

Desenvolvimento e avaliação de um sistema web para geração de planos de treino com a API do ChatGPT

Rafael Borella Strzelecki¹, Ártton Pereira Dorneles²

¹Curso de Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFar) – Frederico Westphalen – RS – Brasil

²Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFar) – Frederico Westphalen – RS – Brasil

bs.rafa@outlook.com, artton.dorneles@iffarroupilha.edu.br

Abstract. *With the increasing number of students in gyms, the demand for Personal Trainers to develop personalized workout plans has grown significantly. However, manually creating these plans requires considerable time, making it difficult to personalize and continuously monitor each student. This study proposes the development of a system based on large-scale language models for the automated generation of workout plans using the ChatGPT API. Different prompt engineering approaches are analyzed to evaluate combinations of techniques that generate more precise and tailored recommendations according to the proposed evaluation criteria. Experiments conducted on a diverse dataset of students demonstrate that combining the Persona and Few-Shot techniques results in structured workout plans that are better aligned with training objectives.*

Resumo. *Com o crescimento do número de alunos em academias, a demanda por Personal Trainers para a elaboração de planos de treino personalizados aumentou significativamente. No entanto, a criação manual desses planos demanda um tempo significativo, dificultando a personalização e o acompanhamento contínuo de cada aluno. Este trabalho propõe o desenvolvimento de um sistema baseado em modelos de linguagem de grande escala para a geração automatizada de planos de treino, utilizando a API do ChatGPT. Diferentes abordagens de engenharia de prompt são analisadas para avaliar combinações de técnicas que geram recomendações mais precisas e adaptadas conforme os critérios de avaliação propostos. Os experimentos realizados em cima de um conjunto de dados de alunos diversos, demonstram que a combinação das técnicas Persona e Few-Shot resultam na geração de treinos organizados e mais alinhados aos objetivos de treinamento.*

1. Introdução

Nos últimos anos, o número de pessoas que buscam academias para melhorar sua saúde e bem-estar tem crescido significativamente, o que aumenta também a demanda por *personal trainers*. Esses profissionais, além de acompanharem os alunos durante os treinos, precisam desenvolver planos de treino personalizados, levando em consideração variáveis como o percentual de gordura, peso, altura, nível de experiência de treino e objetivos. No entanto, a criação de treinos específicos pode ser um desafio devido ao tempo limitado dos personal trainers e à necessidade de atender a múltiplos alunos com diferentes objetivos e restrições físicas (NEGÓCIOS SC, 2024).

Apesar de já existirem plataformas de gestão de treinos, a grande maioria exige que o profissional elabore os planos de forma manual. Embora o uso de modelos de linguagem de larga escala nesse contexto ainda seja pouco explorado, estudos recentes indicam que ferramentas como o ChatGPT têm potencial para produzir treinos realistas e que podem ser aplicados na prática, desde que revisados e adaptados por um especialista para garantir a segurança dos alunos (WASHIF et al., 2024).

Nesse contexto, este trabalho propõe a implementação de um sistema baseado em modelos de linguagem de larga escala para a geração automatizada de planos de treino, utilizando a API do ChatGPT. O sistema é projetado para permitir que *Personal Trainers* insiram informações sobre os alunos e recebam recomendações de treinos adaptadas às suas necessidades específicas. Além disso, diferentes abordagens de engenharia de *prompt* são exploradas para avaliar qual estratégia proporciona os melhores resultados na geração dos planos de treino.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 apresenta o referencial teórico, abordando os fundamentos da elaboração de planos de treino, o funcionamento dos modelos de linguagem de grande escala e tecnologias utilizadas. A Seção 3 descreve a metodologia utilizada para o desenvolvimento do sistema. A Seção 4 apresenta os detalhes de implementação do sistema gerador de treino. A Seção 5, apresenta as estratégias de engenharia de *prompt* aplicadas, critérios de avaliação e experimentos para avaliação do sistema proposto. Por fim, a Seção 6 discute os resultados obtidos e propõe possíveis direções para trabalhos futuros.

2. Referencial Teórico

2.1 Elaboração de Planos de Treino

Nesta seção são apresentados os conceitos fundamentais que orientam a elaboração de um plano de treino de exercícios, assim como as principais variáveis que devem ser consideradas pelo profissional de Educação Física. Ao final, discutem-se diferentes estruturas de treino, exemplificando formas de organização de acordo com o objetivo do aluno.

2.1.1 Definição e fundamentos básicos

Conforme CHARRO et. al (2018), um plano de treino é um conjunto de exercícios prescritos por um profissional de Educação Física, normalmente conhecido como *Personal Trainer*. Para orientar a montagem desse plano, o profissional realiza uma avaliação física inicial do aluno, coletando dados como peso, altura, idade e outras medidas relevantes para o processo. Todos esses dados são encarados como variáveis importantes na elaboração do treino e sua influência a influência de cada uma é tratada em mais detalhes na Seção 2.1.2.

A partir dos dados obtidos, o *Personal Trainer* define, em conjunto com o aluno, o objetivo do treino, que pode englobar ganho de massa muscular, perda de peso, melhoria da resistência ou outras finalidades específicas. Com base no objetivo, se estabelece um tipo de treino que normalmente se encaixa em três tipos apresentados em detalhes na Seção 2.1.3: Hipertrofia, Resistência ou Força. Em seguida, ocorre a avaliação das valências físicas, como força, resistência e velocidade, definindo-se o volume de exercícios, o número de séries, o número de repetições, as cargas iniciais e a organização dos exercícios ao longo da semana. Esse conjunto de informações constitui o plano de treino inicial que um aluno deve seguir com o apoio do *Personal Trainer*.

Embora o plano de treino inicial seja importante para que o aluno possa começar a se exercitar, torna-se necessário alterá-lo com o tempo para que o aluno evolua. Entre as principais estratégias para promover essa progressão, destaca-se o princípio da sobrecarga progressiva, que consiste em ajustar gradualmente variáveis como carga, número de repetições e volume de treino de modo a estimular novas adaptações fisiológicas. Outra forma de progredir é por meio da periodização do treino, que consiste em organizá-lo em ciclos. Cada ciclo é composto de exercícios com parâmetros diferentes e que devem ser executados ao longo de várias semanas. A alternância de ciclos de treinamento, permite que o corpo se adapte gradualmente a diferentes estímulos, prevenindo estagnação, reduzindo o risco de lesões e promovendo melhorias contínuas no desempenho do aluno.

2.1.2 Principais variáveis usadas na elaboração do plano de treino

A elaboração de um plano de treino adequado envolve a análise de diversas variáveis individuais do aluno. Dentre as principais, destacam-se idade, gênero, peso, percentual de gordura, altura, histórico de lesões, experiência de treino e o objetivo do treino. Cada variável e sua influência será apresentada a seguir.

Idade. À medida que o tempo passa, o corpo humano sofre alterações neuromusculares, no sistema cardiorrespiratório e na sua composição corporal. Esta variável é importante, pois, por exemplo, pessoas com mais idade, que estão entrando na terceira idade, tendem a ter uma maior perda de massa muscular e densidade óssea, aumentando o risco de terem uma lesão ao realizarem certos exercícios (MATSUDO et. al., 2000).

Gênero. O diâmetro das fibras musculares, em indivíduos naturais, é significativamente maior no gênero masculino, o que pode permitir a prescrição de exercícios com uma carga inicial mais elevada em relação ao gênero feminino (TEIXEIRA, 2009).

Peso. Em alunos iniciantes que apresentam excesso de peso, pode ocorrer uma sobrecarga significativa nas articulações, o que requer uma seleção de exercícios de menor impacto. Além disso, se torna importante priorizar conjuntos de exercícios que também contribuam com a redução do peso, visando diminuir o risco de estresse articular (ACSM's, 2017).

Percentual de gordura. Quando o aluno possui um percentual de gordura corporal baixo, mas ainda apresenta um peso total elevado em razão de maior massa muscular, é possível introduzir cargas mais altas nos exercícios. Contudo, a biomecânica ainda deve ser monitorada de perto, pois as articulações continuam sob risco de estresse articular que requer monitoramento constante (HEYWARD, 2018).

Altura. A altura do aluno pode afetar significativamente na biomecânica dos movimentos, demandando adaptações para evitar sobrecargas articulares. Pessoas mais altas, por exemplo, podem ter maior dificuldade em exercícios como o agachamento, devido às alavancas corporais (MCGILL, 2007).

Experiência de Treino. A experiência do aluno com o treino em academias é um fator essencial pois define a complexidade dos exercícios que serão prescritos no plano. Alunos iniciantes devem priorizar exercícios básicos, com progressão gradual de carga, de modo a aperfeiçoar a postura e permitir um ganho de massa muscular inicial. Alunos avançados podem adotar métodos de treino com exercícios mais sofisticados e de alta intensidade, já que dispõem de maior controle postural e força adquiridos com a experiência (BAECHLE, 2008).

Histórico de Lesões. Lesões podem limitar a mobilidade e estabilidade articular, bem como a capacidade de suportar pesos. O *Personal Trainer* deve estar ciente de qualquer lesão existente ou anterior para adaptar exercícios que não sobrecarreguem a região afetada, priorizando, quando necessário, o fortalecimento e reabilitação dessa área (KIBLER, 2013).

Objetivo de Treino. A definição do objetivo – hipertrofia, emagrecimento, ganho de força ou condicionamento – direciona a escolha do tipo de exercício, o volume, a intensidade e o tempo de descanso. Por exemplo, cargas elevadas e poucas repetições são recomendadas para o ganho de força, enquanto maiores repetições e intervalos reduzidos entre as séries favorecem o estímulo metabólico (SCHOENFELD, 2010).

2.1.3 Organização e Tipos de Treino

A organização de um programa de treinamento varia conforme o objetivo definido pelo aluno em conjunto com o *Personal Trainer*. Em geral, existem três tipos de treino principais: treino de resistência, treino de hipertrofia e treino de força. Cada um apresenta parâmetros específicos de volume (número de séries e repetições), intensidade (carga) e intervalos de descanso.

O treino de resistência é voltado ao desenvolvimento da resistência cardiovascular e muscular. Geralmente, aplica-se uma intensidade moderada, com 3 a 5 séries de 12 a 20 repetições e pausas de 30 a 90 segundos (JONES, 1985). Essa abordagem pode incluir circuitos de exercícios como agachamentos, flexões e abdominais, mantendo o ritmo contínuo para aprimorar tanto a resistência muscular local quanto a capacidade aeróbia.

Já o treino de hipertrofia visa o aumento de massa muscular e, indiretamente, pode auxiliar na redução de gordura corporal por elevar o metabolismo basal. O protocolo usual varia de 3 a 4 séries, cada qual com 6 a 12 repetições, utilizando cargas maiores. Os intervalos de descanso vão de 30 a 90 segundos, permitindo recuperação parcial do músculo e promovendo alta tensão mecânica e estresse metabólico, fundamentais para o crescimento muscular (SCHOENFELD, 2010).

Por fim, o treino de força foca em cargas mais pesadas e baixas repetições, normalmente de 1 a 5, em 3 a 4 séries. Seu principal objetivo é aprimorar a capacidade de levantar cargas elevadas, priorizando exercícios compostos como agachamento, levantamento terra e supino reto. Nesse caso, o intervalo de descanso recomendado é mais longo, variando entre 2 e 5 minutos, devido à alta exigência neuromuscular e à necessidade de recuperação adequada (RHEA, 2003).

Após definido o tipo de treino, independente do foco em hipertrofia, resistência ou força, o personal trainer normalmente divide o treino em conjuntos de exercícios distintos, cada um, rotulado com uma letra. Um treino ABC, por exemplo, nos indica que existem 3 conjuntos de exercícios. Essa divisão permite que o aluno organize a execução de cada conjunto conforme a sua disponibilidade ou conforme o objetivo definido pelo personal trainer. Por exemplo, em semanas com três dias de treino, o conjunto de exercícios associado a cada letra é executado uma vez, enquanto em semanas com seis dias, cada conjunto pode ser repetido duas vezes, respeitando a recuperação apropriada. Dessa forma, o uso de letras permite criar uma rotina personalizada que favorece o controle de carga, intensidade e recuperação (KRIEGER, 2010).

A Figura 1 ilustra um exemplo de um plano de treino real elaborado por um profissional, usando a divisão ABCDEF. Podemos observar que o plano é composto dos dados do aluno, um plano de treino definindo os conjuntos de exercícios e uma organização semanal do treino.

Treino de Rafael			
Objetivo: Hipertrofia	Idade: 28	Peso: 105 Kg	Gênero: Masculino
Experiência: Avançado	Lesão: Não	Percentual de Gordura: 18%	Altura: 180 cm
Plano de treino			
Treino A – Bíceps, tríceps			
Exercícios: Rosca Martelo (Halter) 4x, Rosca Direta (Cross) 4x, Rosca Scott (Máquina ou Barra) 4x, Rosca Testa Supinada (Barra) 4x, Tríceps Pulley (Pronado) 4x, Rosca Testa Neutra (Halteres) 4x.			
<i>Descanso de 1 a 2 minutos entre as séries. Repetições de 8 a 12 em todos os exercícios.</i>			
Treino B – Ombro, Abdômen			
Exercícios: Desenvolvimento (Máquina) 4x, Elevação Lateral (Halter) 4x, Crucifixo Inverso (Cross) 4x, Elevação Frontal (Halter – Pegada Neutra) 4x, Infra 4x, Oblíquos 1x, Prancha 2x.			
<i>Descanso de 1 a 2 minutos entre as séries. Repetições de 8 a 12 em todos os exercícios.</i>			
Treino C – Coxa			
Exercícios: Mesa Flexora 4x, Agachamento (Hack ou Máquina) 4x, Cadeira Adutora 4x, Cadeira Extensora 4x, Elevação Pélvica (Barra ou Máquina) 3x.			
<i>Descanso de 2 a 3 minutos entre as séries. Repetições de 12 a 16 em todos os exercícios.</i>			
Treino D – Panturrilha, Trapézio, Abdômen			
Exercícios: Gêmeos Sentado (Máquina, Smith ou Livre) 5x, Gêmeos Leg Press 5x, Encolhimento Unilateral (Cross) 4x, Remada Alta (Barra ou Smith) 4x, Remada Alta (Cross) 4x, Supra 4x, Oblíquos 1x, Prancha 2x.			
<i>Descanso de 1 a 2 minutos entre as séries. Repetições de 8 a 12 em todos os exercícios.</i>			
Treino E – Costas, Antebraço			
Exercícios: Puxada Vertical (Pronada) 4x, Remada Máquina (Supinada) 4x, Puxada Vertical (Supinada) 4x, Remada Halter (Neutra) 4x, Extensão Lombar 2x, Rosca Inversa (Barra ou Halter) 4x, Extensão de Punho (Barra ou Halter) 4x.			
<i>Descanso de 1 a 2 minutos entre as séries. Repetições de 8 a 12 em todos os exercícios.</i>			
Treino F – Peito, Abdômen			
Exercícios: Cross Over (Polia Alta) 4x, Supino Reto (Barra ou Smith) 4x, Crucifixo Inclinado 4x, Supino Reto (Halteres) 4x, Oblíquos 1x, Supra 4x, Prancha 2x.			
<i>Descanso de 1 a 2 minutos entre as séries. Repetições de 8 a 12 em todos os exercícios.</i>			
Organização semanal			
Dia 1: Treino A, Dia 2: Treino B, Dia 3: Treino C, Dia 4: Treino D, Dia 5: treino E, Dia 6: Treino F, Dia 7: Repouso.			

Figura 1. Exemplo de plano de treino real elaborado por um profissional.

2.2 Modelos de Linguagem de Grande Escala (LLMs)

Os Modelos de Linguagem de Grande Escala (*Large Language Models* – LLMs) são baseados em redes neurais treinadas sobre grandes volumes de dados e utilizados amplamente para processar e gerar texto em diversas aplicações, como tradução automática, criação de resumos, geração de código-fonte e construção de assistentes virtuais. De acordo com Brown et. al. (2020), esses modelos recebem como entrada um texto e são otimizados para prever a sequência mais provável de palavras que atende ao contexto. Por exemplo, ao receber uma entrada como "O sol está brilhando no", o modelo analisará as palavras fornecidas e preverá que o próximo termo mais provável é "céu", formando a frase "O sol está brilhando no céu". Esse processo continua iterativamente, usando a última previsão como uma nova entrada, podendo acrescentar "azul", resultando na sequência "O sol está brilhando no céu azul". Cada nova palavra gerada pelo modelo é escolhida com base na probabilidade de ocorrência dentro do contexto das palavras anteriores, garantindo que o texto gerado seja coeso e natural.

A grande maioria dos LLMs permitem que o processo de predição seja personalizado por meio de um conjunto de parâmetros, sendo a temperatura um dos parâmetros mais importantes. De acordo com Renze (2024), o parâmetro de temperatura controla a aleatoriedade das previsões de texto realizadas pelo modelo, influenciando diretamente no equilíbrio entre coerência e criatividade. Quando configurado com um valor baixo, próximo de zero, o modelo gera respostas mais previsíveis e determinísticas, com pouca variação entre diferentes execuções da mesma consulta. Por outro lado, temperaturas mais altas, próximas de um, ampliam a diversidade do texto gerado, tornando as respostas mais criativas e variadas, embora potencialmente menos precisas. Por exemplo, com uma temperatura baixa, a entrada “O céu está muito” seria completada por “azul hoje”, enquanto com uma temperatura alta, seria completada por “brilhante, refletindo o tom dourado do entardecer”.

O ajuste dos parâmetros é crucial para adaptar o comportamento do modelo conforme as exigências da aplicação, seja priorizando respostas mais seguras e consistentes ou incentivando maior espontaneidade na geração de texto. Desta forma, é importante observar os parâmetros existentes em cada arquitetura, que costumam ser semelhantes, e então realizar ajustes experimentais naqueles mais significativos.

A base dos LLMs modernos é a arquitetura *Transformer*, proposta por Vaswani et al. (2017), que introduziu o mecanismo de *self-attention*, permitindo que o modelo processe palavras levando em consideração seu relacionamento global dentro de uma sequência de texto. Essa abordagem tornou os modelos baseados em *transformers* significativamente mais eficazes do que abordagens anteriores, como redes neurais recorrentes. Desde então, diversos LLMs foram desenvolvidos, destacando-se o GPT (*Generative Pre-trained Transformer*), o BERT (*Bidirectional Encoder Representations from Transformers*) e o T5 (*Text-to-Text Transfer Transformer*) (Vaswani et al., 2017; Brown et al., 2020). Neste trabalho, iremos utilizar como base o modelo GPT, popularizado pelo ChatGPT da empresa OpenAI.

2.2 Engenharia de *Prompt*

A engenharia de *prompt* refere-se às estratégias utilizadas para otimizar as respostas geradas por modelos de linguagem, como o ChatGPT. Diferentes técnicas podem ser empregadas para direcionar o modelo a produzir respostas mais precisas e contextualizadas (NEPTUNE.AI, 2024). A seguir são apresentadas as principais técnicas de engenharia de *prompt*.

2.2.1 Técnica *Zero-Shot*

Consiste em fornecer apenas a pergunta ao modelo em uma mensagem de usuário, sem exemplos prévios. A resposta é gerada pelo assistente com base no conhecimento geral aprendido durante o treinamento do modelo. Embora seja rápida e simples, essa abordagem pode resultar em respostas genéricas ou pouco estruturadas (RAJ, 2025).

2.2.2 Técnica *One-Shot*

Nesta técnica, é fornecido um único exemplo antes da solicitação real usando uma pergunta em uma mensagem de usuário, seguida de uma resposta do assistente e da pergunta real. Essa estratégia ajuda o modelo a compreender melhor o contexto esperado, gerando respostas mais alinhadas às necessidades do usuário (RAJ, 2025).

2.2.3 Técnica *Few-Shot*

Diferente da técnica One-Shot, essa abordagem inclui múltiplos exemplos antes da pergunta principal. Isso permite que o modelo entenda melhor o padrão desejado e produza respostas mais consistentes e relevantes (NEPTUNE.AI, 2024).

2.2.4 Técnica *Chain-of-Thought (CoT)*

Essa técnica incentiva o modelo a estruturar o raciocínio em etapas, explicitando o processo lógico antes da resposta final. Esse método tem se mostrado eficaz para melhorar a precisão e clareza das respostas, especialmente em problemas complexos. Essa técnica pode ser inserida em uma mensagem de *sistema*, ou, mais comum, inserindo “Vamos pensar passo a passo” no final de uma mensagem de *usuário* (FILHO, M, 2024).

2.2.5 Técnica *Persona*

Consiste em instruir o modelo a assumir um papel específico, como o de um *personal trainer* ou um especialista técnico. Essa técnica ajuda a gerar respostas mais coerentes com a perspectiva desejada, proporcionando interações mais naturais e alinhadas ao contexto pretendido (SAUNIER, 2023).

2.3. Tecnologias utilizadas

Nesta seção serão apresentadas as tecnologias utilizadas para o desenvolvimento do sistema, bem como os principais conceitos e definições necessários para a compreensão do trabalho.

2.3.1. ChatGPT

O ChatGPT é um Modelo de Linguagem de Grande Escala (LLM) baseado na família GPT (*Generative Pre-trained Transformer*), desenvolvido pela OpenAI para gerar respostas contextualizadas e coerentes em interações de linguagem natural. Diferentemente de modelos anteriores, sua arquitetura incorpora a técnica de *Reinforcement Learning from Human Feedback* (RLHF), permitindo o ajuste fino das respostas com base em preferências humanas, o que aprimora sua qualidade e relevância (Ouyang et al., 2022).

A evolução do modelo teve marcos importantes, começando com o lançamento do GPT-3, em junho de 2020. Esse modelo representou um salto significativo em capacidade de geração de texto, contando com 175 bilhões de parâmetros, o que possibilitou uma compreensão mais sofisticada e fluida das interações. Posteriormente, em março de 2023, a OpenAI lançou o GPT-4, expandindo suas capacidades para processar não apenas texto, mas também imagens, aumentando seu potencial de aplicação. Com essa versão, também foi disponibilizada uma API, possibilitando a integração do ChatGPT com diversas linguagens de programação, como C#, Python e PHP, viabilizando seu uso integrado em diversas aplicações.

A API do ChatGPT organiza as interações em um histórico de mensagens estruturado, no qual cada mensagem assume um dos três papéis: *sistema*, *usuário* ou *assistente*. O papel de *sistema* define o comportamento e o estilo das respostas do modelo, definindo o nível de formalidade e eventuais restrições no conteúdo gerado para toda a conversa. Já o papel *usuário* representa as entradas enviadas, incluindo perguntas, comandos e instruções específicas. Por fim, o *assistente* corresponde às respostas produzidas pelo modelo com base

no contexto das interações anteriores, garantindo coerência e continuidade na conversação (OPENAI, 2024). A Figura 2 ilustra um exemplo de interação com o ChatGPT onde um usuário solicita ajuda para elaborar um resumo acadêmico. Os papéis aparecem em negrito.

<p>Sistema: <i>Você é um assistente especializado em escrita acadêmica. Suas respostas devem ser formais e objetivas, evitando opiniões pessoais.</i></p> <p>Usuário: <i>Como estruturar um resumo acadêmico?</i></p> <p>Assistente: <i>Um resumo acadêmico deve conter uma breve introdução ao tema, os objetivos do trabalho, a metodologia utilizada, os principais resultados obtidos e a conclusão. Ele deve ser conciso, geralmente com até 250 palavras, e apresentar uma visão clara do conteúdo do artigo.</i></p>
--

Figura 2. Exemplo de conversação com o ChatGPT.

2.3.2. HTML

Conforme VIEIRA (2023), o HTML é uma linguagem de marcação criada para exibir conteúdo e construir a estrutura de páginas web. HTML é a sigla em inglês para *HyperText Markup Language* ou Linguagem para marcação de hipertexto. Uma página é formada por hipertexto, imagens, vídeos e gráficos que são chamados de hiperlinks. O hipertexto é o conteúdo textual do site que tem como sua maior característica a possibilidade de se interligar com outros documentos da web por meio de links.

2.3.3. CSS

O CSS é a abreviação de *Cascading Style Sheet* ou folha de estilos em cascata. O seu uso permite que os programadores criem layouts modulares deixando o HTML apenas para realizar a estrutura do site. Por meio do CSS é possível alterar a fonte, tamanho, cor e posições dos elementos de uma página ajustando o estilo conforme necessário. Além disso, por meio de seletores, do CSS é possível especificar os elementos alvo da estilização com bastante agilidade (EQUIPE TOTVS, 2020).

2.3.4. JavaScript

JavaScript é uma linguagem de programação orientada a objetos e multiplataforma, amplamente utilizada para tornar as páginas *web* mais interativas, oferecendo funcionalidades como animações, botões dinâmicos, menus e muito mais. Sua biblioteca padrão inclui objetos como *Array*, *Date* e *Math*, permitindo realizar diversas operações. Além disso, JavaScript pode ser utilizado para interagir com servidores, possibilitando a criação de aplicações *web* dinâmicas e a troca de dados em tempo real entre cliente e servidor (MOZILLA, 2025).

2.3.5. PHP

O PHP é uma linguagem de uso geral de *script open source*, muito utilizada em especial utilizada e adequada para o desenvolvimento web e com o seus recursos ela pode ser embutida dentro do HTML, apenas inserindo as suas tags de abertura e fechamento, isso permite entrar e sair do modo PHP no HTML. O diferencial desta linguagem é o processamento dentro do servidor junto com o HTML antes de ser encaminhado para o servidor. Desta forma, se pode configurar o servidor para processar o HTML junto com o PHP e encaminhar para o navegador do cliente (PHP, 2024).

2.3.6. Bootstrap

Originalmente o Bootstrap foi criado por Jacob e Mark Otto no Twitter no ano de 2010. Antes de ser lançado como uma estrutura de código aberto ele era conhecido como Twitter Blueprint. Conforme a documentação oficial, com o Bootstrap podem ser criados protótipos e aplicações completas com agilidade. Ele é uma ótima ferramenta para o desenvolvimento HTML, CSS e JS, contando com sistemas de *grid* responsivo, componentes pré-construídos e *plugins* com JQuery (BOOTSTRAP, 2018).

2.4 Trabalhos Relacionados

O uso de tecnologia da informação é bastante difundido na gestão de exercícios físicos. Hoje em dia existem diversas plataformas para gestão de exercícios físicos tanto para academias quanto para *personal trainers*. Eles se dividem em plataformas *web*, *desktop* e *mobile*, utilizando diversas linguagens, como PHP, Delphi, python e DBA, mas todos com o mesmo propósito que é ter o cadastro de alunos, exercícios e horários. Além disso, a grande maioria dos sistemas permite criar as fichas técnicas dos seus alunos e permite a criação de uma ficha de treinos com os exercícios, carga e quantidade de repetições. Essa ficha é feita em cima do cálculo do IMC, quantidade de dias que o aluno pode frequentar a academia e também se o aluno é iniciante ou já praticou ou pratica uma atividade física. Finalmente, é comum a todas as aplicações pesquisadas que o próprio personal ou professor tenha que gerar os treinos manualmente com os dados coletados nas fichas técnicas.

Na literatura científica é escasso o uso de inteligência artificial no apoio da gestão de exercícios. No entanto, Washif et al. (2024) apresenta um estudo avaliando os treinos produzidos pelo ChatGPT. Os autores propuseram para o chatGPT criar treinos a partir de métricas encaminhadas pelo profissional de educação física, informando parâmetros como quantidades de semanas, a experiência do cliente (intermediário ou avançado) com exercícios de academia, bem como o objetivo como perda de peso ou hipertrofia. O estudo testou esses dados de entrada em 3 versões do chatGPT e classificou as respostas produzidas como fraco, moderado e forte. As respostas foram comparadas com literatura confiável como o livro Associação Nacional de Força e Condicionamento (NSCA) e vários artigos de revisão. Os autores concluíram que o chatGPT criou informações realistas de treinos, mas que precisam ser revisadas ou alteradas para corrigir partes do treino produzido.

3. Metodologia de projeto

Nessa seção serão apresentados os processos utilizados para o desenvolvimento do projeto do sistema: o levantamento dos requisitos funcionais e não funcionais, o desenvolvimento do diagrama de casos de uso.

3.1. Requisitos do sistema

Conforme SOMMERVILLE (2011), a análise de requisitos tem como objetivo descobrir, analisar, documentar e verificar os requisitos do negócio, descrevendo o que um sistema deve fazer. Os requisitos são divididos em funcionais e não funcionais.

Os requisitos funcionais descrevem as funcionalidades do software que precisa ser desenvolvido. Por isso, as especificações dos requisitos funcionais devem ser completas e consistentes para evitar que equívocos resultem em retrabalho, manutenção e aumento do custo do sistema. Já os requisitos não funcionais não estão diretamente relacionados com os

serviços específicos do sistema. Ao invés disto, estes requisitos normalmente especificam ou restringem as características do sistema como desempenho, proteção e disponibilidade. Caso deixe um usuário encontrar uma maneira de contornar uma função e não ser verificado pelo desenvolvedor, isto pode ocasionar a inutilização de todo o sistema. Os requisitos não funcionais surgem por meio de necessidades do usuário, como a necessidade de interoperabilidade com outros softwares e hardwares ou também fatores externos como regulamentos de segurança ou legislação de privacidade (SOMMERVILLE, 2011). Na Tabela 1 são apresentados os requisitos funcionais (1-5) e não funcionais (6) elaborados de acordo com o estudo das necessidades do sistema proposto.

Tabela 1. Levantamento de requisitos

Id	Requisito	Descrição
1	Gerar um plano de treino	O usuário irá realizar a entrada de dados do aluno via formulário para geração do plano de treino.
2	Edição do plano gerado	O usuário terá a possibilidade de realizar alteração no plano gerado pela API do ChatGPT.
3	Salvar plano em PDF	O usuário terá a possibilidade de salvar o seu treino em PDF.
4	Realizar a impressão	O usuário poderá realizar a impressão direta do plano de treino.
5	Gerar um novo treino	Caso o usuário fique insatisfeito com o plano gerado deve ser possível gerar um novo treino usando um prompt diferente.
6	Linguagem de programação	O sistema deverá ser desenvolvido a partir da linguagem PHP.

3.2. Diagrama de Casos de Uso

Segundo SOMMERVILLE (2011), casos de uso servem para realizar descobertas de requisitos de software. Nesse diagrama, são identificados atores que irão realizar ações em uma ou mais interações com o sistema, dando nome ao tipo de interação. O conjunto destes casos de uso irá representar todas as possíveis interações de usuário ou com sistemas externos. Os atores, representados por pessoas ou outros softwares, são responsáveis por realizar as interações no sistema. Ações são os comportamentos que um ator (usuário ou software externo) irá realizar para atingir o objetivo específico dentro do caso de uso. Em resumo, o diagrama de caso de uso é composto pelos atores e suas ações possíveis para conseguir acessar as funcionalidades dos sistemas desenvolvidos. De acordo com os requisitos levantados na seção anterior foi elaborado o diagrama de casos de uso da Figura 3.

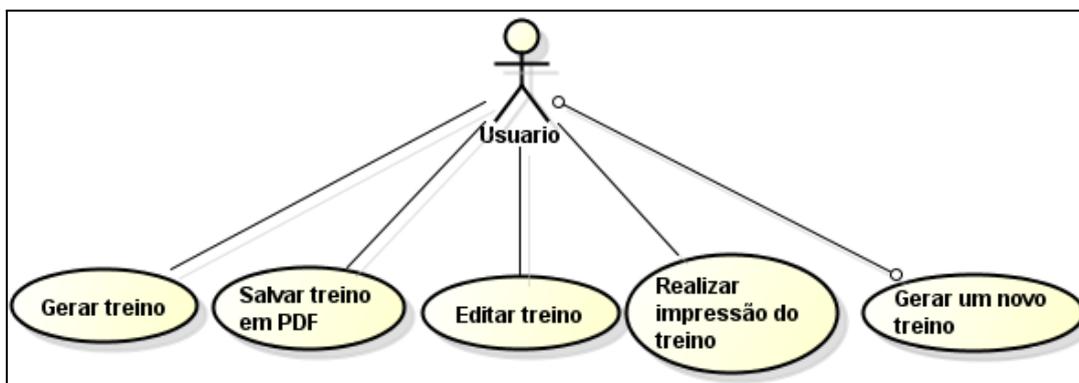


Figura 3. Diagrama de casos de uso

4. Implementação

Após a conclusão das etapas metodológicas do projeto iniciou-se o desenvolvimento do sistema. Na implementação, o primeiro passo foi utilizar a linguagem HTML para elaborar a estrutura inicial das páginas. A seguir, os requisitos do projeto foram implementados com o uso da linguagem PHP e a API do ChatGPT. Finalmente, foi realizada a sua estilização com CSS, Javascript e o *framework* Bootstrap. O código fonte do sistema, assim como os treinos gerados em experimentos estão disponíveis no seguinte repositório do Github em: <https://github.com/Rafael-Borella-Strzelecki/Gerador-de-treinos-Chat-GPT>. Nas seções seguintes são apresentados os resultados do sistema construído e suas principais funções.

4.1. Telas do Sistema

Ao acessar o sistema o usuário tem acesso ao formulário exibido na Figura 4. Após o envio dos dados do formulário, o usuário é encaminhado para a página de “Recomendação de treino” exibido na Figura 5. Nesta tela é apresentado um campo com o texto do treino gerado pela API do ChatGPT que pode ser editado. Ao clicar no botão "Salvar", uma versão em PDF do treino será gerada. Também é possível imprimir o treino clicando em "Imprimir" ou criar um novo clicando no botão "Gerar novo treino".

Dados do Aluno para Geração do Treino

Escolha o tipo de Prompt: Few Shot

Escolha a temperatura: 0.2

Nome: Rafael Borella Strzelecki

Genero: Masculino

Idade: 28

Experiencia de Treino: Avançado

Peso em Kg: 110

Altura em cm: 182

Possui lesão ou cirurgia? Não

Porcentagem de Gordura Corporal: 20

Selecione o objetivo do aluno: Hipertrofia

Enviar

Figura 4. Formulário para coleta de dados

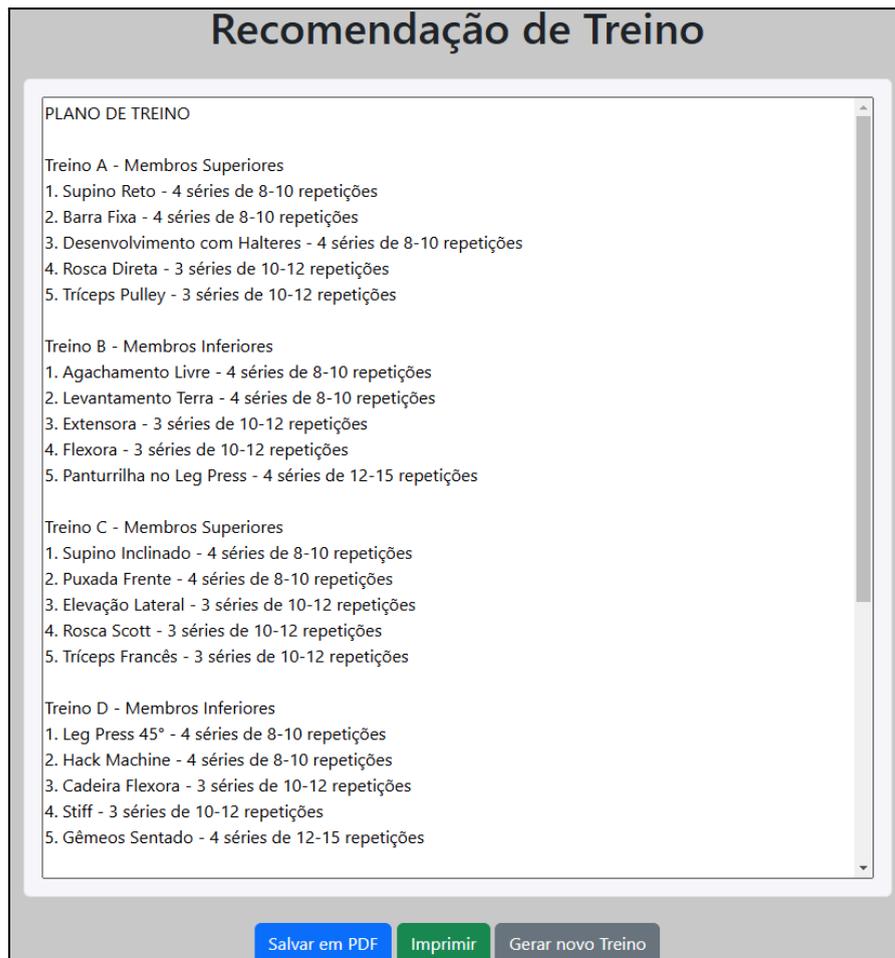


Figura 5. Tela de recomendação de treino

4.3. Geração de treinos usando a API do ChatGPT

Para integrar a API do ChatGPT ao sistema, é necessário gerar uma chave de acesso (API key) no site oficial da OpenAI. Essa chave autentica as requisições e permite que o sistema se comunique com o modelo de linguagem. Conforme ilustrado na Figura 6, são definidas duas variáveis: a primeira armazena a chave de acesso gerada, garantindo a segurança da comunicação, enquanto a segunda contém a rota da API responsável por processar as requisições e gerar respostas com base nos dados fornecidos pelo usuário.

```
1 // Configurações para a API da OpenAI
2 $api_key = 'Key chatGPT';
3 $url = 'https://api.openai.com/v1/chat/completions';
```

Figura 6. Configuração chave de acesso a API do ChatGPT

A Figura 7 apresenta um exemplo de implementação de geração de treino utilizando uma combinação da técnica Persona com a técnica Few-Shot. O código define a estrutura necessária para enviar a solicitação à API, encapsulando os dados em um *array* na variável `$data`. Nesta estrutura, “Model” define o modelo utilizado para a geração das respostas, neste

trabalho estamos usando a versão "gpt-3.5-turbo". A estrutura "Messages" define o conjunto de mensagens da conversação, cada uma com uma definição de papel. Finalmente, temos o parâmetro "temperature" que define a temperatura do modelo em valores entre 0 e 1.

Na estrutura "messages" a técnica Persona é informada no papel "System". A técnica Few-Shot é implementada por meio de três exemplos de interação entre "user" e "assistant", seguidos de uma solicitação final com o papel "user".

```
1  $data = [  
2      "model" => "gpt-3.5-turbo",  
3      "messages" => [  
4          ["role" => "system", "content" => $persona],  
5          ["role" => "user", "content" => $pergunta1],  
6          ["role" => "assistant", "content" => $exemplo1],  
7          ["role" => "user", "content" => $pergunta2],  
8          ["role" => "assistant", "content" => $exemplo2],  
9          ["role" => "user", "content" => $pergunta3],  
10         ["role" => "assistant", "content" => $exemplo3],  
11         ["role" => "user", "content" => $pergunta]  
12     ],  
13     "temperature" => $temperatura  
14 ];
```

Figura 7. Configuração do prompt

Após a inicialização do processo é configurada para encaminhar a requisição para a API no formato JSON. Em seguida, a resposta gerada é capturada e decodificada, garantindo a conversão adequada dos dados para uma "Recomendação de Treino" na Figura 8.

```
1  // Inicializa o cURL  
2  $ch = curl_init($url);  
3  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
4  curl_setopt($ch, CURLOPT_POST, true);  
5  curl_setopt($ch, CURLOPT_HTTPHEADER, [  
6      "Content-Type: application/json",  
7      "Authorization: Bearer $api_key",  
8      "Cache-Control: no-cache, no-store, must-revalidate",  
9      "Pragma: no-cache"  
10 ]);  
11 // Executa a requisição e captura a resposta  
12 $response = curl_exec($ch);  
13 curl_close($ch);  
14  
15 // Decodifica a resposta  
16 $resposta = json_decode($response, true);  
17 $conteudo = $resposta['choices'][0]['message']['content'] ?? 'Erro ao gerar a resposta.';
```

Figura 8. Envio e captura da resposta

5. Experimentos e Resultados

Nesta seção, são apresentados os experimentos realizados para avaliar a eficácia do sistema desenvolvido.

5.1 Conjunto de dados

A Tabela 2 apresenta um conjunto com dados de 8 alunos fictícios para avaliar os treinos produzidos pelo gerador proposto neste trabalho. Nessa tabela, a primeira coluna apresenta um identificador para o caso de teste, enquanto as demais apresentam valores de teste para as variáveis destacadas na Seção 2.1.2. Os valores foram definidos para contemplar cenários variados, abrangendo tanto indivíduos iniciantes quanto avançados, bem como fatores que podem impactar o plano de treino, como excesso de peso, lesões articulares e percentuais de gordura variados.

Id	Idade	Gênero	Peso (kg)	Altura (cm)	% Gordura	Experiência	Lesão	Objetivo
1. Maria	16	Feminino	50	160	25%	Iniciante	Não	Resistência
2. João	25	Masculino	80	175	18%	Iniciante	Não	Força
3. Ricardo	40	Masculino	90	180	22%	Intermediário	Não	Hipertrofia
4. Aline	28	Feminino	120	170	45%	Iniciante	Não	Hipertrofia
5. Paulo	45	Masculino	100	175	26%	Avançado	joelho esquerdo	Força
6. Bruna	60	Feminino	70	160	32%	Iniciante	Não	Resistência
7. Livia	20	Feminino	63	165	24%	Avançado	ombro direito	Hipertrofia
8. César	65	Masculino	75	170	28%	Intermediário	Não	Resistência

Tabela 2. Conjunto de dados de teste.

5.2. Prompts propostos para avaliação

Visto que a forma como um *prompt* é organizado interfere diretamente na qualidade e precisão das respostas do ChatGPT, neste trabalho foram propostos cinco *prompts* distintos que combinam técnicas de engenharia de *prompt*. A seguir, na Figura 9, apresenta-se uma descrição resumida de cada *prompt* avaliado. No conteúdo das mensagens, denotamos o uso de variáveis oriundas do formulário do gerador usando a notação: <genero>, <idade>, <objetivo>. Além disso, para fins de compacidade, abreviamos as *strings* longas ou repetidas usando "...". Finalmente, denotamos a resposta final do assistente que contém o treino gerado como <resposta>.

Prompt P1 = Persona + Zero-Shot
<p>Sistema: <i>Você é um profissional de educação física experiente que atua como Personal Trainer prescrevendo treinos de academia.</i></p> <p>Usuário: <i>Crie um plano de treino detalhado para academia considerando as seguintes informações de um aluno: - Gênero: <genero> - Idade: <idade> - ... - Objetivo: <objetivo></i></p> <p>Assistente: <resposta></p>
Prompt P2 = Persona + One-Shot
<p>Sistema: <i>Você é um profissional de educação física experiente que atua como...</i></p> <p>Usuário: <i>Crie um plano de treino detalhado para academia considerando as seguintes informações de um aluno: - Gênero: Masculino - Idade: 28 - ... - Objetivo: Hipertrofia.</i></p> <p>Assistente: <i>PLANO DE TREINO Treino A Exercícios: Rosca Martelo (Halter) 4x, Rosca Direta (Cross) 4x, ... ORGANIZAÇÃO SEMANAL Dia 1: Treino A, Dia 2: Treino B, ... Dia 7: Repouso.</i></p> <p>Usuário: <i>Crie um plano ... um aluno: - Gênero: <genero> - Idade: <idade> - ... - Objetivo: <objetivo></i></p> <p>Assistente: <resposta></p>
Prompt P3 = Persona + Few-Shot
<p>Sistema: <i>Você é um profissional de educação física experiente que atua como...</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: Masculino - Idade: 28 - ... - Objetivo: Hipertrofia.</i></p> <p>Assistente: <i>PLANO DE TREINO Treino A Exercícios: Rosca Martelo (Halter) 4x, Rosca Direta (Cross) 4x, ... ORGANIZAÇÃO SEMANAL Dia 1: Treino A, Dia 2: Treino B, ... Dia 7: Repouso.</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: Masculino - Idade: 29 - .. - Objetivo: Hipertrofia</i></p> <p>Assistente: <i>PLANO DE TREINO Treino A Exercícios: Remada curvada 4x, Puxada alta 3x, Remada cavalinho 4x, ORGANIZAÇÃO SEMANAL Dia 1: Treino A, ..., Dia 7: Repouso.</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: Feminino - Idade: 45 - ... - Objetivo: Resistência</i></p> <p>Assistente: <i>PLANO DE TREINO Treino A Exercícios: Mesa Flexora + Stiff 3x, Bulgaro 3x, Elevação Pélvica 4x, ... ORGANIZAÇÃO SEMANAL Dia 1: Treino A, ..., Dia 7: Repouso.</i></p> <p>Usuário: <i>Crie um plano ... um aluno: - Gênero: <genero> - Idade: <idade> - ... - Objetivo: <objetivo></i></p> <p>Assistente: <resposta></p>
Prompt P4 = Persona + CoT
<p>Sistema: <i>Você é um profissional de educação ... prescrevendo treinos de academia. Por favor, explique seu raciocínio passo a passo e depois apresente um plano de treino de academia completo.</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: <genero> - Idade: <idade> - - Objetivo: <objetivo>. Vamos pensar passo a passo.</i></p> <p>Assistente: <resposta></p>
Prompt P5 = Persona + CoT + One-Shot
<p>Sistema: <i>Você é um profissional de educação ... prescrevendo treinos de academia. Por favor, explique seu raciocínio passo a passo e depois apresente um plano de treino de academia completo.</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: Masculino - Idade: 28 - ... - Objetivo: Hipertrofia.</i></p> <p>Assistente: <i>PLANO DE TREINO Treino A Exercícios: Rosca Martelo (Halter) 4x, Rosca Direta (Cross) 4x, ... ORGANIZAÇÃO SEMANAL Dia 1: Treino A, Dia 2: Treino B, ... Dia 7: Repouso.</i></p> <p>Usuário: <i>Crie um plano ... de um aluno: - Gênero: <genero> - Idade: <idade> - - Objetivo: <objetivo>. Vamos pensar passo a passo.</i></p> <p>Assistente: <resposta></p>

Figura 9. Estrutura dos Tipos de Prompt avaliados.

5.3 Critérios de Avaliação

A avaliação de cada plano de treino gerado pelo sistema proposto foi realizada com base em quatro critérios principais: clareza e organização, adequação ao objetivo, lesões e segurança e coerência prática. Esses critérios foram definidos com base no referencial teórico apresentado na Seção 2.1. O objetivo desta avaliação não é substituir a avaliação de um profissional de Educação Física, mas fornecer diretrizes gerais mínimas para avaliar a qualidade dos treinos. Cada critério será pontuado em uma escala de 1 a 3 e serão detalhados a seguir.

5.3.1 Critério 1 - Clareza e Organização

Esse critério avalia se o plano está apresentado de forma estruturada e compreensível, garantindo que mesmo um iniciante consiga interpretá-lo sem dificuldades. A avaliação desse critério é focada na apresentação e não no conteúdo do treino, ou seja, considera se os exercícios, séries, repetições e intervalos de descanso estão exibidos de maneira padronizada e se há uma organização semanal presente. Esse critério será pontuado assim:

1. **Insatisfatório:** informações importantes faltando, desorganizadas ou incompreensíveis.
2. **Adequado:** apresenta dados básicos, mas com pequenos problemas de formatação.
3. **Excelente:** informações claras, bem estruturadas e de fácil leitura.

5.3.2 Critério 2 - Adequação ao Objetivo

Esse critério avalia se as diretrizes gerais do treino, como o número de séries, repetições e intervalos são compatíveis com o objetivo, seja hipertrofia, força ou resistência. Este critério será pontuado assim:

1. **Insatisfatório:** séries/repetições/intervalos divergem significativamente do objetivo.
2. **Adequado:** condiz com o objetivo de modo geral, mas pode ter inconsistências.
3. **Excelente:** atende as diretrizes gerais de acordo com o objetivo.

5.3.3 Critério 3 - Lesões e Segurança

Esse critério avalia se o plano de treino leva em consideração as limitações de grupos de risco, como pessoas com lesões, iniciantes, idosos ou pessoas com peso excessivo. Este critério será pontuado assim:

1. **Insatisfatório:** ignora completamente as limitações de alunos em grupo de risco.
2. **Adequado:** reconhece a presença de uma limitação, mas a orientação é superficial.
3. **Excelente:** apresenta orientação clara para proteger lesões e/ou manter a segurança em grupos de risco.

5.3.4 Critério 4 - Coerência Prática

Esse critério avalia se o conteúdo do treino faz sentido no dia a dia, considerando a divisão de grupos musculares, os dias de descanso, a ordem dos exercícios e a viabilidade de execução. Este critério será pontuado assim:

1. **Insatisfatório:** treino impraticável ou grande desequilíbrio entre grupos musculares..
2. **Adequado:** treino com divisão razoável mas com algum nível de desequilíbrio.
3. **Excelente:** treino equilibrado e aplicável que separa os grupos musculares de forma racional e com descanso apropriado.

5.4. Resultados dos Experimentos

A análise dos experimentos foi realizada com base na avaliação dos planos de treino gerados pelo sistema proposto. Para isso, foram conduzidos 80 testes, combinando os oito casos de teste previamente definidos, cinco variações de *prompts* e duas temperaturas ($T=0,2$ e $T=0,7$) na API do ChatGPT. O objetivo desta avaliação é identificar qual combinação gera planos de treino mais coerentes, seguros e adequados aos objetivos dos alunos.

Os resultados obtidos são exibidos na Tabela 3, onde cada tipo de prompt foi avaliado com base em quatro critérios: clareza e organização, adequação ao objetivo, consideração de lesões e segurança e coerência prática. Cada critério foi pontuado em uma escala de 1 a 3, conforme descrito na seção 5.3. Os melhores resultados são destacados em negrito.

T	Tipo de Prompt	Critério 1: Clareza e Organização	Critério 2: Adequação ao objetivo	Critério 3: Lesões e Segurança	Critério 4: Coerência Prática	Média
0,2	P1 = Persona + Zero-Shot	2,38	1,38	2,88	1,50	2,03
0,2	P2 = Persona + One-Shot	2,88	2,25	2,50	2,13	2,44
0,2	P3 = Persona + Few-Shot	3,00	2,38	2,75	3,00	2,78
0,2	P4 = Persona + CoT	2,50	2,00	2,88	2,00	2,34
0,2	P5 = Persona + CoT + One-Shot	2,88	1,75	2,88	2,25	2,44
0,7	P1 = Persona + Zero-Shot	2,50	1,88	2,63	2,13	2,28
0,7	P2 = Persona + One-Shot	2,88	2,38	2,63	2,63	2,63
0,7	P3 = Persona + Few-Shot	3,00	2,38	2,00	2,88	2,56
0,7	P4 = Persona + CoT	2,75	2,13	2,75	2,50	2,53
0,7	P5 = Persona + CoT + One-Shot	2,63	2,00	2,75	2,50	2,47

Tabela 3. Resultados computacionais.

A análise dos resultados revela que a geração de treinos usando o *prompt* P3, baseado na técnica Few-Shot com temperatura 0,2, apresentou o melhor desempenho geral, atingindo uma média de 2,78. Essa abordagem, alcançou pontuação máxima nos critérios 1 e 4, gerando gerar treinos claros e organizados, bem como garantindo que os parâmetros de séries, repetições e intervalos de descanso estivessem alinhados com as diretrizes apresentadas na Seção 2.1. Também podemos perceber que essa mesma abordagem com temperatura 0,7 reduziu significativamente as precauções de segurança, indicando que respostas determinísticas favorecem essa abordagem que utiliza vários exemplos.

Em contrapartida, o pior desempenho foi observado na geração de treinos com o *prompt* P1, baseado na abordagem Zero-Shot com temperatura 0,2, que obteve uma média de 2,03. Isso demonstra que a ausência de exemplos fornecidos ao modelo prejudica a adequação dos treinos aos objetivos dos alunos, além de resultar em instruções pouco estruturadas e, em alguns casos, incoerentes com a segurança necessária para alunos com restrições físicas. A elevação da temperatura para 0,7 contribuiu para um leve aprimoramento

dos resultados nesta abordagem, mas não o suficiente para alcançar o desempenho dos demais métodos.

Em relação ao *prompt* P2, baseado em One-Shot, foi obtido desempenho intermediário com uma média de 2,44 na temperatura 0,2 e 2,63 na temperatura 0,7. Esse resultado sugere que a inclusão de um exemplo inicial ajudou a estruturar melhor os treinos, embora ainda apresentasse inconsistências em alguns casos. Já o *prompt* P4, baseado na técnica Chain-of-Thought (CoT), mostrou-se mais eficaz na temperatura 0,7, obtendo uma média de 2,53, também com um desempenho intermediário. Essa abordagem permitiu ao modelo gerar planos de treino com uma das maiores pontuações no critério de lesão e segurança, sugerindo que com um raciocínio mais passo a passo é possível trazer melhores considerações aos grupos de risco. Outro *prompt* com desempenho semelhante foi o P5, que combinou CoT e One-Shot, reforçando que a explicitação do raciocínio favorece a segurança e pode ser levemente melhorando com o fornecimento de um exemplo.

Finalmente, entre os critérios de avaliação, o critério com melhor pontuação foi a consideração de lesões e segurança, atingindo valores médios superiores a 2,5 em quase todas as abordagens. Isso demonstra que o modelo foi capaz de levar em conta restrições físicas de forma satisfatória. Por outro lado, a adequação ao objetivo obteve as pontuações mais baixas, indicando que, em alguns casos, as repetições, séries e intervalos de descanso não seguiram as diretrizes recomendadas para cada tipo de treinamento (hipertrofia, resistência ou força) da mesma forma que foi explicitado na Seção 2.1.

6. Considerações finais

Com o objetivo de agilizar a personalização de planos de treino de alunos de academia, esse trabalho apresentou a construção de um sistema *web* implementado na linguagem de programação PHP com foco em atender os *Personal Trainers* que necessitam agilizar o processo de prescrição de treinos. O sistema desenvolvido compreende desde a geração de um plano de treino com a API do ChatGPT até a impressão em PDF ou papel.

O sistema proposto possibilita a criação dos planos de treinos personalizados e também a edição deste plano para melhor atender cada aluno do *Personal Trainer*, fornecendo comodidade e agilidade nas criações, reduzindo o tempo necessário para criar um novo plano de treino. Também foram conduzidos experimentos computacionais em um conjunto de dados fictício, avaliando 5 tipos de *prompts* em 2 temperaturas diferentes. Os resultados dos experimentos demonstraram que a abordagem que combina as técnicas de *Persona* e *Few-Shot* com temperatura 0,2 é mais eficiente para a geração de treinos personalizados, conciliando clareza, segurança e coerência prática.

São sugestões de trabalhos futuros: (i) possibilidade de avaliação do banco de dados de treinos gerados por um profissional de educação física; (ii) estender o sistema para permitir o cadastro de alunos e treinos em um banco de dados; (iii) avaliar novas estratégias de combinação de prompts com outros parâmetros além da temperatura.

Referências

AMERICAN COLLEGE OF SPORTS MEDICINE. ACSM's Guidelines for Exercise Testing and Prescription. 10. ed. Philadelphia: Wolters Kluwer, 2017.

- BAECHLE, T. R.; EARLE, R. W. Essentials of Strength Training and Conditioning. 3. ed. Champaign: Human Kinetics, 2008.
- BOOTSTRAP. Documentação oficial do Bootstrap. Disponível em: <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/>. Acesso em: 29 abr. 2024.
- BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; AMODEI, D. Language models are few-shot learners. In: Advances in Neural Information Processing Systems (NeurIPS), v. 33, p. 1877–1901, 2020.
- CHARRO, M. A.; JOÃO, G. A.; FIGUEIRA JR., A. J. Musculação: estruturação do treinamento e controle de carga. São Paulo: CREF4/SP, 2018.
- FILHO, M. Chain-of-Thought Prompting: O que é e como usar? Mario Filho, 10 jul. 2024. Disponível em: https://mariofilho.com/chain-of-thought-prompting/?utm_source=chatgpt.com. Acesso em: 26 jan. 2025.
- HEYWARD, V. H. Advanced Fitness Assessment and Exercise Prescription. 7. ed. Champaign: Human Kinetics, 2018.
- JONES, N. L. et al. Human Physiology: The Mechanisms of Body Function. 3. ed. Philadelphia: W.B. Saunders, 1985.
- KIBLER, W. B. et al. Rehabilitation of sports injuries: scientific basis. Medicine and Science in Sports and Exercise, v. 45, n. 7, p. 1270-1277, 2013.
- KRIEGER, J. W. Single vs. multiple sets of resistance exercise for muscle hypertrophy. Journal of Strength and Conditioning Research, v. 24, n. 4, p. 1150-1159, 2010.
- MATSUDO, V. K. R.; MATSUDO, S. M. M.; BARROS NETO, T. L. Efeitos do exercício no envelhecimento: revisão e recomendações. Revista Brasileira de Medicina do Esporte, v. 6, n. 6, p. 333-340, 2000.
- MCGILL, S. Low Back Disorders: Evidence-Based Prevention and Rehabilitation. 2. ed. Champaign: Human Kinetics, 2007.
- MOZILLA. O que é JavaScript? MDN Web Docs. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Introduction#o_que_%C3%A9_javascript. Acesso em: 2 fev. 2025.
- NEGÓCIOS SC. Mercado fitness: como as academias podem crescer no Brasil. Disponível em: <https://www.negociossc.com.br/blog/mercado-fitness-como-as-academias-podem-crescer-no-brasil/>. Acesso em: 17 mai. 2024.
- NEPTUNE.AI. Zero-shot and few-shot learning with LLMs. Disponível em: <https://neptune.ai/blog/zero-shot-and-few-shot-learning-with-llms>. Acesso em: 26 jan. 2025.

- OPENAI. ChatGPT. Disponível em: <https://openai.com/index/chatgpt/>. Acesso em: 08 set. 2024.
- OUYANG, L.; WU, J.; JIANG, X.; ALMEIDA, D.; WAINWRIGHT, C.; MISHKIN, P.; LEIKE, J. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155, 2022.
- PHP. Histórico de PHP. Disponível em: https://www.php.net/manual/pt_BR/history.php.php. Acesso em: 14 abr. 2024.
- RAJ, R. Zero-shot, one-shot and few-shot learning with examples. Medium, 2025. Disponível em: <https://rahulrajpv7d.medium.com/zero-shot-one-shot-and-few-shot-learning-with-examples-8a3efdccb158>. Acesso em: 26 jan. 2025.
- RENZE, M.; GUVEN, E. The effect of sampling temperature on problem solving in large language models. arXiv preprint arXiv:2402.05201, 2024.
- RHEA, M. R. et al. A comparison of linear and undulating periodized programs with equated volume and intensity for novice and trained college women. *Journal of Strength and Conditioning Research*, v. 17, n. 1, p. 82-87, 2003.
- SAUNIER, Carlos. ChatGPT: Prompts avançados. Blog de Carlos Saunier, 2023. Disponível em: <https://carlossaunier.com/blog/chatgpt-prompts-avanzados/> Acesso em: 26 jan. 2025.
- SCHOENFELD, B. J. The mechanisms of muscle hypertrophy and their application to resistance training. *Journal of Strength and Conditioning Research*, v. 24, n. 10, p. 2857-2872, 2010.
- SOMMERVILLE, Ian. Engenharia de Software. Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves. Revisão técnica de Kechi Hiramã. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
- TEIXEIRA, M. D. M. et al. Estudo comparativo da força muscular da mão entre cadetes homens e mulheres da Força Aérea Brasileira. *Fisioterapia e Pesquisa*, São Paulo, v. 16, n. 2, p. 143-147, abr./jun. 2009.
- TOTVS. O que é CSS? Disponível em: <https://www.totvs.com/blog/developers/o-que-e-css/> . Acesso em: 14 abr. 2024.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; POLOSUKHIN, I. Attention is all you need. In: *Advances in Neural Information Processing Systems (NeurIPS)*, v. 30, 2017.
- VIEIRA, DOUGLAS, HOSTGATOR. Conheça o HTML. Disponível em: <https://www.hostgator.com.br/blog/conheca-o-html/>. Acesso em: 14 abr. 2024.
- WASHIF, J. A.; PAGADUAN, J.; JAMES, C.; DERGAA, I.; BEAVEN, C. M. Artificial intelligence in sport: Exploring the potential of using ChatGPT in resistance training prescription. Vol. 41, 2/2024.