proporcionando um fluxo contínuo de informações e experiências mais coesas. Dessa forma, o projeto se configura como uma contribuição significativa para o avanço de práticas digitais integradas na saúde esportiva, oferecendo bases sólidas para futuras melhorias e expansões do sistema.

## **6 CONSIDERAÇÕES FINAIS**

Os resultados obtidos demonstram a viabilidade técnica e conceitual do projeto, alcançando plenamente os objetivos propostos. A API desenvolvida, composta por 101 rotas distribuídas entre operações CRUD, comprova a capacidade de integração entre as funcionalidades de treino, nutrição e acompanhamento de rotina em um ambiente unificado. O desempenho médio de 577 ms mostrou-se adequado para a maioria dos casos de uso, garantindo respostas rápidas e consistentes.

A principal contribuição deste trabalho reside na criação de um ecossistema colaborativo que promove a interação entre diferentes profissionais da saúde esportiva, como treinadores e nutricionistas e seus alunos ou pacientes, solucionando uma lacuna crítica identificada no mapeamento sistemático da literatura, em que nenhuma das soluções analisadas oferecia integração completa entre todos os atores envolvidos.

Dessa forma, a solução proposta representa um avanço significativo na integração de ecossistemas digitais voltados à saúde esportiva, oferecendo uma base técnica sólida para o desenvolvimento de aplicações que proporcionem experiências mais coesas, eficientes e seguras, contribuindo para a evolução das práticas de cuidado e acompanhamento no contexto esportivo.

## 6.1 LIMITAÇÕES E TRABALHOS FUTUROS

Esta seção tem como objetivo apresentar as principais limitações identificadas durante o desenvolvimento da pesquisa, bem como propor direcionamentos promissores para investigações futuras na área. Entre as limitações observadas, destacam-se os testes de desempenho realizados em ambiente controlado e a quantidade ainda limitada de artigos publicados sobre o tema.

Para trabalhos futuros, sugere-se a expansão das integrações com dispositivos de IoT, permitindo maior conectividade e automação no ecossistema estudado. Além disso, a inclusão de um módulo de substituições alimentares poderá ampliar a personalização e a flexibilidade das recomendações fornecidas. Também se considera relevante a implementação de filas assíncronas, com o objetivo de reduzir o tempo de resposta em rotas que envolvem o envio de e-mails, otimizando o desempenho geral do sistema. Por fim, propõe-se a integração com plataformas que disponibilizam alimentos e suas informações nutricionais, possibilitando maior precisão e praticidade no acesso aos dados essenciais para o funcionamento do modelo desenvolvido.

## REFERÊNCIAS

- ALT, Luis et al. O que é Design Thinking. **Revista Coaching Brasil**, v. 213, p. 14-17, 2017. Disponível em: <https://mamtra.com.br/wp-content/uploads/2017/08/Coaching-e-Thinking-Desing.pdf>. Acesso em: 03 nov. 2025.
- ALVES, Maximiliano. VADER: Heurística para teste de API na prática. **Medium**, [S. ], 28 ago. 2020. Disponível em: <https://maximilianoalves.medium.com/vader-heuristica-para-teste-de-api-na-pratica-fcf78c6ac ec>. Acesso em: 25 out. 2025.
- ARIYANTO, Yuri; RACHMAD, Mochamad Farhan Fitrahtur; PUSPITASARI, Dwi. Laravel framework and native PHP: Comparison in the crea-tion of rest API. **Matrix: Jurnal Manajemen Teknologi Dan Informatika**, v. 14, n. 2, p. 66-73, 2024. Disponível em: <https://ojs2.pnb.ac.id/index.php/MATRIX/article/view/1413>. Acesso em: 25 out. 2025.
- ATLASSIAN. **Scrumban**: mastering two Agile methodologies. c2025. Disponível em: <https://www.atlassian.com/agile/project-management/scrumban>. Acesso em: 25 out. 2025.
- BEN, Farouk. What is a Good API Response Time? **Odown Blog**, 5 jul. 2024. Disponível em: <https://odown.com/blog/what-is-a-good-api-response-time>. Acesso em: 25 out. 2025.
- BOEG, Jesper. Kanban em 10 passos. **Tradução de Leonardo Campos, Marcelo Costa, Lúcio Camilo, Rafael Buzon, Paulo Rebelo, Eric Fer, Ivo La Puma, Leonardo Galvão,**

**Thiago Vespa, Manoel Pimentel e Daniel Wildt. C4Media**, p. 27, 2010. Disponível em: <https://cdn3.gnarususercontent.com.br/2692-cultura-metodos-ageis/InfoQBrasil-Kanban10Pasos.pdf>. Acesso em: 03 nov. 2025.

BIEHL, Matthias. **API architecture**. API-University Press, 2015. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=6D64DwAAQBAJ&oi=fnd&pg=PA15&dq=+BIEHL,+Matthias.+API+architecture.+API-University+Press,+2015>. Acesso em: 03 nov. 2025.

BROWN, Tim et al. Design thinking. **Harvard business review**, v. 86, n. 6, p. 84, 2008. Disponível em: <https://designthinkingmeite.web.unc.edu/wp-content/uploads/sites/22337/2020/02/Tim-Brown-Design-Thinking.pdf>. Acesso em: 03 nov. 2025.

CXAPP. **The very real problem of app overload in the workplace**. Worktech Academy, [S. l.], 2025. Disponível em: <https://worktechacademy.com/the-very-real-problem-of-app-overload-in-the-workplace>. Acesso em: 27 out. 2025.

FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. University of California, Irvine, 2000. Disponível em: <https://www.proquest.com/openview/fc2d064044b971dda476dfb429a2b344/1>. Acesso em: 3 nov. 2025.

**FREELETICS GmbH**. Freeletics: aplicativo para planos de treino e acompanhamento nutricional. 2025. Disponível em: <https://www.freeletics.com>. Acesso em: 2 nov. 2025.

GRAND VIEW RESEARCH. Brazil MHealth Apps Market Size & Outlook, 2024-2030. **Horizon Databook**, 2024. Disponível em: <http://www.grandviewresearch.com/horizon/outlook/mhealth-apps-market/brazil>. Acesso em: 25 maio 2025.

GROWTH SUPPLEMENTS. **Growth Dieta e Treino** [Aplicativo]. [S. l.]: Growth Supplements Produtos Alimentícios LTDA, 2025. Disponível em: <https://www.gsuplementos.com.br/app>. Acesso em: 7 jun. 2025.

IQVIA INSTITUTE. **Digital Health Trends 2024**: Implications for Research and Patient Care [Relatório online]. [S. l.]: IQVIA, 2024. Disponível em: <https://www.iqvia.com/insights/the-iqvia-institute/reports/digital-health-trends-2024>. Acesso em: 3 nov. 2025.

JSON WEB TOKEN. **Introduction to JSON Web Tokens**. [S. l.]: JWT.io, 2025. Disponível em: <https://jwt.io/introduction>. Acesso em: 7 jun. 2025.

KITCHENHAM, Barbara et al. Guidelines for performing systematic literature reviews in software engineering. 2007.

KOVALSKIY, Vlad. **Ditch the App Overload**: Why a Unified Work Ecosystem is the Smarter Choice. Bitrix24, 14 abr. 2025. Disponível em:

<https://www.bitrix24.com/articles/ditch-the-app-overload-why-a-unified-work-ecosystem-is-the-smarter-choice.php>. Acesso em: 27 out. 2025.

LADAS, Corey. **Scrumban-essays on kanban systems for lean software development**. Lulu. com, 2009. Disponível em: [https://books.google.com.br/books?id=SQFdAgAAQBAJ&redir\\_esc=y](https://books.google.com.br/books?id=SQFdAgAAQBAJ&redir_esc=y). Acesso em: 03 nov. 2025.

LARAVEL. **Laravel Documentation**. [S. l.]: Laravel, c2025. Disponível em: <https://laravel.com/docs>. Acesso em: 7 jun. 2025.

MYFITNESSPAL, Inc. **MyFitnessPal**: aplicativo para acompanhamento alimentar. 2025. Disponível em: <https://www.myfitnesspal.com>. Acesso em: 2 nov. 2025.

MURTY, Rohan Narayana; DADLANI, Sandeep; DAS, Rajath B. **How much time and energy do we waste toggling between applications?** Harvard Business Review, 29 Aug. 2022. Disponível em: <https://hbr.org/2022/08/how-much-time-and-energy-do-we-waste-toggling-between-applications>. Acesso em: 27 out. 2025.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para gerenciar projetos de forma ágil. **Mundo PM**, v. 1, n. 14, p. 64-71, 2007. Disponível em: [https://pos.faculdadeprojecao.nucleoad.net/pluginfile.php/2123/mod\\_resource/content/36/Entendendo\\_Scrum\\_para\\_Gerenciar\\_Projetos.pdf](https://pos.faculdadeprojecao.nucleoad.net/pluginfile.php/2123/mod_resource/content/36/Entendendo_Scrum_para_Gerenciar_Projetos.pdf). Acesso em: 03 nov. 2025.

PHILIP, Ben Joseph et al. Toward a unified mhealth platform: A survey of current user challenges and expectations. **IEEE Access**, v. 11, p. 19876-19891, 2023. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10054378>. Acesso em: 03 nov. 2025.

POSTGRESQL. **PostgreSQL Documentation**. [S. l.]: PostgreSQL, c2025. Disponível em: <https://www.postgresql.org/about>. Acesso em: 7 jun. 2025.

POSTMAN, Inc. **Postman Learning Center**. 2025. Disponível em: <https://learning.postman.com>. Acesso em: 7 jun. 2025.

RED HAT. **O que é uma API (Interface de Programação de Aplicações)?** 19 jan. 2023. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em: 7 jun. 2025.

REIS, Augusto Albuquerque. **Scrumban-metodologia híbrida com scrum e kanban para desenvolvimento de software**. 2021. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/8502>. Acesso em: 3 nov. 2025.

ROCK HEALTH. **The New Era of Consumer Engagement**: Insights from Rock Health's Ninth Annual Consumer Adoption Survey. Rock Health, 2024. Disponível em: <https://rockhealth.com/insights/the-new-era-of-consumer-engagement-insights-from-rock-healths-ninth-annual-consumer-adoption-survey>. Acesso em: 3 maio 2025.

SACHDEVA, Sakshi. Scrum methodology. **Int. J. Eng. Comput. Sci**, v. 5, n. 16792, p. 16792-16800, 2016. Disponível em: <https://www.ijecs.in/index.php/ijecs/article/view/1989>. Acesso em: 3 nov. 2025.

STRAVA, Inc. **Strava**: aplicativo para monitoramento de corridas e ciclismo. 2025. Disponível em: <https://www.strava.com>. Acesso em: 2 nov. 2025.

VAN VELSEN, Lex; BEAUJEAN, Desirée JMA; VAN GEMERT-PIJNEN, Julia EWC. Why mobile health app overload drives us crazy, and how to restore the sanity. **BMC medical informatics and decision making**, v. 13, n. 1, p. 23, 2013. Disponível em: <https://link.springer.com/article/10.1186/1472-6947-13-23>. Acesso em: 3 nov. 2025.

WORLD HEALTH ORGANIZATION (WHO). **Obesity and overweight**. [Geneva]: WHO, 2025. Disponível em: <https://www.who.int/health-topics/obesity>. Acesso em: 7 jun. 2025.

## Apêndice A - Protocolo do mapeamento sistemático

Este apêndice utiliza uma abordagem metodológica baseada em diretrizes para revisões sistemáticas da literatura (Kitchenham et al., 2007), com o objetivo de identificar, analisar e sintetizar as evidências disponíveis sobre aplicações digitais voltadas à saúde esportiva.

O objetivo principal do mapeamento é compreender o estado da arte e identificar lacunas em soluções existentes que integrem funcionalidades de treino, nutrição e monitoramento de rotina, além de promover a colaboração entre diferentes perfis de usuários, como nutricionistas, treinadores e alunos de forma totalmente independente de aplicações externas para tal.

A realização deste mapeamento justifica-se pela necessidade de embasamento teórico e prático para o desenvolvimento da API *RESTful* proposta, que se apresenta como uma resposta às deficiências encontradas em soluções atuais. Assim, a metodologia adotada permite validar a originalidade da proposta e reforça sua relevância frente ao mercado de tecnologias voltadas ao bem-estar e à saúde do esportista.

Ao longo deste apêndice, são descritas a metodologia de pesquisa adotada, os critérios de inclusão e exclusão utilizados, as bases de dados consultadas, os principais resultados obtidos e uma discussão crítica a partir da análise de artigos selecionados.

O MS foi conduzido para responder às seguintes questões de pesquisa (QPs):

- QP1: Quais aplicações integram funcionalidades de treino, nutrição e monitoramento da rotina?
- QP2: Quais sistemas integram nutricionistas, treinadores e alunos de academia no mesmo ambiente?
- QP3: Quais são as principais limitações percebidas nas soluções existentes segundo a literatura?
- QP4: Quais funcionalidades poderiam ser agregadas às soluções atuais para torná-las mais completas?

Coordenou-se uma pesquisa em diferentes bases com o propósito de encontrar artigos relacionados ao tema, buscando extrair o maior número possível de informações relevantes para a construção deste estudo. A estratégia adotada baseou-se no uso de operadores booleanos, permitindo a combinação de termos em inglês e português relacionados à temática central.

As buscas foram realizadas em fontes acadêmicas amplamente reconhecidas pela comunidade científica, incluindo o Google Acadêmico, a ACM Digital Library, a

ScienceDirect e o Portal de Periódicos da CAPES. A seleção dessas bases visou garantir a diversidade e a confiabilidade dos resultados, contemplando tanto estudos voltados ao desenvolvimento tecnológico quanto à aplicação prática de soluções na área da saúde esportiva digital. Essa abordagem multidisciplinar proporcionou uma visão abrangente do estado atual das tecnologias e sistemas utilizados, servindo como base para a análise crítica apresentada nas seções subsequentes.

Para garantir a abrangência e a relevância dos artigos encontrados, foram definidas expressões de busca (*strings*) elaboradas com base nos principais termos relacionados ao tema deste estudo. Essas *strings* combinam palavras-chave em inglês e português, explorando variações que englobam desde termos técnicos, como “API” e “interoperabilidade”, até conceitos específicos da área fitness, como “dieta”, “treino” e “fitness”.

A aplicação ocorreu nas bases de dados selecionadas, permitindo filtrar os resultados e focar nos trabalhos mais pertinentes para o escopo da pesquisa. No Quadro 10, apresenta-se as expressões utilizadas no processo de busca.

Quadro 10 - Strings de busca

String	Expressão
1	("mHealth") OR (("API" OR "interoperability" OR "web" OR "app") AND ("diet" OR "training" OR "fitness"))
2	("API" OR "web" OR "aplicativo") AND ("dieta" OR "treino" OR "fitness")

Fonte: Elaborado pelo autor, 2025.

Para assegurar que a análise contemplasse apenas trabalhos relevantes e alinhados ao objetivo da pesquisa, foram definidos critérios rigorosos de inclusão e exclusão. Esses critérios orientaram a seleção dos artigos que compõem a amostra final do MS, garantindo que os estudos considerados apresentassem contribuições específicas à área de saúde do esportista e ao desenvolvimento de sistemas digitais relacionados.

Os critérios de inclusão estabeleceram quais características os artigos deveriam conter para serem considerados adequados, como o foco em sistemas voltados à saúde esportiva, a disponibilidade das aplicações no mercado e a publicação dentro do período temporal delimitado (2019-2024). Por outro lado, os critérios de exclusão foram aplicados para eliminar duplicidades ou resultados irrelevantes .

Esse processo criterioso de seleção é fundamental para manter a qualidade e a pertinência da análise, garantindo que as conclusões do estudo sejam baseadas em evidências sólidas e representativas do estado da arte no campo investigado.

Critérios de Inclusão (CI):

- CI1: Artigos focados na construção de sistemas de saúde do esporte.
- CI2: Estudos sobre aplicações disponíveis no mercado voltadas à saúde do esporte.
- CI3: Publicações entre 2019-2024;

Critérios de Exclusão (CE):

- CE1: Artigos duplicados.
- CE2: Artigos que não mencionam o uso ou desenvolvimento de aplicações em seus resumos.

Após a seleção dos artigos que atenderam aos CI e CE, procedeu-se à extração sistemática dos dados relevantes para a pesquisa. Esta etapa consistiu em coletar informações essenciais sobre cada aplicação estudada, incluindo os serviços integrados que oferecem no contexto da saúde esportiva, como funcionalidades de treino, nutrição e monitoramento.

A extração de dados permite organizar e sintetizar os conteúdos encontrados, facilitando a análise comparativa e a identificação das principais características e lacunas das soluções existentes no mercado. O Quadro 11 exemplifica as aplicações analisadas e os respectivos serviços integrados, evidenciando a diversidade de funcionalidades abordadas, como controle de dieta, integração com dispositivos *IoT*, aconselhamento nutricional, entre outros.

Essa etapa é fundamental para embasar a discussão dos resultados, que buscará responder às questões de pesquisa formuladas e fornecer uma visão crítica do estado atual das tecnologias e práticas na área.

Quadro 11 - Extração de dados

Aplicação	Serviços integrados
Fitness&Health <sup>3</sup>	Controle de dieta, nutrição e conexão com <i>personal trainers</i> .

---

<sup>3</sup>SILVA, Caio Coutinho da; BATISTA, Gabriel. Fitness&Health: site para controle de dieta e nutrição. 2022. Disponível em: <http://ric-cps.eastus2.cloudapp.azure.com/handle/123456789/18184>. Acesso em: 03 nov. 2025.

Diet Engine <sup>4</sup>	Identifica e classifica alimentos e informações nutricionais por imagens e recomenda refeições e exercícios físicos personalizados.
e-Nutri <sup>5</sup>	Aconselhamento nutricional personalizado e avaliação da alimentação com <i>feedback</i> sobre hábitos e progresso.
FOCVS <sup>6</sup>	Organização de treinos, metas e evolução.
NutriCP <sup>7</sup>	Auxílio a nutricionistas na padronização de condutas, registro de ações realizadas e suporte à tomada de decisão para melhorar bem-estar e qualidade de vida de idosos em cuidados paliativos.
Smart List <sup>8</sup>	Sugestões alimentares saudáveis na aquisição de produtos.
Fit Pet <sup>9</sup>	Monitora atividades físicas e hábitos de vida para crianças com sobrepeso, promovendo saúde e bem-estar.
Virtual Fitness Trainer <sup>10</sup>	Monitoramento e <i>feedback</i> sobre treinos e planos personalizados.

<sup>4</sup>SAAD, Asim Moin et al. Diet engine: A real-time food nutrition assistant system for personalized dietary guidance. **Food Chemistry Advances**, v. 7, p. 100978, 2025. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2772753X25000942>. Acesso em: 03 nov. 2025.

<sup>5</sup>ZENUN FRANCO, Rodrigo. Online recommender system for personalized nutrition advice. In: **Proceedings of the Eleventh ACM Conference on Recommender Systems**. 2017. p. 411-415. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3109859.3109862>. Acesso em: 03 nov. 2025.

<sup>6</sup>CAMPANARI, Enzo Firmino et al. FOCVS: aplicativo de treino. 2024. Disponível em: <http://ric-cps.eastus2.cloudapp.azure.com/handle/123456789/31219>. Acesso: 03 nov. 2025.

<sup>7</sup>MOURA, Raquel Bezerra Barbosa de et al. Aplicativo de nutrição para idosos em cuidados paliativos. 2022. Disponível em: <http://plone.ufpb.br/ipe/contents/menu/produtos-do-mestrado-em-gerontologia/aplicativo-de-nutricao-para-idosos-em-cuidados-paliativos.pdf>. Acesso em: 03 nov. 2025.

<sup>8</sup>DE SOUZA MARTINS, Esther; ISAIA, Paloma; SEZINI, Angela Maria. DESENVOLVIMENTO DE UM APLICATIVO DE NUTRIÇÃO PARA DISPOSITIVOS MÓVEIS: SMART LIST. **e-Scientia**, v. 13, n. 2, p. 10-30, 2021. Disponível em: <https://openaccesslegada.emnuvens.com.br/dcbas/article/view/3085>. Acesso em: 03 nov. 2025.

<sup>9</sup>ADAM, Carolina Cittadin. Exergames: Um aplicativo de exercício físico para crianças e pré adolescentes com sobrepeso e obesas. 2021. Disponível em: [https://www.oasisbr.ibict.br/vufind/Record/UEPB\\_2ef5511cb2b41639dca83c60f128cd9a](https://www.oasisbr.ibict.br/vufind/Record/UEPB_2ef5511cb2b41639dca83c60f128cd9a). Acesso em: 03 nov. 2025.

<sup>10</sup>GUPTA, Lakshay; GURBUXANI, Shrey; MADAN, Kapil. Virtual Fitness Trainer using Artificial Intelligence. In: **Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing**. 2024. p. 226-233. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3675888.3676056>. Acesso em: 03 nov. 2025.

Nutrition Garden <sup>11</sup>	Motiva hábitos alimentares saudáveis usando gamificação, recomenda consumo de alimentos específicos para prevenir doenças não transmissíveis e fornece informações nutricionais para educar os usuários.
SugarControl <sup>12</sup>	Monitoramento de glicose, estresse, alimentação e atividades.
Aplicativo Fitness <sup>13</sup>	Monitora dados de saúde e rotina do usuário, como batimentos cardíacos, sono, massa corporal e passos, permite registrar refeições, sugere hábitos saudáveis e incentiva a prática de exercícios físicos.

Fonte: Elaborado pelo autor, 2025.

<sup>11</sup>AIGNER, Christoph et al. Nutrition garden-a gamified mobile app for motivating people to eat specific food to prevent non-communicable diseases. In: **Proceedings of the 2023 7th International Conference on Medical and Health Informatics**. 2023. p. 203-207. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3608298.3608336>. Acesso em: 03 nov. 2025.

<sup>12</sup>KAPUR, Ritu et al. SugarControl: an integrative mobile app for type 2 diabetes management through yoga meditation and lifestyle interventions. In: **Proceedings of the 17th International Conference on Pervasive Technologies Related to Assistive Environments**. 2024. p. 647-653. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3652037.3663926>. Acesso em: 03 nov. 2025.

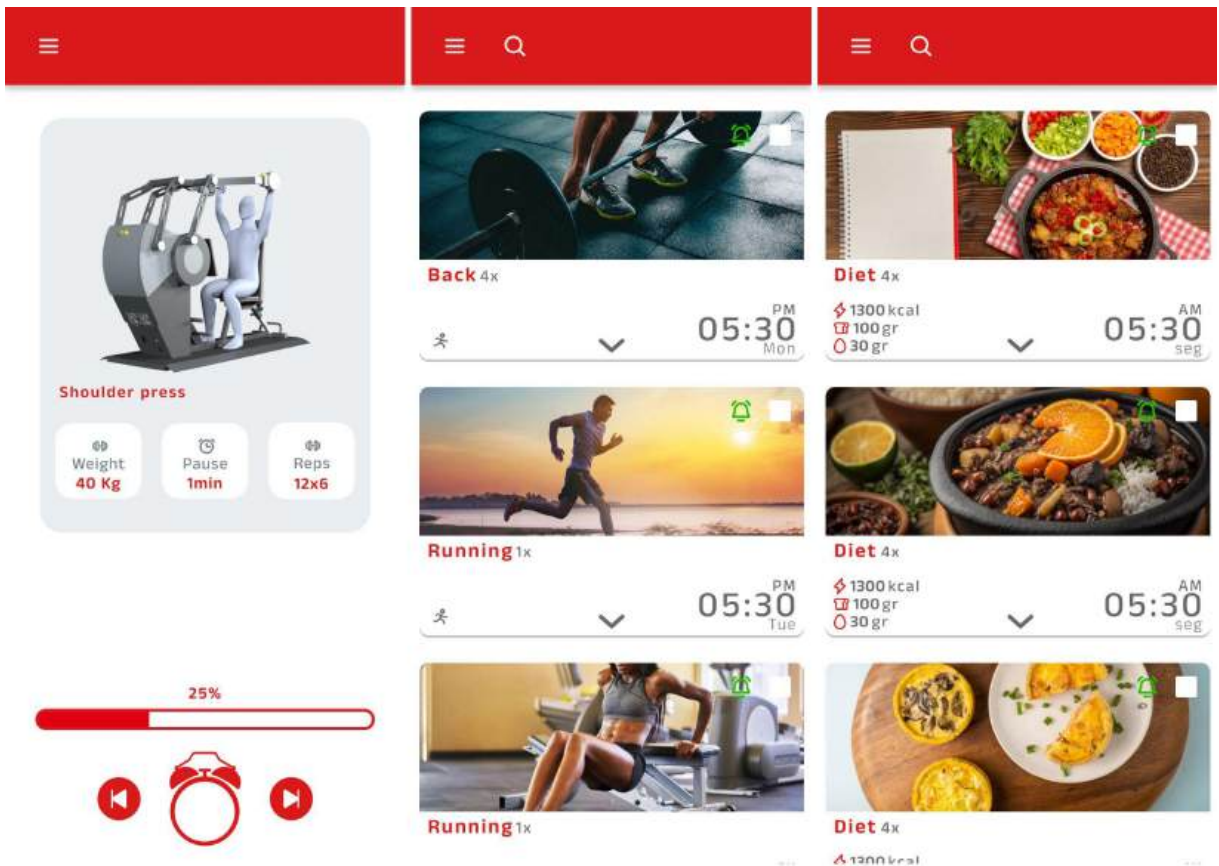
<sup>13</sup>DE ALMEIDA, Arthur Silva; VIEGAS, Silvio Cesar. APLICATIVO FITNESS. **REFAQI-REVISTA DE GESTÃO EDUCAÇÃO EE TECNOLOGIA**, v. 17, n. 2, p. 11-11, 2024. Disponível em: <https://refaqi.faqi.edu.br/index.php/refaqi/article/view/227>. Acesso em: 03 nov. 2025.

## Apêndice B - Protótipo

O protótipo, conforme ilustrado nas Figuras 7, 8, 9 e 10, foi desenvolvido na fase inicial de imersão e ideação do *Design Thinking*. Sua função foi materializar conceitos e fluxos de interação de forma rápida, servindo como material de discussão e validação heurística antes do desenvolvimento do projeto. Esses esboços foram cruciais para obter uma visão macro do funcionamento da aplicação. Embora não tenha sido mantido na versão final, uma grande parte das funcionalidades se mantiveram.

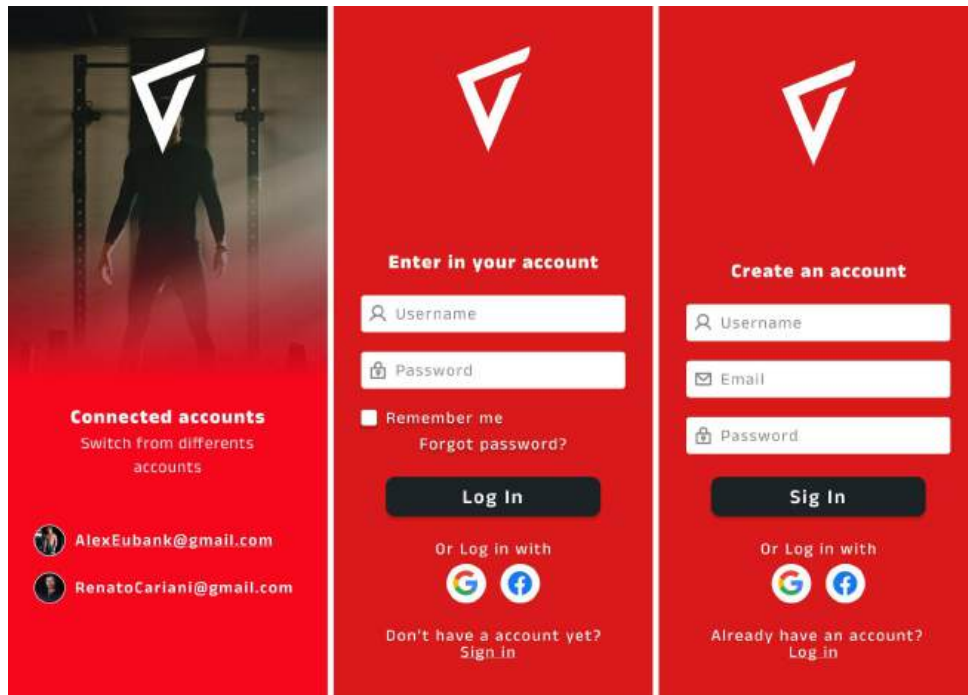
É importante frisar que as informações apresentadas em ambas as interfaces (protótipo e versão final) são meramente ilustrativas; portanto, não devem ser consideradas como dados nutricionais reais, incluindo valores de densidade calórica, quantidade de proteínas e demais informações alimentares.

Figura 7 - Protótipo de interface para perfil de Aluno



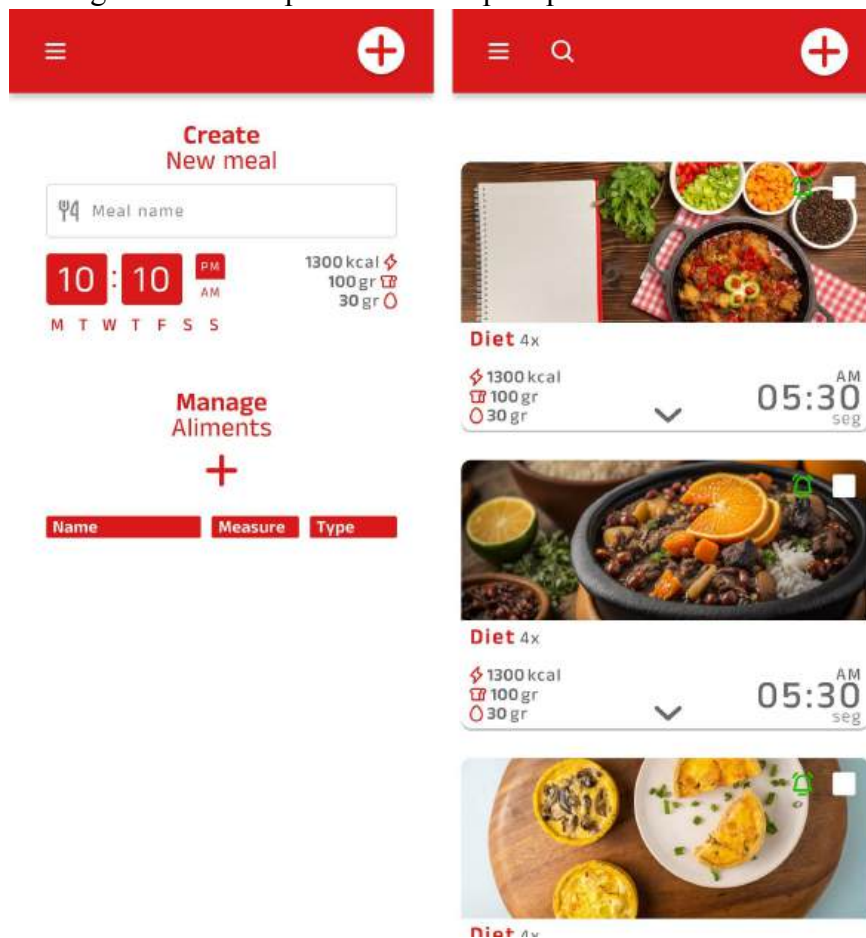
Fonte: Elaborado pelo autor, 2025.

Figura 8 - Protótipo de interface para tela de entrada da aplicação



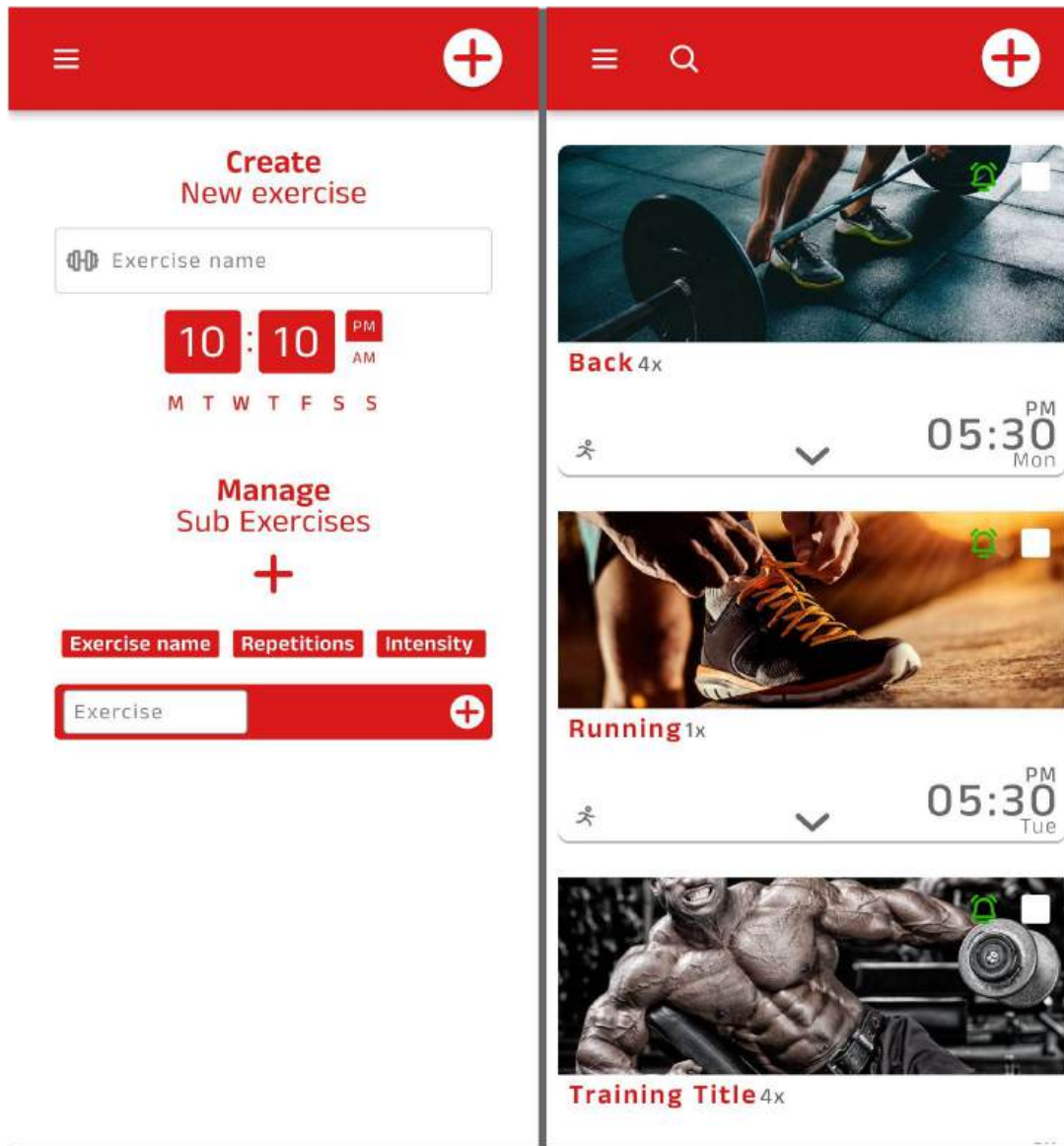
Fonte: Elaborado pelo autor, 2025.

Figura 9 - Protótipo de interface para perfil de nutricionista



Fonte: Elaborado pelo autor, 2025.

Figura 10 - Protótipo de interface para perfil de personal

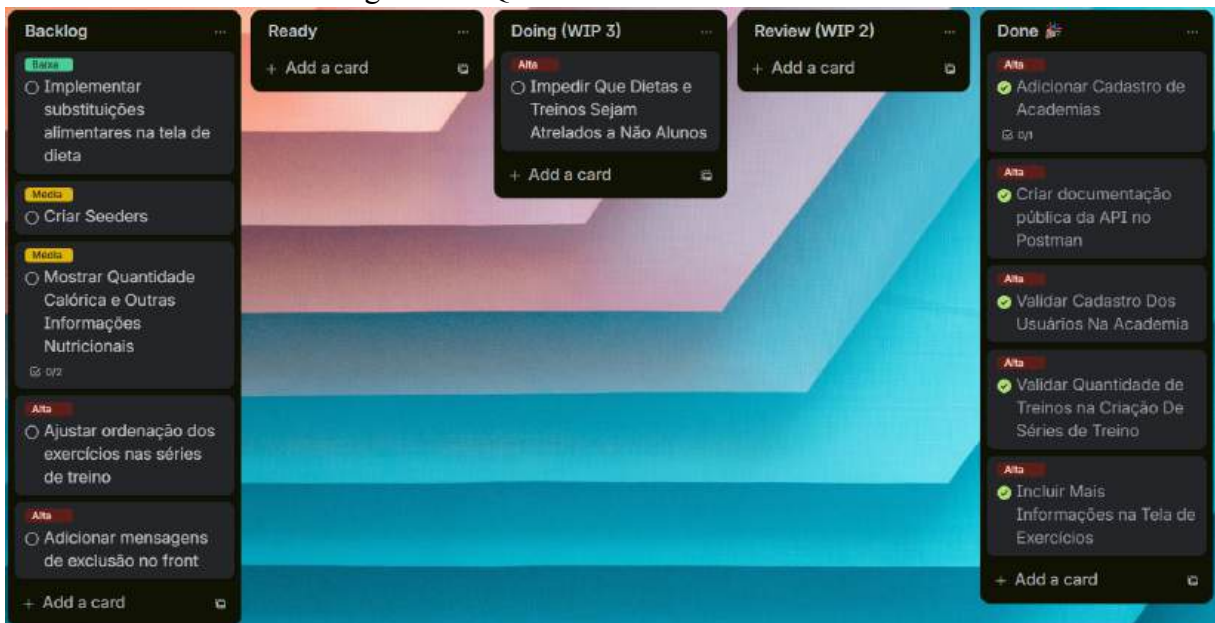


Fonte: Elaborado pelo autor, 2025.

## Apêndice C - Gerenciamento Ágil com Scrumban

O gerenciamento do projeto foi realizado utilizando os princípios do *Scrumban*. O *backlog* do produto, priorizado com base nos objetivos do projeto, foi mantido em um fluxo contínuo de trabalho. O fluxo foi visualizado e gerenciado através de um quadro digital na ferramenta Trello, com as colunas: *Backlog*, *Ready*, *Doing*, *Review* e *Done*. As tarefas eram puxadas naturalmente de acordo com a capacidade da equipe, seguindo uma abordagem de trabalho contínuo sem iterações com prazos fixos, conforme representado na Figura 11.

Figura 11 - Quadro Kanban de Atividades



Fonte: Elaborado pelo autor, 2025.

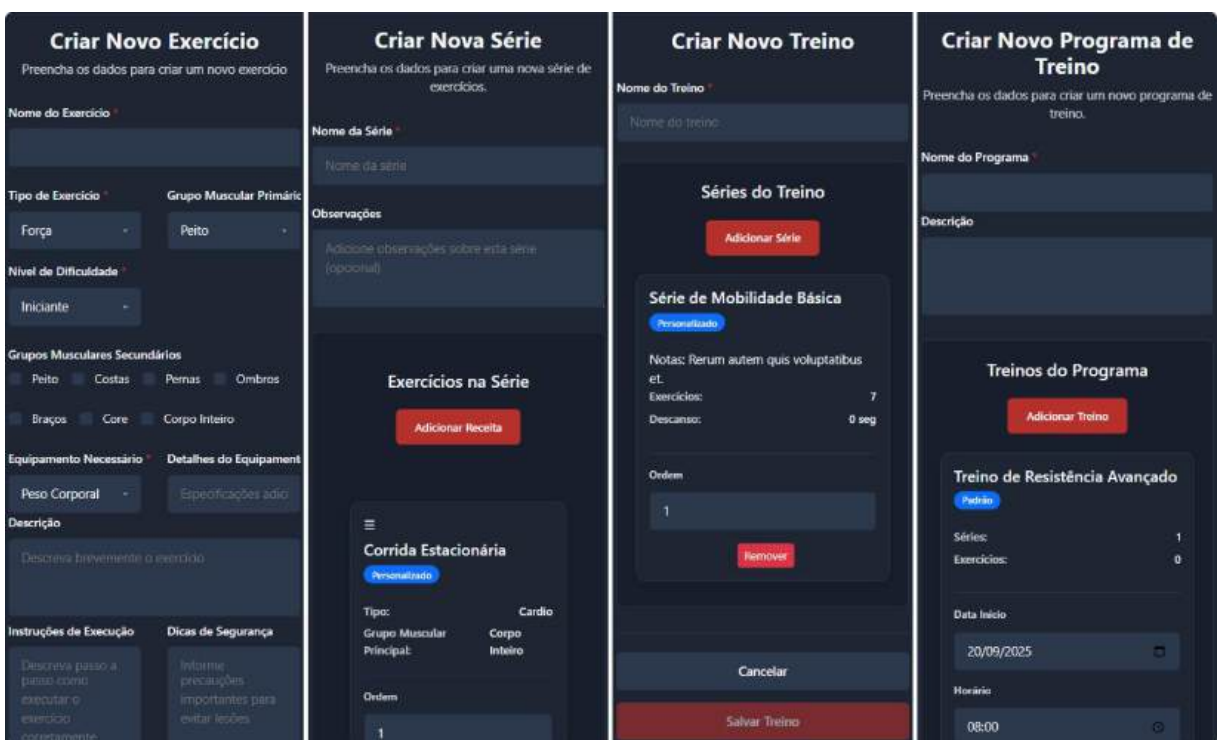
## Apêndice D - Interfaces Finais

Este apêndice apresenta as interfaces finais desenvolvidas para os três perfis de usuários do sistema: treinador, nutricionista e aluno. Cada interface foi projetada com base em princípios de usabilidade, organização hierárquica de dados e fluxo funcional coerente com as responsabilidades de cada perfil. O objetivo é demonstrar como o sistema proporciona uma experiência integrada e intuitiva para todos os profissionais envolvidos no ecossistema da aplicação.

A interface do treinador (Figura 12), permite criar, de forma sequencial, toda a estrutura de treinos:

- Exercício: definição individual de cada exercício;
- Série de treino: agrupamento de vários exercícios;
- Treino: composto por várias séries de treino para um dia específico;
- Programa de treino: organização dos treinos diários distribuídos ao longo da semana, incluindo horários.

Figura 12 - Interface do Treinador.

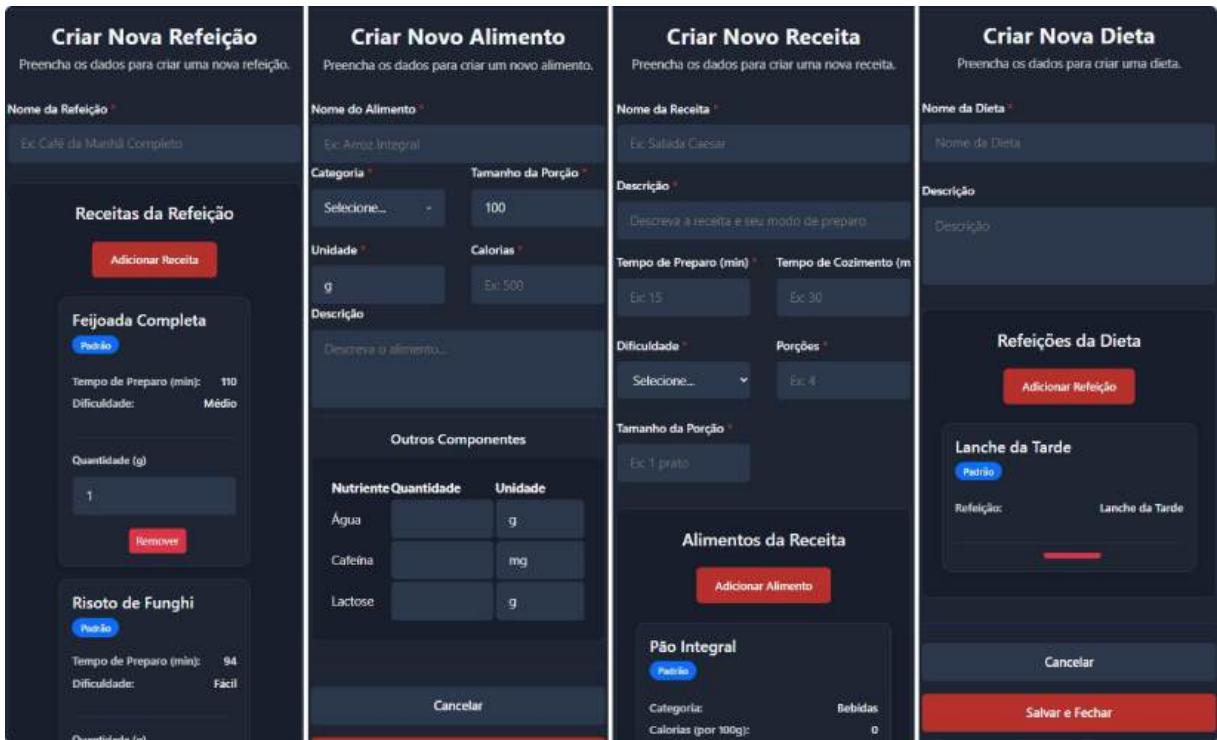


Fonte: Elaborado pelo autor, 2025.

A interface do nutricionista (Figura 13), permite criar, de forma sequencial, toda a estrutura alimentar, desde o alimento presente em uma refeição, até a atribuição para seu paciente:

- Comida: cadastro individual de alimentos;
- Receita: conjunto de comidas;
- Refeição: composta por várias receitas;
- Dieta: conjunto de refeições distribuídas ao longo dos dias da semana, com horários definidos para cada refeição.

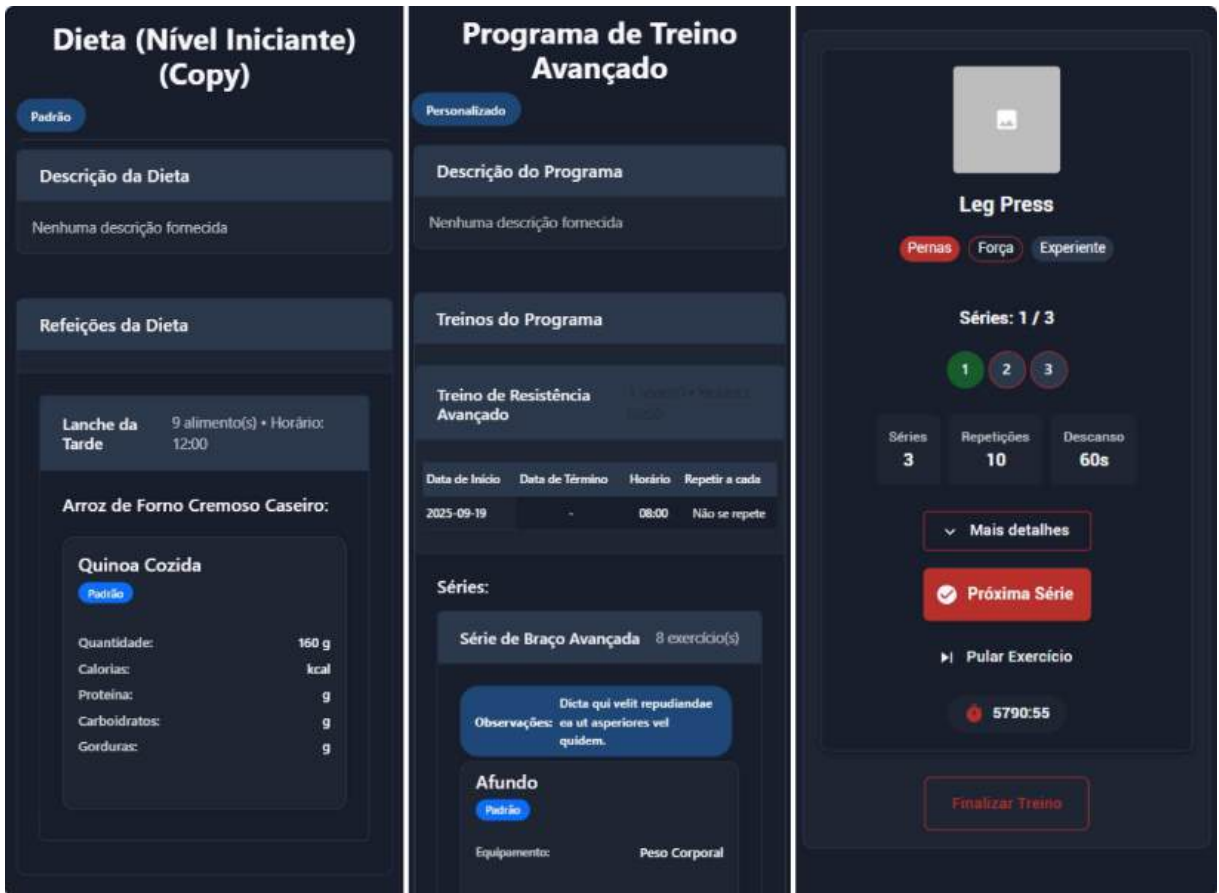
Figura 13 - Interface do Nutricionista.



Fonte: Elaborado pelo autor, 2025

Através da interface do aluno (Figura 14), é possível visualizar o programa de treino e a dieta atribuídos individualmente, bem como acompanhar o progresso diário das atividades. O aluno pode realizar cada exercício e série de forma sequencial, até a conclusão completa do treino do dia.

Figura 14 - Interface do Aluno.



Fonte: Elaborado pelo autor, 2025.

## Apêndice E - Estrutura da base de dados

Este apêndice apresenta os diagramas de entidade-relacionamento (DER) de estrutura da base de dados da API, organizada em três módulos principais: Nutrição, Treinamento e Usuários/Academias. Cada diagrama ilustra as entidades, seus atributos e os relacionamentos.

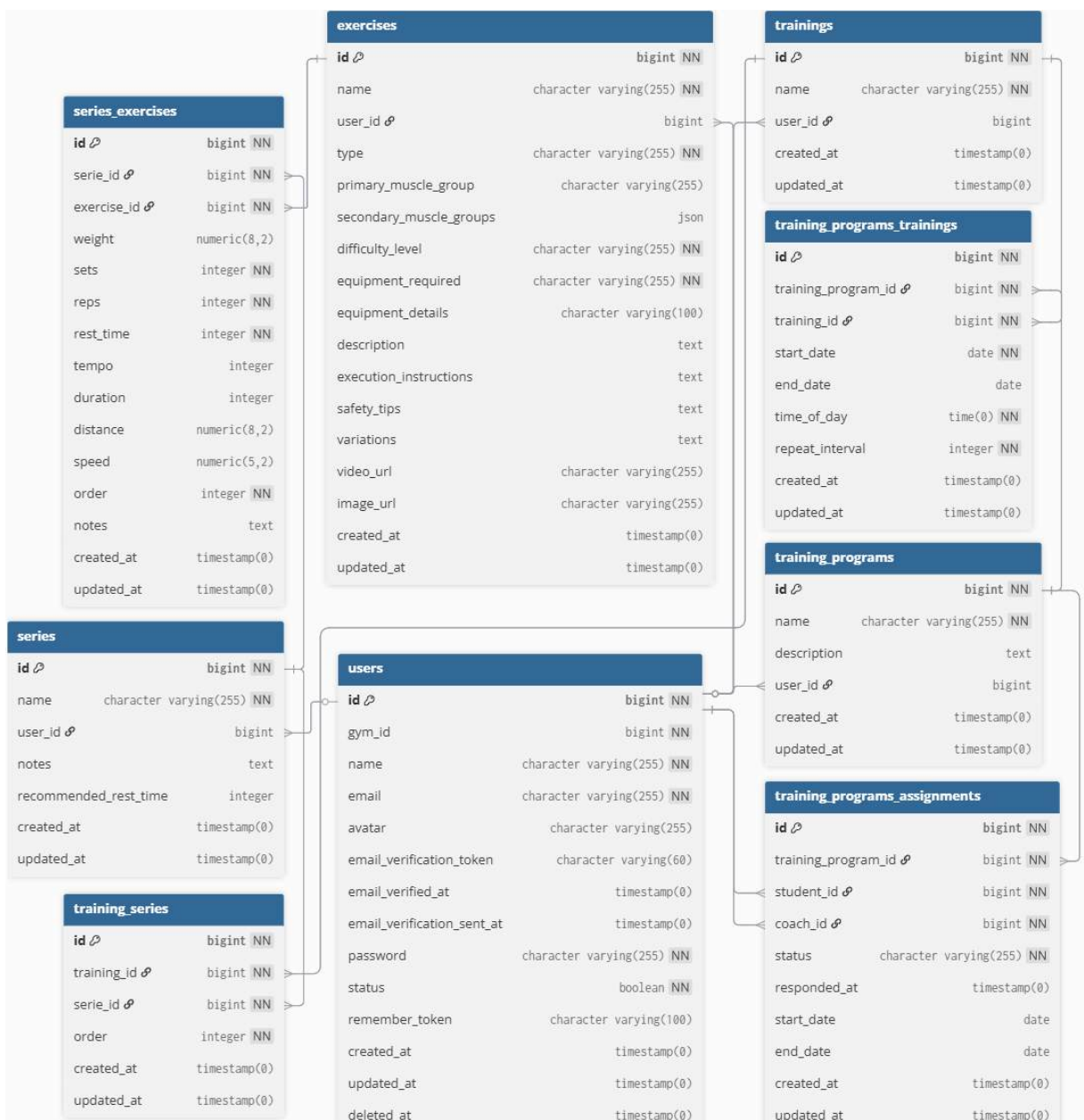
Figura 15 - Diagrama Entidade-Relacionamento do Módulo de Nutrição



Fonte: Elaborado pelo autor, 2025.

A Figura 15 apresenta o diagrama que organiza o gerenciamento das informações nutricionais, desde o cadastro de alimentos até a composição de dietas completas. O modelo inclui as entidades *Food*, para itens alimentares com informações nutricionais; *Recipe*, que combina alimentos para criar uma receita; *Meal*, refeição que agrupa receitas para momentos específicos do dia; e *Diet*, que comporta várias refeições para serem seguidas durante a semana. Os relacionamentos entre essas entidades permitem que nutricionistas criem planos alimentares personalizados para os usuários, garantindo flexibilidade e reutilização dos componentes.

Figura 16 - Diagrama Entidade-Relacionamento do Módulo de Treinamento



Fonte: Elaborado pelo autor, 2025.

A Figura 16 organiza a criação e gestão de programas de treino. O modelo inclui as entidades *Exercise*, para atividades físicas; *Serie*, agrupando exercícios através de uma tabela auxiliar; *Training*, representando sessões de treino compostas por várias séries; e *Training Program*, que organiza os treinos ao longo da semana com horários e repetições. Essa estrutura permite que os treinadores desenvolvam programas complexos e adaptáveis, atribuíveis a múltiplos alunos.

Figura 17 - Diagrama Entidade-Relacionamento do Módulo de Usuários e Academias



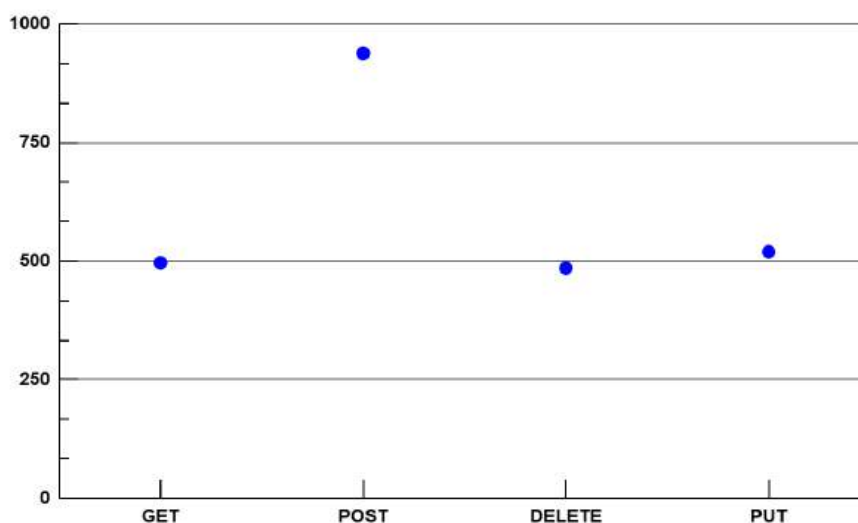
Fonte: Elaborado pelo autor, 2025.

A Figura 17 apresenta a estrutura de usuários e suas relações com academias e perfis. O modelo inclui as entidades *Users*, armazenando todos os usuários e *Gyms*, representando academias ou instituições; Os relacionamentos permitem controlar o cadastro de profissionais e alunos, além de vinculá-los a associações.

## Apêndice F - Tempo de resposta da API

Este apêndice apresenta uma análise detalhada do desempenho da API com base no tempo de resposta, medido desde o envio da requisição até a recepção do primeiro *byte* de retorno (*Time to First Byte*). O gráfico, representado na Figura 18, consolida os tempos médios de resposta agrupados por método HTTP, permitindo identificar padrões de desempenho nas 101 rotas da aplicação.

Figura 18 - Tempo médio de resposta da API por método HTTP



Fonte: Elaborado pelo autor, 2025.

Complementando a análise geral, o Quadro 12 detalha as três rotas com maior latência na aplicação, todas relacionadas ao processo de autenticação. Os tempos elevados de resposta são atribuídos principalmente às operações síncronas de envio de e-mail realizadas através do serviço gratuito de SMTP, que consome aproximadamente 67% do tempo total de processamento nestes *endpoints*.

Quadro 12 - Rotas com Maior Tempo de Resposta na Aplicação

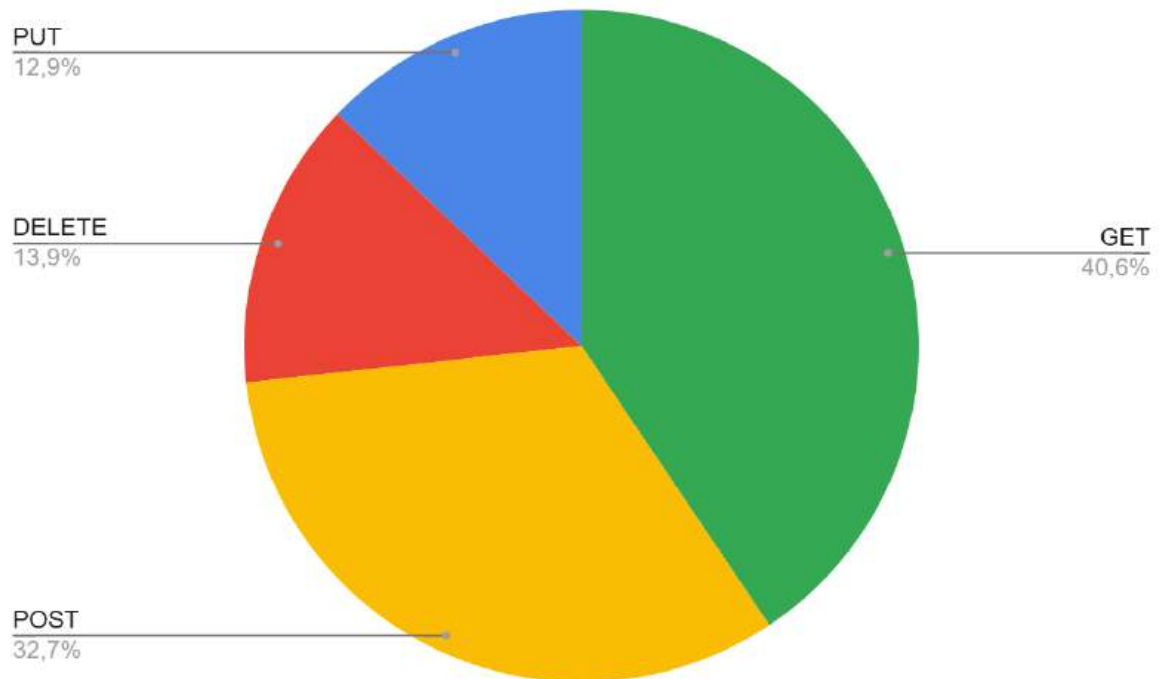
Rota	Método HTTP	Descrição	Tempo de Resposta (ms)
/auth/send-verification	POST	Reenvio de verificação	5770
/auth/register	POST	Registrar usuário	4204
/auth/forgot-password	POST	Solicitar reset de senha	4182

Fonte: Elaborado pelo autor, 2025.

## Apêndice G - Distribuição total de rotas, fracionadas por método HTTP

Este apêndice oferece uma visualização alternativa da distribuição dos métodos HTTP na arquitetura da API, discutida no Quadro 7. O gráfico de pizza presente na Figura 16 reforça visualmente o predomínio de operações de leitura (GET - 40,6%), seguido pela criação de recursos (POST - 32,7%), o que é consistente com o padrão de uso esperado para uma API de gestão e consulta. A presença significativa de operações PUT e DELETE (12,9% e 13,9%, respectivamente) evidencia a capacidade da aplicação em suportar um ambiente dinâmico, onde treinos e dietas são frequentemente ajustados pelos profissionais.

Figura 16 - Gráfico de pizza da distribuição total de rotas, fracionadas por método HTTP.



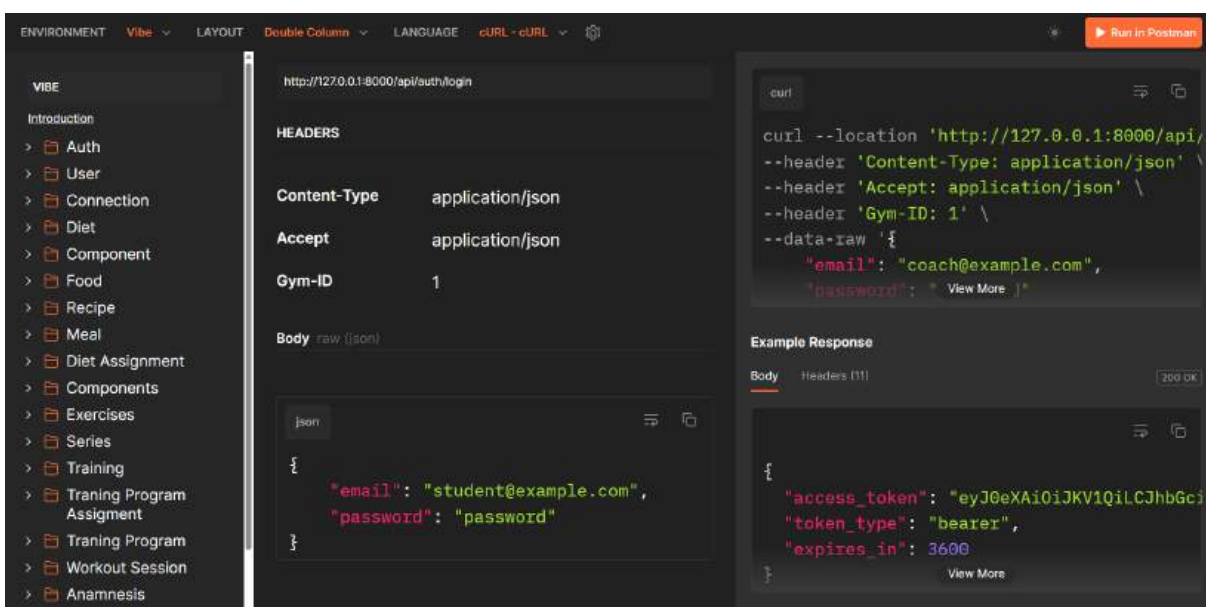
Fonte: Elaborado pelo autor, 2025.

## Apêndice H - Documentação POSTMAN

Este apêndice apresenta a documentação interativa e pública da API, gerada automaticamente pela ferramenta Postman (Figura 20). A documentação serve como o principal ponto de referência para desenvolvedores que desejam integrar suas aplicações *front-end* (como aplicativos *móveis* ou *web*) com a API *RESTful* desenvolvida neste trabalho.

A documentação completa da API desenvolvida neste projeto pode ser acessada por meio de um link público disponibilizado na plataforma Postman<sup>14</sup>.

Figura 20 - Documentação da API.



Fonte: Elaborado pelo autor, 2025.

<sup>14</sup> Disponível em: <https://documenter.getpostman.com/view/25294402/2sB3QNo7ur>. Acesso em: 30 out. 2025.

## Apêndice I - Utilização de autenticação JWT

Este apêndice apresenta a implementação do mecanismo de autenticação segura baseado em JWT, utilizado para o controle de acesso às rotas protegidas da API.

Na Figura 21, observa-se a aplicação do *middleware* de autenticação JWT sobre as rotas *logout*, *refresh* e *me*, responsáveis, respectivamente, por encerrar a sessão do usuário, renovar o *token* de acesso e retornar informações detalhadas do próprio usuário autenticado.

Além dessas, existem rotas públicas, *verify-email*, *send-verification*, *login*, *register*, *forgot-password* e *reset-password*, que permanecem fora do escopo do *middleware*, permitindo o acesso sem necessidade de autenticação prévia.

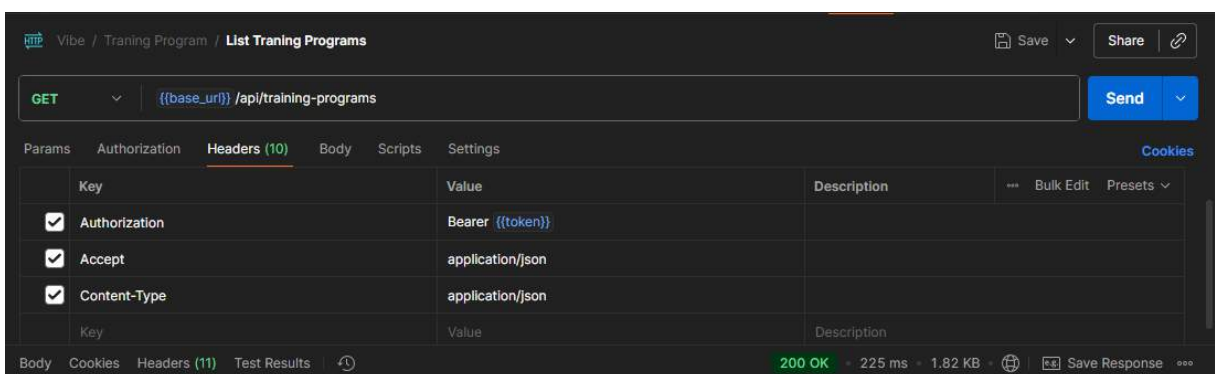
Figura 21 - Utilização do Middleware de Autenticação Nas Rotas da API

```
Route::prefix(prefix: 'auth')->group(callback: function (): void {
    Route::get(uri: '/verify-email/{token}', action: [AuthController::class, 'verifyEmail']);
    ->name(name: 'verification.verify');
    Route::post(uri: '/send-verification', action: [AuthController::class, 'sendVerificationEmail']);
    Route::post(uri: '/login', action: [AuthController::class, 'login']);
    Route::post(uri: '/register', action: [AuthController::class, 'register']);
    Route::post(uri: '/forgot-password', action: [AuthController::class, 'forgotPassword']);
    Route::post(uri: '/reset-password', action: [AuthController::class, 'resetPassword'])
    ->name(name: 'password.reset');
});
Route::middleware(middleware: ['jwt.auth', 'user.verified', 'gym.requests.counter'])
->group(callback: function (): void {
    Route::get(uri: '/users/{id}', action: [UserController::class, 'showProfile']);
    Route::prefix(prefix: 'auth')->group(callback: function (): void {
        Route::post(uri: '/logout', action: [AuthController::class, 'logout']);
        Route::post(uri: '/refresh', action: [AuthController::class, 'refresh']);
        Route::get(uri: '/me', action: [AuthController::class, 'me']);
    });
});
```

Fonte: Elaborado pelo autor, 2025.

A Figura 22 demonstra o envio do *token* de autenticação por meio do cabeçalho de uma requisição HTTP. O teste foi realizado no Postman, utilizando variáveis de ambiente, neste caso, a variável (*token*) para simular o processo de autenticação em ambiente controlado.

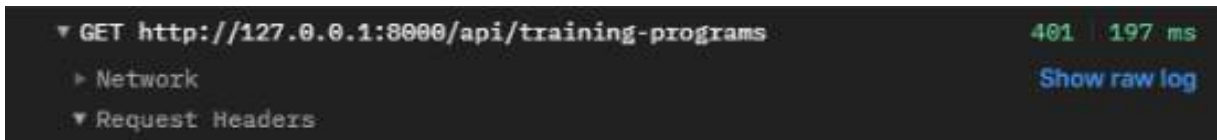
Figura 22 - Interface De Requisição no Postman



Fonte: Elaborado pelo autor, 2025.

Por fim, a Figura 23 ilustra o comportamento da API diante de uma requisição sem o cabeçalho de autenticação. Como esperado, a resposta retorna o *status code* 401 (*Unauthorized*), confirmando a eficácia do mecanismo de segurança implementado.

Figura 23 - Log de Erro de Autenticação



Fonte: Elaborado pelo autor, 2025.

Apêndice J - Comparação do tempo de resposta da API antes e depois da implementação de *Cache*

Este apêndice apresenta a comparação do desempenho da API antes e após a implementação do mecanismo de *cache*, com foco no tempo médio de resposta das principais rotas de acesso. A análise (Quadro 13) visa demonstrar a eficiência obtida com o uso dessa técnica de otimização.

Quadro 13 - Comparação do tempo de resposta da API para as principais rotas.

Rota	Método	Descrição	Sem Cache	Com Cache
meals	GET	Listar refeições	200	210
exercises	GET	Listar exercícios	385	213
recipes	GET	Listar receitas	510	213
training-programs	GET	Listar programas	424	214
components	GET	Listar componentes	407	215
diets	GET	Listar dietas	423	216
trainings	GET	Listar Treinos	616	216
foods	GET	Listar alimentos	551	218
series	GET	Listar Séries	371	220

Fonte: Elaborado pelo autor, 2025.

O tempo médio de resposta para as rotas listadas sem a implementação do *cache* foi de 371 ms, enquanto com o *cache* ativado reduziu-se para 215 ms, representando uma melhoria de aproximadamente 42% no desempenho da API.