doi.org/10.51891/rease.v12i11.33141

GYMTRAINER: DESENVOLVIMENTO DE UM APLICATIVO MOBILE DE PERSONAL TRAINER VIRTUAL COM INTELIGÊNCIA ARTIFICIAL PARA GERAÇÃO E ACOMPANHAMENTO DE TREINOS DE MUSCULAÇÃO

GYMTRAINER: DEVELOPMENT OF A MOBILE VIRTUAL PERSONAL TRAINER APPLICATION WITH ARTIFICIAL INTELLIGENCE FOR STRENGTH-TRAINING GENERATION AND TRACKING

Guilherme Vinicius Rizo[1]
André Luiz da Silva[2]
Fabiana Florian[3]

RESUMO: Este trabalho apresenta o GymTrainer, um aplicativo de celular para musculação que funciona como um personal trainer virtual. Ele usa inteligência artificial (IA) para montar treinos e acompanha o usuário enquanto ele treina. O sistema tem duas partes: o aplicativo, feito em React Native (Expo), e um servidor com uma API REST em Node.js/Express, que guarda os dados no MongoDB e envia o perfil do usuário à IA Google Gemini para gerar os treinos. O aplicativo foi avaliado por testes funcionais, por um teste automatizado de geração com cinco perfis e por um teste inicial com dois usuários reais. As 15 gerações testadas foram concluídas no formato correto, depois que duas falhas causadas pelo limite de uso do serviço de IA foram repetidas, e todos os planos semanais tiveram o número certo de treinos. O tempo médio de resposta foi de 2,1 segundos para treinos avulsos e de 4,9 segundos para planos semanais. Pelos critérios do American College of Sports Medicine, 80,4% dos exercícios tiveram repetições adequadas ao objetivo e 99,4% tiveram séries adequadas ao nível, e o perfil com lesão no joelho não recebeu exercícios a evitar. Os resultados mostram que é possível unir a IA ao acompanhamento do treino em um aplicativo de baixo custo, mas os treinos ainda precisam ser avaliados por profissionais de educação física.

Palavras-chave: Aplicativo móvel. Inteligência artificial. Personal trainer virtual. React Native. Treino de musculação.

[1] Discente do curso de Sistemas de Informação da Universidade de Araraquara (UNIARA), Araraquara-SP.
[2] Orientador. Doutor e Mestre em Educação Escolar pela Universidade Estadual Paulista (UNESP), MBA em Gestão de Projetos, Bacharel em Informática e Engenheiro de Computação. Docente dos cursos de Sistemas de Informação, Engenharia de Computação e do Mestrado Profissional em Engenharia de Produção da Universidade de Araraquara (UNIARA), Araraquara-SP. Head de Dados e Arquitetura na Serasa Experian.
[3] Coorientadora. Doutora em Alimentos e Nutrição pela Faculdade de Ciências Farmacêuticas de Araraquara (FCFAr/UNESP), Mestre em Desenvolvimento Regional e Meio Ambiente pela Universidade de Araraquara (UNIARA), Economista e Bacharel em Direito. Docente do curso de Sistemas de Informação da Universidade de Araraquara (UNIARA), Araraquara-SP.

ABSTRACT: This work presents GymTrainer, a mobile strength-training application that acts as a virtual personal trainer. It uses artificial intelligence (AI) to build workouts and tracks the user while they train. The system has two parts: the app, built with React Native (Expo), and a server with a Node.js/Express REST API, which stores data in MongoDB and sends the user's profile to Google Gemini to generate the workouts. The app was evaluated through functional tests, an automated generation test with five profiles and a preliminary test with two real users. All 15 generations tested were completed in the correct format, after two failures caused by the usage limit of the AI service were repeated, and all weekly plans had the right number of workouts. The average response time was 2.1 seconds for single workouts and 4.9 seconds for weekly plans. According to American College of Sports Medicine criteria, 80.4% of the exercises had repetitions suited to the goal and 99.4% had sets suited to the level, and the profile with a knee injury received no exercises to avoid. The results show that it is possible to combine AI with workout tracking in a low-cost app, but the workouts still need to be evaluated by physical education professionals.

Keywords: Artificial intelligence. Mobile application. React Native. Strength training. Virtual personal trainer.

1 INTRODUÇÃO

A musculação traz muitos benefícios para a saúde física e mental, mas boa parte das pessoas não tem acompanhamento de um profissional, seja pelo preço, seja pela falta de horário de um personal trainer. Ao mesmo tempo, a inteligência artificial (IA) generativa evoluiu muito nos últimos anos e hoje consegue montar textos organizados a partir dos dados de uma pessoa. Isso abriu a possibilidade de automatizar parte do planejamento de treino, que antes dependia apenas de um profissional. Por isso, a pergunta deste trabalho é: é possível criar um aplicativo de celular que gere treinos de musculação personalizados com IA, acompanhe o usuário enquanto ele treina e o oriente durante o treino, usando ferramentas de baixo custo?

O objetivo geral deste trabalho é desenvolver um aplicativo de celular para treinos de musculação que use IA para criar treinos personalizados e orientar o usuário durante o treino, registrando o que ele realmente fez. Para isso, os objetivos específicos são criar um sistema seguro de cadastro e login, com acesso controlado por token; permitir que o usuário cadastre seu perfil físico e seus objetivos, que servem de base para montar os treinos; ligar o aplicativo a uma IA generativa para criar treinos avulsos e planos para a semana inteira; acompanhar a execução

do treino, registrando séries, repetições e carga, com cronômetro automático de descanso entre as séries; criar um assistente de conversa (personal trainer virtual) na tela do treino, que puxe conversa com o usuário e responda suas perguntas; e separar os treinos que ainda não foram feitos dos treinos já concluídos, mostrando estatísticas ao usuário.

O tema foi escolhido por dois motivos. Primeiro, muitas pessoas não têm acesso a um profissional que acompanhe seus treinos, e muitos aplicativos ainda oferecem treinos prontos e iguais para todos, enquanto a IA hoje já consegue montar planos de treino a partir do perfil de uma pessoa. Segundo, o projeto permite aplicar na prática o que foi aprendido no curso de Sistemas de Informação, como a criação de servidores e APIs, o uso de serviços de IA, bancos de dados não relacionais e o desenvolvimento de aplicativos para celular. Além disso, como o aplicativo registra o que o usuário realmente fez, e não só o treino sugerido, esses dados podem ser usados no futuro para ajustar a carga e a quantidade de treino automaticamente.

A hipótese é que, com ferramentas de baixo custo, é possível unir a geração de treinos por IA ao acompanhamento do treino, e que a IA, ao receber os dados do perfil do usuário de forma organizada (prompt) e ter sua resposta conferida pelo sistema, gera treinos de acordo com o perfil informado. A proposta é oferecer uma orientação contínua parecida com a de um personal trainer, mas sem substituí-lo.

Quanto à metodologia, esta é uma pesquisa aplicada, pois seu resultado é uma solução prática: o aplicativo GymTrainer. Também é exploratória, pois busca entender melhor como a IA pode ser usada nesse contexto, e qualitativa, pois os resultados foram analisados principalmente por observação, com apoio de indicadores simples, como porcentagens e tempos médios. Primeiro, foi feita uma pesquisa bibliográfica para embasar o trabalho. Depois, o aplicativo e o servidor foram desenvolvidos aos poucos, em ciclos: cada funcionalidade era criada, testada e melhorada antes da próxima.

A avaliação foi feita em três etapas. Na primeira, o desenvolvedor testou manualmente os principais caminhos de uso do aplicativo, comparando o resultado obtido com o esperado: cadastro e login, preenchimento do perfil, geração de treinos, execução do treino com registro das séries, conversa com o personal trainer virtual, consulta ao histórico e troca automática entre modelos de IA, testada simulando falhas no modelo principal.

Na segunda etapa, um script em Node.js usou a própria API do servidor para cadastrar cinco usuários de teste com perfis diferentes, variando nível, objetivo, dias por semana, local e equipamentos, sendo um deles com lesão no joelho (Quadro 1). Para cada perfil, foram gerados

dois treinos avulsos e um plano semanal, num total de 15 chamadas reais ao Gemini. Na primeira rodada, as duas últimas chamadas falharam por limite de uso do serviço de IA; por isso, o perfil com lesão no joelho foi testado de novo, sozinho, alguns minutos depois, e os resultados finais consideram essa segunda rodada. O script conferiu automaticamente o formato, isto é, se a resposta veio em JSON válido, com nome do treino e lista de exercícios com nome, séries e repetições; o número de dias, isto é, se o plano semanal trouxe exatamente um treino para cada dia disponível; a quantidade de exercícios por treino; e o tempo de resposta de cada chamada. Ao final, salvou os resultados e os treinos completos em arquivos e apagou as contas de teste.

Para avaliar se os treinos combinavam com o perfil, os 29 treinos gerados foram conferidos pelo autor, com apoio de um script, a partir de critérios baseados nas recomendações do American College of Sports Medicine para treinos de força (AMERICAN COLLEGE OF SPORTS MEDICINE, 2009). Os critérios foram: objetivo, em que a faixa de repetições sugerida deveria ficar entre 8 e 12 nos perfis de hipertrofia de iniciantes e intermediários, entre 6 e 12 no de avançados e em 10 ou mais nos perfis de emagrecimento e condicionamento, tomando como referência a faixa de 10 a 15 repetições recomendada para resistência muscular, considerando o valor médio da faixa e desconsiderando exercícios medidos em segundos; nível, com até 3 séries por exercício para iniciantes e pelo menos 3 para intermediários e avançados; equipamentos, em que só poderiam aparecer exercícios possíveis com o que foi informado; tempo, em que a duração estimada do treino, considerando 40 segundos por série mais o descanso sugerido, deveria caber no tempo disponível; distribuição, em que o plano semanal não deveria repetir o mesmo grupo muscular em dias seguidos; e limitação, em que o perfil com lesão no joelho, cuja observação pedia para evitar agachamento profundo, saltos e impacto, não deveria receber esses exercícios.

Na terceira etapa, o aplicativo foi testado por dois usuários reais, escolhidos por conveniência entre familiares do autor e que não participaram do desenvolvimento. Eles usaram o aplicativo para fazer o cadastro, preencher o perfil e gerar e executar treinos, e suas opiniões e dificuldades foram anotadas pelo autor. Por se tratar de um teste inicial de funcionamento do software, com participação voluntária, o estudo não foi submetido a comitê de ética em pesquisa, e nenhum dado pessoal dos participantes é apresentado neste trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 Aplicativos de treino e inteligência artificial

Os aplicativos de celular são muito usados na área fitness para ajudar na prática de exercícios e organizar a rotina de treinos, com recursos como cronogramas, controle de séries, histórico e acompanhamento de metas. Trabalhos recentes sobre aplicativos para academias mostram que eles ajudam a melhorar a experiência do usuário e a organização dos treinos (ARAUJO, 2024; GUIMARÃES; BATISTA, 2024).

A IA aplicada aos treinos consegue considerar ao mesmo tempo várias informações, como nível físico, objetivo, frequência e evolução da pessoa. Porém, Oliveira e Fraga (2021) fazem uma crítica: uma pessoa não se resume a esses dados, e o acompanhamento humano continua necessário para perceber coisas que a máquina não percebe. Roik (2025) também mostra que aplicativos com IA, como Freeletics e Fitbod, ajustam os treinos de acordo com o desempenho e o histórico do usuário, mas alerta que, se a IA não levar em conta as limitações de cada pessoa, pode gerar treinos inadequados.

Um trabalho parecido com este é o de Pelizzoni (2024), que criou um sistema web para academias em que os dados do aluno são enviados a uma IA da OpenAI (GPT-3.5 Turbo), que sugere um treino para o instrutor revisar. O autor conclui que a IA pode economizar tempo dos profissionais, desde que eles continuem revisando as sugestões. A diferença do GymTrainer é que ele é feito para o próprio aluno e, além de gerar o treino, acompanha sua execução e oferece um assistente de conversa.

Alguns estudos já avaliaram treinos criados por chatbots de IA. Washif et al. (2024) observaram que o ChatGPT monta programas de musculação organizados e adequados ao nível do praticante, mas que ainda precisam de ajustes de um profissional para serem usados. Dergaa et al. (2024) pediram ao GPT-4 programas para cinco perfis de pacientes e concluíram que eles eram seguros, porém pouco precisos para condições de saúde específicas e cautelosos demais. Zaleski et al. (2024) verificaram que 90,7% das recomendações de exercício de um chatbot estavam corretas, mas que elas cobriam apenas 41,2% dos itens recomendados pelo American College of Sports Medicine (ACSM) e eram difíceis de ler.

Nesses estudos, a pessoa conversa livremente com o chatbot, e a qualidade da resposta depende do que ela sabe pedir; o treino sai em texto solto, que não é registrado nem acompanhado. O GymTrainer se diferencia desses trabalhos em três pontos: o próprio sistema

monta o prompt a partir do perfil, sempre com as mesmas informações, inclusive limitações e equipamentos; a IA devolve o treino em um formato fixo, que o aplicativo consegue mostrar, executar e registrar; e o usuário conta com um assistente que conhece o treino em andamento. Em relação a Pelizzoni (2024), o GymTrainer é voltado ao próprio praticante e funciona no celular. Já os aplicativos comerciais citados por Roik (2025) ajustam automaticamente a carga com base no desempenho, recurso que o GymTrainer ainda não tem, embora já registre as séries, repetições e cargas necessárias para isso.

2.2 Modelos de linguagem e inteligência artificial generativa

Os grandes modelos de linguagem são programas de IA treinados com uma enorme quantidade de textos. A maioria usa uma estrutura chamada Transformer (VASWANI et al., 2017) e consegue realizar várias tarefas apenas a partir de instruções escritas, chamadas de prompts (BROWN et al., 2020). O Gemini, do Google, é um desses modelos e pode devolver respostas em um formato organizado, como o JSON (GOOGLE, [s.d.]). Assim, um sistema pode pedir à IA um plano de treino completo e personalizado, em vez de depender de regras fixas programadas à mão.

Por outro lado, esses modelos às vezes inventam informações que parecem certas, mas estão erradas, o que é chamado de alucinação (JI et al., 2023). Quando se trata de treinos, isso mostra que é importante enviar à IA dados claros sobre o usuário, conferir as respostas antes de mostrá-las e manter o profissional de educação física como referência.

2.3 Tecnologias de desenvolvimento e segurança

Um aplicativo de celular pode ser feito de duas formas: nativa, com um código diferente para cada sistema (Android e iOS), ou multiplataforma, com um único código que funciona nos dois. O React Native, mantido pela Meta, permite criar as telas em JavaScript e funciona nos dois sistemas (REACT NATIVE, [s.d.]). Em comparação com outras ferramentas parecidas, como o Flutter, o React Native se destaca por ter uma comunidade grande e por permitir usar muitas bibliotecas JavaScript prontas (SANTOS, 2025). O Expo, um conjunto de ferramentas feito para o React Native, facilita testar o aplicativo no celular e gerar a versão final (EXPO, [s.d.]).

Na arquitetura cliente-servidor, as regras do sistema e os dados ficam em um servidor, e o cliente (neste caso, o aplicativo) se comunica com ele para usar essas funções. O padrão

REST define regras para essa comunicação pela internet, usando comandos como GET, POST, PUT e DELETE em endereços específicos (FIELDING, 2000). O Node.js permite rodar JavaScript no servidor (OPENJS FOUNDATION, [s.d.]), e o Express é um framework que organiza as rotas de uma API feita em Node.js (EXPRESS, [s.d.]).

Os bancos de dados NoSQL guardam as informações de forma mais flexível que as tabelas dos bancos tradicionais. Por isso, são bons para dados que mudam de formato ou que têm listas dentro deles, como um treino com vários exercícios. O MongoDB guarda os dados em documentos parecidos com JSON, organizados em coleções, e é muito usado em aplicativos web e de celular (MONGODB, [s.d.]).

Para o login, o JSON Web Token (JWT) é um padrão em que, depois que o usuário informa e-mail e senha corretos, o servidor cria um código assinado (token), que o aplicativo envia nas próximas requisições para provar quem é o usuário, sem precisar mandar a senha de novo (JONES; BRADLEY; SAKIMURA, 2015). Já o bcrypt é um método para proteger senhas: ele transforma a senha em um código que não pode ser desfeito e que é difícil de descobrir por tentativa e erro, mesmo que alguém tenha acesso ao banco de dados (PROVOS; MAZIÈRES, 1999).

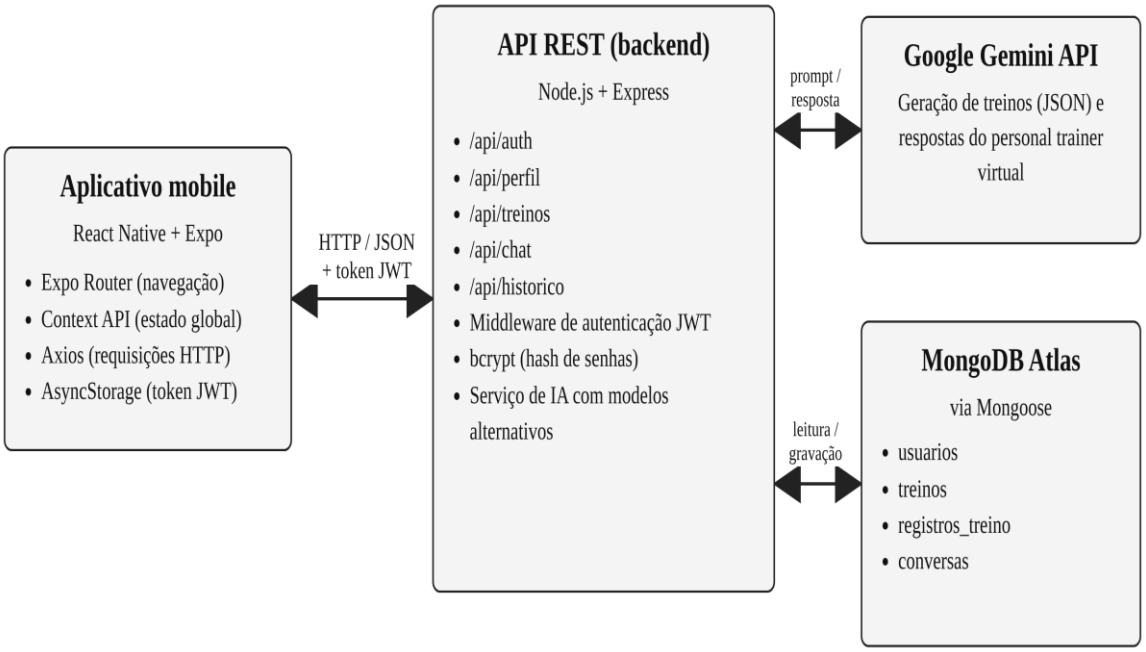
3 DESENVOLVIMENTO

Esta seção mostra como o GymTrainer foi construído, como cada funcionalidade funciona e como as partes do sistema se comunicam.

3.1 Arquitetura geral

O sistema segue a arquitetura cliente-servidor, mostrada na Figura 1. O aplicativo nunca acessa diretamente o banco de dados nem a IA: tudo passa pelo servidor (backend). Assim, a chave de acesso ao Gemini fica guardada apenas no servidor e não aparece no código do aplicativo.

Figura 1 – Arquitetura do sistema GymTrainer



Fonte: o autor, 2026.

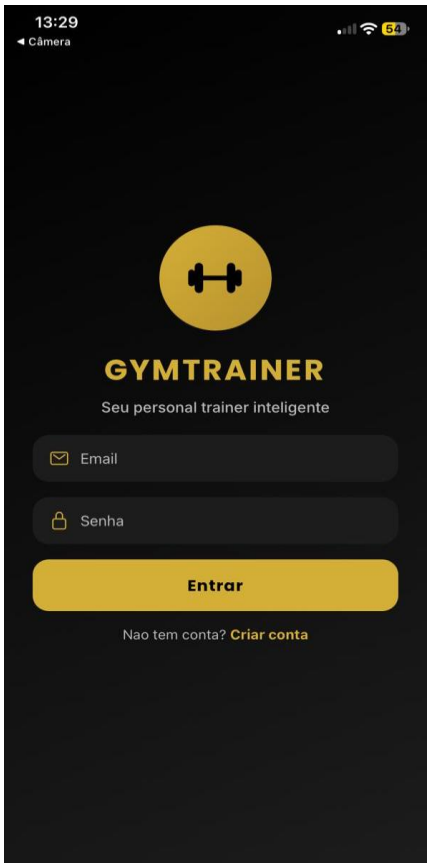
O servidor oferece uma API REST dividida em grupos de rotas: /api/auth, para cadastro e login; /api/perfil, para os dados do usuário; /api/treinos, para gerar e consultar os treinos; /api/chat, para a conversa com o personal trainer virtual; e /api/historico, para os treinos concluídos. Por exemplo, para gerar um treino: o aplicativo envia um pedido ao servidor junto com o token de login; o servidor confere o token, busca o perfil do usuário no MongoDB e monta um texto (prompt) descrevendo esse perfil; esse texto é enviado ao Gemini, que responde com o treino em formato JSON; o servidor confere a resposta, salva o treino no banco e o devolve ao aplicativo, que mostra a tela do treino gerado.

No aplicativo, foram usados o React Native com Expo, o Expo Router para a navegação entre telas, a Context API do React para guardar os dados usados em várias telas, o Axios para a comunicação com a API e o AsyncStorage para manter o token de login salvo no celular, para que o usuário continue conectado mesmo depois de fechar o aplicativo. O React Native foi escolhido pela familiaridade do autor com o JavaScript e pela grande comunidade de suporte (SANTOS, 2025). No servidor, foram usados o Node.js com Express, o MongoDB Atlas (banco de dados na nuvem) com a biblioteca Mongoose, tokens JWT e o bcrypt.

3.2 Cadastro, autenticação e segurança

O usuário cria uma conta com nome, e-mail e senha e entra no aplicativo pela tela de login, mostrada na Figura 2. Antes de ser salva, a senha é transformada em um código pelo bcrypt, então o servidor nunca guarda a senha original. No login, a senha digitada passa pelo mesmo processo e é comparada com o código salvo. Se estiver correta, o servidor cria um token JWT, que o aplicativo envia em todos os pedidos seguintes para provar quem é o usuário. O token vale por sete dias; depois disso, é preciso fazer login de novo.

Figura 2 – Tela de login do GymTrainer



Fonte: o autor, 2026.

Outras medidas de segurança complementam o login. O bcrypt é usado com 10 rodadas, e o segredo usado para assinar os tokens fica em uma variável de ambiente no servidor. Com exceção do cadastro, do login e de uma rota de verificação do servidor, todas as rotas da API exigem um token válido. A chave da API do Gemini também fica em uma variável de ambiente, em um arquivo que não é enviado ao repositório de código, e não aparece em nenhuma parte do

aplicativo. Para a IA são enviados apenas os dados necessários para montar o treino, como idade, peso, altura, objetivo, nível, rotina, equipamentos e limitações, sem nome ou e-mail.

3.3 Perfil do usuário

No primeiro acesso, o usuário preenche um formulário com seu perfil (Figura 3): idade, sexo, peso, altura, objetivo (hipertrofia, emagrecimento, definição ou condicionamento), nível (iniciante, intermediário ou avançado), dias disponíveis por semana, tempo de cada treino, local de treino, equipamentos disponíveis e possíveis limitações ou lesões. Essas informações são usadas pela IA para personalizar os treinos e podem ser editadas depois.

Figura 3 – Tela de cadastro do perfil do usuário



Fonte: o autor, 2026.

3.4 Geração de treinos com inteligência artificial

O treino pode ser gerado de duas formas, pelos botões da tela inicial (Figura 4). Na primeira, o usuário pede um único treino e pode escolher o grupo muscular que quer treinar naquele dia. O servidor monta um prompt com todo o perfil e pede ao Gemini uma resposta em JSON, com o nome do treino, os grupos musculares e a lista de exercícios, cada um com séries, repetições, tempo de descanso e observações. Na segunda, o usuário pede o plano da semana

inteira, e o prompt pede à IA, de uma só vez, um treino para cada dia disponível do usuário, dividindo bem os grupos musculares ao longo da semana. Os treinos gerados ficam na tela de treinos (Figura 5) até serem feitos.

O texto a seguir é o prompt usado para gerar um treino avulso, copiado do código do servidor; os campos entre chaves são preenchidos com os dados de cada usuário:

> Você é um personal trainer especialista. Gere um treino personalizado com base no perfil abaixo e responda APENAS com um JSON válido, sem texto adicional, seguindo exatamente este formato:
> {"nomeTreino": "string", "gruposMusculares": ["string"], "exercicios": [{"nome": "string", "grupoMuscular": "string", "series": number, "repeticoes": "string (ex: 8-12)", "descansoSegundos": number, "observacoes": "string"}], "observacoes": "string com dicas gerais sobre o treino"}
> Perfil do aluno: Idade: {idade} anos. Peso: {peso} kg. Altura: {altura} cm. Objetivo: {objetivo}. Nível: {nível}. Dias disponíveis por semana: {dias}. Tempo por treino: {tempo} minutos. Local de treino: {local}. Equipamentos disponíveis: {equipamentos ou "nenhum"}. Lesões ou limitações: {limitações ou "nenhuma"}. Observação adicional do aluno: {observação ou "nenhuma"}.

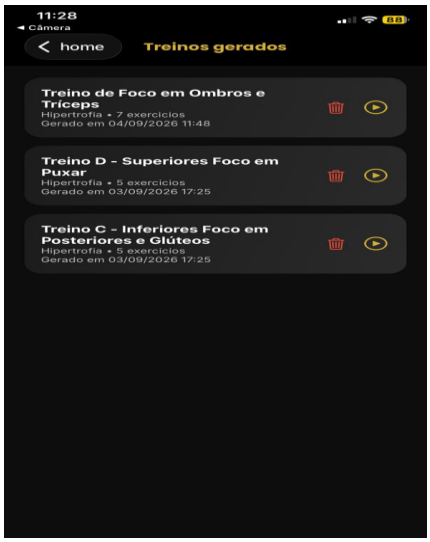
No plano semanal, o prompt segue a mesma estrutura, mas pede uma lista com exatamente o número de treinos igual aos dias disponíveis e repete esse pedido três vezes no texto. Quando a resposta chega, o servidor confere se ela é um JSON válido e, no plano semanal, se é uma lista não vazia; se a IA enviar mais treinos do que o pedido, os excedentes são descartados. Se a resposta não for válida, nada é salvo e o usuário recebe uma mensagem para tentar de novo.

Figura 4 – Tela inicial do GymTrainer



Fonte: o autor, 2026.

Figura 5 – Lista de treinos gerados pela IA



Fonte: o autor, 2026.

3.5 Execução do treino e registro de desempenho

Ao abrir um treino, o usuário vê os exercícios com as séries, repetições e tempo de descanso sugeridos (Figura 6). Quando ele aperta o botão Iniciar treino, o progresso passa a ficar guardado no aplicativo, então ele pode ir para outras telas, como o chat, e voltar sem perder

nada. A cada série feita, o usuário informa quantas repetições fez e qual carga usou. Esse registro é salvo no servidor, separado do treino sugerido pela IA, e o cronômetro de descanso começa sozinho, com aviso visual e sonoro no final, podendo ser pausado, retomado ou pulado. Ao terminar, o usuário confirma no botão Treino concluído, e o treino passa para o histórico.

Figura 6 – Tela de execução do treino



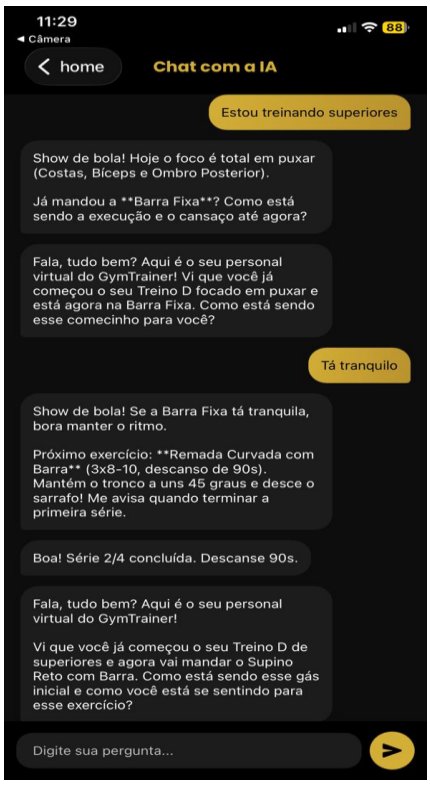
Fonte: o autor, 2026.

3.6 Personal trainer virtual

Durante o treino, aparece na tela um ícone (avatar) do personal trainer virtual, que pode ser arrastado e se encaixa sozinho na borda mais próxima, parecido com o AssistiveTouch do iPhone. Quando o usuário termina a primeira série, o servidor chama o Gemini para que o assistente puxe conversa, considerando em que parte do treino o usuário está (início, meio ou fim), e a mensagem aparece em um balão perto do avatar. Nas séries seguintes, para economizar chamadas à IA, o assistente mostra mensagens curtas criadas pelo próprio aplicativo, que também ficam salvas no histórico do chat.

Ao tocar no avatar, o usuário abre o chat completo (Figura 7), onde pode tirar dúvidas sobre a execução dos exercícios, cansaço, dor ou técnica. Para responder, a IA recebe o perfil do usuário, o treino atual e as últimas mensagens da conversa, o que deixa as respostas de acordo com a situação do usuário. O assistente não muda o treino salvo por conta própria: o prompt do chat pede respostas curtas em português e proíbe a IA de alterar ou confirmar alterações no treino salvo, permitindo apenas sugestões, como reduzir a carga, trocar um exercício ou descansar mais.

Figura 7 – Chat com o personal trainer virtual

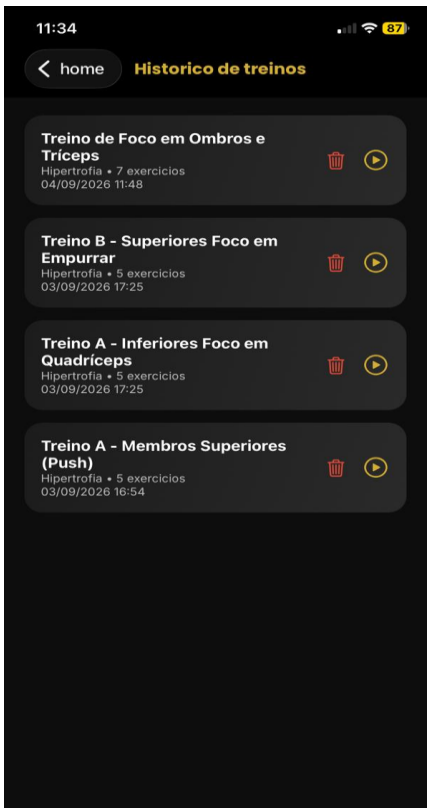


Fonte: o autor, 2026.

3.7 Histórico e estatísticas de desempenho

O aplicativo separa os treinos que ainda não foram feitos dos treinos concluídos, usando uma marcação no treino que só muda quando o usuário confirma que terminou. Cada treino do histórico (Figura 8) mostra o que a IA sugeriu e o que o usuário realmente fez. A tela inicial também mostra o total de treinos concluídos e quantos dias seguidos o usuário treinou. Essa sequência aumenta quando o último treino foi feito no dia anterior, continua igual quando o usuário já treinou no mesmo dia e volta para um quando ele fica dois dias ou mais sem treinar.

Figura 8 – Histórico de treinos concluídos



Fonte: o autor, 2026.

3.8 Estrutura de dados e resiliência da integração com a IA

Os dados ficam guardados em quatro coleções no MongoDB: usuarios, com os dados da conta e do perfil; treinos, com os planos gerados pela IA (exercícios, séries, repetições, descanso e observações) e a marcação de concluído; registros_treino, com as séries que o usuário realmente fez, com as repetições e a carga de cada uma; e conversas, com as mensagens trocadas com o personal trainer virtual.

Como o sistema depende de um serviço externo de IA, o servidor foi preparado para quando esse serviço falhar: se o modelo principal estiver sobrecarregado ou atingir o limite de uso, o sistema tenta automaticamente o próximo modelo de uma lista de modelos alternativos do Gemini, esperando no máximo 12 segundos em cada tentativa, antes de avisar o aplicativo que houve uma falha. Outros tipos de erro não acionam essa troca.

4 RESULTADOS

O resultado deste trabalho é um aplicativo que funciona do começo ao fim: cadastro e login, preenchimento do perfil, geração de treinos personalizados (avulsos e semanais) pela IA,

acompanhamento do treino com registro das séries, cronômetro de descanso, assistente de conversa e separação entre treinos pendentes e histórico, com as estatísticas do usuário. As Figuras 2 a 8 mostram as principais telas.

O cadastro e o login funcionaram como esperado: as senhas foram salvas apenas como hash, e as rotas protegidas da API só responderam quando o aplicativo enviou um token JWT válido. Depois do cadastro, o usuário foi levado ao preenchimento do perfil, que é necessário para gerar os treinos.

A geração de treinos pela IA funcionou tanto no treino avulso quanto no plano semanal. No teste automatizado, considerando a segunda rodada do perfil com lesão no joelho, as 15 chamadas foram concluídas e geraram 29 treinos: 10 avulsos e 19 dentro de cinco planos semanais. Todas as respostas vieram em JSON válido, e os cinco planos semanais trouxeram exatamente um treino para cada dia disponível. Cada treino teve entre 5 e 7 exercícios. O tempo médio de resposta foi de 2,1 segundos para treinos avulsos, variando de 1,9 a 2,6 segundos, e de 4,9 segundos para planos semanais, variando de 3,8 a 6,6 segundos; os planos com mais dias demoraram mais para ser gerados. O Quadro 1 mostra os resultados por perfil.

Quadro 1 – Resultados do teste automatizado de geração de treinos, por perfil de teste

Perfil de teste	Chamadas concluídas	Treinos gerados	Exercícios por treino	Tempo médio avulso (s)	Tempo do plano semanal (s)
Iniciante, hipertrofia, 3 dias, academia	3 de 3	5	5 a 7	2,24	3,80
Intermediário, emagrecimento, 4 dias, academia	3 de 3	6	5 a 6	2,02	5,13
Avançado, hipertrofia, 5 dias, academia	3 de 3	7	5 a 6	2,06	6,61
Iniciante, condicionamento, 3 dias, em casa com halteres	3 de 3	5	5 a 6	2,09	4,08
Intermediário, hipertrofia, 4 dias, academia, lesão no joelho*	3 de 3	6	5 a 7	2,31	4,81
Total	15 de 15 (100%)	29	5 a 7	2,14	4,89

Fonte: o autor, 2026.

*Resultados da segunda rodada; na primeira, 2 das 3 chamadas falharam por limite de uso do serviço de IA.

Na primeira rodada, porém, o segundo treino avulso e o plano semanal do perfil com lesão no joelho, que foi o último a ser testado, falharam depois de 24,4 e 22,6 segundos. Os registros do servidor mostraram que, até ali, já tinham sido feitas cerca de 13 chamadas seguidas, o que esgotou o limite de 15 requisições por minuto do plano gratuito do Gemini no modelo principal (erro 429). O servidor então tentou os modelos alternativos, que estavam

sobrecarregados naquele momento (erro 503) ou não responderam dentro de 12 segundos. Nesses casos, nada foi salvo e o aplicativo recebeu uma mensagem de erro. Quando o mesmo perfil foi testado de novo, sozinho, as três gerações deram certo. Nesse perfil, o treino avulso foi nomeado pela IA como "Treino Superior Foco Hipertrofia (Proteção para Joelho)" e trouxe apenas exercícios de peito, costas, ombros, bíceps e tríceps, sem nenhum exercício de pernas, com observações voltadas à segurança, como "Mantenha a coluna neutra para proteger a lombar" na remada curvada com barra. No plano semanal, os exercícios de pernas escolhidos foram de baixo impacto, como mesa flexora, cadeira extensora com "amplitude limitada" e leg press 45 com a indicação "(Com Cuidado no Joelho)" no próprio nome, e quatro observações citaram diretamente o joelho ou a perna direita.

Na avaliação dos 29 treinos pelos critérios do ACSM, 123 dos 153 exercícios medidos em repetições (80,4%) ficaram na faixa esperada para o objetivo. Nenhum exercício dos perfis de hipertrofia ficou abaixo de 6 repetições: os que ficaram fora da faixa usaram de 12 a 20 repetições, quase sempre em exercícios isolados, como elevação lateral e panturrilhas. Nos perfis de emagrecimento e condicionamento, os 7 exercícios fora da faixa usaram de 8 a 10 repetições em exercícios que trabalham vários músculos ao mesmo tempo. O número de séries combinou com o nível em 158 dos 159 exercícios (99,4%); a única exceção foi um exercício de panturrilha com 4 séries para um iniciante. No perfil que treina em casa só com halteres, nenhum dos 27 exercícios exigiu máquina, barra ou polia. Todos os 29 treinos couberam no tempo disponível, com duração estimada entre 23 e 48 minutos, o que corresponde a 44% a 71% do tempo informado. Em quatro dos cinco planos semanais, nenhum grupo muscular se repetiu em dias seguidos; no plano do perfil avançado, os ombros apareceram em dois dias seguidos, em exercícios diferentes. No perfil com lesão no joelho, nenhum dos seis treinos trouxe agachamento profundo, saltos ou exercícios de impacto. O Quadro 2 resume essa avaliação.

Quadro 2 – Avaliação dos 29 treinos gerados pelos critérios baseados no ACSM

Critério	Regra usada	Resultado
Objetivo (repetições)	Hipertrofia: 8 a 12 (iniciante e intermediário) e 6 a 12 (avançado); emagrecimento e condicionamento: 10 ou mais	123 de 153 exercícios (80,4%)
Nível (séries)	Iniciante: até 3; intermediário e avançado: 3 ou mais	158 de 159 exercícios (99,4%)
Equipamentos	Só exercícios possíveis com o que foi informado	27 de 27 exercícios do perfil em casa (100%)
Tempo	Duração estimada dentro do tempo disponível	29 de 29 treinos (100%)
Distribuição semanal	Sem o mesmo grupo muscular em dias seguidos	4 de 5 planos (80%)
Limitação (joelho)	Sem agachamento profundo, saltos ou impacto	6 de 6 treinos (100%)

Fonte: o autor, 2026.

Na execução do treino, o registro das séries, o cronômetro de descanso e a conclusão do treino funcionaram corretamente. Como as séries feitas são salvas separadas do treino gerado, o plano da IA continua guardado como foi sugerido, independentemente do que o usuário fez em cada dia. Isso também permitiu separar a tela de Treinos, com os treinos pendentes, da tela de Histórico, com os treinos concluídos.

O personal trainer virtual puxou conversa depois da primeira série, levando em conta o momento do treino e sem dizer que o treino já tinha acabado. As mensagens seguintes, mais simples, foram criadas pelo próprio aplicativo, o que diminuiu as chamadas à IA sem quebrar a continuidade da conversa. No chat livre, a IA respondeu às dúvidas sobre execução, cansaço e dificuldade levando em conta o perfil e o treino em andamento.

As estatísticas de treinos concluídos e de dias seguidos foram atualizadas corretamente a cada treino concluído. O sistema de modelos alternativos foi testado simulando falhas no modelo principal do Gemini, e o servidor passou automaticamente para o próximo modelo da lista, como esperado. O Quadro 3 resume as funcionalidades e como foram testadas.

Quadro 3 – Funcionalidades implementadas e situação da validação

Funcionalidade	Situação da validação
Cadastro e autenticação (JWT)	Implementada e testada
Preenchimento do perfil físico	Implementada e testada
Geração de treino avulso pela IA	Implementada e testada
Geração de plano semanal pela IA	Implementada e testada
Execução de treino e registro de séries	Implementada e testada
Separação entre Treinos e Histórico	Implementada e testada
Personal trainer virtual proativo	Implementada e testada
Chat livre com o personal trainer virtual	Implementada e testada
Estatísticas de uso (treinos e sequência de dias)	Implementada e testada
Resiliência por modelos de IA alternativos	Implementada e testada
Teste exploratório com usuários reais (2 participantes)	Realizado (preliminar)

Fonte: o autor, 2026.

Além dos testes feitos pelo desenvolvedor, o aplicativo foi usado por dois usuários reais, familiares do autor, que não participaram do desenvolvimento. Os dois conseguiram fazer o cadastro, preencher o perfil e gerar e fazer treinos. Como foram só duas pessoas próximas ao autor, esse teste é apenas inicial e não permite tirar conclusões gerais, mas ajudou a ver o aplicativo funcionando no uso real.

Durante o desenvolvimento, percebeu-se que separar o treino sugerido pela IA do que o usuário realmente fez foi importante para manter o histórico correto. Também foi necessário guardar o progresso do treino em um lugar comum a todo o aplicativo, e não em cada tela, para não perder o progresso ao trocar de tela. Por fim, percebeu-se que depender de um único modelo de IA deixava o sistema parado quando o serviço falhava, o que levou à criação do sistema de modelos alternativos descrito na seção 3.8.

De modo geral, os resultados mostram que a IA, quando recebe um prompt organizado com os dados do perfil do usuário e tem sua resposta conferida pelo servidor, gera treinos no formato esperado e de acordo com as informações fornecidas, o que confirma a hipótese dentro dos testes realizados. Os tempos de resposta, de poucos segundos, são adequados para uso no celular, e as falhas da primeira rodada não foram erros da IA ao montar o treino, mas efeito do limite de requisições por minuto do plano gratuito do Gemini somado a uma sobrecarga momentânea dos modelos alternativos. Em uso real, com vários usuários ao mesmo tempo, esse limite seria atingido com facilidade, o que exigiria um plano pago ou um controle do número de chamadas. O caso do perfil com lesão no joelho sugere que enviar as limitações de forma explícita ajuda a IA a considerá-las, ponto em que Dergaa et al. (2024) encontraram pouca precisão, embora um único exemplo não seja suficiente para comprovar isso. A avaliação pelos critérios do ACSM mostra que os treinos seguiram, na maior parte, as recomendações para o objetivo e o nível, respeitaram os equipamentos e a limitação informada e couberam no tempo disponível. Ela também mostra pontos a melhorar: os treinos ficaram bem mais curtos que o tempo disponível, e alguns exercícios de emagrecimento e condicionamento usaram poucas repetições. Esses ajustes poderiam ser pedidos no próprio prompt, por exemplo informando a faixa de repetições de cada objetivo e pedindo que o treino ocupe melhor o tempo do usuário. Ainda assim, essa avaliação foi feita pelo autor, com base em faixas gerais, e não substitui a análise de um profissional de educação física.

5 CONCLUSÃO

A resposta à pergunta deste trabalho é sim: foi possível criar um aplicativo de celular que gera treinos de musculação com IA e acompanha o usuário durante o treino, usando ferramentas de baixo custo. Os objetivos específicos foram alcançados: foram criados o login seguro, o cadastro de perfil, a geração de treinos personalizados, o acompanhamento do treino com

registro do que foi feito, o assistente de conversa e a separação entre treinos pendentes e histórico.

Em relação aos trabalhos relacionados, o principal diferencial do GymTrainer é a forma de uso da IA: em vez de uma conversa livre com um chatbot, o próprio sistema envia à IA o perfil do usuário de forma organizada e recebe o treino em um formato que o aplicativo consegue mostrar, executar e registrar, junto com um assistente que conhece o treino em andamento.

Quanto à proteção dos dados, o protótipo já adota medidas básicas, como senhas com hash, rotas protegidas por token, chaves fora do código do aplicativo e envio à IA apenas dos dados necessários para o treino. Ainda assim, há pontos a melhorar antes de uma versão pública. Nos testes, o aplicativo se comunicou com o servidor por HTTP em uma rede local, sem criptografia; em uso real, essa comunicação precisaria ser feita por HTTPS. Também não há limite de requisições contra abusos nem uma opção para o usuário excluir a conta e os seus dados. Esse último ponto é importante porque o perfil guarda dados de saúde, como lesões, que a Lei Geral de Proteção de Dados Pessoais (LGPD) considera dados pessoais sensíveis e que exigem cuidado especial, incluindo o direito do titular de pedir a eliminação dos seus dados (BRASIL, 2018). Como esses dados também são enviados ao Gemini, uma versão pública precisaria pedir o consentimento do usuário e explicar como os dados são usados.

As recomendações da IA também têm limites. O servidor confere o formato da resposta, mas não confere cada campo, como se as séries são números ou se a lista de exercícios está vazia. Como a IA pode errar ou inventar informações (JI et al., 2023) e, segundo Dergaa et al. (2024), tende a ser imprecisa em condições de saúde específicas, os treinos gerados devem ser vistos como sugestões. O aplicativo limita o assistente a orientar, sem mudar o treino salvo, mas uma versão pública deveria mostrar um aviso de que as recomendações não substituem um profissional de educação física, como apontam Oliveira e Fraga (2021).

O trabalho também tem limitações de método. O teste automatizado foi feito com apenas cinco perfis e uma única rodada completa, e o teste com usuários reais teve apenas duas pessoas, familiares do autor, o que não permite generalizar os resultados. Os treinos não foram avaliados por profissionais de educação física, o aplicativo depende de um serviço de IA externo, que pode ficar fora do ar ou mudar de preço e de regras de uso, e não foram criados testes automatizados do código.

Como sugestões para trabalhos futuros, recomenda-se repetir os testes com mais perfis e mais usuários, durante várias semanas; informar no prompt as faixas de repetições de cada

objetivo e pedir que o treino aproveite melhor o tempo disponível; pedir que profissionais de educação física avaliem os treinos com critérios baseados nas recomendações do ACSM; conferir cada campo da resposta da IA; usar HTTPS, limite de requisições e uma opção de exclusão de conta; adequar o aplicativo à LGPD; criar testes automatizados; e usar o histórico de cargas e repetições para sugerir automaticamente o aumento de carga nos próximos treinos.

REFERÊNCIAS

AMERICAN COLLEGE OF SPORTS MEDICINE. Progression models in resistance training for healthy adults. **Medicine and Science in Sports and Exercise**, v. 41, n. 3, p. 687-708, 2009. DOI: 10.1249/MSS.0b013e3181915670.

ARAUJO, Gabriel Roveri de. Desenvolvimento de um aplicativo mobile para gerenciamento de treinos em academias. **Revista FT**, v. 29, ed. 140, nov. 2024. DOI: 10.69849/revistaft/ch10202411181857.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). **Diário Oficial da União**: Brasília, DF, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 2 nov. 2026.

BROWN, Tom B. et al. Language models are few-shot learners. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 34., 2020, Vancouver. **Advances in Neural Information Processing Systems**. Red Hook: Curran Associates, 2020. v. 33, p. 1877-1901.

DERGAA, Ismail 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, 2024. DOI: 10.5114/biolsport.2024.133661.

EXPO. **Expo documentation**. [S. l.]: Expo, [s.d.]. Disponível em: https://docs.expo.dev/. Acesso em: 2 nov. 2026.

EXPRESS. **Express framework documentation**. [S. l.]: OpenJS Foundation, [s.d.]. Disponível em: https://expressjs.com/. Acesso em: 2 nov. 2026.

FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000.

GOOGLE. **Gemini API documentation**. [S. l.]: Google AI for Developers, [s.d.]. Disponível em: https://ai.google.dev/gemini-api/docs. Acesso em: 2 nov. 2026.

GUIMARÃES, Carlos Eduardo Prado; BATISTA, Matheus Pereira Amaral. **FitFlow**: aplicativo para otimizar a comunicação no mercado fitness e incentivar práticas saudáveis. 2024. Trabalho de Graduação (Tecnologia em Informática para Negócios) – Faculdade de Tecnologia do Estado de São Paulo, 2024. Disponível em: https://ric.cps.sp.gov.br/handle/123456789/32924. Acesso em: 2 nov. 2026.

JI, Ziwei et al. Survey of hallucination in natural language generation. **ACM Computing Surveys**, v. 55, n. 12, art. 248, p. 1-38, 2023. DOI: 10.1145/3571730.

JONES, M. B.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. RFC 7519. [S. l.]: Internet Engineering Task Force (IETF), maio 2015. Disponível em: https://www.rfc-editor.org/rfc/rfc7519. Acesso em: 2 nov. 2026.

MONGODB. **MongoDB documentation**. [S. l.]: MongoDB Inc., [s.d.]. Disponível em: https://www.mongodb.com/docs/. Acesso em: 2 nov. 2026.

OLIVEIRA, Braulio Nogueira de; FRAGA, Alex Branco. Prescrição de exercícios físicos por inteligência artificial: a educação física vai acabar? **Revista Brasileira de Ciências do Esporte**, v. 43, e002921, 2021. DOI: 10.1590/rbce.43.e002921.

OPENJS FOUNDATION. **Node.js documentation**. [S. l.]: OpenJS Foundation, [s.d.]. Disponível em: https://nodejs.org/en/docs. Acesso em: 2 nov. 2026.

PELIZZONI, Natan Osvald. **Aplicação de gestão de treinos em academias com suporte de Inteligência Artificial (IA)**. 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Passo Fundo, Passo Fundo, 2024. Disponível em: https://repositorio.upf.br/handle/123456789/7497. Acesso em: 2 nov. 2026.

PROVOS, Niels; MAZIÈRES, David. A future-adaptable password scheme. In: USENIX ANNUAL TECHNICAL CONFERENCE, 1999, Monterey. **Proceedings** [...]. Berkeley: USENIX Association, 1999. Disponível em: https://www.usenix.org/legacy/events/usenix99/provos.html. Acesso em: 2 nov. 2026.

REACT NATIVE. **React Native documentation**. [S. l.]: Meta Open Source, [s.d.]. Disponível em: https://reactnative.dev/. Acesso em: 2 nov. 2026.

ROIK, Peterson. **Aplicativos e inteligência digital na educação física**: inovação na prescrição e monitoramento de atividades físicas. 2025. Monografia (Especialização em Treinamento de Força e Hipertrofia) – Setor de Ciências Biológicas, Universidade Federal do Paraná, Curitiba, 2025.

SANTOS, Darlan Kennedy Oliveira dos. A comparação de performance entre React Native e Flutter no desenvolvimento de aplicativos móveis. **Revista FT**, v. 29, ed. 151, out. 2025. DOI: 10.69849/revistaft/cl10202510211202.

VASWANI, Ashish et al. Attention is all you need. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31., 2017, Long Beach. **Advances in Neural Information Processing Systems**. Red Hook: Curran Associates, 2017. v. 30, p. 5998-6008.

WASHIF, Jad Adrian et al. Artificial intelligence in sport: exploring the potential of using ChatGPT in resistance training prescription. **Biology of Sport**, v. 41, n. 2, p. 209-220, 2024. DOI: 10.5114/biolsport.2024.132987.

ZALESKI, Amanda L. et al. Comprehensiveness, accuracy, and readability of exercise recommendations provided by an AI-based chatbot: mixed methods study. **JMIR Medical Education**, v. 10, e51308, 2024. DOI: 10.2196/51308.