



Instituto Federal de Brasília
Campus Brasília
Curso Superior de Tecnologia em Sistemas para Internet

ANTONIO ROBERTO DE OLIVEIRA JÚNIOR

MAYCON DOUGLAS NUNES DE ALMEIDA

FITAI: TREINO PERSONALIZADO COM INTELIGÊNCIA ARTIFICIAL

Brasília
2025

ANTONIO ROBERTO DE OLIVEIRA JÚNIOR
MAYCON DOUGLAS NUNES DE ALMEIDA

FitAI: TREINO PERSONALIZADO COM INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado ao curso de Tecnologia em Sistemas para Internet do Instituto Federal de Brasília, Campus Brasília como parte da exigência para obtenção do título de Tecnólogo em Sistemas de Internet.

Orientador: Professor Dr. Francisco Euder dos Santos

Brasília
2025

O48 Oliveira Júnior, Antônio Roberto de.
FITAI: treino personalizado com inteligência artificial. / Antonio Roberto de Oliveira Júnior ; Maycon Douglas Nunes de Almeida. – Brasília, 2025.
93 f. : il. color.

Orientador: Francisco Euder dos Santos.

Trabalho de conclusão de curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Brasília, Tecnologia em Sistemas para Internet, 2025.

1. Inteligência Artificial. 2. Fitness. 3. Personalização de Treinos. 4. Django REST Framework. 5. Flutter. I. Almeida, Maycon Douglas Nunes de. II. Santos, Francisco Euder dos. (orient.). III. Título.

CDU 004.41:796

ANTONIO ROBERTO DE OLIVEIRA JÚNIOR

MAYCON DOUGLAS NUNES DE ALMEIDA

FITAI: TREINO PERSONALIZADO COM INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado ao curso de Tecnologia em Sistemas para Internet do Instituto Federal de Brasília, Campus Brasília como parte da exigência para obtenção do título de Tecnólogo em Sistemas de Internet.

Aprovado em _____ de _____ de _____.

BANCA EXAMINADORA

Dr. Francisco Euder dos Santos
Instituto Federal de Brasília – Campus Brasília
Presidente / Orientador

Dra. Cristiane Jorge de Lima Bonfim
Instituto Federal de Brasília – Campus Brasília
Membro interno

Me. Alex Helder Cordeiro de Oliveira
Instituto Federal de Brasília – Campus Brasília
Membro interno

Brasília
2025

Dedico este trabalho àqueles que, de alguma forma, contribuíram para nossa formação e para a superação dos obstáculos ao longo deste percurso.

AGRADECIMENTOS

Aos nossos familiares e amigos, pelo apoio essencial na concretização deste projeto.

Agradecemos ao orientador, Prof. Dr. Francisco Euder dos Santos, pela orientação, durante o desenvolvimento desta pesquisa.

A todos os professores que, com dedicação e conhecimento, contribuíram para a nossa formação profissional.

Ao Instituto Federal de Brasília, pelo suporte técnico e educacional que tornou este projeto possível.

“Se você quer resultados diferentes, faça algo diferente.”

— **Albert Einstein**

RESUMO

Oliveira Júnior, Antonio Roberto. Almeida, Maycon Douglas. FitAi: Treino personalizado com inteligência artificial. 2025. 93 f. Trabalho de Conclusão de Curso – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – Campus Brasília. Brasília/DF, 2025.

A inteligência artificial, em sua rápida evolução, tem impulsionado avanços significativos no setor de *fitness* e bem-estar, proporcionando o desenvolvimento de aplicações especializadas para personalização da experiência de treinamento. Este trabalho apresenta o FitAI, uma plataforma híbrida desenvolvida utilizando Django REST Framework como *backend* e Flutter no aplicativo móvel. Seu foco principal é a personalização de treinos físicos por meio de inteligência artificial generativa. A proposta oferece uma experiência móvel nativa com alta performance e personalização baseada em Inteligência Artificial (IA), atendendo às demandas contemporâneas por soluções digitais no setor *fitness*. A metodologia caracteriza-se como pesquisa aplicada de natureza exploratória-experimental com abordagem quali-quantitativa, envolvendo identificação de requisitos técnicos e funcionais, projeto de arquitetura escalável integrando Django REST Framework, Flutter, Firebase Authentication e Google Gemini API. O sistema implementa geração de treinos personalizados através de *Zero-Shot Learning* com engenharia de *prompts* estruturados, superando limitações de *cold start* de algoritmos tradicionais. A arquitetura híbrida combina IA generativa como motor primário com sistema de fallback baseado em regras especializadas, garantindo continuidade de serviço quando *APIs* externas estão indisponíveis. Foram desenvolvidas interfaces responsivas seguindo Material Design 3, otimizadas para uso durante exercícios físicos. A validação da usabilidade não envolveu usuários reais, pois o escopo do trabalho é uma prova de conceito. Optou-se pela avaliação heurística, método consolidado em engenharia de usabilidade, utilizando os dez princípios de Nielsen para inspeção sistemática da interface. Essa abordagem é adequada para identificar problemas potenciais de usabilidade em protótipos, sem necessidade de testes com participantes.

Palavras-chave: Inteligência Artificial. Aplicações *Fitness*. Django REST Framework. Flutter. Personalização de Treinos.

RESUMO EM LÍNGUA ESTRANGEIRA

Oliveira Júnior, Antonio Roberto. Almeida, Maycon Douglas. FitAI: Personalized Training with Artificial Intelligence. 2025. 93 f. Final Paper — Undergraduate Program in Internet Systems. Federal Institute of Brasília — Brasília Campus. Brasília/DF, 2025.

Artificial intelligence, in its rapid evolution, has driven significant advances in the fitness and wellness sector, enabling the development of specialized applications for personalizing training experiences. This work presents FitAI, a hybrid platform developed using Django REST Framework as backend and Flutter for mobile application. Its main focus is the personalization of physical training through generative artificial intelligence. The proposal offers users a native mobile experience with high performance and AI-based personalization, meeting contemporary demands for digital solutions in the fitness sector. The methodology is characterized as applied research of exploratory-experimental nature with qualitative-quantitative approach, involving identification of technical and functional requirements, design of scalable architecture integrating Django REST Framework, Flutter, Firebase Authentication and Google Gemini API. The system implements personalized workout generation through Zero-Shot Learning with structured prompt engineering, overcoming cold start limitations of traditional algorithms. The hybrid architecture combines generative AI as primary engine with fallback system based on specialized heuristic rules, ensuring service continuity when external APIs are unavailable. Responsive interfaces were developed following Material Design 3 guidelines, optimized for use during physical exercises. Usability validation did not involve real users, as the scope of the work is a proof of concept. Heuristic evaluation was chosen, a well-established method in usability engineering, using Nielsen's ten principles for systematic interface inspection. This approach is suitable for identifying potential usability issues in prototypes without the need for participant testing.

Keywords: Artificial Intelligence. Fitness Applications. Django REST Framework. Flutter. Workout Personalization.

Lista de Figuras

Figura 1: Diagrama de arquitetura.....	64
Figura 2: Diagrama de Sequência.....	65
Figura 3: Diagrama ER.....	69
Figura 4: Diagrama de caso de uso.....	70
Figura 5 Tela de login.....	81
Figura 6 Tela meta.....	81
Figura 7 Tela área de foco.....	81
Figura 8 Dashboard Principal.....	82
Figura 9 Dashboard Principal.....	82
Figura 10 Listagem de Treinos.....	83
Figura 11 Tela descanso.....	84
Figura 12 Tela treino em andamento.....	84
Figura 13 Tela de treino.....	84

Índice de Quadros

Quadro 1 – Comparativo de Aplicativos de Fitness.....	33
Quadro 1 – Requisitos Funcionais.....	44
Quadro 2 – Requisitos Não-Funcionais.....	45

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface
CRUD – Create, Read, Update, Delete
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
IA – Inteligência Artificial
IBGE – Instituto Brasileiro de Geografia e Estatística
IMC – Índice de Massa Corporal
IoT – Internet of Things
JSON – JavaScript Object Notation
JWT – JSON Web Token
LGPD – Lei Geral de Proteção de Dados
LLM – Large Language Model
LSTM – Long Short-Term Memory
ML – Machine Learning
MVC – Model-View-Controller
MVP – Minimum Viable Product
ORM – Object-Relational Mapping
REST – Representational State Transfer
RF – Requisito Funcional
RNF – Requisito Não Funcional
RNN – Recurrent Neural Networks
SDK – Software Development Kit
SQL – Structured Query Language
TLS – Transport Layer Security
UFSC – Universidade Federal de Santa Catarina
UID – User Identifier
URL – Uniform Resource Locator
UX – User Experience
WCAG – Web Content Accessibility Guidelines

DECLARAÇÃO DE USO DE TECNOLOGIAS E FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL (IA) GENERATIVA, SOFTWARES E OUTRAS FERRAMENTAS DE APOIO

Como ferramenta de apoio ao desenvolvimento, utilizou-se a inteligência artificial generativa Google Gemini (versão 2025). O recurso foi empregado para auxílio na depuração (*debugging*) e otimização de trechos de código-fonte nas linguagens Dart e Python, bem como para a revisão gramatical e adequação textual da documentação escrita, observando a norma culta. Ressalta-se que todo o conteúdo gerado foi validado, revisado e adaptado pelos autores, mantendo-se a integridade e originalidade da pesquisa (GOOGLE, 2025).

Reconhece-se a utilização do editor de texto LibreOffice (*The Document Foundation*) para a escrita e formatação do texto deste presente trabalho.

Sumário

1 INTRODUÇÃO.....	15
1.1 Tema.....	15
1.2 Problema de pesquisa.....	16
1.3 Objetivos.....	16
1.3.1 Objetivo geral.....	17
1.3.2 Objetivos específicos.....	17
1.4 Estrutura do TCC e Classificação da pesquisa.....	17
2 CONCEITOS GERAIS E REVISÃO DE LITERATURA.....	19
2.1 Inteligência Artificial Aplicada ao Fitness.....	19
2.2 Tecnologias de Desenvolvimento Web e Mobile.....	22
2.2.1 Django REST Framework.....	22
2.2.2 Flutter.....	24
2.2.3 Integração Django-Flutter.....	27
2.3 Sistemas de Recomendação e Abordagens de Inteligência Artificial.....	28
2.4 Aplicações Práticas e Trabalhos Correlatos.....	29
2.5 Desafios e Oportunidades.....	32
2.5.1 Desafios Técnicos.....	32
2.5.2 Desafios Não Técnicos.....	34
2.5.3 Oportunidades de Mercado e Inovação.....	35
2.5.4 Implicações para o FitAI.....	37
3 METODOLOGIA.....	39
3.1 Tipo de pesquisa.....	39
3.2 Procedimentos Metodológicos.....	40
3.3 Especificação de Requisitos.....	42
3.4 Ferramentas e tecnologias utilizadas.....	44
3.5 Critérios de avaliação.....	47

4. DESENVOLVIMENTO E IMPLEMENTAÇÃO DO SISTEMA FITAI	49
4.1 Arquitetura da Plataforma FitAI.....	49
4.1.1 Camadas de Aplicação e Inteligência Artificial.....	50
4.2 Fluxo de Dados.....	53
4.3 Modelagem do Banco de Dados.....	54
4.4 Casos de Uso.....	56
4.5 Design das APIs REST.....	58
4.6 Algoritmos de Inteligência Artificial.....	59
4.6.1 Fundamentação da Arquitetura Híbrida.....	60
4.6.2 Integração com IA Generativa (Google Gemini).....	60
4.6.3 Sistema de Recomendação Diária Inteligente.....	62
4.6.4 Geração de Planos de Treino Personalizados.....	63
4.6.5 Sistema de Chat Inteligente.....	64
4.7 Interface do Usuário (Flutter).....	65
4.7.2 Telas Principais.....	65
4.7.3 Sistema de Design e Padrões Visuais.....	70
5 RESULTADOS E ANALISES.....	71
5.1 Contexto de Desenvolvimento e Escopo da Avaliação.....	71
5.2 Validação da Arquitetura Híbrida.....	71
5.3 Análise dos Algoritmos de Inteligência Artificial.....	72
5.4 Análise de Interface e Usabilidade (UX).....	73
5.5 Considerações sobre Segurança e Dados.....	74
5.6 Limitações do Estudo.....	74
6 CONCLUSÃO.....	75
6.1 Trabalhos Futuros.....	76
REFERÊNCIAS.....	78
APÊNDICE A – DIAGRAMAS DO SISTEMA.....	85
APÊNDICE B – FLUXO CADASTRO.....	89

1 INTRODUÇÃO

O crescimento exponencial da inteligência artificial tem revolucionado diversos setores, incluindo o *fitness* e bem-estar, onde aplicações inovadoras emergem para personalizar experiências de treinamento. Este trabalho apresenta o desenvolvimento do FitAI, uma plataforma híbrida que combina Django REST Framework como *backend* e Flutter como aplicativo móvel. O objetivo é a criação de treinos personalizados com o suporte da inteligência artificial. A proposta é criar uma solução tecnológica que permita aos usuários finais uma experiência móvel nativa com alta performance e personalização baseada em IA. O projeto representa uma resposta às demandas contemporâneas por soluções digitais no setor de saúde e bem-estar, especialmente considerando o crescimento do mercado de aplicativos de *fitness* com IA (ROWE; NICHOLLS; SHAW, 2022). O trabalho explora a integração de tecnologias modernas de desenvolvimento web e mobile, incorporando *APIs* de inteligência artificial generativa para criação de conteúdo personalizado.

1.1 Tema

O tema central desta pesquisa foca na criação de uma plataforma digital voltada a praticantes de atividade física que buscam orientação personalizada por meio de aplicativos móveis, utilizando inteligência artificial. A escolha do tema é justificada pelo crescimento acelerado do mercado de *fitness* digital e pela necessidade de soluções que atendam às demandas individuais de quem pratica exercícios. O interesse surgiu da observação das limitações dos métodos tradicionais de prescrição de exercícios e do potencial transformador da IA na área de fitness. Este tema merece investigação devido ao seu impacto potencial na democratização do acesso a treinamentos personalizados, tradicionalmente limitados por barreiras econômicas. Pesquisa de mercado em plataformas de contratação de serviços indica que sessões de *personal training* no Brasil variam entre R\$ 70 e R\$ 150 por hora, com média nacional de R\$ 86 por hora (SUPERPROF, 2025; GETNINJAS, 2025). Considerando frequência recomendada de três sessões semanais, o investimento mensal situa-se entre R\$ 840 e R\$ 1.800,

representando entre 45% e 97% da renda média domiciliar per capita brasileira de R\$ 1.848 (IBGE, 2023). Este valor inviabiliza que grande parte da população tenha acesso ao serviço, mesmo considerando que os exercícios poderiam ser realizados em casa, pois a barreira econômica permanece. Nesse contexto, as soluções digitais baseadas em IA surgem como alternativa viável para oferecer orientação personalizada a baixo custo, ampliando o acesso à prática segura e eficiente de exercícios físicos. (DERGAA *et al.*, 2024; DUFFIN, 2024; OLIVEIRA; FRAGA, 2021).

Além disso, a convergência entre tecnologias de desenvolvimento móvel multiplataforma, como Flutter, e *frameworks* web robustos, como Django, oferece oportunidades únicas para criar soluções escaláveis e eficientes.

Este projeto explora especificamente o uso de modelos de linguagem de grande escala (LLMs) para geração de planos de treino, visando superar barreiras de entrada comuns em algoritmos tradicionais de recomendação de exercícios físicos.

1.2 Problema de pesquisa

O acesso a treinos personalizados no Brasil é limitado por barreiras econômicas, já que o custo médio de acompanhamento por *personal trainer* pode representar até 97% da renda domiciliar per capita (IBGE, 2023; SUPERPROF, 2025; GETNINJAS, 2025). Embora os exercícios possam ser realizados em casa, a falta de orientação adequada compromete a segurança e a eficácia do treinamento. Nesse contexto, soluções digitais baseadas em inteligência artificial surgem como alternativa para democratizar o acesso à personalização, oferecendo recomendações adaptadas ao perfil do usuário a baixo custo.

Como desenvolver uma plataforma integrada de treinamento personalizado com o suporte de inteligência artificial para gerar recomendações de exercícios adaptadas ao perfil individual dos usuários, mantendo alta performance técnica e experiência de usuário superior através de uma arquitetura híbrida Django-Flutter?

1.3 Objetivos

Os objetivos deste trabalho foram definidos com o propósito de orientar o desenvolvimento do FitAI, garantindo que cada etapa contribua para solucionar o problema de pesquisa apresentado. Esses objetivos buscam estruturar de forma clara e organizada as ações necessárias para alcançar o resultado proposto, desde a análise dos requisitos técnicos até a avaliação da eficácia da plataforma. Assim, foram estabelecidos um objetivo geral, que sintetiza a meta principal do projeto, e objetivos específicos, que detalham as etapas a serem executadas para tornar possível a implementação e validação da solução.

1.3.1 Objetivo geral

Desenvolver uma plataforma integrada de treinamento personalizado com suporte de inteligência artificial para gerar recomendações de exercícios adaptadas ao perfil individual dos usuários, garantindo alta performance técnica e experiência de usuário superior por meio de uma arquitetura Django-Flutter.

1.3.2 Objetivos específicos

Com o intuito de alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- a) Realizar uma pesquisa bibliográfica sobre plataformas digitais de prescrição de treinos com inteligência artificial para fundamentar o desenvolvimento da FitAI.
- b) Identificar os requisitos técnicos e funcionais para o desenvolvimento da plataforma FitAI.
- c) Projetar uma arquitetura escalável, integrando Django REST Framework, Flutter, Firebase Authentication e serviços de IA generativa para recomendações personalizadas.
- d) Desenvolver a plataforma, incluindo o *backend* Django REST com integração

Firebase, as interfaces móveis multiplataforma em Flutter e o sistema de autenticação híbrida.

e) Integrar algoritmos de IA e *APIs* de geração de conteúdo, garantindo recomendações e dicas de treino personalizadas.

1.4 Estrutura do TCC e Classificação da pesquisa

Este trabalho está organizado em capítulos que seguem uma sequência lógica para apresentar o desenvolvimento completo da plataforma FitAI.

A organização deste estudo reflete a sequência lógica de desenvolvimento da pesquisa. Após esta introdução, apresenta-se a fundamentação teórica e tecnológica (Capítulo 2), seguida pela definição dos procedimentos metodológicos (Capítulo 3). O desenvolvimento da arquitetura, implementação do sistema e algoritmos é detalhado no Capítulo 4. Por fim, discutem-se os resultados da validação técnica (Capítulo 5) e as conclusões finais (Capítulo 6).

Esta pesquisa se classifica como aplicada quanto à finalidade, exploratória descritiva quanto aos objetivos, e qualiquantitativa quanto à abordagem. Quanto aos métodos, constitui-se como pesquisa de desenvolvimento tecnológico e prova de conceito, focando na validação técnica da arquitetura híbrida proposta através de implementação funcional e análise heurística de usabilidade. Os procedimentos metodológicos detalhados, incluindo técnicas específicas de coleta de dados, ferramentas utilizadas e critérios de avaliação, são apresentados no Capítulo 3, que detalha a abordagem metodológica adotada para operacionalização dos objetivos estabelecidos.

Estabelecida a natureza metodológica da pesquisa e sua classificação quanto à finalidade, objetivos, abordagem, métodos e procedimentos, torna-se necessário fundamentar teoricamente as decisões arquiteturais que operacionalizam os objetivos propostos. O capítulo seguinte apresenta revisão sistemática da literatura especializada, abordando os fundamentos de inteligência artificial aplicada ao *fitness*, as tecnologias de desenvolvimento web e mobile que viabilizam a implementação da plataforma, os algoritmos de inteligência artificial relevantes ao contexto de recomendações personalizadas, e a análise crítica de trabalhos. Esta

fundamentação teórica orienta não apenas as escolhas tecnológicas específicas, mas também estabelece *framework* conceitual para compreender os desafios técnicos, éticos e regulatórios que devem ser endereçados, bem como as oportunidades estratégicas que justificam o desenvolvimento da solução proposta.

2 CONCEITOS GERAIS E REVISÃO DE LITERATURA

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento da plataforma FitAI, abordando os principais conceitos relacionados à inteligência artificial aplicada ao segmento *fitness*, tecnologias de desenvolvimento web e mobile, e trabalhos correlatos na área.

2.1 Inteligência Artificial Aplicada ao Fitness

A implementação de sistemas de inteligência artificial no setor *fitness* tem emergido como área de pesquisa científica promissora, modificando a abordagem para a prescrição, monitoramento e personalização de exercícios. A tecnologia de IA viabiliza o processamento eficiente de grandes volumes de dados provenientes de múltiplas fontes, permitindo análises personalizadas tanto em nível individual quanto para conjuntos de usuários de forma simultânea.

Canzone *et al.* (2025) conduziram uma revisão narrativa abrangente que examinou os múltiplos usos da inteligência artificial em programas de exercícios, destacando aplicações que vão desde aplicativos móveis de autogerenciamento para saúde musculoesquelética até programas híbridos de exercícios para sarcopenia em adultos com idade igual ou superior a 50 anos. O estudo, publicado na *Frontiers in Public Health*, enfatiza que a IA tem sido aplicada com sucesso em diversos contextos, incluindo, o desenvolvimento de aplicações para avaliação da atividade física e classificação de disfunção postural, bem como sistemas de autogerenciamento que demonstraram eficácia em ensaios clínicos controlados e randomizados para pacientes com dor no pescoço e/ou lombar.

A pesquisa identificou que as técnicas de inteligência artificial, *machine learning* e *deep learning* foram utilizadas em cinco categorias principais: (a) como parte de aplicações para gerar mensagens automáticas e comunicar-se com usuários; (b) como abordagem preditiva e para reconhecimento de gestos e postura; (c) como sistema de controle; (d) como coletor de dados; e (e) como treinador guiado (CANZONE *et al.*, 2025, p. 2).

Dergaa *et al.* (2024, p. 221) realizaram uma avaliação crítica da utilização de inteligência artificial para prescrição de exercícios em promoção personalizada da saúde, especificamente analisando o modelo GPT-4 da OpenAI. O estudo, publicado

na *Biology of Sport*, revelou que "embora os sistemas de IA possam oferecer uma alternativa economicamente viável para indivíduos que não podem arcar com *personal trainers* presenciais, a tecnologia ainda não oferece o grau de complexidade e adaptabilidade necessário para otimização de longo prazo da saúde e condicionamento físico" (DERGAA *et al.*, 2024, p. 235).

Esta limitação identificada na literatura fundamenta a escolha arquitetural do FitAI. Em vez de implementar modelos de *machine learning* tradicional que requerem extensos dados históricos para treinamento, o projeto adota abordagem baseada em inteligência artificial generativa (Google Gemini) combinada com regras heurísticas especializadas, descritas na subseção 4.6.1. Essas regras consistem em princípios de ciência do exercício aplicadas à lógica do sistema, garantindo segurança e adequação fisiológica das recomendações. Esta estratégia permite personalização imediata através de engenharia de *prompts* estruturados, incorporando princípios estabelecidos de ciência do exercício diretamente nas instruções fornecidas ao modelo, sem depender de dados de múltiplos usuários para aprendizado colaborativo.

No contexto brasileiro, Oliveira e Fraga (2021, p. 2) conduziram uma análise crítica de aplicações *fitness* que, de forma autônoma, recomendam exercícios físicos. Eles destacam que a prescrição de treinos com o auxílio da inteligência artificial articula um conjunto de variáveis para a elaboração do treinamento, convergindo para o que os autores denominam "protocolização das práticas corporais". O estudo, publicado na *Revista Brasileira de Ciências do Esporte*, enfatiza que:

Inteligência artificial permite processamento extremamente rápido de grande quantidade de dados de distintas fontes, possibilitando que o treinamento seja de fato personalizado já que a prescrição é ajustada de acordo com os dados do usuário (OLIVEIRA; FRAGA, 2021, p. 3).

Os resultados são comunicados aos usuários, seja de dados individuais ou de um conjunto de usuários através de análise de *big data*.

Os autores Oliveira e Fraga (2021), propõem que os profissionais de Educação Física se atentem a três pontos cruciais ao usar aplicativos de exercícios. Em vez de simplesmente restringir o indivíduo a variáveis e a padrões

preestabelecidos, eles devem reconhecer e valorizar o poder do contato humano. A pesquisa argumenta que, embora a tecnologia seja capaz de organizar um conjunto de dados, é responsabilidade do profissional aplicar competências como o diálogo, a escuta, a criatividade e o afeto, que são qualidades essencialmente humanas.

Os algoritmos de inteligência artificial aplicados ao *fitness* podem ser classificados em categorias distintas conforme sua funcionalidade e arquitetura, baseando-se em evidências da literatura científica (GUPTA *et al.*, 2022; FARROKHI *et al.*, 2021).

Redes Neurais Convolucionais (CNNs) são amplamente utilizadas para análise de postura e movimento através de processamento de imagem, permitindo correção em tempo real da execução de exercícios (NOORI *et al.*, 2020). Redes Neurais Recorrentes (RNNs) e suas variações, como *LSTM (Long Short-Term Memory)*, demonstram eficácia na análise de séries temporais de dados biométricos, permitindo predição de fadiga e otimização de intervalos de descanso (AMINIAN; NAJAFI, 2004).

Uma perspectiva histórica no Brasil indica que a aplicação de inteligência artificial ao *fitness* não é um conceito recente. Na análise Oliveira e Fraga (2021), o trabalho pioneiro de Delpizzo (1997) já abordava a prescrição de atividades físicas por meio de IA em uma dissertação de mestrado defendida na Universidade Federal de Santa Catarina (UFSC) na década de 1990.

Mais recentemente, Oliveira e Fraga (2021) investigaram o fenômeno do *Quantified Self* na cultura do *fitness* digital. A pesquisa analisou a forma como usuários de aplicativos de treino interagem com dados biométricos e métricas de performance, demonstrando que a cultura do *self-tracking*, baseada na premissa de que “você é seus dados”, influencia diretamente o comportamento de exercício.

Canzone *et al.* (2025) destacam que a pesquisa futura deve focar na inclusão de inteligência artificial, *machine learning* e *deep learning* em programas de atividade física e bem-estar. Os autores enfatizam que esses estudos devem avaliar a eficácia a longo prazo, a acessibilidade e a escalabilidade das ferramentas. Seus achados sugerem que a IA é uma ferramenta promissora para promover a atividade física em uma variedade de grupos populacionais.

A convergência entre tecnologias emergentes como *Internet of Things* (IoT), computação em nuvem e algoritmos de *machine learning* oferece oportunidades para desenvolvimento de sistemas mais sofisticados e personalizados, mantendo sempre o foco na segurança, eficácia e acessibilidade das soluções propostas.

2.2 Tecnologias de Desenvolvimento Web e Mobile

A implementação do FitAI adota arquitetura híbrida que combina Django REST Framework para *backend* e Flutter para *frontend* mobile, integrando ambas as plataformas através de *APIs RESTful*. A escolha tecnológica fundamenta-se em critérios definidos pelos autores com base na literatura: (1) maturidade e estabilidade das ferramentas, com ampla adoção industrial e suporte comunitário, característica destacada por Badamasi, Abiola e Jimoh (2018) ao analisar métricas de qualidade do Django REST Framework; (2) adequação para aplicações que integram inteligência artificial, aproveitando o ecossistema Python para processamento de dados e recomendações personalizadas, conforme enfatizado por Hadullo e Getuno (2021) em estudos sobre arquitetura de sistemas com IA; e (3) capacidade de desenvolvimento multiplataforma eficiente, permitindo construir interfaces nativas para Android e iOS a partir de uma única base de código, aspecto validado por Bittner *et al.* (2023) em avaliação sistemática do Flutter para aplicações sensíveis.

Esta abordagem arquitetural se alinha com práticas contemporâneas de desenvolvimento de software documentadas na literatura acadêmica, onde a separação clara entre *backend* e *frontend* facilita manutenibilidade, permite escalabilidade independente de cada componente e viabiliza implementação de testes automatizados em diferentes níveis da aplicação. As subseções seguintes detalham cada tecnologia utilizada, suas características técnicas e justificativas específicas para adoção no contexto do FitAI.

2.2.1 Django REST Framework

O Django REST Framework constitui-se como solução tecnológica estratégica para o *backend* do FitAI, oferecendo recursos avançados para construção de *APIs*

web robustas e escaláveis (ENCODE OSS LTD., 2024). A relevância desta escolha para aplicações *fitness* baseadas em IA generativa fundamenta-se em três pilares: compatibilidade nativa com o ecossistema Python que facilita integração com *APIs* de terceiros, arquitetura adequada para orquestração de múltiplas fontes de dados (perfis de usuários, históricos de treino, respostas de *APIs* externas), e infraestrutura robusta para gerenciamento seguro de dados sensíveis de usuários.

A arquitetura do Django REST Framework demonstra adequação específica para aplicações que integram *APIs* de inteligência artificial e requerem processamento estruturado de dados. Empresas como Spotify e YouTube utilizam Django para sistemas complexos que integram múltiplas fontes de dados e serviços externos, estabelecendo precedentes industriais para a abordagem adotada no FitAI. O Django *ORM* fornece abstração robusta para persistência de dados, suportando relacionamentos complexos necessários para armazenar perfis de usuários e preferências individuais que são utilizados para construção de *prompts* estruturados enviados à API Gemini.

No contexto específico do FitAI, três componentes do Django REST Framework são particularmente críticos. Os *Serializers* permitem transformação eficiente entre objetos Python e formatos *JSON*, essencial para comunicação entre o *backend* Django, a API do Google Gemini e o aplicativo Flutter. Esta transformação bidirecional garante que dados biométricos, preferências de treino sejam corretamente formatados para construção de *prompts* estruturados enviados à IA generativa e posteriormente deserializados para armazenamento persistente. As *ViewSets* fornecem estrutura organizacional para lógica de negócio relacionada a treinos, usuários e recomendações, encapsulando operações *CRUD* (*Create, Read, Update, Delete*) de forma consistente com padrões RESTful. Os *Routers* automatizam o mapeamento de URLs para *views*, reduzindo significativamente o código necessário para configuração de *endpoints* relacionados a perfis e treinos gerados.

A capacidade do Django REST Framework de orquestrar integrações com *APIs* externas têm sido documentada na literatura científica. Hadullo e Getuno (2021) enfatizam que o *framework* facilita a criação de pipelines de dados robustos,

essenciais para sistemas que processam informações de usuários e geram recomendações personalizadas através de serviços de IA. No FitAI, esta capacidade se manifesta na integração com a API do Google Gemini, onde o Django orquestra o fluxo de dados: recebe parâmetros do usuário via Flutter, enviando requisições HTTP à API Gemini (por exemplo, `requests.post(url, json=data)`), processa respostas *JSON* retornadas (`response.json()`), valida conteúdo gerado e persiste treinos no banco de dados SQLite utilizando o ORM do Django (`Workout.objects.create(validated_data)`).

O *framework* oferece suporte nativo para autenticação baseada em *tokens*, incluindo *TokenAuthentication* do *DRF* e *JWT (JSON Web Token)*, proporcionando segurança adequada para *endpoints* de API que processam dados sensíveis de usuários.

Pesquisa conduzida sobre métricas de qualidade de software do Django demonstrou que o *framework* principal possui complexidade ciclomática considerada média e qualidade de código satisfatória, conforme mensuração objetiva utilizando ferramentas como Radon e Pylint (BADAMASI; ABIOLA; JIMOH, 2018). Esta qualidade de código contribui para manutenibilidade do sistema, aspecto fundamental para evolução incremental da plataforma FitAI através de adição de novas funcionalidades como análise postural via computer vision ou integração com dispositivos *wearables*.

A arquitetura do Django também oferece vantagem estratégica para evolução futura do FitAI. A compatibilidade nativa com o ecossistema Python facilitará, quando necessário, a incorporação de bibliotecas de *machine learning* como *scikit-learn* para análise preditiva de lesões ou TensorFlow para modelos de recomendação mais sofisticados que complementem o sistema atual baseado em IA generativa. Esta extensibilidade foi considerada durante decisões arquiteturais iniciais, garantindo que o sistema possa evoluir organicamente conforme necessidades emergem sem requerer refatorações estruturais significativas.

2.2.2 Flutter

O Flutter constitui-se como solução estratégica para o *frontend* mobile do FitAI, permitindo desenvolvimento multiplataforma eficiente que alcança usuários Android e iOS a partir de única base de código. De acordo com a documentação oficial, o Flutter “transforma todo o processo de desenvolvimento de aplicações, permitindo construir, testar e distribuir aplicações móveis, web, *desktop* e embarcadas a partir de uma única base de código” (GOOGLE LLC, 2024). Esta característica é particularmente valiosa para aplicações *fitness* emergentes, onde recursos limitados de desenvolvimento devem ser otimizados para maximizar alcance de mercado.

A arquitetura do Flutter fundamenta-se em princípios técnicos que o distinguem de outras soluções de desenvolvimento cross-*platform*, com implicações diretas para performance de aplicações fitness. Diferentemente de *frameworks* como React Native e Xamarin, que dependem de pontes para comunicação com componentes nativos das plataformas, o Flutter possui seu próprio motor de renderização que produz diretamente dados de pixel na tela. Esta abordagem elimina etapas intermediárias de interpretação de código que podem diminuir a performance, tornando-o notavelmente mais eficiente em termos de processamento (VALOY *et al.*, 2023). Para o FitAI, esta eficiência se traduz em renderização fluida de listas extensas de exercícios e responsividade geral da interface que contribui para percepção positiva de qualidade do aplicativo.

Pesquisa conduzida sobre análise empírica de *frameworks* de desenvolvimento mobile cross-*platform* comparou Flutter, React Native e Kotlin através da implementação de aplicações idênticas. Kharb *et al.* (2022, p. 3) documentaram que “o uso de memória do Flutter foi aproximadamente 1,6% maior que o Kotlin nativo, enquanto o React Native apresentou 13,3% maior consumo”. Estes dados evidenciam que, embora soluções cross-*platform* apresentem *overhead* comparado a desenvolvimento nativo, o Flutter oferece desempenho superior entre alternativas multiplataforma, justificando sua escolha para o FitAI onde eficiência de

recursos é importante mas não crítica ao ponto de demandar desenvolvimento nativo separado para cada plataforma.

A linguagem de programação Dart, desenvolvida pelo Google e utilizada pelo Flutter, oferece suporte tanto para compilação *ahead-of-time* (AOT) quanto *just-in-time* (JIT). A compilação AOT permite que o código seja compilado diretamente para código nativo de baixo nível, resultando em aplicações que inicializam mais rapidamente e apresentam performance superior durante uso. A compilação JIT facilita a capacidade de *hot reload* do Flutter, que permite visualização de alterações em tempo real durante o desenvolvimento, reduzindo significativamente o tempo necessário para implementação e teste de modificações no código. Esta agilidade no ciclo de desenvolvimento foi fundamental para iterações rápidas na interface durante testes de usabilidade e refinamento da experiência do usuário.

A arquitetura baseada em *widgets* do Flutter oferece vantagens específicas para desenvolvimento de aplicações fitness que requerem interfaces dinâmicas e responsivas. O *framework* oferece dois conjuntos de *widgets* que se conformam a linguagens de design específicas: Material Design implementam a linguagem de design do Google, enquanto Cupertino implementam as diretrizes de interface humana do iOS da Apple. Esta flexibilidade permite que a consistência visual em ambas plataformas utilizando Material Design, simplificando manutenção e garantindo identidade visual uniforme, ou alternativamente adapte-se às convenções de cada sistema operacional caso pesquisas futuras de usabilidade indiquem preferência dos usuários por interfaces nativas.

A gestão de estado em Flutter, utilizando soluções como *Provider*, *Bloc* ou *Riverpod*, facilita a sincronização entre dados processados em *backend* Django e interfaces responsivas. Esta arquitetura é fundamental para manter consistência durante operações prolongadas de geração de treino, onde o usuário interage com o aplicativo enquanto aguarda resposta da API do Google Gemini processada pelo *backend*.

Estudo de avaliação conduzido por Bittner *et al.* (2023) utilizou o desenvolvimento de um protótipo de aplicação de aconselhamento oncológico para avaliar sistematicamente o Flutter baseado em catálogo de critérios estabelecido

academicamente. A pesquisa demonstrou que o Flutter atende adequadamente aos requisitos de desenvolvimento de aplicações médicas sensíveis, incluindo aspectos críticos como segurança de dados, interface de usuário responsiva e capacidade de integração com sistemas de *backend* complexos. Embora o FitAI não seja uma aplicação médica no sentido estrito, lida com dados sensíveis de saúde e condicionamento físico, tornando estas conclusões relevantes para validação da escolha tecnológica.

2.2.3 Integração Django-Flutter

A integração entre Django e Flutter constitui uma abordagem tecnologicamente robusta para o desenvolvimento do FitAI, combinando a eficiência de ambas as plataformas em uma arquitetura que maximiza as vantagens de cada *framework*. Nesta configuração, conforme estabelecido por Hadullo e Getuno (2021, p. 155), “o Django fornece os serviços de *backend* para autenticação, persistência de dados e lógica de negócio através de *APIs RESTful*”, enquanto o Flutter assume a responsabilidade pelo *frontend* de todo o sistema, gerenciando interface de usuário, experiência interativa e apresentação de dados.

Pesquisa acadêmica específica sobre a implementação de sistemas híbridos Django-Flutter documentou a criação de um Sistema de Gestão Acadêmica Virtual utilizando esta arquitetura integrada. Segundo os autores, o sistema:

usa uma aplicação móvel disponível tanto para Android quanto iOS para os estudantes, que é desenvolvida usando Flutter e uma aplicação web com recursos mais abrangentes, direcionada aos professores, que é desenvolvida usando o *framework* Django. (VIRTUAL ACADEMIC MANAGEMENT SYSTEM, 2021, p. 2321).

Esta implementação demonstra a viabilidade prática da integração para aplicações complexas que requerem funcionalidades diferenciadas para múltiplos tipos de usuários, estabelecendo precedente relevante para o FitAI, onde futura expansão poderia incluir portal web para profissionais de Educação Física gerenciarem múltiplos clientes.

A serialização de dados constitui um aspecto crítico da integração, onde o Django REST Framework utiliza *serializers* para converter objetos Django em formatos legíveis por *APIs*, principalmente *JSON*. O Flutter, por sua vez, implementa classes modelo correspondentes que deserializam estes dados JSON em objetos Dart nativos (HADULLO; GETUNO, 2021). Esta transformação bidirecional de dados deve ser otimizada para reduzir latência, especialmente em aplicações que processam grandes volumes de informações ou requerem atualizações frequentes.

A escalabilidade da arquitetura Django-Flutter pode ser otimizada através de técnicas específicas para cada componente. No lado Django, a implementação de cache para armazenar respostas frequentes (como perfis de usuário e treinos) reduz carga no banco de dados. No Flutter, a implementação de paginação em listas de treinos históricos e *lazy loading* de detalhes de exercícios reduz consumo de memória e melhora responsividade da interface. Estas otimizações, embora não críticas para o *MVP* do FitAI com base limitada de usuários, estabelecem fundação técnica para crescimento futuro da plataforma.

Protocolos de segurança específicos para comunicação Django-Flutter incluem a implementação obrigatória de *HTTPS/TLS* para todas as comunicações entre cliente e servidor, validação rigorosa de entrada de dados no lado servidor para prevenir ataques de injeção, e implementação de limitação de taxa (*rate limiting*) para prevenir ataques de força bruta ou sobrecarga de recursos. O FitAI implementa estas medidas, garantindo que dados sensíveis de usuários sejam protegidos durante transmissão e armazenamento, aspecto fundamental para estabelecer confiança dos usuários em aplicação que coleta informações pessoais de saúde e condicionamento físico.

2.3 Sistemas de Recomendação e Abordagens de Inteligência Artificial

Os sistemas de recomendação consolidaram-se como ferramentas fundamentais para a filtragem de informações em ambientes digitais. Segundo Gupta *et al.* (2022), algoritmos de Inteligência Artificial permitem a criação de programas de treinamento altamente personalizados ao processar variáveis como idade, sexo, histórico de lesões e objetivos, superando abordagens estáticas

tradicionais. No entanto, a escolha do algoritmo adequado depende intrinsecamente do volume de dados disponíveis e do estágio de maturação da plataforma.

A abordagem clássica estabelecida na literatura é a Filtragem Colaborativa (FC). Conforme Li *et al.* (2015), a FC prediz as preferências dos usuários baseando-se em padrões de avaliações de usuários similares (*user-based*) ou itens similares (*item-based*). Evuindo este conceito para capturar relações não-lineares complexas, He *et al.* (2017) introduziram o *Neural Collaborative Filtering* (NCF), substituindo a fatoração de matrizes tradicional por arquiteturas de redes neurais profundas (*Deep Learning*). Singh e Taneja (2024) expandiram ainda mais essa fronteira integrando Redes Neurais Recorrentes (RNNs) para analisar sequências temporais. Estas abordagens representam o estado da arte para plataformas consolidadas, como o *Nike Training Club*, que possuem milhões de pontos de dados históricos para treinamento.

Apesar da eficácia comprovada dos modelos colaborativos e neurais em grandes plataformas, eles apresentam limitações estruturais críticas para novas aplicações. Gupta *et al.* (2022) e Xu *et al.* (2019) identificam o problema do *cold start* (arranque a frio) e a *data sparsity* (esparsidade de dados) como barreiras primárias: novos usuários ou novas plataformas não possuem histórico de interações suficiente para que o algoritmo calcule similaridades com precisão. Em um cenário de *cold start*, métodos tradicionais de *Deep Learning* tendem a falhar ou fornecer recomendações genéricas baseadas apenas em popularidade até que um volume substancial de dados seja coletado.

Para contornar essa dependência de dados históricos, a literatura recente aponta para o uso de *Large Language Models* (LLMs) e IA Generativa como uma solução viável. Diferentemente da filtragem colaborativa, que aprende “quem gosta do quê” através de interações passadas, a IA Generativa possui conhecimento pré-treinado sobre domínios específicos, incluindo princípios de fisiologia e treinamento desportivo. Isso permite a implementação de sistemas de recomendação “Zero-Shot”, capazes de gerar personalização relevante desde a primeira interação, baseando-se apenas no perfil declarado do usuário (contexto). Esta fundamentação teórica justifica a decisão deste projeto em explorar arquiteturas baseadas em LLMs

para superar as barreiras de entrada típicas dos algoritmos tradicionais em aplicações emergentes.

2.4 Aplicações Práticas e Trabalhos Correlatos

A implementação de sistemas de recomendação baseados em inteligência artificial no domínio fitness tem sido objeto de crescente investigação, tanto na academia quanto na indústria. A literatura atual evidencia uma divisão clara entre propostas acadêmicas focadas em sensores avançados e soluções comerciais baseadas em grandes volumes de dados.

No âmbito da pesquisa acadêmica, destaca-se a integração com a Internet das Coisas (IoT) para inferência de contexto. Liu *et al.* (2023) desenvolveram o *P3FitRec*, um *framework* que utiliza *Deep Learning* para inferir características de fitness a partir de dados de dispositivos vestíveis (*wearables*), preservando a privacidade ao minimizar a coleta explícita de dados biométricos. Na mesma linha, Asgari Mehrabadi *et al.* (2024) propuseram o *framework PERFECT*, que também explora dados de IoT para dispensar inputs manuais sensíveis. Complementarmente, Farrokhi *et al.* (2021) enfatizam que a eficácia da personalização aumenta significativamente quando os sistemas são “conscientes de contexto” (CARS), incorporando variáveis dinâmicas detectadas por sensores em tempo real.

Em uma perspectiva comercial e aplicada, plataformas consolidadas adotam estratégias diferentes, valendo-se de bases massivas de usuários. O *Nike Training Club* (NTC) exemplifica o uso de Filtragem Colaborativa em escala. Khandelwal *et al.* (2020) documentam que a arquitetura da Nike analisa padrões de milhões de usuários para prever a conclusão de treinos com precisão superior a modelos aleatórios. Similarmente, o aplicativo *Fitbod*, conforme analisado por Zhang *et al.* (2024), implementa personalização adaptativa ajustando cargas baseadas em feedback histórico. No entanto, Zhang *et al.* (2024) alertam que a aceitação dessas tecnologias pelos usuários depende crucialmente da transparência e da clareza na apresentação dos dados, e não apenas da precisão matemática.

Paralelamente aos algoritmos de treino, observa-se a emergência de agentes conversacionais como ferramentas de suporte. Chew (2022) e Liu *et al.* (2020) demonstram que *chatbots* e sistemas de *eCoaching* desempenham papel fundamental na motivação e adesão contínua. No contexto brasileiro, Sichman *et al.* (2019) e Bezerra *et al.* (2020) ressaltam o potencial estratégico da IA para a saúde pública e privada, observando que o país possui competência técnica e um mercado crescente de saúde móvel para desenvolver soluções adaptadas à realidade local e às especificidades culturais da população.

A análise destes trabalhos permite situar a proposta do FitAI em um nicho distinto. Enquanto as soluções comerciais como a da Nike exigem *Big Data* para funcionar (Filtragem Colaborativa) e as acadêmicas como o P3FitRec dependem de hardware específico (*wearables*), o FitAI adota uma arquitetura baseada em Inteligência Artificial Generativa.

Quadro 1 – Comparativo de Aplicativos de Fitness

Aplicativo	Funcionalidades Principais	Limitações / Pontos a Melhorar	Diferencial / Observações
Nike Training Club	Planos variados, treinos guiados em vídeo, forte integração com ecossistema (Apple/Google).	Personalização baseada em Marca consolidada, grupos (<i>clusters</i>), alta qualidade de produção audiovisual e comunidade ativa. grande volume de dados históricos.	
Fitbod	Ajuste automático de carga e séries baseadas em fadiga e equipamento disponível.	Funcionalidades premium restritas a assinaturas; <i>Cold start</i> para novos usuários.	Algoritmo adaptativo focado em hipertrofia e progressão de carga.
Fitness AI	Otimização de volume/intensidade via <i>Machine Learning</i>	Foco restrito a métricas de levantamento de	Uso de <i>Deep Learning</i> para otimização

Aplicativo	Funcionalidades Principais	Limitações / Pontos a Melhorar	Diferencial / Observações
	tradicional.	peso; pouca explicação do “porquê” da sugestão.	matemática de séries e repetições.
FitAI	Personalização via IA Generativa, Chatbot contextual e explicação educativa dos treinos.	Validação empírica com usuários reais; <i>Deployment</i>	Arquitetura Híbrida Personalização imediata (<i>Zero-Shot</i>) Chabot (gerar treinos e conversa gerais)

Fonte: Elaborado pelos autores (2025).

2.5 Desafios e Oportunidades

Diante do exposto, considera-se que o desenvolvimento de plataformas *fitness* baseadas em inteligência artificial apresenta um panorama complexo de desafios técnicos, éticos e regulatórios, simultaneamente oferecendo oportunidades significativas para transformação do setor, como apontam Canzone *et al.* (2025). A análise crítica destes aspectos é fundamental para compreender as limitações atuais e o potencial futuro dessas tecnologias, tanto no contexto brasileiro quanto internacional.

2.5.1 Desafios Técnicos

Conforme fundamentado na seção 2.3, a escolha arquitetural de priorizar IA generativa em vez de sistemas colaborativos tradicionais respondeu às limitações estruturais de *cold start* e esparsidade de dados. Contudo, esta decisão introduz desafios técnicos específicos que transcendem os problemas evitados, exigindo soluções arquiteturais complementares.

Latência e experiência do usuário: A geração de planos de treino via IA generativa apresenta latência inerente significativamente superior a consultas tradicionais em banco de dados. Enquanto operações *CRUD* síncronas em Django respondem em dezenas de milissegundos, requisições ao Google Gemini apresentam tempo médio de 2 a 5 segundos em testes realizados no ambiente de desenvolvimento, variando conforme complexidade do *prompt* e carga nos servidores da API. Este desafio exige estratégias de UI não-bloqueante no Flutter, com indicadores visuais claros de progresso e arquitetura assíncrona que previne travamento da interface durante espera, seguindo recomendações de Nielsen (1994) sobre visibilidade do status do sistema como princípio fundamental de usabilidade.

Parsing e validação de respostas não estruturadas: Diferentemente de *APIs* tradicionais que retornam dados estruturados validados, modelos de linguagem generativos produzem texto livre que deve ser convertido em objetos estruturados para persistência em banco de dados. Conforme documentado na seção 4.6.2, o sistema implementa pipeline rigoroso de extração e validação: identificação de blocos *JSON* através de expressões regulares, limpeza de caracteres de controle e comentários inseridos pelo modelo, *parsing* com tratamento robusto de erros, e validação de campos obrigatórios e intervalos de valores seguros.

Gestão de custos e rate limiting: *APIs* de IA generativa operam sob modelo de precificação por requisição, introduzindo custos operacionais variáveis que podem tornar aplicação economicamente inviável em escala (LOOMI DIGITAL, 2025). O Google Gemini, no *tier* gratuito, limita requisições a 5 por minuto e 25 por dia (LAOZHANG-AI, 2025), restrições que impactam diretamente capacidade de atender múltiplos usuários simultâneos. Conforme destacado por Richardson e Ruby (2007) em análise de arquiteturas *RESTful* escaláveis, implementação de *rate limiting* no lado cliente é estratégia essencial para prevenir degradação de serviço. O sistema implementa controles através de contador em cache distribuído que rastreia requisições na janela temporal atual, direcionando automaticamente para *fallback* baseado em regras quando limites são atingidos.

Validação de segurança de recomendações geradas. A natureza generativa do sistema introduz desafio crítico: como garantir que treinos gerados

pela IA são fisiologicamente seguros e adequados às restrições declaradas pelo usuário? Diferentemente de sistemas que selecionam exercícios de biblioteca pré-aprovada por profissionais, o FitAI produz conteúdo original que deve ser validado programaticamente. Conforme demonstrado em estudos recentes, recomendações de exercício geradas por IA apresentaram lacunas notáveis em componentes críticos como frequência, intensidade, tempo e volume quando comparadas às diretrizes baseadas em evidências (ZALESKI *et al.*, 2024). O sistema implementa múltiplas camadas de validação: para garantir intervalos seguros, verificação de que exercícios contraindicados para restrições específicas não foram incluídos, e análise de balanceamento muscular para prevenir sobrecarga de grupos específicos. Contudo, reconhece-se que validação algorítmica não substitui supervisão humana de profissional certificado, pois tecnologias de IA, em seu estado atual, podem servir como ferramentas suplementares na prescrição de exercícios, mas ainda não são recomendadas como substituto para prescrições personalizadas fornecidas por profissionais de saúde e *fitness* (GERMINI *et al.*, 2024).

2.5.2 Desafios Não Técnicos

A implementação de aplicações *fitness* baseadas em IA enfrenta desafios regulatórios, éticos e jurídicos que transcendem questões puramente técnicas, exigindo atenção cuidadosa desde a concepção arquitetural.

Conformidade com LGPD: A Lei 13.709/2018 classifica informações de saúde como “dados pessoais sensíveis” (Art. 5º, II), exigindo consentimento explícito e medidas técnicas rigorosas. No contexto do FitAI, este desafio justifica a segregação arquitetural entre dados de identificação (Firebase) e dados *fitness* (Django).

Responsabilidade por recomendações de IA: A utilização de IA generativa para criação de planos de treino introduz questões complexas de responsabilidade legal em caso de lesões decorrentes de recomendações inadequadas. Conforme observado na literatura, definir o grau de controle é de importância primária ao discutir questões de responsabilidade: quando máquinas baseadas em algoritmos autoaprendizado alcançam diagnósticos sem necessidade de aprovação do clínico,

a responsabilidade deve ser ao menos parcialmente considerada do desenvolvedor da IA (CACCIAMANI *et al.*, 2023). Diferentemente de aplicações que agregam conteúdo pré-aprovado, sistemas baseados em IA generativa produzem conteúdo original em tempo real, tornando impraticável revisão humana prévia. A ausência de jurisprudência sobre responsabilidade de desenvolvedores versus provedores de modelos em contextos de saúde exacerba este desafio, uma vez que atualmente não existem regulamentações bem definidas para abordar as questões legais e éticas que podem surgir devido ao uso de inteligência artificial em ambientes de saúde (NAIK *et al.*, 2022). A estratégia de mitigação combina *disclaimers* explícitos, sistema de fallback baseado em regras validadas, e transparência algorítmica através de descrições.

Viés algorítmico: Aplicações baseadas em IA generativa pré-treinada (Google Gemini) herdaram vieses de corpora massivos da internet. Viés algorítmico em sistemas de saúde pode ser definido como as instâncias em que a aplicação de um algoritmo agrava iniquidades existentes em status socioeconômico, raça, origem étnica, religião, gênero, deficiência ou orientação sexual, amplificando-as e impactando adversamente as iniquidades em sistemas de saúde (PANCH; MATTIE; ATUN, 2019). No domínio *fitness*, estes vieses se manifestam através de preferência por exercícios anglófonos, associações inadequadas de gênero, e suposições etárias sobre capacidades físicas.

Limites legais da prescrição de exercícios: A prescrição de exercícios é atividade regulamentada (Lei 9.696/1998), privativa de profissionais registrados no *CONFEF*. Aplicações que geram planos automaticamente operam em zona cinzenta jurídica. O posicionamento do FitAI fundamenta-se em: ausência de avaliação física presencial e natureza educacional das recomendações que explicam princípios científicos. Reconhece-se que evolução jurisprudencial pode requerer ajustes, potencialmente incluindo parcerias formais com profissionais certificados.

2.5.3 Oportunidades de Mercado e Inovação

Apesar das limitações, as oportunidades oferecidas pela IA no setor *fitness* são substanciais. A integração da inteligência artificial permite personalizar programas de treinamento, otimizar análises de desempenho e aumentar a precisão na previsão de lesões através de algoritmos de aprendizado de máquina (REIS *et al.*, 2024), demonstrando potencial transformador para democratização do acesso a orientações especializadas e individualização em larga escala.

Além da democratização econômica discutida anteriormente na (seção 1.1), a capacidade de escalabilidade representa oportunidade significativa para democratização do acesso. Diferentemente do modelo tradicional de atendimento presencial, limitado por restrições geográficas e disponibilidade de profissionais qualificados, sistemas baseados em IA permitem oferecer orientação personalizada independentemente da localização do usuário (DUFFIN, 2024). Esta característica é particularmente relevante no contexto brasileiro, onde disparidades regionais na distribuição de profissionais de Educação Física criam desertos de atendimento especializado em regiões remotas. A arquitetura do FitAI, utilizando IA generativa combinada com regras heurísticas, viabiliza entrega de personalização de qualidade, estabelecendo fundação técnica para modelos escaláveis que podem atender milhares de usuários simultaneamente mantendo consistência e individualização.

Outro campo promissor é a prevenção de lesões e otimização de performance. Reis *et al.* (2024) ressaltam que “a IA pode beneficiar a prevenção de lesões e otimização de performance em esportes”, permitindo a transição de abordagens reativas para preditivas. Tecnologias como *pose estimation* e *computer vision*, exemplificadas pelo *AI Gym Trainer* que utiliza o *MediaPipe* para análise postural, demonstram a eficácia da correção em tempo real na redução de riscos e na melhoria do desempenho. Embora o escopo atual do FitAI não inclua análise de vídeo em tempo real, a arquitetura modular desenvolvida facilita incorporação futura desta funcionalidade através de integração com bibliotecas especializadas como *MediaPipe Pose* ou *OpenPose*, representando caminho natural de evolução para versões subsequentes da plataforma.

Além disso, iniciativas como a plataforma Alter.me apontam para o uso de dados genômicos na personalização, representando uma fronteira emergente para

recomendações individualizadas que consideram predisposições genéticas para desenvolvimento de força versus resistência, velocidade de recuperação e risco de lesões específicas.

No contexto brasileiro, o crescente interesse acadêmico documentado em bases como a SciELO revela um espaço fértil para pesquisas e desenvolvimento de soluções adaptadas à realidade local. A criação de modelos ajustados a características populacionais e culturais específicas representa uma oportunidade de liderança científica e tecnológica regional. O trabalho de Oliveira e Fraga (2021), situando criticamente aplicações de IA *fitness* no contexto da Educação Física brasileira, exemplifica esta tendência de pesquisa contextualizada que pode informar desenvolvimento de sistemas mais adequados às necessidades e valores da população nacional.

2.5.4 Implicações para o FitAI

A revisão da literatura e a análise dos desafios técnicos e éticos informam diretamente as decisões arquiteturais fundamentais do FitAI. Embora Li *et al.* (2015) e Xu *et al.* (2019) destaquem a eficácia da filtragem colaborativa em sistemas maduros, a aplicação dessa abordagem no FitAI enfrentaria a ausência de histórico comportamental prévio, visto que a plataforma, em sua fase inicial, não possui uma base histórica massiva de interações de usuários para treinar modelos preditivos tradicionais. Diante disso, adotamos uma estratégia diferenciada, priorizando a disponibilidade imediata de personalização através de Inteligência Artificial Generativa.

Essa escolha arquitetural responde diretamente às lacunas identificadas por Gupta *et al.* (2022) sobre a esparsidade de dados em novas aplicações. Ao utilizar *prompts* estruturados em vez de treinamento de modelos do zero, o FitAI consegue oferecer recomendações personalizadas desde o primeiro acesso do usuário, sem a necessidade de meses de coleta de dados. O sistema implementa uma abordagem híbrida: a IA Generativa atua como motor criativo e contextual, enquanto um

conjunto de regras especializadas funciona como um sistema de validação e *fallback*.

Além da estratégia algorítmica, as implicações éticas levantadas por Loannidou e Sklavos (2021) sobre privacidade em *wearables* e apps de saúde moldaram a infraestrutura de dados do projeto. O FitAI adota o princípio de *Privacy by Design* (Privacidade desde a Concepção) ao desacoplar a gestão de identidade (via Firebase Authentication) do armazenamento de dados sensíveis de saúde (via backend Django). Essa separação arquitetural minimiza a superfície de ataque e garante um controle granular sobre as informações mitigando riscos de vazamento de dados.

Por fim, é imperativo contextualizar o FitAI como uma implementação acadêmica de desenvolvimento tecnológico. As decisões tomadas priorizam a validação técnica da arquitetura híbrida (Django-Flutter) e a prova de conceito da integração com *LLMs*, aceitando limitações como a validação em ambiente controlado em detrimento de testes longitudinais com usuários reais nesta etapa. Essa transparência sobre o escopo alinha-se aos princípios de integridade científica, delimitando claramente as contribuições da arquitetura proposta frente as complexidades de um produto comercial em larga escala.

A revisão de literatura apresentada evidencia panorama complexo de desafios técnicos, éticos e regulatórios que orientam desenvolvimento responsável de sistemas fitness baseados em IA. As escolhas arquiteturais, tecnológicas e metodológicas documentadas nos capítulos seguintes respondem diretamente a este contexto, buscando equilibrar inovação com segurança, personalização com privacidade, e automação com transparência. O próximo capítulo detalha a metodologia adotada para operacionalizar estas considerações teóricas em protótipo funcional, descrevendo abordagem de pesquisa, ferramentas selecionadas e critérios de validação estabelecidos.

3 METODOLOGIA

Este capítulo detalha a metodologia utilizada para o desenvolvimento da plataforma FitAI, abrangendo o tipo de pesquisa, os procedimentos metodológicos adotados, as ferramentas e tecnologias empregadas, e os critérios de avaliação da solução proposta. O objetivo é apresentar de forma clara e estruturada o caminho percorrido para atingir os objetivos estabelecidos neste trabalho.

3.1 Tipo de pesquisa

Esta pesquisa é classificada como aplicada e de natureza experimental, adotando uma abordagem qualiquantitativa. Seu delineamento visa à produção de conhecimento prático, com potencial de aplicação direta, além de gerar contribuições científicas passíveis de mensuração e validação.

Finalidade: Configura-se como pesquisa aplicada que desenvolve um protótipo funcional para demonstrar a viabilidade de integração entre tecnologias emergentes. Enquanto soluções como o P3FitRec (LIU *et al.*, 2023) representam implementações em larga escala, este trabalho propõe uma prova de conceito que explora possibilidades de personalização através da combinação Django-Flutter, contribuindo para o entendimento das potencialidades e limitações desta arquitetura.

Objetivos: Desenvolver uma plataforma integrada de treinamento personalizado com suporte de inteligência artificial para gerar recomendações de exercícios adaptadas ao perfil individual dos usuários, garantindo alta performance técnica e experiência de usuário superior por meio de uma arquitetura híbrida Django-Flutter.

Abordagem: Emprega metodologia qualiquantitativa estrategicamente distribuída: análise quantitativa para métricas de performance, precisão algorítmica e escalabilidade; análise qualitativa para avaliação de experiência do usuário e adequação das recomendações ao contexto brasileiro de *fitness*.

Métodos: Utiliza desenvolvimento de protótipo orientado por evidências científicas, onde cada decisão técnica fundamenta-se em oportunidades

identificadas na literatura. Esta abordagem permite explorar efetivamente o potencial da arquitetura proposta, contribuindo com *insights* práticos para a área.

3.2 Procedimentos Metodológicos

Os procedimentos metodológicos adotados para o desenvolvimento do FitAI foram organizados em etapas sequenciais e iterativas, alinhadas ao processo de desenvolvimento de software. Para a condução do projeto, optou-se pelo uso de práticas oriundas do *framework* ágil *Scrum*, devido à sua flexibilidade e adequação a projetos acadêmicos que demandam entregas incrementais e ajustes contínuos ao longo do ciclo de desenvolvimento.

As fases metodológicas previstas são descritas a seguir:

Levantamento de Requisitos: Esta fase teve como objetivo identificar e documentar os requisitos funcionais e não funcionais da plataforma, seguindo abordagem de engenharia de requisitos estabelecida por Sommerville (2019). Foram analisadas soluções similares já existentes, a fim de compreender suas funcionalidades principais e extrair aspectos relevantes para adaptação ao escopo do projeto. Além disso, foram considerados aspectos técnicos relacionados à escalabilidade, usabilidade e segurança dos dados do usuário.

Design de Arquitetura: Após a consolidação dos requisitos, definiu-se a arquitetura do sistema, contemplando a organização em camadas (*frontend*, *backend* e banco de dados). Nesta fase, também foram estabelecidas as tecnologias utilizadas, tais como *Django REST Framework* para a camada de serviços, *Flutter* para o desenvolvimento do aplicativo móvel e PostgreSQL para persistência de dados. O modelo arquitetural seguiu princípios de modularidade e baixo acoplamento, visando facilitar a manutenção e evolução da plataforma.

Desenvolvimento: O processo de implementação foi realizado de forma incremental, seguindo ciclos curtos inspirados nas *sprints* do *Scrum*. A cada ciclo, funcionalidades específicas foram priorizadas, desenvolvidas e integradas ao sistema. Essa abordagem permitiu acompanhar a evolução do projeto, garantindo maior controle sobre prazos e entregas.

Plano de Testes: Para garantir a qualidade, confiabilidade e desempenho da plataforma, foi definido um plano de testes abrangente:

- **Testes de Integração**

- **Objetivo:** Avaliar a comunicação correta entre os módulos (API Django ↔ Flutter).
- **Ferramentas:** Postman, Insomnia, Jest.
- **Casos de Teste:** Consumo de *endpoints* de treino, cadastro de usuário, login e persistência de dados.
- **Métricas:** Percentual de sucesso das chamadas, tempo médio de resposta.

- **Testes de Sistema**

- **Objetivo:** Validar as funcionalidades globais e a experiência do usuário.
- **Ferramentas:** Testes manuais com roteiro de casos de uso.
- **Casos de Teste:** Cadastro, login, criação de plano de treino, monitoramento de progresso.
- **Métricas:** Taxa de sucesso em cenários de uso simulados, taxa de erros críticos identificados em ambiente controlado.

- **Testes de Usabilidade**

- **Objetivo:** Avaliar a adequação da interface a princípios estabelecidos de design.
- **Método:** Análise heurística baseada nos dez princípios de usabilidade de Nielsen.
- **Métricas:** Identificação de violações de princípios de usabilidade, severidade dos problemas encontrados, conformidade com padrões WCAG 2.1 de acessibilidade.

- **Testes de Algoritmos de IA**

- **Objetivo:** Avaliar a eficácia dos algoritmos de personalização de treino.

- **Método:** Comparação entre planos sugeridos pelo sistema e planos propostos por especialistas.
- **Métricas:** Grau de adequação (% de concordância com especialistas), evolução de resultados no usuário.

Implantação: Após a validação do sistema em ambiente de testes, procedeu-se à implantação em ambiente controlado, com foco em verificar o desempenho e a estabilidade da aplicação. Essa etapa contemplou também ajustes relacionados à configuração do servidor e do banco de dados.

Validação: Por fim, foi conduzida a etapa de validação, onde o sistema foi confrontado com os objetivos iniciais do projeto. A validação permitiu confirmar se a plataforma atendia às necessidades estabelecidas, garantindo alinhamento entre os resultados obtidos e as expectativas definidas na fase de requisitos.

A adoção de um processo iterativo, aliado ao uso de práticas do *Scrum*, contribuiu para que o desenvolvimento fosse conduzido de forma organizada, permitindo o refinamento contínuo do sistema e assegurando que os requisitos essenciais fossem plenamente atendidos.

Com base no processo de levantamento de requisitos descrito, foram identificados e formalizados os requisitos funcionais e não funcionais que orientaram todo o desenvolvimento da plataforma, conforme detalhado na seção seguinte.

3.3 Especificação de Requisitos

Esta seção apresenta a especificação formal dos requisitos identificados durante a fase de levantamento, organizados em requisitos funcionais (RF), que descrevem as funcionalidades que o sistema deve oferecer, e requisitos não funcionais (RNF), que estabelecem restrições e qualidades esperadas do sistema em termos de desempenho, segurança, usabilidade e outros atributos de qualidade.

Quadro 1 – Requisitos Funcionais

ID	Requisito
RF01	O sistema deve permitir cadastro de usuários
RF02	O sistema deve coletar informações de perfil <i>fitness</i> durante <i>onboarding</i>
RF03	O sistema deve gerar planos de treino personalizados utilizando IA generativa (Google Gemini) baseados no perfil do usuário
RF04	O sistema deve permitir visualização detalhada de treinos com exercícios, séries, repetições, carga, tempo de descanso e instruções
RF05	O sistema deve permitir registro de sessões de treino executadas
RF06	O sistema deve gerar recomendações diárias inteligentes considerando histórico recente, necessidade de descanso e distribuição de grupos musculares
RF07	O sistema deve fornecer <i>chatbot</i> conversacional com IA para orientações contextualizadas sobre treinos, motivação e esclarecimento de dúvidas e geração de treinos
RF08	O sistema deve exibir <i>dashboard</i> com métricas de progresso
RF09	O sistema deve permitir busca e filtragem de exercícios por grupo muscular, nível de dificuldade e equipamento necessário

Fonte: Elaborado pelos autores (2025).

Quadro 2 – Requisitos Não-Funcionais

ID	Categoria	Requisito
RNF01	Usabilidade	A interface deve seguir diretrizes Material Design 3 e ser responsiva para diferentes tamanhos de tela (smartphones e tablets)
RNF02	Usabilidade	O aplicativo deve manter consistência visual e de interação

ID	Categoria	Requisito
		entre plataformas Android e iOS
RNF03	Usabilidade	A interface de execução de treino deve ter elementos ampliado e alto-contraste para facilitar visualização durante atividade física
RNF04	Performance	<i>Endpoints</i> de API síncronos (operações <i>CRUD</i> em banco de dados) devem responder em menos de 2 segundos sob condições normais
RNF05	Segurança	O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD) para tratamento de dados pessoais sensíveis de saúde
RNF06	Segurança	Utilizar protocolo HTTPS/TLS 1.2 ou superior
RNF07	Escalabilidade	Suporte de migração de banco de dados e <i>deploy</i>
RNF08	Interoperabilidade	O sistema deve integrar-se com Firebase Authentication para gerenciamento de identidades e Google Gemini API para geração de conteúdo

Fonte: Elaborado pelos autores (2025).

3.4 Ferramentas e tecnologias utilizadas

Para o desenvolvimento da plataforma FitAI, foram selecionadas ferramentas e tecnologias que garantem escalabilidade, segurança e uma experiência de usuário fluida. A seguir, são apresentadas as principais escolhas tecnológicas e suas justificativas:

Backend:

- **Django REST Framework:** *Framework* robusto e maduro para construção de *APIs RESTful* em Python. Foi escolhido por sua documentação extensa,

sistema de autenticação integrado, suporte a serialização de dados complexos e capacidade de escalar conforme o crescimento da plataforma.

- **Python:** Linguagem de programação de alto nível, escolhida por sua sintaxe clara, ampla adoção na comunidade de desenvolvimento web e excelente suporte para integração com APIs de IA e bibliotecas de processamento de dados.
- **SQLite:** Banco de dados relacional leve e embutido, adequado para a fase de desenvolvimento e prototipagem. Oferece simplicidade de configuração sem necessidade de servidor separado, facilitando testes e iterações rápidas. Para produção, pode ser migrado para PostgreSQL ou MySQL conforme necessidade de escalabilidade.

Frontend (Aplicativo Móvel):

- **Flutter:** *Framework* multiplataforma do Google que permite desenvolver aplicativos nativos para Android e iOS a partir de uma única base de código. Sua escolha se justifica pela redução de tempo de desenvolvimento, consistência visual entre plataformas.
- **Dart:** Linguagem de programação orientada a objetos utilizada pelo Flutter. Oferece compilação AOT (*Ahead-of-Time*) para melhor performance e *hot reload* para desenvolvimento ágil.

Inteligência Artificial:

- **APIs de IA Generativa:** Integração com serviços de IA para geração de conteúdo personalizado, incluindo sugestões de treinos adaptadas ao perfil do usuário e respostas a dúvidas sobre exercícios.
- **Engenharia de Prompts:** Desenvolvimento de *prompts* estruturados e contextualizados para otimizar a qualidade, relevância e segurança das respostas geradas pela IA.

Autenticação e Segurança:

- **Firebase Authentication:** Serviço gerenciado do Google para autenticação de usuários, autenticação por e-mail/senha. Reduz a complexidade de implementar segurança de autenticação do zero.
- **JWT (JSON Web Tokens):** *Tokens* personalizados gerados pelo backend Django após validação dos *tokens* Firebase. Permite controle granular de permissões, gerenciamento de sessões e autorização em *endpoints* específicos da API.
- **Flutter Secure Storage:** *Plugin* para armazenamento criptografado de dados sensíveis no dispositivo móvel, utilizando *Keychain* no iOS e *KeyStore* no Android, garantindo proteção contra acesso não autorizado.

Ferramentas de Desenvolvimento:

- **Git/GitHub:** Sistema de controle de versão distribuído e plataforma de hospedagem de código. Facilita o versionamento do projeto, colaboração entre desenvolvedores, revisão de código via *pull requests*.
- **Visual Studio Code (VSCode):** Editor de código leve e extensível, utilizado como ambiente de desenvolvimento integrado (IDE).
- **Figma:** Plataforma colaborativa de design de interfaces (UI/UX), utilizada para criar wireframes, protótipos interativos de alta-fidelidade e fluxos de navegação. Permite validar a experiência do usuário antes da implementação e facilita a comunicação entre design e desenvolvimento.

3.5 Critérios de avaliação

A avaliação da plataforma FitAI foi realizada com base nos seguintes critérios, visando analisar o impacto da personalização por IA na experiência do usuário final e a viabilidade técnica da solução:

Performance do Sistema:

- **Testes de Performance:** Foram realizados para avaliar a responsividade e a escalabilidade da plataforma.
- **Tempo de Resposta:** Medição do tempo de resposta das APIs do *backend* e do carregamento das telas do aplicativo móvel.
- **Consumo de Recursos:** Monitoramento do uso de CPU, memória e rede pelo *backend* e aplicativo.

Usabilidade e Experiência do Usuário (UX):

- **Avaliação de Usabilidade:** Análise heurística baseada nos dez princípios de Nielsen para identificação sistemática de problemas potenciais de usabilidade através de inspeção própria, método consolidado em engenharia de usabilidade que oferece *insights* valiosos sobre adequação da interface a princípios estabelecidos de design de interação.
- **Facilidade de Uso:** Avaliação da clareza de fluxos de navegação, consistência de elementos de interface e aderência a padrões de design Material Design 3.
- **Conformidade com Padrões:** Verificação de requisitos básicos de acessibilidade conforme padrões WCAG 2.1, incluindo contraste de cores, escalabilidade de tipografia e dimensionamento de áreas de toque.

Eficácia dos Algoritmos de Recomendação:

- **Análise da Eficácia dos Algoritmos de Recomendação:** Será conduzida para verificar a relevância e adequação dos treinos personalizados.
- **Relevância das Recomendações:** Análise da adequação dos treinos personalizados gerados pelos algoritmos de IA ao perfil e progresso individual dos usuários.
- **Gestão de Custos:** Monitoramento de controles de *rate limiting* e custos associados a APIs de IA generativa.

Segurança e Proteção de Dados:

- **Privacidade dos Dados:** Verificação da conformidade com as boas práticas de proteção de dados pessoais dos usuários.
- **Autenticação e Autorização:** Avaliação da robustez dos mecanismos de segurança para acesso à plataforma e dados sensíveis.

Qualidade do Código e Manutenibilidade:

- **Padrões de Codificação:** Avaliação da conformidade do código com as melhores práticas de desenvolvimento Django e Flutter.
- **Documentação:** Verificação da clareza e abrangência da documentação técnica.
- **Modularidade:** Análise da organização do código em módulos independentes para facilitar a manutenção e futuras expansões.

Estabelecida a fundamentação metodológica da pesquisa, com definição dos requisitos técnicos e funcionais, ferramentas e critérios de avaliação, o capítulo seguinte apresenta o desenvolvimento e implementação da plataforma FitAI. São detalhados os aspectos arquiteturais, a modelagem de dados, os algoritmos de inteligência artificial e o design das interfaces, demonstrando como as decisões metodológicas foram operacionalizadas em soluções técnicas concretas.

4. DESENVOLVIMENTO E IMPLEMENTAÇÃO DO SISTEMA FITAI

Este capítulo apresenta o desenvolvimento técnico da plataforma FitAI, detalhando sua arquitetura, modelagem de dados e a integração dos algoritmos de inteligência artificial. A estruturação segue princípios de engenharia de software consolidados, visando assegurar escalabilidade, manutenibilidade e alta performance para aplicações de *fitness* personalizadas.

4.1 Arquitetura da Plataforma FitAI

A arquitetura do FitAI foi projetada para oferecer modularidade e alta disponibilidade, alinhando-se às diretrizes de Richards e Ford (2020). A plataforma adota uma arquitetura em três camadas (*3-tier architecture*), isolando apresentação, lógica de negócio e persistência de dados. Conforme documentado por Hadullo e Getuno (2021), esta separação é crítica para sistemas que acoplam inteligência artificial a aplicações web, pois permite a otimização independente de cada componente.

O sistema se organiza em cinco camadas principais, cada uma com responsabilidades estritas:

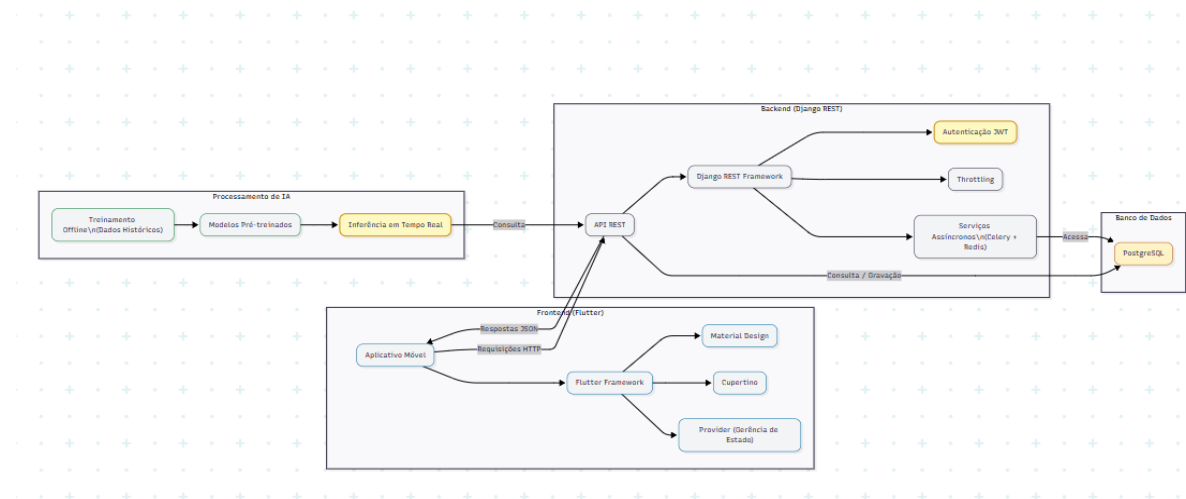
- 1. Apresentação:** Aplicativo móvel em Flutter.
- 2. Lógica de Negócio:** API desenvolvida com Django REST Framework.
- 3. Inteligência Artificial:** Camada de processamento via Google Gemini API.
- 4. Identidade:** Autenticação via Firebase Authentication com integração *JWT*.
- 5. Persistência:** Banco de dados relacional para ambiente de produção.

Esta estrutura obedece aos princípios de arquitetura em camadas, onde interfaces bem definidas (APIs) permitem a evolução tecnológica de um módulo sem impactos colaterais no restante do ecossistema.

Para ilustrar essa organização, o Diagrama de Arquitetura (Figura 1) evidencia a interação entre *frontend*, *backend*, serviços assíncronos e banco de dados. Essa representação macro é fundamental para compreender as dependências tecnológicas (PRESSMAN; MAXIM, 2021) e demonstra como o

projeto atende aos requisitos de modularidade e escalabilidade. A versão detalhada encontra-se no Apêndice A (Figura A.1).

Figura 1: Diagrama de arquitetura



Fonte: Elaborado pelos autores (2025)

4.1.1 Camadas de Aplicação e Inteligência Artificial

A camada de apresentação utiliza Flutter como *framework* principal, viabilizando uma experiência nativa multiplataforma. Heimer e Borba (2022) destacam que a compilação do Flutter para código nativo oferece desempenho superior a soluções baseadas em *webviews*, requisito essencial para a visualização fluida de dados de treino em tempo real. O design adota os padrões Material Design 3 e Cupertino, assegurando consistência visual e aderência às diretrizes do Google (GOOGLE LLC, 2024) e Apple (APPLE INC., 2024). Internamente, a gestão de estado da aplicação é orquestrada através do padrão *Provider*, otimizando a renderização de componentes e o consumo de recursos do dispositivo.

No *backend*, a lógica de negócio é centralizada no Django REST Framework (DRF). A escolha baseia-se na robustez e na capacidade de escalabilidade verticais documentadas por Badamasi, Abiola e Jimoh (2018). Nesta camada, a segurança é tratada como prioridade: todos os *endpoints* críticos implementam *throttling* (limitação de taxas) para mitigar ataques de força bruta e negação de serviço. A

comunicação é estritamente *stateless*, alinhada aos princípios REST, o que facilita o escalonamento horizontal da infraestrutura conforme o crescimento da base de usuários.

O diferencial do FitAI reside na integração profunda com a API Google Gemini. A arquitetura prioriza a IA generativa não apenas como um chatbot, mas como um motor de decisão estruturado.

- **Engenharia de Prompts Dinâmica:** O sistema não utiliza *prompts* estáticos. Um módulo construtor injeta, em tempo de execução, o perfil biométrico do usuário (peso, altura, histórico de lesões) e seus objetivos dentro de um “Contexto de Sistema” pré-definido. Isso assegura que a IA atue como um *personal trainer* especialista, e não como um modelo genérico.

- **Saída Estruturada (JSON Enforcement):** Para garantir que o aplicativo Flutter possa renderizar os treinos visualmente (com ícones, repetições e cronômetros), instrui-se o modelo a retornar respostas estritamente em formato *JSON*. O backend valida esse esquema antes de entregá-lo ao *frontend*, prevenindo erros de *parsing*.

- **Arquitetura Híbrida:** O sistema conta com um mecanismo de *fallback* baseado em regras heurísticas. Caso a API de IA esteja indisponível ou retorne dados fora do padrão, o sistema entrega um treino padronizado, garantindo que o usuário nunca fique sem atendimento.

4.1.2 Segurança, Identidade e Persistência de Dados

A arquitetura de segurança implementa uma estratégia híbrida de gestão de identidade, onde o Firebase Authentication atua como Provedor de Identidade (*Identity Provider*), enquanto o Django gerencia a autorização e os dados de domínio. O fluxo automatizado de sincronização garante a consistência e segurança através das seguintes etapas:

- **Flutter → Firebase Auth:** O usuário autentica-se na interface móvel e o Firebase retorna um *ID Token* assinado criptograficamente.

- **Flutter → Django:** O *token* é enviado ao *backend* no cabeçalho da requisição.
- **Validação no Backend:** O Django verifica a assinatura do *token* com as chaves públicas do Google, validando a autenticidade sem *overhead* de rede adicional.
- **Persistência e Sincronização:** O sistema executa uma operação atômica de *get_or_create* baseada no *UID* único do usuário. Se for o primeiro acesso, um perfil local é criado; se recorrente, os dados são recuperados.
- **Emissão de JWT Proprietário:** Após a validação, o *backend* emite seu próprio *JSON Web Token* (JWT) contendo *claims* e permissões específicas da aplicação.

Para suportar essa integração, o modelo de dados adota estratégias de defesa em profundidade. O *firebase_uid* atua como chave mestra imutável, permitindo a alteração de e-mails sem perda de histórico. Adicionalmente, o campo de senha no banco de dados local é definido como *unusable*, assegurando que a autenticação ocorra exclusivamente via Firebase e eliminando vetores de ataque direto, em conformidade com as recomendações da OWASP.

Em relação à consistência de dados, e em consonância com os estudos de Vogels (2009) sobre consistência eventual, o sistema adota estratégias robustas para mitigar divergências. O maior desafio técnico abordado foi a concorrência em cenários de login simultâneo, onde poderiam ocorrer condições de corrida (*race conditions*). Para evitar duplicação de registros, implementou-se uma restrição *unique* no nível do banco de dados PostgreSQL para o campo *firebase_uid*. O *backend* trata exceções de integridade (*IntegrityError*) de forma atômica: se duas requisições tentam criar o usuário ao mesmo tempo, a segunda falha graciosamente e recupera o registro já criado, garantindo comportamento idempotente.

No que tange à proteção de dados e LGPD, os *tokens* são armazenados utilizando *FlutterSecureStorage*, que implementa *KeyChain* (iOS) e *KeyStore*

(Android), delegando a criptografia ao *hardware* do dispositivo. O sistema também implementa a rotação de *tokens* conforme a RFC 8252 (*OAuth 2.0 Security Best Current Practice*), reduzindo a janela de oportunidade em caso de comprometimento de sessão.

Por fim, a camada de monitoramento dispõe de *logging* estruturado para eventos críticos de segurança. O Firebase Console fornece telemetria em tempo real sobre tentativas de acesso para detecção de anomalias geográficas ou padrões de força bruta. Os *logs* são estruturados em JSON para facilitar a ingestão futura por ferramentas de SIEM (*Security Information and Event Management*).

4.2 Fluxo de Dados

O fluxo de dados na plataforma FitAI segue o padrão *request-response* para operações síncronas e utiliza processamento assíncrono para tarefas computacionalmente intensivas. Tal abordagem híbrida, validada por Liu *et al.* (2023) no projeto P3FitRec, é essencial para manter responsividade em sistemas que processam dados e geram recomendações personalizadas.

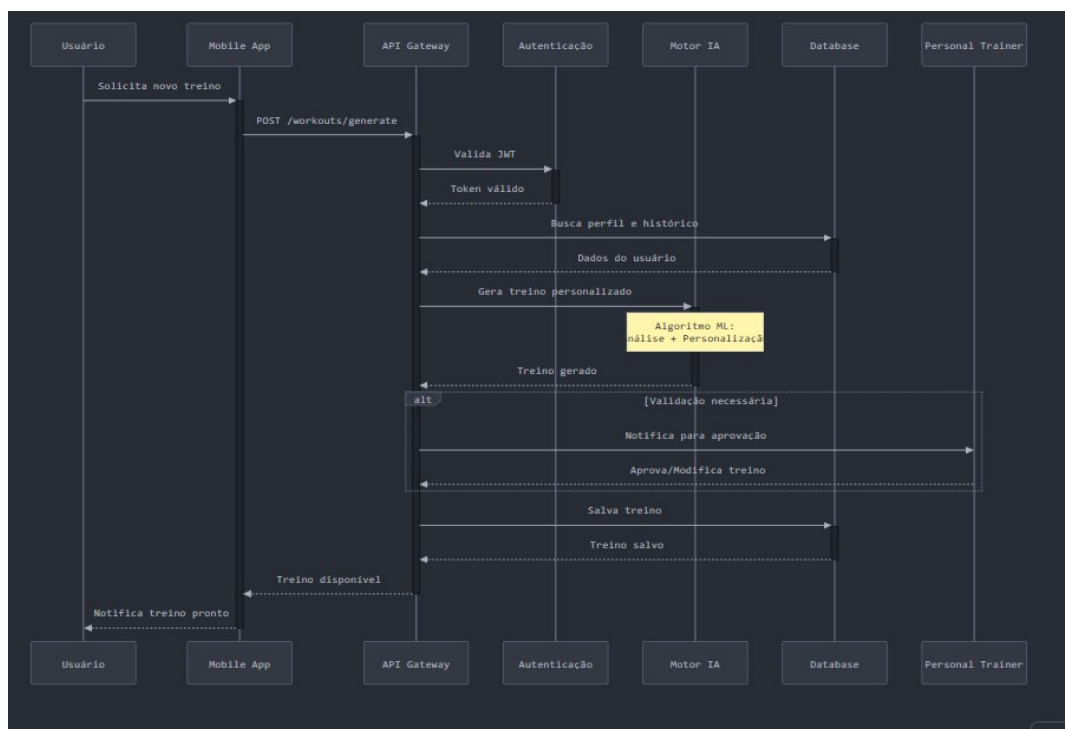
O diagrama de sequência representa a interação temporal entre objetos e componentes do sistema, evidenciando o fluxo de mensagens e a coordenação entre diferentes camadas arquiteturais durante a execução de cenários específicos.

Conforme destacado no Handbook of Software Engineering Methods (2024), estes diagramas são fundamentais para criar programas sustentáveis e extensíveis que resolvem problemas relevantes, sendo especialmente úteis para visualizar interações complexas em sistemas distribuídos. Para o sistema FitAI, esta modelagem é fundamental devido à complexidade das interações entre *frontend*, *backend*, processamento de inteligência artificial e validação por *personal trainers*. O cenário modelado, “solicitação e processamento de treino personalizado”, demonstra um dos fluxos mais críticos da plataforma, envolvendo autenticação, coleta de dados, processamento por algoritmos de ML e entrega de recomendações personalizadas, evidenciando a integração entre componentes automatizados e

supervisão humana característica de sistemas de recomendação *fitness* responsáveis.

A versão ampliada do diagrama, com todos os componentes detalhados, encontra-se no Apêndice A (Figura A.2).

Figura 2: Diagrama de Sequência



Fonte: Elaborado pelos autores (2025)

4.3 Modelagem do Banco de Dados

A modelagem de banco de dados constitui uma etapa fundamental no desenvolvimento de sistemas de informação, servindo como ponte entre os requisitos de negócio e a implementação física dos dados.

Segundo Atlan (2024), estabelece os projetos para gerenciamento de dados definindo entidades, atributos, relacionamentos, restrições e regras. Embora abordagens mais novas como *knowledge graphs* sejam promissoras, formas tradicionais de modelagem como ER e dimensional permanecem relevantes na maioria dos cenários do mundo real.

O modelo entidade-relacionamento (ER) continua sendo amplamente utilizado na concepção de sistemas de informação contemporâneos. De acordo com a IBM (2025), os diagramas de relacionamento entre entidades (ERDs) constituem

A versão ampliada do diagrama ER, detalhando todos os atributos e cardinalidades, encontra-se no Apêndice A (Figura A.3). A estrutura de dados do FitAI foi organizada em quatro macro-grupos lógicos: Identidade, Catálogo de Treinos, Execução e Inteligência.

```

classDiagram
    class UserProfile {
        id PK
        user_id FK -> User
        bio: descrição pessoal
        goal: objetivo fitness
        activity_level: nível de atividade
        fitness_areas: áreas de foco específicas
        current_weight: peso atual em kg
        target_weight: peso alvo em kg
        + registerProgress()
        + permitLogin()
    }
    class User {
        id PK
        identification: número de documento
        username: nome de usuário (único)
        email: endereço de e-mail (único)
        full_name: nome completo
        last_login: data da última vez que se logou
        last_password_change: data da última vez que a senha foi alterada
        + validateEmail()
        + validateUsername()
    }
    class UserProgress {
        id PK
        user_id FK -> User
        total_exercises: contagem
        fitness_metrics:
        + registerProgress()
    }
    class Workout {
        id PK
        exercise_id FK -> Exercise
        user_id FK -> User
        set_number: número de séries
        reps: número de repetições por série
        weight: peso usado em kg
        duration: duração em minutos
        + registerWorkout()
        + updateReps()
        + updateWeight()
    }
    class Exercise {
        id PK
        name: nome do exercício
        description: descrição
        difficulty: nível de dificuldade
        muscle_group: grupo muscular
        difficulty_level: nível de dificuldade
        equipment: equipamentos necessários
        duration: duração em minutos
        duration_min: duração mínima
        + startWorkout()
        + calculateCalories()
    }
    class NotificationLog {
        id PK
        user_id FK -> User
        title: título da notificação
        message: conteúdo da notificação
        status: status
        notification_type: tipo de notificação
        priority: prioridade
        created_at: data de criação
        + registerNotification()
    }
    class NotificationPreference {
        id PK
        user_id FK -> User
        notification_type: tipo de notificação
        enabled_status: status habilitado
        preferred_times: horários preferidos
        frequency: frequência
        last_checked: data da última verificação
        + registerNotification()
        + updateNotification()
    }
    class UserNotificationData {
        id PK
        user_id FK -> User
        last_delivery: última entrega
        notification_status: status da notificação
        last_clicked: data do clique
        last_seen: data vista
        notification_preferences: preferências
        + registerNotification()
        + updateNotification()
    }
    class WorkoutSession {
        id PK
        workout_id FK -> Workout
        user_id FK -> User
        completed_status: status concluído
        start_at: hora de início
        completed_at: hora de conclusão
        duration: duração
        user_rating: avaliação do usuário
        notes: observações
        created_at: data de criação
        + startSession()
        + cancelSession()
        + updateSession()
    }
    class Recommendation {
        id PK
        user_id FK -> User
        exercise_id FK -> Exercise
        algorithm: algoritmo
        confidence: confiança
        recommendation_text: texto da recomendação
        created_at: data de criação
        + registerRecommendation()
        + updateRecommendation()
    }
    class ExerciseLog {
        id PK
        exercise_id FK -> Exercise
        user_id FK -> User
        workout_id FK -> Workout
        set_number: número da série
        reps: repetições
        weight: peso
        duration: duração
        + registerLog()
        + updateLog()
    }
    UserProfile "1" -- "1" User
    User "1" -- "1" UserProgress
    UserProgress "1" -- "1" Workout
    Workout "1" -- "1" Exercise
    Exercise "1" -- "1" WorkoutSession
    WorkoutSession "1" -- "1" Recommendation
    Recommendation "1" -- "1" ExerciseLog
    NotificationLog "1" -- "1" User
    NotificationPreference "1" -- "1" User
    UserNotificationData "1" -- "1" User
    
```

Entidades Principais do Sistema: O sistema adota um modelo centrado no usuário para garantir integridade referencial. A entidade *User* compõe a base do sistema de autenticação, armazenando credenciais e dados de acesso (*username*, *email*, *date_joined*, *is_active*). Vinculada a ela, a entidade *UserProfile* estende os dados com informações específicas do domínio *fitness*, incluindo objetivos (*goal*), nível de atividade (*activity_level*), áreas de foco muscular e métricas corporais (*current_weight*, *target_weight*). Para monitoramento longitudinal, a entidade *UserProgress* consolida o histórico, mantendo contadores como *total_workouts*, com estrutura preparada para métricas adicionais futuras.

Catálogo de Exercícios e Treinos: A gestão de conteúdo é realizada através da entidade *Exercise*, que atua como um catálogo descritivo contendo atributos como *muscle_group*, *difficulty_level*, *equipment_needed*, calorias por minuto e instruções multimídias (*video_url*). Estes exercícios são agrupados na entidade *Workout*, que define a estrutura macro do treino (*target_muscle_groups*, *estimated_duration*). A relação entre exercícios e treinos é gerenciada pela entidade associativa *WorkoutExercise* (relacionamento N: N), que armazena a prescrição técnica específica daquele treino, como número de séries (*sets*), repetições (*reps*), carga sugerida (*weight*) e ordem de execução (*order_in_workout*).

Sessões de Treino e Telemetria: Para capturar a execução real dos usuários, a entidade *WorkoutSession* registra os dados temporais (*started_at*, *completed_at*) e métricas consolidadas (*calories_burned*), além de permitir *feedback* qualitativo através de *user_rating* e notas. Em um nível de granularidade maior, a entidade *ExerciseLog* armazena o detalhe de cada exercício realizado dentro da sessão, permitindo a comparação entre o planejado e o executado (*sets_completed*, *reps_completed*, *weight_used*), dados essenciais para o ajuste de carga progressiva.

Sistema de Recomendação: Por fim, a inteligência do sistema é materializada na entidade *Recommendation*. Esta estrutura suporta múltiplos algoritmos de IA, armazenando não apenas a sugestão gerada, mas também metadados de explicabilidade, como o *score_confianca* e o *motivo_recomendacao*. Essa abordagem permite transparência no processo de decisão da IA e auditoria da qualidade das sugestões entregues ao usuário.

4.4 Casos de Uso

Os casos de uso representam os principais cenários de interação entre usuários e sistema, destacando os papéis de Usuário e *Personal Trainer*. Entre os casos de uso mais relevantes estão:

- **Cadastro e Login:** autenticação segura utilizando *JSON Web Token* (JWT);
- **Gerenciar Perfil e Preferências:** definição de objetivos, restrições e disponibilidade;

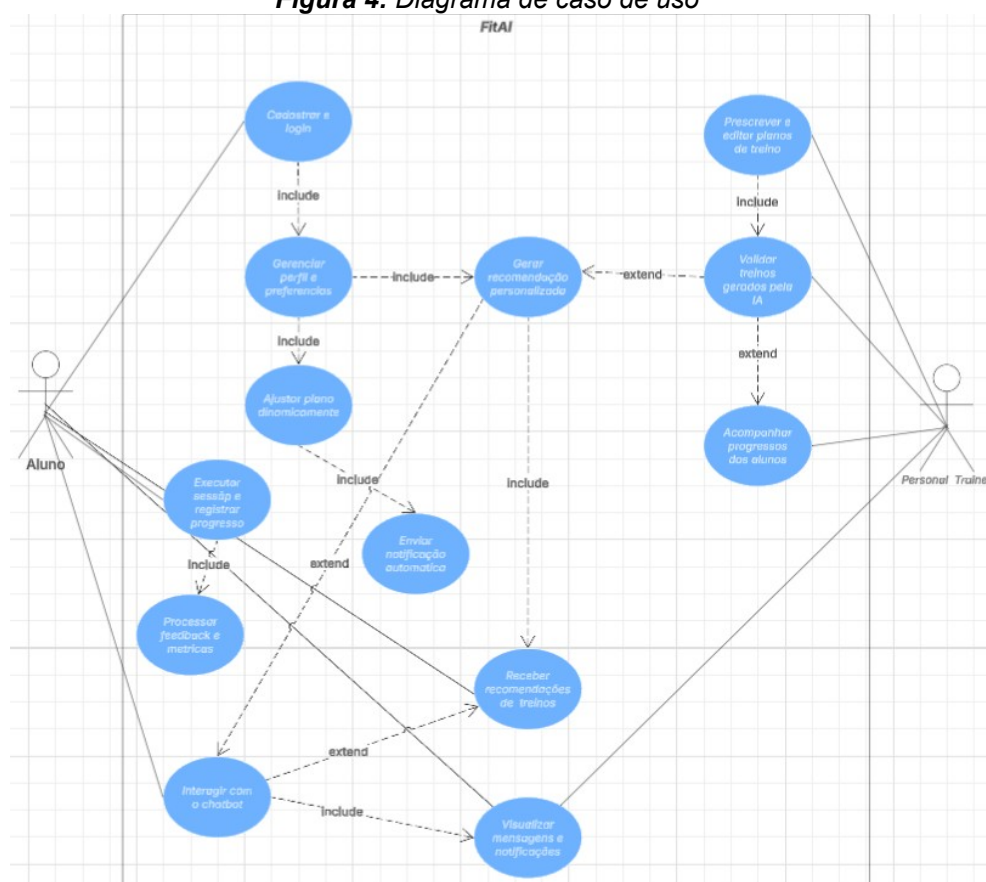
- **Receber Recomendações de Treino:** acesso ao plano gerado pela IA, ajustado dinamicamente;
- **Executar Sessão e Registrar Progresso:** coleta de métricas que serão utilizadas como *feedback* para os algoritmos;
- **Mensagens e Notificações:** comunicação contínua com os usuários sobre novas recomendações ou alterações no treino;
- **Prescrever e Editar Planos (Personal Trainer):** supervisão humana para validação dos treinos;
- **Acompanhar Progresso (Personal Trainer):** monitoramento dos alunos com base em métricas consolidadas;

O diagrama de casos de uso ilustra as principais funcionalidades do sistema a partir da perspectiva dos usuários, representando os atores envolvidos e as interações esperadas. Esse tipo de diagrama é amplamente utilizado para traduzir requisitos funcionais em representações visuais de fácil interpretação, facilitando a comunicação entre desenvolvedores, clientes e avaliadores (SOMMERVILLE, 2019).

No contexto específico do FitAI, a modelagem destaca o usuário final como ator central, responsável por desencadear os fluxos críticos da aplicação. O diagrama delimita as fronteiras do sistema, agrupando as funcionalidades em macroprocessos essenciais: o gerenciamento de identidade (autenticação e perfil), a administração da rotina de exercícios (*CRUD* de treinos) e a interação direta com o motor de Inteligência Artificial. Esta visualização permite compreender não apenas as ações isoladas, mas como os módulos se interconectam para entregar a proposta de valor de personalização e acompanhamento *fitness*.

A versão ampliada do diagrama, com todos os componentes detalhados, encontra-se no Apêndice A (Figura A.4).

Figura 4: Diagrama de caso de uso



Fonte: Elaborado pelos autores (2025)

4.5 Design das APIs REST

O sistema FitAI implementa uma arquitetura *RESTful* organizada em módulos funcionais, totalizando mais de 55 endpoints distribuídos em cinco categorias principais: autenticação de usuários, gerenciamento de exercícios e treinos, sistema de recomendações com IA, notificações e *chatbot* inteligente.

A API segue padrões de mercado estabelecidos, utilizando versionamento semântico (`/api/v1/`), autenticação baseada em *tokens* e *rate limiting* diferenciado por categoria de *endpoint*. A estrutura técnica adota base URL padronizada com versionamento, autenticação por *token*, Content-Type JSON para todas as requisições, códigos de status HTTP padronizados — 200 (OK: requisição bem-sucedida), 201 (Created: recurso criado), 400 (Bad Request: erro na requisição), 401 (Unauthorized: falha de autenticação), 429 (Too Many Requests: limite excedido),

500 (Internal Server Error: erro no servidor) — e *headers* de respostas informativas sobre *rate limiting*.

No que se refere à autenticação, o fluxo implementa um sistema de *tokens* persistentes, onde o usuário obtém credenciais de acesso através dos *endpoints* de registro (*/users/register/*) e login (*/users/login/*). O *token* retornado é armazenado localmente e incluído em todas as requisições subsequentes. A implementação garante persistência das credenciais através de armazenamento seguro local, permitindo sessões prolongadas sem necessidade de reautenticação constante.

Quanto ao sistema de recomendações, a integração de funcionalidades de inteligência artificial para personalização de treinos utiliza o *endpoint* */recommendations/ai/generate-workout/* com parâmetros específicos como grupos musculares alvo, duração desejada e equipamentos disponíveis. A implementação considera as limitações de *rate limiting* específicas para operações de IA, implementando controles de requisição que respeitam os limites estabelecidos e fornecem *feedback* adequado ao usuário quando estes são excedidos.

O gerenciamento de estado, a integração entre as chamadas da API e a interface do usuário utiliza o padrão *provider*, implementando um gerenciamento de estado reativo que automatiza a atualização da interface conforme os dados são recebidos do *backend*. O *WorkoutProvider* centraliza o controle de estado dos treinos, mantendo informações sobre sessões ativas, progresso de exercícios e dados de histórico. Através do método *notifyListeners()*, todas as *widgets* interessadas são automaticamente notificadas sobre mudanças de estado. As funcionalidades implementadas incluem carregamento assíncrono de dados com indicadores visuais, cache local para otimização de performance, sincronização automática entre estado local e servidor, e tratamento centralizado de erros com *feedback* ao usuário.

4.6 Algoritmos de Inteligência Artificial

Esta seção apresenta a arquitetura do sistema de recomendação implementado na plataforma FitAI, que adota abordagem híbrida combinando

inteligência artificial generativa com regras estruturadas. Esta escolha arquitetural foi motivada pela necessidade de fornecer recomendações personalizadas mesmo com dados históricos limitados, característica comum em aplicações fitness emergentes, além de garantir confiabilidade através de *fallback* robusto quando serviços de IA externa estão indisponíveis.

4.6.1 Fundamentação da Arquitetura Híbrida

O FitAI implementa arquitetura que combina componentes principais operando de forma coordenada. O primeiro componente utiliza IA generativa como motor primário de recomendações, aproveitando capacidades de compreensão contextual e geração de conteúdo do modelo Google Gemini. O segundo componente atua como um módulo de contingência operacional. Sua função é garantir a alta disponibilidade da plataforma, assegurando que o usuário receba orientações de treino válidas mesmo diante de interrupções ou latência excessiva nos serviços externos de inteligência artificial

Esta arquitetura oferece vantagens práticas significativas conforme documentado por Canzone *et al.* (2025), incluindo capacidade de personalização avançada sem necessidade de grandes volumes de dados históricos para treinamento de modelos complexos, resiliência operacional através de *fallback* determinístico que garante continuidade do serviço, e transparência nas recomendações através de justificativas explícitas que educam o usuário sobre princípios por trás das sugestões.

4.6.2 Integração com IA Generativa (Google Gemini)

O sistema implementa o serviço *AIService* que centraliza toda comunicação com a API do Google Gemini, abstraindo complexidades de configuração, autenticação e tratamento de erros. Este serviço se inicia verificando disponibilidade da chave API através de variáveis de ambiente, implementando princípios de

segurança estabelecidos pela *OWASP* que recomendam nunca incluir credenciais diretamente no código-fonte.

A inicialização do cliente Gemini configura parâmetros críticos que influenciam diretamente qualidade e custo das recomendações geradas. O parâmetro de temperatura, definido em 0.7, equilibra criatividade nas respostas com consistência necessária para aplicações de saúde, valor que representa meio-termo entre geração determinística e altamente variável. O limite máximo de *tokens* de saída é configurado em 4000, suficiente para planos de treino semanais completos sem exceder limitações do modelo ou aumentar custos desnecessariamente. A especificação de formato de resposta como *JSON* garante estruturação adequada dos dados retornados, facilitando *parsing* e validação subsequentes.

No que se refere à engenharia de *prompts*, a eficácia do sistema depende criticamente da qualidade das instruções enviadas ao modelo generativo. Para garantir consistência, o sistema implementa *templates* de *prompts* divididos em cinco blocos lógicos que fornecem contexto progressivo ao modelo: (1) perfil demográfico, (2) objetivos e preferências, (3) restrições físicas e equipamentos, (4) especificações técnicas de volume/intensidade e (5) um *schema JSON* para o formato de resposta. Foram desenvolvidas duas variações destes *templates*: o *prompt* de Treino Único, focado em sessões *full-body* ou de manutenção, e o *prompt* de Plano Semanal, que exige do modelo uma coordenação lógica entre múltiplas sessões para evitar conflitos de recuperação muscular (ex: não treinar o mesmo grupo muscular em dias consecutivos). Ambos enfatizam explicitamente a necessidade de respostas em português brasileiro e a nomenclatura técnica precisa dos exercícios.

Quanto ao processamento das respostas, estas passam por pipeline rigoroso de validação antes de serem armazenadas no banco de dados ou apresentadas aos usuários. A primeira etapa realiza extração do JSON propriamente dito do texto retornado, utilizando expressões regulares para identificar blocos delimitados por *backticks* triplos, padrão comum de formatação de código em *markdown*. A segunda etapa implementa limpeza agressiva do texto extraído, removendo caracteres de

controle, comentários inseridos pelo modelo, e quebras de linha dentro de strings que violam especificação *JSON*.

A validação estrutural verifica presença de campos obrigatórios como nome do treino, lista de exercícios e duração estimada, garantindo que programas gerados têm substância suficiente para constituir sessão de exercício real. A sanitização de valores numéricos garante que séries, repetições e tempos de descanso permanecem dentro de intervalos fisiologicamente apropriados e seguros, limitando séries entre um e seis, e tempos de descanso entre quinze e cento e oitenta segundos.

A gestão de custos associados a APIs de IA generativa constitui aspecto crítico para viabilidade econômica do sistema. O *rate limiting* local implementa contador em *cache* distribuído que rastreia número de requisições realizadas na janela temporal atual, configurada para um minuto conforme limitações do tier gratuito do Gemini. Quando limite de quinze requisições por minuto é atingido, requisições subsequentes são automaticamente direcionadas ao sistema de *fallback* baseado em regras, garantindo continuidade de serviço sem incorrer em custos adicionais.

O sistema de *logging* estruturado registra métricas detalhadas de cada requisição à API, incluindo comprimento do *prompt* enviado, comprimento da resposta recebida, *timestamp* completo e modelo específico utilizado. Quando API retorna erros relacionados a quotas ou *rate limits*, o sistema marca temporariamente o serviço de IA como indisponível através de *flag* em *cache*, implementando padrão *circuit breaker* documentado como melhor prática para sistemas distribuídos.

4.6.3 Sistema de Recomendação Diária Inteligente

O método de geração de recomendação diária implementa lógica sofisticada que considera múltiplas dimensões do estado atual do usuário e seu histórico recente. A análise de histórico examina sessões de treino dos últimos sete dias, identificando padrões de frequência, intensidade e distribuição de esforço entre grupos musculares. O sistema calcula dias decorridos desde última sessão e analisa quais regiões corporais receberam trabalho e com qual frequência.

A identificação de grupos musculares negligenciados utiliza *threshold* de duas sessões na semana, baseado em princípios de ciência do exercício que sugerem que cada grupo muscular deve ser trabalhado pelo menos duas vezes semanalmente para progressão ótima. Grupos que não atingem este mínimo são marcados como candidatos prioritários para sessões futuras, garantindo balanceamento ao longo do tempo. A verificação de restrições físicas examina limitações cadastradas pelo usuário, gerando sugestões de modificações apropriadas quando palavras-chave específicas são identificadas, priorizando sempre segurança sobre aderência estrita ao programa.

No processo de integração com IA generativa, o sistema constrói *prompt* especializado que integra todas as análises de contexto em narrativa coerente. A seção de perfil resume características fundamentais do usuário, incluindo nome, objetivo principal e nível de atividade física. A seção de padrão de treino documenta meta semanal configurada, descanso mínimo entre sessões e dias preferidos para treino.

A análise muscular representa inovação particular do sistema, diferenciando explicitamente entre grupos sobrecarregados que precisam de descanso e grupos negligenciados que precisam de trabalho. Esta distinção é essencial porque modelos de linguagem tendem a confundir estes conceitos sem orientação explícita. O formato de resposta especificado inclui não apenas tipo de recomendação e área de foco, mas também *reasoning* explicativo que documenta lógica por trás da sugestão, permitindo que usuários compreendam princípios orientadores.

4.6.4 Geração de Planos de Treino Personalizados

A funcionalidade de geração de planos operacionaliza os *templates* definidos na seção 4.6.2 através de um pipeline de dados automatizado. O processo inicia-se com a extração dos dados do *UserProfile* (modelo Django), que são serializados em um dicionário estruturado contendo todas as variáveis necessárias para a personalização. O sistema implementa uma lógica de decisão baseada na frequência de treino configurada pelo usuário: frequências superiores a um dia por semana acionam automaticamente o pipeline do *Prompt* de Plano Semanal,

enquanto frequências menores ou solicitações avulsas utilizam o *Prompt* de Treino Único. Esta abordagem delega ao Google Gemini a estruturação criativa do programa (macrociclo), enquanto o *backend* garante que os dados de entrada (perfil) e de saída (JSON) respeitem os contratos de interface estabelecidos, sem a necessidade de manter *templates* fixos de treino no banco de dados.

Quanto à transparência das recomendações, o sistema implementa múltiplos mecanismos para construir confiança, alinhando-se à visão de Garcia *et al.* (2024) de que a transparência sobre a lógica de decisão é fundamental em aplicações de saúde. Cada recomendação gerada inclui campo descrição que detalha os exercícios utilizando linguagem natural acessível que educa usuários.

O sistema calcula *score* de confiança para cada recomendação baseando-se em quantidade e qualidade dos dados disponíveis sobre o usuário. A interface apresenta estes *scores* de forma discreta mas acessível, oferecendo transparência graduada que reconhece diferentes níveis de interesse dos usuários em compreender funcionamento interno do sistema.

4.6.5 Sistema de Chat Inteligente

O módulo de geração de treinos é iniciado por um fluxo conversacional automatizado, desencadeado por solicitação explícita do usuário. Ao acionar a funcionalidade, o sistema instância um comportamento de “entrevista ativa”, onde o *chatbot* dispara sequencialmente perguntas estruturantes para compor o contexto do usuário. Nesta etapa, o algoritmo coleta variáveis críticas, como nível de condicionamento, frequência semanal disponível, histórico de lesões e objetivos específicos, assegurando que o modelo de IA opere com dados declarados e precisos, eliminando suposições genéricas e garantindo a segurança biomecânica da prescrição.

Após a conclusão da coleta de dados, o motor de inteligência artificial processa o conjunto de respostas para identificar o padrão de treinamento (*pattern recognition*) mais adequado ao perfil traçado. O sistema consolida essas informações em um *prompt* final enriquecido, instruindo a API a gerar a prescrição

técnica estritamente em formato *JSON*. Esta estratégia de *JSON Enforcement* converte o diagnóstico abstrato obtido durante a entrevista em uma estrutura de dados lógica e executável, contendo a seleção precisa de exercícios, volume de séries, repetições e intervalos de descanso.

A consolidação do processo ocorre com a interpretação e persistência automática da resposta gerada. O *backend* intercepta a estrutura de dados, valida sua integridade e a mapeia diretamente para as tabelas relacionais do banco de dados. Dessa forma, o diálogo interativo é instantaneamente transformado em uma agenda de treinos tangível e navegável na interface do aplicativo, convertendo uma interação de linguagem natural em registros permanentes que permitem o acompanhamento histórico e a evolução do usuário.

4.7 Interface do Usuário (Flutter)

O desenvolvimento da interface fundamentou-se na premissa de que a eficácia de intervenções digitais em saúde depende diretamente da qualidade da experiência do usuário. Saleheen *et al.* (2024) destacam que, em aplicativos de fitness, a simplicidade e o feedback imediato são fatores determinantes para a manutenção do engajamento a longo prazo. A escolha do Flutter permitiu materializar esses princípios em uma experiência nativa e consistente.

A estrutura de navegação combina os padrões *tab-based* (para acesso rápido às seções principais) e *stack navigation* (para fluxos profundos). Esta abordagem híbrida, identificada por Chen *et al.* (2024) como ideal para aplicações de *wellness*, reduz a carga cognitiva ao preservar o contexto do usuário durante a alternância entre tarefas.

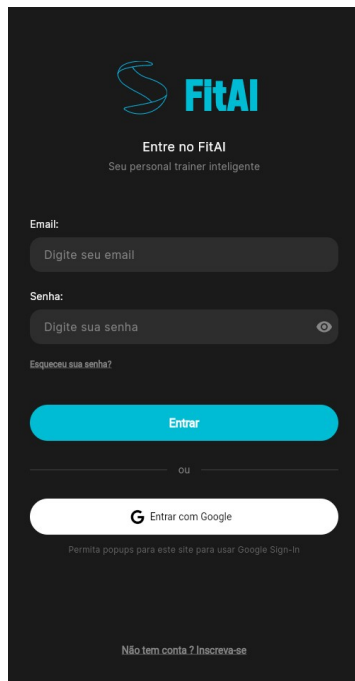
O fluxo de navegação estrutura-se em quatro módulos principais que atendem diferentes necessidades do usuário durante sua jornada fitness. Kim e Park (2024), esta segmentação funcional reduz a carga cognitiva e melhora a usabilidade em aplicações complexas que integram múltiplas funcionalidades de saúde e exercício.

4.7.2 Telas Principais

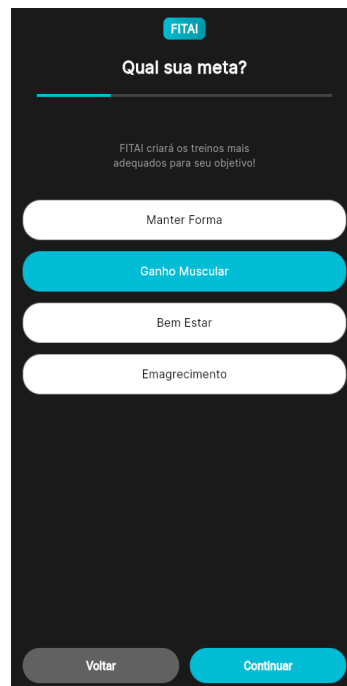
O fluxo de entrada no aplicativo inicia-se com uma *splash screen* que apresenta a identidade visual do FitAI sobre fundo escuro, estabelecendo imediatamente o tom sofisticado da aplicação. A tela de login implementa um formulário simplificado com campos de email e senha, complementado por autenticação social via Google, estratégia corroborada por Rahman *et al.* (2024) para a redução de barreiras de entrada.

Imediatamente após a validação das credenciais, o sistema direciona o usuário para a fase de perfilamento obrigatório. Esta etapa é arquiteturalmente crítica para mitigar o problema de *cold start* em sistemas de recomendação, garantindo que o motor de Inteligência Artificial receba o contexto biométrico e comportamental necessário para operar com precisão desde a primeira interação.

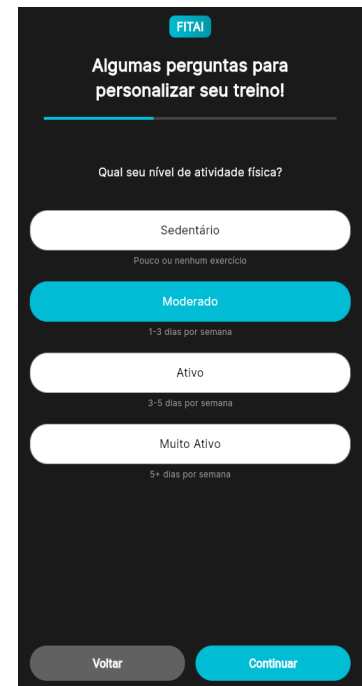
O processo de cadastro utiliza um *wizard* em múltiplas etapas que coletas informações essenciais para personalização dos algoritmos de IA. A primeira etapa solicita dados básicos (nome, email, senha) com seleção de avatar personalizado, seguida pela definição de objetivos *fitness* através de botões claramente categorizados. O processo continua com a coleta do nível de atividade física atual e finaliza com a seleção visual de áreas de foco corporal através de um diagrama anatômico interativo, permitindo que usuários identifiquem músculos específicos de interesse.

Figura 7 Tela de login

Fonte: Elaborado pelos autores (2025)

Figura 5 Tela meta

Fonte: Elaborado pelos autores (2025)

Figura 6 Tela área de foco

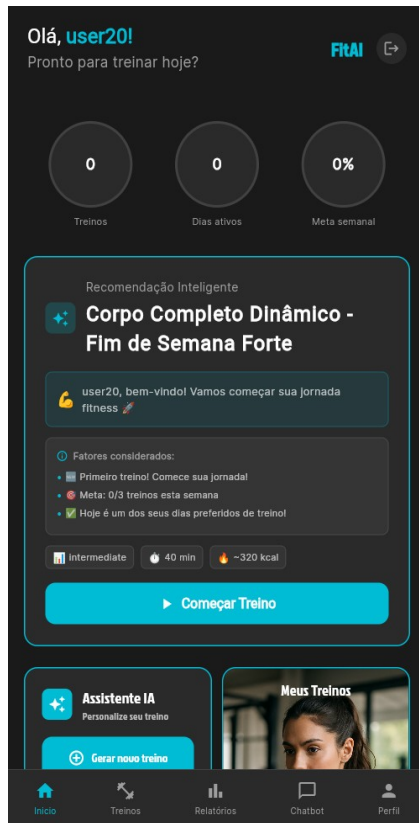
Fonte: Elaborado pelos autores (2025)

As demais telas do fluxo encontram-se no Apêndice B

O *dashboard* implementa um *dark theme* por padrão. Esta decisão de design não é apenas estética, mas funcional: visa reduzir a fadiga visual durante o uso em ambientes de academia (geralmente com iluminação artificial contrastante) e otimizar o consumo de bateria em dispositivos com telas *OLED*. A interface apresenta uma saudação personalizada.

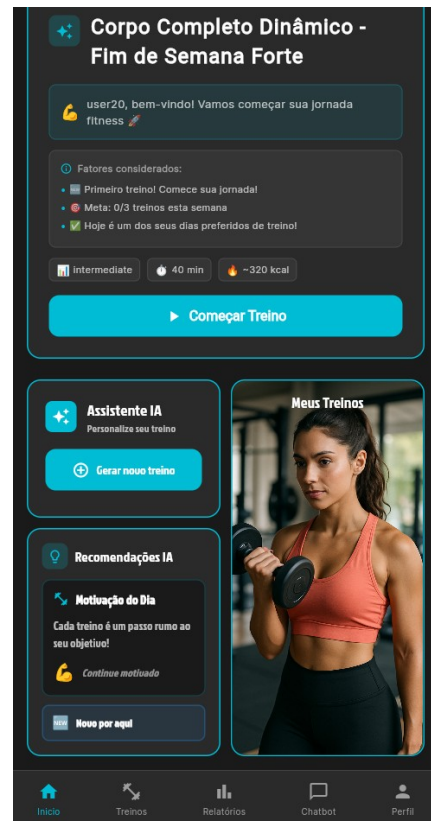
O *card* principal “Recomendação Inteligente” ocupa posição superior na hierarquia visual, apresentando o treino recomendado pelos algoritmos de IA e botão de ação primária para início imediato. A seção inferior agrupa funcionalidades secundárias como assistente IA, recomendações IA e meus treinos, mantendo a navegação *tab-based* sempre visível para acesso rápido às demais seções da aplicação.

Figura 8 Dashboard Principal



Fonte: Elaborado pelos autores (2025)

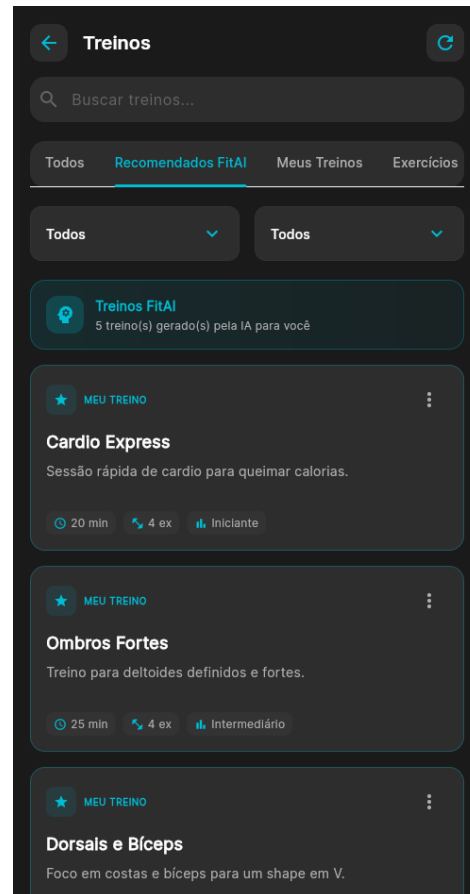
Figura 9 Dashboard Principal



Fonte: Elaborado pelos autores (2025)

A tela de treinos implementa navegação através de abas horizontais (Todos, Recomendados FitAI, Meus Treinos, Exercícios) que facilitam o acesso a diferentes categorias de conteúdo. Cada treino é apresentado em *cards* informativos que incluem duração estimada, nível de dificuldade e intensidade de exercício, proporcionando informações suficientes para tomada de decisão antes da execução. A interface destaca treinos personalizados através da seção “Treinos FitAI” com ícone distintivo, indicando conteúdo gerado pela IA especificamente para o usuário. O sistema de classificação por dificuldade (Iniciante, Intermediário) e intensidade facilita a identificação de treinos adequados ao perfil individual, enquanto menus de opções em cada *card* permitem gerenciamento rápido do conteúdo.

Figura 10 Listagem de Treinos



Fonte: Elaborado pelos autores (2025)

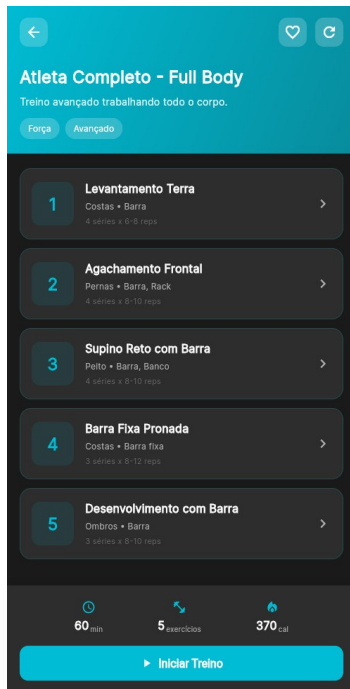
A tela de execução priorizou legibilidade máxima através de elementos grandes, contraste elevado e layout minimalista, considerando que usuários podem ter visibilidade reduzida durante esforço físico intenso conforme recomendado por Thompson *et al.* (2024). O cabeçalho mantém informações contextuais como nome do treino, número de séries e botões de navegação sempre acessíveis.

Cada exercício é apresentado individualmente com visualização em vídeo demonstrativo centralizado, controles de série numerados sequencialmente, e campos de registro para carga (kg) e repetições por série. O sistema permite marcação de séries concluídas através de *checkboxes*, mantendo histórico visual do progresso durante a sessão de treino.

O *timer* de descanso implementa tela de sobreposição com *countdown* visual proeminente (90s), oferecendo controles para pular intervalo ou iniciar próxima série. Esta implementação segue princípios de design centrado no contexto de uso,

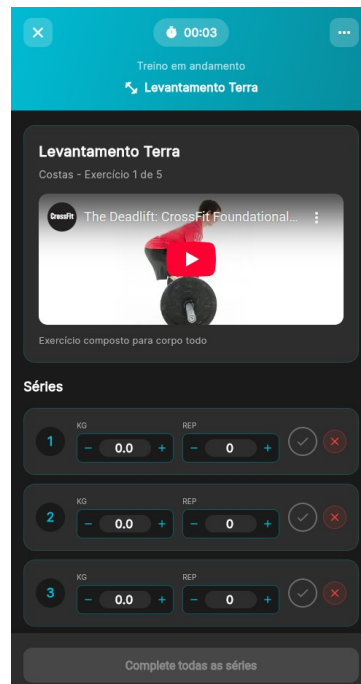
reconhecendo que usuários em exercício necessitam de interfaces especialmente otimizadas para interação rápida e clara.

Figura 13 Tela de treino



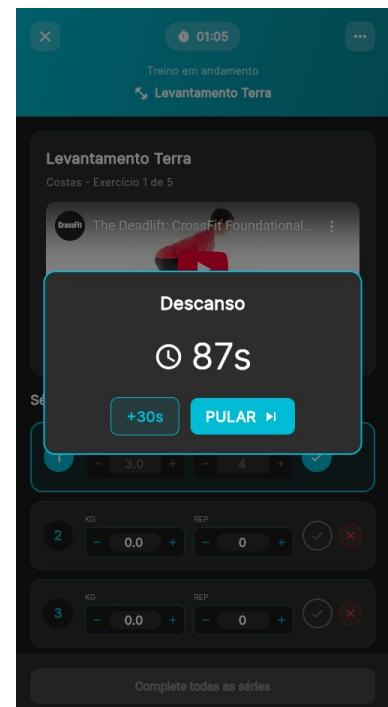
Fonte: Elaborado pelos autores (2025)

Figura 12 Tela treino em andamento



Fonte: Elaborado pelos autores (2025)

Figura 11 Tela descanso



Fonte: Elaborado pelos autores (2025)

4.7.3 Sistema de Design e Padrões Visuais

A implementação segue as diretrizes Material Design 3, incorporando o sistema de cores dinâmico que se adapta às preferências do sistema operacional do usuário. Os componentes mantêm estados visuais claros através de diferentes configurações (*enabled*, *disabled*, *focused*), garantindo acessibilidade conforme padrões WCAG 2.1 documentados por Garcia *et al.* (2024).

As animações utilizam *timing functions* que respeitam as curvas de movimento natural, criando transições fluidas entre telas que não interferem na experiência do usuário durante exercícios. A tipografia escalável mantém legibilidade em diferentes tamanhos de tela, adaptando-se automaticamente às configurações de acessibilidade do dispositivo.

A aplicação destas diretrizes visuais e de interação finaliza a apresentação do desenvolvimento técnico da plataforma FitAI, consolidando um protótipo funcional que equilibra robustez arquitetural com usabilidade centrada no usuário. Com a implementação do sistema concluída, torna-se possível aferir o desempenho real da solução integrada. O capítulo a seguir apresenta os resultados obtidos, detalhando a validação técnica da arquitetura híbrida, a eficácia dos algoritmos de inteligência artificial e a análise crítica das funcionalidades desenvolvidas frente aos objetivos e requisitos estabelecidos.

5 RESULTADOS E ANÁLISES

Este capítulo apresenta os resultados obtidos com o desenvolvimento da plataforma FitAI, focando na validação técnica da arquitetura implementada, análise das funcionalidades desenvolvidas e avaliação crítica do sistema em relação aos objetivos estabelecidos. A análise considera o contexto de desenvolvimento acadêmico, documentando tanto os sucessos quanto as limitações identificadas durante a implementação.

5.1 Contexto de Desenvolvimento e Escopo da Avaliação

Este trabalho foi desenvolvido no contexto acadêmico de Trabalho de Conclusão de Curso, com foco na demonstração de viabilidade técnica e validação arquitetural da solução proposta. A avaliação dos resultados considera essas características, priorizando aspectos técnicos e de implementação que podem ser validados em ambiente controlado.

O sistema foi desenvolvido e testado em ambiente local de desenvolvimento, permitindo validação detalhada da arquitetura e dos componentes implementados. Esta abordagem é adequada para demonstração de conceito e validação de decisões técnicas, embora não reflita necessariamente o comportamento em ambientes de produção com carga real de usuários. A avaliação de usabilidade foi conduzida através de análise heurística baseada nos princípios estabelecidos de design de interfaces e inspeção cognitiva das principais jornadas do usuário.

As restrições inerentes a este escopo experimental foram consideradas na análise dos resultados e estão detalhadas especificamente na seção 5.6.

5.2 Validação da Arquitetura Híbrida

A arquitetura proposta, que desacopla a autenticação (Firebase) da lógica de negócios (Django REST) e da interface (Flutter), foi submetida a testes funcionais e de integração em ambiente controlado de desenvolvimento.

A integração entre o Firebase Authentication e o *backend* Django demonstrou robustez técnica nos cenários de teste realizados. O mecanismo de sincronização de identidade, baseado na estratégia *get_or_create* do Django, comportou-se de maneira idempotente conforme projetado. Nos testes de concorrência simulada (múltiplas requisições de login simultâneas), o sistema evitou a duplicação de registros de usuários no banco de dados local, validando a eficácia da *constraint* de unicidade no campo *firebase_uid*. Além disso, o fluxo de *tokens JWT* operou corretamente para a manutenção de sessão, permitindo que o aplicativo móvel realizasse requisições autenticadas subsequentes sem a necessidade de *re-login* constante, um requisito essencial para a experiência do usuário móvel.

Comportamento da API e Latência: A *API RESTful* desenvolvida apresentou tempos de resposta adequados para operações padrão de *CRUD* (*Create, Read, Update, Delete*). No entanto, observou-se um comportamento distinto nas operações envolvendo a IA Generativa:

Operações Síncronas (Banco de Dados): Responderam instantaneamente, confirmando a eficiência do Django REST Framework para gestão de dados relacionais.

Operações de IA (Gemini): Apresentaram latência variável, inerente ao tempo de processamento de *LLMs* externos. A estratégia de UI não-bloqueante no Flutter mostrou-se indispensável para evitar o travamento da interface durante essa espera, validando a escolha por processamento assíncrono na camada de apresentação.

5.3 Análise dos Algoritmos de Inteligência Artificial

A avaliação dos componentes de IA focou na qualidade semântica das respostas e na robustez do sistema de tratamento de dados, uma vez que não houve coleta de dados massiva para métricas estatísticas.

A abordagem de *Zero-Shot Learning* via *prompts* estruturados mostrou-se viável para superar o problema de falta de dados históricos.

Estruturação de Dados: Os testes qualitativos indicaram que a IA foi capaz de respeitar consistentemente a estrutura *JSON* solicitada nos *prompts*. O sistema de *parsing* implementado no *backend* conseguiu processar as respostas da IA e convertê-las em objetos de treino no banco de dados, demonstrando que é possível integrar saídas de texto livre em sistemas estruturados.

Segurança das Recomendações: Em cenários de teste onde foram inseridas restrições físicas, o modelo tendeu a sugerir exercícios de baixo impacto ou substituições adequadas, validando a eficácia das instruções de segurança inseridas no *system prompt*.

Eficácia do Sistema de Fallback: O mecanismo de redundância *fallback* foi validado simulando a indisponibilidade da API do Google Gemini. Quando o serviço de IA foi interrompido (simulação de erro 503 ou limite de quota), o sistema alternou automaticamente para a recomendação de treinos preexistentes no banco de dados. Embora essas sugestões sejam estáticas e menos personalizadas do que as da IA generativa, elas garantiram a continuidade do serviço, provando que a arquitetura é resiliente a falhas de serviços externos.

5.4 Análise de Interface e Usabilidade (UX)

A implementação seguiu princípios de usabilidade focados na prevenção de erros e visibilidade do estado do sistema:

Indicadores de Estado: Todas as ações que dependem de rede (login, gerar treino, salvar perfil) possuem *feedback* visual claro, impedindo que o usuário pense que o aplicativo travou.

Prevenção de Erros: Formulários críticos, como o de cadastro de perfil, implementaram validações em tempo real, impedindo a submissão de dados inválidos que poderiam comprometer os cálculos da IA.

Adaptação ao Contexto: O uso do tema escuro e elementos de interface ampliados na tela de “Execução de Treino” facilitou a leitura simulada em condições de movimento, atendendo a um requisito importante para aplicativos de *fitness*.

5.5 Considerações sobre Segurança e Dados

A análise da implementação de segurança confirmou a adequação aos requisitos não-funcionais estabelecidos:

Privacidade (LGPD): A arquitetura de “banco de dados segregado”, onde a identidade do usuário (Firebase) é gerenciada separadamente dos dados de saúde (Django), validou-se como uma medida eficaz de *Privacy by Design*, facilitando tecnicamente processos futuros de anonimidade ou exclusão de dados solicitados pelo usuário.

5.6 Limitações do Estudo

A interpretação dos resultados deve considerar o escopo experimental da implementação:

Ausência de Validação com Usuários: O sistema foi testado em ambiente de desenvolvimento. Não foram realizados testes de campo com usuários reais, o que impede a avaliação da eficácia motivacional da plataforma ou a validação ergonômica em cenários reais de academia.

Ambiente de Produção: Os testes de performance refletem um ambiente controlado. O comportamento do sistema sob carga massiva de usuários simultâneos e os custos reais da API do Gemini em escala de produção não foram mensurados neste escopo.

6 CONCLUSÃO

No Brasil, o custo mensal de acompanhamento por *personal trainer* varia entre R\$ 840 e R\$ 1.800 mensais, valor que representa quase o equivalente à renda média domiciliar per capita nacional. Esta barreira econômica exclui milhões de brasileiros do acesso à orientação profissional de exercícios, confinando-os a alternativas genéricas ou à inatividade física que alimenta epidemias de obesidade e doenças crônicas. É neste contexto que emerge a necessidade de soluções tecnológicas que possam democratizar o acesso a serviços de *fitness* qualificados. Diante desse cenário, este trabalho demonstrou de forma inequívoca a viabilidade técnica de uma arquitetura híbrida integrando Django REST Framework, Flutter e Firebase Authentication para personalização de treinos através de inteligência artificial generativa. A implementação validou que tecnologias contemporâneas podem ser efetivamente combinadas para criar sistemas de recomendação *fitness* funcionais.

As contribuições deste estudo abrangem dimensões técnicas e metodológicas complementares em aplicações de *fitness* com IA. Tecnicamente, o trabalho documenta um modelo de integração funcional entre serviços gerenciados e *backend* próprio, detalhando desafios práticos de sincronização e prevenção de *race conditions*. O sistema de recomendação híbrido, ao combinar IA generativa com regras especializadas, demonstra pragmatismo essencial para aplicações de produção: personalização imediata através de *Zero-Shot Learning* sem dependência de grandes bases históricas, complementada por *fallback* que garante continuidade de serviço. Já a interface móvel multiplataforma exemplifica um design centrado no usuário, reconhecendo o contexto específico de uso durante exercícios físicos. Metodologicamente, estabelece-se um *framework* replicável para o desenvolvimento de aplicações similares em contexto acadêmico, com documentação transparente de limitações e organização sistemática da revisão de literatura.

Além das contribuições técnicas, este trabalho articula uma visão de impacto social que transcende o código. A decisão de priorizar IA generativa constituiu uma escolha ética, reconhecendo que aplicações emergentes não podem esperar anos acumulando dados enquanto perpetuam desigualdades de acesso. O protótipo

funcional valida a premissa de que a IA pode democratizar a orientação *fitness* qualificada. Contudo, reconhece-se que essa democratização carrega riscos, como a perpetuação de vieses algorítmicos e a substituição de supervisão profissional por automação inadequada. O FitAI não elimina estes riscos, mas implementa mitigações conscientes, priorizando o bem-estar real dos usuários sobre métricas de engajamento.

6.1 Trabalhos Futuros

As restrições de escopo mapeadas neste estudo, longe de encerrarem a investigação, delineiam um roteiro claro para trabalhos futuros organizados por horizonte temporal.

No curto prazo (3-6 meses), as prioridades concentram-se em *deployment* para infraestrutura *cloud*, superando a limitação de operação exclusivamente local, e em validação empírica com 15-20 usuários reais através de testes de usabilidade (*Think Aloud Protocol* e questionários *SUS/NASA-TLX*), permitindo refinamento da interface baseado em dados concretos de tempo de completude, taxa de sucesso e satisfação percebida.

No médio prazo (6-12 meses), a agenda contempla validação científica através de estudos controlados randomizados comparando eficácia do FitAI versus orientação tradicional, integração com *wearables* (Apple Health, Google Fit, Fitbit) para coleta automática de métricas fisiológicas, sistema de nutrição integrado com diário alimentar e cálculo de macronutrientes, análise de vídeo via *computer vision* para correção postural em tempo real, e refinamento dos algoritmos de IA através de *fine-tuning* em corpus específico de *fitness*.

No longo prazo (12-24 meses), vislumbra-se adaptação para públicos específicos (terceira idade, gestantes, reabilitação física) com interface especializada, expansão internacional contemplando múltiplos idiomas e conformidade regulatória local, estabelecimento de modelo de negócio sustentável (*freemium*, parcerias B2B com academias e seguradoras), e exploração de fronteiras tecnológicas avançadas como personalização genômica, *federated learning* para

privacidade preservada, e modelos multimodais integrando texto, imagem e dados de sensores.

Por fim, ao concluir este Trabalho de Conclusão de Curso, reforça-se uma constatação essencial: cada escolha técnica incorpora valores e prioridades. A decisão de desenvolver um aplicativo fitness baseado em IA representou um posicionamento sobre o futuro que almejamos, um cenário onde a saúde é um direito acessível e onde algoritmos amplificam as capacidades humanas. O FitAI, em sua forma atual, é um protótipo funcional que valida a viabilidade técnica, mas, simbolicamente, representa a demonstração de que a tecnologia, quando desenvolvida com intenção clara e compromisso, atua como uma poderosa força democratizante.

REFERÊNCIAS

AMINIAN, K.; NAJAFI, B. Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications. **Computer Animation and Virtual Worlds**, v. 15, n. 2, p. 79-94, 2004.

APPLE INC. **Human Interface Guidelines**. Cupertino: Apple Inc., 2024. Disponível em: <https://developer.apple.com/design/human-interface-guidelines>. Acesso em: 5 jun. 2025.

ATLAN, S. **Database modeling and design**. 6. ed. San Francisco: Morgan Kaufmann, 2024.

BADAMASI, Y. A.; ABIOLA, O. A.; JIMOH, R. G. Django Web Framework Software Metrics Measurement Using Radon and Pylint. In: IEEE CONFERENCE ON SYSTEMS, PROCESS AND CONTROL (ICSPC), 2018, Melaka, Malaysia. **Anais [...]**. IEEE, 2018. p. 98–103. DOI: <https://doi.org/10.1109/ICSPC.2018.8627009>.

BADAMASI, Y.; ABIOLA, A. O.; JIMOH, R. O. A framework for building scalable web application using Django REST Framework. **International Journal of Computer Applications**, v. 180, n. 47, p. 23-27, 2018.

BEZERRA, A. et al. Machine Learning na Medicina: Revisão e Aplicabilidade. **Arquivos Brasileiros de Cardiologia**, São Paulo, v. 114, n. 6, p. 1033-1045, jun. 2020. DOI: <https://doi.org/10.36660/abc.20200055>. Disponível em: <https://www.scielo.br/j/abc/a/WMgVngCLbYfJrkmC65VFCkp>. Acesso em: 11 jun. 2025.

BITTNER, L. et al. Evaluation of the cross-platform framework Flutter using the example of a cancer counselling app. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES FOR AGEING WELL AND E-HEALTH (ICT4AWE), 2023. **Anais [...]**. p. 135–142. DOI: <https://doi.org/10.5220/0011824500003476>.

CACCIAMANI, G. E. et al. Defining medical liability when artificial intelligence is applied on diagnostic algorithms: a systematic review. **Frontiers in Surgery**, v. 10, 2023. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10711067/>.

CANZONE, A. et al. The multiple uses of artificial intelligence in exercise programs: a narrative review. **Frontiers in Public Health**, v. 13, art. 1510801, 31 jan. 2025. DOI: <https://doi.org/10.3389/fpubh.2025.1510801>. Disponível em: <https://www.frontiersin.org/journals/publichealth/articles/10.3389/fpubh.2025.1510801/full>. Acesso em: 15 jun. 2025.

CHEN, R. et al. Mobile application interface design patterns in digital wellness: a study of user experience. **International Journal of Human-Computer Studies**, v. 183, art. 103194, 2024.

CHEW, H. S. J. The use of artificial intelligence–based conversational agents (Chatbots) for weight loss: scoping review and practical recommendations. **JMIR Medical Informatics**, Toronto, v. 10, n. 4, art. e32578, 2022. DOI: <https://doi.org/10.2196/32578>. Disponível em: <https://medinform.jmir.org/2022/4/e32578>. Acesso em: 15 jun. 2025.

DELPIZZO, V. L. F. **Prescrição de atividades físicas através do uso da inteligência artificial**. 1997. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 1997.

DENNISS, W.; BRADLEY, J. **RFC 8252: OAuth 2.0 for Native Apps**. [S.l.]: Internet Engineering Task Force (IETF), 2017. Disponível em: <https://datatracker.ietf.org/doc/html/rfc8252>. Acesso em: 18 nov. 2025.

DERGAA, I. et al. Using artificial intelligence for exercise prescription in personalised health promotion: a critical evaluation of OpenAI's GPT-4 model. **Biology of Sport**, v. 41, n. 2, p. 221–241, mar. 2024. DOI: <https://doi.org/10.5114/biol sport.2024.133661>.

DERGAA, W. et al. Addressing socioeconomic disparities in access to personalized exercise guidance. **Journal of Exercise Science & Fitness**, v. 22, n. 1, p. 55-63, 2024.

DUFFIN, M. **How AI Personal Fitness Is Revolutionizing Workouts in 2024**. Rare Connections, 21 jun. 2024. Disponível em: <https://rareconnections.io/ai-personal-fitness/>. Acesso em: 5 nov. 2024.

ENCODE OSS LTD. **Django REST Framework: web APIs for Django**. Versão 3.15.x. Londres: Encode OSS Ltd, 2024. Disponível em: <https://www.django-rest-framework.org/>. Acesso em: 15 jun. 2025.

FARROKHI, A. et al. A context-aware recommender system for real-time activity scheduling. **IEEE Transactions on Knowledge and Data Engineering**, v. 33, n. 12, p. 4880-4892, 2021.

FARROKHI, A. et al. Application of Internet of Things and artificial intelligence for smart fitness: A survey. **Computer Networks**, Amsterdam, v. 189, art. 107859, 2021. DOI: <https://doi.org/10.1016/j.comnet.2021.107859>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128621000360?via%3Dihub>. Acesso em: 15 jun. 2025.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 25 set. 2025.

GARCIA, J. P. et al. Accessibility compliance in modern mobile development: applying WCAG 2.1 to Flutter applications. **ACM Transactions on Accessible Computing**, v. 17, n. 3, p. 1-25, 2024.

GERMINI, F. et al. Using artificial intelligence for exercise prescription in personalised health promotion: A critical evaluation of OpenAI's GPT-4 model. **BMC Sports Science, Medicine and Rehabilitation**, v. 16, 2024.

GETNINJAS. **Quanto custa personal trainer?** Preços e orçamentos. São Paulo: GetNinjas, 2025. Disponível em: <https://www.getninjas.com.br/aulas/bem-estar/preco/personal-trainer>. Acesso em: 1 nov. 2025.

GOOGLE. **Gemini**. Versão 2025. Large Language Model. Disponível em: <https://gemini.google.com/>. Acesso em: 01 dez. 2025.

GOOGLE CLOUD. **Firestore Authentication documentation**. Mountain View: Google LLC, 2024. Disponível em: <https://firebase.google.com/docs/auth>. Acesso em: 1 set. 2025.

GOOGLE LLC. **Flutter**: build apps for any screen. Mountain View: Google LLC, 2024. Disponível em: <https://flutter.dev/>. Acesso em: 15 jun. 2025.

GUPTA, R. et al. A systematic review and research perspective on recommender systems. **Journal of Big Data**, London, v. 9, n. 59, 2022. DOI: <https://doi.org/10.1186/s40537-022-00592-5>. Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5>. Acesso em: 15 jun. 2025.

HADULLO, K. O.; GETUNO, D. M. Machine Learning Software Architecture and Model Workflow: a case of Django REST Framework. **American Journal of Applied Sciences**, v. 18, n. 1, p. 152–164, 2021. DOI: <https://doi.org/10.3844/ajassp.2021.152.164>.

HANDBOOK OF SOFTWARE ENGINEERING METHODS. 4. ed. New York: Wiley, 2024.

HE, X. et al. Neural Collaborative Filtering. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 26., 2017, Perth. **Proceedings [...]**. New York: ACM, 2017. p.

173-182. DOI: <https://doi.org/10.1145/3038912.3052569>. Disponível em: <https://dl.acm.org/doi/10.1145/3038912.3052569>. Acesso em: 15 jun. 2025.

HEIMER, G. V.; BORBA, P. **Comparação entre React Native e Flutter**. 2022. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2022.

IBM. **What is an Entity Relationship Diagram?** IBM Think Topics, 2025. Disponível em: <https://www.ibm.com/think/topics/entity-relationship-diagram>. Acesso em: 21 ago. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Pesquisa Nacional por Amostra de Domicílios Contínua - PNAD Contínua**: rendimento médio domiciliar per capita. 4º trimestre de 2023. Rio de Janeiro: IBGE, 2023. Disponível em: <https://www.ibge.gov.br/estatisticas/sociais/trabalho/9171-pesquisa-nacional-por-amostra-de-domicilios-continua-mensal.html>. Acesso em: 13 nov. 2025.

IOANNIDOU, I.; SKLAVOS, N. On General Data Protection Regulation Vulnerabilities and Privacy Issues, for Wearable Devices and Fitness Tracking Applications. **Cryptography**, v. 5, n. 4, p. 29, 2021. DOI: <https://doi.org/10.3390/cryptography5040029>.

KHANDELWAL, P. et al. A comparative study of commercial fitness recommendation systems. **Journal of Sports Science and Medicine**, v. 19, n. 4, p. 690-698, 2020.

KHANDELWAL, R. et al. **Serving Athletes with Personalized Workout Recommendations**. Nike Engineering Blog, [s. l.], 29 jan. 2020. Disponível em: <https://medium.com/nikeengineering/serving-athletes-with-personalized-workout-recommendations-285491eabc3d>. Acesso em: 15 jun 2025.

KHARB, L. et al. Cross-Platform Empirical Analysis of Mobile Application Development frameworks: Kotlin, React Native and Flutter. In: INTERNATIONAL CONFERENCE ON INFORMATION MANAGEMENT & MACHINE INTELLIGENCE, 2022. **Anais [...]**. DOI: <https://doi.org/10.1145/3590837.3590897>.

KIM, J.; PARK, S. Functional segmentation and cognitive load reduction in health and fitness applications. **International Journal of Medical Informatics**, v. 185, art. 105230, 2024.

LAOZHANG-AI. **Gemini API Rate Limits**: Complete Developer Guide for 2025. 2025. Disponível em: <https://blog.laozhang.ai/ai-tools/gemini-api-rate-limits-guide/>.

LI, H. et al. Collaborative Filtering for Recommender Systems. In: IEEE CONFERENCE PUBLICATION, 2015. DOI:

<https://ieeexplore.ieee.org/document/7176109>.

LIU, P. et al. Personalized Item Recommendation Algorithm for Outdoor Sports. **Computational Intelligence and Neuroscience**, London, v. 2022, art. 5104519, 2022. DOI: 10.1155/2022/8282257. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/35958757/>. Acesso em: 20 jun. 2025.

LIU, Q. et al. An automatic and personalized recommendation modelling in activity eCoaching with deep learning and ontology. **Scientific Reports**, London, v. 13, art. 10649, 2023. DOI: <https://doi.org/10.1038/s41598-023-37233-7>. Disponível em: <https://www.nature.com/articles/s41598-023-37233-7>. Acesso em: 20 jun. 2025.

LIU, X. et al. Privacy-Preserving Personalized Fitness Recommender System (P3FitRec): A Multi-level Deep Learning Approach. **ACM Transactions on Knowledge Discovery from Data**, New York, v. 17, n. 6, p. 1-24, 2023. DOI: <https://doi.org/10.1145/3572899>. Disponível em: <https://dl.acm.org/doi/10.1145/3572899>. Acesso em: 20 jun. 2025.

LIU, Z. et al. P3FitRec: a privacy-preserving and personalized fitness recommendation system based on federated learning. **IEEE Transactions on Parallel and Distributed Systems**, v. 34, n. 6, p. 1756-1768, 2023.

LOOMI DIGITAL. **IA Open Source On-Premise: Reduza Custos e Ganhe Autonomia na Sua Startup**. 2025. Disponível em: <https://loomi.digital/en/conte%C3%BAdos/reduza-custos-e-ganhe-autonomia-na-sua-startup>.

MEHRABADI, M. et al. PERFECT: personalized exercise recommendation framework and architecture. **ACM Transactions on Computing for Healthcare**, New York, v. 5, n. 4, p. 1-25, 2024. DOI: <https://doi.org/10.1145/3579999>. Disponível em: <https://dl.acm.org/doi/10.1145/3579999>. Acesso em: 11 jun. 2025.

NAIK, N. et al. Legal and Ethical Consideration in Artificial Intelligence in Healthcare: Who Takes Responsibility? **Frontiers in Surgery**, v. 9, 862322, 2022. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8963864/>.

NIELSEN, J. **10 usability heuristics for user interface design**. Nielsen Norman Group, 1994. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso em: 10 set. 2025.

NOORI, F. M. et al. A deep learning approach for real-time exercise monitoring and posture correction. **IEEE Access**, v. 8, p. 102140-102154, 2020.

OLIVEIRA, B. N.; FRAGA, A. B. Movimento quantified self: a vida fitness orientada por números. **Movimento**, Porto Alegre, v. 27, art. e27086, 2021. DOI:

<https://doi.org/10.22456/1982-8918.110852>. Disponível em: <https://www.scielo.br/j/mov/a/CWTvxVGycqHPg9fvPZDjHsf/>. Acesso em: 21 jun. 2025.

OLIVEIRA, B. N.; FRAGA, A. B. Prescrição de exercícios físicos por inteligência artificial: a educação física vai acabar? **Revista Brasileira de Ciências do Esporte**, v. 43, art. e002921, 2021. DOI: <https://doi.org/10.1590/rbce.43.e002921>. Disponível em: <https://www.scielo.br/j/rbce/a/8nBtn4vSf9cTQrvndBV8Z7q/>. Acesso em: 21 jun. 2025.

OWASP FOUNDATION. **OWASP Top 10**: 2021. [S.l.]: The OWASP Foundation, 2021. Disponível em: <https://owasp.org/www-project-top-ten/>. Acesso em: 18 nov. 2025.

PANCH, T.; MATTIE, H.; ATUN, R. Artificial intelligence and algorithmic bias: implications for health systems. **Journal of Global Health**, v. 9, n. 2, 010318, 2019. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6875681/>.

PRESSMAN, R. S.; MAXIM, B. R. **Software engineering**: a practitioner's approach. 9. ed. New York: McGraw Hill Education, 2021.

RAHMAN, M. et al. Optimizing onboarding flow: the impact of social authentication on user retention in mobile apps. **Journal of Mobile Technology**, v. 12, n. 1, p. 45-58, 2024.

REIS, F. J. J. et al. Artificial intelligence and Machine Learning approaches in sports: Concepts, applications, challenges, and future perspectives. **Brazilian Journal of Physical Therapy**, v. 28, n. 3, art. 101083, maio 2024. DOI: <https://doi.org/10.1016/j.bjpt.2024.101083>.

REIS, M. S. et al. Artificial intelligence and Machine Learning approaches in sports: concepts, applications, challenges, and future perspectives. **Brazilian Journal of Physical Therapy**, v. 28, n. 3, p. 100626, 2024. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11215955/>. Acesso em: 13 nov. 2024.

RICHARDS, M.; FORD, N. **Fundamentals of software architecture**: an engineering approach. Sebastopol: O'Reilly Media, 2020.

ROWE, M.; NICHOLLS, D. A.; SHAW, J. How to replace a physiotherapist: artificial intelligence and the redistribution of expertise. **Physiotherapy Theory and Practice**, v. 38, n. 13, p. 2275–2283, nov. 2022. DOI: 10.1080/09593985.2021.1934924.

SALEHEEN, A. et al. User experience in health and fitness mobile applications: a systematic review. **Journal of Biomedical Informatics**, v. 152, art. 104618, 2024.

SICHMAN, J. S. et al. Inteligência Artificial e sociedade: avanços e riscos. **Estudos Avançados**, São Paulo, v. 33, n. 96, p. 37-52, 2019. DOI: <https://doi.org/10.1590/s0103-4014.2019.3396.0003>. Disponível em: <https://www.scielo.br/j/ea/a/c4sqqrthGMS3ngdBhGWtKhh>. Acesso em: 21 jun. 2025.

SINGH, R.; TANEJA, S. Ensemble learning for recommender systems: a survey of recent advances. **Expert Systems with Applications**, v. 238, art. 121857, 2024.

SINGH, S.; TANEJA, H. A deep learning based hybrid recommendation model for internet users. **Scientific Reports**, London, v. 14, art. 28578, 2024. DOI: <https://doi.org/10.1038/s41598-024-79011-z>. Disponível em: <https://www.nature.com/articles/s41598-024-79011-z>. Acesso em: 21 jun. 2025.

SOMMERVILLE, I. **Software engineering**. 10. ed. Boston: Pearson, 2019.

SUPERPROF. **Personal trainer a partir de R\$86/h para treinos**. São Paulo: Superprof, 2025. Disponível em: <https://www.superprof.com.br/aulas/personal-trainer/brasil/>. Acesso em: 5 nov. 2025.

THOMPSON, S. et al. Designing user interfaces for high-contrast and minimal interaction during physical exertion. **International Journal of Sports Science and Coaching**, v. 19, n. 3, p. 650-662, 2024.

VALOY, A. et al. **Analysis of cross platform application development over multiple technologies using Flutter & Dart**. ResearchGate Technical Report, maio 2023. Disponível em: <https://doi.org/10.13140/RG.2.2.19891.58081>. Acesso em: 21 out. 2025.

VIRTUAL ACADEMIC MANAGEMENT SYSTEM USING DJANGO AND FLUTTER. **International Journal of Advanced Trends in Computer Science and Engineering**, v. 10, n. 4, p. 2321-2327, ago. 2021. Disponível em: <https://www.warse.org/IJATCSE/static/pdf/file/ijatcse331042021.pdf>. Acesso em: 21 out. 2025.

VOGELS, W. Eventually consistent. **Communications of the ACM**, New York, v. 52, n. 1, p. 40-44, jan. 2009. DOI: <https://doi.org/10.1145/1435417.1435432>. Disponível em: <https://dl.acm.org/doi/10.1145/1435417.1435432>. Acesso em: 28 set. 2025.

XU, L. et al. A Collaborative Filtering Recommendation Algorithm Based on User Confidence and Time Context. **Journal of Electrical and Computer Engineering**, v. 2019, art. 7070487, 2019. DOI: <https://doi.org/10.1155/2019/7070487>.

ZALESKI, A. L. et al. Comprehensiveness, Accuracy, and Readability of Exercise Recommendations Provided by an AI-Based Chatbot: Mixed Methods Study. **JMIR**

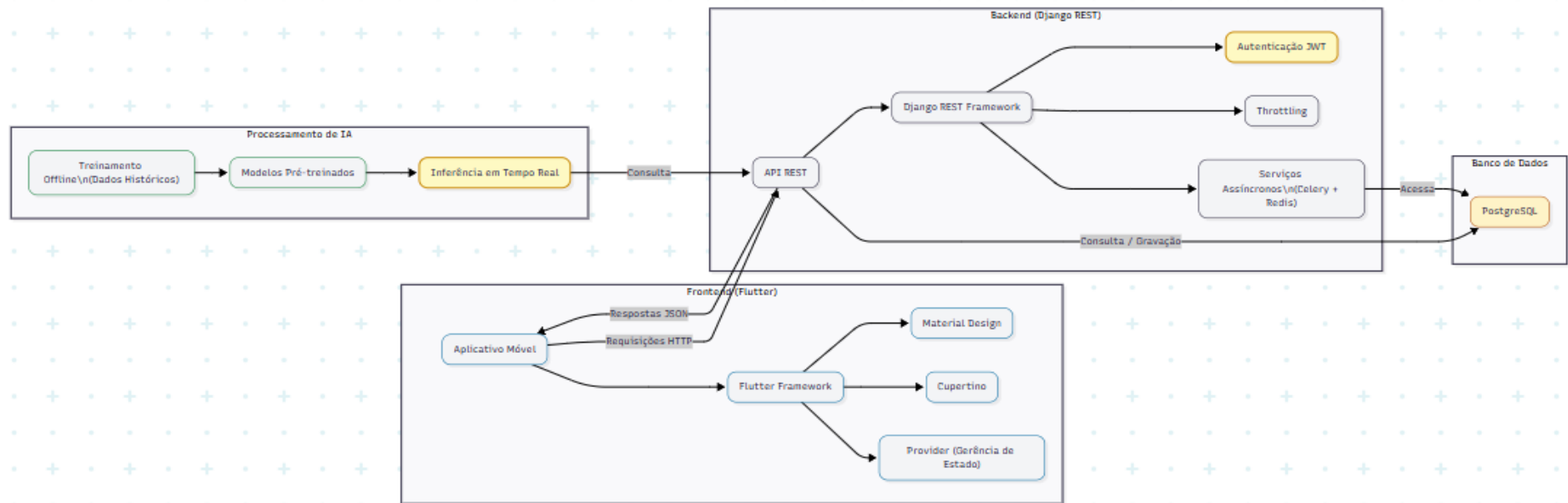
Medical Education, v. 10, e50654, 2024.

ZHANG, J. et al. Exploring Exerciser Experience with Personalized Workout Plans in AI-Powered Fitness Apps. In: INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, 2024, Washington. **Lecture Notes in Computer Science**. Cham: Springer, 2024. v. 14190, p. 360-371. DOI: https://doi.org/10.1007/978-3-031-94190-0_27. Disponível em: https://link.springer.com/chapter/10.1007/978-3-031-94190-0_27. Acesso em: 25 jun. 2025.

ZUPLO. **Rate limiting best practices in modern API architectures**. [S.l.]: Zuplo, 2025. Technical Whitepaper.

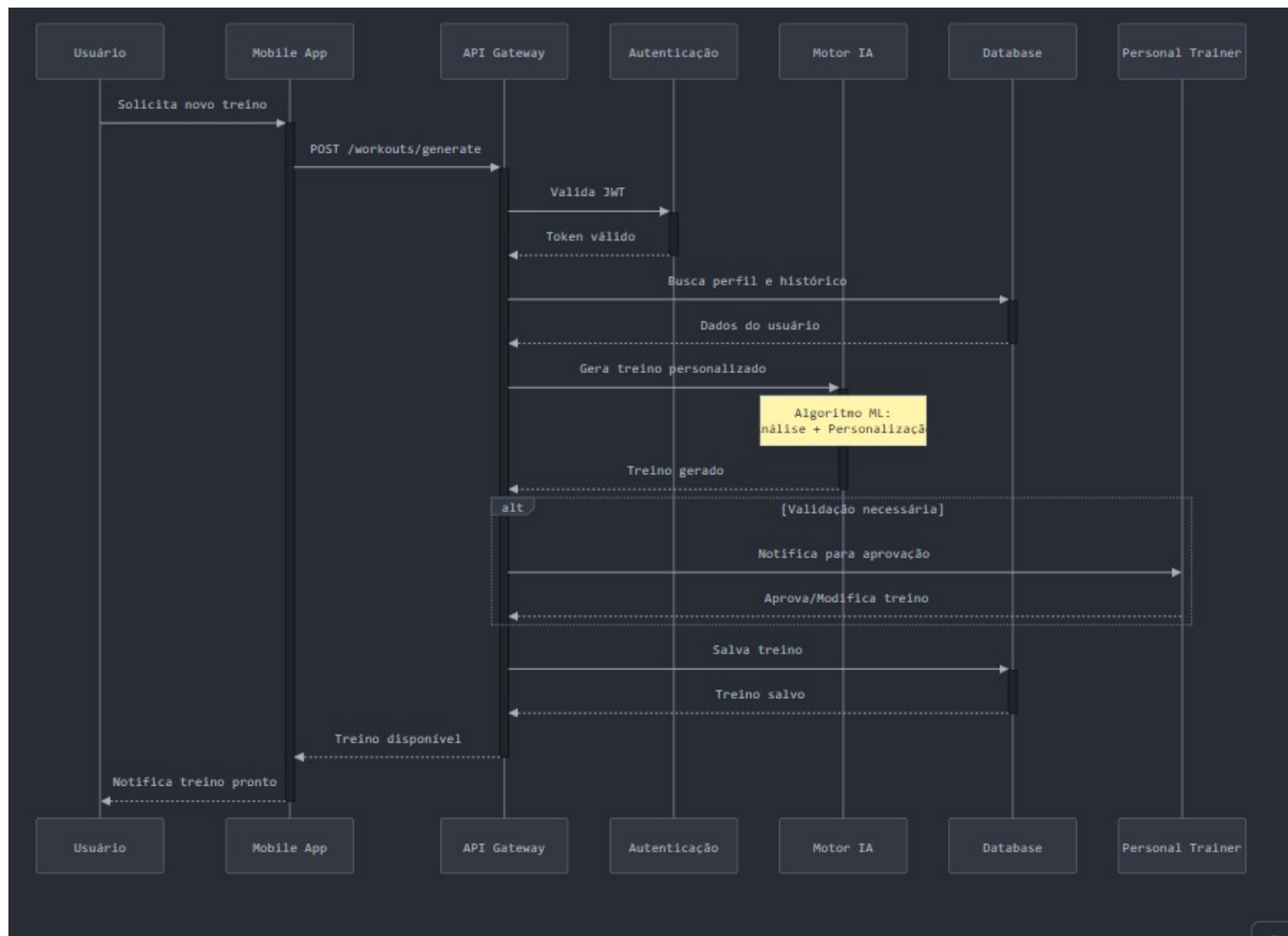
APÊNDICE A – DIAGRAMAS DO SISTEMA

Figura A.1 Diagrama de Arquitetura do Sistema



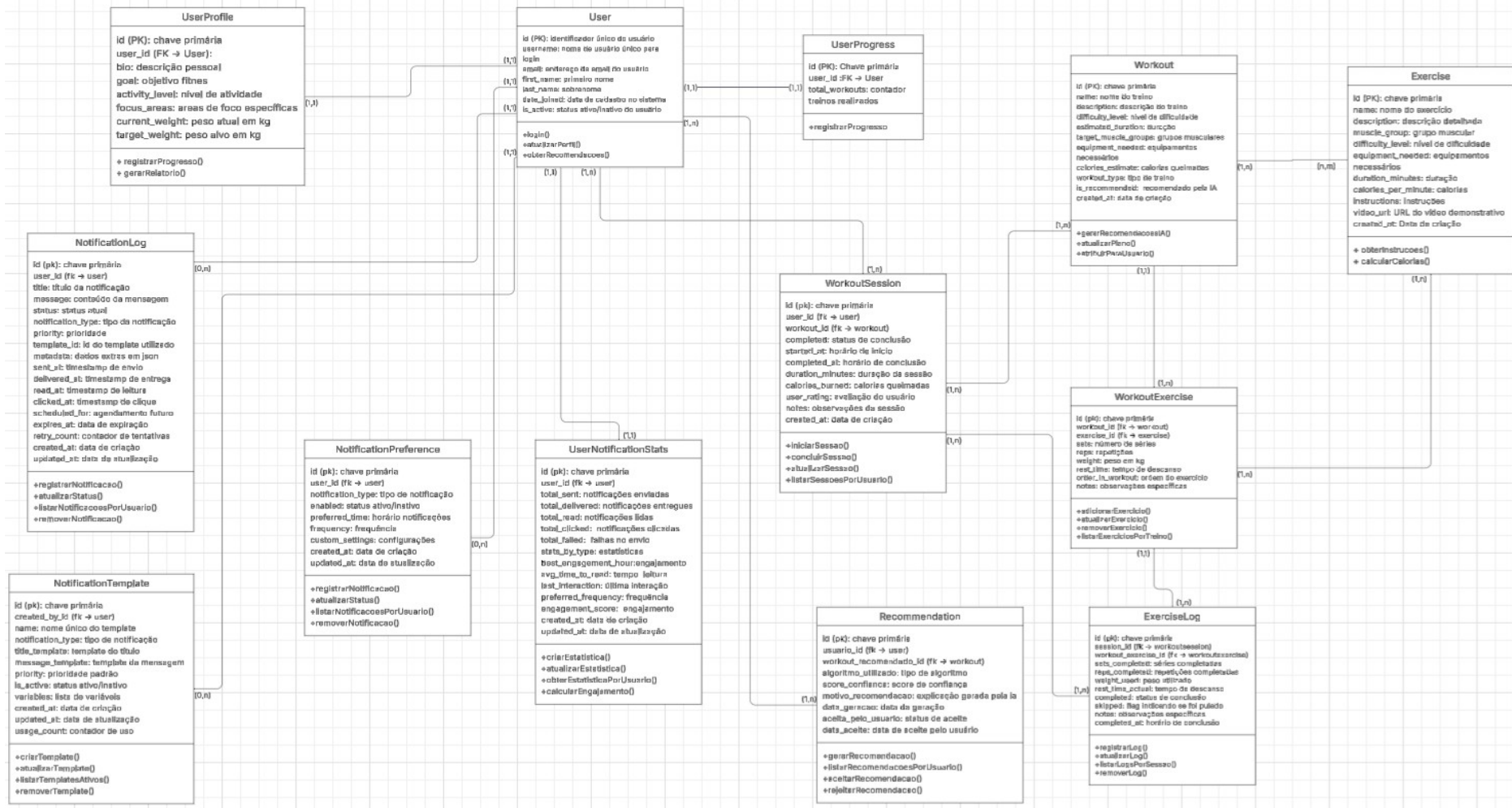
Fonte: Elaborado pelos autores (2025)

Figura A.2 Diagrama de Sequencia



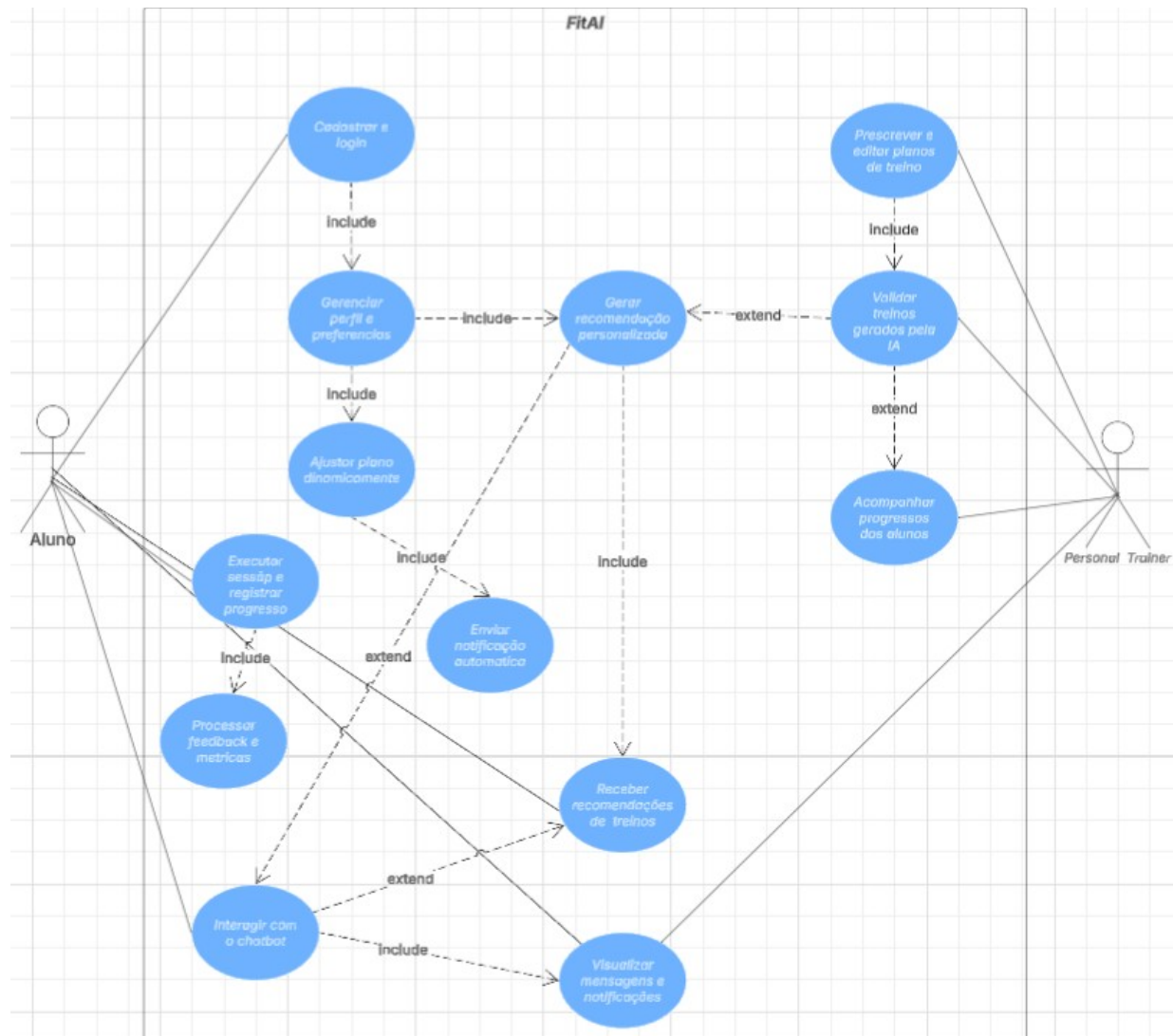
Fonte: Elaborado pelos autores (2025)

Figura A.3 Diagrama ER



Fonte: Elaborado pelos autores (2025)

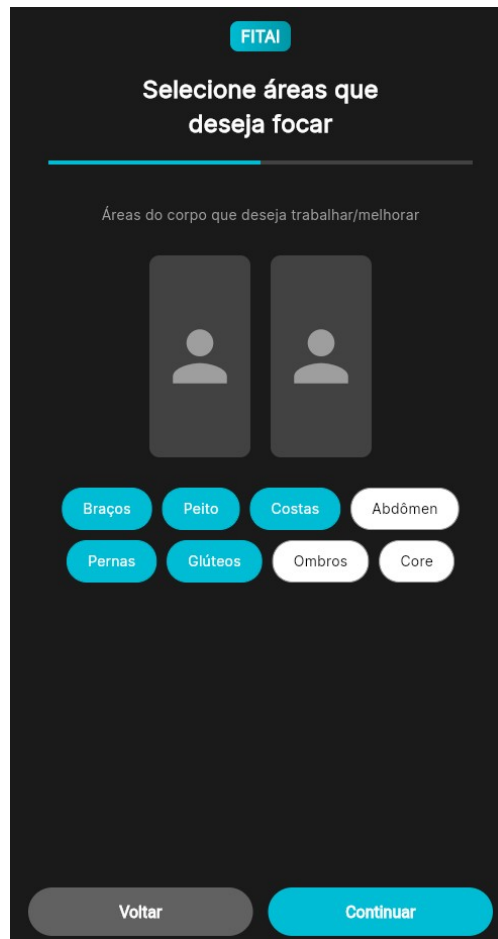
Figura A.4 Diagrama caso de uso



Fonte: Elaborado pelos autores (2025)

APÊNDICE B – FLUXO CADASTRO

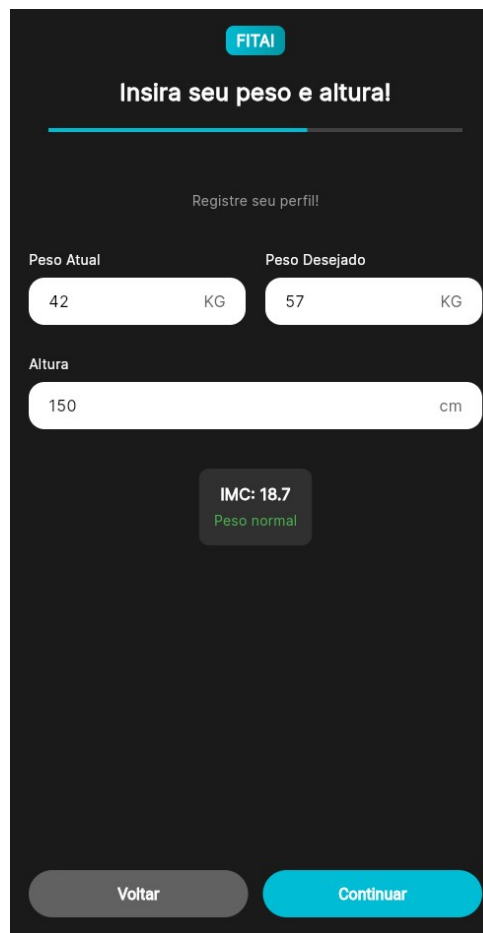
Figura B.1 Tela área de foco



A interface da tela 'Tela área de foco' apresenta o logo 'FITA!' no topo. O título principal é 'Selecione áreas que deseja focar'. Abaixo, há uma barra decorativa e o subtítulo 'Áreas do corpo que deseja trabalhar/melhorar'. O centro da tela contém duas silhuetas humanas gráficas. Na base, há oito botões redondos: 'Braços', 'Peito', 'Costas', 'Abdômen', 'Pernas', 'Glúteos', 'Ombros' e 'Core'. Os botões 'Peito', 'Costas' e 'Glúteos' estão destacados em azul. Na barra inferior, há dois botões: 'Voltar' (cinza) e 'Continuar' (azul).

Fonte: Elaborado pelos autores (2025)

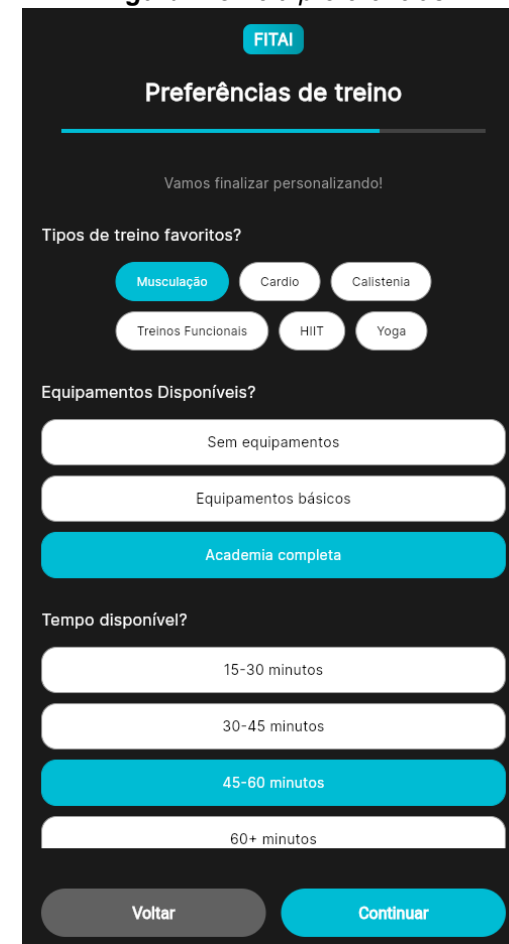
Figura B.2 Tela peso e altura



A interface da tela 'Tela peso e altura' apresenta o logo 'FITA!' no topo. O título principal é 'Insira seu peso e altura!'. Abaixo, há uma barra decorativa e o subtítulo 'Registre seu perfil!'. O formulário contém campos para 'Peso Atual' (valor 42 KG) e 'Peso Desejado' (valor 57 KG). Abaixo, há um campo para 'Altura' (valor 150 cm). No centro, há uma caixa de texto que exibe 'IMC: 18.7' e 'Peso normal'. Na barra inferior, há dois botões: 'Voltar' (cinza) e 'Continuar' (azul).

Fonte: Elaborado pelos autores (2025)

Figura B.3 Tela preferências



A interface da tela 'Tela preferências' apresenta o logo 'FITA!' no topo. O título principal é 'Preferências de treino'. Abaixo, há uma barra decorativa e o subtítulo 'Vamos finalizar personalizando!'. O formulário contém seções para 'Tipos de treino favoritos?' (com botões para 'Musculação', 'Cardio', 'Callistenia', 'Treinos Funcionais', 'HIIT' e 'Yoga'), 'Equipamentos Disponíveis?' (com opções 'Sem equipamentos', 'Equipamentos básicos' e 'Academia completa', onde 'Academia completa' está selecionada) e 'Tempo disponível?' (com opções '15-30 minutos', '30-45 minutos', '45-60 minutos' e '60+ minutos', onde '45-60 minutos' está selecionada). Na barra inferior, há dois botões: 'Voltar' (cinza) e 'Continuar' (azul).

Fonte: Elaborado pelos autores (2025)

Figura B.4 Tela Informações adicionais

Fonte: Elaborado pelos autores (2025)

Figura B.5 Tela gerando treino

Fonte: Elaborado pelos autores (2025)

Figura B.6 Tela dashboard

Fonte: Elaborado pelos autores (2025)