

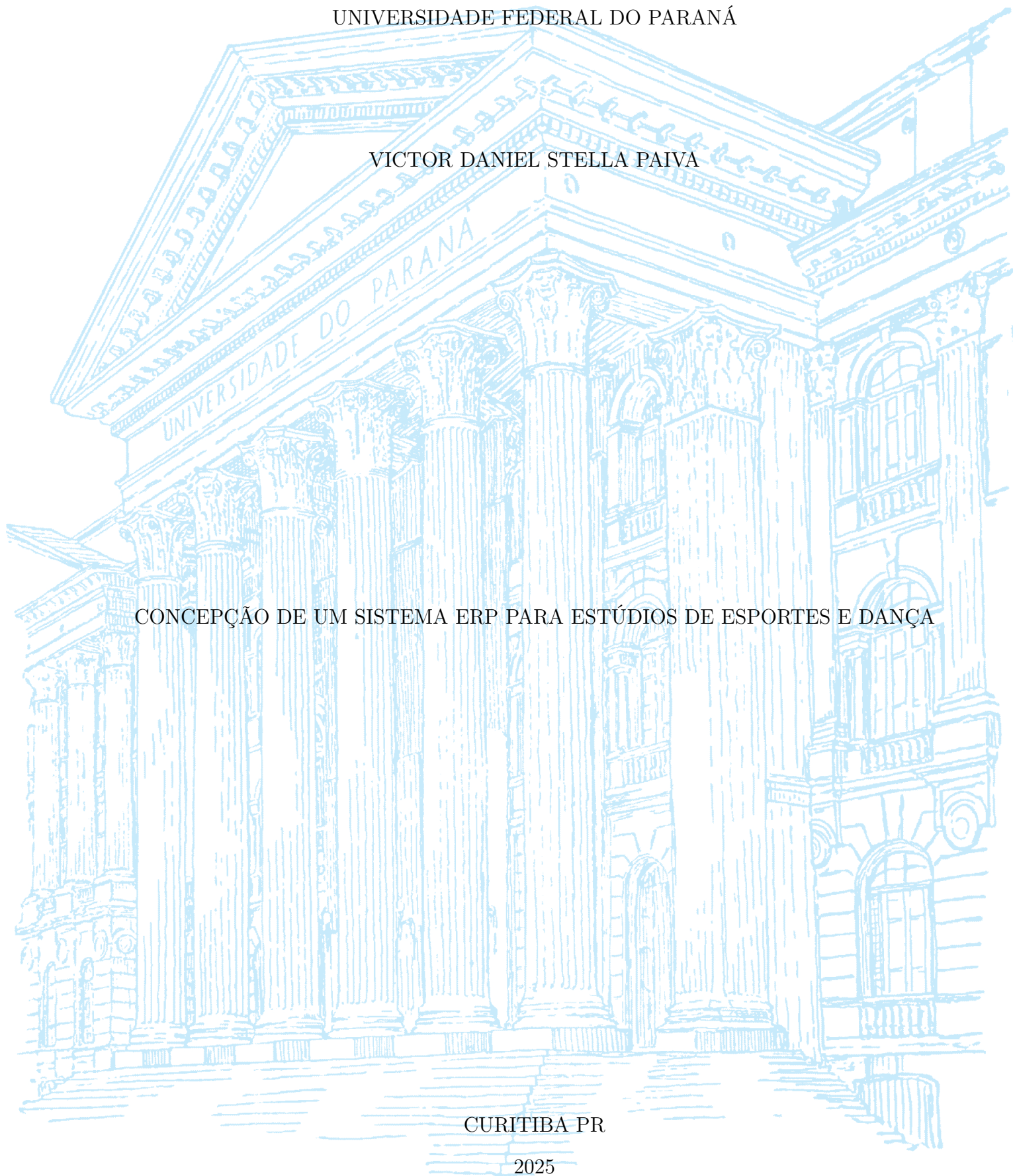
UNIVERSIDADE FEDERAL DO PARANÁ

VICTOR DANIEL STELLA PAIVA

CONCEPÇÃO DE UM SISTEMA ERP PARA ESTÚDIOS DE ESPORTES E DANÇA

CURITIBA PR

2025



VICTOR DANIEL STELLA PAIVA

CONCEPÇÃO DE UM SISTEMA ERP PARA ESTÚDIOS DE ESPORTES E DANÇA

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dra. Leticia Mara Peres.

CURITIBA PR

2025

*Aos meus pais, pelo amor, apoio e por nunca medirem esforços para que eu realizasse meus sonhos. Essa conquista também é de vocês.*

## AGRADECIMENTOS

Agradeço, antes de tudo, aos meus pais, por todo amor, paciência e apoio incondicional. Por nunca medirem esforços para que eu chegasse até aqui, por cada palavra de incentivo e por cada gesto silencioso que me deu forças nos momentos em que mais precisei. Tudo o que conquistei até hoje é reflexo do exemplo de dedicação, honestidade e perseverança que sempre encontrei em vocês. Este trabalho, mais do que uma etapa acadêmica, é também o resultado do amor e das oportunidades que vocês me proporcionaram ao longo da vida.

Agradeço também aos meus professores e orientadores, que compartilharam comigo seus conhecimentos e experiências com tanta generosidade. Cada aula, cada orientação e cada conversa contribuíram não apenas para a construção deste trabalho, mas também para o meu crescimento pessoal e profissional. Em especial, agradeço à professora Dra. Leticia Mara Peres, pela paciência, orientação e confiança depositada em meu potencial durante todo o desenvolvimento do projeto.

Aos colegas de curso, que tornaram essa jornada mais leve e significativa, deixo meu reconhecimento pela parceria, pelas trocas de ideias, pelas risadas e pelo companheirismo nos momentos de desafio. Compartilhar essa caminhada com vocês foi essencial para tornar o percurso acadêmico mais rico e humano.

Por fim, agradeço a mim mesmo, pela persistência, pela coragem de enfrentar os desafios e por não desistir diante das dificuldades. Este trabalho representa não apenas uma conclusão acadêmica, mas também o resultado de uma trajetória construída com esforço, aprendizado e gratidão.

## RESUMO

Este trabalho apresenta as etapas iniciais do processo de desenvolvimento de um sistema ERP (*Enterprise Resource Planning*) voltado ao nicho de estúdios de esportes e danças, abrangendo modalidades como pole dance, artes marciais e atividades similares. O estudo tem como foco o levantamento de requisitos, a análise do modelo de negócio e o design inicial do sistema, com o objetivo de estabelecer uma base sólida para o desenvolvimento futuro de uma solução completa de gestão voltada a esse segmento de mercado.

O levantamento de requisitos foi conduzido por meio de dois questionários estruturados, direcionados a donos de estabelecimentos e instrutores, bem como a alunos praticantes dessas atividades. Ainda, o projeto conta com a participação direta e orientação de uma profissional especialista da área, atuante como instrutora de pole dance, cuja participação foi de relevância estratégica imprescindível para a definição das necessidades reais do público-alvo, elicitação das funcionalidades propostas e das etapas futuras de testes e validações.

A fundamentação teórica do trabalho foi elaborada de forma a sustentar as práticas e técnicas de Engenharia de Software empregadas ao longo do projeto, oferecendo o embasamento conceitual necessário para a escolha dos modelos de processo, metodologias, técnicas de elicitação de requisitos e princípios de design adotados. Com base nesses referenciais, decisões de projeto foram tomadas, tecnologias mais adequadas ao contexto da solução foram selecionadas e outras foram sugeridas, garantindo coerência técnica e consistência metodológica durante o planejamento e a modelagem do sistema.

Os resultados do trabalho incluem a apresentação dos questionários aplicados, dos BMCs (*Business Model Canvas*), dos diagramas UML (*Unified Modeling Language*), desenvolvidos de forma a preservar a anonimização dos detalhes técnicos e do plano de desenvolvimento detalhado, que servirá como guia para as etapas subsequentes do projeto. Esse plano prevê a implementação de um MVP (*Minimum Viable Product*), a ser futuramente testado em um cliente voluntário, com o propósito de coletar *feedback* e orientar o processo de refinamento e correção de problemas antes de uma versão completa.

Dessa forma, o trabalho consolida as fases iniciais de levantamento de requisitos, análise e design do sistema, delimitando um conjunto de requisitos e diretrizes técnicas que servirão de alicerce para o desenvolvimento futuro, mantendo a coerência com os princípios de engenharia de software e com as necessidades específicas do público-alvo identificado.

Palavras-chave: ERP. Software para Gestão. Software para Esportes. Software para Danças. Software para Estúdios. Software para Pole Dance. Software Web. Automação de Processos.

## ABSTRACT

This work presents the initial stages of the development process of an ERP (Enterprise Resource Planning) system designed for the niche of sports and dance studios, encompassing modalities such as pole dance, martial arts, and similar activities. The study focuses on requirements elicitation, business model analysis, and the initial design of the system, with the goal of establishing a solid foundation for the future development of a complete management solution tailored to this market segment.

The requirements elicitation process was conducted through two structured questionnaires—one directed at studio owners and instructors, and another aimed at students who practice these activities. In addition, the project includes the direct participation and guidance of an industry specialist, who works as a pole dance instructor. Her contribution was of strategic importance for identifying the real needs of the target audience, defining the proposed functionalities, and outlining the future stages of testing and validation.

The theoretical foundation of this work was developed to support the Software Engineering practices and techniques applied throughout the project, providing the conceptual basis necessary for selecting process models, methodologies, requirements elicitation techniques, and design principles. Based on these theoretical references, project decisions were made, the most suitable technologies for the proposed solution were selected, and others were suggested, ensuring technical coherence and methodological consistency during the system's planning and modeling stages.

The results of this study include the presentation of the applied questionnaires, the Business Model Canvas (BMCs), and the Unified Modeling Language (UML) diagrams—developed to preserve the anonymity of technical details—along with a detailed development plan that will serve as a guide for the subsequent stages of the project. This plan foresees the implementation of a Minimum Viable Product (MVP), to be later tested with a volunteer client, with the purpose of collecting feedback and guiding the refinement and correction process prior to a full version release.

In this way, the work consolidates the initial phases of requirements elicitation, analysis, and partial system design, outlining a set of requirements and technical guidelines that will serve as a foundation for future development, maintaining consistency with software engineering principles and the specific needs of the identified target audience.

Keywords: ERP. Management Software. Sports Software. Dance Software. Studio Software. Pole Dance Software. Web Software. Process Automation.

## LISTA DE FIGURAS

3.1	<i>Business Model Canvas</i> aplicado ao <i>stakeholder</i> . . . . .	25
3.2	<i>Business Model Canvas</i> aplicado à solução proposta. . . . .	26
3.3	Diagrama de Casos de Uso do sistema (parte um).. . . . .	28
3.4	Diagrama de Casos de Uso do sistema (parte dois). . . . .	29
3.5	Diagrama de Classes do sistema.. . . . .	30
3.6	Diagrama de Sequência - Processo de Matrícula.. . . . .	31
3.7	Diagrama de Sequência - Processo de Mudança de Turma ou Reagendamento de Aula.. . . . .	32
3.8	Diagrama de Sequência - Registro de Presença e Evolução dos Alunos. . . . .	33
3.9	Diagrama de Sequência - Cadastro e Gerenciamento de Instrutores. . . . .	33
3.10	Diagrama de Sequência - Pagamento Mensal de Aluno. . . . .	34
3.11	Diagrama de Sequência - Emissão de Relatórios Financeiros e Operacionais. . . . .	34
3.12	Diagrama de Sequência - Compartilhamento de Materiais de Apoio.. . . . .	35
A.1	Respostas à questão dois do questionário aplicado aos administradores e instrutores. . . . .	45
A.2	Respostas à questão quatro do questionário aplicado aos administradores e instrutores. . . . .	46
A.3	Respostas à questão sete do questionário aplicado aos administradores e instrutores. . . . .	47
A.4	Respostas à questão nove do questionário aplicado aos administradores e instrutores. . . . .	47
B.1	Respostas à questão um do questionário aplicado aos alunos praticantes. . . . .	49
B.2	Respostas à questão dois do questionário aplicado aos alunos praticantes. . . . .	50
B.3	Respostas à questão três do questionário aplicado aos alunos praticantes.. . . . .	50
B.4	Respostas à questão cinco do questionário aplicado aos alunos praticantes. . . . .	51
B.5	Respostas à questão seis do questionário aplicado aos alunos praticantes. . . . .	51

## LISTA DE ACRÔNIMOS

ERP	Enterprise Resource Planning
MVP	Minimum Viable Product
LGPD	Lei Geral de Proteção de Dados
UML	Unified Modeling Language
BMC	Business Model Canvas
MVC	Model-View-Controller
SQL	Structured Query Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA E METODOLOGIA</b>	<b>12</b>
2.1	ENGENHARIA DE SOFTWARE E CICLO DE VIDA DE DESENVOLVIMENTO	12
2.2	LEVANTAMENTO DE REQUISITOS	13
2.2.1	Análise do Modelo de Negócio	13
2.2.2	<i>Business Model Canvas</i>	14
2.3	MODELAGEM DO SISTEMA	15
2.3.1	Diagrama de Casos de Uso	16
2.3.2	Diagrama de Classes	16
2.3.3	Diagrama de Sequência	17
2.4	ARQUITETURA E DESIGN DO SISTEMA	18
2.5	TECNOLOGIAS UTILIZADAS E SUGERIDAS	19
2.6	PLANO DE DESENVOLVIMENTO	20
2.7	VALIDAÇÃO E REFINAMENTO	21
2.8	CONCLUSÃO	22
<b>3</b>	<b>RESULTADOS</b>	<b>24</b>
3.1	QUESTIONÁRIOS	24
3.2	<i>BUSINESS MODEL CANVAS</i>	24
3.3	MODELAGEM	27
3.3.1	Diagrama de Casos de Uso	27
3.3.2	Diagrama de Classes	30
3.3.3	Diagramas de Sequência	30
3.4	TECNOLOGIAS SELECIONADAS	35
3.5	PLANO DE DESENVOLVIMENTO	36
3.5.1	Incremento 1 - Configuração do Ambiente e Arquitetura Base	36
3.5.2	Incremento 2 - Módulo de Autenticação e Controle de Acesso	37
3.5.3	Incremento 3 - Cadastro de Empresa, Instrutores e Alunos	37
3.5.4	Incremento 4 - Módulo de Turmas e Agendamentos	37
3.5.5	Incremento 5 - Registro de Presença e Visualização de Horários	38
3.5.6	Incremento 6 - Módulo Financeiro Básico	38
3.5.7	Incremento 7 - Módulo de Materiais de Apoio	38
3.5.8	Incremento 8 - Testes e Refinamento do MVP	38

3.6	IMPACTO SOCIAL . . . . .	39
3.7	CONCLUSÃO . . . . .	40
4	CONSIDERAÇÕES FINAIS . . . . .	41
	REFERÊNCIAS . . . . .	42
	APÊNDICE A – QUESTIONÁRIO AOS ADMINISTRADORES E INSTRUTORES . . . . .	45
	APÊNDICE B – QUESTIONÁRIO AOS ALUNOS. . . . .	49

## 1 INTRODUÇÃO

De acordo com Castells (2006), vivemos em uma sociedade em rede, na qual a informação e o conhecimento assumem papel determinante na estrutura e no desempenho das organizações. Nesse contexto, a tecnologia pode atuar como um fator democratizador, ao ampliar o acesso a ferramentas e infraestruturas antes restritas a grandes corporações, permitindo que negócios de pequeno porte também inovem e otimizem seus processos.

Um levantamento recente apresentado pelo SEBRAE (2025) indicou que setenta e seis por cento dos pequenos negócios utilizam computadores em suas atividades, um crescimento de seis pontos percentuais em comparação com dois mil e vinte e dois, e que quarenta e sete por cento dos empreendimentos fazem uso de aplicativos ou softwares integrados para gestão, o que representa um aumento de vinte pontos percentuais desde dois mil e dezoito, evidenciando uma tendência de avanço na digitalização ao longo dos últimos anos. Esses dados mostram um movimento relevante de incorporação de tecnologia pelos pequenos negócios, mas também revelam que, apesar de grande parte dos pequenos empreendedores brasileiros já contarem com acesso à internet e ferramentas digitais básicas, a adoção de soluções integradas de gestão ainda é limitada.

A ausência de sistemas adaptados ao nicho observado neste trabalho tende a resultar em ineficiência administrativa, erros manuais, dificuldade de controle financeiro e falhas de comunicação entre instrutores, alunos e gestores. Nesse cenário, os sistemas ERP (*Enterprise Resource Planning*) destacam-se como uma alternativa promissora, pois permitem centralizar informações e automatizar tarefas essenciais, como controle de matrículas, agendamento de aulas, gestão de pagamentos e acompanhamento de desempenho.

Assim, este trabalho propõe as etapas iniciais de levantamento de requisitos, análise do modelo de negócios e design de um sistema ERP web voltado para estúdios e escolas de esportes e dança, incluindo modalidades como pole dance, artes marciais, ginástica, danças no geral e esportes diversos. O objetivo é construir uma base sólida para o futuro desenvolvimento de uma plataforma que ofereça funcionalidades de gestão de turmas, alunos, instrutores e fluxo financeiro, contribuindo para o aumento da eficiência operacional.

A metodologia do projeto contempla o levantamento de requisitos realizado por meio de dois questionários estruturados: um direcionado a instrutores e proprietários de estabelecimentos, e outro a alunos praticantes. Além disso, o trabalho conta com uma análise do modelo de negócio, uma modelagem de sistema e um plano de desenvolvimento. Ainda, o projeto conta com a participação, na forma de consultoria, de uma profissional especialista da área, atuante como instrutora de pole dance no cliente voluntário (*stakeholder* que contribuirá para as etapas de validação e refinamento), cuja participação foi essencial para a validação das necessidades reais do público-alvo e para o direcionamento estratégico das funcionalidades do sistema. Esses instrumentos de coleta de informações possibilitaram uma compreensão mais ampla sobre as dificuldades enfrentadas no cotidiano dos estúdios, permitindo que o design do sistema reflita de maneira mais precisa o contexto e as prioridades dos usuários.

O trabalho também apresenta uma fundamentação teórica que sugere decisões metodológicas e técnicas a serem tomadas ao longo do ciclo de vida de desenvolvimento, abordando modelos de engenharia de software, metodologias ágeis, técnicas de elicitação

de requisitos e princípios de design de sistemas. Como afirma Pressman e Maxim (2016), o sucesso de um projeto de software está fortemente relacionado à qualidade do processo de engenharia, que deve garantir uma base conceitual sólida antes do início do desenvolvimento.

Também, o trabalho inclui diagramas UML (*Unified Modeling Language*) simplificados como parte dos seus resultados, representando as principais funcionalidades e interações do sistema, mantendo a anonimização dos detalhes técnicos por questões de confidencialidade. Também, foi elaborado um plano de desenvolvimento que orientará a implementação do MVP (*Minimum Viable Product*) a ser implantado futuramente no cliente voluntário, possibilitando a coleta de *feedback* a partir de um caso de uso real e a identificação de correções e melhorias a serem feitas.

Além dos possíveis benefícios técnicos relacionados à automatização e à integração de processos, a solução proposta pode gerar efeitos sociais relevantes, especialmente ao ampliar o acesso a ferramentas de gestão e apoiar a inclusão digital de pequenos empreendedores. Ao facilitar a organização e o controle das atividades, a plataforma tende a contribuir para ganhos de eficiência e produtividade, além de potencialmente favorecer a troca de experiências e o compartilhamento de conhecimento entre membros da comunidade atendida. De forma complementar, o uso responsável de tecnologia pode estimular práticas voltadas à sustentabilidade, segurança da informação e responsabilidade social, reforçando o potencial do sistema como instrumento de apoio ao desenvolvimento local e à redução de desigualdades no acesso digital.

Em sua essência, este projeto busca estabelecer bases conceituais e estruturais, que orientarão as etapas subseqüentes do desenvolvimento de uma ferramenta web. Seu propósito, é estabelecer as bases para o desenvolvimento de uma solução acessível a empreendedores atuantes no nicho, contribuindo para a inclusão digital dessas empresas e, potencialmente, favorecendo sua profissionalização, organização, crescimento sustentável e maior eficiência na gestão de processos e no uso de recursos. De acordo com Castells (2006), ao democratizar o acesso a tecnologias de gestão, iniciativas como esta podem fortalecer a eficiência, a competitividade e a sustentabilidade do mercado, alinhando-se ao papel transformador da tecnologia na sociedade contemporânea.

Nos capítulos a seguir, serão apresentadas a fundamentação teórica do trabalho, as decisões técnicas e estratégicas relacionadas ao desenvolvimento futuro do sistema, os resultados obtidos até o momento e os esperados para as próximas etapas, finalizando com uma discussão sobre o impacto da solução proposta sob as perspectivas técnica e social. Este trabalho foi elaborado utilizando o modelo de dissertação/tese do PPGInf/UFPR disponibilizado por Maziero (2025).

## 2 FUNDAMENTAÇÃO TEÓRICA E METODOLOGIA

A concepção de uma solução escalável requer a aplicação dos conceitos fundamentais da Engenharia de Software, disciplina que reúne princípios teóricos e práticas metodológicas, voltados à transformação de necessidades de negócio em soluções tecnológicas, funcionais e sustentáveis. Este capítulo apresenta a fundamentação teórica que orienta o desenvolvimento do sistema proposto, bem como a metodologia aplicada em cada uma de suas etapas, buscando assegurar a coerência entre os objetivos do projeto e as práticas recomendadas da área.

Inicialmente, são discutidos os princípios da Engenharia de Software e o ciclo de vida de desenvolvimento de sistemas, contextualizando a escolha do modelo incremental com metodologias ágeis, adotado como base para o planejamento e a organização das atividades. Em seguida, são apresentados fundamentos do processo de levantamento de requisitos, da análise do modelo de negócio e a ferramenta usada *Business Model Canvas*, da etapa de modelagem do sistema e a ferramenta usada UML (*Unified Modeling Language*). Depois, princípios e técnicas de design de software são discutidos, com ênfase na seleção de tecnologias, ferramentas e arquiteturas adequadas ao contexto do projeto.

Na sequência, são apresentadas as estratégias iniciais e discussões sobre a futura etapa validação e refinamento da solução, considerando um MVP em ambiente real, com o objetivo de obter feedback e promover melhorias contínuas. Então, discute-se os resultados obtidos nesse trabalho, ou seja, os questionários aplicados e suas respostas, os BMCs, os diagramas UML e o plano de desenvolvimento, comentando, por fim, sobre os impactos tecnológicos e sociais da solução. A última seção desse capítulo apresenta uma conclusão parcial, que sintetiza as observações teóricas e metodológicas que embasam as próximas etapas do trabalho.

### 2.1 ENGENHARIA DE SOFTWARE E CICLO DE VIDA DE DESENVOLVIMENTO

A Engenharia de Software, segundo Sommerville (2011), consiste em uma abordagem sistemática e disciplinada que busca garantir a qualidade, confiabilidade e eficiência das soluções produzidas, por meio de métodos, ferramentas e boas práticas consolidadas.

De acordo com o mesmo autor, um projeto de software deve seguir um ciclo de vida de desenvolvimento, que representa a sequência organizada de atividades necessárias para transformar uma necessidade ou ideia em um produto de software operacional. Esse ciclo normalmente inclui etapas bem definidas, como levantamento de requisitos, design, implementação, integração, testes e manutenção, que podem ser estruturadas de diferentes maneiras, conforme o modelo de processo adotado.

No contexto deste trabalho, adotou-se o modelo incremental aliado a metodologias ágeis como base conceitual para a elaboração do plano de desenvolvimento e para a execução do projeto. Inicialmente, o intuito é desenvolver uma versão MVP, com base no plano criado, contendo as funcionalidades essenciais para demonstrar o funcionamento geral da solução. Nessa fase, o modelo incremental será utilizado principalmente como método de organização e controle interno das etapas, estruturando o progresso em *sprints* e iterações planejadas, mas sem entregas contínuas ou ciclos de feedback externos.

Somente após a conclusão do MVP e a realização das etapas de testes e coleta de feedback com o cliente voluntário, o projeto passará a seguir efetivamente o modelo

incremental em sua forma completa, permitindo que o sistema evolua de modo iterativo e incremental, com novas versões e funcionalidades sendo implementadas e aprimoradas com base nos resultados e nas necessidades identificadas.

## 2.2 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma etapa importante no processo de Engenharia de Software, responsável por identificar, compreender e documentar as necessidades, expectativas e restrições dos *stakeholders* envolvidos em um projeto. Segundo Sommerville (2011), essa fase busca traduzir as demandas do usuário em requisitos claros e verificáveis, constituindo a base sobre a qual o sistema poderá ser projetado e desenvolvido. De forma semelhante, Pressman e Maxim (2016) destacam que a eficácia de um projeto de software está diretamente relacionada à qualidade do processo de elicitación, que deve empregar métodos sistemáticos de coleta, análise e validação de informações.

No presente trabalho, adotou-se uma abordagem mista para o levantamento de requisitos, combinando entrevistas confidenciais com a profissional especialista da área, questionários estruturados aplicados a administradores, instrutores e alunos do cliente voluntário e a análise do modelo de negócio representada por meio dos *Business Model Canvas*. Essa combinação de métodos ajudou a integrar a perspectiva prática dos usuários com uma visão sistêmica do funcionamento do setor.

Os questionários foram elaborados levando em consideração a importância da objetividade e padronização na coleta de informações, conforme descrito por Sommerville (2011). As perguntas foram planejadas para identificar o interesse em informatizar atividades administrativas recorrentes, compreender o nível de adoção tecnológica, reconhecer dificuldades operacionais e levantar expectativas sobre funcionalidades desejadas. Segundo Pressman e Maxim (2016), a aplicação de questionários é uma técnica eficaz de levantamento de requisitos, pois permite reunir dados de forma padronizada, reduzindo vieses interpretativos e facilitando a identificação de padrões de comportamento e prioridade.

No âmbito deste projeto, os requisitos levantados foram classificados, à partir de Sommerville (2011), como funcionais — aqueles que descrevem o que o sistema deve fazer, e não funcionais — relacionados a como o sistema deve se comportar em termos de desempenho, segurança e usabilidade. As informações obtidas nessa etapa fornecem subsídios diretos para a modelagem e o design do sistema, influenciando também a escolha de ferramentas e tecnologias a serem empregadas, além de servirem como referência para validação e refinamento em etapas posteriores de desenvolvimento.

Os métodos e resultados do levantamento de requisitos estão documentados no Apêndice A e Apêndice B, que reúnem, respectivamente, os questionários aplicados a administradores e instrutores e a alunos praticantes, acompanhados das respostas obtidas. Esses instrumentos tiveram como propósito compreender o contexto de uso, avaliar a viabilidade e relevância da solução proposta e estabelecer uma base sólida para a definição dos requisitos e da arquitetura conceitual do sistema.

### 2.2.1 Análise do Modelo de Negócio

Antes de iniciar o desenvolvimento de um sistema, é importante compreender o modelo de negócio do setor a que ele se destina. De acordo com Osterwalder e Pigneur (2010), a análise do modelo de negócio consiste em identificar como uma organização cria, entrega e captura valor, mapeando seus principais processos, recursos, atores e fluxos

de informação. Essa compreensão é fundamental para garantir que o sistema atenda às necessidades reais dos usuários e esteja alinhado aos objetivos estratégicos do negócio. Conforme ressaltam Osterwalder et al. (2014), a análise do modelo de negócio ajuda a identificar oportunidades de inovação, otimização de processos e diferenciação competitiva, contribuindo para o sucesso do projeto de software.

Para essa análise, foram considerados os resultados dos questionários estruturados aplicados aos usuários, além da consultoria de uma profissional especialista da área. Essa abordagem metodológica possibilitou compreender, de forma sistemática, as práticas, desafios e oportunidades existentes no contexto estudado.

Com base nas informações obtidas, foi realizada uma análise do modelo de negócio do *stakeholder* e da solução sugerida, a fim de sintetizar as informações coletadas e compreender de forma integrada seus processos, recursos e fluxos operacionais. Os resultados dessa análise forneceram subsídios diretos para o levantamento de requisitos e serviram como base conceitual para as etapas seguintes.

### 2.2.2 *Business Model Canvas*

De acordo com Osterwalder e Pigneur (2010), o *Business Model Canvas* é uma ferramenta de planejamento e análise estratégica que permite visualizar e estruturar, de forma integrada, os principais elementos que compõem um modelo de negócio. Proposto originalmente por Alex Osterwalder em sua tese de doutorado e posteriormente desenvolvido em colaboração com Yves Pigneur e mais de duzentos especialistas ao redor do mundo, o BMC organiza a lógica de funcionamento de uma organização em nove blocos fundamentais: segmentos de clientes, propostas de valor, canais, relacionamento com clientes, fontes de receita, recursos-chave, atividades-chave, parcerias-chave e estrutura de custos.

A seguir, a explicação de cada parte do *canvas*:

- **Proposta de Valor:** Define o que a empresa oferece de relevante aos seus clientes: o problema que resolve ou a necessidade que atende. Representa o núcleo do modelo de negócio.
- **Segmentos de Clientes:** Identifica os diferentes grupos de pessoas ou organizações que a empresa busca atender, reconhecendo suas características e necessidades específicas.
- **Canais:** Descreve como a empresa se comunica e entrega sua proposta de valor aos clientes, incluindo canais de marketing, distribuição e venda.
- **Relacionamento com Clientes:** Explica o tipo de relacionamento que a empresa estabelece com cada segmento, podendo variar entre atendimento personalizado, automação, autoatendimento, comunidades, entre outros.
- **Fontes de Receita:** Indica como o negócio gera receita a partir de cada segmento de clientes, seja por vendas diretas, assinaturas, licenças, comissões, etc.
- **Recursos-Chave:** Engloba os principais ativos físicos, humanos, intelectuais e financeiros necessários para criar e entregar a proposta de valor.
- **Atividades-Chave:** São as ações essenciais que a empresa precisa realizar para operar com sucesso, como produção, marketing, suporte técnico, desenvolvimento ou gestão.

- **Parcerias-Chave:** Identifica os parceiros externos, fornecedores e alianças estratégicas que ajudam a empresa a executar suas atividades e reduzir riscos.
- **Estrutura de Custos:** Resume os principais custos envolvidos na operação do modelo de negócio, incluindo despesas fixas, variáveis, de pessoal e infraestrutura.

No contexto deste trabalho, o BMC foi utilizado de maneira complementar em duas perspectivas distintas. O primeiro modelo (Figura 3.1) foi aplicado ao *stakeholder* principal (cliente voluntário), com o objetivo de melhor compreender o funcionamento do negócio, seus fluxos operacionais e os principais pontos de melhoria que justificam a solução proposta. A construção desse *canvas* contou com o suporte das entrevistas, questionários e da consultoria da profissional especialista da área, que contribuíram de forma imprescindível para garantir que o modelo refletisse a realidade do setor de forma fiel e fundamentada, servindo como base sólida para o levantamento de requisitos funcionais e não funcionais do sistema.

O segundo *canvas* (Figura 3.2) foi elaborado para representar o modelo de negócio da plataforma proposta, descrevendo como a solução cria, entrega e captura valor dentro do ecossistema identificado. Ambos os BMCs foram desenvolvidos com o intuito de representar o contexto de maneira imparcial e realista, mas também sob um viés analítico voltado à concepção de uma solução eficiente e com potencial transformador. Essa abordagem ajudou a alinhar as funcionalidades técnicas e as decisões de design aos objetivos operacionais e estratégicos do projeto, assegurando coerência entre os aspectos tecnológicos e o contexto real de aplicação.

### 2.3 MODELAGEM DO SISTEMA

A modelagem de sistemas constitui uma prática amplamente recomendada na Engenharia de Software, por contribuir para o entendimento e a representação das abstrações que compõem uma aplicação antes de sua implementação. De acordo com Pressman e Maxim (2016), a modelagem auxilia no entendimento das necessidades e restrições do software, fornecendo representações gráficas que ajudam a reduzir a complexidade e a melhorar a comunicação entre os participantes do projeto. De forma semelhante, Sommerville (2011) ressalta que a modelagem não apenas documenta o comportamento esperado do sistema, mas também apoia a análise, validação e comunicação das especificações com os *stakeholders*, funcionando como um instrumento de verificação conceitual ainda nas fases iniciais do ciclo de vida do software.

A metodologia de modelagem adotada neste trabalho foi fundamentada em dados obtidos durante o levantamento de requisitos. Conforme já citado Wieggers e Beatty (2013), a combinação de técnicas qualitativas e quantitativas na elicitação de requisitos é recomendada para aumentar a completude e a confiabilidade das informações levantadas, permitindo que a modelagem subsequente reflita o contexto de uso e as necessidades reais dos usuários. Essa combinação de métodos possibilitou converter as informações coletadas em representações conceituais, que descrevem de forma abstrata as principais entidades, atores e interações do sistema analisado.

A modelagem foi conduzida com base na *Unified Modeling Language* (UML), linguagem padrão para especificação, visualização, construção e documentação de artefatos de software, criada por Grady Booch, James Rumbaugh e Ivar Jacobson na década de mil novecentos e noventa e posteriormente padronizada pelo Group (2024). Segundo Rumbaugh et al. (2005), a UML surgiu como um esforço de unificação de diferentes métodos

de modelagem orientada a objetos, consolidando uma notação capaz de representar de forma integrada as dimensões estrutural, comportamental e interativa de um sistema.

No contexto deste projeto, foram elaborados diferentes diagramas UML para representar as principais perspectivas do sistema proposto, em conformidade com as recomendações de Rumbaugh et al. (2005) e do Group (2024). Dessa forma, os modelos desenvolvidos têm como propósito organizar e representar graficamente as informações extraídas das etapas de análise, estruturando-as em um formato que favorece o raciocínio técnico e o planejamento das próximas etapas. As subseções a seguir apresentam os tipos de diagramas utilizados, descrevendo seus fundamentos conceituais e suas funções no processo de modelagem do sistema.

### 2.3.1 Diagrama de Casos de Uso

O diagrama de casos de uso permite descrever, de forma visual, as interações entre os atores externos e o sistema, representando as funcionalidades esperadas do ponto de vista do usuário. De acordo com Sommerville (2011), essa técnica fornece uma visão de alto nível sobre o que o software deve realizar, sem detalhar sua implementação, auxiliando na validação dos requisitos funcionais. Além disso, conforme destacam Pressman e Maxim (2016), a clareza visual desse tipo de diagrama facilita o diálogo entre as áreas técnica e de negócio, reduzindo ambiguidades e servindo como base para o refinamento das demais representações UML, como os diagramas de sequência e os diagramas de classes.

No contexto deste trabalho, o diagrama de casos de uso foi elaborado a partir da interpretação e síntese das informações obtidas nas etapas anteriores. Essa abordagem auxiliou na compreensão das principais atividades desempenhadas pelos usuários (administradores, gestores, instrutores e alunos) e na identificação das funcionalidades essenciais para o sistema. Conforme também destaca Wiegers e Beatty (2013), a modelagem baseada em dados coletados diretamente dos *stakeholders* tende a aumentar a precisão das representações, pois reflete necessidades observadas.

A relevância do diagrama de casos de uso neste projeto está relacionada à sua função de síntese e comunicação dos requisitos levantados, traduzindo as informações textuais coletadas em representações visuais acessíveis e verificáveis. Como observam Sommerville (2011) e Pressman e Maxim (2016), esse tipo de diagrama é especialmente útil para validar objetivos e limites do sistema antes do plano de desenvolvimento, garantindo que as expectativas dos usuários estejam devidamente representadas.

Por fim, convém observar que o diagrama de classes (??) apresentado no capítulo de Capítulo 3 é uma versão um pouco simplificada propositalmente de um modelo original mais completo. Essa redução foi feita por motivos de confidencialidade do sistema.

### 2.3.2 Diagrama de Classes

O diagrama de classes descreve a estrutura estática do sistema, representando suas principais entidades, atributos, métodos e os relacionamentos entre elas. De acordo com Pressman e Maxim (2016), esse tipo de diagrama é fundamental para o design conceitual de sistemas orientados a objetos, pois fornece uma base formal para a organização e comunicação dos elementos estruturais do software. De forma semelhante, Sommerville (2011) destaca que a modelagem por classes contribui para a compreensão da arquitetura e das dependências internas do sistema, auxiliando na análise e validação de sua consistência antes da implementação.

No contexto deste trabalho, o diagrama de classes foi elaborado a partir dos requisitos identificados e do Diagrama de Casos de Uso. Essa combinação metodológica ajudou a identificar as principais entidades que compõem o domínio do sistema e a compreender as relações lógicas e funcionais entre elas. Conforme já apontado e referenciado, a modelagem derivada diretamente de dados empíricos coletados junto aos usuários tende a produzir representações mais alinhadas às necessidades reais e à terminologia do domínio.

Cada classe foi definida considerando as responsabilidades e atributos julgados necessários ao funcionamento do sistema, bem como os métodos que podem representar comportamentos esperados em seu contexto de uso. Segundo ?, essa etapa é essencial para garantir encapsulamento e modularidade, princípios que favorecem a manutenibilidade e a escalabilidade do software. Assim, o diagrama desenvolvido fornece uma perspectiva mais estruturada do sistema proposto, servindo como base para as etapas posteriores de design detalhado e implementação.

A relevância do diagrama de classes neste projeto reside em sua função de integrar os resultados das etapas anteriores em uma representação formal e organizada da estrutura do sistema. Como destacam Pressman e Maxim (2016) e Sommerville (2011), essa representação é um instrumento essencial de comunicação entre analistas e desenvolvedores, assegurando que as decisões técnicas e de arquitetura sejam fundamentadas em um modelo coerente, verificável e alinhado ao domínio de aplicação.

Ainda, convém observar que o diagrama de classes (Figura 3.5) apresentado no capítulo de Capítulo 3 é uma versão propositalmente simplificada de um modelo original mais extenso. Essa redução foi feita por motivos de confidencialidade do sistema e para facilitar a compreensão dentro do espaço disponível no documento; o modelo completo contém detalhes e relações adicionais que foram omitidos aqui sem prejuízo da validade das decisões de projeto discutidas.

### 2.3.3 Diagrama de Sequência

O diagrama de sequência representa, de forma temporal, as interações entre objetos ou componentes de um sistema. Segundo Sommerville (2011), esse tipo de diagrama permite visualizar o comportamento dinâmico de um sistema ao longo do tempo, destacando a troca de mensagens entre os objetos participantes para a execução de uma funcionalidade específica. De forma semelhante, Pressman e Maxim (2016) observam que a representação sequencial dos eventos auxilia na compreensão dos fluxos de controle e comunicação dentro do sistema, contribuindo para a identificação de dependências e para a verificação da coerência entre a modelagem estática (estrutural) e a dinâmica.

Esses diagramas são particularmente úteis para detalhar cenários descritos nos casos de uso, mostrando como os atores e os componentes internos interagem para atingir um determinado objetivo. Conforme Rumbaugh et al. (2005), esse tipo de diagrama é eficaz para validar a lógica de processos e a integridade da comunicação entre objetos, pois torna explícito o encadeamento das mensagens trocadas, os pontos de decisão e as possíveis variações de fluxo.

A relevância dos diagramas de sequência neste projeto está associada à sua capacidade de integrar e consolidar diferentes aspectos da modelagem do sistema. Como ressaltam Pressman e Maxim (2016) e Sommerville (2011), esse tipo de representação oferece um meio eficiente de comunicação entre projetistas e *stakeholders*, auxiliando na validação das funcionalidades e garantindo que a sequência lógica dos processos esteja coerente com o que foi definido nos casos de uso e estruturado no diagrama de classes. No presente trabalho, portanto, os diagramas de sequência cumprem papel essencial na

verificação conceitual do sistema, assegurando a consistência entre os modelos e a clareza na representação das interações dinâmicas.

Também, convém observar que os sete diagramas de sequência (Figura 3.6) apresentados no capítulo de Capítulo 3 são uma versão propositalmente simplificadas de modelos originais mais completos, gerando mais diagramas de sequência. Essa redução foi feita por motivos de confidencialidade do sistema e para facilitar a compreensão dentro do espaço disponível no documento.

## 2.4 ARQUITETURA E DESIGN DO SISTEMA

De acordo com Sommerville (2011), a arquitetura representa um ponto de convergência entre os requisitos técnicos e de negócio, fornecendo uma visão que facilita a comunicação entre desenvolvedores, analistas e demais partes interessadas. Nessa mesma linha, Pressman e Maxim (2016) observam que a definição de uma arquitetura clara e coerente contribui para reduzir a complexidade do sistema, estabelecer padrões de desenvolvimento e orientar decisões futuras de design e manutenção.

No contexto deste projeto, a arquitetura foi definida com o propósito de oferecer uma estrutura que permita a evolução incremental do sistema e a integração organizada de novos módulos nas etapas subsequentes do trabalho. Assim, buscou-se adotar um modelo que favoreça a modularidade, a escalabilidade e a separação de responsabilidades, características essenciais para assegurar a coerência entre as fases de análise, design e implementação. Essa preocupação com a continuidade e a previsibilidade do desenvolvimento encontra respaldo em Pressman e Maxim (2016), que destacam a importância de arquiteturas planejadas de forma a reduzir riscos técnicos e facilitar o refinamento progressivo do software.

A organização geral do sistema segue o padrão arquitetural *Model-View-Controller* (MVC), amplamente empregado no desenvolvimento de aplicações web. Conforme descrevem Gamma et al. (1995) e Fowler (2004), o MVC divide o sistema em três camadas principais (Modelo, Visão e Controlador) promovendo a separação de responsabilidades entre a lógica de negócio, a interface com o usuário e o gerenciamento das interações entre ambas.

- **Modelo (Model):** Responsável pela representação dos dados e das regras de negócio do sistema, incluindo entidades como alunos, turmas, pagamentos e relatórios.
- **Visão (View):** Compreende os componentes da interface, responsáveis por exibir informações e interagir com o usuário final.
- **Controlador (Controller):** Atua como intermediário, processando as requisições do usuário, manipulando os dados no modelo e atualizando a visão de forma coerente.

Complementarmente, a aplicação poderá ser estruturada conforme o modelo Cliente-Servidor, padrão de arquitetura em que as responsabilidades são distribuídas entre dois ambientes distintos. De acordo com Tanenbaum e Van Steen (2017), nesse modelo o cliente é responsável pela interação direta com o usuário e pelo envio de requisições, enquanto o servidor processa essas requisições, realiza operações de negócio e retorna as respostas correspondentes. Essa abordagem é particularmente adequada

para aplicações web, pois permite o compartilhamento centralizado de dados, controle de acesso e atualização contínua sem a necessidade de intervenções locais nos dispositivos dos usuários.

No projeto em questão, o servidor deverá hospedar a lógica de negócio, os serviços de autenticação, os módulos de gerenciamento e a integração com o banco de dados. Já o cliente — acessado via navegador — poderá oferecer uma interface responsiva e intuitiva, por meio da qual administradores, gestores, instrutores e alunos interagem com o sistema.

Por fim, o design do sistema foi orientado pelos princípios de baixo acoplamento e alta coesão, buscando assegurar clareza, reuso e facilidade de manutenção dos componentes. Como salientam Pressman e Maxim (2016), o design de software deve traduzir os requisitos levantados em uma representação técnica que sirva de base para a implementação, ao mesmo tempo em que preserve a integridade conceitual do projeto.

## 2.5 TECNOLOGIAS UTILIZADAS E SUGERIDAS

Segundo Pressman e Maxim (2016), o uso de ferramentas de apoio técnico e metodológico contribui para padronizar processos, reduzir erros humanos e garantir maior controle sobre as atividades de análise, modelagem e implementação. Além disso, Sommerville (2011) ressalta que a seleção dessas ferramentas deve considerar o alinhamento com os objetivos do projeto, os recursos disponíveis e o nível de complexidade da solução proposta.

Na fase atual do trabalho — centrada no levantamento de requisitos e na modelagem conceitual do sistema — foram empregadas ferramentas que favorecem a coleta estruturada de dados, a análise de negócios e a representação visual de modelos. Para a aplicação dos questionários estruturados, foi utilizado o Forms (2024), que, por sua interface acessível e capacidade de armazenamento automático das respostas, se mostrou adequado para a coleta de informações quantitativas e qualitativas junto a diferentes perfis de usuários. Conforme Marconi e Lakatos (2017), instrumentos digitais de coleta de dados são particularmente úteis em estudos aplicados, por oferecerem praticidade, padronização e ampla acessibilidade aos respondentes.

A ferramenta Canva (2024) foi empregada para a elaboração dos *Business Model Canvas*, possibilitando a representação visual dos blocos do modelo de negócio de maneira organizada e colaborativa. Essa escolha favoreceu a integração entre análise e comunicação, tornando a interpretação dos resultados mais acessível aos envolvidos no projeto. Já para a modelagem UML, optou-se pelo uso do PlantText (2024), que se baseia na linguagem PlantUML (2024) e permite gerar diagramas de forma textual e automatizada. Essa abordagem é coerente com a proposta metodológica de Pressman e Maxim (2016), que destacam a importância de ferramentas capazes de manter consistência entre a documentação e os artefatos técnicos, especialmente em ambientes de prototipagem e modelagem conceitual.

Para as próximas etapas do projeto (voltadas ao desenvolvimento do MVP) foram sugeridas tecnologias que conciliam desempenho, escalabilidade e facilidade de manutenção, adequadas a uma arquitetura modular baseada no padrão MVC e na estrutura Cliente-Servidor. As principais linguagens de desenvolvimento recomendadas são os *frameworks open-source* Next.js (2024), Node.js (2024) e React (2024), todas baseadas em JavaScript (2024) mas usando TypeScript (2024) e oferecendo recursos integrados de renderização híbrida e alta performance para aplicações web. Essa escolha se alinha às necessidades de

construção de interfaces dinâmicas e responsivas, além de dispor de ampla documentação e suporte da comunidade.

O sistema poderá utilizar o PostgreSQL (2024) como banco de dados relacional, devido à sua robustez, conformidade com padrões SQL (*Structured Query Language*) e suporte a operações complexas de transação, conforme destacado por Elmasri e Navathe (2015) ao abordarem bancos de dados orientados à confiabilidade e escalabilidade. O serviço de hospedagem sugerido é a Vercel (2024), serviço que oferece integração nativa com o Next.js (2024) e PostgreSQL (2024) através da plataforma *serverless open-source* Neon (2024), garantindo uma série de recursos de monitoramento, bando de dados distribuído e suporte a implantações contínuas, o que permitirá atualizar o sistema de forma rápida e segura. O código-fonte poderá ser versionado e mantido em um repositório GitHub (2024), garantindo controle de versões, rastreabilidade e colaboração entre os participantes do projeto, práticas reconhecidas como essenciais para a gestão moderna de software, conforme Spinellis (2012).

Por fim, a execução do projeto poderá seguir a metodologia ágil Scrumban, definida por Ladas (2009), que combina os princípios do Scrum.org (2024) — voltados à organização iterativa e incremental — com a flexibilidade visual da ferramenta Kanban, que prioriza o fluxo contínuo de trabalho. Conforme Anderson (2010), o Scrumban é especialmente útil em equipes pequenas e projetos de longo prazo, pois oferece um equilíbrio entre planejamento estruturado e adaptação contínua, sem perder a visibilidade do progresso e das prioridades. Essa metodologia pode ser aplicada para planejar as *sprints*, acompanhar a evolução das tarefas e promover ajustes progressivos com base no *feedback* obtido nas futuras etapas de testes.

## 2.6 PLANO DE DESENVOLVIMENTO

Segundo Sommerville (2011), um plano de desenvolvimento bem estruturado permite estabelecer uma sequência lógica de atividades, definir responsabilidades, prever riscos e assegurar que o produto final atenda aos requisitos especificados. Também, Pressman e Maxim (2016) afirmam que o planejamento não deve ser visto como uma fase estática, mas como um processo contínuo de adaptação e refinamento, que evolui conforme o sistema se desenvolve e novas informações são obtidas.

No contexto deste trabalho, o plano de desenvolvimento tem por objetivo orientar a futura construção do MVP e as etapas subsequentes de evolução do sistema. Durante a criação do MVP, as atividades poderão ser organizadas com base nas diretrizes do modelo incremental, porém sem adoção imediata de ciclos de entrega contínua ou coleta de *feedback*, uma vez que essa fase busca consolidar uma versão inicial completa, funcional e passível de validação pelo cliente voluntário. Após a finalização do MVP e a realização dos testes iniciais, o desenvolvimento poderá adotar efetivamente o modelo incremental, permitindo que novas versões do sistema sejam implementadas de forma iterativa e progressiva, conforme os resultados coletados.

Para o gerenciamento e acompanhamento das próximas etapas, recomenda-se a utilização de metodologias ágeis, que oferecem maior flexibilidade, adaptação e transparência ao processo de desenvolvimento. Conforme destacam Highsmith (2002) e Beck et al. (2001), os métodos ágeis surgiram como uma resposta à rigidez dos modelos tradicionais, enfatizando a entrega de valor contínuo, a colaboração entre equipes e a capacidade de responder rapidamente a mudanças nos requisitos. Em vez de seguir um plano fixo e

linear, as metodologias ágeis promovem o desenvolvimento iterativo e incremental, no qual o software evolui a partir de ciclos curtos de planejamento, implementação e revisão.

Entre as abordagens ágeis mais consolidadas, o Scrum.org (2024) é amplamente adotado para gestão de projetos complexos. Segundo Schwaber e Sutherland (2020), o Scrum.org (2024) organiza o trabalho em *sprints* — ciclos de tempo fixo, geralmente de duas a quatro semanas — dentro dos quais são desenvolvidas entregas funcionais e verificáveis do produto. Cada *sprint* contempla atividades de planejamento, execução, revisão e retrospectiva, promovendo uma melhoria incremental do processo e do produto.

Outra ferramenta muito utilizada é o Kanban, que, segundo Anderson (2010), se baseia na visualização do fluxo de trabalho por meio de quadros e cartões, permitindo controlar a capacidade da equipe e reduzir gargalos. Essa abordagem prioriza a eficiência e o fluxo contínuo, promovendo transparência e foco na entrega constante de valor, sem exigir a estruturação rígida de ciclos temporais. O Kanban é particularmente útil em projetos de longo prazo ou em equipes pequenas, em que a natureza das tarefas pode variar em ritmo e complexidade.

Diante das características do projeto e da limitação de recursos humanos, o presente trabalho sugere a adoção do modelo híbrido Scrumban para as etapas futuras de desenvolvimento. O Scrumban combina a estrutura iterativa do Scrum.org (2024) com a flexibilidade e fluidez do Kanban, permitindo equilibrar planejamento e adaptação contínua. De acordo com Ladas (2009), o Scrumban é especialmente indicado para equipes reduzidas e projetos em evolução, pois mantém a organização de ciclos curtos e objetivos definidos, ao mesmo tempo em que possibilita ajustes de prioridade em tempo real. Essa abordagem favorece o acompanhamento contínuo das atividades, a visualização do progresso e a adaptação do plano conforme surgem novas necessidades ou restrições técnicas.

A sugestão dessa metodologia está alinhada às boas práticas de engenharia contemporânea, valorizando a entrega incremental, a comunicação transparente e o aprendizado contínuo. Assim, o plano de desenvolvimento proposto neste trabalho tenta estabelecer uma base para as etapas de implementação e teste, buscando garantir que o sistema evolua de forma controlada, sustentável e coerente com as diretrizes de qualidade e validação apresentadas nas seções anteriores.

## 2.7 VALIDAÇÃO E REFINAMENTO

A validação de software constitui uma etapa fundamental da Engenharia de Software, voltada a garantir que o produto desenvolvido atenda às expectativas e necessidades dos usuários. Segundo Sommerville (2011), o processo de validação busca assegurar que o sistema “faça o que o cliente realmente deseja”, enquanto a verificação avalia se o software foi construído corretamente de acordo com suas especificações. Juntas, essas atividades formam um ciclo contínuo de controle de qualidade, essencial para reduzir falhas, retrabalho e custos nas fases posteriores do desenvolvimento.

De acordo com Pressman e Maxim (2016), a validação deve ser conduzida desde as primeiras fases do ciclo de vida do software, mesmo antes da implementação, por meio da revisão dos requisitos, da análise de consistência entre modelos e da validação conceitual de diagramas e fluxos de interação. No contexto deste trabalho, em que o foco recai sobre as etapas iniciais de levantamento de requisitos e modelagem do sistema, a validação foi conduzida de forma qualitativa, buscando confirmar a coerência entre as necessidades identificadas e as soluções propostas nos artefatos de design.

Além dos requisitos funcionais, foi considerada a validação dos requisitos não funcionais, que englobam características como desempenho, usabilidade, segurança, confiabilidade e escalabilidade. Segundo Sommerville (2011), esses requisitos exercem influência direta sobre a experiência do usuário e o comportamento do sistema em condições reais de operação, sendo determinantes para sua aceitação e sucesso. Pressman e Maxim (2016) complementam, que a ausência de validação adequada desses aspectos frequentemente resulta em sistemas que, embora funcionais, apresentam limitações práticas significativas.

No contexto deste projeto, a usabilidade poderá ser analisada de forma teórica com base em princípios de interação humano-computador, considerando a necessidade de uma interface acessível e intuitiva para diferentes perfis de usuários (administradores, gestores, instrutores e alunos). O desempenho será considerado em relação à futura implementação web do sistema, priorizando respostas rápidas a solicitações comuns, como o carregamento de turmas, registros de pagamento e comunicações internas. Já a segurança será abordada como requisito crítico, especialmente pela natureza sensível dos dados envolvidos (financeiros e pessoais), prevendo autenticação segura, proteção contra acesso indevido e confidencialidade nas interações — práticas recomendadas por Pfleeger e Atlee (2010) ao discutir a engenharia da segurança de sistemas.

Esses requisitos poderão ser verificados de forma prática nas próximas etapas do projeto, com o desenvolvimento e teste do MVP. Segundo Larman (2004), a prototipagem incremental é uma das formas mais eficazes de validar hipóteses de design e avaliar requisitos não funcionais, pois permite a observação direta do comportamento do sistema sob uso real. O MVP poderá ser testado com o cliente voluntário participante deste estudo, e os futuros resultados obtidos servirão de base para o processo de refinamento — etapa conduzida de forma iterativa e incremental, permitindo que cada nova versão do sistema incorpore melhorias baseadas em *feedback* e dados empíricos.

Conforme Sommerville (2011) e Pressman e Maxim (2016), o refinamento contínuo constitui um mecanismo essencial para assegurar a qualidade de software ao longo de seu ciclo de vida, garantindo que as evoluções implementadas mantenham coerência com as necessidades dos usuários e com os princípios de desempenho, segurança e usabilidade definidos desde a concepção do projeto.

## 2.8 CONCLUSÃO

A construção de um sistema ERP eficaz requer uma combinação de boas práticas de Engenharia de Software, adoção de um ciclo de vida incremental aliado a técnicas ágeis e compreensão clara do contexto de negócio. O levantamento de requisitos, realizado por meio de entrevistas e questionários, permitiu inferir as necessidades reais dos *stakeholders* em requisitos funcionais e não funcionais que guiam o projeto.

A modelagem do sistema, com diagramas de casos de uso, de classes e sequência, fornece uma visão estruturada das entidades e das interações esperadas, servindo como ponte entre requisitos e implementação. A arquitetura proposta, baseada em princípios de modularidade e separação de responsabilidades (*Model-View-Controller*), suporta a escalabilidade e a manutenção contínua do sistema. As escolhas tecnológicas (Next.js (2024), TypeScript (2024), PostgreSQL (2024)) foram orientadas por requisitos de flexibilidade, desempenho e facilidade de integração com serviços modernos de hospedagem.

Finalmente, os procedimentos de validação e refinamento descritos aqui (testes com usuários, avaliação de desempenho, usabilidade e segurança) constituem etapas essenciais para garantir que a solução seja prática, segura e alinhada às expectativas dos usuários

finais. O tratamento explícito dos impactos sociais — inclusão digital, sustentabilidade, e criação de um ambiente online seguro — reforça que projetos tecnológicos bem-sucedidos devem equilibrar objetivos técnicos com responsabilidades éticas e sociais.

Com isso, é possível ter uma visão consolidada das relações entre análise de negócio, requisitos, modelagem e escolhas arquiteturais que orientarão a implementação e a validação do sistema.

### 3 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir das etapas de levantamento de requisitos, análise do modelo de negócio, modelagem do sistema e plano de desenvolvimento definido, com base nas informações coletadas. Os resultados são expostos de forma descritiva e analítica, com ênfase em sua contribuição para a estruturação da proposta do sistema e para as decisões relacionadas às próximas fases do projeto.

#### 3.1 QUESTIONÁRIOS

Os questionários estruturados aplicados aos administradores, instrutores e alunos do cliente voluntário ajudaram na compreensão das rotinas operacionais e das dificuldades enfrentadas no gerenciamento das atividades administrativas e pedagógicas. A coleta de dados possibilitou identificar pontos críticos, como a ausência de integração entre controles financeiros, turmas e agendamentos, além da carência de ferramentas adequadas para comunicação e compartilhamento de conteúdos com os alunos.

As respostas indicam que todas as principais funcionalidades relacionadas à automação de processos de agendamento e reagendamento de aulas, comunicação eficiente em tempo real com alunos, controle financeiro, compartilhamento de conteúdos e segurança digital são muito valorizadas pelos administradores, instrutores e alunos praticantes.

Os resultados detalhados dos questionários estão apresentados no Apêndice A e Apêndice B, onde são discutidas as principais tendências observadas, bem como as sugestões e observações fornecidas pelos participantes.

#### 3.2 BUSINESS MODEL CANVAS

Com base nos dados coletados através dos questionários, foram elaborados dois BMCs — um representando o modelo de negócio do *stakeholder* (cliente voluntário) e outro descrevendo o modelo da plataforma proposta. O primeiro permitiu uma melhor compreensão dos fluxos de valor, os recursos, as necessidades e as atividades essenciais que sustentam o funcionamento dos estúdios e escolas de esportes e dança. Já o segundo *canvas* representou o posicionamento estratégico do sistema sugerido, incluindo as propostas de valor, os segmentos de clientes e as potenciais fontes de receita.

Esses modelos contribuíram para a identificação das oportunidades de automação, formulação dos requisitos do sistema e para a estruturação das funcionalidades centrais que posteriormente foram representadas nos diagramas UML. Além disso, auxiliaram na escolha das ferramentas tecnológicas mais adequadas às demandas observadas, reforçando a coerência entre o contexto de aplicação e as soluções projetadas.

A seguir, a Figura 3.1 apresenta a ferramenta aplicada ao *stakeholder* e a Figura 3.2 aplicada à solução proposta, servindo como bases conceituais para as etapas seguintes. Os dois modelos foram elaborados utilizando a ferramenta online Canva (2024).

Business Model Canvas				Stakeholder
Recursos-Chave	Atividades-Chave	Proposta de Valor	Canais	Relacionamento com Clientes
<p><b>Equipe Técnica:</b></p> <ul style="list-style-type: none"> <li>Proprietário/Administrador que representa a empresa;</li> <li>Gestores, que cuidam de processos documentais, agendas e/ou financeiro;</li> <li>Instrutores profissionais.</li> </ul> <p><b>Tecnologia:</b></p> <ul style="list-style-type: none"> <li>Cadernos e livros-caixa;</li> <li>Planilhas e ferramentas de gestão de tarefas;</li> <li>Smartphones individuais;</li> <li>Grupos em aplicativos de conversa.</li> </ul> <p><b>Parcerias:</b></p> <ul style="list-style-type: none"> <li>Especialistas do nicho (pole dancers e instrutores de dança e esportes);</li> <li>Eventos com participação de outros estabelecimentos;</li> <li>Influenciadores do setor esportivo e artístico.</li> </ul>	<ul style="list-style-type: none"> <li>Controle de fluxo de caixa (lucros, mensalidades, salários, impostos, etc.), cadastro de alunos, trabalho de instrutores e de turmas.</li> <li>Oferta de turmas de horário fixo disponíveis à escolha do aluno, com instrutor(es) profissionais.</li> <li>Agendamento de reposição de aulas, seja em turma existente ou criação de turma alternativa de encontro único, exclusiva para reposições.</li> <li>Acompanhamento da presença e evolução do aluno pelo instrutor responsável de forma individual.</li> <li>Divulgação do estabelecimento em mídias sociais e promoção de eventos comemorativos temáticos, com descontos e condições especiais para alunos e visitantes convidados.</li> </ul>	<ul style="list-style-type: none"> <li>Aprendizado e desenvolvimento pessoal, com metodologia didática acessível e instrutores qualificados.</li> <li>Ambiente acolhedor, seguro, inclusivo e motivador, onde alunos e instrutores constroem relações de confiança e pertencimento.</li> <li>Flexibilidade e conveniência, por meio de horários flexíveis, opções de reposição de aulas e formas simplificadas de agendamento e pagamento.</li> <li>Profissionalismo e credibilidade, através de gestão, organização, comunicação clara e compromisso com a evolução dos alunos.</li> <li>Saúde e bem-estar por meio de práticas que aliam atividade corporal, disciplina, expressão artística e autoconhecimento.</li> </ul>	<p><b>Digitais:</b></p> <ul style="list-style-type: none"> <li>Site oficial da empresa;</li> <li>Campanhas e divulgações em redes sociais (Instagram, Facebook, TikTok, etc.);</li> <li>E-mail marketing para alunos.</li> </ul> <p><b>Presenciais:</b></p> <ul style="list-style-type: none"> <li>Apresentações e demonstrações em eventos locais de dança ou esportes;</li> <li>Parcerias com associações de dança ou federações esportivas.</li> </ul>	<p><b>Suporte:</b></p> <ul style="list-style-type: none"> <li>Atendimento aos alunos via chat online, telefone e e-mail para tirar dúvidas e resolver problemas.</li> </ul> <p><b>Comunidade:</b></p> <ul style="list-style-type: none"> <li>Relacionamento por meio de mídias sociais.</li> </ul> <p><b>Onboarding:</b></p> <ul style="list-style-type: none"> <li>Matrículas manuais de alunos por meio de fichas.</li> </ul> <p><b>Feedback Contínuo:</b></p> <ul style="list-style-type: none"> <li>Coleta de feedback dos alunos para idéias e melhorias.</li> </ul>
Parcerias-Chave			Segmentos de Clientes	
<ul style="list-style-type: none"> <li>Divulgações com <i>Influencers</i>.</li> <li>Presença em eventos de apresentações, divulgações e demonstrações.</li> <li>Parcerias com associações esportivas.</li> </ul>			<p><b>Primários:</b></p> <ul style="list-style-type: none"> <li>Alunos.</li> </ul> <p><b>Secundários:</b></p> <ul style="list-style-type: none"> <li>Instrutores profissionais.</li> <li>Pequenos empreendedores no setor esportivo ou artístico.</li> </ul>	
Estrutura de Custos			Fontes de Receita	
<p><b>Custos Fixos:</b></p> <ul style="list-style-type: none"> <li>Custo de infraestrutura do estabelecimento, como aluguel e IPTU;</li> <li>Salários da equipe técnica e administrativa;</li> <li>Depreciação de equipamentos;</li> <li>Custos de limpeza e manutenção preventiva.</li> </ul> <p><b>Custos Variáveis:</b></p> <ul style="list-style-type: none"> <li>Energia elétrica, água e gás;</li> <li>Manutenção emergencial;</li> <li>Custo de meios de pagamento;</li> <li>Publicidade com <i>Influencers</i> e mídias sociais.</li> </ul>			<ul style="list-style-type: none"> <li>Assinatura Mensal/Anual dos alunos.</li> <li>Lucro com a venda de alguns consumíveis, como barras de proteína, isotônicos, etc.</li> <li>Lucro com a venda de roupas de treino e equipamentos específicos.</li> </ul>	

Figura 3.1: *Business Model Canvas* aplicado ao stakeholder.

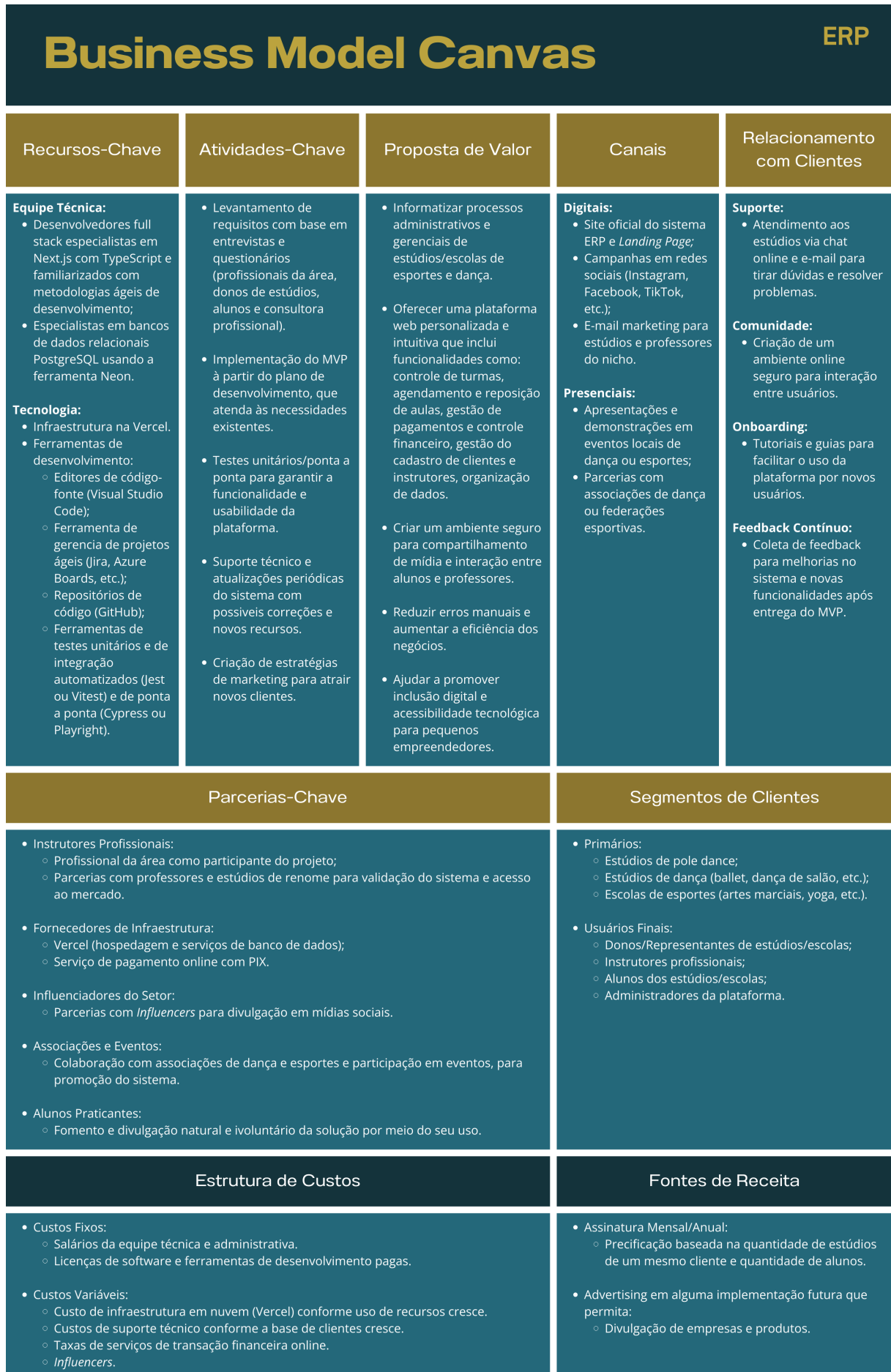


Figura 3.2: *Business Model Canvas* aplicado à solução proposta.

### 3.3 MODELAGEM

A partir dos requisitos e informações sintetizadas nas etapas anteriores, foram elaborados três diagramas UML — de casos de uso, de classes e de sequência —, buscando representar de forma visual e organizada a estrutura e o comportamento do sistema. O diagrama de casos de uso proporcionou uma visão de alto nível sobre as funcionalidades esperadas, permitindo compreender melhor as interações entre os diferentes perfis de usuários. O diagrama de classes auxiliou na estruturação das principais entidades do sistema, seus atributos e relacionamentos, servindo como base para o planejamento conceitual da arquitetura e do design. Por fim, os diagramas de sequência representam os principais fluxos dinâmicos de mensagens entre objetos e componentes críticos, como matrícula, registro de presença, agendamento e pagamento.

Em conjunto, esses diagramas contribuíram para a validação dos requisitos levantados, para a organização lógica da arquitetura proposta e para o planejamento técnico das etapas subsequentes. A representação integrada das dimensões estrutural e comportamental do sistema também auxiliaram na identificação de possíveis dependências e pontos de atenção a serem avaliados durante a implementação do MVP.

A seguir, são apresentados os diagramas UML desenvolvidos. Detalhes técnicos sensíveis foram omitidos para preservar a confidencialidade do projeto.

#### 3.3.1 Diagrama de Casos de Uso

O diagrama de casos de uso descreve as principais interações entre os atores (Administrador da Plataforma, Administrador da Empresa, Instrutor e Aluno) e o sistema. Cada caso de uso representa uma funcionalidade crítica, como gerenciamento de empresas, turmas, matrículas e pagamentos, refletindo as necessidades identificadas durante o levantamento de requisitos. Por praticidade, algumas palavras usadas nas funcionalidades denotam ações simplificadas, como "Gerenciar", que engloba criar, ler, atualizar e deletar (*CRUD*).



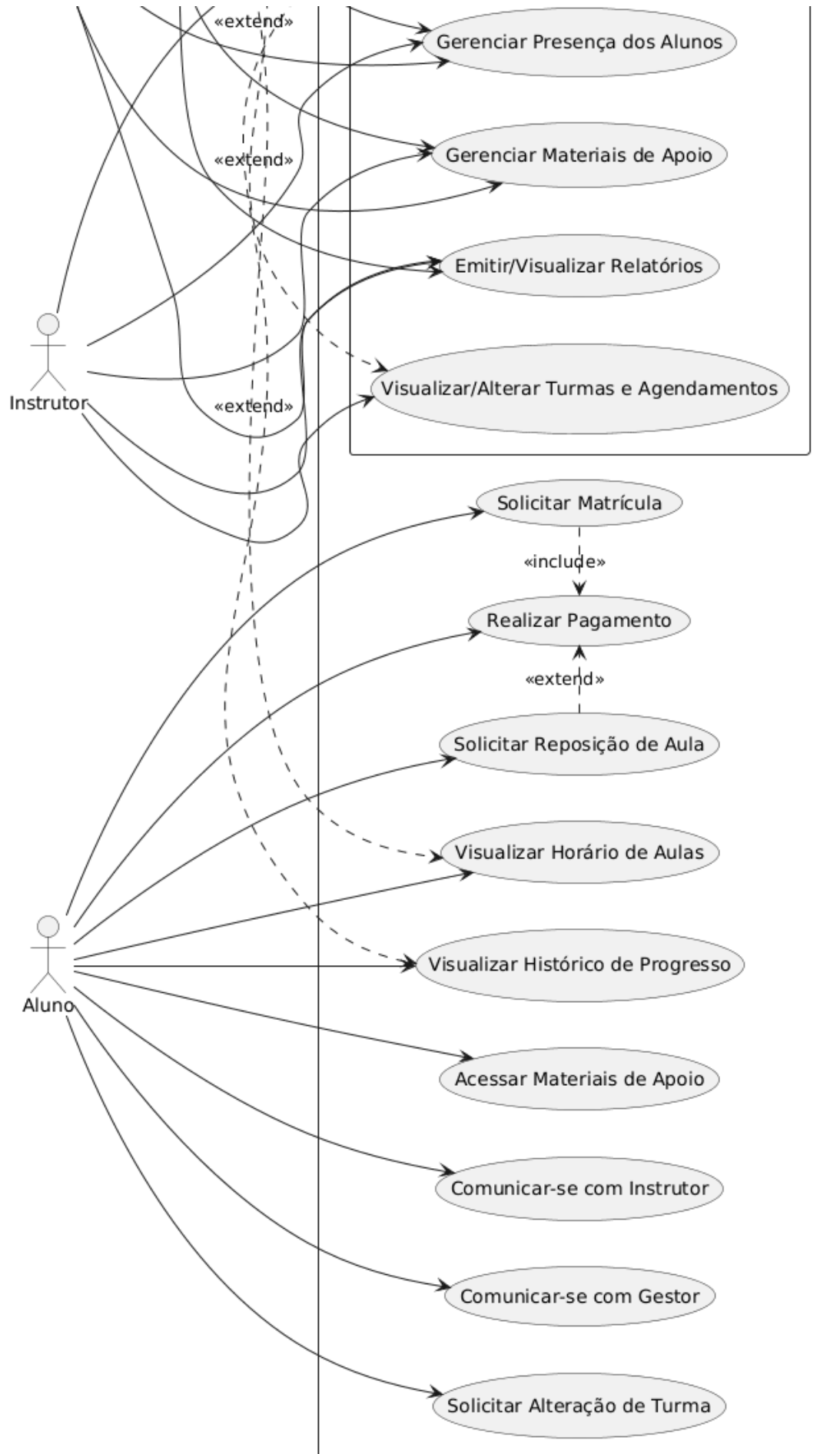


Figura 3.4: Diagrama de Casos de Uso do sistema (parte dois).

### 3.3.2 Diagrama de Classes

O diagrama de classes apresenta as principais entidades do sistema, seus principais atributos, métodos e os relacionamentos entre elas. Cada classe foi definida com base nos requisitos funcionais identificados, buscando representar de forma clara e organizada os dados e comportamentos essenciais para o funcionamento do sistema. As associações entre as classes refletem as interações e dependências necessárias para a execução das funcionalidades propostas.

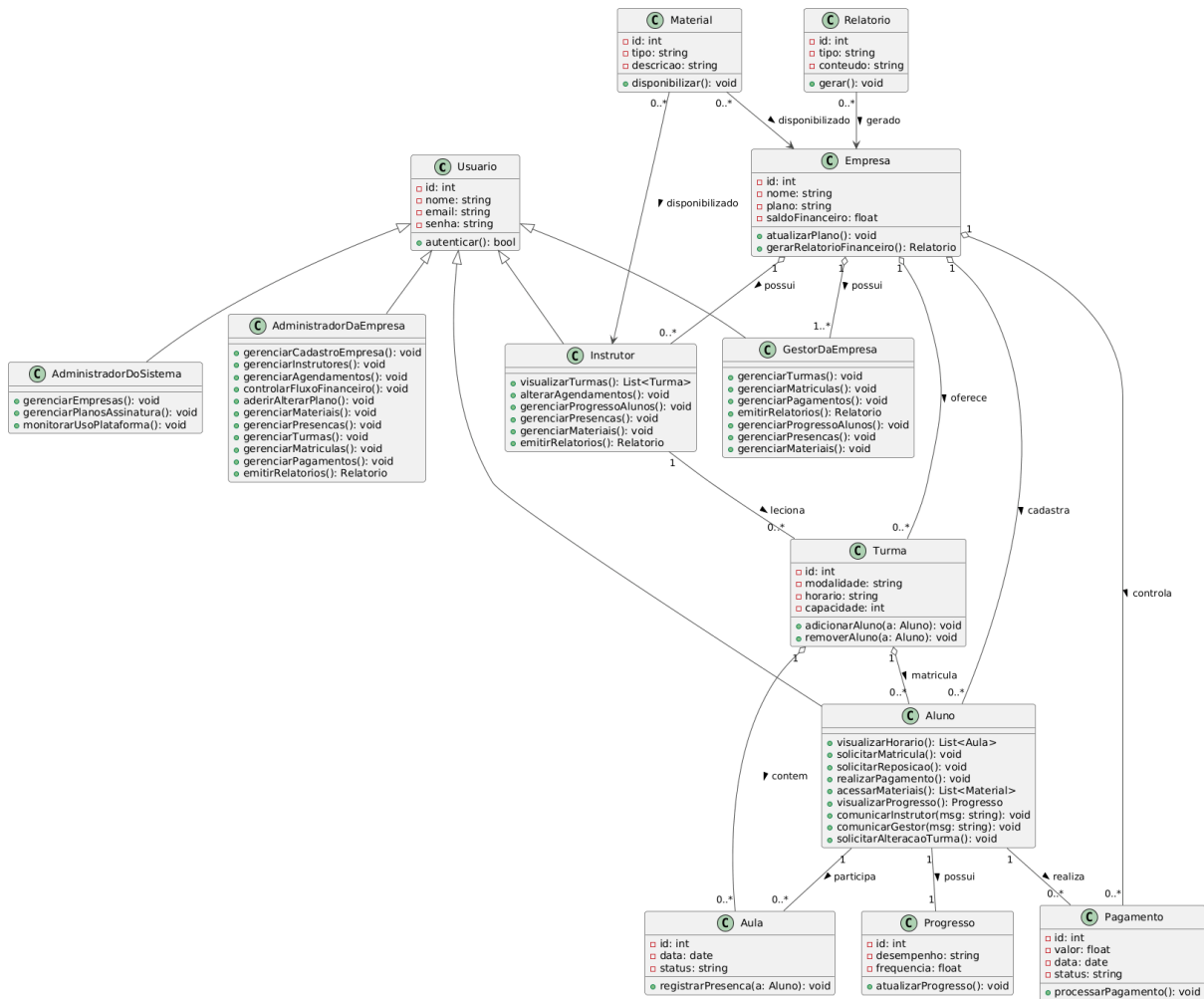


Figura 3.5: Diagrama de Classes do sistema.

### 3.3.3 Diagramas de Sequência

Os sete diagramas de sequência buscam detalhar os principais fluxos dinâmicos de mensagens entre objetos e componentes do sistema para funcionalidades críticas, como o processo de matrícula, agendamento de aulas, registro de presença e pagamento de mensalidades. Cada diagrama ilustra a ordem das interações ao longo do tempo, evidenciando como os diferentes atores e objetos colaboram para realizar cada tarefa. Esses diagramas são fundamentais para compreender o comportamento do sistema em cenários específicos, auxiliando na identificação de possíveis pontos de melhoria e na validação dos requisitos levantados.

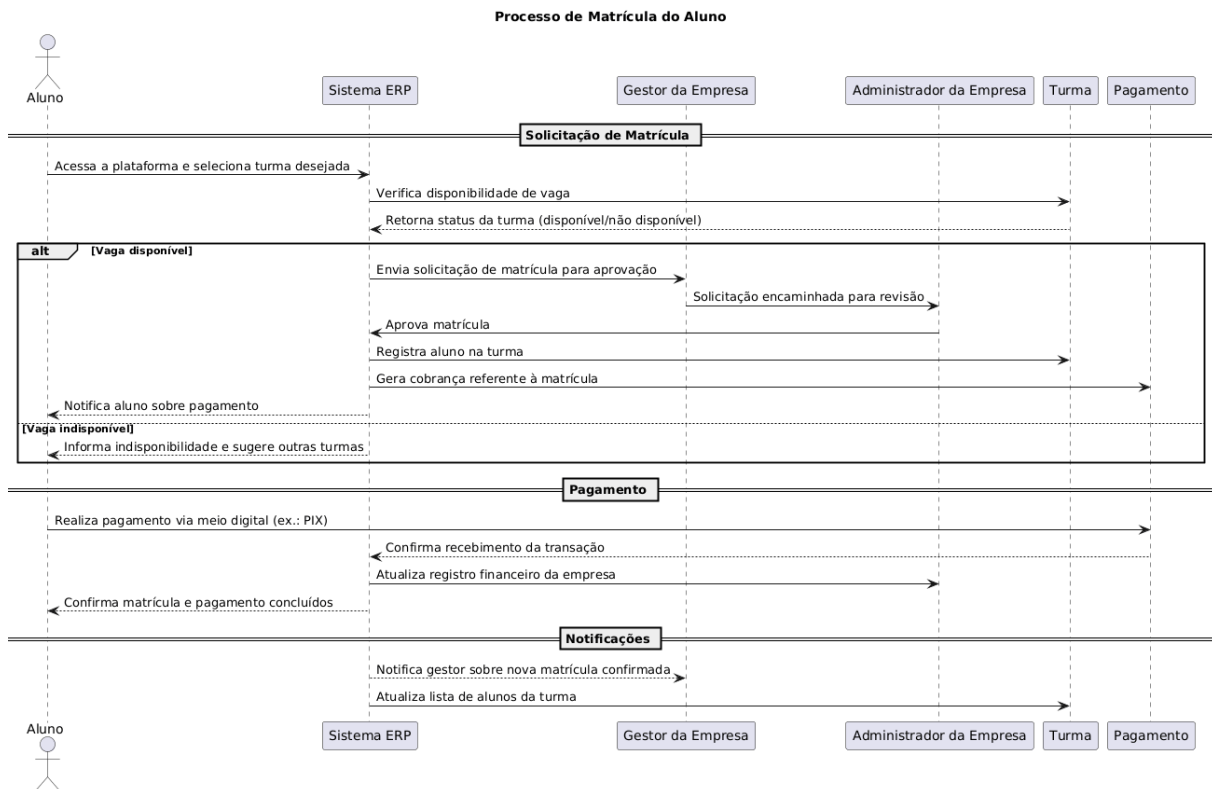


Figura 3.6: Diagrama de Sequência - Processo de Matrícula.

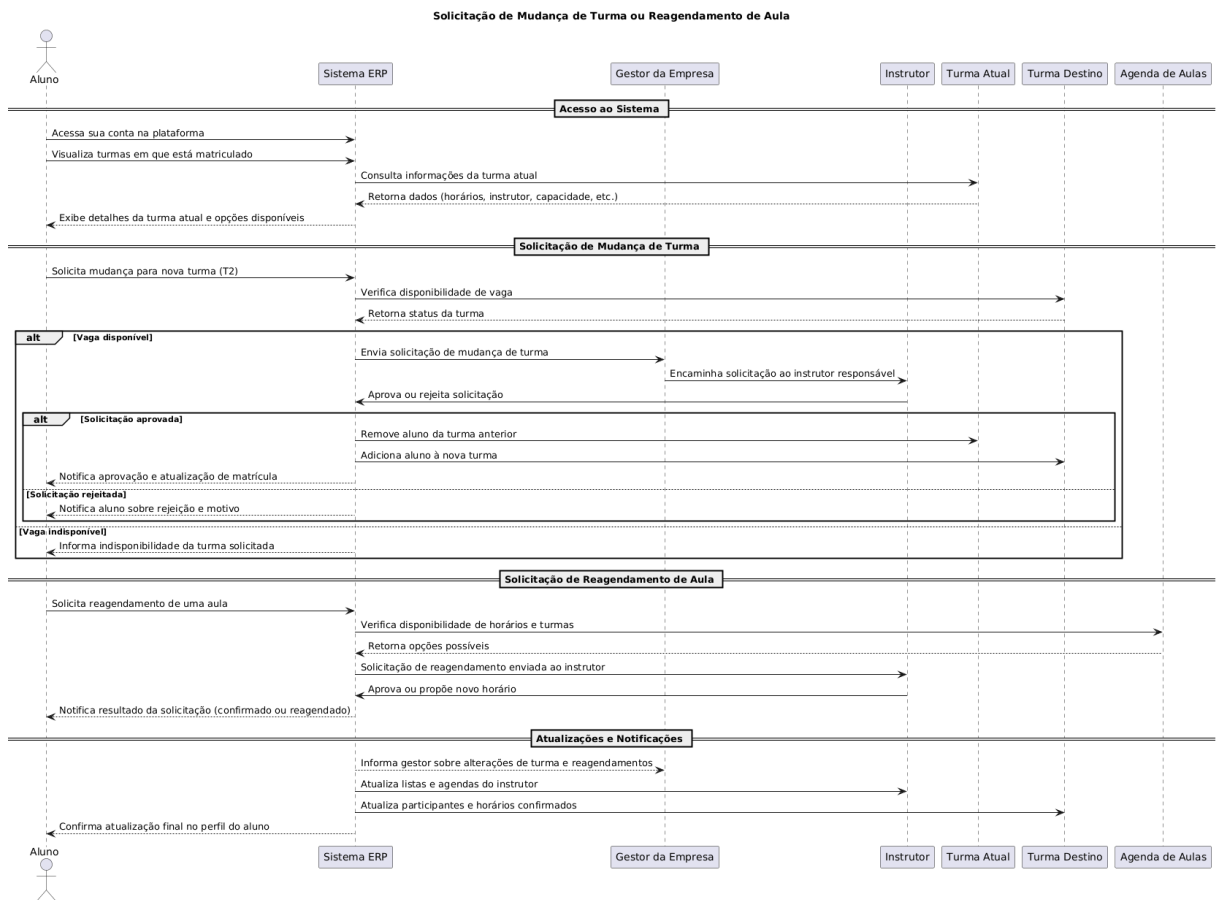


Figura 3.7: Diagrama de Sequência - Processo de Mudança de Turma ou Reagendamento de Aula.

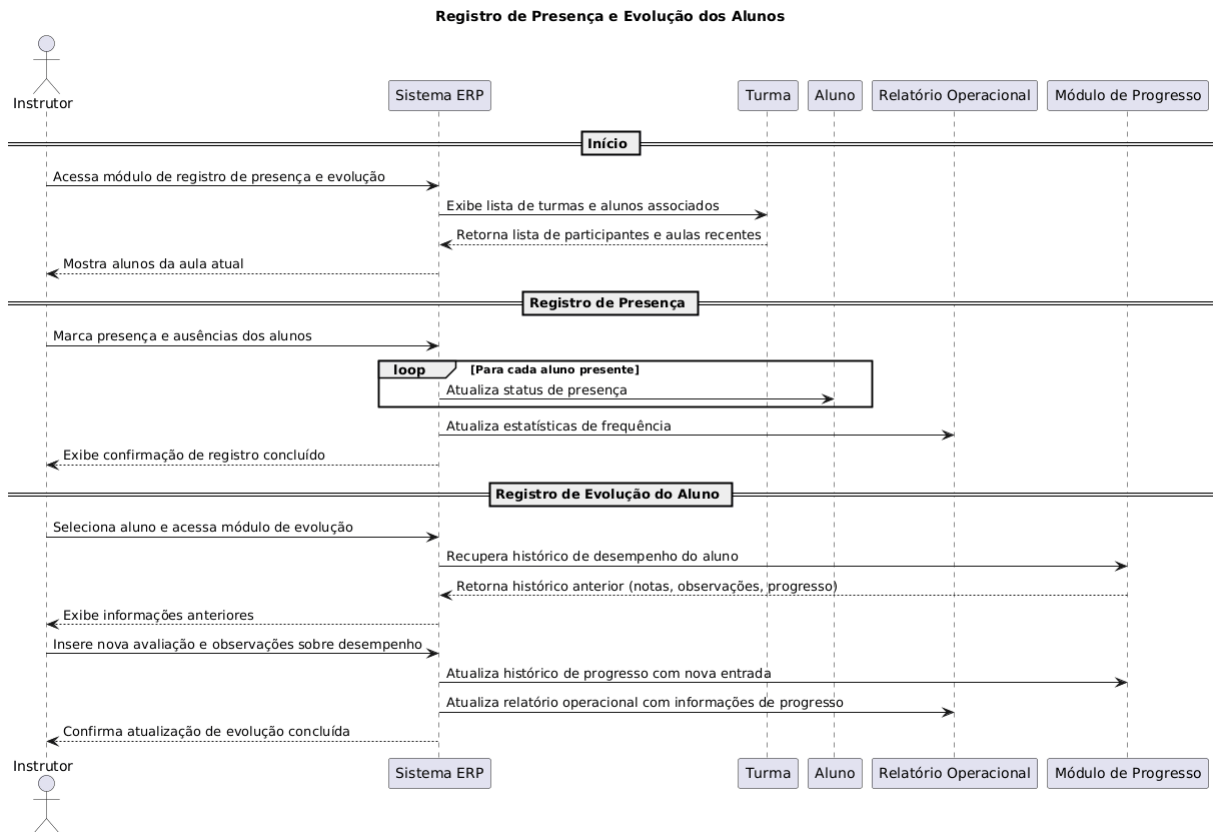


Figura 3.8: Diagrama de Sequência - Registro de Presença e Evolução dos Alunos.

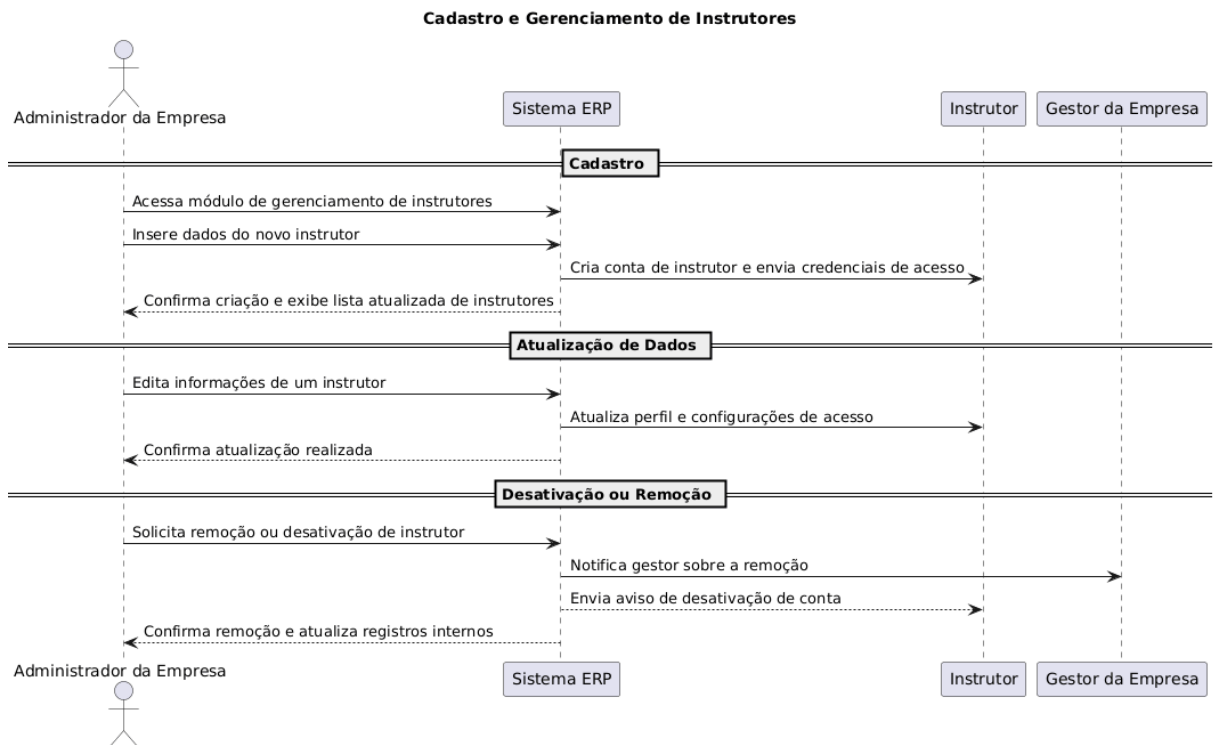


Figura 3.9: Diagrama de Sequência - Cadastro e Gerenciamento de Instrutores.

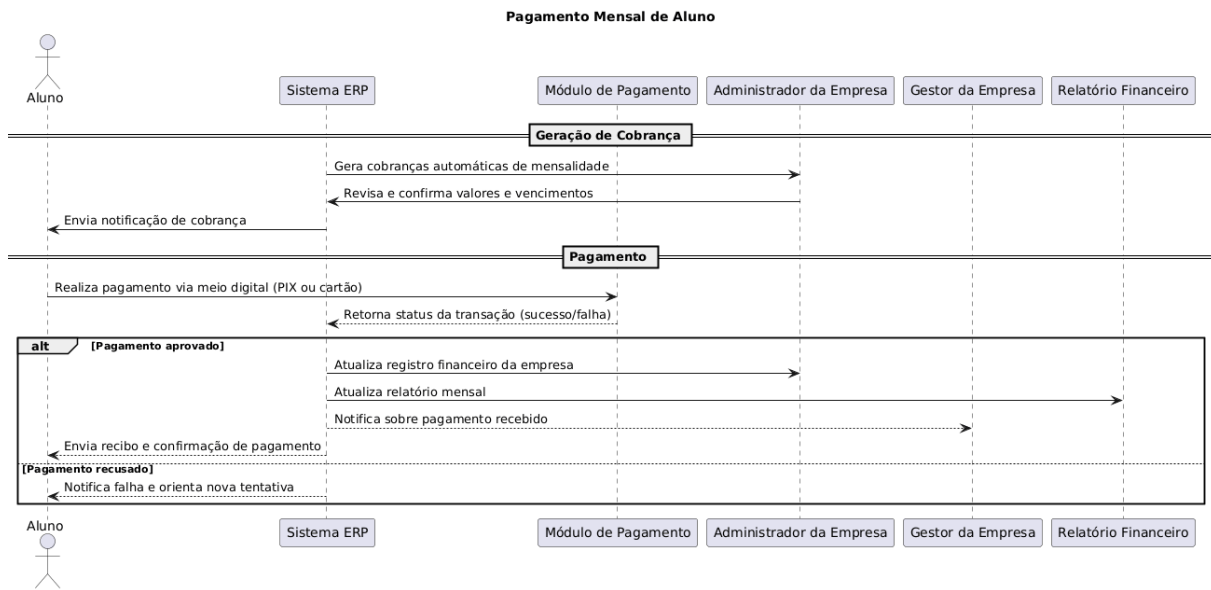


Figura 3.10: Diagrama de Sequência - Pagamento Mensal de Aluno.

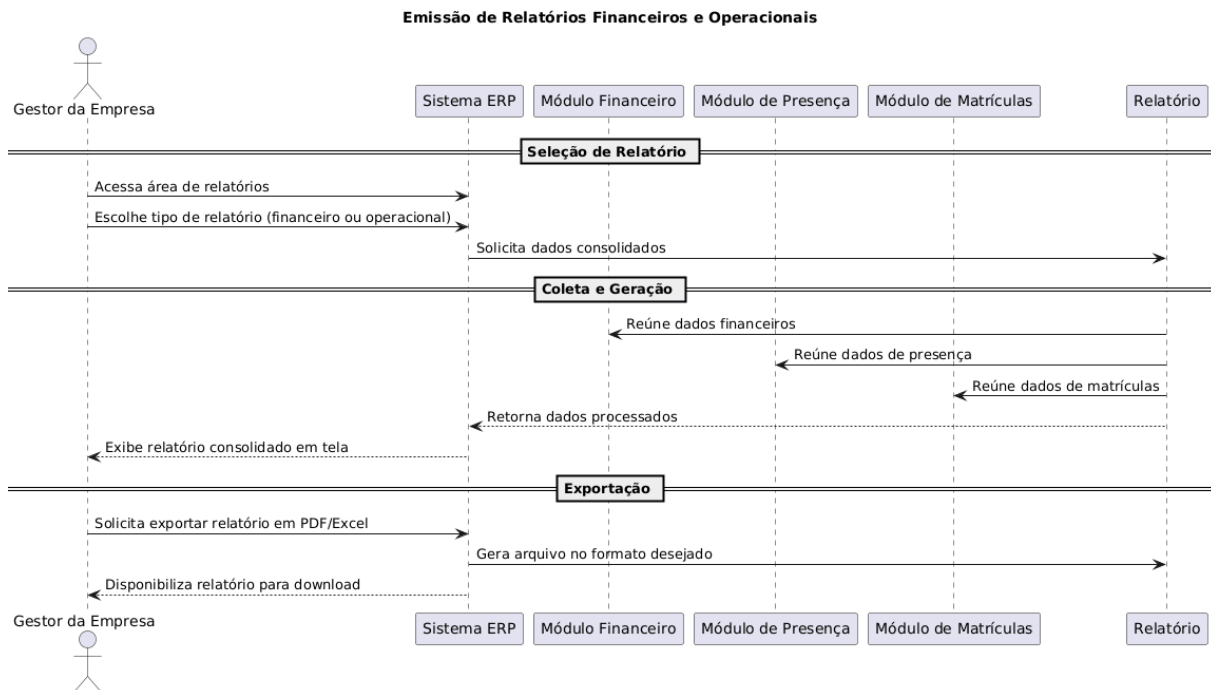


Figura 3.11: Diagrama de Sequência - Emissão de Relatórios Financeiros e Operacionais.

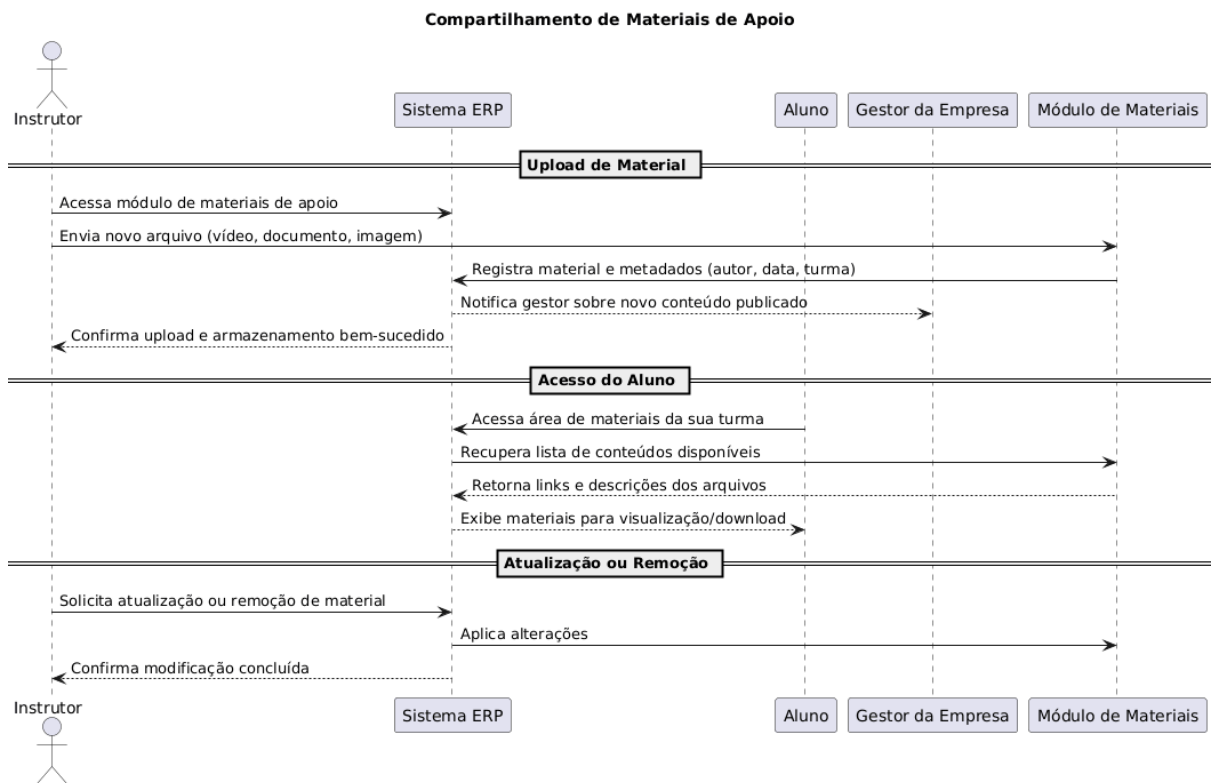


Figura 3.12: Diagrama de Sequência - Compartilhamento de Materiais de Apoio.

### 3.4 TECNOLOGIAS SELECIONADAS

Com base nos requisitos funcionais e não funcionais identificados, bem como nas necessidades específicas do cliente voluntário e na praticidade de implementação, foram sugeridas as seguintes tecnologias para o desenvolvimento do MVP:

- Linguagens de Programação:** Next.js (2024) usando TypeScript (2024) para o *frontend* e *backend*, proporcionando uma experiência de usuário dinâmica. A escolha se deve à facilidade de desenvolvimento e integração ao utilizar serviços da empresa Vercel (2024), pois toda a infraestrutura oferecida é otimizada para essa *stack* uma vez que a própria empresa criou o *framework* em questão. Tanto o Next.js (2024) quanto as ferramentas que o compõem são *Open-Source*, o que contribui para a redução de custos e maior flexibilidade no desenvolvimento.
- Banco de Dados:** PostgreSQL (2024), um sistema de gerenciamento de banco de dados relacional robusto e amplamente utilizado, conhecido por sua confiabilidade, desempenho e conformidade com padrões SQL. A escolha do PostgreSQL (2024) se deve à sua capacidade de lidar com dados estruturados e relacionais, além de oferecer recursos avançados como suporte a transações, integridade referencial e extensibilidade. Além disso, o PostgreSQL (2024) é uma solução *Open-Source*, o que contribui para a redução de custos e maior flexibilidade na gestão dos dados. As soluções da Vercel (2024) oferecem integração nativa com a plataforma *serverless Open-Source* Neon (2024), que disponibiliza bancos de dados PostgreSQL (2024) gerenciados, facilitando a configuração, escalabilidade e manutenção do banco de dados para o MVP.

- **Controle de Versão:** É recomendado Git (2024), utilizando a plataforma GitHub (2024) para hospedagem do repositório, controle de versões e colaboração entre desenvolvedores.
- **Hospedagem:** Vercel (2024), escolhida pela facilidade de integração com Next.js (2024) e pela oferta de diversos serviços otimizados para aplicações web modernas.
- **Metodologia Ágil:** Scrumban é sugerido para gerenciar o desenvolvimento, combinando a estrutura de *sprints* do Scrum.org (2024) com o fluxo visual do Kanban, permitindo flexibilidade e adaptação contínua às mudanças nos requisitos, prevendo uma equipe pequena de desenvolvimento.

### 3.5 PLANO DE DESENVOLVIMENTO

O plano de desenvolvimento busca organizar as atividades e entregas previstas para o ciclo de criação e evolução do sistema. Baseado no modelo incremental e orientado por práticas ágeis, o plano propõe uma sequência de incrementos de projeto, cada um composto por funcionalidades específicas e associadas a histórias de usuário.

Ainda que o uso do *framework* Scrumban seja apenas uma recomendação, sua estrutura híbrida foi considerada adequada para o contexto do projeto, por permitir acompanhar o progresso de forma visual e flexível, mantendo o foco em entregas organizadas e prioridades definidas. O plano tem por objetivo servir como guia estratégico para o desenvolvimento do MVP e suas versões futuras, sugerindo que as etapas de implementação sigam uma lógica evolutiva coerente com o planejamento técnico e com os resultados obtidos.

Assim, foi criado um plano de desenvolvimento detalhado para o MVP, dividido em *features*, incrementos e *user stories*, considerando uma equipe pequena (um a dois desenvolvedores) e *sprints* de três semanas, totalizando sessenta e três semanas, divididas em vinte e uma *sprints*, somando trinta e três *user stories*. A seguir, uma breve descrição das etapas deste plano de desenvolvimento:

#### 3.5.1 Incremento 1 - Configuração do Ambiente e Arquitetura Base

O objetivo desse incremento é estabelecer a estrutura técnica do projeto e preparar o ambiente de desenvolvimento. Com duração estimada de três *sprints*, serão implementados os seguintes *user stories*:

- **US01:** Como desenvolvedor, quero configurar o repositório no GitHub (2024) para controle de versões.
- **US02:** Como desenvolvedor, quero criar a estrutura base do projeto em Next.js (2024) seguindo o padrão MVC.
- **US03:** Como desenvolvedor, quero configurar o banco de dados PostgreSQL (2024) e conectar à aplicação.
- **US04:** Como desenvolvedor, quero configurar o ambiente de hospedagem na Vercel (2024) para futuras publicações.
- **US05:** Como desenvolvedor, quero implementar a tela inicial genérica para validar a comunicação cliente-servidor.

### 3.5.2 Incremento 2 - Módulo de Autenticação e Controle de Acesso

O objetivo desse incremento é permitir o cadastro e login de usuários com papéis distintos. Com duração estimada de três *sprints*, serão implementadas as seguintes funcionalidades:

- **US06:** Como usuário, quero criar uma conta com e-mail e senha para acessar o sistema.
- **US07:** Como usuário, quero fazer login e logout de forma segura.
- **US08:** Como administrador da empresa, quero cadastrar novos usuários (instrutores e alunos).
- **US09:** Como sistema, quero diferenciar os tipos de usuário (Administrador da Empresa, Gestor, Instrutor e Aluno) e aplicar permissões distintas.
- **US10:** Como usuário, quero poder redefinir minha senha por e-mail.

### 3.5.3 Incremento 3 - Cadastro de Empresa, Instrutores e Alunos

O objetivo desse incremento é implementar o módulo básico de cadastros, permitindo o vínculo entre empresa e usuários. Com duração estimada de três *sprints*, serão implementadas as seguintes funcionalidades:

- **US11:** Como administrador da empresa, quero cadastrar as informações da minha empresa (nome, endereço, contato).
- **US12:** Como administrador da empresa, quero cadastrar instrutores vinculados à minha empresa.
- **US13:** Como administrador da empresa, quero cadastrar alunos vinculados à minha empresa.
- **US14:** Como gestor, quero visualizar a lista de instrutores e alunos cadastrados.
- **US15:** Como administrador do sistema, quero visualizar todas as empresas cadastradas.

### 3.5.4 Incremento 4 - Módulo de Turmas e Agendamentos

O objetivo desse incremento é estruturar o gerenciamento de turmas, aulas e agendamentos. Com duração estimada de três *sprints*, serão implementadas as seguintes funcionalidades:

- **US16:** Como gestor, quero criar turmas e associar instrutores a elas.
- **US17:** Como instrutor, quero visualizar as turmas nas quais leciono.
- **US18:** Como aluno, quero visualizar o horário das minhas aulas.
- **US19:** Como gestor, quero editar e excluir turmas existentes.
- **US20:** Como instrutor, quero marcar reagendamentos de aulas conforme disponibilidade.

### 3.5.5 Incremento 5 - Registro de Presença e Visualização de Horários

O objetivo desse incremento é permitir o controle de presença e acompanhamento de frequência dos alunos. Com duração estimada de três *sprints*, serão implementadas as seguintes funcionalidades:

- **US21:** Como instrutor, quero registrar presença e ausência dos alunos em cada aula.
- **US22:** Como gestor, quero consultar relatórios de presença por turma.
- **US23:** Como aluno, quero visualizar meu histórico de presenças e faltas.
- **US24:** Como aluno, quero solicitar reposição de aula quando estiver ausente.

### 3.5.6 Incremento 6 - Módulo Financeiro Básico

O objetivo desse incremento é registrar e consultar pagamentos de forma simplificada. Com duração estimada de dois *sprints*, serão implementadas as seguintes funcionalidades:

- **US25:** Como gestor, quero registrar pagamentos realizados pelos alunos.
- **US26:** Como aluno, quero visualizar o status dos meus pagamentos.
- **US27:** Como administrador da empresa, quero emitir um relatório simples de pagamentos.

### 3.5.7 Incremento 7 - Módulo de Materiais de Apoio

O objetivo desse incremento é permitir upload e acesso a conteúdos compartilhados pelos instrutores. Com duração estimada de dois *sprints*, serão implementadas as seguintes funcionalidades:

- **US28:** Como instrutor, quero fazer upload de arquivos (vídeos, PDFs ou imagens) para minha turma.
- **US29:** Como aluno, quero acessar os materiais de apoio das turmas em que estou matriculado.
- **US30:** Como sistema, quero restringir o acesso aos materiais apenas a usuários autenticados da turma.

### 3.5.8 Incremento 8 - Testes e Refinamento do MVP

O objetivo desse incremento é realizar validação funcional e ajustes de usabilidade. Com duração estimada de dois *sprints*, serão implementadas as seguintes atividades:

- **US31:** Como desenvolvedor, quero testar todos os fluxos críticos (login, cadastro, agendamento, pagamentos).
- **US32:** Como usuário, quero navegar pelo sistema sem erros visuais ou quebras de layout.
- **US33:** Como cliente voluntário, quero avaliar a usabilidade geral e relatar eventuais melhorias.

### 3.6 IMPACTO SOCIAL

Conforme observa Castells (2006), vivemos em uma sociedade em rede, onde o acesso à informação e ao conhecimento torna-se elemento central da inclusão e da competitividade.

Os impactos sociais decorrentes do desenvolvimento e uso desses sistemas podem ser analisados sob diferentes perspectivas complementares:

1. **Inclusão Digital e Redução de Desigualdades:** Segundo Warschauer (2003), a inclusão digital não se limita ao acesso físico à tecnologia, mas envolve também a capacidade de utilizá-la de forma significativa e produtiva. No caso de sistemas ERP direcionados a pequenos empreendedores, a disponibilização de ferramentas simplificadas e adaptadas pode contribuir para reduzir desigualdades estruturais, possivelmente ampliando as oportunidades de gestão eficiente e competitiva. De acordo com Castells (2006), a tecnologia tem potencial de atuar como um fator democratizador, ao descentralizar o acesso à informação e à automação.
2. **Eficiência e Produtividade:** Pressman e Maxim (2016) observam que a automação possibilita não apenas ganhos de produtividade, mas também a reorientação de esforços humanos para atividades de maior valor cognitivo e criativo. No caso analisado, a centralização de dados de turmas, pagamentos e agendamentos pode contribuir para que gestores e instrutores possam dedicar mais tempo à qualidade pedagógica e ao relacionamento com os alunos, em vez de tarefas administrativas repetitivas.
3. **Impactos Culturais e Comunitários:** O desenvolvimento de sistemas voltados a nichos culturais ou esportivos pode contribuir para a preservação e fortalecimento de práticas comunitárias. Geertz (1973) argumenta que o valor simbólico de práticas culturais se mantém e se transforma à medida que novas formas de mediação — como plataformas digitais — possibilitam sua reinterpretação e difusão. Assim, uma aplicação voltada a estúdios e escolas de dança e esportes, ao facilitar a comunicação entre praticantes e a divulgação dessas atividades, pode acabar colaborando para o fortalecimento de comunidades locais, a ampliação da visibilidade cultural e a valorização de práticas muitas vezes marginalizadas no discurso social.
4. **Sustentabilidade e Responsabilidade Social:** Segundo Elkington (1998), o conceito de *triple bottom line* (econômico, social e ambiental) reflete a necessidade de que as inovações tecnológicas equilibrem desempenho econômico e responsabilidade ambiental. A adoção de soluções digitais em pequenas empresas pode acabar contribuindo para práticas mais sustentáveis e transparentes, reduzindo desperdícios e promovendo maior controle sobre recursos.
5. **Troca de Experiências e Conhecimento:** A integração entre alunos e instrutores em um ambiente digital pode facilitar o compartilhamento de experiências, saberes e boas práticas, criando uma rede de aprendizado colaborativo. Segundo Wenger (1998), comunidades de prática se formam quando indivíduos com interesses comuns interagem continuamente, trocando conhecimento e fortalecendo sua identidade coletiva. No contexto da plataforma, essa dinâmica pode ocorrer por meio de *chats*, grupos de discussão e compartilhamento de conteúdos educativos,

possivelmente fortalecendo o senso de comunidade e o aprendizado contínuo entre os participantes.

6. **Ambiente Online Seguro:** Pfleeger e Atlee (2010) destacam que a segurança em sistemas de informação deve abranger não apenas aspectos técnicos, mas também sociais, garantindo proteção contra abusos e uso indevido de dados. Nesse sentido, a proposta de controle de acesso restrito, moderação interna e políticas de privacidade alinhadas à LGPD (2018) busca promover a confiança entre os usuários e evitar situações de exposição indevida ou assédio. Um ambiente protegido tende a favorecer a participação, o engajamento e o senso de pertencimento.
7. **Inclusão:** A plataforma têm o potencial de favorecer a inclusão social e o sentimento de pertencimento entre os praticantes do esporte. Ao conectar alunos de diferentes níveis, origens e experiências, o sistema pode contribuir para a criação de um espaço que valoriza e promove interações respeitadas. Conforme Sen (2000), a inclusão social está intrinsecamente ligada à ampliação das liberdades individuais e à capacidade de participação plena na vida comunitária, objetivos que podem ser reforçados pelo uso ético e acessível de tecnologias digitais.

### 3.7 CONCLUSÃO

De forma geral, os resultados obtidos demonstram uma coerência metodológica entre as etapas de levantamento, modelagem e planejamento, em que cada fase contribuiu para o refinamento e amadurecimento da proposta. Os questionários e entrevistas forneceram dados sobre o contexto operacional do público-alvo, os BMCs transformaram essas informações em uma visão estratégica do negócio, os diagramas UML consolidaram o modelo conceitual e comportamental do sistema e o plano de desenvolvimento traduziu esses elementos em uma proposta prática e escalonável para a construção do MVP.

Essa interligação entre etapas metodológicas favoreceu que as decisões tomadas fossem baseadas em evidências e alinhadas às necessidades reais dos usuários, conforme recomendam Wieggers e Beatty (2013). Assim, o conjunto de resultados apresentados neste capítulo, constitui o produto das atividades desenvolvidas e a base técnica para o desenvolvimento futuro da solução completa.

## 4 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou as etapas de levantamento de requisitos, análise de modelo de negócio de design de um sistema ERP voltado para tentar atender às necessidades de estúdios/escolas de esportes e dança. A proposta surgiu da identificação de desafios enfrentados por esses negócios, como a dificuldade de gerenciar turmas, agendamentos, pagamentos e interação com os alunos de forma eficiente.

Ao longo do desenvolvimento do MVP, é sugerida a adoção de uma abordagem baseada em práticas ágeis para organizar o trabalho, o que futuramente facilitará a entrega incremental de novas funcionalidades e a incorporação de *feedback* de usuários reais. O sistema poderá ser implementado com tecnologias, ferramentas e técnicas de desenvolvimento modernas, o que contribuiria para sua robustez, escalabilidade e segurança na gestão das informações em conformidade com a LGPD (2018).

Os testes serão realizados com usuários reais, incluindo alunos, instrutores e administradores de um estúdio piloto, permitindo validar a eficácia do sistema em resolver problemas práticos e otimizar processos operacionais. Entre os principais possíveis benefícios identificados, destacam-se a redução de erros manuais, a automação de tarefas administrativas, a centralização de informações e a criação de um ambiente seguro para a interação entre alunos e instrutores.

O sistema também terá o potencial de promover contribuições sociais, através a inclusão digital de pequenos empreendedores do setor esportivo e artístico, especialmente do nicho do pole dance, incentivando a troca de experiências e do senso de comunidade dos praticantes. Para implementação futura, sugere-se a ampliação do sistema, como integração com mais sistemas de pagamento diversificados.

## REFERÊNCIAS

- Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. e Thomas, D. (2001). Manifesto for agile software development. <https://agilemanifesto.org/>.
- Canva (2024). Canva - create beautiful designs & documents for free. <https://www.canva.com/>.
- Castells, M. (2006). *A Sociedade em Rede*. Paz e Terra, Rio de Janeiro, 9 edition.
- Elkington, J. (1998). *Cannibals with Forks: The Triple Bottom Line of 21st Century Business*. New Society Publishers, Gabriola Island, BC and Stony Creek, CT.
- Elmasri, R. e Navathe, S. B. (2015). *Fundamentos de Sistemas de Banco de Dados*. Pearson, 7 edition.
- Forms, G. (2024). Google forms - create online forms and surveys with google. <https://www.google.com/forms/about/>.
- Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3 edition.
- Gamma, E., Helm, R., Johnson, R. e Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Geertz, C. (1973). *The Interpretation of Cultures*. Basic Books, New York.
- Git (2024). Git - distributed version control system. <https://git-scm.com/>.
- GitHub, I. (2024). Github: Where the world builds software. <https://github.com/>.
- Group, O. M. (2024). Unified modeling language (uml) – object management group. <https://www.omg.org/spec/UML/>.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison-Wesley, 1 edition.
- JavaScript (2024). Javascript. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- Ladas, C. (2009). *Scrumban - Essays on Kanban Systems for Lean Software Development*. Modus Cooperandi Press.
- Larman, C. (2004). *Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Processo Unificado*. Bookman, 3 edition.

- LGPD (2018). Lei geral de proteção de dados pessoais (lgpd) - lei nº 13.709/2018. <https://www.gov.br/secretariageral/pt-br/assuntos/lei-geral-de-protECAo-de-dados-lgpd>.
- Marconi, M. d. A. e Lakatos, E. M. (2017). *Fundamentos de metodologia científica*. Atlas, São Paulo, 7 edition.
- Maziero, C. (2025). Modelo latex de dissertação/tese do ppginf/ufpr. <https://gitlab.c3sl.ufpr.br/maziero/tese>.
- Neon (2024). Neon - postgresql as a service. <https://neon.tech/>.
- Next.js (2024). Next.js. <https://nextjs.org/>.
- Node.js (2024). Node.js. <https://nodejs.org/>.
- Osterwalder, A. e Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley.
- Osterwalder, A., Pigneur, Y., Bernarda, G. e Smith, A. (2014). *Value Proposition Design: How to Create Products and Services Customers Want*. Wiley.
- Pfleeger, S. L. e Atlee, J. M. (2010). *Software Engineering: Theory and Practice*. Prentice Hall, 4 edition.
- PlantText (2024). Planttext - free online plantuml editor. <https://www.planttext.com/>.
- PlantUML (2024). Plantuml - open source tool to create uml diagrams from plain text. <https://plantuml.com/>.
- PostgreSQL (2024). Postgresql: The world's most advanced open source relational database. <https://www.postgresql.org/>.
- Pressman, R. S. e Maxim, B. R. (2016). *Engenharia de software: uma abordagem profissional*. AMGH, 8 edition.
- React (2024). React. <https://react.dev/>.
- Rumbaugh, J., Jacobson, I. e Booch, G. (2005). *The Unified Modeling Language Reference Manual*. Addison-Wesley, 2 edition.
- Schwaber, K. e Sutherland, J. (2020). *The Scrum Guide*. Scrum.org, 2020 edition.
- Scrum.org (2024). Scrum.org - home of scrum. <https://scrum.org/>.
- SEBRAE (2025). Digitalização: recorde de pequenos negócios no brasil atingem nível histórico em 2025. <https://agenciasebrae.com.br/inovacao-e-tecnologia/digitalizacao-recorde-pequenos-negocios-no-brasil-atingem-nivel-historico-em-2025/>.
- Sen, A. (2000). *Development as Freedom*. Anchor Books.
- Sommerville, I. (2011). *Engenharia de Software*. Pearson, 9 edition.

- Spinellis, D. (2012). *Code Quality: The Open Source Perspective*. Addison-Wesley.
- Tanenbaum, A. S. e Van Steen, M. (2017). *Distributed Systems: Principles and Paradigms*. Pearson, 3 edition.
- TypeScript (2024). Typescript: Javascript with syntax for types. <https://www.typescriptlang.org/>.
- UFPR, B. (2015). Manual de normalização de documentos científicos de acordo com as normas da ABNT. Relatório Técnico ISBN 9788584800025, Sistema de Bibliotecas – Universidade Federal do Paraná, Curitiba PR.
- Vercel (2024). Vercel - develop. preview. ship. <https://vercel.com/>.
- Warschauer, M. (2003). *Technology and Social Inclusion: Rethinking the Digital Divide*. MIT Press.
- Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press.
- Wiegers, K. E. e Beatty, J. (2013). *Software Requirements*. Microsoft Press, 3 edition.

## APÊNDICE A – QUESTIONÁRIO AOS ADMINISTRADORES E INSTRUTORES

Este apêndice apresenta, de forma detalhada, os resultados obtidos por meio do questionário estruturado aplicado aos administradores e instrutores do cliente voluntário (dois participantes), com o objetivo de melhor compreender as rotinas administrativas, desafios, práticas de gestão e percepções sobre o uso de tecnologia nos estúdios e escolas de esportes e dança.

O questionário foi composto por doze perguntas, sendo majoritariamente fechadas (de múltipla escolha ou seleção simples), com algumas questões abertas para coleta de observações descritivas e sugestões. A aplicação ocorreu de forma online e anônima, entre os dias sete de abril de dois mil e vinte e cinco e sete de junho de dois mil e vinte e cinco, e os resultados estão apresentados individualmente para cada questão, acompanhados de alguns gráficos e representações visuais que ilustram as proporções e tendências observadas.

A seguir, são apresentados os resultados de cada pergunta, com uma breve descrição interpretativa e, quando pertinente, exemplos de respostas abertas fornecidas pelos participantes. A ferramenta utilizada para a aplicação do questionário e para a criação dos gráficos foi o Forms (2024).

1. **Cargo/Função.** Esta questão buscou identificar o papel de cada participante dentro do estúdio ou escola. Um deles exerce a função de proprietário, enquanto o outro atua como instrutor.
2. **Quais atividades administrativas você realiza com maior frequência?**

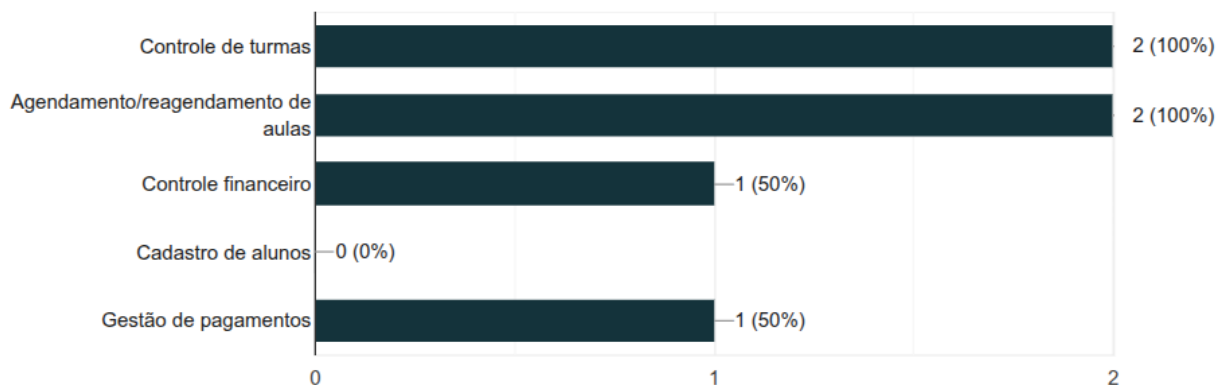


Figura A.1: Respostas à questão dois do questionário aplicado aos administradores e instrutores.

As atividades administrativas mais frequentes incluem agendamento de aulas, controle financeiro e gestão de pagamentos, refletindo as principais demandas operacionais enfrentadas pelos administradores e instrutores no dia a dia.

3. **Você utiliza algum software ou ferramenta para gerenciar essas atividades?** Esta questão investigou o nível atual de digitalização. As duas participantes responderam que utilizam planilhas eletrônicas e aplicativos de gerencia de atividades para gerenciar suas atividades administrativas, indicando uma adoção parcial de ferramentas digitais, mas sem o uso de sistemas integrados de gestão.

#### 4. Quais desafios você enfrenta atualmente nesses processos?

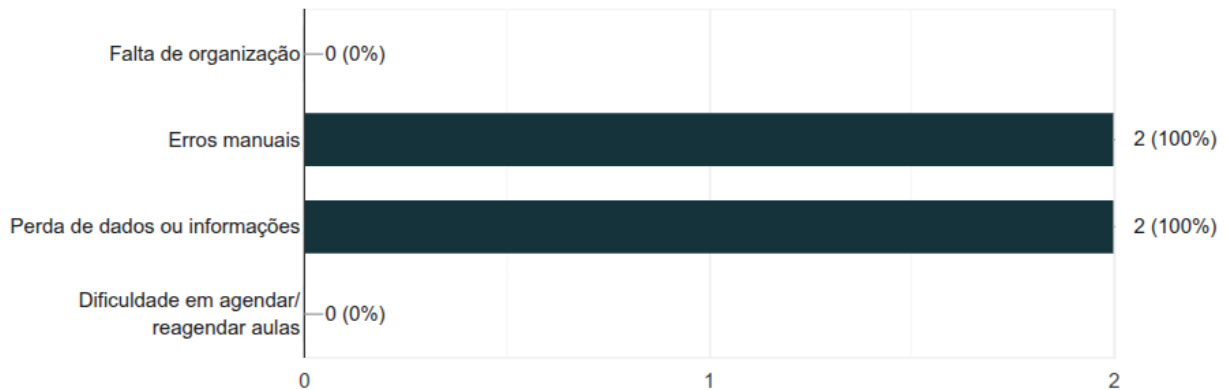


Figura A.2: Respostas à questão quatro do questionário aplicado aos administradores e instrutores.

Os desafios mais recorrentes identificados foram erros manuais e perda de dados ou informações, confirmando a carência de ferramentas integradas de gestão no setor.

#### 5. Há alguma funcionalidade específica que você gostaria que o sistema oferecesse? (Descreva brevemente). Tendo as seguintes respostas:

- "Cálculo de pagamento da professoras automatizado, por porcentagem.";
- "Reagendamento automático direto pelo aluno, envio de lembrete de aula. Hoje fazemos tudo manualmente.".

Ambas as respostas destacam a necessidade de automação de processos administrativos, como cálculo de pagamentos, agendamento e reagendamento de aulas, reforçando a importância dessas funcionalidades no design do sistema proposto.

#### 6. Como você organiza a comunicação com os alunos? A comunicação ocorre quase que unicamente via aplicativos de mensagem de texto. Este dado evidencia a importância de integrar canais de comunicação, notificação e tomada de decisão em tempo real ao sistema proposto, nas interações entre os alunos e os instrutores/administradores.

#### 7. Você gostaria de um espaço exclusivo na plataforma para interação com os alunos (em grupo e individualmente)?

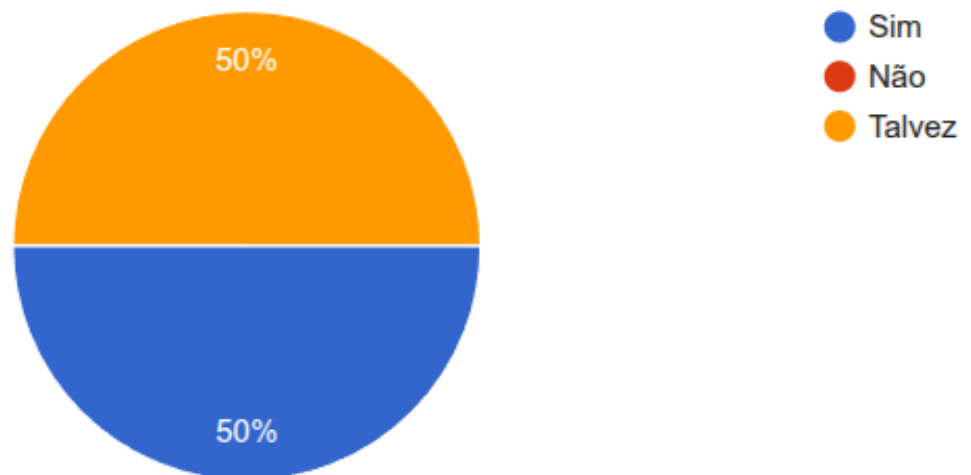


Figura A.3: Respostas à questão sete do questionário aplicado aos administradores e instrutores.

Um participante respondeu que "sim", destacando a importância de um canal oficial e seguro para comunicação, enquanto o outro respondeu que "talvez". Ainda assim, um recurso de notificação e interação em tempo real permanece fundamental para o design do sistema.

8. **Você gostaria de compartilhar conteúdos com os alunos por meio do sistema (ex.: vídeos, fotos, comunicados)?** Todos os respondentes consideraram útil ou muito relevante poder compartilhar vídeos, fotos e comunicados diretamente com os alunos, o que reforça o potencial pedagógico e de engajamento da solução proposta.
9. **Quais medidas de segurança são importantes para você em uma plataforma online? (além de usuário com senha).**



Figura A.4: Respostas à questão nove do questionário aplicado aos administradores e instrutores.

Todas as opções de segurança foram marcadas pelos dois participantes. Isso indica uma preocupação crescente com a segurança e privacidade digital, aspectos fundamentais no design do sistema.

10. **Você acredita que um ambiente seguro para compartilhamento de mídia ajudaria a melhorar a interação com os alunos? Por quê?** Ambos os participantes reconheceram o valor de um ambiente seguro para compartilhamento

de mídia, destacando benefícios relacionados à organização do conteúdo, ao acompanhamento da evolução dos alunos e à criação de um espaço online inclusivo e acolhedor.

11. **Se pudesse melhorar um único aspecto da gestão do estúdio/escola com o sistema, qual seria?** Com as respostas:

- "Facilidade em as próprias alunas agendarem suas reposições.";
- "Envio de notificações automáticas para alunas.".

As respostas abertas apontaram, com maior frequência, aspectos da comunicação em tempo real facilitada, agendamento e reagendamento de aulas como áreas prioritárias para melhoria.

12. **Está disposto a testar um protótipo ou versão inicial do sistema?** Todos os participantes demonstraram interesse em testar uma versão inicial do sistema, evidenciando abertura à inovação e disposição para contribuir nas próximas fases do projeto.

## APÊNDICE B – QUESTIONÁRIO AOS ALUNOS

Este apêndice apresenta, de forma detalhada, os resultados obtidos por meio do questionário estruturado aplicado aos alunos praticantes vinculados ao cliente voluntário (dez participantes), com o objetivo de compreender suas experiências, hábitos de uso de ferramentas digitais, principais dificuldades e expectativas em relação a uma plataforma de gestão de aulas e pagamentos.

O questionário foi composto por sete perguntas, sendo majoritariamente fechadas (de múltipla escolha ou seleção simples), com algumas questões abertas voltadas à coleta de percepções individuais e sugestões. A aplicação ocorreu de forma online e anônima, entre os dias sete de abril de dois mil e vinte e cinco e sete de junho de dois mil e vinte e cinco, assegurando a espontaneidade das respostas.

Os resultados estão apresentados individualmente para cada questão, acompanhados de alguns gráficos e representações visuais que ilustram as proporções, preferências e tendências observadas entre os participantes. Quando pertinente, também são incluídos exemplos de respostas abertas, que ajudam a contextualizar a percepção dos alunos sobre a organização das aulas, os meios de comunicação utilizados e a importância de um ambiente digital mais seguro e centralizado. A ferramenta utilizada para a aplicação do questionário e para a criação dos gráficos foi o Forms (2024).

### 1. Qual atividade você pratica no estúdio/escola?

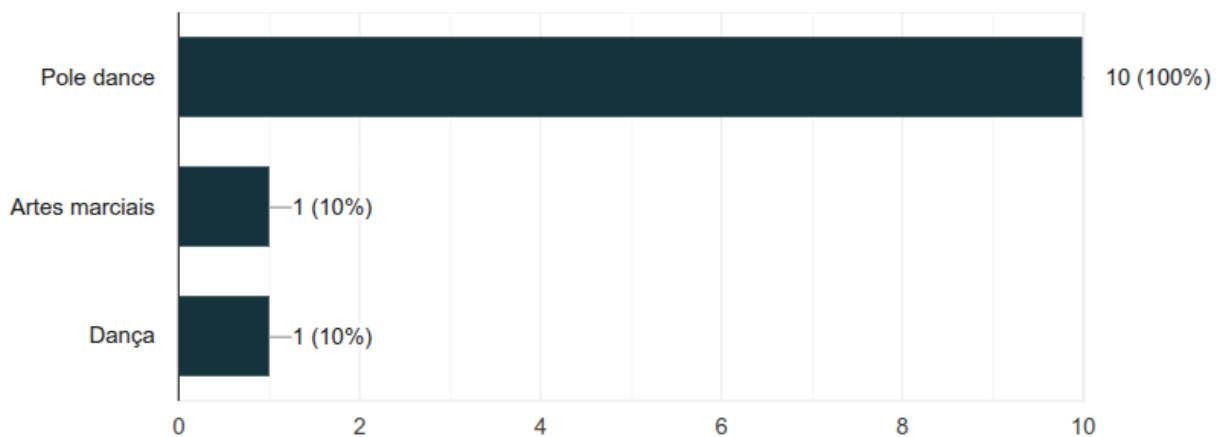


Figura B.1: Respostas à questão um do questionário aplicado aos alunos praticantes.

Esta questão buscou identificar a diversidade de atividades praticadas pelos alunos no estúdio/escola. As respostas indicam que todos os participantes praticam pole dance e vinte por cento estão envolvidos em outras modalidades, como artes marciais e dança.

### 2. Como você costuma agendar suas aulas/reposições de aula?

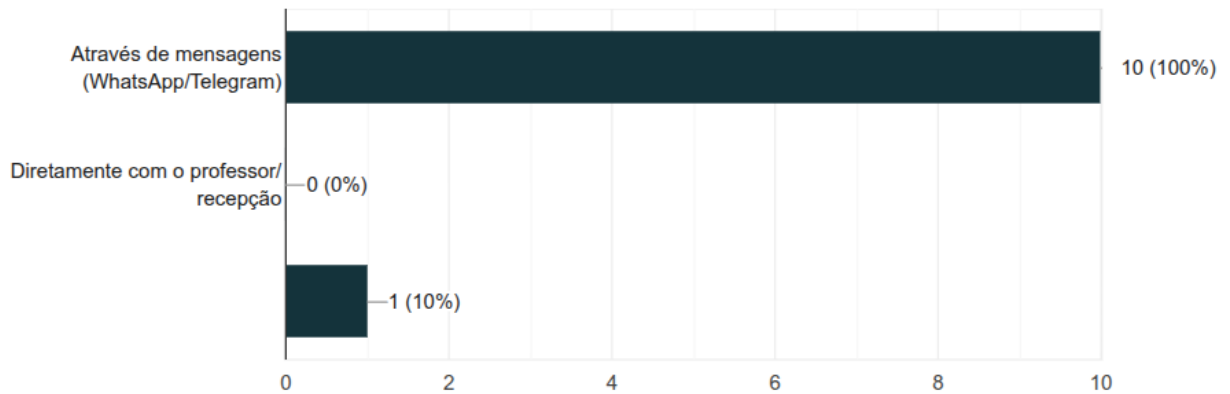


Figura B.2: Respostas à questão dois do questionário aplicado aos alunos praticantes.

Todos os alunos realizam o agendamento de aulas e reposições por meio de aplicativos de mensagem de texto. Esse dado evidencia a ausência de uma plataforma digital integrada para gerenciamento de aulas, reforçando a necessidade de um sistema que centralize essas operações.

### 3. Quais problemas você já enfrentou na organização de aulas?

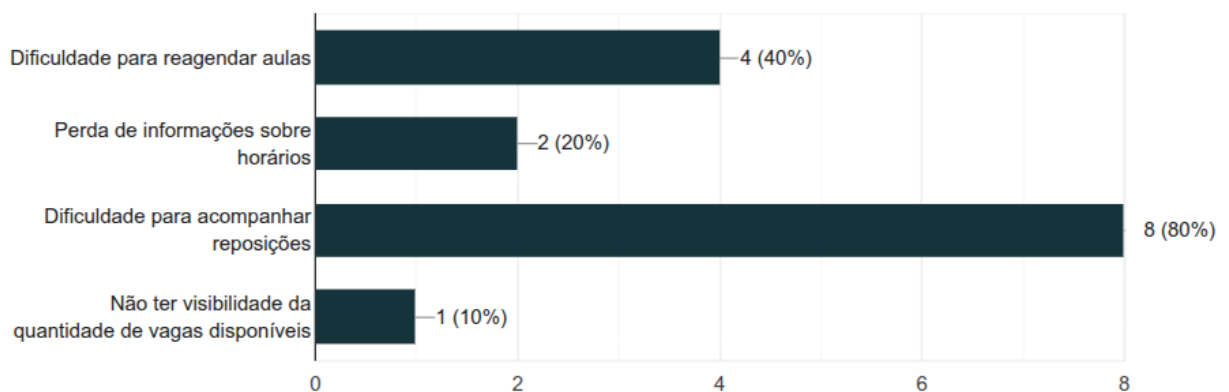


Figura B.3: Respostas à questão três do questionário aplicado aos alunos praticantes.

oitenta por cento dos participantes relatam dificuldade para acompanhar reposições de aulas, quarenta por cento alega dificuldade para reagendar aulas, vinte por cento disse ocorrer perda de informações sobre horários e dez por cento mencionou a falta de visibilidade da quantidade de vagas disponíveis nas turmas. Esses desafios destacam a necessidade de uma solução digital que facilite o gerenciamento e o controle das turmas.

4. **Como você prefere receber informações ou comunicados do estúdio?**  
Aqui, oitenta por cento dos participantes indicam preferência por aplicativos de mensagem de texto, enquanto quarenta por cento também dizem preferir uma hipotética solução online centralizadora. dez por cento mencionou e-mail como canal de comunicação.
5. **Você gostaria de ter acesso a um sistema para.**

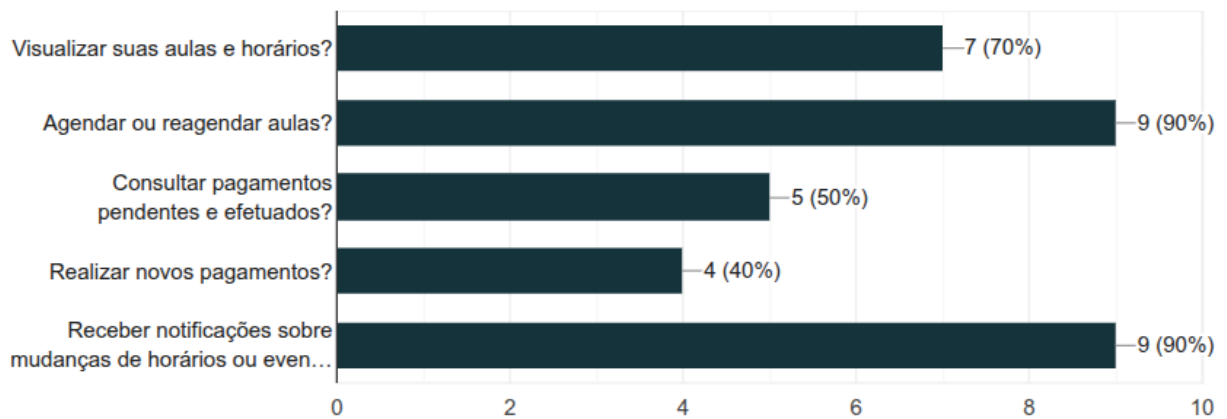


Figura B.4: Respostas à questão cinco do questionário aplicado aos alunos praticantes.

Os participantes demonstram grande interesse em todas as funcionalidades propostas, com destaque para os recursos de notificações, agendamento e reagendamento de aulas (noventa por cento), visualização de agenda de turmas (setenta por cento), consulta a pagamentos pendentes e efetuados (cinquenta por cento) e realização de pagamentos (quarenta por cento). Esses dados reforçam a demanda por uma plataforma digital que ofereça todos esses recursos.

#### 6. Quais tipos de conteúdo você gostaria de acessar na plataforma?

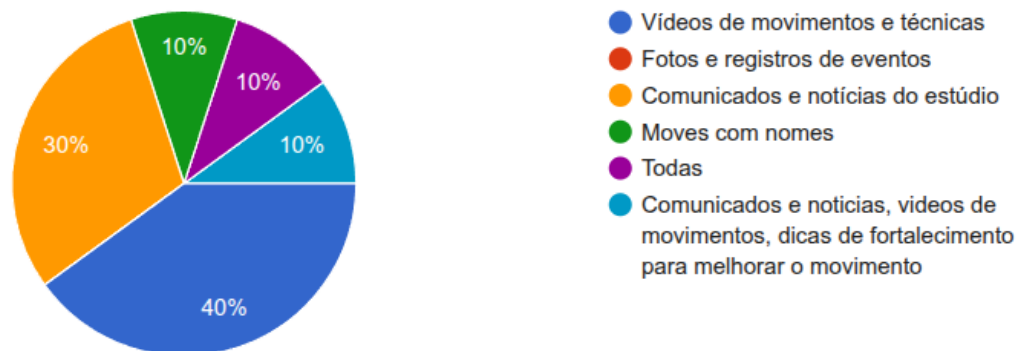


Figura B.5: Respostas à questão seis do questionário aplicado aos alunos praticantes.

Esses resultados indicam que o acesso a comunicados, notícias do estabelecimento e conteúdos multimídia são funcionalidades valorizadas pelos alunos.

#### 7. O que você acha que poderia melhorar na sua experiência como aluno do estúdio/escola? As respostas destacam diversos aspectos, como:

- "Visibilidade de vaga em outros horários de aulas";
- "Maior agilidade para agendar aulas e reposições, em um sistema mais organizado com calendário e facilidade na hora de usar.";
- "Gostaria de poder ver minhas aulas no mês e anotar coisas como desempenho e movimentos realizados.";
- "Ter as regras e contrato do estúdio com facil acesso para consulta de quando posso ou nao repor aulas, taxas, etc.";

- "Sistema com reagendamento, controle de pagamentos online.";
- "Facilidade com comunicados gerais.";
- "Acompanhar as notícias do estúdio e reagendar com mais praticidade as aulas.".

Essas sugestões reforçam a importância de uma plataforma digital que centralize informações, facilite o agendamento e o controle financeiro, além de melhorar a comunicação entre alunos e estúdio.