

# Aplicação de gestão de treinos em academias com suporte de Inteligência Artificial(IA)

Natan Osvald Pelizzoni<sup>1</sup>, Jaqson Dalbosco<sup>2</sup>

<sup>1</sup> Instituto de Tecnologia – Universidade de Passo Fundo

178332@upf.br

**Abstract.** *This project presents a web application designed to support the management of gym training routines, incorporating integration with Artificial intelligence (AI). The system utilizes user data, such as age, weight, height, fitness goals, weekly frequency, health restrictions, and more, to generate personalized workout plans through the GPT-3.5 Turbo AI model from OpenAI. By automating the workout planning process, the application aims to optimize the time of fitness professionals, allowing them to focus on providing better guidance and support to gym members during their exercises. The project details the selection of technologies, their application, the design and implementation process, as well as the results obtained from validating the application in a real gym environment. It concludes that integrating AI into a management platform can be a valuable ally in optimizing professionals' time while maintaining expert oversight of the AI-generated suggestions.*

**Resumo.** *Este projeto apresenta uma aplicação web de apoio a gestão de treinos para academias, incorporando integração com Inteligência Artificial (IA). O sistema utiliza dados do usuário, como idade, peso, altura, objetivos fitness, frequência semanal e restrições de saúde, entre outros para gerar planos de treino personalizados por meio do modelo de IA GPT-3.5 Turbo da empresa OpenAI. Ao automatizar o processo de planejamento de treinos, a aplicação tem como objetivo otimizar o tempo dos profissionais de fitness, permitindo que se concentrem em oferecer melhor orientação e suporte aos membros da academia durante seus exercícios. São apresentadas informações sobre a escolha de tecnologias e sua aplicação, detalhes do projeto e implementação, assim como, os resultados obtidos com a validação da aplicação em uma academia real. Conclui-se que utilizar IA integrada à uma plataforma de gestão pode ser uma grande aliada na otimização de tempo dos profissionais, mantendo a supervisão do especialista sobre a sugestão da IA.*

## 1. Introdução

Atualmente em nosso cotidiano nos deparamos com alguns problemas na sociedade que vem prejudicando as pessoas de várias formas, uma delas é na área da saúde que acaba por afetar o bem-estar de cada pessoa. Alguns dos fatores para isso são a obesidade, o sedentarismo, má alimentação, entre outros hábitos do dia a dia que trazem malefícios para a saúde. Com todos esses problemas tem-se notado que há campanhas de conscientização para a realização de atividades físicas e ter hábitos mais saudáveis, nas quais refletem em uma qualidade de vida melhor.

Ao mesmo tempo, observa-se grandes avanços no campo da tecnologia, especialmente no que se refere a IA. Demonstrando ser muito eficiente e poderosa em diversas áreas, oferecendo auxílio, suporte e soluções inovadoras para uma variedade de atividades.

No atual cenário das academias, principalmente as de menor porte, a partir da observação realizada em algumas academias no início do trabalho, percebeu-se que ainda necessitam do básico para a sistematização do controle de treinos, o qual ainda segue sendo feito através de fichas impressas onde as atividades a serem realizadas são marcadas manualmente, muitas vezes de forma verbal, apenas falando qual exercício o cliente deve executar. Em alguns casos estas fichas acabam sendo descartadas ou extraviadas, resultando em perda do histórico do cliente e retrabalhos. Geralmente não possuem o controle das fichas utilizadas para registrar os exercícios que os clientes devem realizar durante as atividades. Isso na maior parte dos casos leva a perda ou ao descarte sem intenção da mesma. Para a elaboração da ficha de treino, usualmente levam cerca de 4 a 6 minutos do tempo do instrutor ou do *personal trainer* para cada cliente, onde o mesmo poderia estar auxiliando os clientes a corrigir a postura, mostrando como é executado o exercício que ele esteja com dúvida, entre outros afazeres dentro do estabelecimento.

Reconhecendo o problema enfrentado na maioria das academias, este artigo tem por objetivo o desenvolvimento de uma aplicação web para gerenciamento de treinos em academias, integrando uma IA para auxiliar profissionais da área na criação de planos de exercícios. A aplicação combinará dados coletados dos clientes durante o cadastro como por exemplo a idade, o peso, a altura, os objetivos, as restrições de saúde, entre outros. E com base nos dados fornecidos e na capacidade analítica da IA, será gerada uma ficha de treino personalizada e otimizada para um melhor desempenho do cliente, considerando os equipamentos disponíveis na academia.

Com esta solução, busca-se minimizar o tempo que o instrutor ou *personal trainer* leva para selecionar os exercícios que o cliente vai realizar durante os treinos. Para isso, a aplicação web permite o cadastro dos clientes e dos exercícios que podem ser realizados no estabelecimento. Na tela de criação da ficha de treino, existe um botão que, ao ser acionado, gera sugestões de exercícios com o suporte de uma IA. Todas as informações do cliente, juntamente com os exercícios cadastrados na academia, são utilizadas para montar um *prompt* de comando, que é enviado para a IA. Esta, por sua vez, retorna uma lista de exercícios que é exibida na tela, permitindo que o instrutor avalie se está adequada ao cliente. O instrutor também pode adicionar exercícios manualmente ou remover, caso necessário.

A estrutura deste artigo é delineada da seguinte maneira: a Seção 2 apresentará uma visão geral da aplicação, com uma descrição do sistema e suas funcionalidades principais, incluindo o uso da IA para a geração de treinos personalizados. A Seção 3 detalhará o desenvolvimento do projeto, com ênfase na integração do modelo de IA da OpenAI e nos principais desafios e soluções adotadas durante o desenvolvimento. Na Seção 4, serão exibidos os resultados obtidos, destacando as telas principais do sistema e como estas facilitam o processo de criação e gestão de treinos. A Seção 5 abordará a validação do sistema com feedback de profissionais da área de educação física, analisando a eficácia e as limitações do uso da IA para a montagem de treinos. Por fim, a Seção 6 discutirá as considerações finais, incluindo as dificuldades encontradas, as lições

aprendidas e sugestões para futuras melhorias e desenvolvimentos do sistema.

## **2. Trabalhos Relacionados**

Nesta seção, são brevemente apresentados trabalhos relacionados encontrados a partir da base de dados do Google Acadêmico [6], no mês de março de 2024, onde foi realizada uma busca por palavras chaves em português que abordam o desenvolvimento de um software para gestão de treinos em academias.

### **2.1. SOUZA, Eliezer Vinicius de et al. SGA: sistema gestão de academia**

O estudo [21] teve como objetivo a implementação de um software para facilitar a rotina dos frequentadores de uma academia. A partir de um código de barras único gerado para cada cliente pelo sistema, foi possível realizar o gerenciamento dos treinos, controlar a frequência no estabelecimento, monitorar os pagamentos realizados e exibir os exercícios do dia.

Para o desenvolvimento do sistema os autores utilizaram da linguagem de programação Delphi [5], pela possibilidade de gerar um software para dispositivos desktop que utilizem os sistemas operacionais Windows [20] e MacOs [11] bem como aplicativos para os dispositivos móveis que utilizem os sistemas operacionais Android [2] e iOS [8].

### **2.2. OLIVEIRA, Romário Luz de. Ficha de treino digital: software de apoio a alunos de academia**

Já o trabalho [18] teve por objetivo a criação de uma aplicação web com foco no auxílio dos praticantes de musculação e exercícios aeróbicos durante os treinos na academia. A motivação do autor foi desenvolver uma aplicação gratuita que pudesse ser acessada a partir de um desktop ou pelo smartphone durante os treinos. Outro recurso provido foi o histórico de treinos e relatório de evolução.

O desenvolvimento da solução obteve-se a partir da utilização da linguagem de programação JavaScript [9], da linguagem de marcação HTML [7] e de estilização CSS [4]. Além dessas ferramentas também foi utilizado o framework progressivo VueJS [16] na parte do *front-end*. E na implementação do *back-end* utilizou-se a plataforma Supabase [15].

### **2.3. DE ALMEIDA, Ariel Lopes; DA SILVA, Ricardo Frohlich. Desenvolvimento de um sistema para academias com acompanhamento da rotina de treino**

A pesquisa [17] apresenta o desenvolvimento de um aplicação web que visa o gerenciamento dos treinos de clientes em academias. Além da gestão do treino, foi possível fazer avaliações como o próprio autor menciona através dos anamneses, com o intuito de diagnosticar alguma lesão. Com isso montar o treino de forma personalizada para o praticante, e assim ter um acompanhamento e evolução dos clientes. Além da aplicação web, os autores também desenvolveram uma aplicação mobile onde o cliente pudesse acompanhar seus dados.

Para a construção de da aplicação, o autor optou por utilizar o framework Laravel [10] utilizando como principal linguagem o PHP[13]. Juntamente com o pacote de estilos AdminLTE [1] para que a construção visual seja feita de forma mais fácil por possuir um

vasto acervo de componentes prontos onde é necessário realizar somente a personalização dos mesmo conforme a necessidade.

## **2.4. Considerações sobre os trabalhos relacionados**

Os trabalhos analisados têm como objetivo comum auxiliar ou melhorar o gerenciamento e controle das fichas de treino dos clientes em academias, seja por meio de um sistema de código de barras ou de uma aplicação web controlada pelo pessoal responsável do estabelecimento. Além disso, foram desenvolvidas variações de aplicações móveis: uma que complementa a versão web, permitindo ao cliente acompanhar sua evolução e realizar agendamentos, e outra que propõe uma aplicação mobile gratuita e aberta, focada nos atletas para gravar e monitorar seus feitos.

Dentre os trabalhos analisados, o que mais se assemelha a proposta deste projeto é o de Oliveira [18]. Além de utilizar o mesmo framework para o *back-end*, seu conceito é parcialmente semelhante, com algumas diferenças na construção e concepção do sistema.

Ao analisar esses trabalhos, percebe-se que nenhum aborda especificamente o tempo que o instrutor leva para criar fichas de treino personalizadas para cada cliente. Este é um ponto crítico que este projeto pretende solucionar, buscando reduzir esse tempo através do uso de Inteligência Artificial, na qual optou-se pelo modelo do gpt-3.5-turbo da OpenAI. O uso da IA busca agilizar esse processo e melhorar a experiência tanto dos instrutores quanto dos clientes.

## **3. Desenvolvimento da aplicação**

Nesta seção, uma avaliação minuciosa do sistema é feita, englobando desde sua estrutura interna (API) até a interface visual destinada aos usuários. Serão discutidos aspectos como os requisitos, a estrutura geral do sistema e as tecnologias utilizadas. Adicionalmente, são mencionadas as funcionalidades práticas que o sistema oferecerá aos usuários.

É fundamental ressaltar que a criação deste projeto envolve duas áreas diferentes. Uma delas é o *back-end*, responsável pela lógica dos dados, conexão com o banco de dados e integração com a OpenAI. O outro ambiente refere-se ao *front-end*, que é responsável pela construção visual e focado na interação direta com os usuários finais.

### **3.1. Seleção de tecnologias e ferramentas para o desenvolvimento da aplicação**

Para a construção do *back-end* como uma Application Programming Interface (API) RESTful, foi escolhido a linguagem PHP[13], versão 8, pois é uma das linguagens mais utilizadas no ambiente web, juntamente com o *framework* Laravel[10], versão 11, que auxilia a ter um controle melhor do projeto e uma estrutura de código mais padronizada e escalável a longo prazo. Para o desenvolvimento do mesmo foi utilizada a plataforma da empresa JetBrains, PhpStorm, devido as funcionalidades oferecidas e integração com o framework Laravel[10]. Além de ter ferramentas para debugging, conexão com o banco de dados, testes, entre outras.

Para a concepção da API foram utilizados alguns padrões de design que seriam eles:

Repository Pattern: Esse padrão de projeto é utilizado para abstrair a lógica de acesso a dados, criando uma separação entre a aplicação e a camada de persistência.

Dessa forma, o código de negócios interage com uma interface de repositório em vez de fazer chamadas diretas ao banco de dados. Isso ajuda a manter o código mais modular, o que facilita a manutenção e a realização de testes.

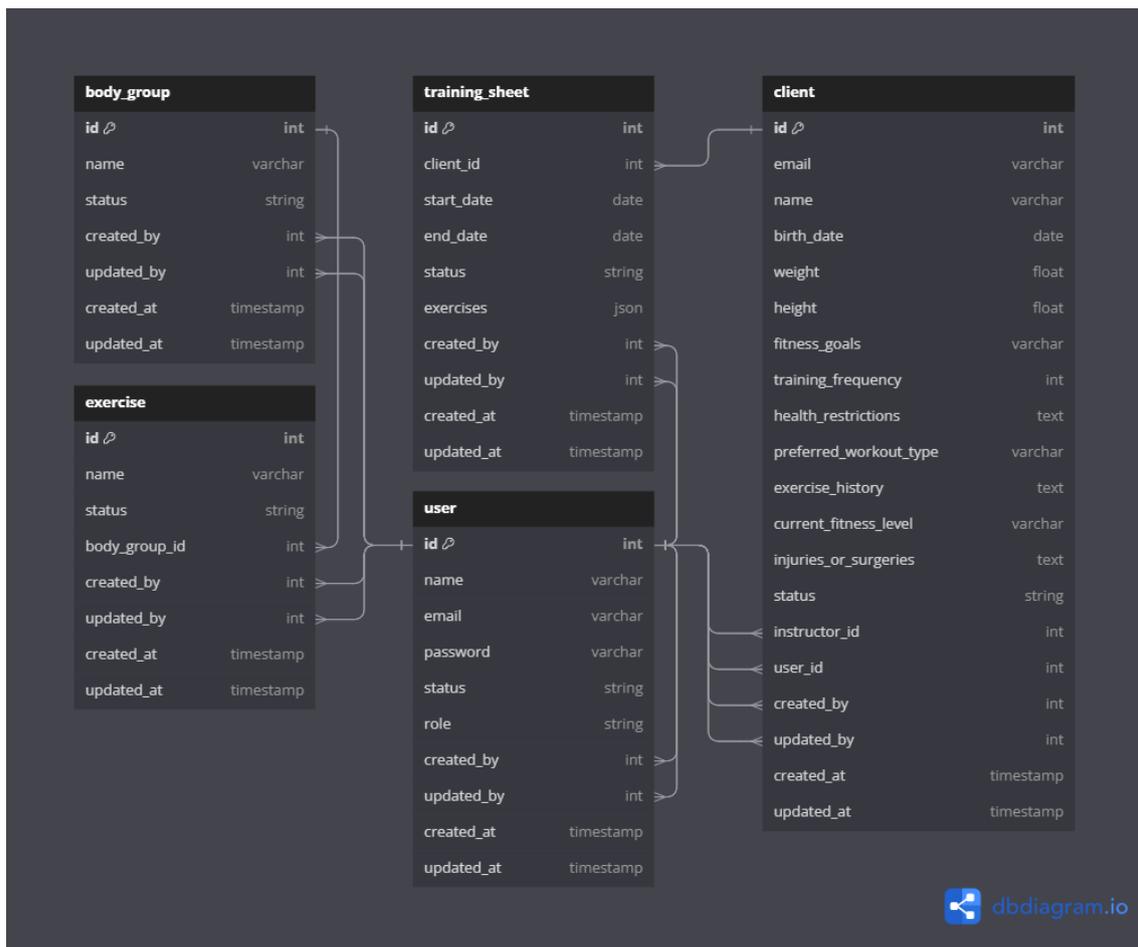
**Service Pattern:** O Service Pattern é usado para encapsular a lógica de negócios, isolando-a tanto da camada de controle quanto da camada de acesso a dados. Essa abordagem organiza a aplicação de forma que regras de negócios mais complexas sejam agrupadas em serviços específicos, promovendo um código mais coeso e fácil de entender.

**Dependency Injection:** A injeção de dependência é um princípio de design que visa inverter o controle na criação de objetos. Em vez de instanciar dependências diretamente dentro das classes, a injeção de dependência permite que essas dependências sejam passadas de fora, normalmente por meio de construtores ou métodos de configuração. Essa prática melhora a flexibilidade e a testabilidade do código.

**Active Record Pattern:** O padrão Active Record é uma abordagem em que uma classe representa uma linha de uma tabela do banco de dados e contém métodos para manipular esses dados diretamente. No projeto, foi utilizado o Eloquent ORM do Laravel[10], que segue esse padrão e permite uma interação simplificada e intuitiva com o banco de dados.

Para o banco de dados, optou-se por um modelo relacional devido à integridade dos dados e à disponibilidade. Embora todos os SGBDs relacionais implementem as propriedades ACID, a escolha do PostgreSQL[19] se deu por sua confiabilidade, suporte a transações complexas e ser um sistema de código aberto e gratuito. Essas características tornam o PostgreSQL uma opção robusta e escalável para o projeto.

A Figura 1 apresenta o modelo lógico de banco de dados utilizado pela aplicação, elaborado com a ferramenta online dbdiagram.io. Esta ferramenta foi crucial para a visualização e organização das relações entre as tabelas e campos do sistema, garantindo uma estrutura eficiente e coesa para o gerenciamento dos dados. A partir dessa análise, foi possível abstrair uma tabela intermediária para fazer a relação entre a tabela training sheet e os exercícios associados a ela. Essa estrutura ajudou a entender que o controle dos exercícios gerados pela IA, bem como a adição, exclusão e edição manual dos mesmos, apresentaria complexidade na implementação. Esse ponto foi levado em consideração para simplificar a gestão dos dados na aplicação.



**Figura 1. Modelo do banco de dados.**

Para garantir a segurança no sistema, foi utilizado o pacote integrado ao Laravel[10] chamado Sanctum, o qual fornece um sistema de autenticação leve para SPAs (single page applications), aplicativos móveis e APIs simples. É particularmente útil para gerenciar tokens e oferece várias funcionalidades, como emissão de tokens de API personalizados e a habilidade de revogá-los.

Foi empregado o Postman para testar o envio e recebimento de informações da API desenvolvida, facilitando o processo de envio de requisições HTTP e recebimento de respostas do servidor.

Para o desenvolvimento do *front-end*, foi utilizado o Angular[3], versão 18, um framework de JavaScript. A plataforma WebStorm, também da JetBrains, foi empregada para testar e depurar a aplicação. O Angular, conhecido por sua versatilidade e base em componentes, permite a modularização. A partir da versão 14, o Angular introduziu os componentes Standalone, que tornam cada componente independente, possibilitando o Lazy Loading. Isso significa que a aplicação carrega apenas componentes e módulos e códigos necessários, resultando em uma aplicação mais leve. Além dessas funcionalidades nativas, foi implementado o PrimeNG[14], uma biblioteca de componentes de interface de usuário (UI) amplamente utilizada para criar aplicações web modernas, responsivas e com design consistente.

### 3.2. Requisitos

Os requisitos do sistema são divididos em duas categorias principais: Funcionais e Não Funcionais.

Os Requisitos Funcionais (Tabela 1) descrevem as funcionalidades e operações específicas que o sistema deve realizar para atender as necessidades dos usuários e aos objetivos do projeto. Estes requisitos são diretamente relacionados as atividades e tarefas que o sistema deve executar.

Os Requisitos Não Funcionais (Tabela 2), por sua vez, definem as características de qualidade do sistema, como desempenho, segurança, usabilidade. Eles não estão diretamente relacionados as funcionalidades do sistema, mas garantem que ele opere de forma eficiente, segura e confiável.

A seguir, as Tabelas 1 e 2 apresentam os requisitos identificados para o sistema, organizados de forma clara e objetiva, com uma breve descrição para cada item.

**Tabela 1. Requisitos Funcionais**

<b>ID</b>	<b>Categoria</b>	<b>Descrição</b>
RF01	Gerenciamento de Usuários	Cadastro, edição, exclusão e visualização de detalhes de usuários.
RF02	Gerenciamento de Clientes	Cadastro, edição, exclusão, listagem e visualização de detalhes de clientes. Ao criar cliente, gerar um usuário para o mesmo.
RF03	Gerenciamento de Grupos Musculares	Cadastro, edição, exclusão e visualização de detalhes dos grupos musculares.
RF04	Gerenciamento de Exercícios	Cadastro, edição, exclusão, listagem e visualização de detalhes de exercícios. Associação de exercícios a grupos musculares.
RF05	Gerenciamento de Fichas de Treino	Criação, edição, exclusão, listagem e visualização de fichas de treino, com adição e remoção de exercícios. Geração de fichas de treino personalizadas utilizando IA.
RF06	Autenticação e Autorização	Login de usuários utilizando Laravel[10] Sanctum para autenticação baseada em tokens. O controle de acesso por perfil será implementado em trabalhos futuros.
RF07	Notificações e Alertas	Exibição de mensagens de sucesso, erro e alerta para ações realizadas.
RF08	Integração com OpenAI	Geração de fichas de treino personalizadas utilizando a API da OpenAI.

**Tabela 2. Requisitos Não Funcionais**

<b>ID</b>	<b>Categoria</b>	<b>Descrição</b>
RNF/INTERFACE-01	Usabilidade	Interface amigável, intuitiva e responsiva, com feedback visual para ações do usuário e elementos interativos que melhoram a experiência do usuário.
RNF/SEGURANCA-01	Segurança	Proteção contra acessos não autorizados, criptografia de dados sensíveis, validação de entrada no <i>front-end</i> e <i>back-end</i> .
RNF/MANUTENIBILIDADE-01	Manutenibilidade	Código modular e estruturado, com práticas de desenvolvimento e testes bem definidas para facilitar alterações e manutenção.
RNF/MANUTENIBILIDADE-02	Modularização	Arquitetura em camadas para garantir escalabilidade e a possibilidade de adicionar novas funcionalidades com facilidade.
RNF/COMPATIBILIDADE-01	Compatibilidade	Suporte aos principais navegadores web (Google Chrome, Firefox, Microsoft Edge, Safari) e responsividade para diferentes dispositivos, garantindo uma experiência consistente.
RNF/CONFIABILIDADE-01	Confiabilidade	Tratamento adequado de erros e exceções, com mensagens de erro claras para o usuário e mecanismos de recuperação.
RNF/CONEXÃO-01	Conectividade com a Internet	O aplicativo precisa de uma conexão estável com a Internet para funcionar corretamente e para geração de exercícios personalizados por IA.

### 3.3. Integração com Inteligência Artificial

Com este trabalho, busca-se diminuir o tempo necessário para um instrutor de academia criar a ficha de treino de um cliente. Para essa finalidade, optou-se pela utilização de uma inteligência artificial moderna e amplamente reconhecida, capaz de gerar resultados de alta qualidade de forma eficiente.

Atualmente, existem diversas ferramentas de IA disponíveis no mercado, cada uma com características específicas que as tornam adequadas para diferentes áreas, sejam elas mais especializadas, como análise médica e automação industrial, ou generalistas, como processamento de linguagem natural. No contexto deste trabalho, foi utilizada uma IA com foco em processamento de texto, escolhida por sua capacidade de compreender e interpretar informações detalhadas fornecidas pelo usuário, gerando respostas personalizadas com base em dados específicos.

Depois de pesquisar no mercado um modelo que atendesse as necessidades do projeto, levando em conta o custo de uso, optou-se pelo "gpt-3.5-turbo" da OpenAI, que possui um valor de 3 dólares por 1 milhão de tokens [12]. Um token é a menor unidade de texto que um modelo de linguagem consegue processar, podendo ser uma palavra completa, parte dela ou um caractere. Por exemplo, o termo "chatbot" é contado como um token, ao passo que "GPT-3.5" pode ser visto como dois tokens. A determinação do custo

de uso e da quantidade de texto processado é baseada na contagem de tokens, sendo cerca de 1.000 tokens equivalentes a 750 palavras em inglês.

A escolha se deu por sua ampla aplicação em diferentes campos, como a área da saúde, satisfazendo, dessa forma, a necessidade das academias ao abordar o corpo humano e seus limites. Em comparação com outros modelos disponíveis, este foi selecionado devido ao seu ótimo custo-benefício e capacidade de se integrar facilmente com várias linguagens. Na sequência, será descrito a integração do serviço à aplicação.

#### 4. Resultados Alcançados

Nesta seção, serão exibidas imagens das telas mais importantes do sistema, acompanhadas por uma descrição minuciosa de seus campos e funcionalidades. A intenção é mostrar como cada componente do sistema foi planejado para atender as demandas dos usuários e aprimorar a experiência como um todo. As telas consistem em telas de registro de clientes, registro de exercícios, registro da ficha de treino, listagem das fichas de treino.

Cada figura terá uma explicação dos elementos visuais e da lógica por trás de cada função, enfatizando a conexão com a API da IA e sua importância para a eficácia do sistema.

A Figura 2 mostra o formulário de cadastro de um cliente. Este formulário possui diversos campos importantes e necessários para garantir uma maior assertividade nos exercícios fornecidos. Esses dados serão utilizados pela IA para gerar os treinos personalizados, levando em conta as necessidades e restrições de cada cliente.

The image shows a web form titled "Editar cliente" (Edit client). The form is organized into several rows of input fields. The first row contains "Nome \*" (Name) with the value "Cliente Iniciante/Intermediário" and "Email \*" (Email) with the value "teste2@teste.com". The second row contains "Status \*" (Status) with a dropdown menu showing "Ativo", "Data de nascimento \*" (Date of birth) with the value "31/12/1990", "Peso \*" (Weight) with the value "85", and "Altura \*" (Height) with the value "1,78". The third row contains "Objetivos \*" (Goals) with the value "Perda de peso, hipertrofia" and "Tipo de Treino Preferido \*" (Preferred Training Type) with the value "Musculação". The fourth row contains "Histórico de Exercícios \*" (Exercise History) with the value "5 horas de atividades físicas frequentes durante a semana" and "Nível de Fitness Atual \*" (Current Fitness Level) with the value "Iniciante". The fifth row contains "Lesões ou Cirurgias" (Injuries or Surgeries) with the value "Cirurgia de LCA, Menisco" and "Restrições de Saúde" (Health Restrictions) with the value "Nenhum". The sixth row contains "Frequência de Treino \*" (Training Frequency) with a dropdown menu showing "2" and "Instrutor \*" (Instructor) with a dropdown menu showing "Instrutor".

**Figura 2. Formulário do cliente.**

Já a Figura 3 exibe o formulário de cadastro dos exercícios. Embora possua poucos campos, este formulário é de extrema importância, pois os exercícios gerados pela IA serão baseados nos dados cadastrados aqui.

Editar exercício

Nome \*

Status \*

Grupo Muscular \*

**Figura 3. Formulário do exercício.**

A Figura 4 mostra a tela mais importante do sistema, que após selecionar o cliente e clicar no botão azul com o ícone de robô, será feita uma chamada para a API do *back-end*, passando o ID do cliente selecionado. Esse ID é utilizado para obter as informações do cliente e também os exercícios existentes na academia, garantindo que não sejam sugeridos exercícios inexistentes na academia como mostra a Figura 6, que após todo esse processo de coletar as informações serão utilizados para construir o *prompt* explicando como o modelo deve atuar e retornar sua resposta.

Por exemplo, o *prompt* pode ser: "Você é um profissional altamente capacitado na área de educação física, saúde e *personal trainer*, com vasto conhecimento em anatomia, fisiologia e montagem de treinos personalizados...". Além disso, é adicionado ao *prompt* um comando explicando como o modelo deve retornar a sua resposta, seguindo uma estrutura JSON formada por um array de objetos com os seguintes campos: *exercise\_id*, *series*, *repetitions*, *weight*, *order*, que pode ser visualizado na Figura 7 todo esse processo da criação e execução do *prompt* a ser enviado para a API da OpenAI.

Após o processamento da requisição e retorno da API da IA, será utilizado o objeto json contido dentro do retorno que será enviado para o *front-end*, onde será tratado e disponibilizado visualmente em forma de tabela, permitindo que o instrutor avalie se o treino está adequado ao cliente. Caso não esteja, é possível solicitar alterações como a quantidade de séries a serem executadas, a quantidade de repetições, o peso a ser usado e a ordem de execução, tudo isso clicando no botão verde com o ícone de lápis. Para remover algum exercício indesejado basta clicar no botão vermelho com o ícone de lixeira correspondente a linha do mesmo. Também é possível realizar a remoção múltipla, marcar os checkboxes ao lado esquerdo de cada exercício, e então clicar no botão superior vermelho com ícone de lixeira que se encontra ao lado do botão de IA. Para adicionar um novo exercício que não se encontra na lista, basta clicar no botão superior verde com o ícone de sinal mais (+), que mostrará um pequeno formulário onde será preenchida as informações necessárias, como mostra a Figura 5, que logo após preencher, basta clicar em salvar que será adicionado a lista de exercícios juntamente com os de mais.

**Editar ficha de treino**

Cliente \* Cliente Iniciante/Intermediário × ▼
 Data de início \* 19/11/2024 ×
 Data de término \* 19/02/2025 ×
 Status \* Ativo × ▼

Gerar treino com IA

🔍 🗑️ +

<input type="checkbox"/>	Exercício ↑↓	Grupo Muscular ↑↓	Séries ↑↓	Repetições ↑↓	Peso ↑↓	Ordem ↑↓	
<input type="checkbox"/>	Voador Frontal	Peito	4	12		1	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Supino inclinado com barra	Peito	4	12		1	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Barra fixa pegada pronada	Costas	4	12		2	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Remada TRX	Costas	4	12		2	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Desenvolvimento com barra	Ombros	4	12		3	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Rosca direta	Biceps	4	12		4	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Tríceps testa com barra	Tríceps	4	12		5	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Agachamento livre	Pernas	4	12		6	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Barra fixa neutra	Costas	4	12		7	<span>✎</span> <span>🗑️</span>
<input type="checkbox"/>	Levantamento terra com barra hexagonal	Pernas	4	12		8	<span>✎</span> <span>🗑️</span>

Mostrando 1 a 10 de 12 registros « < 1 2 > » 10 ▼

**Figura 4. Formulário da ficha de treino.**

### Detalhes do exercício ✕

Exercício \*

Selecione um exercício ∨

Séries \*

Repetições \*

Peso

Ordem

✕ Cancelar ✓ Salvar

Figura 5. Detalhes do exercício.

```

public function generateTrainingSheet($request)
{
    $clientData = $this->clientRepository->getById($request->client_id);

    if (!$clientData) {
        return ApiResponseService::error( message: 'Client id not founded or provided', status: 404);
    }

    $clientData = $clientData->only([
        'name',
        'birth_date',
        'weight',
        'height',
        'fitness_goals',
        'training_frequency',
        'health_restrictions',
        'preferred_workout_type',
        'exercise_history',
        'current_fitness_level',
        'injuries_or_surgeries',
    ]);

    $bodyGroupsData = $this->bodyGroupService->getAll()->pluck('name', 'id')->toArray();
    $exercises = $this->exerciseRepository->getAll();

    $exerciseData = [];
    foreach ($exercises as $exercise) {
        $exerciseData[] = [
            'id' => $exercise->id,
            'name' => $exercise->name,
            'body_group_name' => $bodyGroupsData[$exercise->body_group_id],
        ];
    }

    $data = [
        'client' => $clientData,
        'exercises' => $exerciseData,
    ];

    $response = $this->openAIService->generateTrainingSheet($data);

    if ($response) {
        return ApiResponseService::success($response, message: 'Exercises generated', status: 201);
    } else {
        return ApiResponseService::error( message: null, status: 'Error on generate exercises', 500);
    }
}

```

Figura 6. Código onde é montado o *array* de dados para acoplar ao *prompt*.

```

public function generateTrainingSheet(array $data)
{
    $response = $this->openAiClient->chat()->create([
        'model' => 'gpt-3.5-turbo',
        'messages' => [
            [
                'role' => 'system',
                'content' => 'Você é um profissional altamente capacitado na área de educação física, saúde e personal trainer, com vasto conhecimento em anatomia, fisiologia e montagem de treinos personalizados. Que fornece respostas obrigatoriamente no formato JSON sem textos além da lista de exercícios e somente retorne a lista e nada mais, não precisa indentar a lista! Seu objetivo é criar a melhor ficha de treino possível, coerente com os objetivos do cliente pensando na quantidade de dias da semana(pensando que cada dia de treino deve ter pelo menos 1 hora, então monte o treino com exercícios suficientes para atender esse requisito) que o cliente vai treinar. E baseado nos exercícios disponíveis na academia, no qual vai montar as fichas somente com esses exercícios, adicionar nenhum exercício fora da lista fornecida, sempre retornando os ids(exercise_id) de cada exercício conforme o modelo de retorno fornecido a seguir. O formato de retorno deve obrigatoriamente ser uma lista de objetos(JSON) com os campos: exercise_id, series, repetitions, weight, order. O campo "order" deve ser um sequencial para cada grupo muscular, exemplo o grupo "peito" vai ter o "order" do 1 ao 5, o grupo "biceps" vai ter o order do 1 ao 4, e assim por diante. O campo "weight" é opcional retornar preenchido. Aqui está a estrutura que você deve retornar a resposta: [{"exercices": [{"exercise_id": example_exercise_id,"series": example_series,"repetitions": example_repetitions,"weight": example_weight,"order": example_order},{ "exercise_id"(type number): example_exercise_id,"series"(type number): example_series,"repetitions"(type number): example_repetitions,"weight"(type number): example_weight,"order"(type number): example_order}]}]
            ],
            [
                'role' => 'user',
                'content' => 'Por favor, crie uma ficha de exercícios completa com base nas informações a seguir. O objeto de dados "client" contém as informações do aluno, incluindo objetivos, saúde, restrições, etc, nos quais devem ser levados em consideração na montagem de treino, pensando na melhor performance possível dentro dos limites(físicos) que o cliente possui, e também levando em conta a experiência que ele tem. O objeto de dados "exercices" contém os exercícios disponíveis na academia(não sugerir exercícios que não estão na lista fornecida). A ficha de treino deve cobrir todos os grupos musculares, levando em consideração os problemas de saúde e lesões do cliente, e atender aos seus objetivos de fitness. Aqui estão os dados: '. json_encode($data)
            ]
        ],
        'max_tokens' => 2000,
    ]);

    return json_decode($response['choices'][0]['message']['content'], associative: true);
}

```

Figura 7. Código onde é montado o *prompt* e realizado a requisição para a API da OpenAI.

Por fim, mas não menos importante, a Figura 8 ilustra a visualização das informações do cadastro da ficha de treino. Os demais cadastros possuem uma visualização semelhante, oferecendo basicamente as mesmas funcionalidades: criar, editar, visualizar e excluir um registro.

Além disso, há a opção de exclusão em massa de registros, que pode ser feita marcando os checkboxes ao lado esquerdo de cada linha e, em seguida, clicando no botão vermelho com o texto "Deletar selecionados".

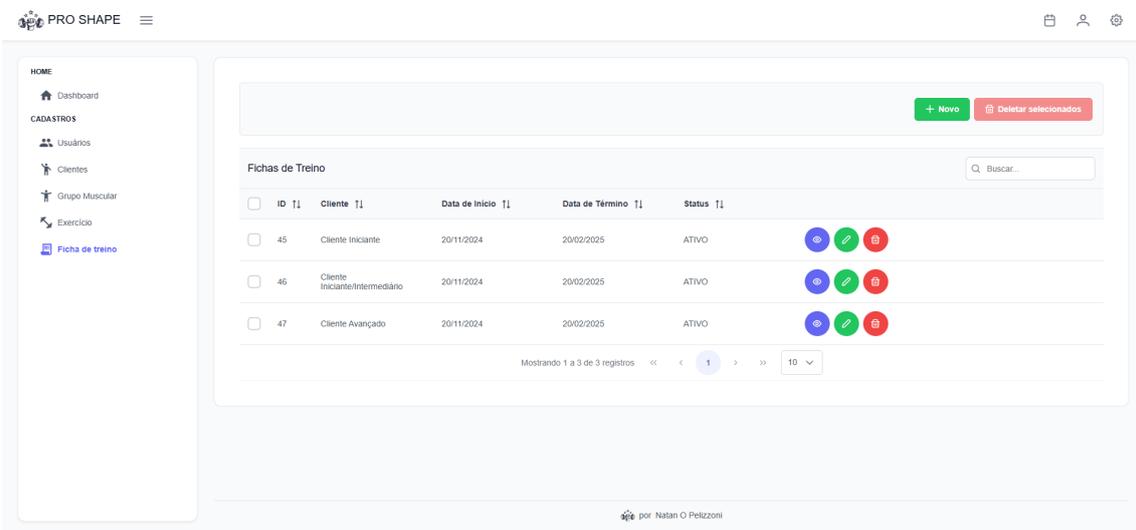


Figura 8. Listagem das fichas de treino.

## 5. Validação

Nesta seção, será discutido o processo de validação dos resultados fornecidos pela IA. Para isso, foi contatado um único profissional da área, proprietário de uma academia focada exclusivamente em atendimento como *personal trainer*. Esse profissional também possui experiência prévia como instrutor em academias abertas, pois trabalhou nesse tipo de ambiente antes de abrir seu próprio espaço. Essa vivência permitiu obter feedbacks tanto da perspectiva de um *personal trainer* quanto de instrutores em academias abertas.

A validação foi realizada *in loco* na academia, onde foram cadastrados três clientes com perfis, objetivos, idades e níveis de experiência diferentes, além de considerações sobre possíveis problemas físicos ou de saúde. Os perfis escolhidos foram: iniciante, intermediário e avançado. Após o cadastro dos grupos musculares, foram inseridos os exercícios disponíveis na academia.

Com todos os preparativos concluídos, passou-se ao objetivo principal: gerar um treino para cada cliente cadastrado. No formulário de ficha de treino, foram selecionados cada cliente, definidas informações como a data de início e término do treino e enviados para a IA para gerar os exercícios correspondentes.

Após a IA gerar os exercícios para cada ficha de treino, o proprietário avaliou a adequação dos treinos. Caso algum treino não estivesse adequado, ele explicou o que poderia ser ajustado para torná-lo mais agradável e eficiente para o cliente.

Os feedbacks apontaram que, para o cliente iniciante, o treino estava muito extenso (16 exercícios) no qual ele sugeriu de oito a dez exercícios e poderia haver uma melhor alternância entre os grupos musculares, priorizando os músculos maiores (peito, costas e pernas) antes dos menores (bíceps, tríceps, ombros, etc.).

Para os clientes intermediário e avançado, houve uma ênfase desequilibrada entre alguns grupos musculares. Por exemplo, no treino do cliente avançado, apenas dois exercícios foram destinados ao grupo peito e quatro ao grupo costas. Se fossem realizados todos os exercícios gerados pela IA em um único dia, o treino ficaria exaustivo, comprometendo a performance e os resultados desejados, além de aumentar o risco de lesão.

Uma sugestão foi dividir o treino em partes, para serem executadas em dias diferentes. Por exemplo, no dia A, executar exercícios para peito e bíceps; no dia B, para costas e tríceps; no dia C, para pernas e ombros, e assim por diante, facilitando a execução de todos os exercícios da ficha de treino.

Por fim, ele avaliou a eficácia do sistema na montagem de treinos para economizar tempo. Ele apresentou dois cenários: no primeiro, como *personal trainer*, o uso da IA não seria tão viável, já que o treino varia conforme os dias e o desempenho do cliente, com ajustes sendo feitos durante a execução dos exercícios.

No segundo cenário, em uma academia aberta com mais clientes, o uso da IA seria bastante útil. Nesse contexto, os treinos são fixos por um período de tempo, e em alguns dias é necessário planejar treinos para dois clientes, enquanto em outros esse número pode chegar a até dez. Isso aumenta significativamente o tempo demandado pelo instrutor, já que, a cada novo aluno que ingressa e precisa de uma ficha de treino, o processo se torna mais trabalhoso e consome ainda mais tempo. Com o auxílio da IA, a montagem dos

treinos seria facilitada, entregando exercícios prontos ou com pequenos ajustes para os clientes.

## 6. Considerações finais

Neste trabalho, foi desenvolvida e validada uma aplicação web para criação de treinos personalizados, utilizando IA. A IA provou ser uma ferramenta eficiente para a criação de treinos, embora tenha apresentado algumas sugestões que precisaram de ajustes para um dos cenários de uso. Mesmo assim, oferece uma solução prática para instrutores de academias abertas, pelo fato de otimizar o tempo de planejamento. No entanto, a utilização pode ser mais limitada para *personal trainers* que ajustam os treinos conforme o desempenho do cliente.

As tecnologias selecionadas atenderam as necessidades do projeto, por serem atuais e possuírem bom suporte e manutenibilidade, especialmente a integração com o modelo "gpt-3.5-turbo" da OpenAI, que se mostrou adequado tanto em termos de custo quanto de desempenho.

Durante o desenvolvimento, foram encontradas algumas dificuldades e limitações, como a necessidade de ajustes para encontrar o melhor *prompt* a ser enviado para a API do modelo, de modo que a IA pudesse gerar treinos garantindo maior adequação as necessidades específicas de cada cliente.

No quesito do projeto, houveram alguns desafios para encontrar encontrar uma boa estrutura de código tanto para o *back-end* quanto para o *front-end*, que resultasse em maior facilidade para implementar novos módulos no futuro. Além disso, também houve dificuldades na escolha dos melhores padrões de código utilizados no mercado que pudessem ser aplicados no projeto, garantindo melhor manutenção e qualidade.

Para a escolha do modelo a ser utilizado na geração dos exercícios, foi necessária uma pesquisa de mercado para encontrar aquele que melhor se adequasse e que entregasse resultados satisfatórios a um custo baixo ou razoável. Optou-se pelo modelo "gpt-3.5-turbo" da OpenAI, que entre diversas opções, se mostrou efetivo para o objetivo proposto, por possuir conhecimento em múltiplas áreas, incluindo saúde e exercícios.

Para trabalhos futuros, são sugeridos os seguintes pontos:

- Realizar uma validação com uma amostra maior de clientes, visando obter um feedback mais abrangente.
- Treinar um modelo de IA específico para o contexto de academias, resultando em maior assertividade nos treinos gerados.
- Implementar o controle por perfil de usuário (Admin, Instrutor e Cliente), definindo o que cada um poderá realizar e ver dentro da aplicação.
- Criar novos módulos dentro da ferramenta, como por exemplo o controle de pagamentos e a gestão de planos de nutrição, incluindo a integração com profissionais de nutrição.
- Hospedar a aplicação completa em uma plataforma de cloud como a Amazon Web Services (AWS), garantindo escalabilidade e disponibilidade da ferramenta.
- Desenvolver um aplicativo móvel e disponibiliza-lo nas principais lojas (App Store e Play Store), permitindo que os clientes acompanhem seus treinos e que os instrutores também possam utilizar a ferramenta.

- Integração com dispositivos wearables na aplicação móvel, possibilitando que o instrutor acompanhe o rendimento dos treinos executados pelo cliente.

Além disso, é importante coletar feedbacks dos clientes da ferramenta, para que possam ser realizadas correções e melhorias contínuas, aprimorando cada vez mais a aplicação.

## Referências

- [1] AdminLTE Bootstrap Admin Dashboard Template. <https://adminlte.io/>. [Acesso: 25-11-2024].
- [2] Android — Faça mais com o Google em telefones e dispositivos Android — android.com. [https://www.android.com/intl/pt\\_br/](https://www.android.com/intl/pt_br/). [Acesso: 16-04-2024].
- [3] Angular — angular.dev. <https://angular.dev/>. [Acesso: 25-11-2024].
- [4] CSS — MDN — developer.mozilla.org. <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. [Acesso: 16-04-2024].
- [5] Delphi - Embarcadero — embarcadero.com. <https://www.embarcadero.com/br/products/delphi>. [Acesso: 16-04-2024].
- [6] Google Scholar — scholar.google.com.br. <https://scholar.google.com.br/>. [Acesso: 16-04-2024].
- [7] HTML: Linguagem de Marcação de Hipertexto — MDN — developer.mozilla.org. <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. [Acesso: 16-04-2024].
- [8] iOS 17 — apple.com. <https://www.apple.com/br/ios/ios-18/>. [Acesso: 16-04-2024].
- [9] JavaScript — MDN — developer.mozilla.org. <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. [Acesso: 16-04-2024].
- [10] Laravel - The PHP Framework For Web Artisans — laravel.com. <https://laravel.com/>. [Acesso: 25-11-2024].
- [11] macOS Sequoia — apple.com. <https://www.apple.com/br/macOS/sonoma/>. [Acesso: 16-04-2024].
- [12] OpenAI. <https://openai.com/api/pricing/>. [Acesso: 24-12-2024].
- [13] PHP: Hypertext Preprocessor — php.net. <https://www.php.net/>. [Acesso: 25-11-2024].
- [14] PrimeNG - Angular UI Component Library — primeng.org. <https://primeng.org/>. [Acesso: 25-11-2024].
- [15] Supabase — The Open Source Firebase Alternative — supabase.com. <https://supabase.com/>. [Acesso: 16-04-2024].
- [16] Vue.js — vuejs.org. <https://vuejs.org/>. [Acesso: 16-04-2024].
- [17] DE ALMEIDA, A. L., AND DA SILVA, R. F. Desenvolvimento de um sistema para academias com acompanhamento da rotina de treino.

- [18] DE OLIVEIRA, S. J. Proposta de um software para gerenciamento de ficha de treinamento para pequenas academias.
- [19] GROUP, P. G. D. PostgreSQL — postgresql.org. <https://www.postgresql.org/>. [Acesso: 16-04-2024].
- [20] MICROSOFT. Windows — Microsoft. <https://www.microsoft.com/pt-br/windows>. [Acesso: 16-04-2024].
- [21] SOUZA, E. V. D., SOUZA, F. S. D., OLIVEIRA, G. T. D., SILVA, L. F. P. D., AND PAIVA, V. J. R. Sga: sistema gestão de academia.