CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE FRANCA

“Dr. THOMAZ NOVELINO”

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GUILHERME DE OLIVEIRA CANTO CARDOSO

LEONARDO SANTOS PARRA

SISTEMA DE GERENCIAMENTO DE ACADEMIA

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Técnologo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2025

SISTEMA DE GERENCIAMENTO DE ACADEMIA

Leonardo Santos Parra[1]

Guilherme de Oliveira Canto Cardoso[2]

Carlos Alberto Lucas[3]

Resumo

Este trabalho apresenta o desenvolvimento de um sistema voltado para o gerenciamento de academias, com o objetivo de organizar processos internos e facilitar o dia a dia dos funcionários. A aplicação permite cadastrar alunos, controlar matrículas, gerenciar fichas de treino e registrar pagamentos de forma simples e eficiente. A proposta surge como uma alternativa aos métodos manuais ainda usados em muitas academias de pequeno porte, buscando reduzir erros, ganhar tempo e melhorar o atendimento. A pesquisa também discute os benefícios da digitalização na gestão de pequenos negócios e reforça a importância de ferramentas acessíveis e funcionais para acompanhar a evolução do setor fitness. Com isso, o projeto pretende contribuir para uma administração mais prática e moderna dentro das academias.

Palavras-chave: academia. gestão. sistema digital. matrícula. treino. pagamento.

Abstract

This paper presents the development of a system aimed at gym management, with the goal of organizing internal processes and simplifying the staff's daily routine. The application allows for student registration, membership control, workout plan management, and payment tracking in a practical and efficient way. The proposal small gyms, seeking to reduce errors, save time, and improve service. The research

[1] Graduando em Análise e Desenvolvimento de sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: leonardoparra1k@gmail.com

[2] Graduando em Análise e Desenvolvimento de sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: guilherme.cardoso8@fatec.sp.gov.br

[3] Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: carlos.lucas@fatec.sp.gov.br

also discusses the benefits of digitalization in the management of small businesses and highlights the importance of accessible and functional tools to keep up with the evolution of the fitness sector. With this approach, the project aims to contribute to a more practical and modern gym administration.

Keywords: gym. management. digital system. membership. workout. Payment.

1 Introdução

Nos últimos anos, a busca por um estilo de vida mais saudável se intensificou, especialmente após a pandemia da COVID-19, que despertou uma preocupação maior com o bem-estar físico e mental da população. Nesse cenário, academias de musculação passaram a ter uma procura ainda maior, tanto por parte de iniciantes quanto de frequentadores antigos que passaram a dar mais valor à prática regular de exercícios físicos. Com esse aumento da demanda, ficou evidente a necessidade de modernização dos processos de gestão desses espaços, principalmente nas academias de pequeno porte que ainda utilizam métodos manuais, como o uso de cadernos para controle de matrículas, pagamentos e treinos.

Este trabalho tem como tema o desenvolvimento de um software de gestão voltado para academias de musculação de pequeno porte. A questão-problema que orienta o projeto é: como informatizar e otimizar a gestão de uma academia que atualmente utiliza métodos manuais, melhorando o controle das informações e a qualidade do atendimento aos alunos? A hipótese é que, por meio da implementação de um sistema informatizado simples, porém funcional, será possível centralizar dados importantes, reduzir falhas humanas, automatizar processos rotineiros e tornar a administração mais eficiente e segura.

O objetivo geral deste trabalho é desenvolver uma ferramenta digital personalizada para atender às necessidades específicas da academia analisada. Como objetivos específicos, destacam-se: permitir o cadastro e controle de alunos, gerenciar fichas de treino, acompanhar o status das matrículas e registrar pagamentos. A justificativa para a realização deste projeto se baseia na observação de que muitos estabelecimentos pequenos ainda enfrentam dificuldades com a gestão, o que compromete o crescimento do negócio e a experiência dos clientes. Um sistema simples e acessível pode representar um avanço significativo, trazendo organização, agilidade e profissionalismo ao ambiente.

A relevância deste trabalho está na sua aplicação prática e imediata. Além de resolver uma dor real vivida pelo proprietário da academia envolvida no projeto, a solução proposta pode ser adaptada e replicada em outros estabelecimentos com características semelhantes. Trata-se de uma contribuição concreta para a

comunidade, especialmente para pequenos empreendedores do setor fitness que ainda não têm acesso a soluções tecnológicas sofisticadas.

A metodologia adotada envolveu inicialmente a observação do funcionamento da academia, seguida por uma visita técnica ao local. Com base nesse contato direto, foi elaborado um questionário aplicado ao proprietário, o que permitiu uma compreensão mais profunda das necessidades específicas do negócio. A partir dessas informações, o software foi modelado com base em práticas de engenharia de requisitos e princípios de usabilidade.

1.1 Termo da Abertura do Projeto (TAP)

O TAP (Termo de Abertura do Projeto) é um documento fundamental que marca oficialmente o início de um projeto. Ele tem como função principal definir os aspectos mais relevantes da proposta, como o objetivo do projeto, seu escopo, as premissas, restrições, prazos, custos e os principais envolvidos (stakeholders). Sua elaboração ocorre nas etapas iniciais do planejamento e serve como uma referência clara e compartilhada entre todos os participantes.

Segundo o Project Management Institute (PMI), o TAP funciona como uma ferramenta de alinhamento estratégico, garantindo que todos os envolvidos tenham compreensão mútua das entregas, responsabilidades e critérios de sucesso do projeto. Isso contribui diretamente para reduzir riscos associados a falhas de comunicação e interpretações equivocadas.

Ainda conforme Kerzner (2017), o TAP é essencial para a formalização do projeto, pois estabelece de maneira oficial que ele foi aprovado, autorizado e que os recursos necessários foram disponibilizados. Ele também funciona como base para o planejamento detalhado das atividades e etapas subsequentes.

Tabela 1 - TAP

Seção	Detalhes

Identificação do projeto	Desenvolvimento de um sistema de gestão para academias de musculação
Justificativa do projeto	Academias de pequeno porte frequentemente realizam seus processos administrativos de forma manual, utilizando cadernos ou planilhas simples. Esse modelo é suscetível a erros, atrasos e perda de informações importantes. Diante disso, torna-se essencial a criação de um sistema informatizado que centralize o controle de alunos, matrículas, treinos e pagamentos, otimizando a gestão e melhorando a experiência tanto para os administradores quanto para os alunos.
Objetivos e metas	● Automatizar o controle de matrículas, treinos e pagamentos. ● Reduzir o uso de registros físicos. ● Melhorar a qualidade do atendimento ao aluno. ● Facilitar a tomada de decisões com base em dados organizados.
Descrição do projeto	Será desenvolvido um sistema web voltado para a gestão de academias, com funcionalidades como cadastro de alunos, criação de fichas de treino, controle de matrículas e pagamentos, envio de notificações e geração de relatórios.
Premissas	● O sistema será desenvolvido conforme os requisitos identificados. ● Deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD).

	● Serão utilizadas tecnologias gratuitas ou de código aberto sempre que possível.
Restrições	● O prazo de entrega vai até o final do segundo semestre de 2025. ● o escopo foca no desenvolvimento de uma ferramenta web.
Escopo	● Cadastro e gerenciamento de alunos. ● Criação e edição de fichas de treino. ● Controle de matrículas (status, duração e planos). ● Registo de pagamentos manuais.
Resultados esperados	● Maior organização e controle dos dados da academia. ● Redução de falhas humanas e retrabalhos.
Não-escopo	● Integrações com maquininhas de cartão ou bancos. ● Controle de estoque ou vendas de produtos. ● Funcionalidades de marketing ou redes sociais.
Prazos Previstos	Início: Agosto de 2024 Término: Novembro de 2025
Custo Previsto	O projeto será desenvolvido utilizando recursos internos e tecnologias open-source. O único custo estimado é relacionado à hospedagem e infraestrutura mínima necessária para o funcionamento do sistema.

Fonte: Autores

2 Viabilidade do projeto

Esta seção apresenta as ações estratégicas que garantem a viabilidade do projeto, as quais serão analisadas por meio da aplicação do Business Model Canvas (BMC). Com essa ferramenta, será possível identificar os principais componentes do modelo de negócio.

2.1 Canvas de Negócio (Business Model Canvas - BMC)

O Business Model Canvas (BMC) é uma ferramenta visual desenvolvida por Alexander Osterwalder que permite descrever, analisar e projetar modelos de negócio de forma simplificada e estruturada. Ele é composto por nove blocos que representam as áreas fundamentais de uma empresa: segmentos de clientes, proposta de valor, canais, relacionamento com clientes, fontes de receita, recursos principais, atividades principais, parcerias-chave e estrutura de custos. O objetivo do Canvas é proporcionar uma visão clara e compartilhada do funcionamento do negócio, facilitando o entendimento e a comunicação entre os envolvidos (OSTERWALDER; PIGNEUR, 2010).

Figura 1 – Canvas

<table>
<tr><th>PARCERIAS CHAVE</th><th>ATIVIDADES CHAVE</th><th>PROPOSTA DE VALOR</th><th>RELAÇÕES COM CLIENTE</th><th>SEGMENTOS DE MERCADO</th></tr>
<tr><td rowspan="2">• Colaboração com profissionais de educação física para definição de fichas de treino.
• Apoio de professores e gestores da academia para validação dos processos.</td><td>• Desenvolvimento e manutenção do sistema web da academia.
• Controle de matrículas, treinos, pagamentos.</td><td rowspan="2">• Sistema digital para gerenciamento de academias de forma prática e centralizada.
• Controle de fichas de treino e matrículas de forma eficiente.
• Relatórios de alunos ativos, inativos e inadimplentes para tomada de decisão.</td><td>• Comunicação direta com os gestores da academia para ajustes no sistema.
• Implementação de funcionalidades com base nas necessidades reais do cliente.</td><td rowspan="2">• Academias de pequeno a médio porte que precisam digitalizar seu controle de gestão.
• Profissionais de educação física que desejam organizar fichas de treino e acompanhamento de alunos.</td></tr>
<tr><td>RECURSOS CHAVE
• Servidor e banco de dados PostgreSQL.
• Funcionário devidamente treinados.</td><td>CANAIS
• Sistema web acessível por navegadores em computadores.</td></tr>
<tr><td colspan="3">ESTRUTURA DE CUSTOS
• Desenvolvimento e manutenção do sistema.
• Custos com servidor e hospedagem de banco de dados.</td><td colspan="2">FONTES DE RENDA
• Possibilidade de cobrança por licença de uso do sistema para outras academias.</td></tr>
</table>

Fonte: Autores

O modelo é dividido em nove blocos fundamentais, a seguir a explicação de cada bloco.

1 – Segmentos de clientes: identifica os diferentes grupos de pessoas ou organizações que a empresa pretende atingir e atender. Compreender bem quem são os clientes é essencial para criar produtos e serviços relevantes e personalizados.

2 - Propostas de Valor: define o que torna a empresa única para seus clientes. Refere-se aos benefícios e diferenciais que a empresa oferece, solucionando problemas ou satisfazendo necessidades de maneira específica. Pode envolver inovação, desempenho, personalização ou design.

3 - Canais: representam os meios pelos quais a empresa se comunica com os clientes e entrega sua proposta de valor. Isso inclui desde a divulgação até a entrega do produto ou serviço. Os canais influenciam a experiência do cliente e a eficiência das operações.

4 **-** Relacionamento com Clientes: descreve como a empresa interage com seus diferentes segmentos de clientes. Pode variar entre atendimento personalizado, autoatendimento, comunidades ou serviços automatizados. Esse relacionamento impacta diretamente na fidelização e na satisfação do público.

5 - Fontes de Receita: explicam como a empresa obtém renda a partir da oferta de valor aos seus clientes. Podem vir de vendas diretas, mensalidades, licenças, aluguéis ou outras formas de pagamento. Esse bloco mostra quais atividades são financeiramente mais relevantes.

6 - Recursos-Chave: são os principais ativos necessários para que o modelo de negócio funcione. Incluem recursos físicos, intelectuais, humanos e financeiros. Eles sustentam a entrega da proposta de valor, os canais, o relacionamento com clientes e a geração de receita.

7 - Atividades-Chave: abrangem as ações fundamentais que a empresa deve realizar para operar com sucesso. São atividades diretamente ligadas à criação, entrega e manutenção da proposta de valor, como produção, marketing, logística ou desenvolvimento.

8 - Parcerias-Chave: envolvem os relacionamentos estratégicos com terceiros, fornecedores, aliados ou parceiros, que ajudam a otimizar o modelo de negócio. Essas parcerias podem trazer recursos, reduzir riscos ou ampliar a atuação da empresa no mercado.

9 - Estrutura de Custos: detalha os principais custos envolvidos para manter o funcionamento do negócio. Refere-se aos gastos com recursos, atividades e parcerias-chave. Ter clareza sobre esses custos permite que a empresa otimize seus processos e aumente a rentabilidade.

2.1 SWOT

A Análise SWOT é uma ferramenta essencial no campo do planejamento estratégico empresarial. Desenvolvida por Albert Humphrey durante um projeto de pesquisa na Universidade de Stanford, nas décadas de 1960 e 1970, essa metodologia busca oferecer uma visão completa do ambiente interno e externo de uma organização.

No contexto da Administração de Empresas, a Análise SWOT é amplamente utilizada como um instrumento para o planejamento estratégico. Seu objetivo é identificar e organizar informações relevantes que compõem o ambiente interno (forças e fraquezas) e o ambiente externo (oportunidades e ameaças) de uma empresa. Segundo Maximiano (2012), essa análise ambiental serve como base para o planejamento estratégico, sendo aplicável a qualquer tipo de organização, devido à sua simplicidade e flexibilidade.

A seguir, na Imagem 2, é apresentada a matriz SWOT referente ao projeto da academia.

Figura 2 - SWOT

Pontos fortes	**Pontos fracos**
Catraca com biometria. Fichas de treino personalizadas. Fúncionarios prestativos.	Falta de gerenciamento financeiro, fluxo de caixa e fechamento mensal. Um caderno é usado para gerenciar o cadastro e pagamento dos clientes. Avisos importantes comunicados de forma verbal. Não há nenhum tipo de programa fidelização. Falta de segurança nos dados. Não há controle de clientes inativos, ou que não vão regularmente. Não há um canal para receber feedback. Não há marketing.
Oportunidades	**Ameaças**
Pouca concorrência. Atualmente está havendo um aumento no interesse por academias devido à um projeto no Youtube.	Flutuação no preço dos equipamentos. Perda de interesse dos clientes durante horário de pico Possibilidade de perda financeira devido à falta de controle. Risco de danificar equipamentos.

Fonte: Autores

Sobre as fraquezas:

- Falta de gerenciamento financeiro, fluxo de caixa e fechamento mensal: A falta de gerenciamento financeiro, incluindo controle de fluxo de caixa e fechamento mensal, representa um desafio crítico para a estabilidade da academia.

- Um caderno é usado para gerenciar o cadastro e pagamento dos clientes: A dependência de um caderno para gerenciar pagamentos e cadastros dos clientes evidencia uma fragilidade no sistema de registro, resultando em inconsistências e falta de eficiência no controle administrativo.

- Avisos importantes comunicados de forma verbal: A comunicação de avisos importantes de forma verbal pode ser uma fraqueza, pois há o risco de perda de informações essenciais, falta de registro formal e possíveis mal-entendidos entre a equipe ou clientes.

- Não há nenhum tipo de programa fidelização: A ausência de um programa de fidelização representa uma fraqueza, pois a academia pode perder oportunidades de reter clientes, incentivar a frequência e construir relacionamentos mais sólidos, impactando potencialmente a fidelidade e a satisfação dos membros.

- Falta de segurança nos dados: A falta de segurança nos dados é uma vulnerabilidade significativa, expondo a academia ao risco de perda, acesso não autorizado ou uso indevido de informações confidenciais, comprometendo a confiança dos clientes e a conformidade com normas de privacidade.

- Não há controle de clientes inativos, ou que não vão regularmente: A falta de controle sobre clientes inativos ou irregulares é uma fraqueza que pode resultar na perda de oportunidades de reengajamento e na subutilização dos recursos da academia.

- Não há canal para receber feedback: A ausência de um canal para receber feedback representa uma lacuna na comunicação com os clientes, dificultando a compreensão das necessidades e expectativas.

- Não há marketing: A falta de estratégias de marketing é uma fraqueza, pois a academia pode perder oportunidades de atrair novos clientes e reter os existentes.

2.2 5W2H

O 5W2H é uma ferramenta de gestão e planejamento que tem como objetivo detalhar e organizar as ações necessárias para a realização de um projeto, tarefa ou objetivo. Seu nome deriva das iniciais de sete perguntas em inglês, que ajudam a orientar e estruturar a execução das atividades:

- What (O quê): quais atividades ou tarefas serão realizadas.
- Why (Por quê): os motivos e objetivos da execução.
- Where (Onde): o local onde a ação ocorrerá.
- When (Quando): o cronograma das ações.
- Who (Quem): os responsáveis pela execução.
- How (Como): os métodos e processos utilizados.
- How much (Quanto): os custos envolvidos.

De acordo com Chiavenato (2004), o 5W2H é um instrumento fundamental para a aplicação prática do planejamento, pois facilita o desdobramento de ações e responsabilidades de forma clara e objetiva. Sua importância está na capacidade de proporcionar clareza na execução de projetos, ajudando no alinhamento das equipes, definição de papéis, antecipação de problemas e melhor comunicação entre os envolvidos.

Na Imagem 3 a seguir, é apresentado o plano 5W2H do projeto da academia Gigantes.

Figura 3 – matriz 5w2h

	5H					2H	
	O quê?	Porque?	Quem?	Quando?	Onde?	Como?	Quanto
Solução 1	Implementar um software de gestão de clientes, funcionários e finanças.	Melhorar a eficiência, precisão e acessibilidade das informações.	Gerente: Adriano Rodrigues Gomes da Silva. Equipe: Leonardo Santos Parra.	9/20/2023	Secretaria da academia	Usando técnicas de elicitação de requisitos, BPMN, linguagem JAVA.	R$ 5.816,4
Solução 2	Proporcionar um treino de como utilizar o software para os funcionários.	Aumentar a eficiência do funcionário no uso do software.	Gerente: Adriano Rodrigues Gomes da Silva. Equipe: Leonardo Santos Parra.	9/20/2023	Secretaria da academia	Fornecendo aulas semanais aos funcionários.	R$ 5.816,4
Solução 3	Digitalizar os dados financeiros e dos clientes para o software.	Melhorar a segurança das informações.	Gerente: Adriano Rodrigues Gomes da Silva. Equipe: Leonardo Santos Parra.	9/20/2023	Secretaria da academia	Reescrevendo as informações do caderno para o software.	R$ 5.816,4

Fonte: Autores

A questão problema do 5w2h acima foi: “como melhorar a gestão das atividades e das finanças de uma academia de musculação que ainda utiliza registros em cadernos”

2.3 VIABILIDADE

Foi escolhida uma solução entre as três propostas, e a opção considerada mais adequada consiste na implementação de um *software* de gestão de clientes, funcionários e finanças. Essa escolha se baseia na capacidade do *software* em otimizar processos, reduzir erros e aprimorar a experiência do cliente, contribuindo significativamente para uma gestão mais eficiente. Além disso, a ferramenta irá proporcionar um controle mais preciso dos paga,emtos, destacando-se como uma estratégia vital para o sucesso e crescimento sustentável do negócio da academia de pequeno porte.

A implementação do *software* de gestão aborda de maneira eficaz os desafios previamente identificados. Em relação à falta de gerenciamento financeiro, o *software* pode proporcionar uma solução abrangente automatizando as transações e facilitando a vida do usuário.

Enquanto a utilização de um caderno, a implementação do *software* elimina essa prática antiquada. Ao centralizar o cadastro de clientes e o controle de pagamentos no sistema, reduz-se significativamente a possibilidade de erros e inconsistências. O *software* oferece uma plataforma organizada e segura, substituindo o método manual por um processo automatizado, promovendo assim maior eficiência no controle administrativo. Essa transição contribui não apenas para a precisão dos registros, mas também para uma gestão mais ágil e profissional, melhorando a experiência do cliente e assim podendo fortalecer a reputação da academia

3 Levantamento de Requisitos

3.1 Elicitação e especificação dos Requisitos

A elicitação de requisitos é uma das etapas mais importantes no desenvolvimento de software, pois é nela que buscamos entender o que realmente o cliente e os usuários esperam do sistema. É como se fosse o momento de "levantar as cartas na mesa" para saber exatamente o que deve ser feito.

Segundo Balieiro e Pinto (2024), essa fase é essencial para definir as funcionalidades e restrições do sistema, garantindo que o produto final atenda às necessidades do cliente e evite retrabalhos futuros.

No nosso projeto, optamos por uma abordagem prática: realizamos visitas à academia para observar o ambiente e entender melhor o contexto. Essa técnica de observação é bastante eficaz para captar detalhes que, muitas vezes, passam despercebidos em entrevistas formais (Couto, 2023).

Com base nessas visitas, elaboramos um questionário direcionado ao proprietário da academia. O uso de questionários é uma técnica comum na engenharia de requisitos, pois permite coletar informações de forma estruturada e eficiente (Alflen & Prado, 2021).

Essa combinação de observação e questionário nos proporcionou uma visão mais completa das necessidades e expectativas do cliente, permitindo que a

especificação dos requisitos fosse mais precisa e alinhada com a realidade da academia.

Além disso, ter requisitos bem definidos desde o início facilita a comunicação entre todos os envolvidos no projeto e ajuda a controlar possíveis mudanças ao longo do desenvolvimento.

Em resumo, a elicitação de requisitos não é apenas uma formalidade, mas sim uma etapa crucial para o sucesso de qualquer projeto de software. Investir tempo e esforço nessa fase inicial pode evitar muitos problemas no futuro e garantir que o produto final realmente atenda às expectativas dos usuários.

3.2 BPMN

BPMN (*Business Process Model and Notation*) é uma representação gráfica feita a partir de ícones que simbolizam o fluxo de processo. Ou seja, a partir dessa notação é possível fazer o mapeamento dos processos. Portanto, cada ícone representa uma etapa do processo de produção.

A principal finalidade do BPMN é fornecer uma linguagem comum para descrever os processos de negócios, independentemente da indústria ou setor. Ele utiliza símbolos gráficos intuitivos para representar atividades, eventos, gateways, fluxos de sequência e outras partes dos processos de negócios. Isso torna mais fácil para as organizações documentarem, analisarem, otimizarem e comunicarem seus processos internos e externos.

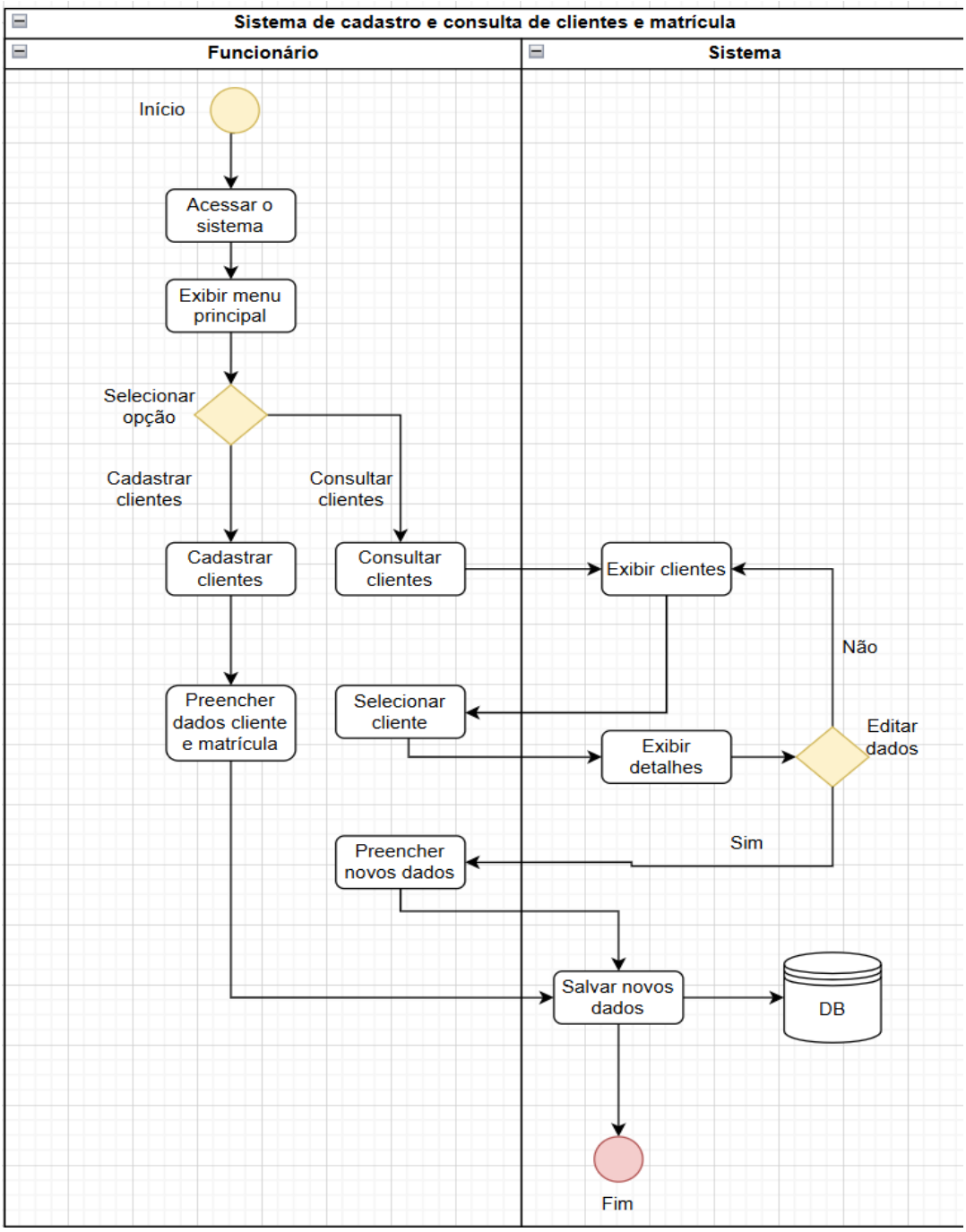
É importante criar um BPMN porque ele gera vários benefícios como:

- Padronização: O BPMN oferece uma padronização na representação de processos, o que facilita a compreensão e colaboração entre diferentes partes interessadas.
- Automação: O BPMN é frequentemente utilizado em conjunto com sistemas de automação de processos, permitindo a implementação direta dos processos modelados em ferramentas de software.
- Melhoria de Processos: Ao visualizar os processos de negócios, as organizações podem identificar áreas para melhoria e otimização, contribuindo para uma maior eficiência operacional.

Ele desempenha um papel fundamental na gestão de processos e na busca por operações mais eficazes e alinhadas com os objetivos estratégicos da organização.

No contexto deste projeto, o BPMN foi utilizado para representar os processos da academia de forma visual e padronizada, facilitando a análise do funcionamento do sistema. As figuras a seguir apresentam os diagramas BPMN que descrevem os fluxos de cadastro de alunos, matrículas, ficha de treino e controle de pagamento.

Figura 4 – BPMN Cliente e Matrícula

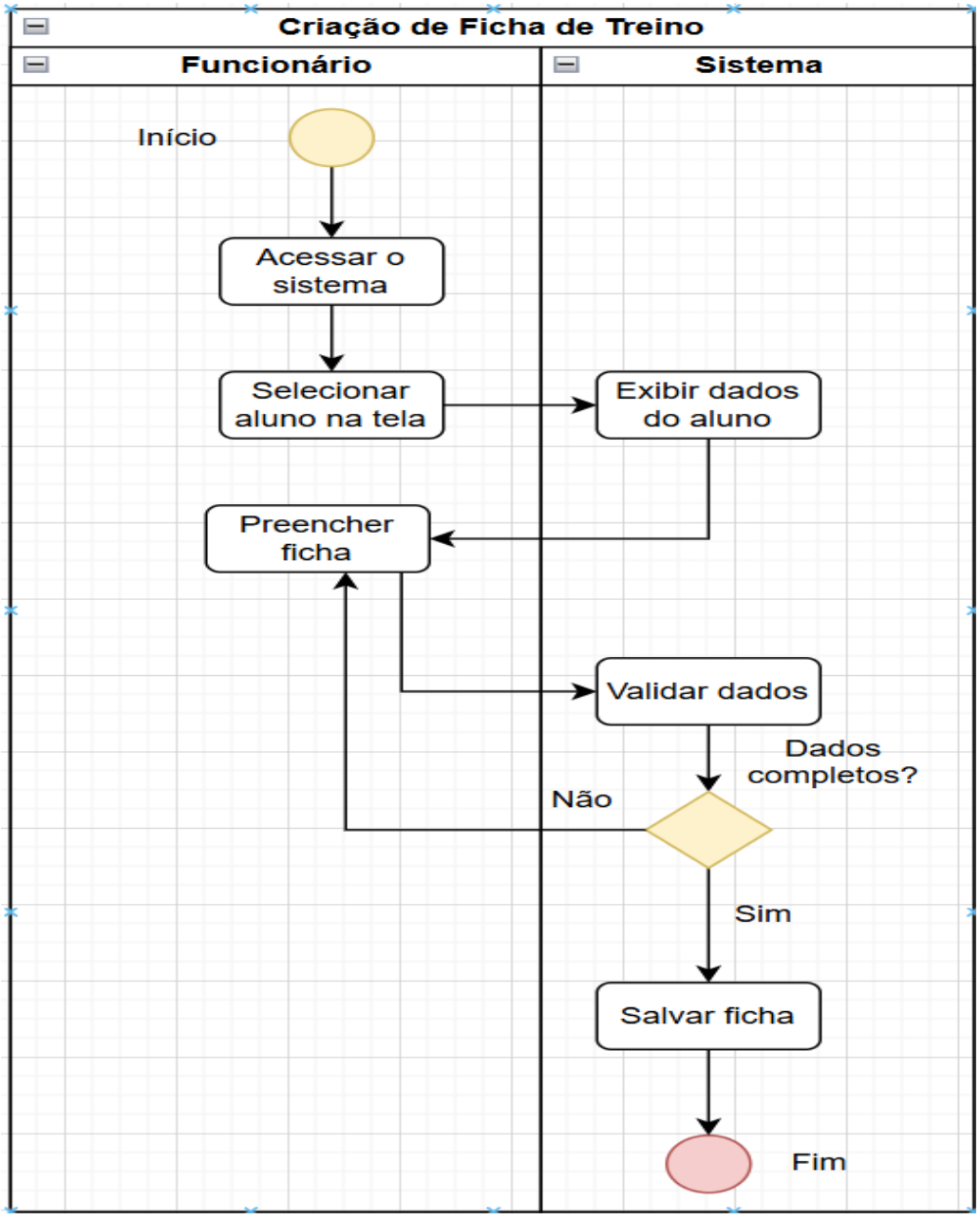


Fonte: Autores

Esse BPMN acima mostra o fluxo de um funcionário acessando o sistema para cadastrar ou consultar clientes. Ao consultar, ele pode visualizar detalhes, editar dados se necessário e, ao final, os dados são salvos no banco. Esse processo facilita a gestão e atualização das informações dos alunos.

A seguir na figura 5, temos o BPMN sobre o processo da criação de ficha de treino.

Figura 5 – BPMN Ficha de Treino

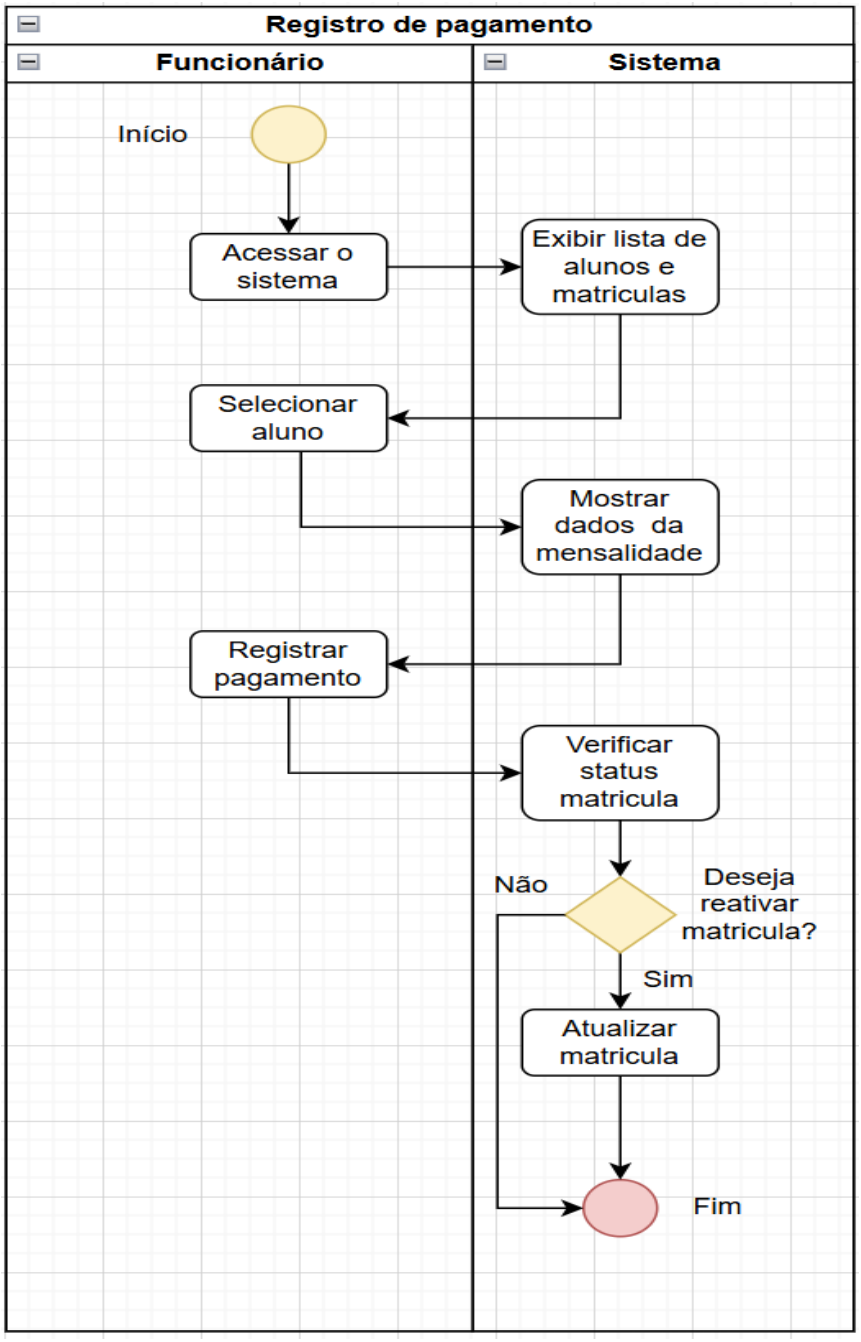


Fonte: Autores

Aqui o funcionário acessa o sistema, seleciona um aluno e preenche a ficha de treino. O sistema valida os dados e, se estiver tudo certo, salva a ficha. O fluxo é simples e direto, garantindo que a ficha só seja salva quando estiver completa.

E por fim, na figura 6, o BPMN sobre o processo de registro de pagamento e atualização de matrícula.

Figura 6 – BPMN registro de pagamento



Fonte: Autores

Esse último BPMN representa o processo de registrar pagamentos. O funcionário acessa o sistema, seleciona o aluno e registra o pagamento. O sistema verifica se é necessário reativar a matrícula e, caso sim, atualiza o status. Isso mantém o controle financeiro e de matrícula em dia.

3.3 Requisitos Funcionais

Requisitos funcionais são especificações detalhadas das funcionalidades que um sistema, software ou produto deve possuir para atender às necessidades e expectativas do usuário. Eles descrevem o que o sistema deve fazer em termos de operações, serviços ou atividades específicas, eles basicamente definem as características funcionais que o produto ou sistema deve apresentar para satisfazer seus usuários.

Segundo Sommerville (2011), requisitos funcionais “descrição dos serviços que o sistema deve oferecer e como ele deve reagir a entradas específicas”. Em outras palavras, eles dizem **“**o que o sistema deve fazer**”**, e não como ele deve fazer.

Além disso, os requisitos funcionais são essenciais para guiar o desenvolvimento e garantir que o sistema entregue valor real ao usuário, pois representam diretamente o que foi solicitado. Eles costumam ser levantados durante entrevistas, reuniões e análises com os usuários ou clientes.

Quadro 1 - Documentação de Requisitos

RF001- Exibir menu principal	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir a exibição de uma tela com todas as opções que o usuário pode selecionar.		
RF002-Cadastrar novo cliente e matrícula	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média

		( ) Baixa
Descrição: O sistema deve permitir ao usuário registrar os dados pessoais e plano de matrícula de um novo cliente.		
RF003-Consultar lista de clientes	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir ao usuário visualizar a lista completa de clientes cadastrados.		
RF004-Exibir dados do cliente	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir ao usuário editar as informações de um cliente previamente cadastrado.		
RF005-Selecionar aluno para ficha	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir ao funcionário selecionar o aluno desejado para preenchimento de ficha de treino.		
RF006-Preencher ficha de treino	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir o preenchimento dos dados da ficha de treino.		
RF007-Validar dados da	Categoria:	Prioridade:

ficha	( ) Oculto (X)Evidente	(X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve verificar se todos os campos obrigatórios da ficha foram preenchidos.		
RF008-Exibir alunos com matrículas	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir a apresentação de alunos com os status de suas matrículas.		
RF009-Exibir dados da mensalidade	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir a exibição os valores, datas e plano da mensalidade do aluno selecionado.		
RF010-Registrar pagamento	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o usuário registre um pagamento recebido do aluno.		
RF011-Verificar status da matrícula	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa

Descrição: O sistema deve verificar se a matrícula está vencida ou ativa no momento do pagamento.		
RF012-Atualizar status da matrícula	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve atualizar a matrícula para ativa caso esteja vencida e o pagamento seja confirmado.		
RF013-Atualizar dados do cliente	Categoria: ( ) Oculto (X)Evidente e	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir que o usuário altere informações previamente cadastradas do cliente.		
RF014-Atualizar ficha de treino	Categoria: ( ) Oculto (X)Evidente e	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir ao usuário editar fichas de treino já cadastradas.		
RF015-Gerar relatórios simples	Categoria: ( ) Oculto (X)Evidente	Prioridade: (X) Altíssima ( ) Alta ( ) Média ( ) Baixa
Descrição: O sistema deve permitir a geração de relatórios em tela com informações como: Alunos ativos/inativos, planos mais vendidos		

Fonte: Autores

3.4 Casos de Uso

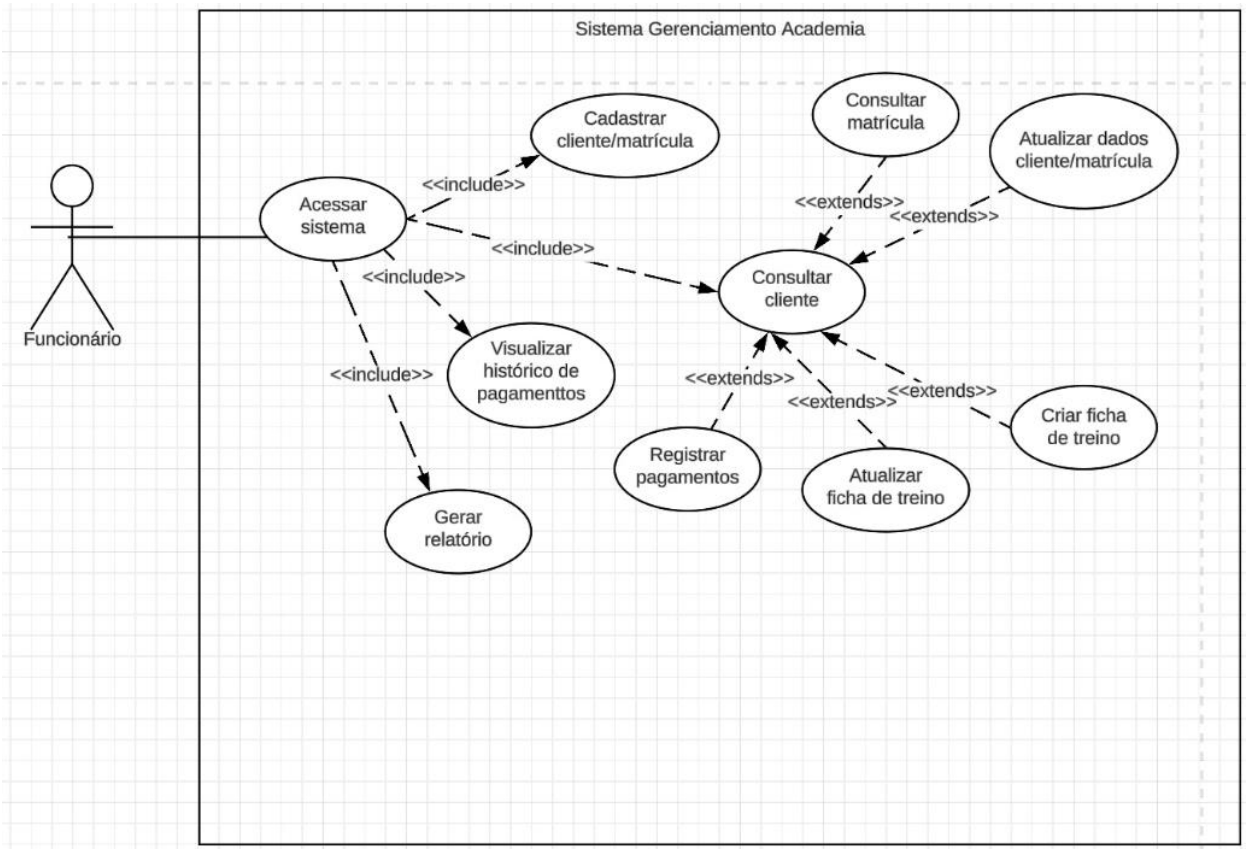
O Caso de Uso é uma ferramenta essencial na análise de sistemas, utilizada para descrever as interações entre os usuários (ou outros sistemas) e o sistema em desenvolvimento. De acordo com Cockburn (2001), um caso de uso define uma sequência de ações executadas por um ator externo com o objetivo de alcançar um resultado específico, utilizando os recursos oferecidos pelo sistema. Essa abordagem permite visualizar o comportamento esperado do sistema em resposta às ações dos usuários, fornecendo uma perspectiva prática e centrada nas funcionalidades.

A principal relevância dos casos de uso está em sua capacidade de aproximar os requisitos do cliente da realidade técnica da equipe de desenvolvimento. Eles funcionam como uma ponte de comunicação entre os envolvidos no projeto, garantindo que todos compreendam de forma clara o que o sistema deve realizar. Além disso, contribuem diretamente para a identificação dos requisitos funcionais, facilitam o planejamento dos testes e promovem um desenvolvimento mais alinhado com as necessidades reais dos usuários.

3.4.1 Diagrama de Caso de Uso

Neste projeto, foi desenvolvido o caso de uso representado na figura 5. A utilização dessa ferramenta permitiu descrever com clareza como os usuários irão interagir com o sistema de gerenciamento da academia, facilitando o alinhamento entre os desenvolvedores e o cliente. Dessa forma, garantiu-se que funcionalidades essenciais, como o cadastro de alunos, a criação de fichas de treino e o controle de matrículas, fossem contempladas entre os requisitos do sistema.

Figura 7 - diagrama de Caso de Uso



Fonte: Autores

O diagrama apresentado na figura 5 reune todas as funcionalidades disponveis para o funcionário do sistema.

3.4.2 Documentação de Caso de Uso

A documentação de caso de uso (quadro 2) é uma forma estruturada de descrever os comportamentos esperados do sistema a partir das interações dos usuários com suas funcionalidades. Segundo Sommerville (2011), ela oferece uma visão clara dos fluxos de trabalho e das ações executadas pelos usuários, auxiliando na identificação precisa dos requisitos e evitando interpretações ambíguas ou inconsistentes. Assim, torna-se um recurso fundamental tanto para a equipe técnica quanto para os stakeholders, servindo como referência ao longo de todo o desenvolvimento do sistema.

Quadro 2 - Documentação de Caso de Uso

Caso de Uso – Acessar sistema	
ID	UC 001
Descrição	Este caso de uso descreve o processo pelo qual o funcionário acessa o sistema para utilizar suas funcionalidades.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional.
Cenário Principal	• O funcionário acessa a página inicial do sistema. • O funcionário tem acesso às funcionalidades do sistema.
Pós-condição	O funcionário visualiza a tela inicial com as funcionalidades principais.
Cenário Alternativo	2a. Se Houver erro ao carregar o sistema, uma mensagem informativa é exibida.
Caso de Uso – Cadastrar Ciente	
ID	UC 002
Descrição	Este caso de uso permite que o funcionário cadastre um novo aluno no sistema, inserindo seus dados pessoais.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional.
Cenário Principal	• O funcionário acessa a opção “cadastrar cliente” • O sistema exibe o formulário de cadastro de cliente e sua respectiva matrícula • O funcionário preenche os dados e confirma. • O sistema salva os dados e confirma o cadastro
Pós-condição	O funcionário cadastrou o cliente no sistema com sucesso.
Cenário Alternativo	3a. Se houver campos obrigatórios não preenchidos, o sistema exibe uma mensagem de erro.
Caso de Uso – Consultar Cliente	
ID	UC 003
Descrição	Este caso de uso permite que o funcionário busque e visualize os dados de um cliente já cadastrado no sistema.

Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e o cliente deve estar cadastrado.
Cenário Principal	• O funcionário acessa a opção "Consultar cliente". • O sistema solicita o critério de busca. respectiva matrícula • O funcionário informa os dados. • O sistema exibe as informações do cliente.
Pós-condição	Os dados do cliente são exibidos para o funcionário e uma decisão é tomada.
Cenário Alternativo	4a. Se o cliente não for encontrado, o sistema informa que não há resultados.
Caso de Uso – Consultar matrícula	
ID	UC 004
Descrição	Este caso de uso permite que o funcionário verifique o status de matrícula do cliente.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e o cliente deve estar cadastrado.
Cenário Principal	• A partir da consulta do cliente, o funcionário acessa o status de matrícula. • O sistema exibe o plano, validade e status (ativo/inativo).
Pós-condição	O status da matrícula é exibido ao funcionário e uma decisão é tomada.
Cenário Alternativo	2a. Se o cliente não tiver uma matrícula, o sistema exibe uma mensagem.
Caso de Uso – Atualizar dados cliente/matrícula	
ID	UC 005
Descrição	Este caso de uso permite ao funcionário atualizar os dados do cliente ou informações da matrícula.
Ator Primário	Funcionário

Pré-condição	O sistema deve estar operacional e o cliente deve estar cadastrado.
Cenário Principal	• O funcionário acessa a lista de clientes. • Seleciona "Atualizar dados". • O sistema exibe os campos editáveis. • O funcionário altera e os novos dados são salvos.
Pós-condição	Os novos dados são atualizados com sucesso.
Cenário Alternativo	4a. Se houver erro de validação, o sistema exibe uma mensagem e não salva.
Caso de Uso – Criar ficha de treino	
ID	UC 006
Descrição	Este caso de uso permite criar uma nova ficha de treino associada a um cliente.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e o cliente deve estar cadastrado.
Cenário Principal	• O funcionário seleciona o cliente. • O funcionário acessa "Criar ficha de treino". • O funcionário preenche os dados do treino. • A ficha é salva no sistema.
Pós-condição	A ficha de treino é criada com sucesso e fica disponível para visualização
Cenário Alternativo	3a. Se faltar preencher algum dado obrigatório, o sistema informa o erro.
Caso de Uso – Atualizar ficha de treino	
ID	UC 007
Descrição	Este caso de uso permite editar uma ficha de treino existente de um cliente.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e o cliente deve possuir uma ficha de treino.
Cenário Principal	• O funcionário localiza a ficha do cliente.

	• O funcionário seleciona "Editar". • O funcionário preenche os dados do treino. • Os novos dados são salvos
Pós-condição	A ficha de treino é atualizada com sucesso e fica disponível para visualização
Cenário Alternativo	3a. Se os dados forem inválidos, o sistema exibe uma mensagem de erro.
Caso de Uso – Registrar pagamentos	
ID	UC 008
Descrição	Este caso de uso permite registrar o pagamento da mensalidade de um cliente já cadastrado no sistema.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e o cliente deve estar cadastrado.
Cenário Principal	• O funcionário acessa a área do cliente. • O funcionário seleciona a opção "Registrar pagamento". • Insere os dados (valor, data) e confirma. • O sistema salva o pagamento. • O sistema atualiza a matrícula se for necessário.
Pós-condição	O pagamento é registrado e fica disponivel para visualização no histórico de pagamentos.
Cenário Alternativo	3a. Se houver erro no preenchimento, o sistema impede o registro.
Caso de Uso – Visualizar histórico de pagamentos	
ID	UC 009
Descrição	Este caso de uso permite ao funcionário visualizar os pagamentos realizados por um cliente.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e deve existir pagamentos já realizados anteriormente.

Cenário Principal	• O funcionário acessa a área de histórico de pagamentos. • O sistema exibe a lista com datas e valores. • Uma decisão é tomada pelo funcionário
Pós-condição	O histórico de pagamento é exibido e então o funcionário pode tomar uma decisão
Cenário Alternativo	3a. Se não houver pagamentos, o sistema exibe uma mensagem informando que não há pagamentos registrados.
Caso de Uso – Gerar relatório	
ID	UC 010
Descrição	Este caso de uso permite gerar relatórios administrativos como número de alunos ativos, inativos e planos.
Ator Primário	Funcionário
Pré-condição	O sistema deve estar operacional e com dados cadastrados.
Cenário Principal	• O funcionário acessa "Gerar relatório". • Escolhe o tipo de relatório. • O sistema processa e exibe o resultado.
Pós-condição	O relatório é exibido ao funcionário e uma decisão é tomada a partir disso.
Cenário Alternativo	3a. Se não houver dados, o sistema avisa que não há informações disponíveis.

Fonte: Autores

3.5 Diagrama Entidade-Relacionamento

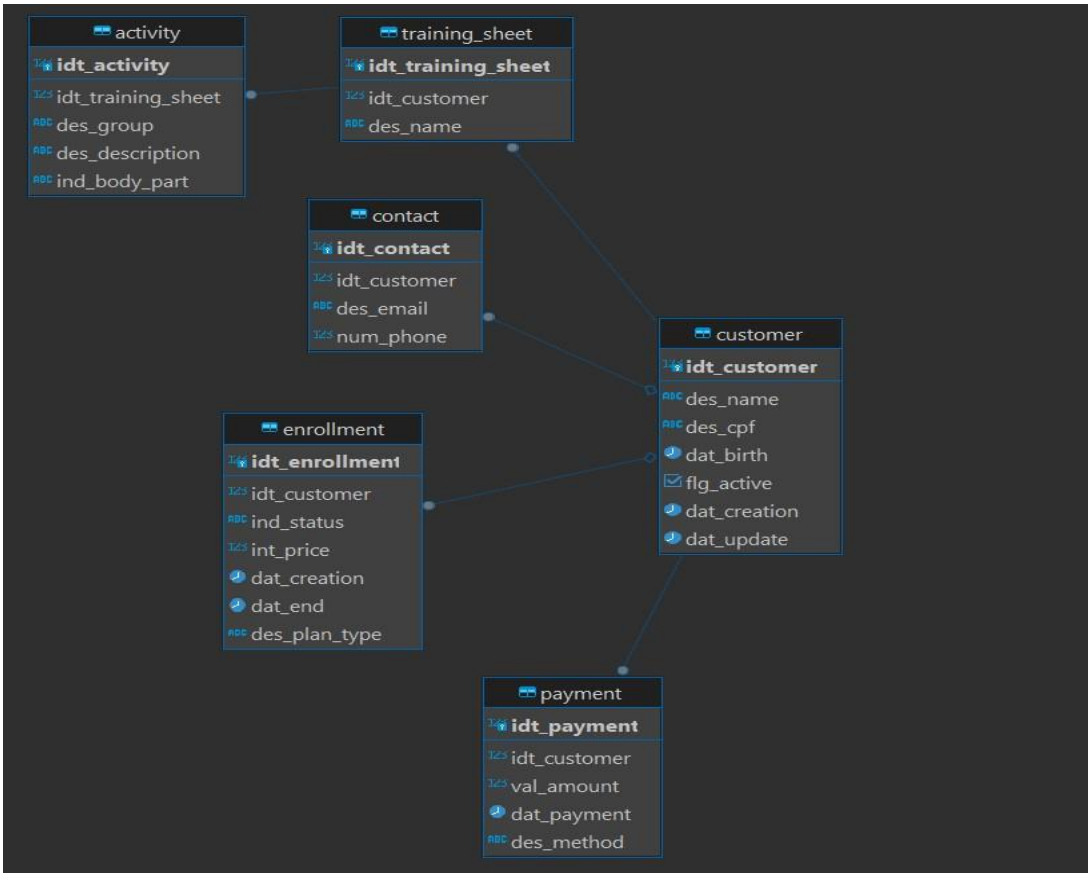
O Diagrama Entidade-Relacionamento (ER) é uma ferramenta amplamente utilizada na modelagem de banco de dados, com o objetivo de representar de forma visual as entidades envolvidas em um sistema, seus atributos e os relacionamentos existentes entre elas. Segundo Elmasri e Navathe (2019), o modelo ER permite

abstrair a estrutura lógica dos dados, facilitando tanto o processo de criação do banco quanto sua posterior manutenção e compreensão por parte da equipe de desenvolvimento.

No contexto do sistema de academia desenvolvido, o diagrama ER foi essencial para organizar as principais entidades envolvidas, como cliente, matrícula, pagamento, ficha de treino e atividade. Com ele, foi possível definir com clareza os relacionamentos, como por exemplo: um cliente pode ter várias matrículas, cada matrícula possui um valor e um período de validade, e um cliente também pode ter diversas fichas de treino e registros de pagamento.

Além de servir como base para a implementação do banco de dados relacional, o diagrama ER também atua como uma forma de documentação visual, facilitando o entendimento da estrutura geral do sistema e auxiliando na integridade e consistência dos dados. Sua utilização contribui para o desenvolvimento de um banco bem estruturado, normalizado e de fácil manutenção.

Figura 8 – Diagrama Enridade-Relacionamento



Fonte: Autores

O diagrama apresentado modela a estrutura do banco de dados do sistema. Ele representa as principais entidades envolvidas nas operações do sistema, seus

atributos e os relacionamentos existentes entre elas.

A entidade central do modelo é customer, que representa os alunos da academia. Cada cliente possui informações básicas como nome (des_name), CPF (des_cpf), data de nascimento (dat_birth), status de atividade (flg_active) e registros de criação e atualização.

Diversas entidades se relacionam diretamente com o cliente:

- contact: guarda os dados de contato do cliente, como e-mail (des_email) e telefone (num_phone). A relação é de 1:N, ou seja, um cliente pode ter vários registros de contato (embora na prática, normalmente tenha apenas um).

- enrollment: representa a matrícula do cliente em algum plano. Contém atributos como status da matrícula (ind_status), valor do plano (int_price), data de criação e término, e o tipo de plano (des_plan_type). Um cliente pode ter múltiplas matrículas ao longo do tempo.

- payment: registra os pagamentos realizados pelo cliente, com valor pago (val_amount), data do pagamento (dat_payment) e método utilizado (des_method). Um cliente pode realizar vários pagamentos.

- training_sheet: refere-se à ficha de treino atribuída ao cliente. Inclui um nome para identificação (des_name) e está vinculada diretamente ao cliente.

A ficha de treino, por sua vez, se relaciona com a entidade activity, que representa os exercícios dentro da ficha. Cada atividade possui um grupo muscular (des_group), uma descrição (des_description) e a parte do corpo trabalhada (ind_body_part). Assim, uma ficha pode conter várias atividades.

Esse modelo é um exemplo clássico de relacionamento 1:N, no qual a entidade customer é a origem de várias outras entidades dependentes. O diagrama ER fornece uma visão clara, organizada e relacional da base de dados, garantindo coesão e integridade na modelagem do sistema da academia.

4 Ferramentas e Métodos

As ferramentas e os métodos estão detalhados a seguir.

4.1 Ferramentas

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade. Além disso, essas ferramentas têm documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e implementarem as soluções. A escolha dessas ferramentas também foi influenciada

pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas. As licenças das ferramentas são de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais.

4.1.1 Diagramas – Draw.io

O Draw.io é uma ferramenta online gratuita usada para criação de diversos tipos de diagramas, como fluxogramas, diagramas de caso de uso, BPMN, ER (entidade-relacionamento) e diagramas UML. Com uma interface simples, é possível montar representações visuais de forma rápida e organizada. A plataforma permite salvar os diagramas direto na nuvem (Google Drive, Dropbox, OneDrive) e exportar em formatos como PNG, JPEG, SVG e PDF, facilitando a documentação de projetos e a comunicação entre equipes. É bastante utilizado na área de desenvolvimento de software por sua praticidade e versatilidade. (Draw.io, 2024)

4.1.2 Documentação - Microsoft Word

O Microsoft Word é um dos editores de texto mais populares e utilizados no mundo, sendo parte integrante do pacote Microsoft Office. Ele permite a criação, edição e formatação de documentos de forma prática e eficiente, com recursos como correção ortográfica, inserção de imagens, tabelas, gráficos e referências. É amplamente usado para produção de trabalhos acadêmicos, relatórios e documentação técnica, já que possui ferramentas específicas para criação de sumário automático, citações, bibliografia e estilos personalizados. Sua interface amigável e os recursos de exportação para PDF facilitam a formatação e entrega de trabalhos como o TCC. (Microsoft Word, 2024)

4.1.3 Desenvolvimento

4.1.3.1 IntelliJ IDEA

O IntelliJ IDEA é uma IDE (Ambiente de Desenvolvimento Integrado) poderosa e muito utilizada por desenvolvedores Java. Desenvolvida pela JetBrains, ela oferece recursos inteligentes como autocompletar código, análise de erros em tempo real, refatoração automática e integração com ferramentas de versionamento como Git. A interface é organizada e voltada para produtividade, facilitando o desenvolvimento de aplicações Java robustas. É bastante usada em projetos acadêmicos e profissionais por sua estabilidade e suporte a diversos frameworks. (JetBrains, 2024)

4.1.3.2 Visual Studio Code (VSCode)

O Visual Studio Code é um editor de código-fonte leve, gratuito e altamente personalizável, desenvolvido pela Microsoft. Suporta várias linguagens de programação, como JavaScript, Java, Python, entre outras, e conta com uma enorme biblioteca de extensões. Possui recursos como terminal integrado, depuração, controle de versão com Git e destaque de sintaxe, o que o torna ideal para projetos web e aplicações front-end. Sua interface simples e agilidade tornam o VSCode uma escolha comum entre desenvolvedores iniciantes e experientes. (VSCode, 2024)

4.1.3.3 Java

Java é uma linguagem de programação orientada a objetos, conhecida por sua portabilidade, segurança e robustez. Muito utilizada no desenvolvimento de sistemas corporativos, aplicativos Android e back-ends de aplicações web, ela permite a construção de soluções escaláveis e bem estruturadas. Por ser fortemente tipada e ter uma vasta comunidade, Java se mantém como uma linguagem sólida tanto no meio acadêmico quanto no mercado de trabalho. No TCC, foi utilizada principalmente na construção da lógica de negócio e na comunicação com o banco de dados. (Oracle, 2024)

4.1.3.4 React

O React é uma biblioteca JavaScript voltada para a construção de interfaces de usuário, criada pelo Facebook. Ele permite criar aplicações web modernas e dinâmicas por meio de componentes reutilizáveis. Sua abordagem reativa torna a experiência do usuário mais fluida, já que atualiza apenas os elementos necessários na tela. React também facilita a organização do código front-end e a integração com APIs. No projeto do TCC, foi utilizado para desenvolver a interface do sistema, tornando o acesso mais visual e amigável. (React, 2024)

4.1.3.5 DBeaver

O DBeaver é uma ferramenta gráfica para gerenciamento de bancos de dados, compatível com sistemas como PostgreSQL, MySQL, Oracle, entre outros. Ele permite executar comandos SQL, visualizar tabelas e relações de forma prática, além de facilitar a modelagem e análise dos dados. No projeto, foi utilizado para gerenciar e testar o banco de dados PostgreSQL durante o desenvolvimento. (DBeaver Community, 2024)

4.2 Métodos

O desenvolvimento do sistema da plusfit seguiu uma linha bem prática e direta, sempre pensando nas necessidades reais da academia. Tudo começou com a elaboração de um questionário, com o objetivo de entender os principais desafios enfrentados na gestão — como controle de matrícula, treinos e pagamentos.

Com base nas respostas, foi feita uma análise SWOT para identificar os pontos fortes, fracos, oportunidades e ameaças do negócio. A partir das fraquezas identificadas, utilizamos a matriz 5W2H para planejar ações que pudessem ser resolvidas com o sistema.

Na sequência, desenvolvemos um diagrama BPMN (Business Process Model and Notation), que nos ajudou a visualizar o funcionamento dos principais processos da academia e serviu de base para o desenvolvimento técnico.

A partir desse mapeamento, criamos a documentação de requisitos e os diagramas de casos de uso, que ajudaram a organizar melhor as funcionalidades. As telas da aplicação foram criadas diretamente no desenvolvimento, sem prototipação prévia, o que deixou o processo mais ágil e focado.

Com todas essas etapas, conseguimos estruturar um sistema funcional, prático e pensado para a realidade da plusfit.

5 Desenvolvimento

O sistema foi desenvolvido com o objetivo de atender as necessidades de uma academia, garantindo organização, desempenho e facilidade de manutenção. Para isso, foram aplicadas boas práticas de programação, como separação por camadas (*controller*, *service*, *repository*), uso de DTOs e validações, além de uma estrutura limpa e bem definida.

O *backend* foi construído em Java, utilizando o *framework Spring Boot*, seguindo os princípios da orientação a objetos e respeitando regras de negócio específicas, como controle de matrícula, status de pagamentos e fichas de treino. O frontend está sendo desenvolvido com React, visando criar uma interface moderna, responsiva e fácil de usar. Embora as telas ainda estejam em finalização, elas já foram planejadas e seguirão o fluxo das funcionalidades implementadas na API.

Neste capítulo, são apresentados os principais pontos do desenvolvimento, com trechos relevantes de código, explicações sobre técnicas utilizadas e prints dos testes realizados via *Insomnia*.

5.1 Tecnologias e Ferramentas Utilizadas

A seleção das tecnologias foi feita com base na experiência do desenvolvedor e na maturidade das ferramentas no mercado, visando estabilidade, documentação rica e comunidade ativa. As principais utilizadas foram:

- Java: Linguagem de programação orientada a objetos, segura e performática, amplamente utilizada em sistemas corporativos.
- Spring Boot: Framework que facilita a criação de aplicações Java, oferecendo configuração automática, suporte a APIs REST e integração com banco de dados.
- Spring Data JPA: Biblioteca que simplifica a persistência de dados, permitindo a criação de repositórios com mínima configuração.
- PostgreSQL: Banco de dados relacional robusto e open-source, com suporte a tipos complexos e transações ACID.
- Lombok: Biblioteca que reduz a verbosidade do código Java com anotações para geração de getters, setters, builders e construtores.
- Insomnia: Ferramenta de testes de API, utilizada para simular requisições e validar respostas.
- Gradlew: Gerenciador de dependências e build da aplicação.
- React: Biblioteca JavaScript para construção do frontend, focada em componentes reutilizáveis e interfaces reativas.

Essas ferramentas permitiram maior agilidade no desenvolvimento, bem como padronização e clareza na organização do projeto.

5.2 Estrutura do Projeto

O sistema foi estruturado seguindo a arquitetura em camadas, que separa as responsabilidades de cada componente da aplicação:

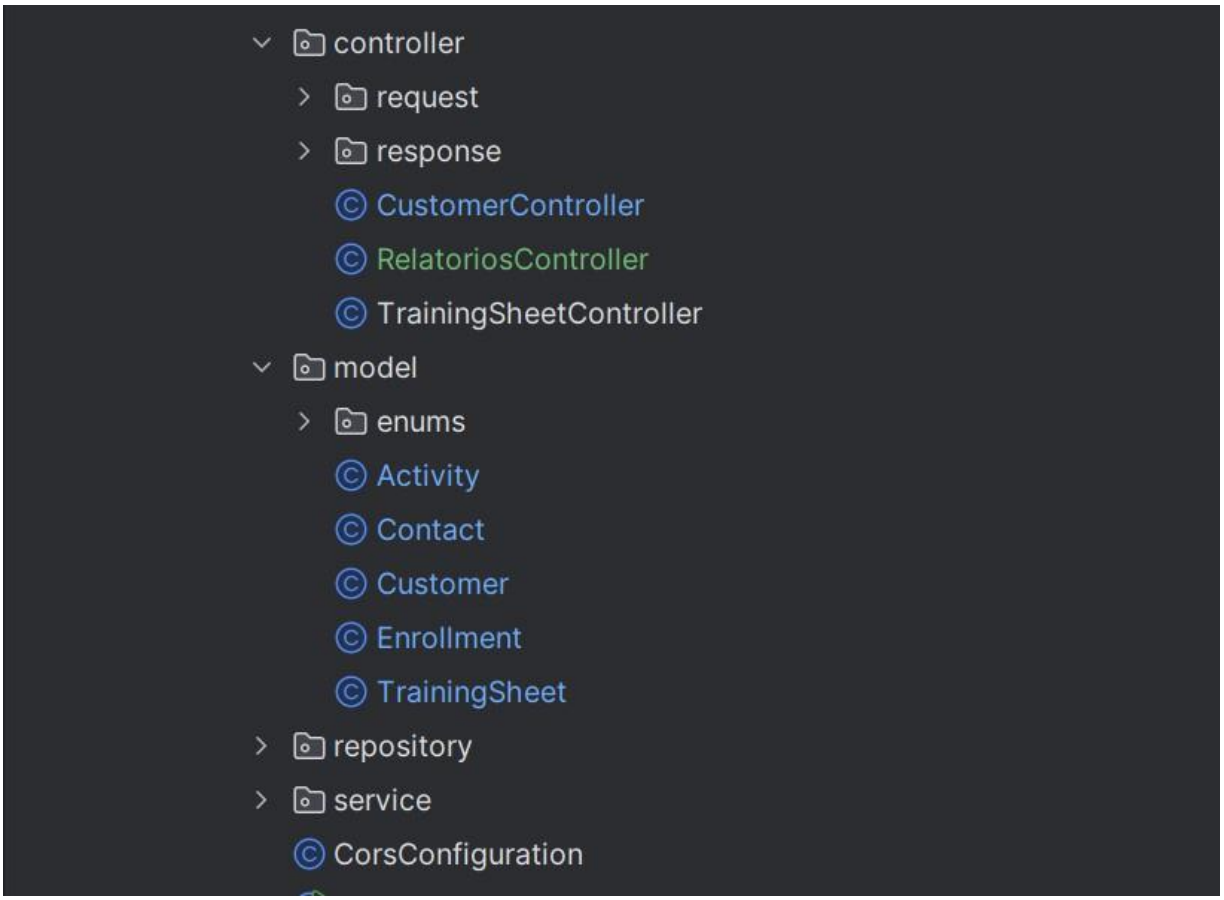
- Controller:Recebe e responde às requisições HTTP.
- Service: Contém a lógica de negócio. Aqui são validadas regras específicas como status da matrícula ou vencimento de plano.
- Repository: Responsável pela persistência dos dados. Utiliza a interface do Spring Data JPA para comunicação com o banco.
- DTOs (Data Transfer Objects): Divididos entre request e response,

os DTOs evitam o acoplamento direto entre as entidades JPA e os dados que trafegam na API.

- Model: Contém as entidades da aplicação, anotadas com JPA para mapeamento no banco de dados.

Essa separação promove a escalabilidade e facilita a manutenção do sistema ao longo do tempo. Na figura a seguir podemos ver como todas as camadas são separadas.

Figura 9 - Separação de camadas



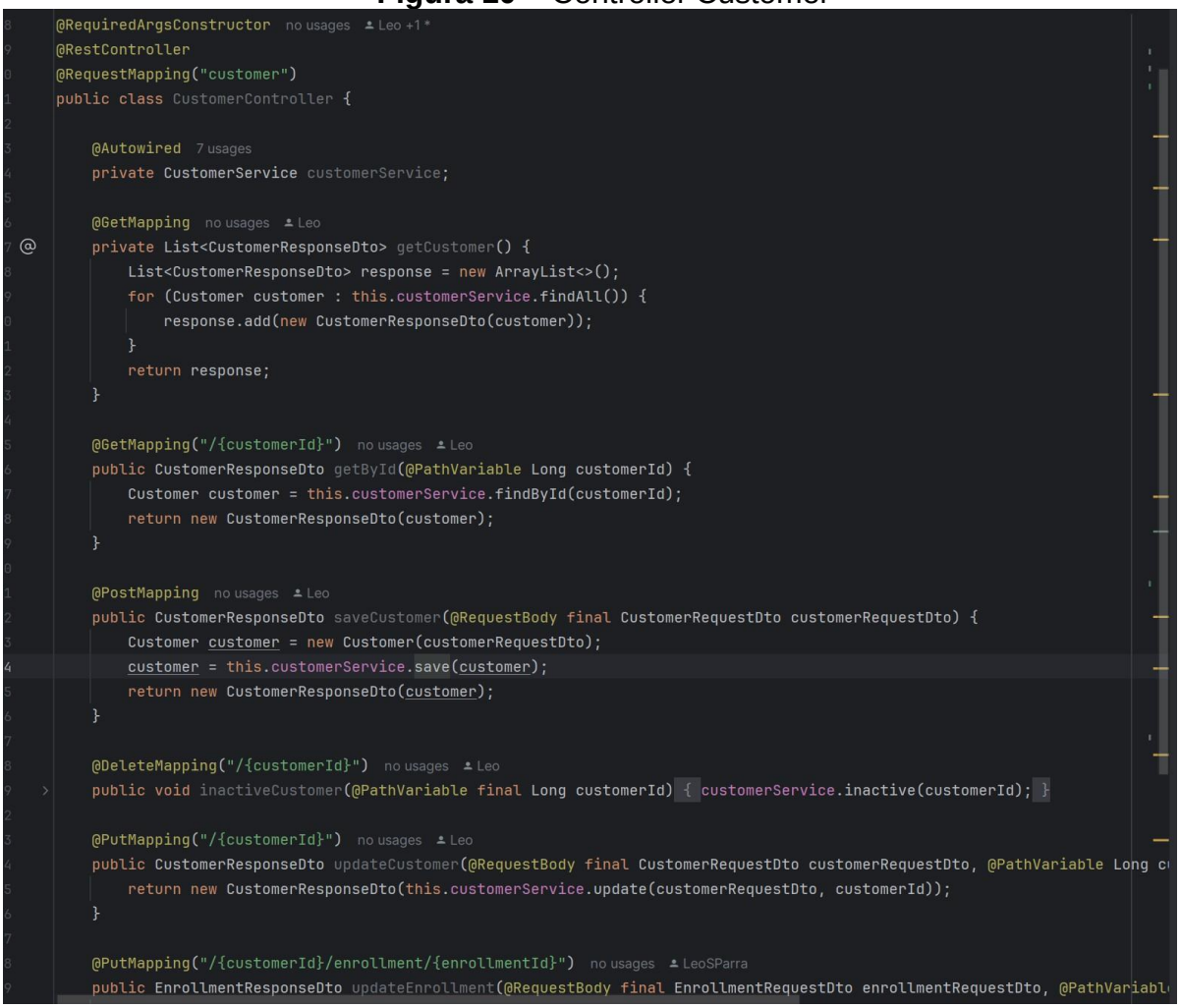
Fonte : Autores

5.3 Implementação das Funcionalidades

5.3.1 Cadastro de Cliente

O cadastro de cliente é realizado via requisição POST para o *endpoint* /customer. A camada *Controller* recebe o DTO com os dados do cliente, valida as informações e envia à camada de serviço, que por sua vez chama o repositório para persistir os dados. Na figura a seguir podemos ver o trecho relevante do código do controller de *customer*.

Figura 10 – Controller Customer



```
@RequiredArgsConstructor  no usages  Leo +1 *
@RestController
@RequestMapping("customer")
public class CustomerController {

    @Autowired  7 usages
    private CustomerService customerService;

    @GetMapping  no usages  Leo
    private List<CustomerResponseDto> getCustomer() {
        List<CustomerResponseDto> response = new ArrayList<>();
        for (Customer customer : this.customerService.findAll()) {
            response.add(new CustomerResponseDto(customer));
        }
        return response;
    }

    @GetMapping("/{customerId}")  no usages  Leo
    public CustomerResponseDto getById(@PathVariable Long customerId) {
        Customer customer = this.customerService.findById(customerId);
        return new CustomerResponseDto(customer);
    }

    @PostMapping  no usages  Leo
    public CustomerResponseDto saveCustomer(@RequestBody final CustomerRequestDto customerRequestDto) {
        Customer customer = new Customer(customerRequestDto);
        customer = this.customerService.save(customer);
        return new CustomerResponseDto(customer);
    }

    @DeleteMapping("/{customerId}")  no usages  Leo
    public void inactiveCustomer(@PathVariable final Long customerId) { customerService.inactive(customerId); }

    @PutMapping("/{customerId}")  no usages  Leo
    public CustomerResponseDto updateCustomer(@RequestBody final CustomerRequestDto customerRequestDto, @PathVariable Long c
        return new CustomerResponseDto(this.customerService.update(customerRequestDto, customerId));
    }

    @PutMapping("/{customerId}/enrollment/{enrollmentId}")  no usages  LeoSParra
    public EnrollmentResponseDto updateEnrollment(@RequestBody final EnrollmentRequestDto enrollmentRequestDto, @PathVariabl
```

Fonte: Autores

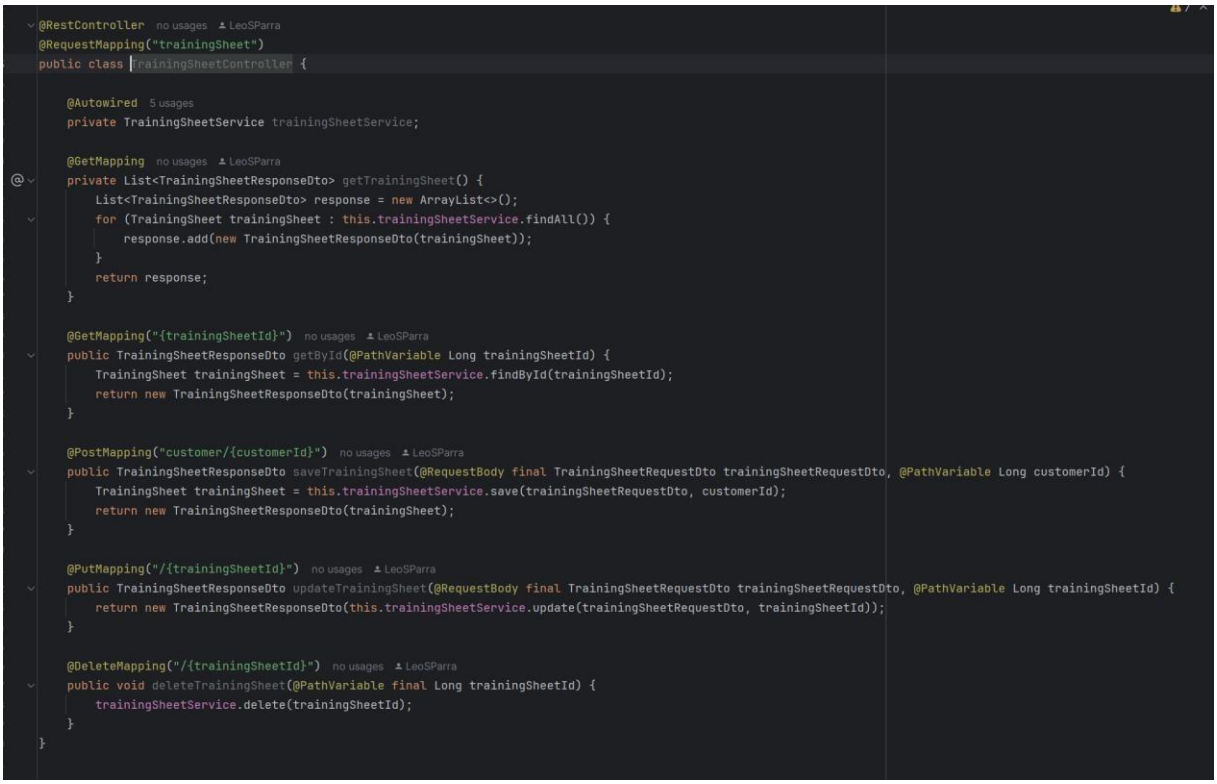
Esse padrão se repete em toda a aplicação, garantindo coesão e padronização no tratamento de requisições.

5.3.2 Criação de Ficha de Treino

Cada aluno pode ter uma ficha de treino associada com exercícios, séries, repetições e observações. A lógica permite associar uma ficha a um cliente e editar os exercícios posteriormente.

Na próxima imagem será apresentado o *controller* da ficha de treino para maior entendimento.

Figura 11 – Controller ficha de treino



```java
@RestController  no usages  LeoSParra
@RequestMapping("trainingSheet")
public class TrainingSheetController {

    @Autowired  5 usages
    private TrainingSheetService trainingSheetService;

    @GetMapping  no usages  LeoSParra
    private List<TrainingSheetResponseDto> getTrainingSheet() {
        List<TrainingSheetResponseDto> response = new ArrayList<>();
        for (TrainingSheet trainingSheet : this.trainingSheetService.findAll()) {
            response.add(new TrainingSheetResponseDto(trainingSheet));
        }
        return response;
    }

    @GetMapping("{trainingSheetId}")  no usages  LeoSParra
    public TrainingSheetResponseDto getById(@PathVariable Long trainingSheetId) {
        TrainingSheet trainingSheet = this.trainingSheetService.findById(trainingSheetId);
        return new TrainingSheetResponseDto(trainingSheet);
    }

    @PostMapping("customer/{customerId}")  no usages  LeoSParra
    public TrainingSheetResponseDto saveTrainingSheet(@RequestBody final TrainingSheetRequestDto trainingSheetRequestDto, @PathVariable Long customerId) {
        TrainingSheet trainingSheet = this.trainingSheetService.save(trainingSheetRequestDto, customerId);
        return new TrainingSheetResponseDto(trainingSheet);
    }

    @PutMapping("/{trainingSheetId}")  no usages  LeoSParra
    public TrainingSheetResponseDto updateTrainingSheet(@RequestBody final TrainingSheetRequestDto trainingSheetRequestDto, @PathVariable Long trainingSheetId) {
        return new TrainingSheetResponseDto(this.trainingSheetService.update(trainingSheetRequestDto, trainingSheetId));
    }

    @DeleteMapping("/{trainingSheetId}")  no usages  LeoSParra
    public void deleteTrainingSheet(@PathVariable final Long trainingSheetId) {
        trainingSheetService.delete(trainingSheetId);
    }
}
```

Fonte: Autores

5.3.3 Controle de Pagamentos

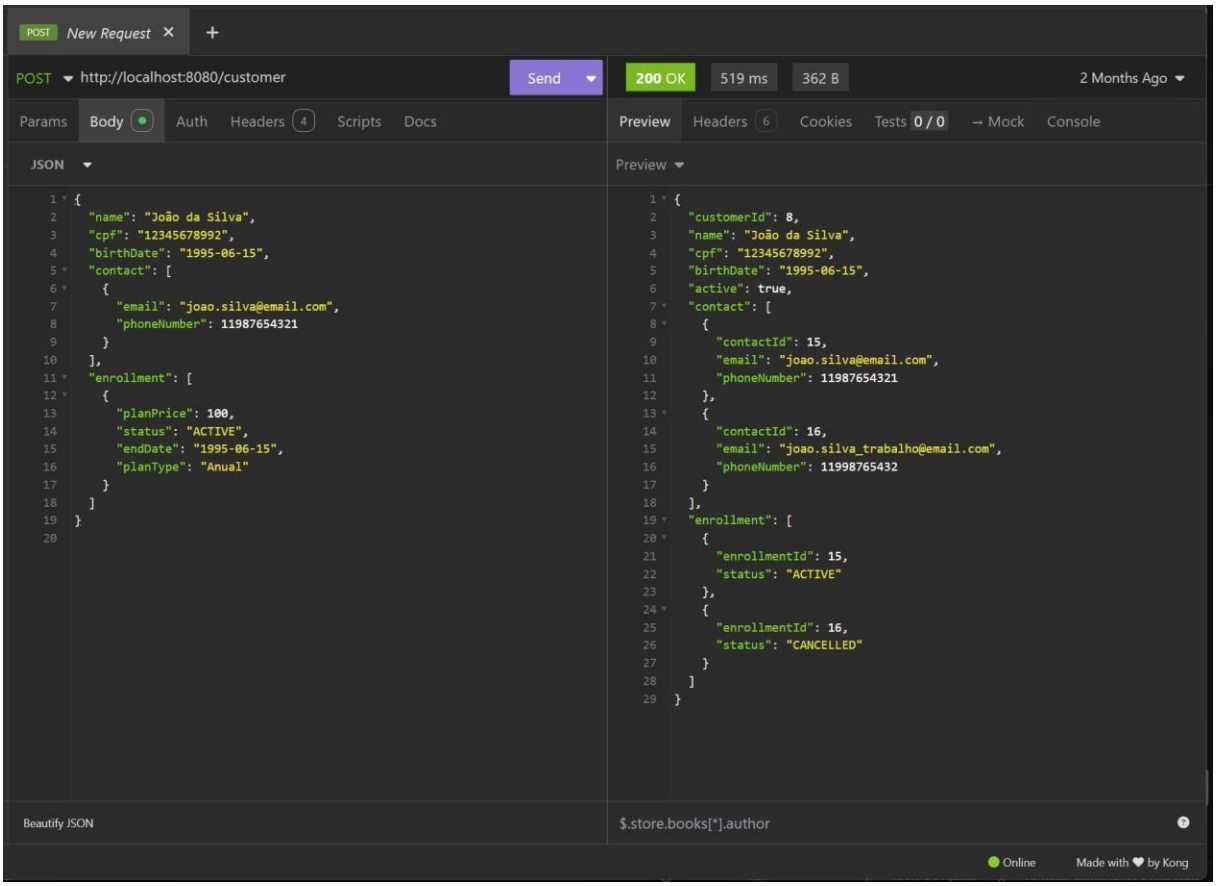
Os pagamentos podem ser registrados manualmente via endpoint POST /payment, e o status de matrícula é atualizado conforme o último pagamento registrado.

5.4 Testes realizados

Os testes realizados foram manuais utilizando-se o insomnia, para simular diferentes cenários, como cadastro com dados inválidos, buscar por ID inexistentes e alterações em registros.

Na figura a seguir um exemplo de um teste para a rota /customer.

Figura 12 – Teste insomnia para cadastro de clientes



Fonte: Autores

5.5 Boas Práticas Adotadas

5.5.1 Separação por camadas

A estrutura do backend foi baseada no padrão de camadas, organizando as responsabilidades da aplicação entre Controller, Service, Repository, Model e DTO. Essa separação facilita a manutenção do código, o isolamento das regras de negócio e a futura implementação de testes unitários.

5.5.2 Utilização de DTOs

Para evitar o acoplamento direto entre as entidades JPA e os dados trafegados pela API, foram utilizados objetos DTO (Data Transfer Object) nas operações de entrada e saída. Isso trouxe mais segurança e controle sobre quais dados são recebidos e enviados.

5.5.3 Uso de Lombok

Para reduzir a repetição de código (como getters, setters e construtores), foi utilizada a biblioteca Lombok. Isso deixou o código mais limpo e fácil de ler,

principalmente nas classes de modelo e DTO.

5.5.4 Padrão de nomenclatura

Todos os nomes de pacotes, classes, variáveis e métodos seguiram convenções claras e consistentes, utilizando inglês técnico e padrões como *camelCase* para variáveis e *PascalCase* para classes. Isso melhora a legibilidade do código e facilita a colaboração com outros desenvolvedores.

5.5.5 Organização por pacotes

A estrutura de pacotes foi planejada para refletir a organização do sistema, separando claramente as funcionalidades e mantendo os arquivos agrupados por responsabilidade. Isso contribuiu para a escalabilidade do projeto e facilitou a navegação durante o desenvolvimento.

6 Resultados e Discussão

6.1 PORTABILIDADE

A portabilidade é um ponto-chave nesse tipo de ferramenta, já que a mobilidade dos profissionais e alunos exige que o sistema funcione bem tanto em notebooks quanto em smartphones ou tablets. Com isso, garantimos mais praticidade no dia a dia e evitamos gargalos no atendimento ou no acompanhamento dos treinos.

Requisitos mínimos para rodar o sistema com fluidez:

- Processador**:** Intel Core i3 ou equivalente
- Memória RAM: 4 GB
- Armazenamento: 128 GB (preferencialmente SSD)
- Sistema operacional: Windows 10 ou superior, Linux, Android 9+ ou iOS 12+
- Navegador**:** Versões atualizadas de Chrome, Firefox, Edge ou Safari
- Banco de dados: MySQL ou PostgreSQL
-

Essas configurações foram pensadas para garantir que o sistema funcione bem durante todo o expediente da academia, mesmo com multitarefas sendo executadas ao mesmo tempo, como cadastro de novos alunos, atualização de fichas de treino ou controle de pagamentos. Embora o acesso fique concentrado nos funcionários, a fluidez e a estabilidade do sistema são essenciais para que o atendimento continue ágil e sem travamentos, principalmente em horários com maior movimento interno,

como abertura e fechamento da academia.

6.2 PROPOSTA COMERCIAL

O sistema digital proposto foi desenvolvido com foco na rotina real de academias, trazendo mais praticidade e organização para quem gerencia o negócio. A ferramenta centraliza as principais funções do dia a dia, como cadastro de alunos, criação de fichas de treino personalizadas e controle das matrículas — incluindo o tempo restante de cada plano.

Com uma interface simples e objetiva, o sistema ajuda a manter o controle básico sem complicações, oferecendo uma visão clara sobre quais alunos estão ativos ou inativos, qual plano escolheram e quando esse plano vai vencer. Tudo isso contribui para um atendimento mais ágil e uma gestão mais eficiente.

Funcionalidades principais:

- Cadastro de Alunos: Registro com informações pessoais e dados de contato.
- Ficha de Treino: Criação de treinos de acordo com o objetivo do aluno.
- Matrículas: Visualização de status (ativa, inativa, vencida), com cálculo de tempo restante.
- Relatórios Simples: Como lista de alunos ativos/inativos e planos mais utilizados.
- Gerenciamento de pagamentos manuais.

A proposta é oferecer uma solução prática para academias que querem sair do papel ou das planilhas, mas sem depender de sistemas complicados ou caros.O foco está na simplicidade, no controle básico e na melhoria do atendimento, sem prometer automações que ainda não fazem parte do escopo — tudo dentro da realidade da maioria das academias de pequeno e médio porte

Considerações finais

O objetivo principal deste trabalho foi desenvolver uma solução digital voltada para a gestão de academias, trazendo mais organização, controle e agilidade para o dia a dia tanto dos responsáveis quanto dos alunos. A proposta nasceu da necessidade real de melhorar processos como o cadastro de clientes, o controle de

matrículas e pagamentos, além da criação de fichas de treino personalizadas — tarefas que, na prática, muitas vezes ainda são feitas no papel ou de forma desorganizada.

Durante o desenvolvimento, o sistema foi estruturado de forma a centralizar essas funções em um só lugar, com lógica de negócio clara, divisão de responsabilidades entre camadas (Controller, Service, Repository) e uso de boas práticas no código. Apesar do foco ter sido na parte administrativa, a ideia sempre foi criar uma base sólida, capaz de crescer e se adaptar com o tempo.

Algumas funcionalidades importantes, como o login com autenticação de usuários, o painel exclusivo para alunos acompanharem seus treinos e até a integração com sistemas de catraca, ficaram fora do escopo inicial, mas são vistas como evoluções naturais do projeto. Também se imagina, para o futuro, o uso de dashboards com gráficos, envio de notificações via app ou e-mail, e até uma interface mais amigável para celular, pensando na rotina do aluno moderno.

Essa experiência foi mais do que codar: foi entender a real necessidade de um negócio, aplicar conhecimento técnico em algo prático e, principalmente, aprender a lidar com decisões e limitações de um projeto de verdade. O sistema pode ainda não estar completo, mas ele representa um passo importante na digitalização da rotina de academias, abrindo portas para soluções mais eficientes, conectadas e alinhadas com o que o público espera hoje.

Este projeto não apenas atendeu aos objetivos iniciais, mas também abriu caminho para futuras melhorias que podem ser implementadas para aprimorar ainda mais a experiência dos usuários e a eficiência operacional da academia. Com a base sólida estabelecida, há um vasto potencial para expandir as funcionalidades dosistema, tornando-o uma ferramenta ainda mais robusta e adaptável às necessidades em constante evolução do setor fitness.

Referências

ALFLEN, T.; PRADO, T. Técnicas de elicitação de requisitos no desenvolvimento de software: uma revisão sistemática da literatura. AtoZ: novas práticas em informação e conhecimento, 2021.

BALIEIRO, A. F.; PINTO, G. S. Importância do levantamento de requisitos no desenvolvimento de softwares. Faculdade de Tecnologia de Taquaritinga (Fatec), 2024.

CARVALHO, B. D. S. Boas práticas para elicitação de requisitos. Revista Ada Lovelace, 2018.

CHIAVENATO, Idalberto. Administração: Teoria, Processo e Prática. 3. ed. Rio de Janeiro: Elsevier, 2004.

COCKBURN, Alistair. Writing Effective Use Cases. Addison-Wesley, 2001.

COUTO, D. A importância da engenharia de requisitos. eduCAPES, 2023.

DBEAVER CORP. DBeaver Community Edition. Versão 23.2. Disponível em: https://dbeaver.io/. Acesso em: 8/5/2025

ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 7. ed. São Paulo: Pearson, 2019.

JETBRAINS. IntelliJ IDEA. Versão 2023.2. Disponível em: https://www.jetbrains.com/idea/. Acesso em: 8/5/2025

JGRAPH LTD. draw.io. Disponível em: https://www.drawio.com/. Acesso em: 8 maio 2025.

KONG INC. Insomnia. Versão 2023.5. Disponível em: https://insomnia.rest/. Acesso em: 8 maio 2025.

MAXIMIANO, Antonio Cesar Amaru. Introdução à Administração. 10. ed. São Paulo: Pearson, 2012.

META PLATFORMS, INC. React: A JavaScript library for building user interfaces. Versão 18.2.0. Disponível em: https://reactjs.org/. Acesso em: 8/5/2025.

MICROSOFT CORPORATION. Microsoft Excel. Versão 2024. Disponível em: https://www.microsoft.com/excel. Acesso em: 8/5/2025.

MICROSOFT CORPORATION. Microsoft Word. Versão 2024. Disponível em: https://www.microsoft.com. Acesso em: 8/5/2025.

MICROSOFT CORPORATION. Visual Studio Code. Versão 1.89. Disponível em: https://code.visualstudio.com/. Acesso em: 8/5/2025.

OBJECT MANAGEMENT GROUP (OMG). Business Process Model and Notation (BPMN) Version 2.0. 2011. Disponível em: https://www.omg.org/spec/BPMN/2.0. Acesso em: 8/5/2025.

ORACLE CORPORATION. Java Platform, Standard Edition. Versão 21. Disponível em: https://www.oracle.com/java/technologies/javase-downloads.html. Acesso em: 8/5/2025.

OSTERWALDER, A.; PIGNEUR, Y. Business Model Generation: Inovação em Modelos de Negócios. Rio de Janeiro: Alta Books, 2010.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7ª ed. McGraw Hill, 2011.SOMMERVILLE, Ian. Engenharia de Software. 9ª ed. Pearson, 2011.

APÊNDICE A - MISSÃO, VISÃO E VALORES

Missão: Proporcionar um ambiente acolhedor e acessível, promovendo a saúde e o bem-estar de nossos membros por meio de treinamentos personalizados, incentivando a prática regular de exercícios físicos e construindo uma comunidade saudável.

Visão: Ser reconhecida como a academia referência em nosso bairro, comprometida em oferecer serviços de qualidade, criar vínculos sólidos com os membros e contribuir para a melhoria da qualidade de vida da comunidade local.

Valor: Fomentar um ambiente amigável e inclusivo, promovendo o senso de comunidade entre nossos membros, priorizar a saúde e o bem-estar, proporcionando programas de treinamento seguros e eficazes.

APÊNDICE B – ENTREVISTA

Questionário ao Cliente “plusfit”

1. Quais os principais desafios enfrentados na administração da academia?
R: Os principais desafios enfrentados são o gerenciamento financeiro, controle de fluxo de caixa, fechamento mensal.

2. Como você gerencia as inscrições e os pagamentos dos clientes atualmente?
R: O gerenciamento de inscrições e pagamentos são por meio de anotações no caderno.

3. Quantos funcionários atuam na academia?
R: 3 funcionários.

4. Como é feita a comunicação aos clientes e funcionários de informações importantes, como mudanças de horário e promoções?R: A comunicação é realizada por meio de comunicação verbal, mensagens via whatsapp e publicações no instagram.

5. Existe um controle de entrada e saída de clientes na academia?
R: O controle de entrada e saída de clientes é por meio de uma catraca.

6. Como você coordena os treinos de cada cliente?
R: Por meio de uma elaboração de um plano de treino, descritos em fichas individuais para cada cliente.

7. Quais recursos e funcionalidades você gostaria de ver no software de gestão de Clientes? Prefere algo mais simples ou com mais funcionalidades?
R: Simples e de fácil manuseio, como por exemplo: Dados do cliente, mensalidades pagas e a vencer, faturamento mensal e reconhecimento facial.

8. Como você lida com cancelamentos e reembolsos?
R: De forma que fique bom para o cliente e para a academia, sem constrangimentos.

9. Existe algum programa de fidelidade para fidelização de clientes?
R: No momento não.

10. Quantos clientes em média vão ao estabelecimento por dia e quais os horários de pico?
R: Média de 250 clientes. Horário de pico das 18h às 20:30h

11. Como você lida com a segurança e a privacidade dos dados dos clientes no sistema?
R: Com acessos restritos.

12. Você já viu alguma funcionalidade em algum outro software de academia que você gostaria de ter nesse novo software
R: Reconhecimento facial.

13. Qual a função de cada funcionário na academia?
R: Recepcionar os clientes, promover treinos individualizados, orientar e acompanhar os clientes durante a pratica de exercícios físicos.

14. Como é feita a personalização dos treinos para cada cliente? E com qual frequência isso acontece?
R: Realizada de forma individualizada de acordo com as necessidades de cada cliente por meio de fichas individuais a cada 3 meses.

15. Você tem algum canal para recebimento de feedbacks e reclamações dos membros?
R: Não.

16. Como vocês lidam com clientes inativos ou que não vão regularmente?
R: Atualmente não possuímos um controle exato de clientes inativos ou que não vão regularmente.