

Universidade Tiradentes
CIÊNCIA DA COMPUTAÇÃO

Ana Sofia Ribeiro de Meneses e Rocha Almeida
Heitor Fernando Feitosa Vieira
Ianca Carolinne Carregosa Silva
Luan Gouveia Falcão dos Anjos
Mateus da Silva Machado
Rayelen de Jesus Oliveira

Documento de Especificação de Software da Plataforma de
Gerenciamento de Cursos e Treinamentos de Vértida

Projeto técnico desenvolvido como parte da disciplina de
Projeto de Programação

Ana Sofia Ribeiro de Meneses e Rocha Almeida
Heitor Fernando Feitosa Vieira
Ianca Carolinne Carregosa Silva
Luan Gouveia Falcão dos Anjos
Mateus da Silva Machado
Rayelen de Jesus Oliveira

**Documento de Especificação de Software da Plataforma de
Gerenciamento de Cursos e Treinamentos de Vértida**

**Projeto técnico desenvolvido como parte da disciplina de
Engenharia de Software**

Atividade sobre desenvolvimento de software de plataforma de gerenciamento de cursos e treinamentos do município de Vértida, apresentada como requisito parcial da avaliação da disciplina Projeto de Programação, ministrada pela Prof.^a Layse Santos Souza, no 2º semestre de 2025.

Sumário

1	INTRODUÇÃO	4
2	JUSTIFICATIVA	5
3	OBJETIVOS	6
3.1	OBJETIVO GERAL	6
3.2	OBJETIVOS ESPECÍFICOS	6
4	METODOLOGIA	7
4.1	FASES DO PROJETO	7
4.2	REQUISITOS DO SISTEMA	8
4.2.1	Requisitos Funcionais	8
4.2.2	Requisitos Não Funcionais	9
5	RESULTADOS E DISCUSSÕES	11
5.1	MODELAGEM DE CASOS DE USO	11
5.1.1	Ator Aluno	11
5.1.2	Ator Instrutor	11
5.1.3	Ator Administrador	12
5.2	MODELAGEM ESTRUTURAL (DIAGRAMA DE CLASSES)	13
5.3	IMPLEMENTAÇÃO DO SISTEMA	15
5.4	DISCUSSÃO DOS RESULTADOS	15
6	CONSIDERAÇÕES FINAIS	18
7	REFERÊNCIAS	19
8	ANEXOS	20

1 INTRODUÇÃO

Este documento apresenta a análise e o planejamento inicial para o desenvolvimento de um sistema de gerenciamento de cursos. O projeto visa criar uma plataforma centralizada para otimizar a operação de uma instituição de ensino, focando na automação de processos e na facilitação da interação entre usuários. Os objetivos centrais do sistema incluem facilitar o processo de matrícula e pagamento de mensalidades, automatizar processos internos, como transferências e seleção de cursos, organizar e armazenar de forma centralizada os dados de funcionários e alunos, Gerenciar cancelamentos de inscrição e garantir o reembolso quando aplicável. Além disso, o sistema busca promover maior transparência nos ganhos mensais da instituição e estabelecer regras de negócio claras, como o limite de 60 alunos por curso e o repasse de 40% do valor da compra de cada curso ao instrutor. O público-alvo é dividido em três perfis principais, alunos, focados em se inscrever e participar dos cursos, instrutores, responsáveis por gerenciar as aulas e administradores, funcionários que gerenciam a operação geral do sistema.

2 JUSTIFICATIVA

O desenvolvimento da Plataforma de Gerenciamento de Cursos e Treinamentos para a cidade de Vértida se justifica pela grande dificuldade de organização dos pagamentos e inscrições de cursos educacionais e profissionalizantes ofertados pelo município. A dispersão de informações e a ineficiência nos processos manuais de inscrição, controle de vagas, pagamentos e comunicação representam um obstáculo significativo ao acesso da população a oportunidades de capacitação de instituições públicas e privadas.

A ausência de uma plataforma centralizada e digitalizada impacta diretamente na transparência da oferta de cursos, na dificuldade de gerenciamento por parte dos administradores e na experiência frustrante dos alunos, que não possuem um canal unificado para pagamento, inscrição e acompanhamento.

Portanto, a ideia da plataforma é mitigar essa situação. Com essa aplicação, a administração organiza as inscrições e pagamentos, os instrutores controlam melhor os cursos, e os alunos conseguem ver com clareza a situação de suas inscrições e pagamentos.

3 OBJETIVOS

3.1 OBJETIVO GERAL

O objetivo geral deste projeto é desenvolver e implementar um sistema de gerenciamento de cursos abrangente, projetado para modernizar, automatizar e centralizar os processos acadêmicos e administrativos da instituição. Através desta plataforma, busca-se otimizar todo o ciclo de vida do aluno, desde a matrícula e seleção de cursos até o processamento de pagamentos e a gestão de eventuais cancelamentos ou transferências.

3.2 OBJETIVOS ESPECÍFICOS

Para alcançar este objetivo principal, os seguintes objetivos específicos foram definidos:

1. Facilitar o processo de matrícula e pagamento de mensalidade dos alunos;
2. Facilitar o cancelamento da inscrição do aluno em cursos e garantir o seu reembolso quando possível;
3. Ter um local para organizar e armazenar os dados dos funcionários e dos alunos da instituição;
4. Automatizar processos internos, como transferência, matrícula e seleção de cursos;
5. Limitar a quantidade de alunos por curso, estabelecendo um limite de 60 alunos;
6. Promover a melhor transparência dos ganhos da instituição no respectivo mês;
7. Determinar que o instrutor fique com 40% do valor da compra de cada curso oferecido;

4 METODOLOGIA

Para a concepção deste projeto, a metodologia de trabalho foi estruturada com base em um cronograma interno dividido em quatro semanas. Cada fase foi planejada para abordar uma etapa distinta da análise e planejamento, garantindo um progresso sistemático desde a concepção até a entrega parcial.

4.1 FASES DO PROJETO

As fases do projeto foram definidas da seguinte maneira:

1. Semana 01: Levantamento de Requisitos e Definição de Escopo Fase inicial dedicada à definição dos objetivos , público-alvo , e ao levantamento dos requisitos funcionais e não funcionais.
2. Semana 02: Modelagem do Sistema Foco na modelagem dos diagramas de classe , de uso e na descrição das funcionalidades.
3. Semana 03: Elaboração do Documento Final Consolidação de todo o material produzido na elaboração do documento final.
4. Semana 04: Apresentação e Entrega Parcial Conclusão das atividades para a apresentação e entrega parcial do trabalho.
5. Semana 05: Estrutura Inicial do Sistema Objetivo de criar a base do projeto em Java, estruturando as classes e pacotes.
6. Semana 06: Regras de Negócio e Funcionalidades Internas Implementação das regras de negócio e os fluxos principais de cada módulo.
7. Semana 07: Persistência de Dados Adição de persistência em banco de dados ou arquivos.
8. Semana 08: Interface e Integração entre Módulos Integração dos módulos e criação de uma interface de interação (gráfica ou textual).
9. Semana 09: Finalização e Relatórios Fase dedicada a testar, documentar e apresentar o sistema final.
10. Semana 10: Correção de erros, ajuste da integração e entrega da versão final revisada.

4.2 REQUISITOS DO SISTEMA

Como resultado da "Semana 01: Levantamento de Requisitos", foram definidas as seguintes especificações funcionais e não funcionais que nortearam o desenvolvimento.

4.2.1 Requisitos Funcionais

Os requisitos funcionais (Tabela 1) descrevem as ações e funcionalidades que o sistema deve executar, detalhando as interações dos usuários e as operações de cada módulo.

Tabela 1 – Requisitos Funcionais do Sistema

Nº	Descrição	Módulo	Criticidade
RF001	Os alunos devem realizar a inscrição na plataforma	Aluno	Alto
RF002	Os alunos devem poder cancelar a inscrição na plataforma	Aluno	Alto
RF003	Os alunos devem escolher o curso na plataforma	Aluno	Alto
RF004	Os alunos devem efetuar o pagamento na plataforma via PIX, cartão ou boleto	Aluno	Alto
RF005	Os alunos devem poder fazer pedido de reembolso parcial ou integral do curso com base na quantidade de conteúdo consumido	Aluno	Baixo
RF006	Os alunos devem poder realizar transferência de curso	Aluno	Baixo
RF007	Os instrutores devem realizar a inscrição na plataforma	Instrutor	Alto
RF008	Os instrutores devem escolher o curso que vão administrar	Instrutor	Alto
RF009	Os instrutores podem cancelar o curso por falta de alunos, erro de pagamento, abandono ou trancamento de matrícula	Instrutor	Médio
RF010	A administração deve cadastrar os novos cursos do sistema	Administração	Alto
RF011	A administração deve gerenciar as matrículas e os pagamentos	Administração	Alto
RF012	A administração deve gerenciar as inscrições dos cursos	Administração	Alto
RF013	A administração deve criar lista de espera	Administração	Baixo
RF014	A administração deve alertar os alunos sobre a lista de espera	Administração	Baixo
RF015	A administração deve gerar relatórios sobre participação dos alunos, desempenho dos cursos e desempenho dos instrutores	Administração	Médio

4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais (Tabela 2) definem os critérios de qualidade, desempenho, segurança e restrições técnicas do sistema.

Tabela 2 – Requisitos Não Funcionais do Sistema

Nº	Descrição	Módulo	Criticidade
RNF001	O sistema deve validar o pagamento	Sistema	Alto
RNF002	O sistema deve liberar o acesso e as aulas oferecidas ao aluno na plataforma após a efetivação do pagamento	Sistema	Alto
RNF003	O sistema deve possuir alta escalabilidade para período de inscrições	Sistema	Alto
RNF004	O sistema deve realizar a validação de usuários (instrutor, aluno ou administrador)	Sistema	Alto
RNF005	O sistema deve emitir comprovante de inscrição	Sistema	Alto
RNF006	O sistema deve permitir que alunos se inscrevam até atingir o limite de vagas em cada curso	Sistema	Alto
RNF007	O sistema deve enviar notificação aos alunos e instrutores em caso de trocas de sala ou fechamento de cursos	Sistema	Baixo
RNF008	O sistema deve garantir que todos os carregamentos realizados pelo site não excedam o tempo de 3 segundos	Sistema	Médio
RNF009	O sistema deve emitir o comprovante em até 2 segundos	Sistema	Médio
RNF010	O sistema deve enviar o boleto para o e-mail do aluno	Sistema	Médio
RNF011	O sistema deve gerenciar as formas de pagamento disponíveis para o aluno	Sistema	Alto
RNF012	O sistema deve criptografar todos os dados sensíveis e obedecer à LGPD	Sistema	Médio
RNF013	O sistema deve bloquear automaticamente as inscrições quando o limite de vagas for atingido	Sistema	Alto
RNF014	O sistema deve gerenciar e analisar o desempenho do aluno com base em suas participações em aulas e notas	Sistema	Baixo
RNF015	O sistema deve oferecer suporte para diferentes sistemas operacionais	Sistema	Alto
RNF016	O sistema deve validar o status de pagamento do aluno	Sistema	Alto
RNF017	O sistema deve validar a disponibilidade de vagas nos cursos	Sistema	Alto
RNF018	O sistema deve validar os pré-requisitos dos cursos a serem realizados	Sistema	Médio
RNF019	O sistema deve possuir uma interface intuitiva para o usuário final	Sistema	Médio
RNF020	O sistema deve permitir atualizações contínuas de status de pagamento e inscrição sem interrupção de serviço	Sistema	Alto
RNF021	O sistema deve calcular o reembolso parcial ou total conforme a porcentagem do curso assistido	Sistema	Médio
RNF022	O sistema deve processar os pagamentos realizados pelos alunos	Sistema	Alto
RNF023	O sistema deve notificar o aluno sobre a conclusão da inscrição e confirmação de pagamento	Sistema	Baixo

5 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os artefatos de modelagem e a implementação do sistema, seguidos de uma discussão que conecta os resultados obtidos com os requisitos definidos na metodologia.

5.1 MODELAGEM DE CASOS DE USO

Para atender aos objetivos do projeto e traduzir as necessidades do público-alvo (Alunos, Instrutores e Administradores) em funcionalidades, a primeira etapa da modelagem consistiu na elaboração dos Diagramas de Casos de Uso, detalhados a seguir por ator.

5.1.1 Ator Aluno

O ator Aluno (Figura 1) foca nas ações de consumo e gerenciamento de seus cursos. O caso de uso "Realizar Inscrição" é central e, conforme o requisito RNF017 e RNF018, obrigatoriamente (via `<<include>>`) verifica vagas e pré-requisitos antes de "Efetuar pagamento" (RF004).

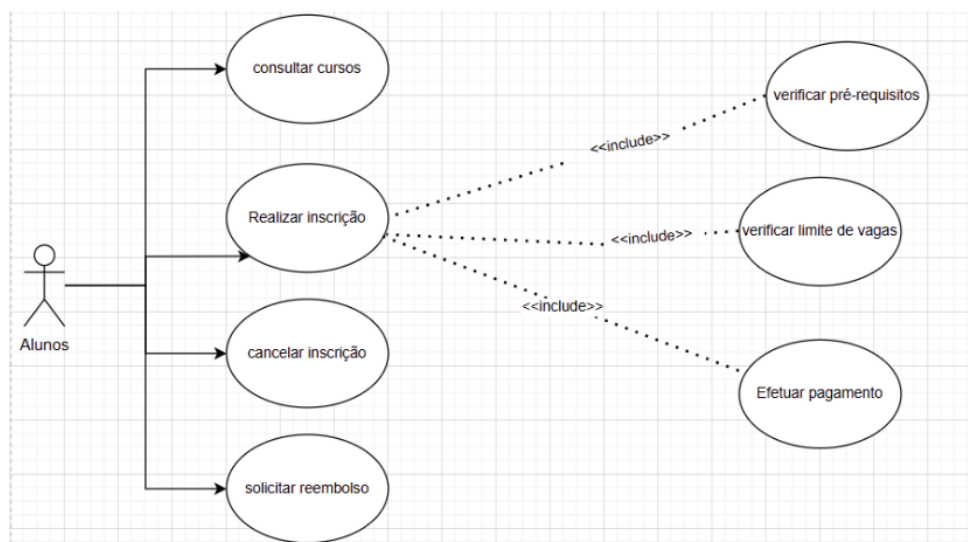


Figura 1 – Diagrama de Casos de Uso - Ator Aluno.

5.1.2 Ator Instrutor

O ator Instrutor (Figura 2) possui funcionalidades voltadas ao gerenciamento financeiro e acompanhamento de seus cursos, como "Gerenciar valores das matrículas" e "Gerenciar relatórios".

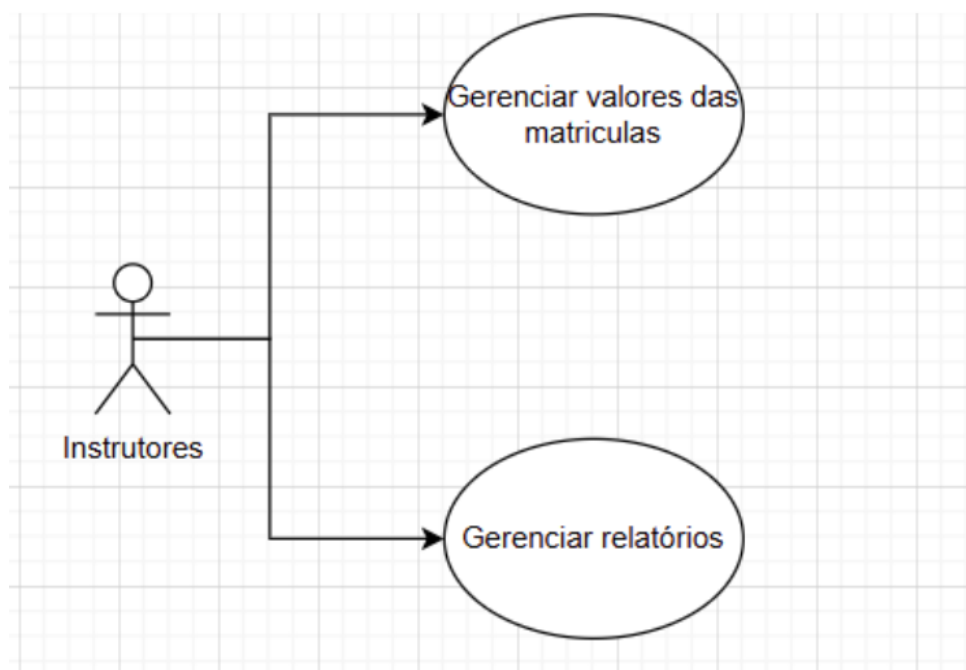


Figura 2 – Diagrama de Casos de Uso - Ator Instrutor.

5.1.3 Ator Administrador

O ator Administrador (Figura 3) centraliza o controle operacional da plataforma (Figura 3), alinhado aos requisitos RF010, RF011 e RF012 (cadastrar cursos, gerenciar matrículas e pagamentos).

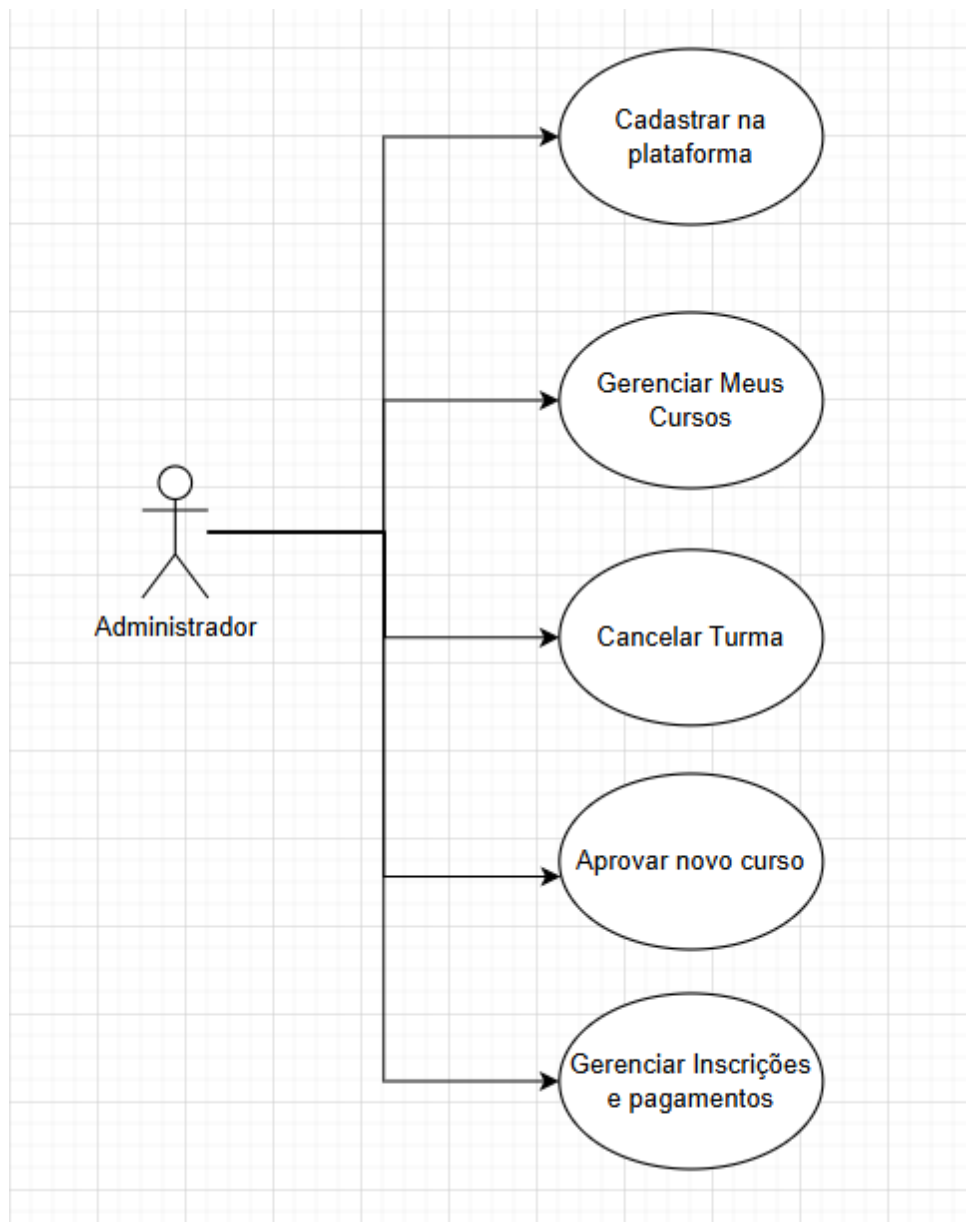


Figura 3 – Diagrama de Casos de Uso - Ator Administrador.

5.2 MODELAGEM ESTRUTURAL (DIAGRAMA DE CLASSES)

Uma vez definidas as funcionalidades macro nos casos de uso, foi necessário modelar a estrutura estática do sistema. O Diagrama de Classes (Figura 4) define as entidades centrais, seus atributos e os relacionamentos que dão suporte às regras de negócio.

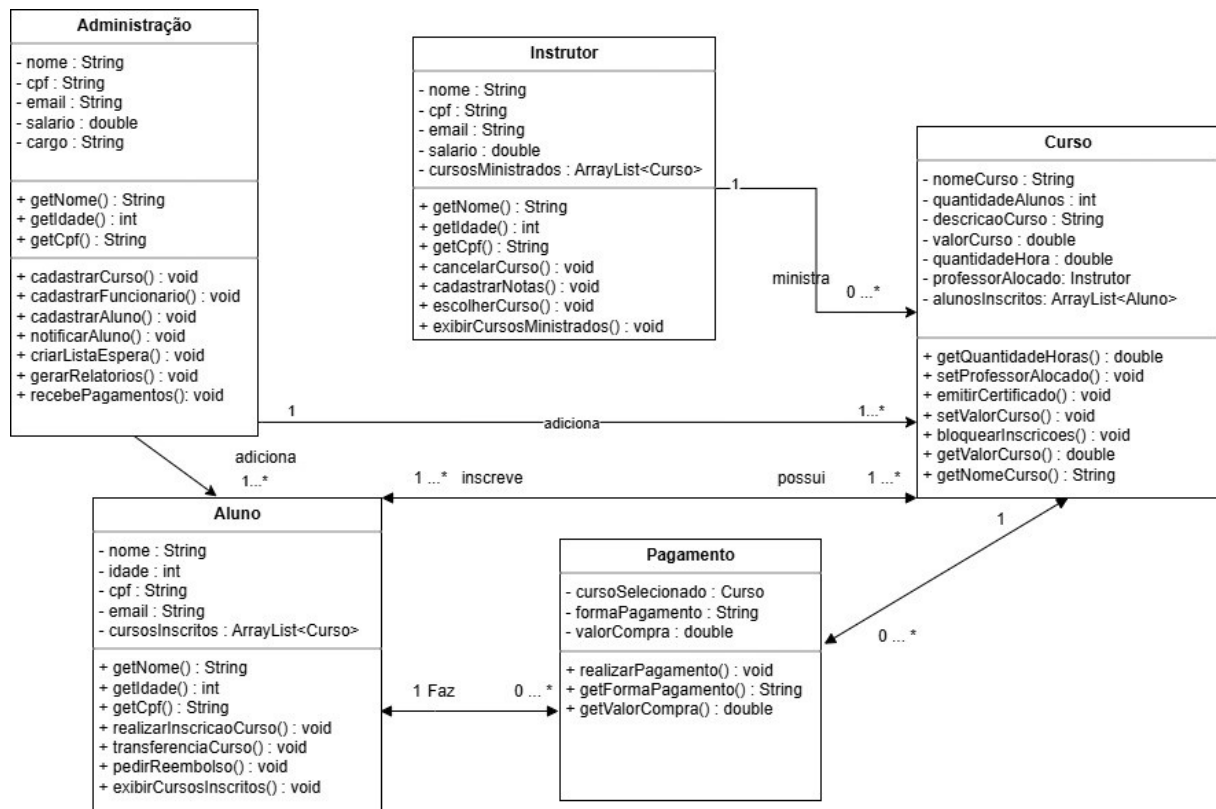


Figura 4 – Diagrama de Classes do sistema.

Como detalhado na Figura 4, o modelo gira em torno das entidades principais: **Aluno**, **Instrutor**, **Curso** e **Administração**. O diagrama estabelece as seguintes lógicas:

- Um **Instrutor** "ministra" de 0 a muitos **Cursos** (0..*), e um **Curso** é ministrado por exatamente 1 **Instrutor**. Isso atende à regra de negócio do repasse de 40instrutor responsável.
- Um **Aluno** "inscreve-se" em 1 ou muitos **Cursos** (1..*).
- Um **Aluno** "Faz" 0 ou muitos **Pagamentos** (0..*), sendo que cada pagamento está associado a 1 **Curso** selecionado. Isso formaliza a relação de compra e atende aos requisitos RF004 e RNF024 (múltiplos métodos de pagamento).
- A classe **Administração** atua como controladora do sistema, possuindo métodos para gerenciar as demais entidades, como "cadastrarCurso()" e "gerarRelatorios()", centralizando as operações do requisito RF011.

Este modelo estrutural serviu como base para a implementação do banco de dados e a construção das classes no BackEnd.

5.3 IMPLEMENTAÇÃO DO SISTEMA

O sistema de gerenciamento de cursos foi desenvolvido conforme planejado, utilizando Spring Boot 3.5.6, Spring Data JPA, Hibernate, H2 Database 2.3.232, Jackson, Lombok e Maven para o BackEnd e HTML5, CSS3, JavaScript (ES6+) e Fetch API para o FrontEnd.

As Figuras 10 a 14 apresentam as principais telas da aplicação final.

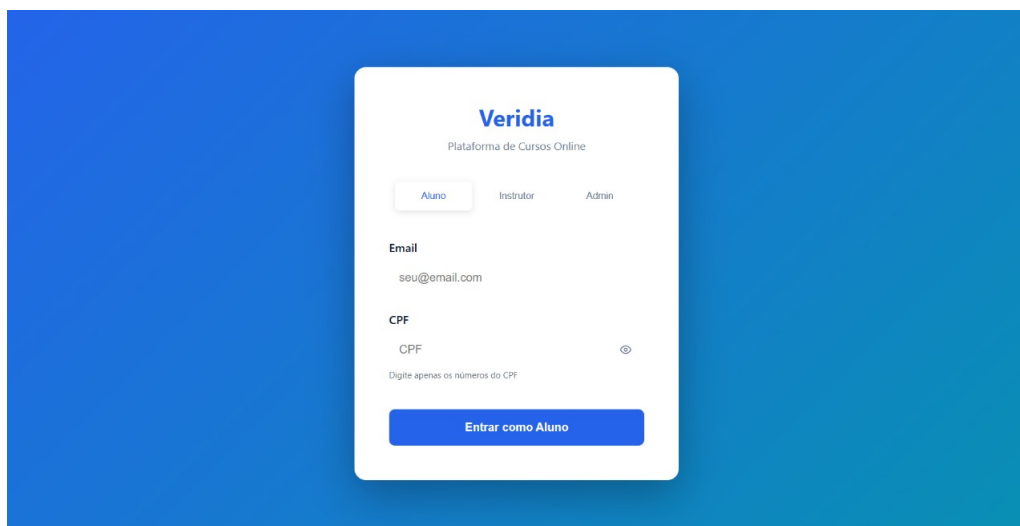


Figura 5 – TELA DE LOGIN

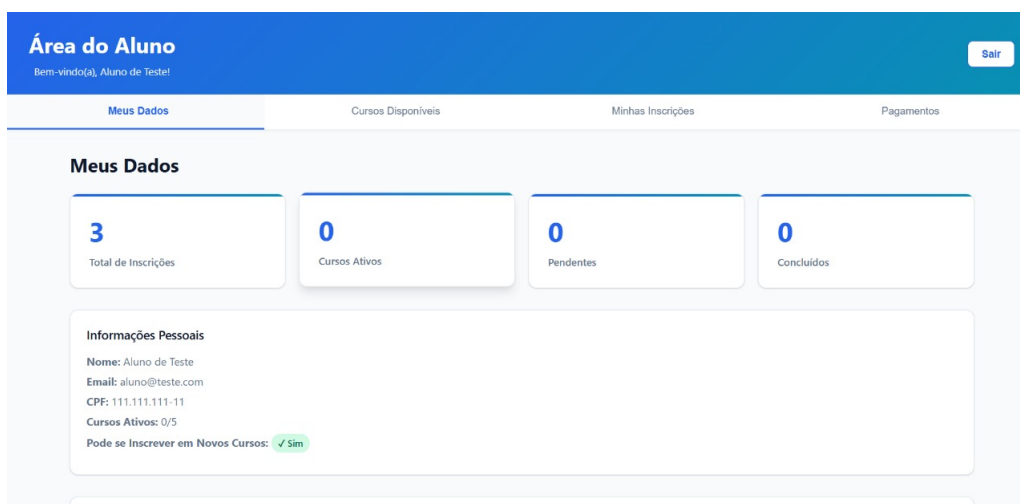


Figura 6 – DASHBOARD DE ALUNO

5.4 DISCUSSÃO DOS RESULTADOS

O objetivo geral de criar uma plataforma centralizada foi plenamente alcançado. A implementação atende diretamente aos requisitos levantados na metodologia.

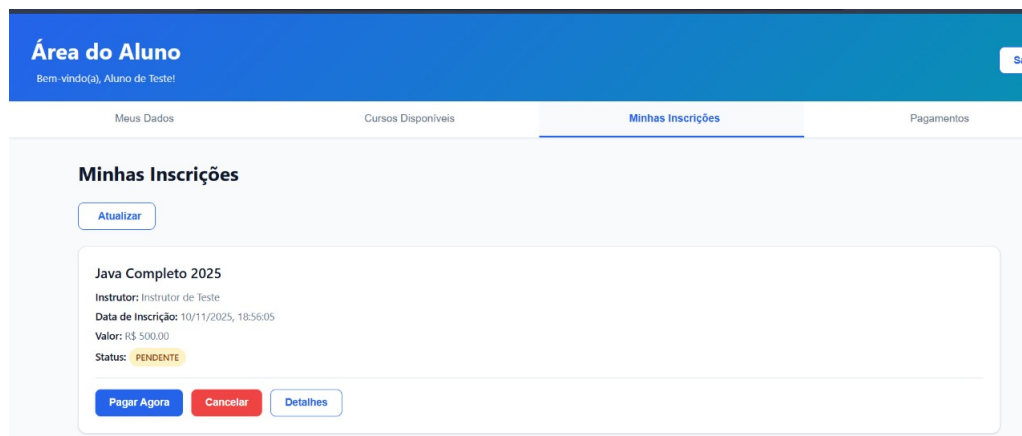


Figura 7 – MINHAS INSCRIÇÕES

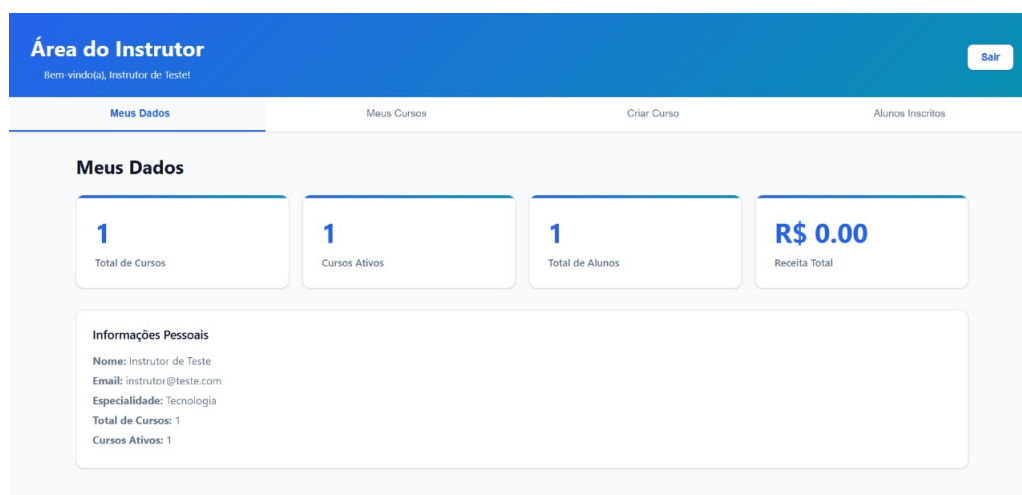


Figura 8 – DASHBOARD DE INSTRUTOR

A "Tela de Login" (Figura 10) implementa o requisito RNF004 (validação de usuários), separando os acessos para os três atores. O "Dashboard de Aluno" (Figura 11) e a tela "Minhas Inscrições" (Figura 12) são a concretização dos requisitos RF001, RF003 e RF004, permitindo ao aluno consultar cursos, se inscrever e acompanhar o status de pagamento.

O "Dashboard do Instrutor" (Figura 13) disponibiliza as ferramentas para o RF008 (escolher curso) e gera relatórios, enquanto o "Dashboard de Administrador" (Figura 14) centraliza a gestão de alunos, instrutores e cursos, atendendo aos requisitos RF010, RF011 e RF012.

O sistema final realiza, portanto, a gestão completa de alunos, instrutores e cursos, com um processo de matrícula validado e múltiplos métodos de pagamento (RNF024).

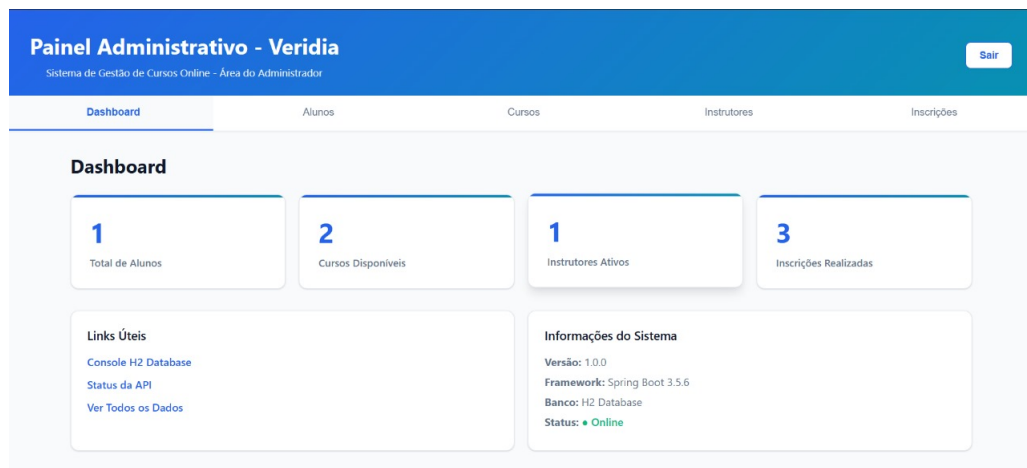


Figura 9 – DASHBOARD DE ADMINISTRADOR

6 CONSIDERAÇÕES FINAIS

A elaboração deste projeto permitiu compreender de forma mais prática como ocorre o processo de especificação e planejamento de um sistema de software. A Plataforma de Gerenciamento de Cursos e Treinamentos de Vértida mostrou-se uma solução necessária para organizar, centralizar e facilitar o acesso às informações acadêmicas, atendendo tanto alunos quanto instrutores e administradores.

Durante o desenvolvimento do documento, foi possível identificar e estruturar os requisitos, modelar os casos de uso, definir os módulos do sistema e organizar um plano de implementação coerente com as demandas do projeto. Esse processo contribuiu para ampliar o entendimento sobre a importância da análise detalhada antes do início da programação, garantindo maior clareza e direcionamento para as etapas futuras.

Além disso, a divisão do projeto na metodologia de 10 semanas demonstrou como a divisão em etapas e a colaboração entre os integrantes favorecem um trabalho mais organizado, flexível e eficiente. Assim, espera-se que este material sirva como base sólida para a continuação do projeto e para o aperfeiçoamento da plataforma, contribuindo para uma experiência acadêmica mais estruturada e acessível.

7 REFERÊNCIAS

ORACLE. Java Documentation. Disponível em: <https://docs.oracle.com/javase/> . Acesso em: 10 nov. 2025.

SPRING. Spring Boot Reference Documentation. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 10 nov. 2025.

MORAIS, Izabelly S.; ZANIN, Aline. Engenharia de software. Porto Alegre: SAGAH, 2020. E-book. ISBN 9788595022539. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788595022539/> . Acesso em: 15 set. 2025.

BRASIL. Lei Geral de Proteção de Dados Pessoais (Lei n. 13.709/2018). Disponível em: <https://www.planalto.gov.br/ccivil03/ato2015-2018/2018/lei/L13709.htm>. Acesso em: 10 nov. 2025.

8 ANEXOS

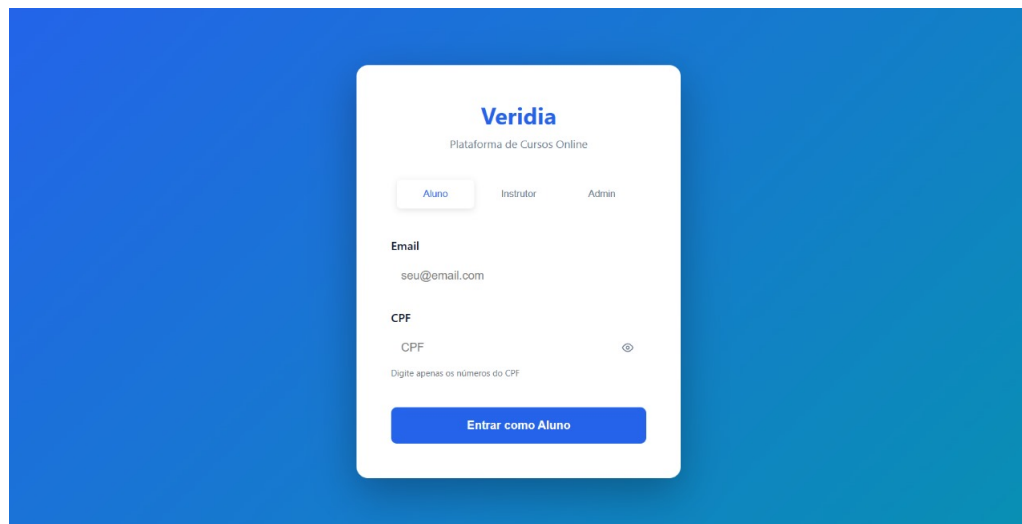


Figura 10 – TELA DE LOGIN

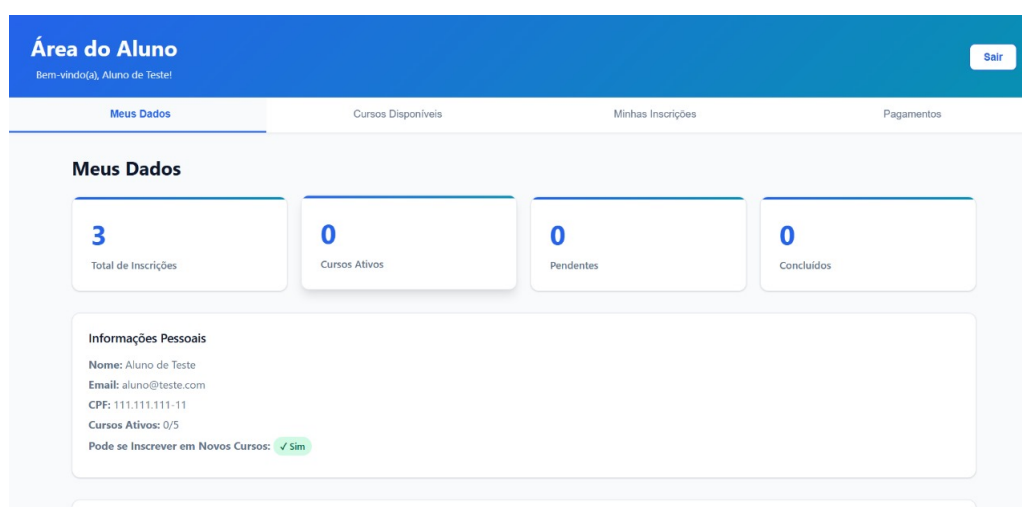


Figura 11 – DASHBOARD DE ALUNO

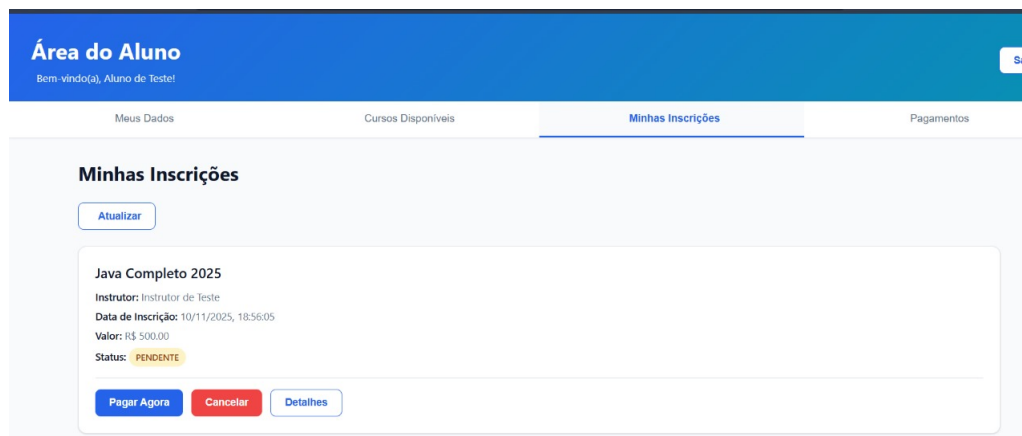


Figura 12 – MINHAS INSCRIÇÕES

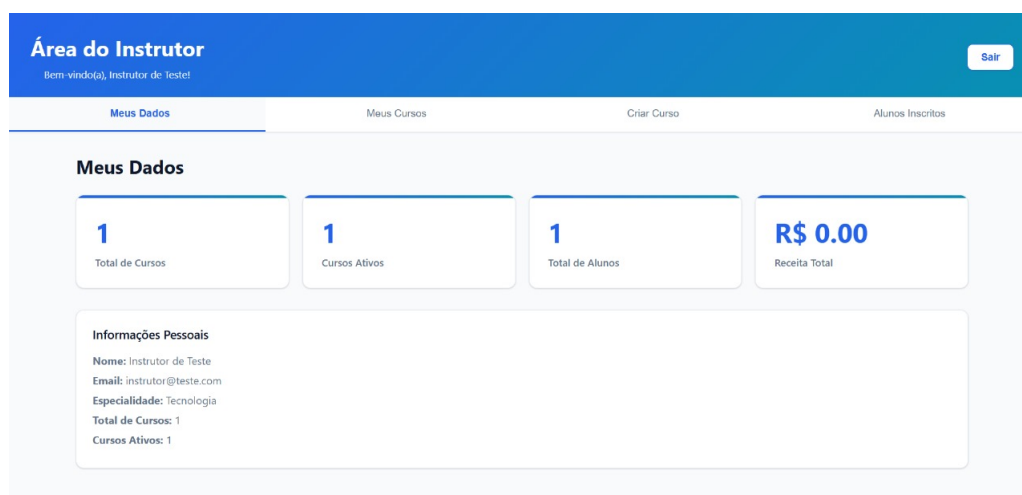


Figura 13 – DASHBOARD DE INSTRUTOR

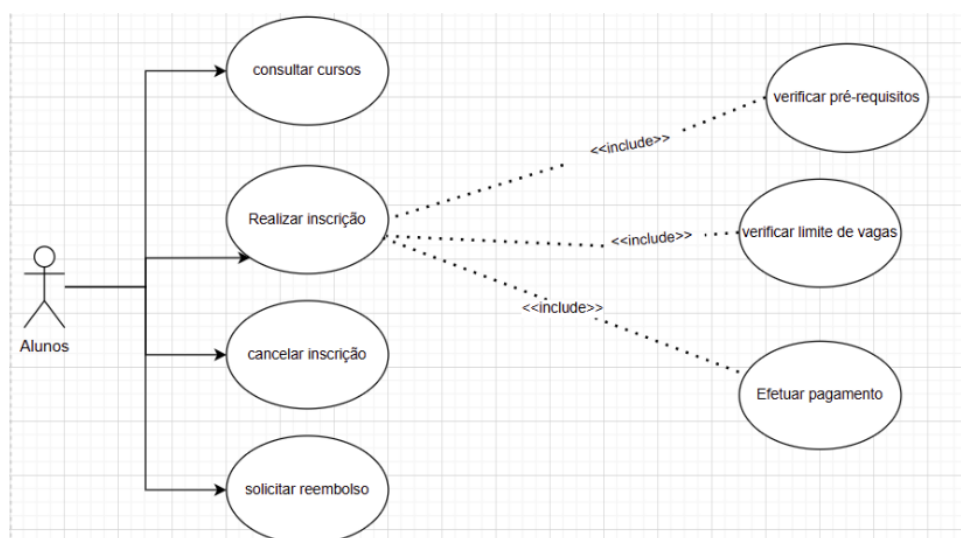


Figura 15 – Diagrama de Casos de Uso - Ator Aluno.

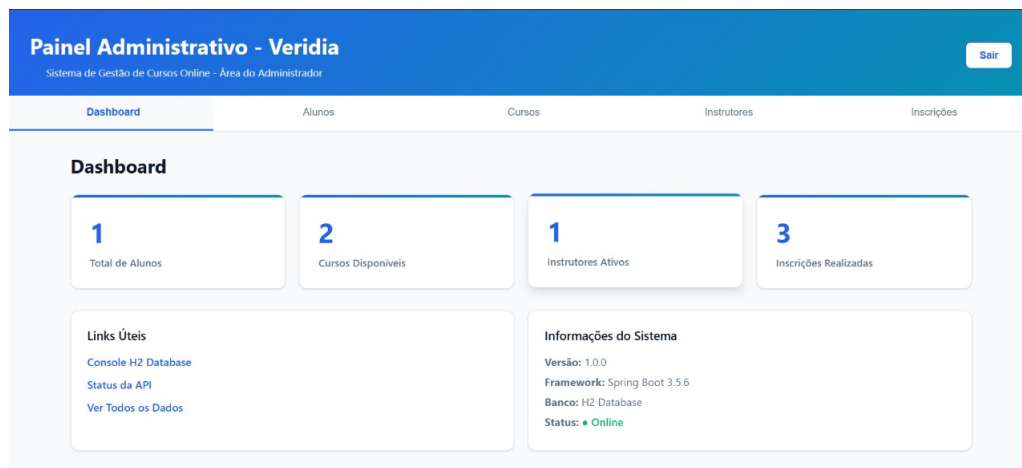


Figura 14 – DASHBOARD DE ADMINISTRADOR

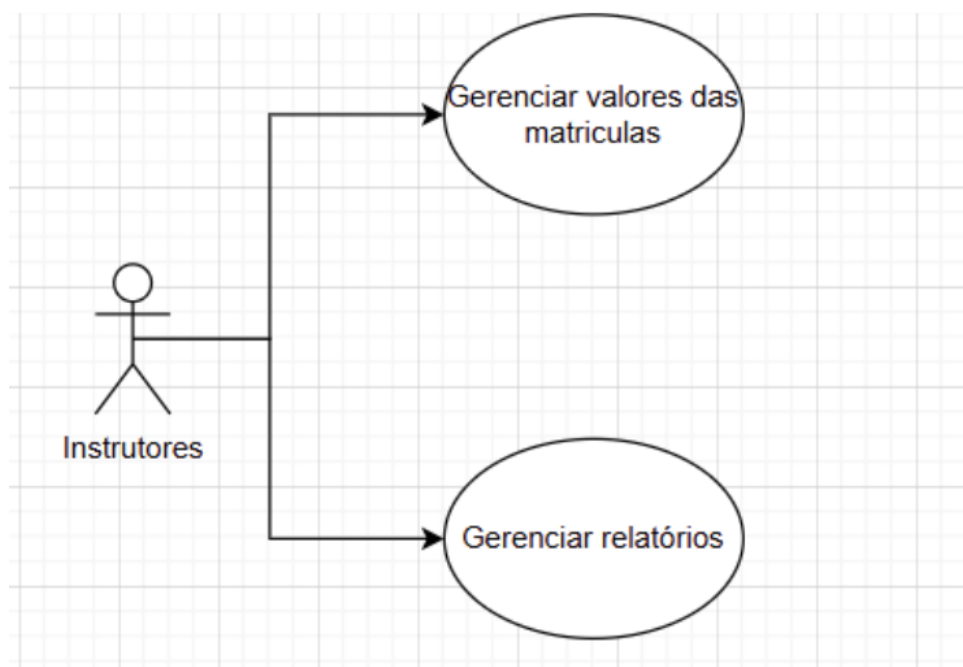


Figura 16 – Diagrama de Casos de Uso - Ator Instrutor.

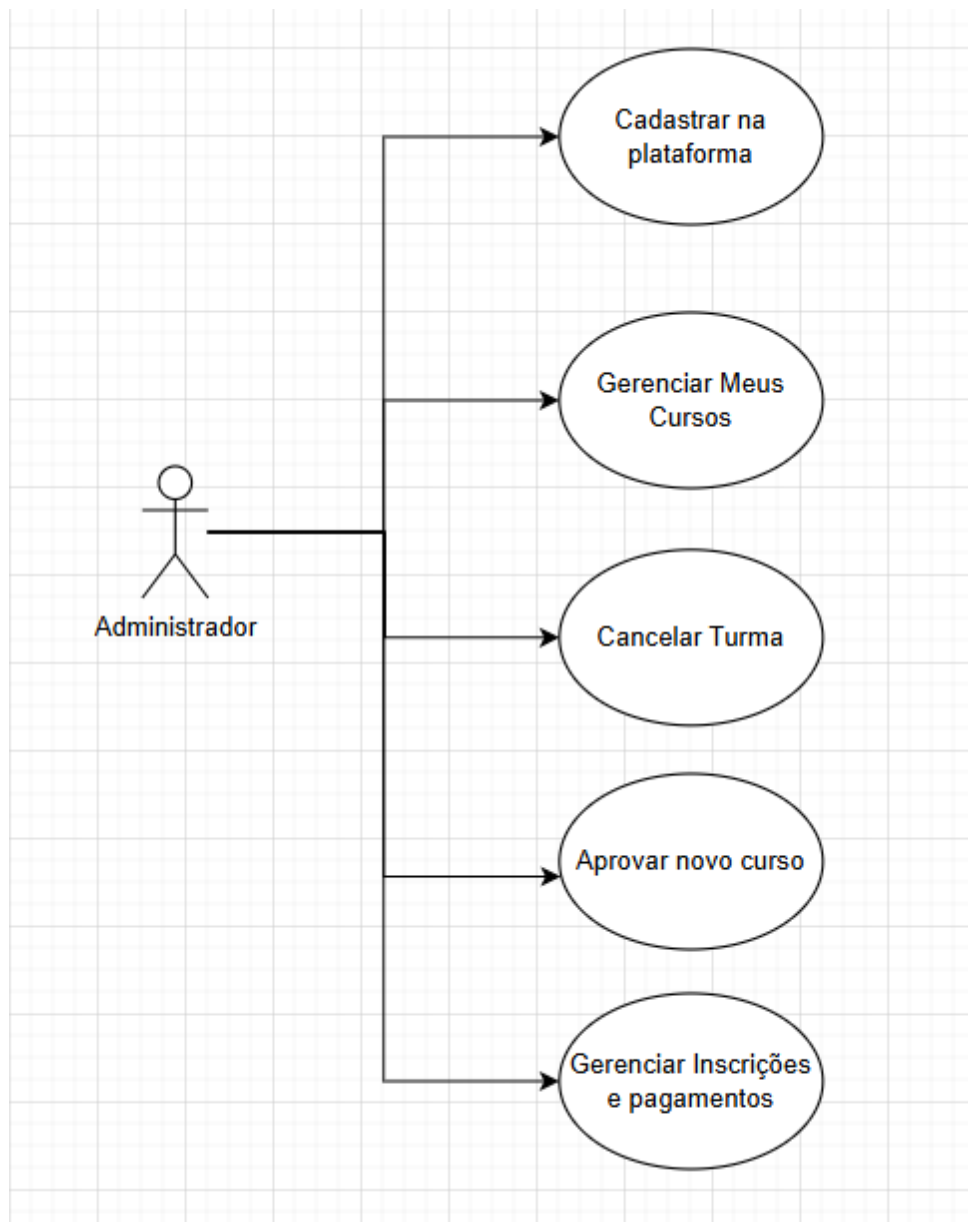


Figura 17 – Diagrama de Casos de Uso - Ator Administrador.

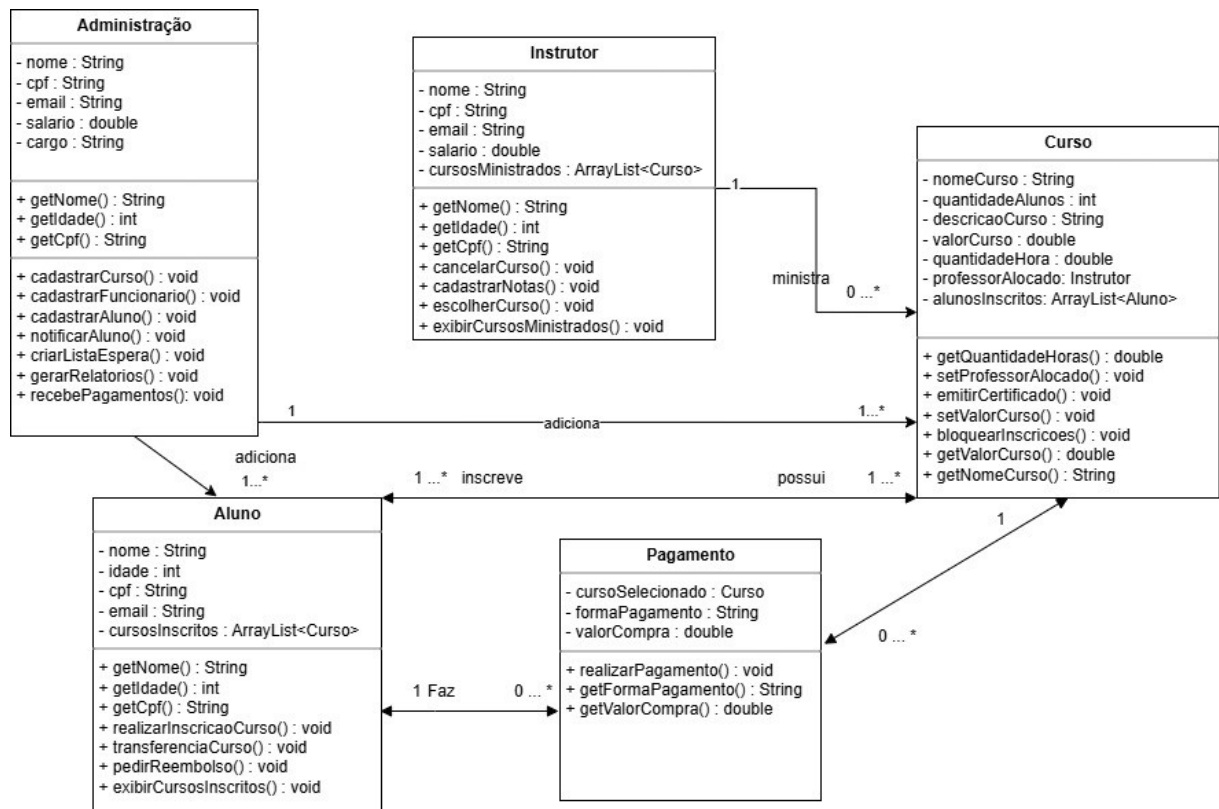


Figura 18 – Diagrama de Classes do sistema.