

Sistema de gestão de treinos de musculação

Leonardo Barros Schroter, Herysson Rodrigues Figueiredo

Curso de Sistemas de Informação

UFN - Universidade Franciscana

Santa Maria - RS

leonardo.schroter@ufn.edu.br, herysson.figueiredo@ufn.edu.br

Resumo—Este trabalho mostra o desenvolvimento de um sistema web para gestão de treinos de musculação, voltado a personal trainers. A solução propõe facilitar o acompanhamento, a personalização e a organização dos treinos de alunos, integrando conceitos de usabilidade, segurança da informação e inteligência artificial. A aplicação é construída utilizando a plataforma .NET, com arquitetura MVC, C#, Entity Framework e SQL Server, e permite o gerenciamento de usuários, anamneses, exercícios e fichas de treino. Como diferencial, o sistema incorpora um módulo de geração automática de treinos com base no modelo de linguagem Gemini, possibilitando sugestões personalizadas conforme o perfil do aluno. A metodologia ágil Kanban foi empregada para o gerenciamento do desenvolvimento, garantindo flexibilidade e acompanhamento contínuo do progresso.

Palavras-chave : Sistema de Gestão; Musculação; Aplicação Web; .NET; Kanban.

I. INTRODUÇÃO

A prática regular de atividade física é um fator essencial para a promoção da saúde e a prevenção de diversas doenças. Segundo a Organização Mundial da Saúde (OMS), a realização de exercícios físicos contribui para a redução do risco de doenças cardiovasculares, diabetes tipo 2 e determinados tipos de câncer, além de minimizar sintomas de transtornos mentais, como ansiedade e depressão [1].

A crescente digitalização do setor fitness tem impulsionado o desenvolvimento de soluções tecnológicas voltadas para a gestão de treinos e acompanhamento de alunos por personal trainers. Um levantamento realizado pela *Tecnofit*, uma das principais plataformas de gestão fitness no Brasil, apontou um aumento de 32,76% no número de personal trainers em atividade entre os anos de 2022 e 2023, considerando a base de clientes da empresa. Além disso, foi identificado um crescimento de 119% no número de treinos confirmados por alunos no mesmo período, evidenciando a crescente busca por acompanhamento profissional na prescrição e execução de atividades físicas [2].

Como demonstrado no estudo de Santos (2023), o uso de ferramentas digitais tem-se mostrado cada vez mais relevante e benéfico para os profissionais da área. Os aplicativos são amplamente aceitos e contribuem significativamente para a organização e eficiência no trabalho dos educadores físicos, proporcionando um acompanhamento mais eficaz dos treinos e uma melhor gestão das agendas [3].

A. Objetivo

Este trabalho teve como objetivo desenvolver uma aplicação web voltada para personal trainers, com o intuito de facilitar

o gerenciamento e a organização dos treinos de seus alunos.

B. Objetivos específicos

- Realização de pesquisa sobre sistemas de gestão de treinos existentes.
- Estudo sobre o funcionamento do processo de criação de treinos.
- Implementação da utilização de IA (Google Gemini) para sugestão de treinos ao personal trainer.
- Disponibilização de fácil acesso à ficha de treino para alunos e personals.
- Garantia de que o treino pudesse ser alterado pelo personal trainer, conforme a necessidade do aluno.

A Metodologia detalha o uso do Kanban para o planejamento e gerenciamento das tarefas, bem como a fase de projeto, que apresenta a modelagem de dados e os protótipos de interface que guiaram o desenvolvimento. A seção de Resultados demonstra a aplicação funcional finalizada, evidenciando a integração com Inteligência Artificial e a responsividade do sistema através das telas reais de uso, enquanto a conclusão valida o cumprimento dos objetivos propostos.

II. REFERENCIAL TEÓRICO

Nesta seção, são apresentados os fundamentos teóricos que sustentam o desenvolvimento deste trabalho, abrangendo conceitos relacionados ao exercício físico, sistemas de informação, aplicações web, experiência do usuário e as tecnologias e ferramentas utilizadas.

A. Exercício físico

O exercício físico é uma forma específica de atividade física que se caracteriza por ser planejado, estruturado e repetitivo, com o propósito de aprimorar ou manter componentes da aptidão física [4]. Além de resultar em um gasto energético controlado, o exercício físico também melhora a autoestima do praticante e seu perfil lipídico, o que contribui para uma melhor qualidade de vida [1].

1) *Musculação*: A musculação, também chamada de treinamento de força, é uma prática que evoluiu para uma ciência focada no aumento da força, resistência e hipertrofia muscular [5]. Para otimizar os resultados, é essencial a utilização da divisão de treinos uma estratégia que organiza a rotina de exercícios (seja por grupos musculares, corpo inteiro ou periodização) e deve ser personalizada, considerando o nível de experiência do praticante, seus objetivos e a frequência de

treinamento [5]. A prática da musculação, ao exigir acompanhamento e individualização, leva diretamente à necessidade de uma avaliação inicial detalhada, o que evidencia a importância da anamnese no contexto do treinamento.

2) *Anamnese*: A anamnese consiste na coleta sistemática de dados sobre a saúde e hábitos do praticante, sendo um passo essencial na avaliação inicial. A anamnese permite ao profissional de Educação Física obter informações detalhadas sobre o histórico de saúde e hábitos de vida do indivíduo. Segundo as orientações propostas por Cesana et al. [6], a anamnese deve abranger os seguintes aspectos:

- Histórico médico: levantamento de doenças pré-existent, lesões anteriores, cirurgias e uso de medicamentos.
- Estilo de vida: informações sobre hábitos alimentares, padrão de sono, nível de estresse, consumo de álcool, tabagismo e outras substâncias.
- Nível de atividade física: frequência, intensidade, tipo e duração das atividades físicas praticadas regularmente.
- Objetivos pessoais: metas relacionadas à saúde, condicionamento físico ou desempenho esportivo que o indivíduo deseja alcançar.

A realização de uma anamnese detalhada é essencial para identificar possíveis fatores de risco, estabelecer um plano de exercícios seguro e eficaz, e monitorar a evolução do praticante ao longo do tempo. Além disso, contribui para a prevenção de lesões e para a promoção de um estilo de vida mais saudável.

Com base nas informações obtidas na anamnese, o papel do personal trainer torna-se ainda mais relevante, pois é ele quem transforma esses dados em um plano de ação individualizado e eficaz.

3) *Personal trainer*: O personal trainer é um profissional de Educação Física que atua na prescrição e supervisão de exercícios personalizados, auxiliando indivíduos na busca por objetivos relacionados à saúde e ao condicionamento físico. Segundo Bossle [7], sua função vai além do treinamento, pois ele também influencia aspectos subjetivos e comportamentais de seus clientes, promovendo o "cuidado de si", que envolve a atenção e o investimento na própria saúde e bem-estar.

Vale destacar que a atuação desse profissional é regulamentada pela Lei nº 9.696/98, que reconhece a profissão de Educação Física e institui os respectivos Conselhos, e pela Resolução CONFEF nº 082/2004, que estabelece critérios para o serviço de treinamento personalizado [8].

B. Sistemas de informação

Os sistemas de informação são fundamentais para o funcionamento das organizações atuais, pois oferecem suporte à tomada de decisão, à coordenação e ao controle das atividades empresariais. Segundo Laudon e Laudon [9], um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados que coleta, processa, armazena e distribui informações para dar suporte às operações, à gestão e à análise estratégica de uma organização.

1) *Sistemas de gestão*: Os sistemas de gestão são aplicações desenvolvidas para organizar, controlar e disponibilizar informações de maneira prática e eficiente. Segundo Laudon e Laudon [9], sistemas de gestão consistem em ferramentas que capturam e processam dados para apoiar atividades organizacionais, facilitando a tomada de decisão e a otimização de processos. Quando hospedados na web, esses sistemas oferecem vantagens como acesso remoto, atualização centralizada de dados, escalabilidade e integração facilitada com outras tecnologias.

2) *Segurança*: A segurança da informação desempenha um papel essencial nos sistemas computacionais modernos, sobretudo em aplicações web que lidam com dados sensíveis dos usuários. Segundo Whitman e Mattord [10], a segurança da informação envolve práticas e políticas que visam proteger os dados e os sistemas contra acessos não autorizados, modificações indevidas, destruição acidental ou intencional, assegurando assim os três pilares fundamentais: confidencialidade, integridade e disponibilidade da informação. A confidencialidade garante que apenas pessoas autorizadas tenham acesso aos dados; a integridade assegura que as informações não sejam alteradas indevidamente; e a disponibilidade garante que os dados estejam acessíveis sempre que necessário.

No contexto brasileiro, destaca-se a Lei Geral de Proteção de Dados (Lei nº 13.709/2018) [11], também conhecida como LGPD, que estabelece diretrizes rigorosas para o tratamento de dados pessoais. A LGPD exige que organizações adotem medidas técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão. A lei reforça a importância de políticas de segurança bem estruturadas, especialmente em sistemas digitais voltados ao público.

C. Aplicação web

As aplicações web são sistemas de software que operam sobre uma rede, como a internet ou uma intranet, utilizando navegadores para interação com o usuário. De acordo com Laudon e Laudon [9], essas aplicações desempenham papel fundamental na transformação dos negócios modernos, permitindo comunicação, colaboração e execução de processos administrativos e operacionais através de plataformas digitais acessíveis de qualquer lugar.

1) *Arquitetura MVC*: A arquitetura MVC é um padrão de design utilizado em aplicações web para separar a lógica de negócios, a interface de usuário e o controle das interações. O Modelo (*Model*) gerencia os dados e a lógica de negócios, o Controlador (*Controller*) interage com o Modelo e a Visão, e a Visão (*View*) exibe a interface com o usuário. Esse padrão facilita a organização e manutenção do código, permitindo a modificação de uma parte da aplicação sem afetar as outras [12].

D. UX/UI

A experiência do usuário (*User experience/UX*) e a interface do usuário (*User interface/UI*) são componentes essenciais

no desenvolvimento de aplicações web, especialmente em sistemas voltados para o público final. Enquanto a *UI* se refere à apresentação visual e à interação direta com o sistema, a *UX* engloba toda a jornada do usuário, desde a navegação até a satisfação final com o uso da aplicação. Segundo Garrett [13], a construção de uma boa experiência digital exige atenção a cinco planos: estratégia, escopo, estrutura, esqueleto e superfície, todos conectados para garantir que o produto atenda às necessidades do usuário de forma eficiente e intuitiva.

1) *Responsividade*: O design responsivo, introduzido por Frain [14], visa adaptar o layout de uma aplicação a diferentes tamanhos de tela, começando pelo menor viewport e progressivamente aprimorando-o para telas maiores. A ideia central é apresentar o conteúdo de maneira acessível em qualquer dispositivo, garantindo uma experiência otimizada para o usuário.

2) *Usabilidade*: No contexto de aplicações web, concerne à capacidade de usuários específicos alcançarem seus objetivos com eficácia, eficiência e satisfação em um dado contexto [15]. Nielsen [16] destaca cinco atributos-chave: facilidade de aprendizado, eficiência, memorabilidade, poucos erros e satisfação. Uma aplicação com alta usabilidade permite que os usuários naveguem intuitivamente, realizem tarefas sem frustração e percebam valor na sua utilização, impactando diretamente a adoção e retenção.

E. Kanban

O *Kanban* é uma abordagem ágil voltada à visualização do trabalho, limitação de tarefas em andamento e melhoria contínua dos processos. Baseado nos princípios do *Lean Thinking*, o *Kanban* busca aumentar a eficiência e a previsibilidade do fluxo de trabalho ao reduzir desperdícios e tornar os processos mais transparentes e adaptáveis. Não se limita a um simples quadro de tarefas, mas constitui um sistema completo de gerenciamento ágil, aplicável a diversas áreas, especialmente no desenvolvimento de software e na gestão de produtos [17].

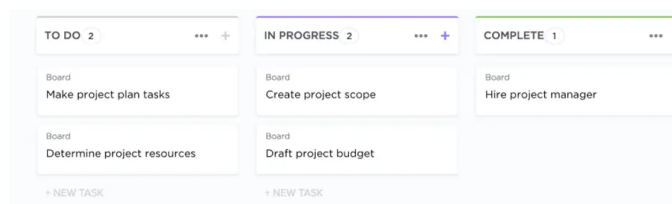


Figura 1. Quadro Kanban com estágios do fluxo de trabalho [18]

De acordo com a figura 1, no *Kanban*, o trabalho é representado por cartões que se movem por colunas organizadas conforme os estágios de execução das tarefas. O quadro facilita a visualização em tempo real do fluxo de trabalho, a identificação de gargalos e a atuação rápida na resolução de problemas. Diferentemente de métodos iterativos, como o *Scrum*, o *Kanban* opera em fluxo contínuo: as tarefas são puxadas do backlog de acordo com a capacidade da equipe, sem necessidade de ciclos temporais fixos.

A Figura 1 apresenta um exemplo de quadro *Kanban* com três colunas básicas, que representam os estágios comuns de uma tarefa:

- *A Fazer/To Do*: reúne as tarefas ainda não iniciadas, funcionando como um backlog visual, com itens priorizados.
- *Em Andamento/In Progress*: mostra as tarefas em execução, respeitando os limites de *WIP*.
- *Concluído/Complete*: apresenta as tarefas finalizadas, servindo de base para controle e análise de desempenho.

F. Inteligência Artificial (IA)

Refere-se ao desenvolvimento de sistemas computacionais capazes de executar tarefas que normalmente exigem inteligência humana, como reconhecimento de padrões, tomada de decisão e aprendizado a partir de dados [19]. O avanço recente de técnicas como redes neurais profundas tem impulsionado a aplicação da IA em diversas áreas, incluindo o Processamento de Linguagem Natural (PLN) [20].

1) *Modelos de Linguagem de Grande Escala (LLMs/Large Language Models)*: Estes modelos são redes neurais treinadas com grandes volumes de texto, utilizando arquiteturas como Transformers. Esses modelos, como o GPT-4, são capazes de compreender e gerar linguagem natural com alta precisão, sendo amplamente utilizados em aplicações como assistentes virtuais, tradução e sumarização automática [21], [22].

2) *Gemini 1.5 (Google DeepMind)*: É um modelo lançado em 2024 com capacidades multimodais e integração profunda com o ecossistema Google. Seu maior diferencial está no suporte a janelas de contexto extremamente amplas (mais de 1 milhão de *tokens*), o que o torna ideal para análise de documentos extensos e tarefas que requerem memória contextual de longo prazo [23].

G. Tecnologias e ferramentas

A seguir, serão descritas as principais tecnologias adotadas no projeto, com destaque para suas funcionalidades.

1) *.NET*: É uma plataforma de desenvolvimento de software criada pela Microsoft, que oferece um conjunto abrangente de ferramentas, bibliotecas e APIs para criar aplicativos de alta performance e escaláveis. A plataforma é open-source e suportada por diversas linguagens de programação, como C#, F#, e Visual Basic. O *.NET* oferece um ambiente robusto para o desenvolvimento de aplicações em diferentes áreas, como *desktop*, web, e mobile, com suporte para sistemas de alta escalabilidade e performance. Uma das principais características do *.NET* é o seu ciclo de vida de desenvolvimento ágil, que permite a criação, manutenção e evolução de software de forma eficiente [24].

2) *C# (C-Sharp)*: É uma linguagem de programação desenvolvida pela Microsoft, lançada em 2000, que faz parte da plataforma *.NET*. Ela é amplamente utilizada para o desenvolvimento de aplicações *desktop*, web, móveis e até jogos, especialmente com o uso do framework Unity. C# oferece uma sintaxe simples e moderna, combinando conceitos de linguagens como C++ e Java, o que a torna altamente produtiva e robusta para o desenvolvimento de sistemas corporativos e aplicações de grande escala [25].

3) *ASP.NET Core*: É um framework de código aberto e multiplataforma para o desenvolvimento de aplicações web atuais e de alto desempenho. Segundo a documentação oficial da Microsoft, o ASP.NET Core permite a criação de aplicações web dinâmicas usando C# e .NET, com suporte à arquitetura MVC. Ele oferece uma série de recursos importantes, como middleware para processamento de solicitações HTTP (Hypertext Transfer Protocol/Protocolo de Transferência de Hipertexto), integração com bancos de dados, segurança robusta por meio de autenticação e autorização, e suporte a testes automatizados, entre outros [26].

4) *Entity Framework (EF)*: É uma tecnologia de mapeamento objeto-relacional (ORM/Object-Relational Mapping) desenvolvida pela Microsoft que facilita o acesso e a manipulação de dados em aplicativos .NET. Ele permite que os desenvolvedores interajam com bancos de dados utilizando objetos e modelos de dados, eliminando a necessidade de escrever complexas consultas SQL. O EF é integrado ao .NET e é amplamente utilizado para simplificar o processo de persistência de dados em aplicativos de diferentes tipos, como aplicações web e de *desktop*. Ele suporta o conceito de migrações, permitindo que as mudanças no modelo de dados sejam refletidas automaticamente no banco de dados [27].

5) *ASP.NET Identity*: É um sistema de gerenciamento de identidades utilizado em aplicações web com ASP.NET Core, que oferece funcionalidades robustas de autenticação e autorização. Ele permite que os desenvolvedores implementem facilmente recursos como login, registro de usuários, recuperação de senha e autenticação multifatorial. Além disso, o ASP.NET Identity integra-se ao Entity Framework Core, permitindo o armazenamento seguro de dados de usuários em bancos de dados relacionais [28].

6) *SQL Server*: Desenvolvido pela Microsoft, é um sistema de gerenciamento de banco de dados relacionais. Sua arquitetura é voltada para oferecer desempenho, escalabilidade e segurança no armazenamento e na manipulação de dados. Com suporte ao T-SQL (Transact-SQL), linguagem estendida baseada no SQL padrão, o SQL Server permite a criação de procedimentos armazenados, *triggers* e consultas complexas de forma otimizada. Outro destaque da ferramenta é sua integração com o ecossistema Microsoft, como .NET e Visual Studio [29].

7) *Bootstrap*: O Bootstrap [30] é um framework front-end de código aberto, amplamente utilizado para o desenvolvimento de aplicações web responsivas e *mobile-first*. Ele fornece uma biblioteca robusta de componentes de interface pré-estilizados, baseados em HTML, CSS e JavaScript, além de um sistema de grid flexível que agiliza a construção de layouts consistentes e adaptáveis a diferentes dispositivos.

III. TRABALHOS CORRELATOS

Nesta seção, serão apresentados os trabalhos que possuem envolvimento com gestão de treino de musculação utilizando tecnologias e que contribuíram para a realização deste projeto.

A. Sistema para Academias com Acompanhamento da Rotina de Treino

A aplicação web desenvolvida por Almeida apresenta uma interface intuitiva voltada à rotina dos profissionais de educação física, com funcionalidades que abrangem desde o cadastro de praticantes até o planejamento e acompanhamento de treinos. Ao integrar recursos como agendamento de avaliações físicas, registro de anamnese e acesso remoto por meio de dispositivos móveis, o sistema promove uma experiência mais fluida e organizada tanto para os profissionais quanto para os alunos. Esse tipo de abordagem evidencia o potencial da tecnologia da informação na promoção de práticas mais sistemáticas e personalizadas no contexto do condicionamento físico.

Os resultados obtidos no trabalho de Almeida demonstram que a aplicação proposta atendeu satisfatoriamente aos objetivos definidos, proporcionando maior controle e organização no acompanhamento da rotina de treino dos praticantes. A principal contribuição para o desenvolvimento deste trabalho está na estruturação modular da aplicação apresentada por Almeida, que serviu como base para a definição e organização dos módulos funcionais deste projeto, como cadastro de alunos, elaboração de treinos e acompanhamento do progresso.

B. Impactos do Uso de Aplicativos Fitness por Profissionais de Educação Física em academias de musculação de Belém do Pará

No trabalho de conclusão de curso de R. C. dos Santos[3], intitulado *Impactos do uso de aplicativos fitness por profissionais de educação física em academias de musculação de Belém do Pará*, foi realizada uma análise sobre a influência dos aplicativos fitness no cotidiano profissional de educadores físicos. A pesquisa envolveu 12 professores atuantes em academias de musculação, que responderam a questionários estruturados utilizando a escala de Likert de 5 pontos. Os resultados indicaram que os aplicativos digitais estão cada vez mais presentes no setor, sendo bem aceitos pelos profissionais. O estudo conclui que essas ferramentas tecnológicas colaboram significativamente com o trabalho dos educadores físicos, contribuindo para a modernização e eficiência das práticas nas academias.

Os dados coletados e analisados por R. C. dos Santos indicam que os aplicativos fitness são bem aceitos pelos profissionais da educação física, proporcionando melhorias significativas na organização e na personalização do treino dos alunos. Esses aplicativos têm sido apontados como ferramentas que agilizam o processo de planejamento e acompanhamento das atividades físicas, permitindo aos educadores físicos um controle mais preciso sobre o progresso de seus alunos e a capacidade de realizar ajustes de forma eficiente. Além disso, o estudo destaca que essas tecnologias contribuem para a modernização das práticas no setor, evidenciando como o uso de plataformas digitais pode transformar a dinâmica do trabalho em academias de musculação.

C. Aplicativo Mobile para Gerenciamento de Academias de Treinamento Funcional

O trabalho de da Silva [31], intitulado *Aplicativo Mobile para Gerenciamento de Academias de Treinamento Funcional*, apresenta o desenvolvimento de um aplicativo híbrido, utilizando a tecnologia React Native, voltado à gestão de estúdios e academias de treinamento funcional. O objetivo principal do projeto é facilitar o controle de horários, o agendamento de aulas e a comunicação entre gestores e alunos, atendendo especialmente academias de pequeno porte que carecem de soluções tecnológicas específicas.

Os resultados apresentados neste trabalho possuem relevância para o presente projeto, especialmente no que diz respeito à definição de requisitos funcionais e não funcionais, à adoção de metodologias ágeis de desenvolvimento e ao foco na experiência do usuário. Além disso, reforça a importância de soluções tecnológicas acessíveis e adaptadas às rotinas de academias e profissionais da área da educação física.

IV. METODOLOGIA

Para o desenvolvimento deste trabalho, optou-se pela metodologia ágil *Kanban*, devido à sua simplicidade, foco no fluxo contínuo de tarefas e facilidade na visualização do progresso do projeto [32].

Neste projeto, foi adotado um *backlog* detalhado com todas as atividades necessárias para a conclusão, priorizando tarefas de maior impacto e urgência, a fim de manter o foco nas entregas mais relevantes. Foram realizadas reuniões semanais com o orientador para acompanhamento do progresso, alinhamento de expectativas e resolução de impedimentos. Conforme Anderson [32], embora o Kanban não imponha cerimônias fixas, encontros regulares são recomendados para promover a comunicação, revisar o fluxo de trabalho e fomentar a melhoria contínua, adaptando o processo às necessidades da equipe e do projeto.

A. Aplicação Prática com Base no Diagrama de Casos de Uso

O sistema desenvolvido é uma aplicação web para gestão de treinos de musculação, voltada a personal trainers. A plataforma digitaliza o acompanhamento de alunos e permite a automatização de processos por meio de inteligência artificial (IA).

Conforme a Figura 2, A estrutura do sistema é:

- Criar Conta: Permite que novos usuários, tanto *Personal Trainers* quanto *Alunos*, se registrem na plataforma.
- Fazer Login: Garante o acesso seguro ao sistema, diferenciando os perfis de *Personal Trainer* e *Aluno*. Um cadastro inicial também foi implementado.
- Manter Alunos: Permite que o personal trainer cadastre, edite, consulte e remova alunos, associando-os ao seu perfil profissional.
- Registrar Anamnese: Funcionalidade essencial onde o personal armazena dados vitais do aluno, como histórico médico, hábitos, restrições físicas, nível de experiência e objetivos.

- Manter Exercícios: Módulo que funciona como um banco de dados de movimentos, onde o personal cadastra e gerencia os exercícios (com grupo muscular, equipamento, etc.) que serão usados nas fichas.
- Manter Ficha: Permite ao personal criar ou editar manualmente uma ficha de treino, selecionando exercícios do banco e definindo séries, repetições e divisões (ex: ABC).
- Gerar Treino IA: Caso de uso central onde o sistema, utilizando os dados da anamnese, consulta o modelo Gemini para sugerir uma ficha de treino personalizada. O personal pode revisar e aprovar esta sugestão.
- Visualizar Ficha: Permite que o aluno, através de seu próprio login, acesse e consulte a ficha de treino final que lhe foi atribuída pelo personal.

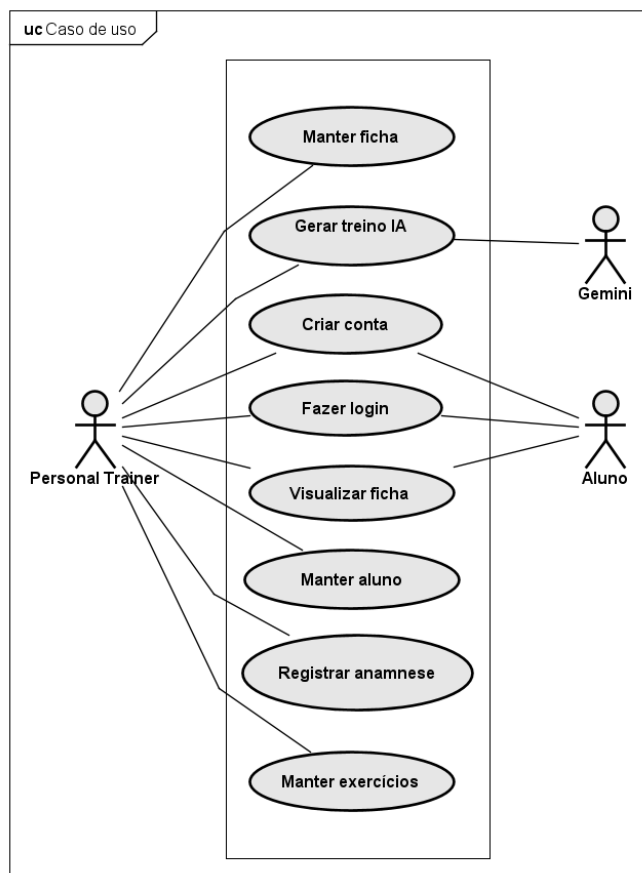


Figura 2. Diagrama de Caso de uso

B. Backlog do Projeto

Com base nas informações coletadas e no entendimento do sistema, o *backlog* foi gerado, estruturando as atividades necessárias para o desenvolvimento. Cada item foi definido com base nas necessidades levantadas na etapa de levantamento de requisitos e visou guiar o desenvolvimento de forma estruturada e incremental.

Os principais itens que compõem o *Product Backlog* são:

- Configurar ambiente de desenvolvimento: Preparar o ambiente técnico com as ferramentas, bibliotecas, banco de dados e sistema de versionamento necessários.
- Implementar autenticação com ASP.NET Identity: Criar o sistema de login e autenticação de usuários, garantindo segurança e integridade dos acessos.
- Manter usuários: Implementar funcionalidades *Create, Retrieve, Update, Delete (CRUD)* para os usuários, com campos como nome, e-mail e papel (Personal ou Aluno).
- Criar modelo e formulário de cadastro de alunos: Desenvolver o formulário específico para cadastro de alunos, incluindo campos como sexo, endereço e data de nascimento.
- Criar modelo e formulário para registrar dados de anamnese: Estruturar o registro da anamnese do aluno, contendo informações relevantes para o planejamento dos treinos, como histórico médico, hábitos alimentares e intensidade da atividade física.
- Estruturar modelo de ficha de treino: Criar a estrutura para representar fichas de treino, com objetivo, divisão do treino e associação com exercícios.
- Criar e editar ficha de treino manualmente: Permitir ao personal trainer criar fichas personalizadas para os alunos e editá-las conforme necessário.
- Criar modelo e formulário para cadastrar exercícios: Desenvolver o cadastro de exercícios com informações como nome, execução, grupo muscular e equipamento necessário.
- Implementar integração com API Gemini: Conectar o sistema à API Gemini para recursos avançados de recomendações automáticas com base em inteligência artificial.
- Implementar controle de acesso por perfil: Controlar o acesso às funcionalidades com base no papel do usuário, garantindo que apenas usuários autorizados realizem determinadas ações.

C. Projeto

Antes da implementação do código, foi realizada a fase de projeto da aplicação, onde os requisitos foram traduzidos em artefatos conceituais, arquiteturais e visuais.

1) *Modelagem de Dados*: O Diagrama de Entidade-Relacionamento (DER) lógico, apresentado na Figura 3, ilustra as tabelas principais geradas no banco de dados e seus relacionamentos.

Destacam-se as tabelas *AspNetUsers* (utilizada pelo ASP.NET Identity para gerenciar os papéis de Aluno e Personal), e as entidades centrais da aplicação: *Fichas*, *Treinos*, *ItemTreino*, *Exercicios* e *Anamneses*. O diagrama detalha as chaves estrangeiras que conectam o sistema, como o duplo relacionamento da *Fichas* com *AspNetUsers* (para *IdAluno* e *IdPersonal*) e a relação mestre-detalle entre *Ficha*, *Treino* e *ItemTreino*, que constitui o núcleo da ficha de treino.

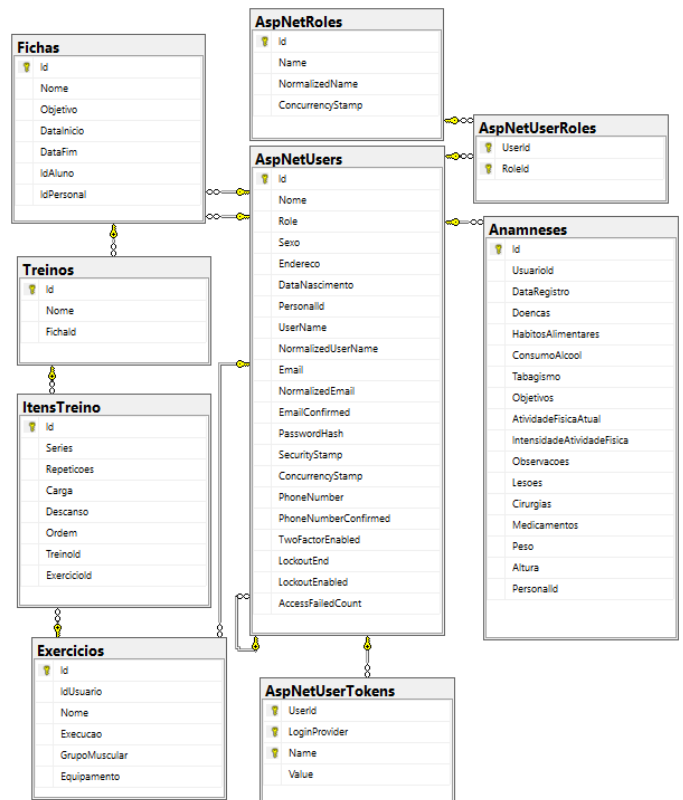


Figura 3. Diagrama de Entidade-Relacionamento (DER) do banco de dados.

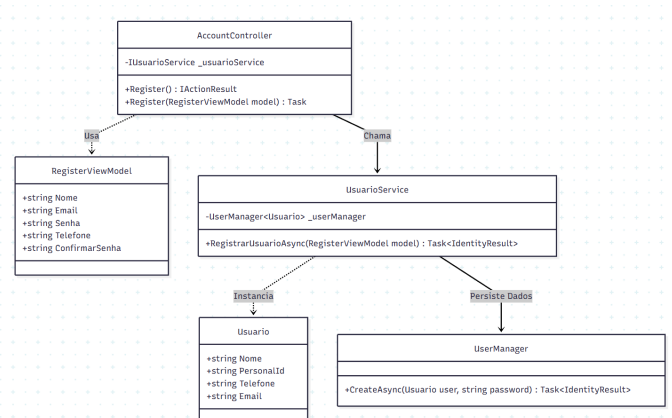


Figura 4. Diagrama de Classes da funcionalidade cadastro de Usuário.

2) *Projeto da Arquitetura*: Para exemplificar a arquitetura de software adotada, foi elaborado o Diagrama de Classes (Figura 4) focado na funcionalidade de cadastro de usuários. Este diagrama detalha o fluxo de dados iniciando pelo *RegisterViewModel*, que atua como um Objeto de Transferência de Dados entre a View e o *AccountController*. O controlador recebe a requisição e delega o processamento para o *UsuarioService*, que encapsula toda a lógica de negócios. Para a persistência, o serviço interage diretamente com a classe *UserManager* do ASP.NET Core Identity, abstraindo as operações de banco de dados.

3) Projeto de Interface e Experiência do Usuário (UI/UX):

Para definir o fluxo de navegação e a experiência do usuário, foram desenvolvidos protótipos de tela de baixa fidelidade. A Figura 5 detalha a tela de criação de ficha, que seria o "hub" central do sistema. Este protótipo foi essencial para validar a interação do personal com a seleção de alunos, a definição de objetivos e a integração com a sugestão de exercícios por Inteligência Artificial. Estes protótipos guiaram a construção do front-end com Bootstrap 5.

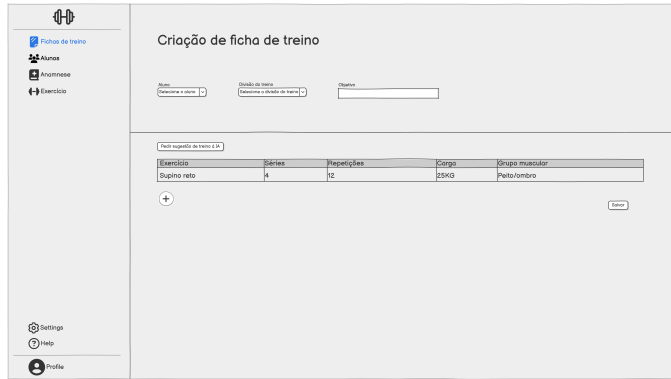


Figura 5. Protótipo da tela de visualização de alunos.

D. Desenvolvimento

A construção da aplicação seguiu os princípios da arquitetura Model-View-Controller (MVC) provida pelo framework ASP.NET Core, utilizando a linguagem C#.

A persistência de dados foi gerenciada pelo Entity Framework Core 8, utilizando a abordagem *Code-First*. Esta metodologia permitiu que o esquema do banco de dados SQL Server fosse gerado e versionado diretamente a partir das classes de modelo C#.

A segurança e o gerenciamento de usuários foram implementados utilizando o ASP.NET Core Identity. A adoção desta ferramenta contribui diretamente para a conformidade com a LGPD (Lei nº 13.709/2018) [11], pois implementa nativamente medidas técnicas de proteção exigidas pela legislação. Tais mecanismos mitigam os riscos de acessos não autorizados e vazamento de informações sensíveis, alinhando o sistema às diretrizes de segurança digital voltadas ao público.

No front-end, a interface foi construída com Bootstrap 5 (template SB Admin 2 [33]), e enriquecida com bibliotecas JavaScript como SweetAlert2 [34] para diálogos modais e DataTables [35] para a interatividade das tabelas de dados.

1) *Padrão de Desenvolvimento*: Os controllers são responsáveis apenas por receber requisições HTTP e orquestrar o fluxo, delegando toda a lógica de negócios para uma camada de serviço dedicada (ex: IFichasService). O método de criação de uma nova Ficha de Treino exemplifica este padrão. O Listing 1 mostra o método no FichasController, que valida o modelo e invoca o serviço.

```
1 [Authorize(Roles = "Personal,Admin")]
2 [HttpPost]
3 [ValidateAntiForgeryToken]
```

```
4 public async Task<IActionResult> Create(
    FichaCreateViewModel viewModel)
5 {
6     if (ModelState.IsValid)
7     {
8         // 1. Delega a logica para o servico
9         var novoId = await _fichasService.
            CreateFichaAsync(viewModel, User);
10        return RedirectToAction(nameof(Edit),
            new { id = novoId });
11    }
12
13    // 2. Se falhar, busca dados de suporte (
14    // tambem do servico)
15    viewModel.Alunos = await _fichasService.
16    GetAlunosSelectListAsync(
17    viewModel.IdAluno
18    );
19    return View(viewModel);
20 }
```

Listing 1. Método Create (POST) no FichasController.

A lógica de negócios como obter o ID do personal logado, mapear a ViewModel para a entidade e salvar no banco reside inteiramente na camada de serviço, como visto no Listing 2.

```
1 public async Task<string> CreateFichaAsync(
2     FichaCreateViewModel viewModel,
3     ClaimsPrincipal user)
4 {
5     // 1. Logica de negocio: Obter o ID do
6     // usuario logado
7     var personalId = _userManager.GetUserId(
8     user);
9
10    // 2. Mapeamento de ViewModel para Modelo
11    // de Dominio
12    var ficha = new Ficha
13    {
14        Id = Guid.NewGuid().ToString(),
15        Nome = viewModel.Nome,
16        IdAluno = viewModel.IdAluno,
17        IdPersonal = personalId,
18        DataInicio = viewModel.DataInicio,
19        DataFim = viewModel.DataFim,
20        Objetivo = viewModel.Objetivo
21    };
22
23    // 3. Logica de persistencia de dados
24    _context.Add(ficha);
25    await _context.SaveChangesAsync();
26
27    return ficha.Id;
28 }
```

Listing 2. Implementação do CreateFichaAsync no FichasService.

Esta interação entre controlador e serviço foi a base para a implementação de todos os outros módulos de gerenciamento, como o cadastro de Exercícios e Anamneses.

2) *Integração com Inteligência Artificial (Gemini)*: O sistema possui a funcionalidade de sugestão de treinos, que utiliza a API do modelo de linguagem Gemini do Google. Esta ferramenta atua como um assistente para o Personal Trainer. A implementação também segue o padrão Controller-Service.

O fluxo de execução é iniciado quando o usuário clica no botão "Sugerir com IA" na tela de edição, onde um modal (SweetAlert2) coleta instruções de texto opcionais. A requisição é submetida para a action GerarTreinosIA no FichasController (Listing 3), que por sua vez delega imediatamente a operação completa para o método GerarTreinosComIAAsync na camada de serviço.

O FichasService (Listing 4) então orquestra toda a operação: ele busca a Anamnese do aluno e a lista de Exercícios disponíveis no banco de dados. Em seguida, invoca a classe que envia requisição para a API do Gemini, que é responsável pela Engenharia de Prompt (construindo as instruções para a IA) e pela chamada de API. O JSON retornado pelo Gemini é desserializado, e o serviço aplica a lógica "Remove-e-Adiciona" para substituir os treinos antigos da ficha pelos novos sugeridos. Ao final, o serviço retorna uma tupla (bool Success, string Message) para o controller, que exibe uma notificação de sucesso ou erro para o usuário.

```
1 [HttpPost]
2 [ValidateAntiForgeryToken]
3 public async Task<IActionResult>
4 GerarTreinosIA(
5     string fichaId, string instrucoes)
6 {
7     if (string.IsNullOrEmpty(fichaId))
8     {
9         return NotFound();
10    }
11
12    // 1. Delega toda a operacao complexa para
13    // o servico
14    var (success, message) = await
15        _fichasService.GerarTreinosComIAAsync(
16            fichaId, instrucoes, User
17        );
18
19    // 2. Retorna o feedback para a View
20    if (success) { TempData["Success"] =
21        message; }
22    else { TempData["Error"] = message; }
23
24    return RedirectToAction(nameof(Edit), new
25        { id = fichaId });
26 }
```

Listing 3. Método GerarTreinosIA no FichasController.

```
1 public async Task<(bool Success, string
2     Message)> GerarTreinosComIAAsync(
3     string fichaId, string instrucoes,
4     ClaimsPrincipal user)
5 {
6     var personalId = BuscarPersonal(user);
7     var anamnese = BuscarAnamnese(ficha.
8         IdAluno);
9
10    if (anamnese == null)
11    {
12        return (false, "Aluno não possui
13            anamnese cadastrada.");
14    }
15 }
```

```
var exercicios = BuscarExercicios(
    personalId);

var fichaSugeridaJson = await
    _geminiService.
    GerarFichaDeTreinoAsync(anamnese,
        exercicios, instrucoes);

var fichaVmSugerida =
    DesserializarFicha(
        fichaSugeridaJson)

// Logica "Remove-e-Adiciona"
if (ficha.Treinos.Any())
{
    _context.Treinos.RemoveRange(ficha.
        Treinos);
}

foreach (var treinoVm in
    fichaVmSugerida.Treinos)
{
    // (Mapeamento de ViewModel para
    // Modelo omitido por brevidade)
    var novoTreino = new Treino { ...
    };
    ficha.Treinos.Add(novoTreino);
}
await _context.SaveChangesAsync();
return (true, "Novas divisões geradas
    com sucesso!");
}
```

Listing 4. Trecho do método GerarTreinosComIAAsync no FichasService.

V. RESULTADOS

Esta seção apresenta os resultados práticos obtidos com a implementação do sistema, construído sobre a arquitetura MVC e os padrões de projeto detalhados na Metodologia.

A. Dashboard de Gestão do Personal Trainer

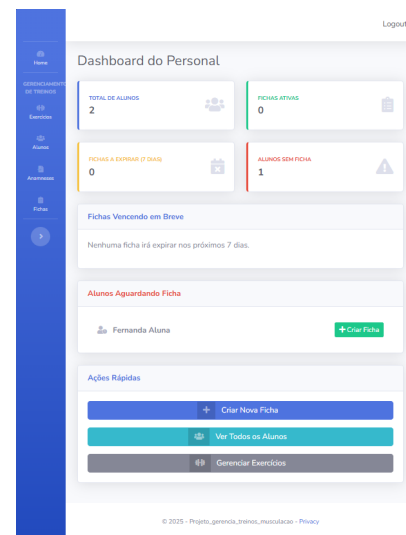


Figura 6. Dashboard principal do Personal Trainer.

O primeiro objetivo do sistema era "facilitar o gerenciamento e a organização dos treinos". Para o Personal Trainer, isso foi materializado através de um Dashboard dinâmico, apresentado na Figura 6. Esta tela atua como uma central de controle que aplica conceitos de gerenciamento visual de tarefas.

Ao invés de simplesmente listar dados, o dashboard fornece indicadores de ação, como "Alunos Aguardando Ficha" e "Fichas Vencendo em Breve". Estas listas informam ao profissional exatamente onde sua atenção é necessária, permitindo uma gestão proativa e o atendimento direto ao objetivo de organização do fluxo de trabalho.

B. Montagem de Ficha e Integração com IA

O núcleo funcional do sistema é a tela de montagem de fichas (Figura 7), que cumpre dois objetivos específicos: "garantir que o treino possa ser mudado"(flexibilidade) e "implementar a utilização de IA".

Como detalhado na Metodologia (Listing 3), esta interface permite ao personal adicionar divisões de treino manualmente ou acionar o diferencial do sistema: a sugestão por Inteligência Artificial. A Figura 7 demonstra o modal de interação (construído com SweetAlert2) no momento em que o personal trainer invoca a IA. O sistema coleta instruções adicionais e envia o contexto do aluno (anamnese) e a lista de exercícios disponíveis para o GeminiService.

Esta abordagem cumpre o objetivo de usar a IA como uma ferramenta de assistência, que respeita o catálogo de exercícios do personal e exige sua validação final, ao mesmo tempo que oferece a flexibilidade para edição manual.

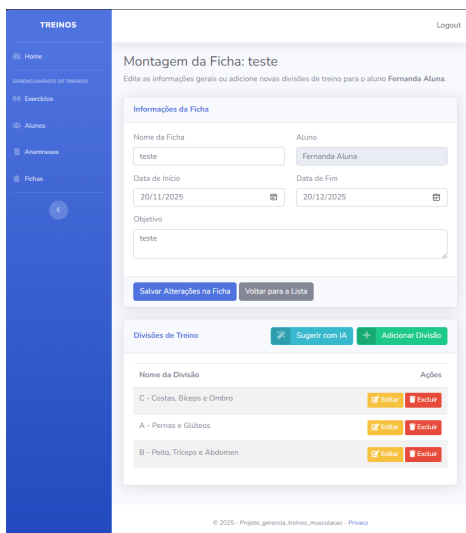


Figura 7. Tela de montagem de ficha com modal.

C. Acesso do Aluno e Responsividade

Finalmente, para atender ao objetivo de "possibilitar fácil acesso à ficha de treino", a aplicação foi desenvolvida utilizando Bootstrap 5 para garantir um design responsivo, adaptando-se a dispositivos móveis, que é o principal meio de acesso do aluno dentro da academia.

A Figura 8 demonstra a tela de detalhes da ficha de treino (o "Dashboard do Aluno") renderizada em um dispositivo móvel. A interface utiliza um padrão de "accordion", permitindo ao aluno expandir e focar em cada divisão de treino (ex: "Treino A"), onde a tabela de exercícios é apresentada com os dados organizados verticalmente para a leitura em telas de formato retrato (portrait).

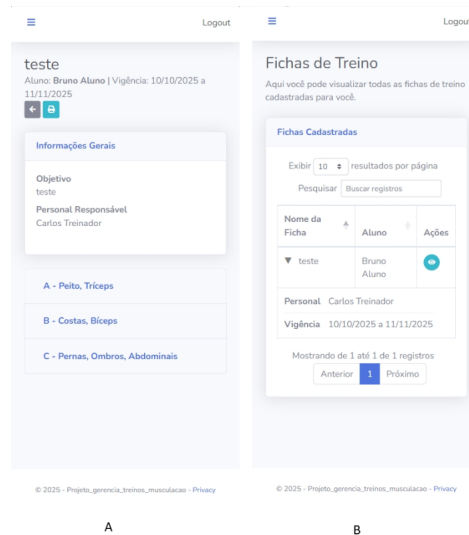


Figura 8. Tela de detalhes da ficha(A) e tela de index das fichas(B) mobile.

Para garantir a usabilidade em todo o sistema, as telas de listagem (como Anamneses, Exercícios e Fichas) utilizaram a biblioteca DataTables com a extensão *Responsive*. Foi implementada a funcionalidade de *responsive priority* (prioridade de coluna) e *child rows* (linhas filhas). Esta técnica oculta colunas de menor prioridade em telas menores e as move para uma "linha filha" expansível, conforme mostrado na figura 8. Esta abordagem elimina a necessidade de rolagem horizontal, um conhecido ponto de atrito na navegação móvel. Ao exibir apenas os dados essenciais por padrão, a interface reduz a carga cognitiva do usuário e otimiza a apresentação da informação para o dispositivo, alinhando-se aos princípios de design eficiente e boa usabilidade.

Desta forma, os resultados práticos da aplicação funcional demonstram que todos os objetivos específicos propostos neste trabalho foram atingidos com sucesso.

VI. CONCLUSÃO

Este artigo apresentou o desenvolvimento de uma aplicação web voltada para a gestão de treinos de musculação, destinada a personal trainers que buscam otimizar a organização e o acompanhamento de seus alunos.

Os objetivos específicos foram alcançados conforme o planejado, com a implementação bem-sucedida da centralização do gerenciamento de alunos, anamneses e a criação de fichas de treino. O sistema também garantiu a correta segregação de dados entre os diferentes perfis de usuário (Aluno, Personal e Admin), assegurando que cada um acesse apenas as informações relevantes ao seu contexto de uso.

A integração da API do Google Gemini permitiu ao personal trainer gerar sugestões de treino de forma assistida, baseando-se na anamnese do aluno e no catálogo de exercícios do profissional. Esta funcionalidade otimiza o tempo de prescrição e auxilia na criação de rotinas personalizadas. Como resultado, o atendimento se torna mais técnico e eficiente, permitindo ao personal dedicar mais tempo ao acompanhamento direto dos alunos.

Visando a evolução da aplicação, foi observado que, para trabalhos futuros, o desenvolvimento de um aplicativo móvel nativo para o aluno agregaria de forma significativa. Este possibilitaria a implementação de um "diário de treino" para o registro em tempo real de cargas e repetições. Isso permitiria, futuramente, a geração automática de relatórios de progressão de carga e o aprimoramento da IA para sugerir evoluções de treino, possibilitando um gerenciamento de desempenho mais completo e eficiente.

REFERÊNCIAS

- [1] G. Granchi and B. Paludeti. (2021) Exercício é remédio: entenda como atividade física previne e trata doenças. [Online]. Available: <https://www.uol.com.br/vivabem/reportagens-especiais/exercicio-e-remedio-entenda-como-atividade-fisica-previne-e-trata-doencas->
- [2] G. Ojea. (2024) Número de personal trainers em atividade cresce 32,76% no Brasil. [Online]. Available: <https://revistaproativa.com.br/numero-de-personal-trainers-em-atividade-cresce-3276-no-brasil>
- [3] R. C. dos Santos, "Impactos do uso de aplicativos fitness por profissionais de educação física em academias de musculação de Belém do Pará," Trabalho de Conclusão de Curso, Universidade Federal do Pará, 2022. [Online]. Available: https://www.bdm.ufpa.br/bitstream/prefix/5084/1/TCC_ImpactosUsoAplicativos.pdf
- [4] D. P. Guedes and J. E. R. Guedes, "Atividade física, aptidão física e saúde," *Revista Brasileira de Atividade Física & Saúde*. [Online]. Available: <https://rbafs.org.br/RBAFS/article/view/451/495>
- [5] M. G. A. Campos Filho, Ed., *Musculação: bases teóricas*. Rio de Janeiro: Freitas Bastos, 2024.
- [6] J. C. et al. (2019) Orientações para avaliação e prescrição de exercícios físicos direcionados à saúde. [Online]. Available: <https://www.crefsp.gov.br/storage/app/arquivos/7e02a5c44298e22ad31dce23f52948b9.pdf>
- [7] C. B. Bossle, "O personal trainer e o cuidado de si: Uma perspectiva de mediação profissional," *Movimento*, vol. 14, no. 1, p. 187198, abr. 2008. [Online]. Available: <https://seer.ufrgs.br/index.php/Movimento/article/view/3764>
- [8] Brasil. Lei nº 9.696, de 12 de setembro de 1998. [Online]. Available: https://www.planalto.gov.br/ccivil_03/leis/19696.htm
- [9] K. C. Laudon and J. P. Laudon, *Sistemas de informação gerenciais: administrando a empresa digital*, 17th ed. São Paulo, SP: Bookman, 2023.
- [10] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, 7th ed. Cengage Learning, 2021.
- [11] "Lei geral de proteção de dados pessoais (LGPD) - Lei nº 13.709/2018," Presidência da República, 2018, disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/13709.htm. Acesso em: 10 abr. 2025.
- [12] Microsoft, "ASP.NET Core MVC," 2025. [Online]. Available: <https://learn.microsoft.com/pt-br/aspnet/core/mvc/>
- [13] J. J. Garrett, *The Elements of User Experience: User-Centered Design for the Web and Beyond*, 2nd ed. Berkeley: New Riders, 2011.
- [14] B. Frain, *Responsive Web Design with HTML5 and CSS3*, 2nd ed. Birmingham, UK: Packt Publishing, 2019.
- [15] International Organization for Standardization, "ISO 9241 ergonomics of human-system interaction," 2019.
- [16] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, 1993.
- [17] Atlassian, "O que é Kanban?" <https://www.atlassian.com/pt-br/agile/kanban>, 2024.
- [18] ClickUp, "30+ best Kanban board examples to streamline your workflow," <https://clickup.com/blog/kanban-board-examples/>, 2024.
- [19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: <https://arxiv.org/abs/2005.14165>
- [22] OpenAI, "GPT-4 technical report," <https://openai.com/research/gpt-4>, 2023.
- [23] Google DeepMind, "Gemini 1.5 technical report," <https://deepmind.google/technologies/gemini/>, 2024.
- [24] Microsoft, ".NET," 2023. [Online]. Available: <https://dotnet.microsoft.com/>
- [25] Microsoft, "C# programming guide," <https://docs.microsoft.com/en-us/dotnet/csharp/>, 2020.
- [26] Microsoft. (2024) Visão geral do ASP.NET Core. [Online]. Available: <https://learn.microsoft.com/pt-br/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-9.0>
- [27] Microsoft, "Entity framework," 2025. [Online]. Available: <https://learn.microsoft.com/pt-br/ef/>
- [28] Microsoft, "ASP.NET Core Identity," 2023. [Online]. Available: <https://learn.microsoft.com/pt-br/aspnet/core/security/authentication/identity?view=aspnetcore-9.0&tabs=visual-studio>
- [29] Microsoft, "SQL Server documentation," 2024, available at: <https://learn.microsoft.com/en-us/sql/sql-server/>
- [30] The Bootstrap Authors, "Bootstrap: The most popular HTML, CSS, and JS library," <https://getbootstrap.com/>, 2025, acesso em: 20 nov. 2025.
- [31] G. M. da Silva. (2021) Aplicativo mobile para gerenciamento de academias de treinamento funcional. [Online]. Available: <https://repositorio-api.animaeducacao.com.br/server/api/core/bitstreams/6b903906-4504-4851-b437-f2a6d54c5d1e/content>
- [32] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, WA: Blue Hole Press, 2010.
- [33] Start Bootstrap, "SB Admin 2: Free Bootstrap admin theme," <https://startbootstrap.com/theme/sb-admin-2>, 2025, acesso em: 20 nov. 2025.
- [34] SweetAlert2, "SweetAlert2: A beautiful, responsive, customizable replacement for JavaScript's popup boxes," <https://sweetalert2.github.io/>, 2024, acesso em: 20 nov. 2024.
- [35] SpryMedia Ltd, "Datatables: Table plug-in for jQuery," <https://datatables.net/>, 2024, acesso em: 20 nov. 2024.