

UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 6º Período

DISCIPLINA: *Oficina de Integração 1 - ES46F-ES61*

PROFESSOR: *Eduardo Cotrin Teixeira*

Documento de Projeto de Software

ELLP-Gerenciamento de famílias e notas

Gustavo Betiati Ferreira

Lucas Gabriel Costa

Samuel Penha Jacobsen

Samuel Ribeiro da Costa

Cornélio Procópio

2024

Sumário

1	INTRODUÇÃO	2
1.1	Contexto	2
1.2	Justificativa	4
1.3	Proposta	5
1.4	Organização do Documento	5
2	DESCRIÇÃO GERAL DO SISTEMA	6
2.1	Objetivos (Gerais e Específicos)	6
2.2	Limites e Restrições	7
2.3	Descrição dos Usuários do Sistema	9
3	DESENVOLVIMENTO DO PROJETO	11
3.1	Tecnologias e ferramentas	11
3.2	Metodologia de desenvolvimento	12
3.3	Cronograma previsto	13
4	REQUISITOS DO SISTEMA	14
4.1	Requisitos Funcionais	14
4.2	Requisitos Não-funcionais	16
4.3	Diagramas de Casos de Uso	16
4.4	Protótipos de Telas	21
5	ANÁLISE DO SISTEMA	33
5.1	Dicionário de dados	33
5.2	Modelo do Banco de Dados	37
5.3	Diagrama de Classes	38
5.3	Diagrama de Atividades	39
6	IMPLEMENTAÇÃO	
6.1	Descrição do código	46
6.2	Implantação	51
6.3	Telas principais	54
7	CONSIDERAÇÕES FINAIS	56
8	BIBLIOGRAFIA	

1 Introdução

1.1 Contexto

O projeto de extensão ELLP (Ensino Lúdico de Lógica de Programação) foi desenvolvido com o objetivo de democratizar o acesso ao conhecimento de lógica e programação para estudantes do ensino público na região de Cornélio Procopio, Paraná. A iniciativa busca preencher uma lacuna educacional ao fornecer habilidades tecnológicas essenciais, alinhadas com as demandas do atual cenário de transformação digital. O projeto é direcionado a jovens que, de outra forma, teriam pouco ou nenhum acesso a esse tipo de educação complementar.

O projeto oferece oficinas estruturadas com uma progressão pedagógica, começando com fundamentos de informática e evoluindo para habilidades mais avançadas, como programação e desenvolvimento web. O público-alvo compreende alunos do quinto ano do ensino fundamental ao terceiro ano do ensino médio. Além disso, o projeto inclui atividades de robótica prática, que tornam o aprendizado mais interativo e motivador. As oficinas ocorrem regularmente aos sábados, com duas sessões pela manhã e duas no período da tarde.

O grupo de voluntários que executa essas atividades é composto por alunos de diversos cursos da Universidade Tecnológica Federal do Paraná (UTFPR). A estrutura de ensino inclui um voluntário atuando como instrutor em cada oficina, com suporte de outros voluntários que atuam como tutores. Essa abordagem permite um acompanhamento personalizado para cada aluno e oferece aos voluntários uma experiência prática valiosa.

O sistema digital desenvolvido trará uma série de benefícios ao projeto ELLP, incluindo:

- **Automatização do gerenciamento:** permitirá o registro detalhado de dados dos alunos e suas famílias, substituindo os registros manuais em papel.
- **Monitoramento socioeconômico:** possibilitará a análise de informações socioeconômicas, contribuindo para a identificação de necessidades específicas e estratégias mais eficazes.
- **Acompanhamento acadêmico:** facilitará o controle de notas e o progresso educacional de cada aluno, promovendo um acompanhamento mais eficiente.
- **Geração de relatórios:** permitirá a criação de relatórios rápidos e precisos, reduzindo o risco de erros e otimizando o tempo dos voluntários.
- **Acesso centralizado às informações:** fornecerá aos voluntários e coordenadores uma plataforma unificada para consultar e atualizar dados, aumentando a organização e a produtividade.

Essa ferramenta será essencial para aprimorar a gestão do projeto, fortalecer o impacto educacional e proporcionar uma experiência mais estruturada para os alunos e suas famílias, além de facilitar a atuação dos voluntários.

1.2 Justificativa

Atualmente, o projeto ELLP enfrenta dificuldades significativas devido ao uso de métodos manuais para o acompanhamento do desempenho dos alunos e a geração de relatórios. Esse processo manual é demorado, ineficiente e altamente suscetível a erros, o que impacta negativamente a organização e o controle do progresso dos alunos. A falta de um sistema automatizado torna o acompanhamento impreciso e compromete a capacidade de análise dos dados, resultando em atrasos na geração de relatórios e na tomada de decisões estratégicas. O público diretamente afetado por essa situação são os voluntários do projeto, responsáveis pela coleta e organização das

informações. Além disso, a ausência de um sistema centralizado dificulta o acesso eficiente às informações sobre o progresso dos alunos, impactando a qualidade do acompanhamento pedagógico. A falta de uma solução adequada para gerenciar o desempenho e o histórico dos alunos acarreta uma série de problemas, como a demora na elaboração dos relatórios, o risco elevado de erros dos dados e a falta de controle efetivo a longo prazo. Essa situação compromete a transparência e a precisão das informações repassadas tanto aos voluntários quanto às famílias dos alunos, prejudicando o acompanhamento contínuo e a análise do desempenho.

Com a implementação de um sistema automatizado de **gerenciamento de famílias e notas**, será possível melhorar significativamente a produtividade dos voluntários e a qualidade nas análises. O sistema permitirá a geração de relatórios detalhados de forma automatizada, reduzindo o tempo necessário para esse processo e minimizando os riscos de erros. Além disso, a centralização dos dados garantirá um acompanhamento mais eficaz do progresso dos alunos, proporcionando maior transparência e controle.

1.3 Proposta

A solução proposta é o desenvolvimento de um sistema de gerenciamento de famílias e notas que integrará dados socioeconômicos e acadêmicos em uma plataforma digital. O sistema permitirá que os voluntários do ELLP registrem e acessem informações de forma organizada, facilitando a geração de relatórios que analisem o impacto do projeto sobre o desempenho escolar. O sistema será capaz de registrar informações detalhadas sobre as famílias e acompanhar as notas em diferentes disciplinas, permitindo uma análise bimestral do impacto das oficinas. A geração de relatórios incluirá listas de alunos com renda per capita abaixo de um valor específico, além de análises do progresso ao longo do tempo. Com essa solução, espera-se melhorar a gestão do projeto e promover uma comunicação mais eficaz entre voluntários e famílias.

1.4 Organização do Documento

Este documento foi elaborado para apresentar o desenvolvimento do sistema de gerenciamento de famílias e notas. Ele está organizado em seções que abrangem desde o contexto inicial até os resultados alcançados, com destaque para os aspectos técnicos e metodológicos do projeto. A introdução apresenta o contexto do projeto, a justificativa para sua realização e a proposta de solução desenvolvida.

A descrição geral do sistema detalha os objetivos principais, as limitações e restrições do sistema, bem como a caracterização dos usuários. A seção de desenvolvimento do projeto expõe as tecnologias, ferramentas e metodologia utilizadas, além do cronograma planejado para a implementação. Já a parte de requisitos do sistema define os requisitos funcionais e não funcionais, complementados por diagramas de casos de uso e protótipos das interfaces. A modelagem e análise do sistema inclui o modelo de dados, diagramas UML e demais elementos da modelagem, como dicionários de dados.

A implementação e resultados descreve a organização do código, as etapas de implantação e as telas principais do sistema, além de uma análise dos resultados obtidos. As considerações finais apresentam os desafios enfrentados, os aprendizados adquiridos e sugestões para evoluções futuras do sistema. Por fim, a bibliografia lista todas as fontes e materiais consultados durante o desenvolvimento do projeto. Essa organização visa facilitar a navegação pelo documento, oferecendo uma visão clara e objetiva do projeto e de seus resultados.

2 Descrição Geral do Sistema

2.1 Objetivos (Gerais e Específicos)

Objetivo Geral:

Desenvolver um sistema de gerenciamento de famílias e notas para o projeto ELLP que automatize o processo de controle de desempenho dos alunos, geração de relatórios e acompanhamento do progresso em uma plataforma web. O sistema deve oferecer funcionalidades para registrar e monitorar dados educacionais e socioeconômicos de forma centralizada e organizada, facilitando o acesso e o uso pelos voluntários do projeto.

Objetivos Específicos:

Automatizar a geração de relatórios – Substituir o processo manual de registro de dados por um sistema automatizado que permita a geração precisa e ágil de relatórios sobre o desempenho dos alunos. Esse desempenho será avaliado com base em fatores como notas acadêmicas, acessibilidade a recursos tecnológicos (como internet, celular e computador), e condições socioeconômicas, incluindo a renda familiar.

Centralizar o controle de dados acadêmicos – Implementar uma base de dados única e centralizada, onde todas as informações sobre notas e progresso dos alunos sejam armazenadas e acessíveis de maneira eficiente pelos voluntários e coordenadores do projeto.

Aprimorar o acesso à informação pelos voluntários – Garantir que os voluntários tenham acesso rápido e confiável às informações sobre o desempenho dos alunos, facilitando a tomada de decisões pedagógicas e melhorando o planejamento das oficinas.

Melhorar a precisão no controle de notas – Reduzir o risco de erros na inserção e organização das notas dos alunos, aumentando a confiabilidade dos dados e a precisão nos resultados gerados pelo sistema.

Oferecer uma interface amigável e intuitiva – Desenvolver uma interface web de fácil uso que permita aos voluntários e coordenadores do projeto

navegar e utilizar o sistema de forma eficiente, sem a necessidade de conhecimentos técnicos avançados.

2.2 Limites e Restrições

O sistema de gerenciamento de famílias e notas será acessado exclusivamente pelos voluntários do projeto ELLP, e terá como foco o acompanhamento do desempenho acadêmico dos alunos, bem como a geração de relatórios que trazem essas informações com dados socioeconômicos das famílias. A presença das famílias será limitada aos dados fornecidos para análise de desempenho e contexto socioeconômico, sem acesso direto ao sistema.

Limites:

Acesso restrito aos voluntários – O sistema será acessado exclusivamente pelos voluntários do projeto ELLP, não havendo suporte para que alunos, familiares ou outras partes externas possam interagir diretamente com a aplicação. As famílias estarão presentes apenas como parte dos dados utilizados para gerar relatórios e análises.

Dados Socioeconômicos – O sistema coletará e organizará informações socioeconômicas das famílias dos alunos para cruzar esses dados com o desempenho escolar. No entanto, ele não terá funcionalidades adicionais voltadas para o gerenciamento detalhado de perfis familiares ou interações diretas entre voluntários e familiares.

Foco na Geração de Relatórios – O sistema será limitado à geração de relatórios relacionados ao desempenho acadêmico dos alunos, com foco em cruzar essas informações com o grupo socioeconômico, participação nas oficinas e desempenho geral. Funcionalidades mais amplas, como análises comportamentais ou desenvolvimento de habilidades extracurriculares, não serão contempladas na versão inicial.

Interface de Usuário – A aplicação será desenvolvida com uma interface simples e voltada para facilitar o uso pelos voluntários do projeto. Não haverá

funcionalidades avançadas de personalização ou complexidade no design, visando garantir uma implementação eficiente e direta.

Restrições:

Restrições tecnológicas e de projeto: O sistema será desenvolvido como uma aplicação web, utilizando React.js, Java Spring Boot e o banco de dados sendo o MongoDB. O prazo máximo previsto para a conclusão é 16/02/2025.

Prazo de Desenvolvimento:

O prazo de desenvolvimento do projeto é até o final do semestre. Devido a essa restrição de tempo, funcionalidades avançadas ou integrações complexas serão deixadas para futuras fases do projeto.

Segurança

- **Sanitização de Rotas:** Todas as rotas da API serão protegidas com sanitização de entradas, prevenindo ataques de injeção de código e garantindo a integridade dos dados transmitidos.
- **Criptografia de Senhas:** As senhas dos usuários voluntários serão criptografadas antes de serem armazenadas no banco de dados, assegurando a proteção das credenciais.
- Funcionalidades de segurança mais avançadas, como autenticação multifator, poderão ser consideradas em futuras fases do projeto, mas não estarão presentes na versão inicial.

Ambiente de Implantação

O sistema completo será hospedado na AWS. Caso haja algum problema relacionado ao custo, optamos por rodar o front-end na Vercel e o back-end na Railway.

Escalabilidade e Manutenção

A aplicação será desenvolvida com foco em fácil manutenção e expansão futura, mas as funcionalidades iniciais serão limitadas ao escopo do controle de notas e geração de relatórios. O suporte à escalabilidade será simples,

sem previsão de alta disponibilidade ou balanceamento de carga avançado para a fase inicial.

Compatibilidade

A aplicação será projetada para ser compatível com navegadores modernos e dispositivos desktop e móveis.

2.3 Descrição dos Usuários do Sistema

O sistema será utilizado principalmente pelos voluntários e coordenadores do projeto de extensão ELLP, cada grupo com diferentes responsabilidades e funcionalidades específicas para atender às suas necessidades.

Voluntários: Os voluntários são usuários que participam ativamente das oficinas do projeto, auxiliando no acompanhamento dos alunos e na coleta de dados. Eles precisam de acesso a funcionalidades que lhes permitam:

- **Gerenciar alunos e famílias:** Cadastrar e atualizar informações dos alunos, incluindo dados acadêmicos e detalhes socioeconômicos das famílias, para construir relatórios mais completos e informativos.
- **Inserir e acompanhar notas:** Adicionar, visualizar e editar as notas dos alunos em diferentes disciplinas, permitindo um controle detalhado do progresso acadêmico.
- **Histórico de desempenho:** Consultar e editar o histórico de notas dos alunos ao longo do tempo para monitorar o desenvolvimento educacional.
- **Geração de relatórios:** Criar relatórios que cruzem dados de desempenho acadêmico com aspectos socioeconômicos, ajudando na compreensão das condições e no impacto das oficinas.

Coordenadores: Os coordenadores possuem um papel mais abrangente e estratégico dentro do projeto. Além das funcionalidades dos voluntários, eles têm acesso a ferramentas adicionais para supervisão e organização:

- **Cadastro e gerenciamento de usuários:** Registrar novos voluntários e coordenadores no sistema, garantindo que apenas usuários autorizados possam acessar as informações.
- **Gerenciamento de turmas e atividades:** Criar e organizar turmas, além de planejar cronogramas de atividades e oficinas, permitindo uma visão clara e centralizada do andamento do projeto.
- **Análise de desempenho:** Utilizar funcionalidades específicas de análise de desempenho para identificar padrões e tendências nos resultados dos alunos, facilitando a identificação de pontos de melhoria e a tomada de decisões estratégicas.

Ambos os tipos de usuários compartilham o acesso a um sistema de **login seguro** e são protegidos por níveis de acesso que garantem a segurança e a integridade dos dados armazenados. A interface do sistema foi projetada para ser intuitiva e acessível, de forma que tanto voluntários quanto coordenadores possam realizar suas atividades de maneira eficiente e eficaz.

3 Desenvolvimento do Projeto

3.1 Tecnologias e ferramentas

Tecnologia de Desenvolvimento

- **Front-end:** A interface do usuário será construída utilizando **React.js**, uma biblioteca popular para construção de interfaces dinâmicas e responsivas.
- **Back-end:** O back-end será desenvolvido em **Java Spring Boot**, uma estrutura robusta e escalável para a criação de APIs RESTful.
- **Banco de Dados:** O sistema utilizará **MongoDB**, um banco de dados NoSQL que oferece flexibilidade para o armazenamento dos dados dos alunos e famílias.

Ambiente de Desenvolvimento

- **Visual Studio Code:** Ambiente de desenvolvimento integrado (IDE)

Organizacional

- **Google Docs** - Edição e compartilhamento de documentos.
- **Git** - Sistema de controle de versão distribuído.
- **GitHub** - Controle de versionamento de código.

Gerenciamento de atividades

- **Trello** - Ferramenta de gerenciamento de tarefas

Prototipação de telas

- **Figma** - Ferramenta de design de interface e prototipagem de alta fidelidade.

3.2 Metodologia de desenvolvimento

Como modelo de ciclo de vida, foi escolhida a metodologia de gestão ágil Scrum, amplamente utilizada no desenvolvimento de software. Ele possui

práticas que estimulam a colaboração, flexibilidade e interação entre os membros da equipe.

O ciclo de vida do Scrum começa com a definição da visão do produto, liderada pelo Product Owner, que identifica os objetivos do projeto e as funcionalidades principais. Esses itens são organizados em um backlog do produto (Product Backlog), que representa uma lista priorizada de tudo o que precisa ser feito.

O projeto é dividido em sprints, períodos de tempo curtos e fixos, geralmente de 2 a 4 semanas. Em cada sprint, uma seleção de itens do backlog do produto é retirada e trabalhada, formando o backlog da sprint (Sprint Backlog). A definição desses itens é feita na reunião de planejamento da Sprint, onde a equipe estima sua capacidade e escolhe o que pode ser concluído no tempo disponível.

Durante cada sprint, há reuniões diárias chamadas de Daily Scrums, em que a equipe discute o progresso do dia anterior, o plano para o dia atual, e identifica qualquer bloqueio. Essas reuniões são curtas e ajudam a manter o foco da equipe e identificar rapidamente problemas.

Ao final de cada sprint, duas reuniões principais acontecem: a Sprint Review, onde o progresso é apresentado e as funcionalidades desenvolvidas são avaliadas e possivelmente adaptadas; e a Sprint Retrospective, onde a equipe reflete sobre o processo de trabalho, discutindo o que funcionou bem e o que precisa ser melhorado para aumentar a eficiência nas próximas sprints.

Para este projeto específico, o Scrum permite um acompanhamento contínuo, garantindo que as prioridades do backlog possam ser ajustadas conforme surgem novas necessidades, mantendo a equipe alinhada e o produto em constante evolução.

Os papéis foram divididos em:

- **Scrum Master** (Lucas): Responsável por garantir que a equipe siga as práticas e valores do Scrum, removendo impedimentos e facilitando a comunicação entre os membros da equipe.
- **Product Manager** (Samuel Jacobsen): Atua como a ponte entre o cliente e a equipe de desenvolvimento, definindo as prioridades e garantindo que o backlog esteja alinhado com os objetivos do projeto
- **Desenvolvedores** (Gustavo e Samuel): Responsáveis pela implementação das funcionalidades do sistema, trabalhando em estreita colaboração com o designer e o product manager para transformar os requisitos em um produto funcional.

Antes de cada sprint, são definidas as partes do backlog do produto a serem abordadas, e ao final de cada sprint, reuniões são conduzidas para revisar e complementar as informações, tornando assim o desenvolvimento incremental do projeto. Durante cada sprint, serão desenvolvidos diferentes componentes do sistema.

3.3 Cronograma previsto

Fase 1: Planejamento e Inicialização

30/09 a 06/10

- **Atividade:** Apresentação da disciplina e alinhamento inicial.
- **Descrição:** Reunião para apresentar o projeto, os objetivos e o cronograma geral.
- **Responsável:** Todos.
- **Entrega:** Documento inicial com visão geral e responsabilidades.

07/10

- **Atividade:** Reunião com o professor antônio
- **Descrição:** Alinhamento inicial com o professor, referente ao projeto da ELLP, trazendo e entendendo os principais problemas enfrentados

- **Responsável:** Todos.

08/10 a 09/10

- **Atividade:** Introdução e análise do documento do projeto.
- **Descrição:** Primeira leitura detalhada do documento, revisão de objetivos e levantamento de dúvidas para a reunião com o professor. Refinamento das atividades do backlog
- **Responsável:** Todos.

10/10 a 13/10

- **Atividade:** Descrição Geral do Sistema.
- **Descrição:** Estruturar e documentar a visão geral do sistema, incluindo principais funcionalidades e stakeholders. Distribuição e análise das atividades selecionadas para a sprint
- **Responsável:** Todos.

14/10 a 17/10

- **Atividade:** Reunião com o professor e distribuição de atividades no Trello.
- **Descrição:** Alinhar expectativas do projeto, validar os objetivos e dividir as tarefas iniciais no Trello.
- **Responsável:** Todos.

Fase 2: Levantamento de Requisitos e Prototipação

18/10 a 20/10

- **Atividade:** Levantamento de Requisitos do Sistema.
- **Descrição:** Detalhar requisitos funcionais e não funcionais, incluindo user stories no Trello. Alinhamento sobre o uso da ferramenta trello e sobre a metodologia scrum
- **Responsável:** Todos.

21/10 a 27/10

- **Atividade:** Participação na SECOMP.

- **Descrição:** Participar do evento
- **Responsável:** Todos.

28/10 a 30/10

- **Atividade:** Validação e prototipagem inicial.
- **Descrição:** Revisão do backlog e validação do protótipo no Figma. Alinhamento e QA por parte de todos os integrantes do squad
- **Responsável:** Todos.

31/10 a 02/11

- **Atividade:** Ajustes finais da documentação, Fase de QA nas últimas atividades. Alinhamento(daily) pelo grupo do whatsapp.
- **Descrição:** Produzir o vídeo explicativo com visão geral do projeto na parte do item 5.
- **Responsável:** Todos.

03/11

- **Atividade:** Desenvolvimento do vídeo de apresentação.
- **Descrição:** Produzir o vídeo explicativo com visão geral do projeto na parte do item 5.
- **Responsável:** Todos.

04/11 a 06/11

- **Atividade:** Envio do documento e vídeo da primeira fase.
- **Descrição:** Revisar e submeter os entregáveis para a primeira entrega. Fazer etapa de QA com todos os integrantes, validando a segunda entrega com qualidade
- **Responsável:** Todos.

Fase 3: Análise e Modelagem do Sistema

11/11 a 17/11

- **Atividade:** Análise do Sistema.

- **Descrição:** Criar o modelo de banco de dados e diagramas UML..
Validação com o time referente ao modelo a ser desenvolvido

- **Responsável:** Todos.

25/11 a 30/11

- **Atividade:** Validação da etapa 5.
- **Descrição:** Revisar os modelos e diagramas para alinhamento interno e validação com o professor. Discussão e análise do time completo por meio da daily.
- **Responsável:** Todos.

01/12

- **Atividade:** Ajustes da primeira etapa
- **Descrição:** Revisão, retro e alinhamento com o time referente a primeira entrega. Corrigindo os erros e arrumando os problemas apontados pelo professor
- **Responsável:** Todos.

02/12

- **Atividade:** Etapa final dos ajustes de documentação da segunda entrega
- **Descrição:** Revisão final e envio dos materiais para avaliação. Validação dos modelos desenvolvidos por meio do QA, daily feita pelo whatsapp e revisão da segunda entrega
- **Responsável:** Todos.

03/12 a 04/12

- **Atividade:** Gravação do video da segunda etapa
- **Descrição:** Revisão final e desenvolvimento e apresentação do vídeo do material apresentado
- **Responsável:** Todos.

Fase 4: Implementação e Refinamento

09/12 a 22/12

- **Atividade:** Protótipos de Telas e Descrição do Código (Item 6).
- **Descrição:** Implementar os protótipos no Figma e documentar o código-fonte.
- **Responsável:** Todos.

03/02 a 09/02

- **Atividade:** Alinhamento final do projeto.
- **Descrição:** Realizar revisão geral e preparação para a entrega final.
- **Responsável:** Todos.

10/02 a 11/02

- **Atividade:** Desenvolvimento e apresentação do vídeo de entrega final.
- **Descrição:** Fazer planejamento e refinamento com o time, ajustando a distribuição dos tópicos do vídeo. QA referente a apresentação
- **Responsável:** Todos.

12/02

- **Atividade:** Envio do documento e vídeo da 3ª Fase (Entrega Final).
- **Descrição:** Concluir e enviar os materiais finais. QA com o time todo durante a daily
- **Responsável:** Todos.

Fase 5: Encerramento e Retrospectiva

17/02 a 23/02

- **Atividade:** Retrospectiva do Projeto.
- **Descrição:** Avaliar os resultados, discutir aprendizados e documentar melhorias para futuros projetos.
- **Responsável:** Todos.

4 Requisitos do Sistema

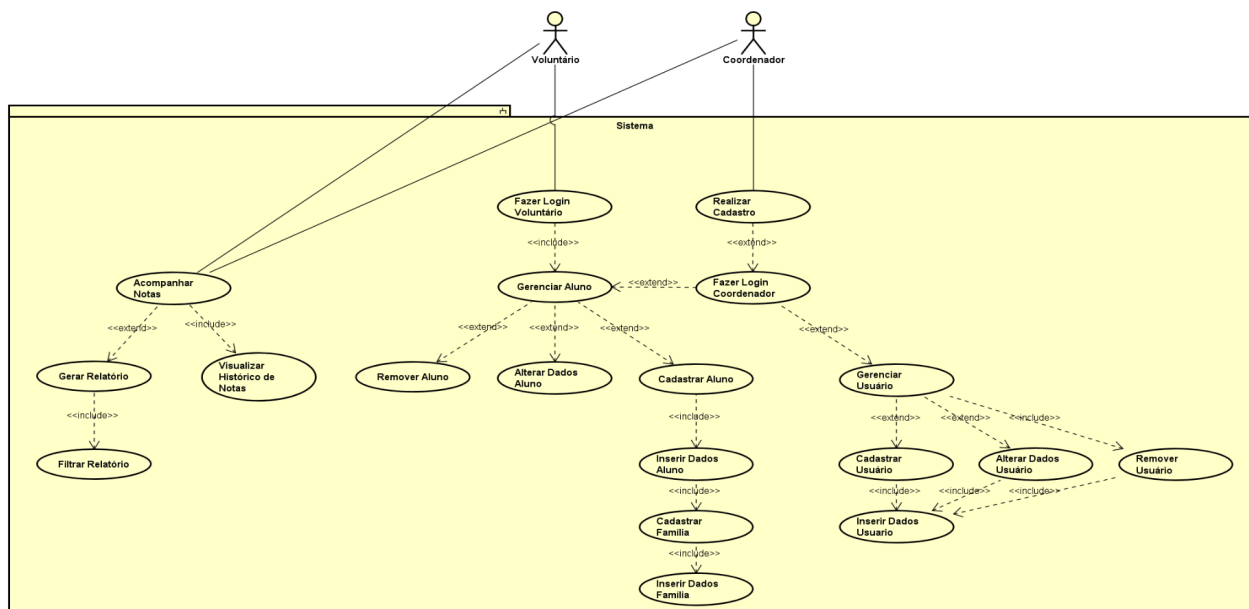
4.1 Requisitos Funcionais

ID	Funcionalidade	Prioridade
RF01	O sistema deve permitir o cadastro de alunos, com informações como nome, idade, escola e série.	Essencial
RF02	O sistema deve permitir o cadastro de famílias, incluindo informações socioeconômicas, como renda média, número de membros e acesso a internet.	Essencial
RF03	O sistema deve permitir o cadastro e acompanhamento das notas dos alunos em diferentes disciplinas (Ciências, Matemática, etc.), organizadas por bimestres.	Essencial
RF04	Gerar relatórios sobre o desempenho acadêmico dos alunos, filtrando por diferentes critérios (ex: renda per capita, evolução das notas).	Essencial
RF05	O sistema deve permitir filtrar os relatórios por critérios, como renda per capita, escola ou progresso nas notas.	Essencial
RF06	O sistema deve permitir a visualização do histórico de notas de um aluno individual.	Essencial
RF07	O sistema deve permitir o controle de acesso para garantir que somente usuários autorizados possam acessar os dados.	Essencial
RF08	O sistema deve permitir que os voluntários insiram ou editem informações sobre os alunos e suas famílias.	Importante
RF09	O sistema deve registrar as alterações feitas nas informações dos alunos e famílias para fins de auditoria.	Desejável
RF10	O sistema deve gerar alertas automáticos quando houver mudanças significativas nas notas dos alunos (melhora ou queda).	Importante
RF11	Possibilitar a edição e atualização dos dados dos alunos e famílias.	Essencial
RF12	Permitir a visualização do histórico de notas dos alunos ao longo do tempo.	Importante
RF13	Implementar um sistema de busca avançada para localizar rapidamente alunos com base em diferentes critérios (idade, notas, etc.).	Importante

4.2 Requisitos Não-funcionais

ID	Requisito	Categoria
RNF01	O sistema deve ser fácil de usar e interagir, com uma interface gráfica amigável e intuitiva.	Requisito de Usabilidade.
RNF02	O sistema deve ser rápido e responsivo, sem atrasos ou latência perceptível para os clientes.	Requisito de Eficiência.
RNF03	O sistema deve garantir a integridade dos dados do cliente.	Requisito de Segurança.
RNF04	O sistema deve ser compatível com diferentes navegadores e dispositivos, incluindo desktops, laptops, tablets e smartphones.	Requisito de Compatibilidade.
RNF05	O sistema deve ser escalável, permitindo adicionar novos usuários e recursos sem degradação significativa de desempenho.	Requisito de Desempenho.

4.3 Diagramas de Casos de Uso



	Especificação do Caso de Uso - Cadastro de Alunos
Descrição	Esse caso de uso descreve as etapas percorridas por um voluntário para cadastrar um aluno no sistema.
Atores	Voluntário
Pré-condições	O voluntário deve estar autenticado no sistema
Pós-condições	O aluno estará cadastrado no sistema e disponível para futuras consultas e edições.
Ações Normais	<ol style="list-style-type: none"> 1. O voluntário acessa o sistema de gerenciamento. 2. O voluntário escolhe a opção de "Cadastrar Aluno". 3. O voluntário preenche o formulário de cadastro com as seguintes informações: nome, idade, escola e série. 4. O voluntário clica no botão "Cadastrar". 5. O sistema valida os dados preenchidos. 6. O sistema salva as informações do aluno no banco de dados. 7. O sistema exibe uma mensagem confirmando que o aluno foi cadastrado com sucesso.
Observação	Todos os campos de cadastro são obrigatórios. Se algum dado estiver incorreto ou faltar, o sistema exibirá uma mensagem de erro para o voluntário corrigir as informações.
Requisitos Relacionados	RF01, RF04, RF06

	Especificação do Caso de Uso - Acompanhamento de Notas
Descrição	Este caso de uso descreve as etapas para que um voluntário insira ou visualize as notas de um aluno em diferentes disciplinas, organizadas por bimestres.
Atores	Voluntário
Pré-condições	O voluntário deve estar autenticado e o aluno já deve estar cadastrado no sistema.
Pós-condições	As notas serão inseridas ou atualizadas no sistema, podendo ser visualizadas futuramente.
Ações Normais	<ol style="list-style-type: none"> 1. O voluntário acessa o sistema e navega até a área de acompanhamento de notas. 2. O voluntário seleciona o aluno e o bimestre desejado. 3. O sistema exibe as disciplinas disponíveis (ex: Ciências, Matemática). 4. O voluntário insere ou edita as notas de cada disciplina. 5. O voluntário confirma as alterações. 6. O sistema salva as notas no banco de dados. 7. O sistema exibe uma mensagem confirmando que as notas

	foram registradas com sucesso.
Observação	O sistema deve validar que todas as notas inseridas estão dentro de um intervalo aceitável e organizadas por bimestre.
Requisitos Relacionados	RF03, RF04.

	Especificação do Caso de Uso - Geração de Relatórios Acadêmicos e Socioeconômicos
Descrição	Este caso de uso descreve como um usuário autorizado gere relatórios que cruzem dados socioeconômicos com o desempenho acadêmico dos alunos.
Atores	Voluntário, Coordenador.
Pré-condições	O usuário deve estar autenticado e ter permissões adequadas para gerar relatórios.
Pós-condições	O relatório será gerado e exibido para o usuário.
Ações Normais	<ol style="list-style-type: none"> 1. O usuário acessa a área de relatórios no sistema. 2. O sistema solicita os critérios de filtro (renda, número de membros, acesso à internet, etc.). 3. O usuário seleciona os critérios desejados. 4. O sistema processa os dados e gera o relatório cruzando informações acadêmicas e socioeconômicas. 5. O sistema exibe o relatório para o usuário.
Observação	O relatório pode ser filtrado por diversos critérios, como renda, escola e desempenho nas notas.
Requisitos Relacionados	RF04, RF05.

	Especificação do Caso de Uso - Visualização do Histórico de Notas
Descrição	Este caso de uso descreve as etapas para que um usuário autorizado visualize o histórico de notas de um aluno individual.
Atores	Voluntário, Coordenador.
Pré-condições	O aluno já deve estar cadastrado no sistema e o usuário deve estar autenticado.
Pós-condições	O histórico de notas será exibido para o usuário.
Ações Normais	<ol style="list-style-type: none"> 1. O usuário acessa a área de histórico de notas. 2. O usuário seleciona o aluno cujo histórico deseja visualizar. 3. O sistema exibe todas as notas do aluno organizadas por bimestres e disciplinas.

	Especificação do Caso de Uso - Visualização do Histórico de Notas
Observação	O sistema deve garantir que o usuário tenha permissão para acessar as informações do aluno.
Requisitos Relacionados	RF06.

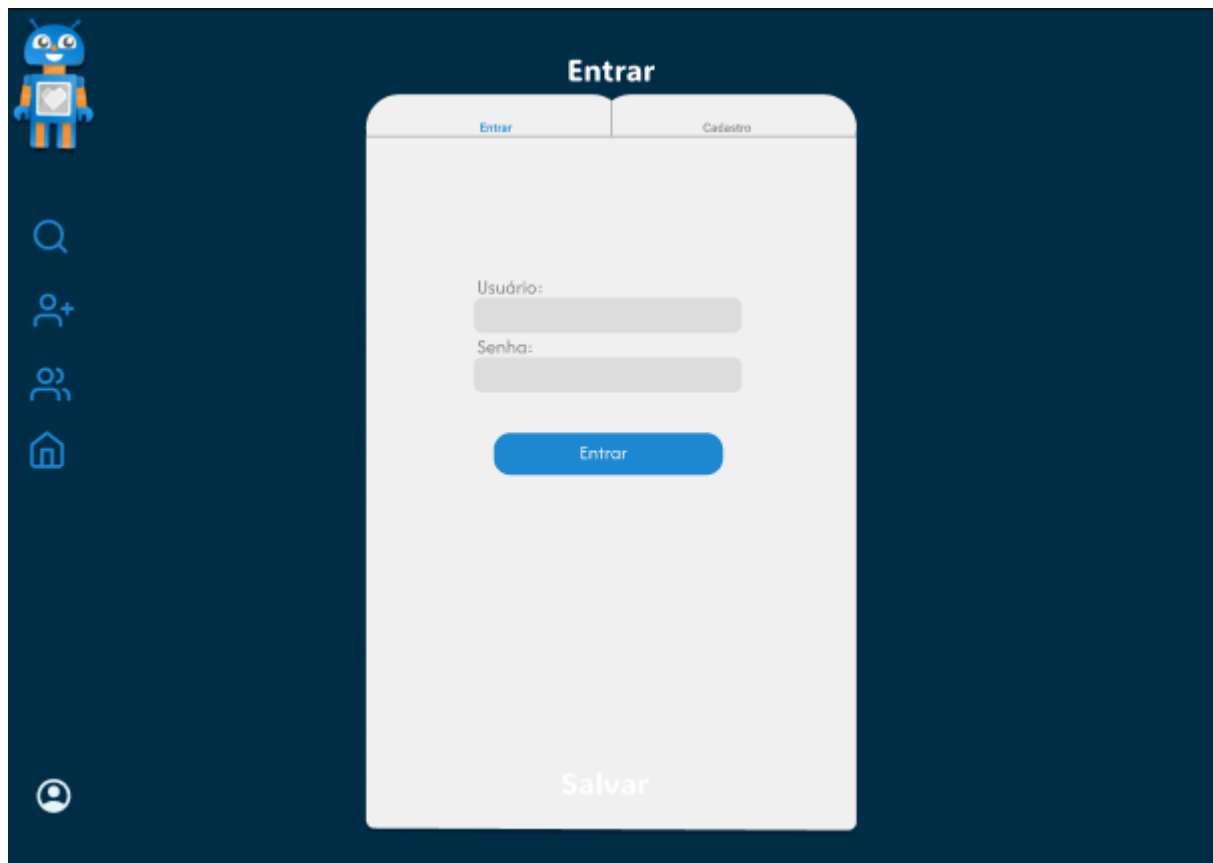
	Especificação do Caso de Uso - Controle de Acesso
Descrição	Este caso de uso descreve como o sistema garante que apenas usuários autorizados possam acessar ou modificar os dados.
Atores	Voluntário, Coordenador
Pré-condições	O usuário deve estar autenticado no sistema.
Pós-condições	O acesso ao sistema será concedido ou negado com base nas permissões do usuário.
Ações Normais	<ol style="list-style-type: none"> 1. O usuário tenta acessar uma área do sistema. 2. O sistema verifica o nível de acesso do usuário. 3. Se autorizado, o sistema concede o acesso. Se não, o acesso é negado. 4. O sistema registra todas as tentativas de acesso para fins de auditoria.
Observação	As permissões de acesso são configuradas pelo administrador, que pode definir diferentes níveis de privilégio.
Requisitos Relacionados	RF07.

	Especificação do Caso de Uso - Inserir ou Editar Informações de Alunos e Famílias
Descrição	Este caso de uso descreve como um voluntário pode inserir ou editar informações sobre alunos e suas famílias.
Atores	Voluntário.
Pré-condições	O voluntário deve estar autenticado no sistema.
Pós-condições	As informações atualizadas serão salvas no sistema e registradas para fins de auditoria.
Ações Normais	<ol style="list-style-type: none"> 1. O voluntário acessa o cadastro de um aluno ou de uma família. 2. O sistema exibe as informações já registradas.

	<ol style="list-style-type: none"> 3. O voluntário insere ou edita os dados. 4. O sistema valida os novos dados. 5. O sistema salva as alterações no banco de dados e registra a mudança para auditoria. 6. O sistema exibe uma mensagem confirmando a atualização.
Observação	O sistema deve garantir que todas as alterações sejam auditadas para fins de segurança.
Requisitos Relacionados	RF08, RF09.

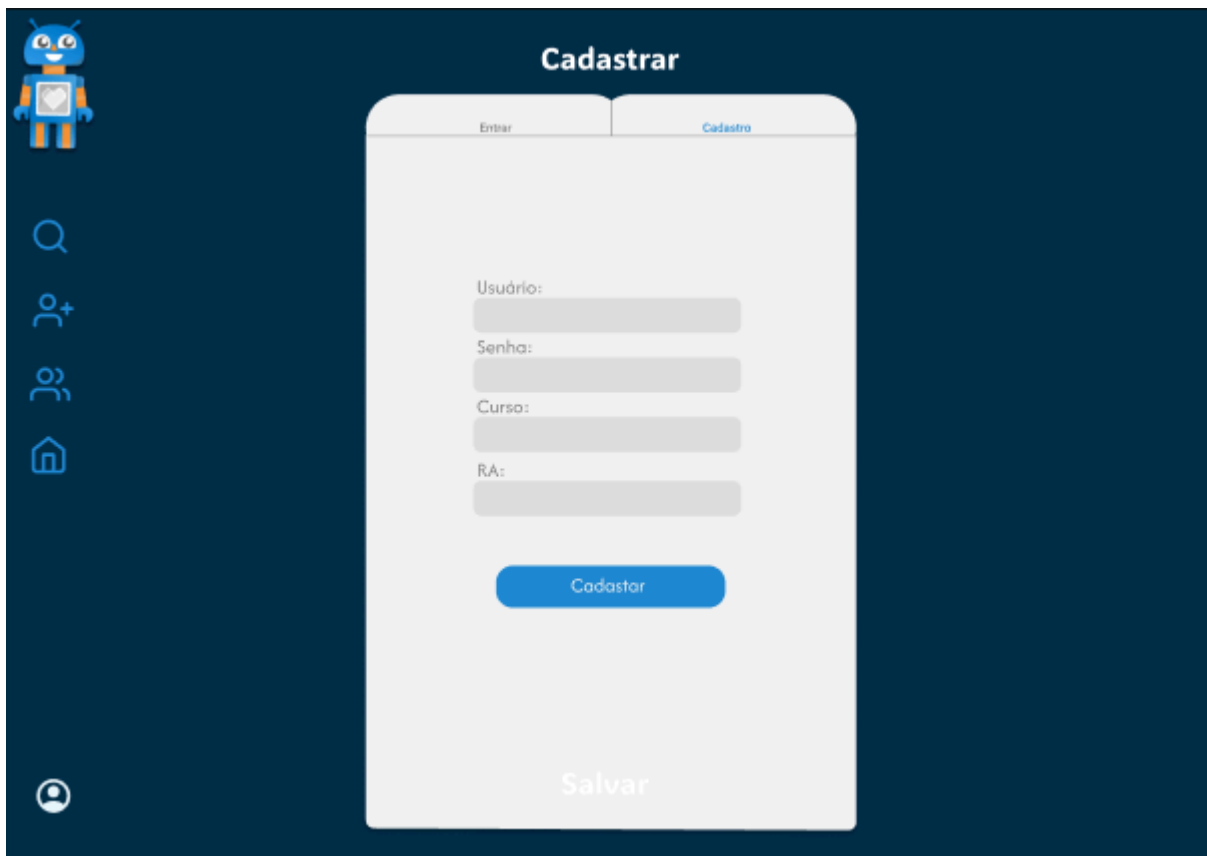
	Especificação do Caso de Uso - Geração de Alertas Automáticos para Mudanças nas Notas
Descrição	Este caso de uso descreve como o sistema gera alertas automáticos quando há mudanças significativas nas notas dos alunos.
Atores	Sistema (Automático).
Pré-condições	O aluno deve ter notas registradas e o sistema deve monitorar as alterações.
Pós-condições	Um alerta será enviado se houver uma mudança significativa nas notas.
Ações Normais	<ol style="list-style-type: none"> 1. O sistema monitora as notas dos alunos. 2. O sistema detecta uma mudança significativa nas notas (melhora ou queda). 3. O sistema gera um alerta automático. 4. O alerta é enviado para os usuários autorizados (ex: coordenadores, professores).
Observação	O sistema deve permitir a personalização do que constitui uma "mudança significativa" nas notas.
Requisitos Relacionados	RF10.

4.4 Protótipos de Telas



Tela 1: Tela de login do usuário

A Tela 1 mostra a tela de login do usuário, onde deve ser informado usuário e senha cadastrados. Ao clicar em “Entrar” são verificadas as credenciais e então o usuário tem acesso ao sistema.



Tela 2: Tela de cadastro do usuário

A Tela 2 mostra a tela de cadastro do usuário, onde é possível criar um novo usuário para acessar o sistema, são necessárias as informações, usuário, senha, curso e RA do aluno.

Cadastro de Aluno

Aluno:

Nome completo: Série: Idade:

Escola:

O aluno recebe atendimento médico/clínico: ☐

Notas:

Nota Antiga
Bimestre: 2024/3
Matemática: Ciências:
Alterar

Nota Nova
Bimestre: 2025/1
Matemática: Ciências:
Adicionar

Frequencia:

Data de matrícula: Data de encerramento:

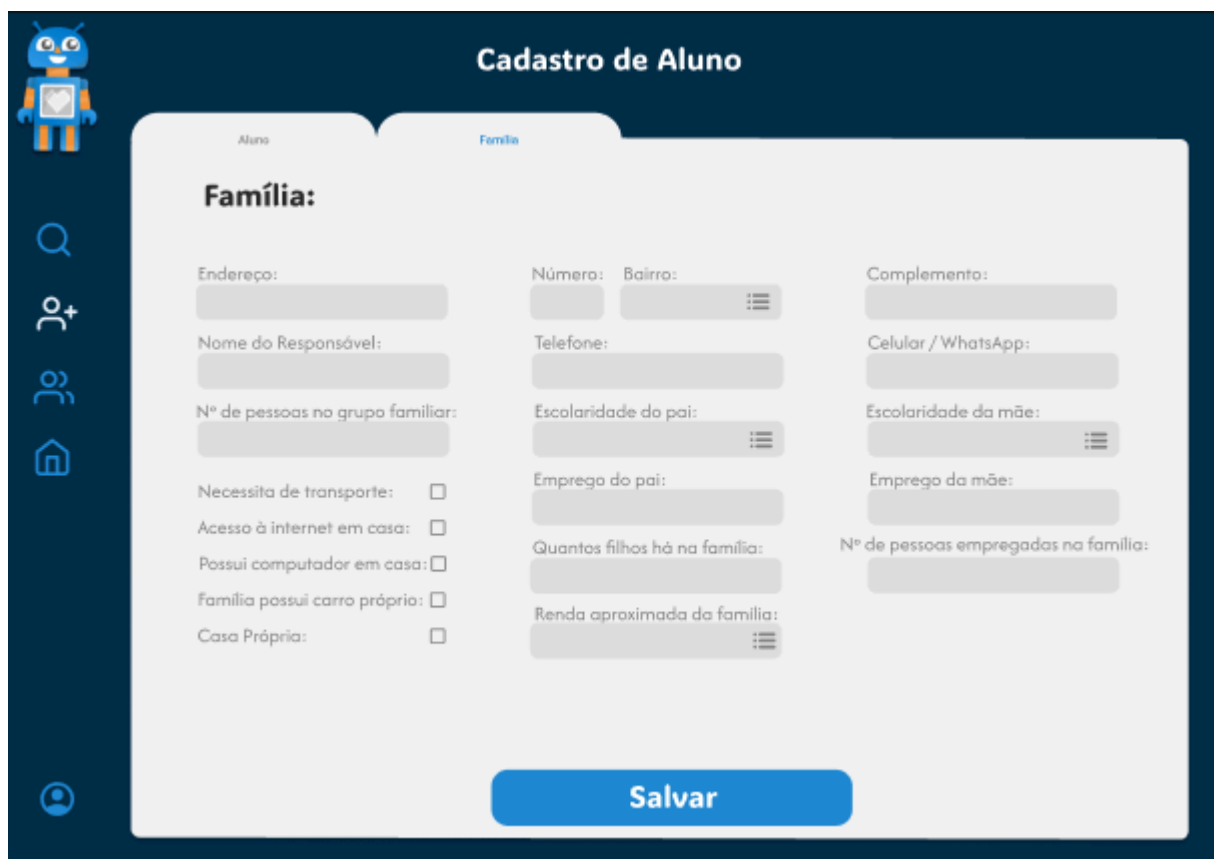
Ativo: ☒

Salvar

Tela 3: Tela de cadastro de aluno

A Tela 3 mostra a tela de cadastro de aluno, onde é possível criar um novo registro de aluno e também atualizar um aluno antigo, informações como Nome, Série e Idade podem ser preenchidas apenas digitando no teclado, a Escola por sua vez deve conter uma lista de escolas para ser selecionada, caso seja uma nova escola também pode ser cadastrada, “o aluno recebe atendimento médico/clínico” é uma variável booleana que vem por padrão como falsa.

Em Notas é possível alterar alguma nota já cadastrada caso algum erro ocorra na hora de cadastrar, e também criar uma nova data no semestre atual, em frequência o Campo Ativo vem como verdadeiro e se torna falso ao adicionar data de encerramento.



Cadastro de Aluno

Aluno Família

Família:

Endereço:

Número: Bairro:

Complemento:

Nome do Responsável:

Telefone:

Celular / WhatsApp:

Nº de pessoas no grupo familiar:

Escolaridade do pai:

Escolaridade da mãe:

Necessita de transporte: ☐

Emprego do pai:

Emprego da mãe:

Acesso à internet em casa: ☐

Quantos filhos há na família:

Possui computador em casa: ☐

Renda aproximada da família:

Família possui carro próprio: ☐

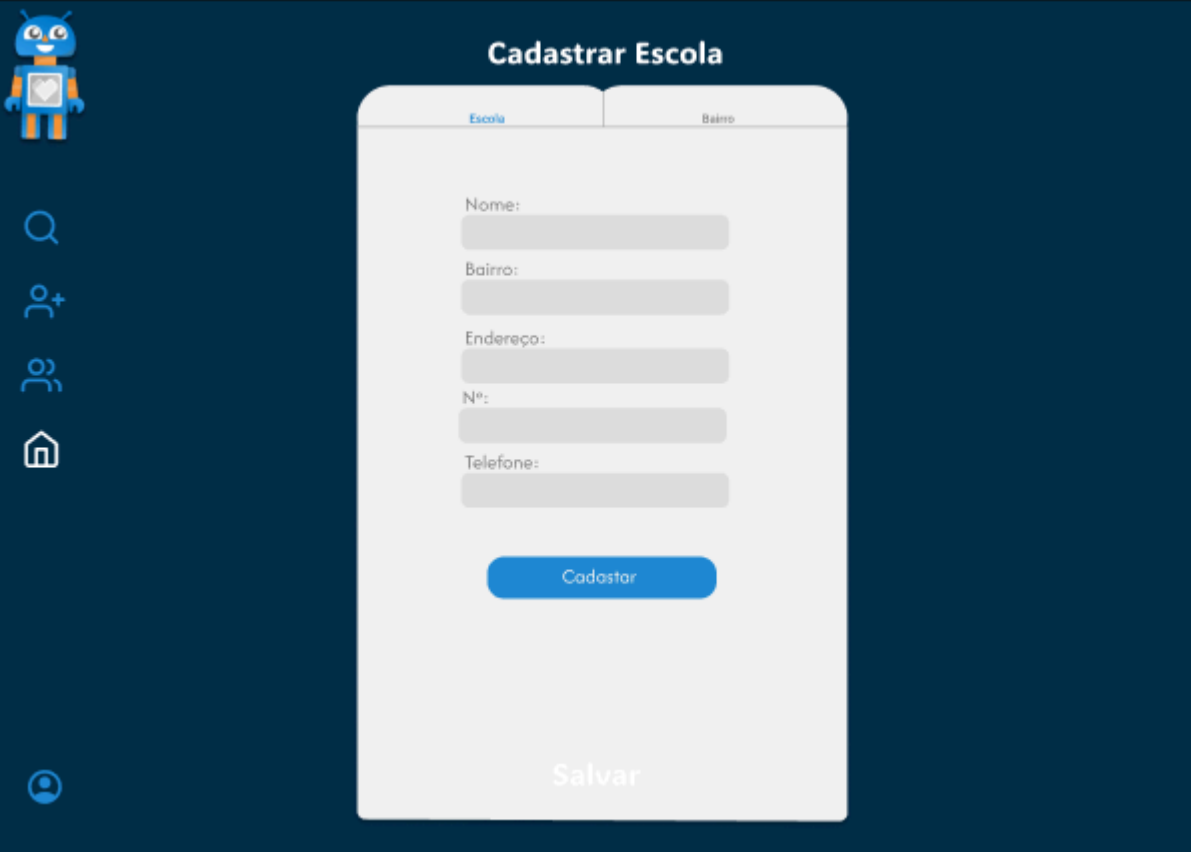
Nº de pessoas empregadas na família:

Casa Própria: ☐

Salvar

Tela 4: Segunda tela de cadastro de aluno

A Tela 4 mostra a continuação da tela de cadastro do aluno, onde são inseridas informações a respeito da família do aluno, como geralmente essas informações provavelmente não serão alteradas após o cadastro elas ficam na segunda aba. Informações como bairro, escolaridade dos pais e renda devem conter uma lista com as possíveis respostas.



Cadastrar Escola

Escola Bairro

Nome:

Bairro:

Endereço:

Nº:

Telefone:

Cadastrar

Salvar

Tela 5: Cadastrar escola

A Tela 5 só deve ser usada em caso de uma nova escola, informações além do nome da escola podem ser preenchidas.

Cadastrar Bairro

Escola Bairro

Nome do Bairro:

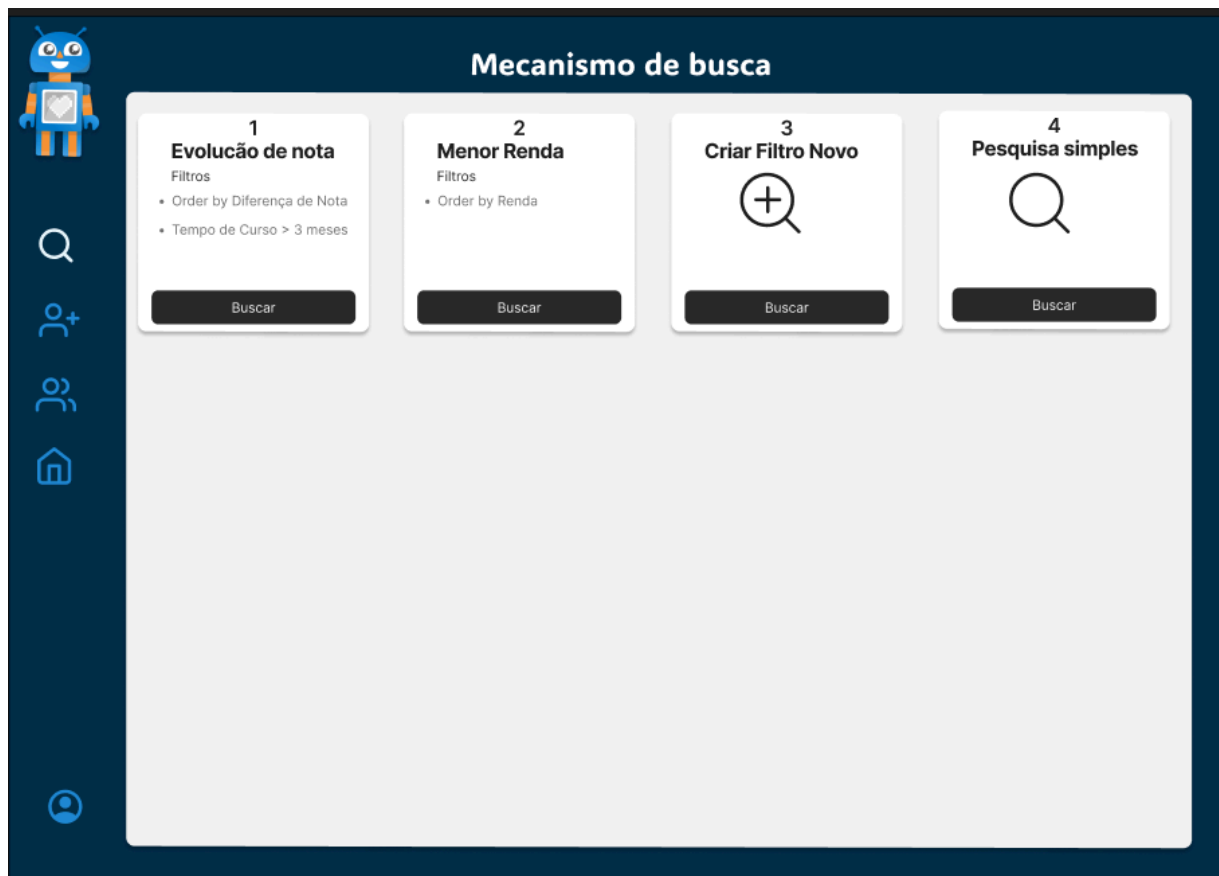
Cidade

Cadastrar

Salvar

Tela 6: Cadastrar bairro

A Tela 6 foi feita para ser usada caso um aluno resida em um bairro ainda não cadastrado, como por exemplo bairros de outras cidades próximas.



Tela 7: Mecanismo de busca

A Tela 7 mostra o mecanismo principal de busca, onde o usuário pode criar novos filtros e salvá-los para pesquisar novamente mais tarde, o usuário também pode fazer uma busca simples e não salvar os filtros. Ao clicar em “buscar” o usuário é direcionado para tabela de alunos com os filtros que ele escolheu.

5 Análise do Sistema

5.1 Modelo do Banco de Dados

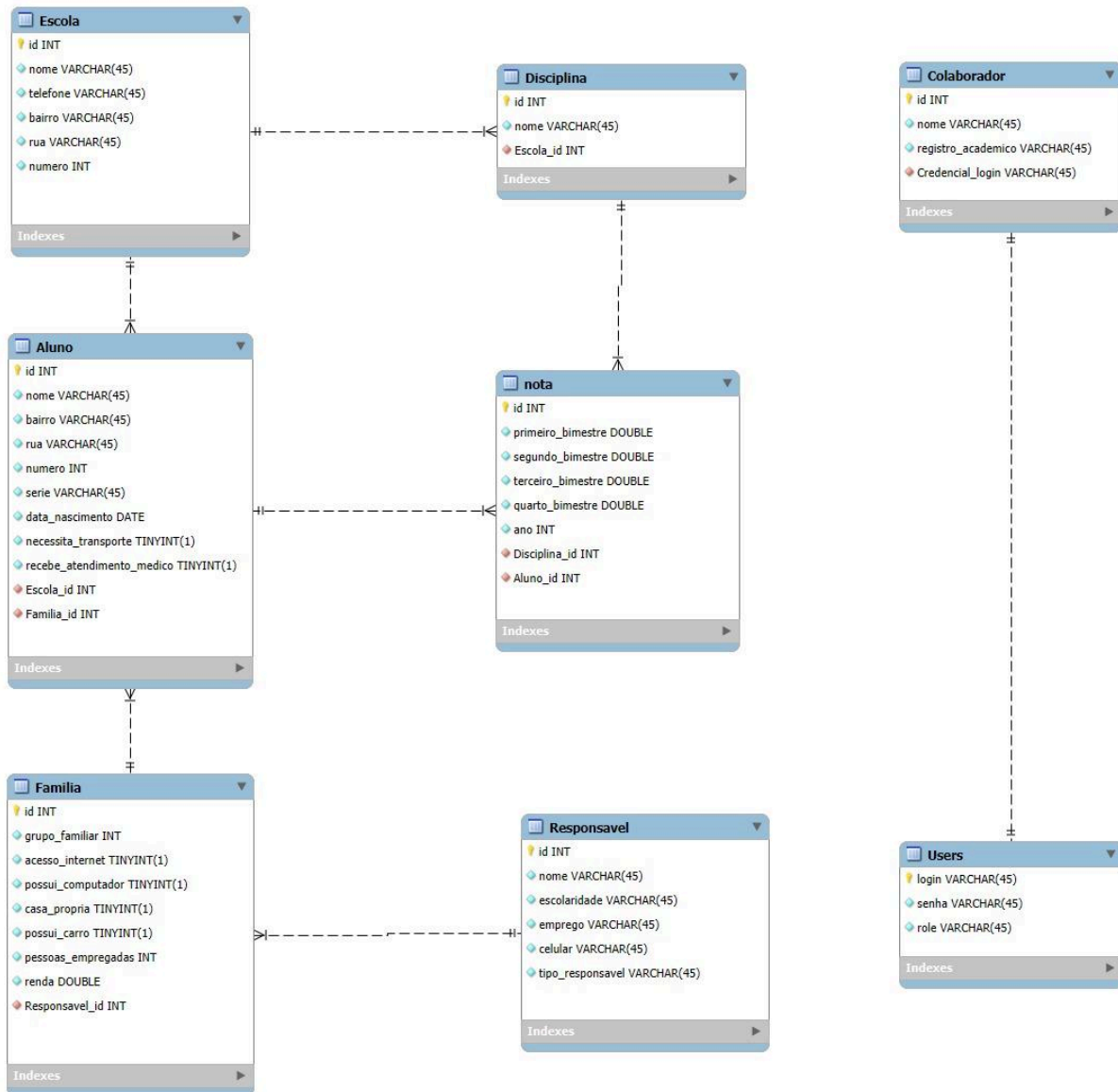


Figura 1: Modelo do banco de dados

5.2 Dicionário de dados

Tabela		Escola		
Descrição		Armazenará as informações das escolas		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único da escola	INT		PK/IDENTITY
nome	Nome da escola	VARCHAR	45	NOT NULL
telefone	Telefone de contato	VARCHAR	45	NOT NULL
bairro	Bairro da escola	VARCHAR	45	NOT NULL
rua	Rua da escola	VARCHAR	45	NOT NULL
numero	Número do endereço	INT		NOT NULL

Tabela		Disciplina		
Descrição		Armazenará as informações das disciplinas, estudadas pelo aluno na escola.		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único da disciplina	INT		PK/IDENTITY
nome	Nome da disciplina	VARCHAR	45	NOT NULL
Escola_id	chave estrangeira identificador único da Escola	INT		FK (Escola.id)

Tabela		nota		
Descrição		Armazenará as notas dos alunos nas disciplinas estudadas na escola durante o ano em que participou do ELLP.		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único da nota	INT		PK/IDENTITY
Aluno_id	chave estrangeira identificador único do Aluno	INT		FK(Aluno.id)
Disciplina_id	chave estrangeira identificador único da Disciplina	INT		FK (Disciplina.id)
primeiro_bimestre	nota do aluno durante o primeiro bimestre na disciplina vinculada	DOUBLE		
segundo_bimestre	nota do aluno durante o segundo bimestre na disciplina vinculada	DOUBLE		
terceiro_bimestre	nota do aluno durante o terceiro bimestre na disciplina vinculada	DOUBLE		
quarto_bimestre	nota do aluno durante o quarto bimestre na disciplina vinculada	DOUBLE		
ano	ano em que o aluno participou do ELLP	INT		NOT NULL

Tabela		Aluno		
Descrição		Armazenará dados sobre os alunos		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único do aluno	INT		PK/IDENTITY
nome	Nome completo do aluno	VARCHAR	45	NOT NULL
bairro	Nome do bairro do aluno	VARCHAR	45	NOT NULL
rua	Nome da rua onde reside aluno	VARCHAR	45	NOT NULL
numero	Número residencial de onde reside o aluno	INT		NOT NULL
serie	Série do aluno na escola.	VARCHAR	45	NOT NULL
data_nascimento	Data de nascimento do aluno	DATE		NOT NULL
necessita_transporte	Indica se o aluno tem necessidade de ser levado ao ELLP: 1 = Sim, 0 = Não	TINYINT	1	NOT NULL

recebe_atendimento_medico	Indica se o aluno recebe atendimento médico: 1 = Sim, 0 = Não	TINYINT	1	NOT NULL
Familia_id	chave estrangeira identificador único da Família	INT		FK (Familia.id)

Tabela		Familia		
Descrição		Armazenará dados sobre a família que o aluno pertence, somente das pessoas que moram na mesma casa que ele.		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único da familia	INT		PK/IDENTITY
grupo_familiar	Quantidade de pessoas que moram na mesma casa	INT		NOT NULL
acesso_internet	Indica se a família tem acesso a internet: 1 = Sim, 0 = Não.	TINYINT	1	NOT NULL
possui_computador	Indica se a família possui computador 1 = Sim, 0 = Não.	TINYINT	1	NOT NULL
casa_propria	Indica se a família possui casa própria 1 = Sim, 0 = Não	TINYINT	1	NOT NULL

possui_carro	Indica se a família possui carro próprio 1 = Sim, 0 = Não	TINYINT	1	NOT NULL
peessoas_employadas	Indica quantidade de pessoas que possuem emprego na família	INT		NOT NULL
renda	Renda total da família	DOUBLE		NOT NULL

Tabela		Responsável		
Descrição		Armazenará dados sobre o responsável do aluno		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único do responsável	INT		PK/IDENTITY
nome	Nome completo do responsável	VARCHAR	45	NOT NULL
escolaridade	Escolaridade do responsável	VARCHAR	45	NOT NULL
emprego	Cargo do responsável	VARCHAR	45	NOT NULL
celular	Número do celular do responsável	VARCHAR	45	NOT NULL

tipo_responsavel	Representa o parentesco do responsável com o aluno	VARCHAR	45	NOT NULL
Familia_id	chave estrangeira identificador único da Família	INT		FK (Familia.id)

Tabela		Colaborador		
Descrição		Armazenará as informações dos colaboradores do ELLP		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
id	Identificador único de colaborador	INT		PK/IDENTITY
Credencial_login	chave estrangeira identificador único de credencial	VARCHAR	45	FK(Credencial.login)
nome	Nome do colaborador	VARCHAR	45	NOT NULL
registro_academico	Registro acadêmico do colaborador	VARCHAR	45	NOT NULL

Tabela		Credencial		
Descrição		Armazenará as informações de login dos colaboradores		
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio (PK, FK, NOT NULL, IDENTITY)
login	Identificador único de login, login do colaborador no sistema	VARCHAR	45	PK/IDENTITY
senha	Senha do colaborador no sistema	VARCHAR	45	NOT NULL
tipo_usuario	Indicador de tipo de usuário do sistema 1 = Coordenador, 0 = Voluntário	TINYINT	1	NOT NULL

Usuário do banco de dados	
Nome do usuário	Permissão
Coordenador	Acesso total ao banco de dados.
Voluntário	Permissão para inserir, alterar e excluir dados nas tabelas Aluno, Família e Responsável, além de visualizar relatórios.

5.3 Diagrama de Classes

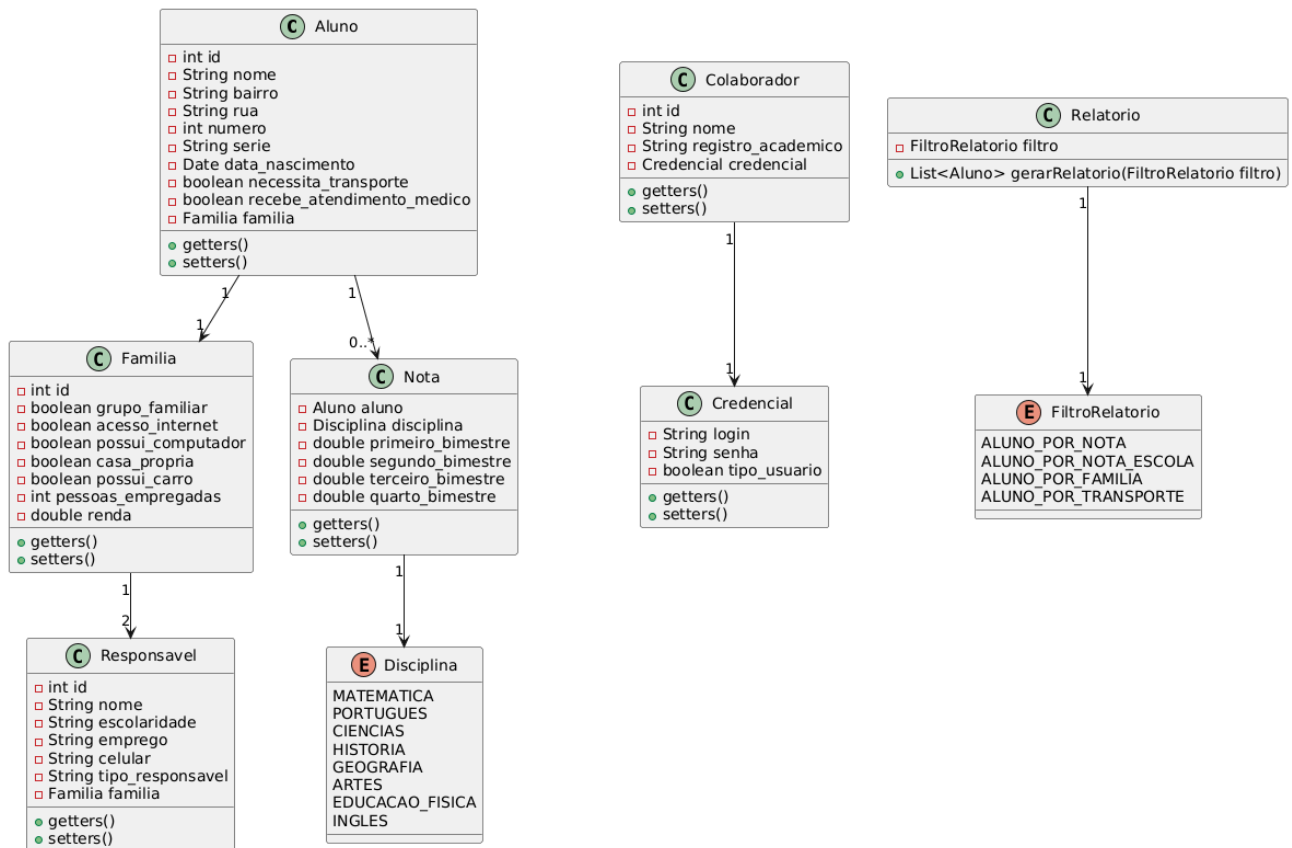


Figura 2: Diagrama de classes

5.4 Diagrama de Atividades

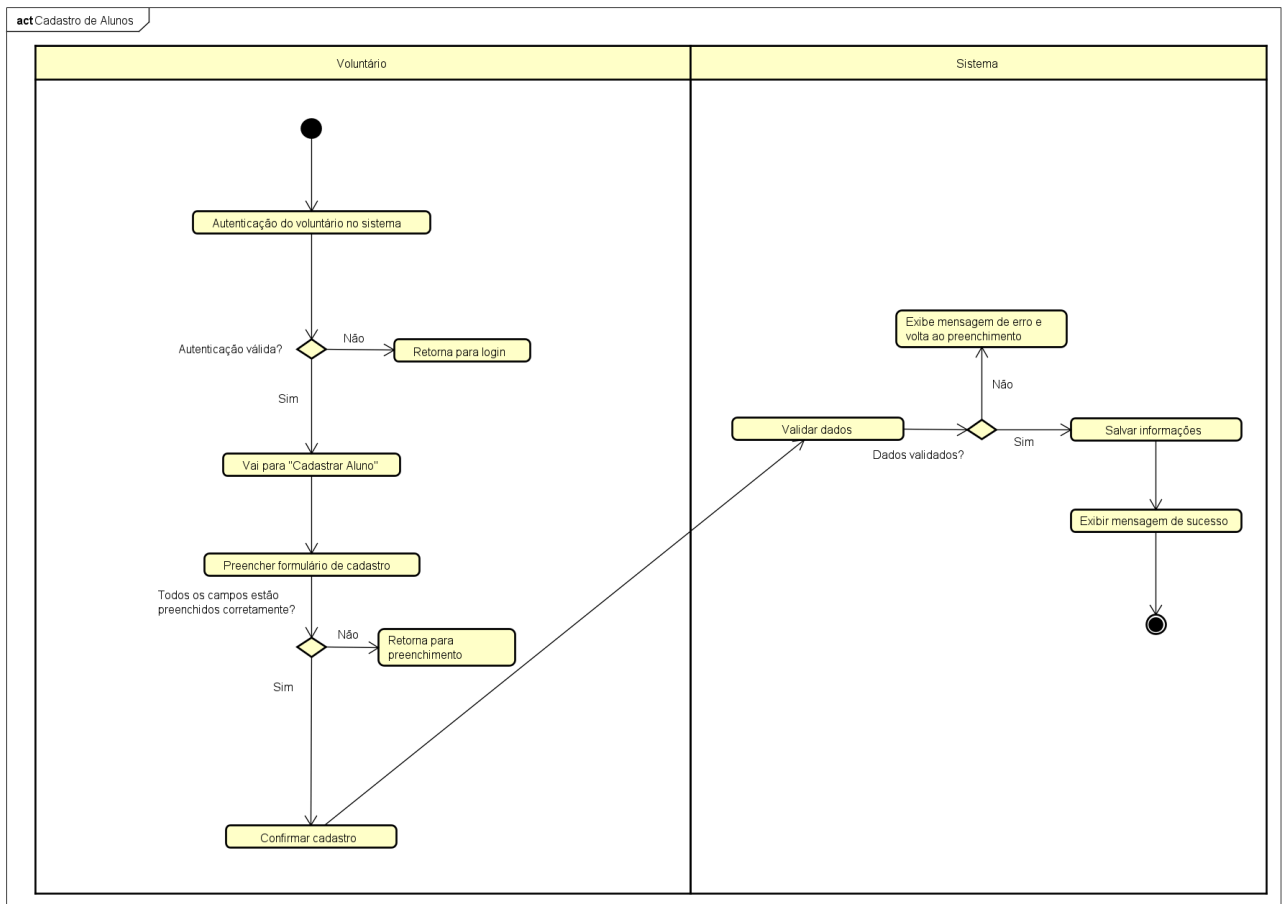


Figura 3: Diagrama de Atividades - Cadastro de Alunos

O diagrama da figura 3 descreve o fluxo de atividades para cadastrar um aluno no sistema. O processo começa com a autenticação do voluntário, segue para o preenchimento do formulário de cadastro com informações obrigatórias, e termina com a validação e salvamento dos dados no banco. Caso ocorram erros na validação, o sistema retorna ao preenchimento para correção.

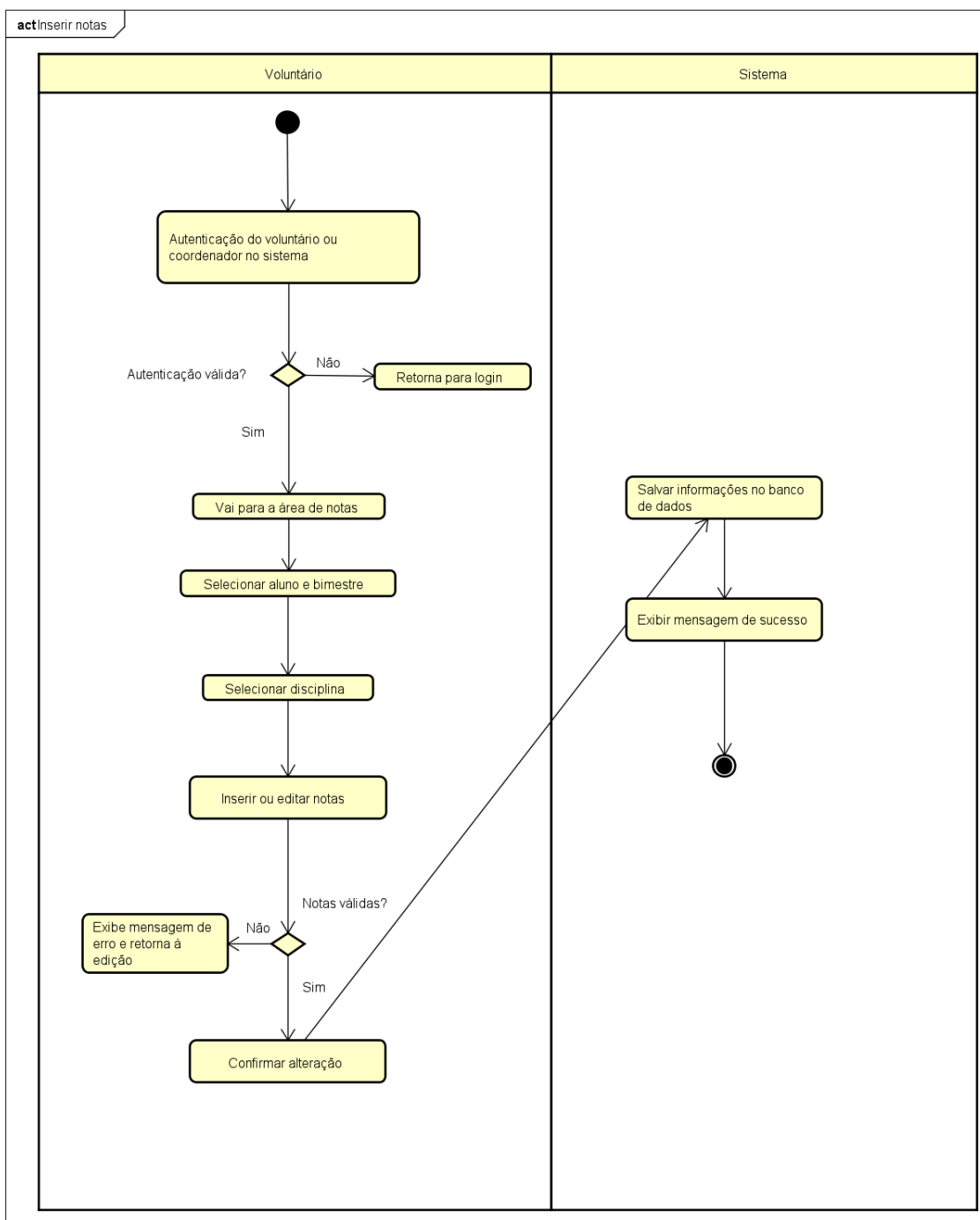


Figura 4: Diagrama de Atividades - Inserir notas

O diagrama da figura 4 apresenta o fluxo para inserir ou atualizar as notas dos alunos. Após a autenticação, o voluntário seleciona o aluno e o bimestre, insere ou edita as notas, e confirma as alterações. O sistema valida as informações e salva os dados no banco. Mensagens de sucesso ou erro são exibidas conforme a validação

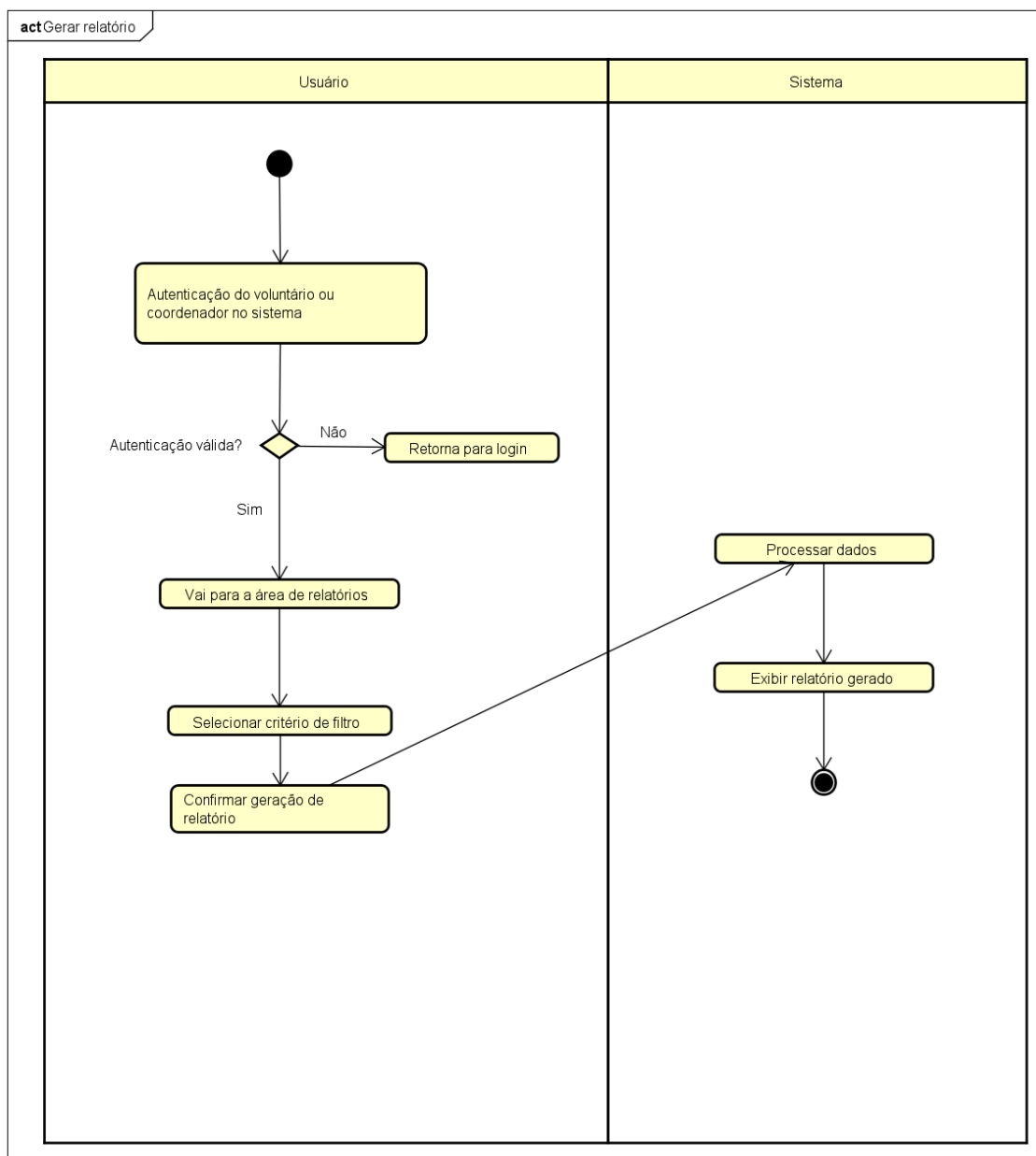


Figura 5: Diagrama de Atividades - Gerar relatório

O diagrama da figura 5 ilustra o processo de geração de relatórios acadêmicos e socioeconômicos. O usuário, após se autenticar, seleciona os critérios de filtro desejados e solicita a geração do relatório. O sistema processa os dados e exibe o relatório gerado. Caso os filtros estejam incompletos, o usuário é orientado a ajustá-los.

6 Análise do Sistema

6.1 Descrição do código

Front-end

O front-end do projeto está organizado em uma estrutura modular, seguindo boas práticas de desenvolvimento com React. Abaixo está a descrição da organização dos arquivos e componentes:

- src/
 - assets/ # Arquivos estáticos (imagens, ícones, etc.)
 - components/ # Componentes reutilizáveis
 - ListaBimestre/ # Componente para seleção de bimestres
 - Dashboard.jsx # Página inicial do sistema
 - DetalhesAluno.jsx # Página de detalhes do aluno
 - FormAluno.jsx # Formulário de cadastro de aluno
 - LancamentoNotas.jsx # Página para lançamento de notas
 - ... # Outros componentes
 - contexts/ # Contextos do React (gerenciamento de estado global)
 - AuthContext.js # Contexto de autenticação
 - services/ # Serviços para comunicação com o back-end
 - alunoService.js # Serviço para operações com alunos
 - familiaService.js # Serviço para operações com famílias
 - notaService.js # Serviço para operações com notas
 - App.js # Componente principal da aplicação
 - index.js # Ponto de entrada da aplicação

Componentes Principais

App.js:

- Componente raiz da aplicação.
- Define as rotas da aplicação usando react-router-dom.

AuthContext.js:

- Contexto de autenticação para gerenciar o estado de login do usuário.

Dashboard.jsx:

- Página inicial do sistema, exibindo estatísticas e links para outras funcionalidades.

FormAluno.jsx:

- Formulário para cadastro e edição de alunos.

LancamentoNotas.jsx:

- Página para lançamento de notas dos alunos.

DetalhesAluno.jsx:

- Página que exibe detalhes completos de um aluno, incluindo notas e informações familiares.

ListaBimestre.jsx:

- Componente reutilizável para seleção de bimestres.

Serviços

- alunoService.js: Realiza operações CRUD para alunos.
- familiaService.js: Realiza operações CRUD para famílias.
- notaService.js: Realiza operações CRUD para notas.
- escolaService.js: Realiza operações CRUD para escolas.

Bibliotecas Utilizadas

- React: Biblioteca principal para construção da interface.
- React Router DOM: Para gerenciamento de rotas.
- Axios: Para requisições HTTP ao back-end.
- React Hook Form: Para validação e gerenciamento de formulários.
- React Toastify: Para exibição de notificações.

Back-end (Spring Boot)

O back-end do projeto foi desenvolvido utilizando Spring Boot, seguindo uma arquitetura em camadas (controller, service, repository). Abaixo está a descrição da organização dos arquivos e classes:

- src/main/java/com.app.ellp/
 - config # Configurações da aplicação
 - CorsConfig.java # Configuração de CORS
 - DotenvConfig.java # Configuração de variáveis de ambiente
 - SecurityConfig.java # Configuração de segurança
 - Exception # Tratamento de exceções
 - ExceptionHandlerClass.java # Classe para tratamento global de exceções
 - Module # Módulos de aplicação
 - Aluno # Módulo de alunos
 - Colaborador # Módulo de colaboradores
 - Disciplina # Módulo de disciplina
 - Escola # Módulo de escola
 - Familia # Módulo de família
 - Nota # Módulo de nota
 - Responsavel # Módulo de responsáveis
 - User # Módulo de usuários
 - Security # Configurações de segurança
 - DTOs # DTOs relacionados à segurança
 - SecurityFilter.java # Filtro de segurança
 - TokenService.java # Serviço para geração e validação de tokens JWT
 - Web # Controladores e documentação da API
 - AuthenticationController.java # Controlador de autenticação
 - AuthenticationDocumentation.java # Documentação da API de autenticação
 - EllpApplication.java # Classe principal do Spring Boot

Descrição dos Pacotes e Classes

Pacote config

Contém as classes de configuração da aplicação.

CorsConfig.java:

- Configura as políticas de CORS (Cross-Origin Resource Sharing) para permitir requisições de diferentes origens.

DotenvConfig.java:

- Configura o carregamento de variáveis de ambiente a partir de um arquivo .env.

SecurityConfig.java:

- Configura a segurança da aplicação, incluindo autenticação, autorização e proteção de endpoints.

Pacote exception

Contém o tratamento global de exceções.

ExceptionHandlerClass.java:

- Classe que centraliza o tratamento de exceções da aplicação, retornando respostas padronizadas para erros.

Pacote module

Contém os módulos da aplicação, cada um com sua própria estrutura interna (controller, service, repository, domain).

Pacote security

Contém as configurações e serviços relacionados à segurança da aplicação.

Pacote web

Contém os controladores e a documentação da API.

DTOs/:

- Contém DTOs (Data Transfer Objects) relacionados à segurança, como LoginRequest e TokenResponse.

SecurityFilter.java:

- Filtro que intercepta requisições e valida tokens JWT.

TokenService.java:

- Serviço responsável por gerar e validar tokens JWT.

AuthenticationController.java:

- Controlador REST para autenticação de usuários (login, logout, etc.).

AuthenticationDocumentation.java:

- Documentação da API de autenticação usando Swagger.

EllpApplication.java:

- Classe principal do Spring Boot, responsável por iniciar a aplicação.

Fluxo de Funcionalidades

Autenticação:

- O usuário faz login através do AuthenticationController.
- O TokenService gera um token JWT, que é retornado ao front-end.
- O SecurityFilter valida o token em requisições subsequentes.

Cadastro de Alunos:

- O front-end envia os dados do aluno para o AlunoController.
- O AlunoService valida os dados e persiste no banco de dados através do AlunoRepository.

Lançamento de Notas:

- O front-end envia as notas para o NotaController.
- O NotaService valida e persiste as notas no banco de dados.

Visualização de Detalhes:

- O front-end solicita os detalhes de um aluno ao AlunoController.
- O AlunoService busca os dados no banco de dados e retorna ao front-end.

6.2 Implantação

Passo a Passo para Execução do Software

Requisitos Necessários

Antes de executar o projeto, certifique-se de que os seguintes requisitos estão instalados e configurados no seu ambiente:

Front-end (React)

- Node.js: Versão 16.x ou superior.
- NPM ou Yarn: Gerenciadores de pacotes para instalar as dependências do projeto.
- O NPM é instalado automaticamente com o Node.js.
- Yarn (opcional).

Back-end (Spring Boot)

- Java Development Kit (JDK): Versão 17 ou superior.
- Maven: Ferramenta de build e gerenciamento de dependências.

Ferramentas Adicionais

- Git: Para clonar o repositório do projeto.
- IDE: Recomenda-se o uso de uma IDE para desenvolvimento, como:
 - IntelliJ IDEA (para back-end)
 - Visual Studio Code (para front-end)

Configuração do Ambiente

Clonar o Repositório

- Abra o terminal ou prompt de comando.
- Execute o seguinte comando para clonar o repositório do projeto:

```
git clone https://github.com/gubetiati/Projeto-ELLP.git
```

```
git clone https://github.com/rxluk/ELLP.git
```

Configuração do Back-end

- Abra o arquivo de configuração do Spring Boot:
 - Localização: ELLP/src/main/resources/application.properties
- Configure as propriedades do banco de dados MongoDB:

```
spring.application.name=ellp  
spring.data.mongodb.uri=<URI do MongoDB>  
server.port=8080  
api.security.token.secret=<chave-secreta>
```

Executar o Back-end

- Navegue até a pasta do back-end com 'cd ELLP'
- Compile e execute o projeto usando o Maven:
 - mvn clean install
 - mvn spring-boot:run

Configuração do Front-end

Instalar as dependências

- Navegue até a pasta do front-end:
 - cd Projeto-ELLP
- Instale as dependências do projeto
 - npm install

Configurar a API do Back-end

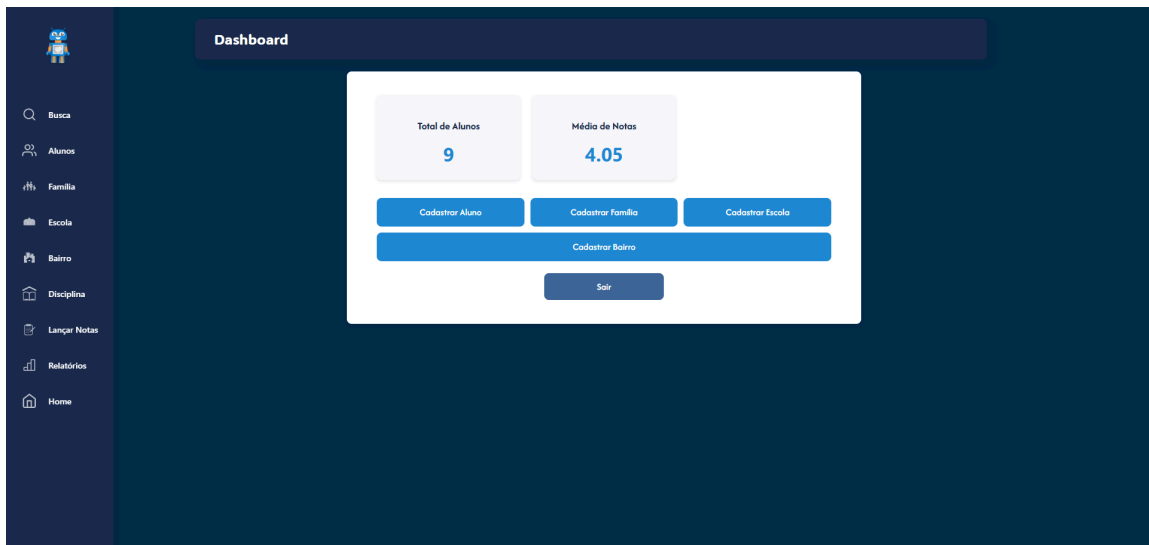
- Abra o arquivo de configuração da API 'setupProxy.js:

- Em 'target' verifique se a URL está apontando para a API:
 - target: 'http://localhost:8080',

Executar o Front-end

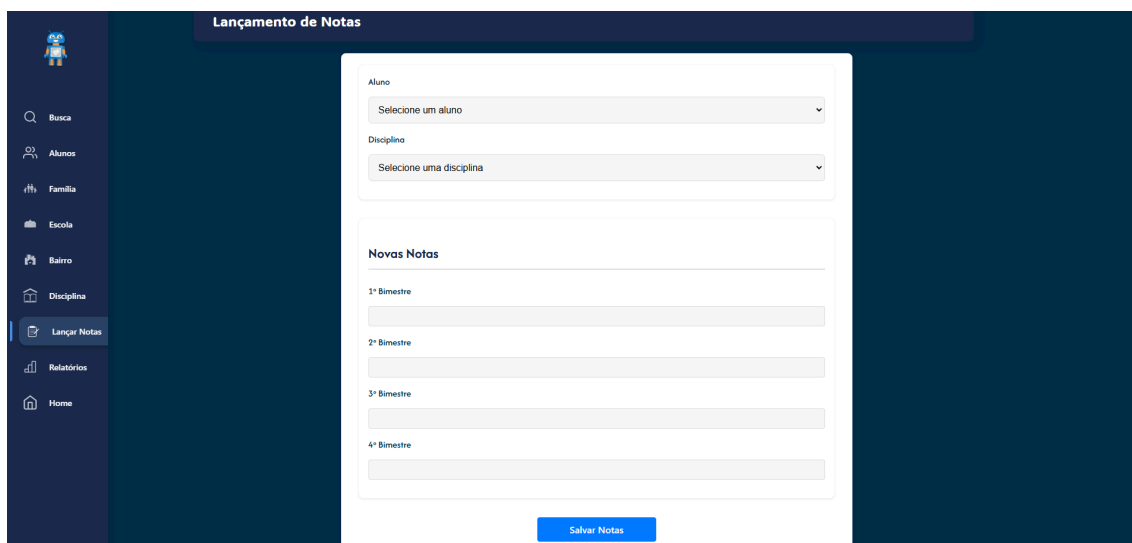
- Inicie o servidor de desenvolvimento
 - npm start
- Inicie a aplicação:
 - npm run dev

6.3 Telas principais



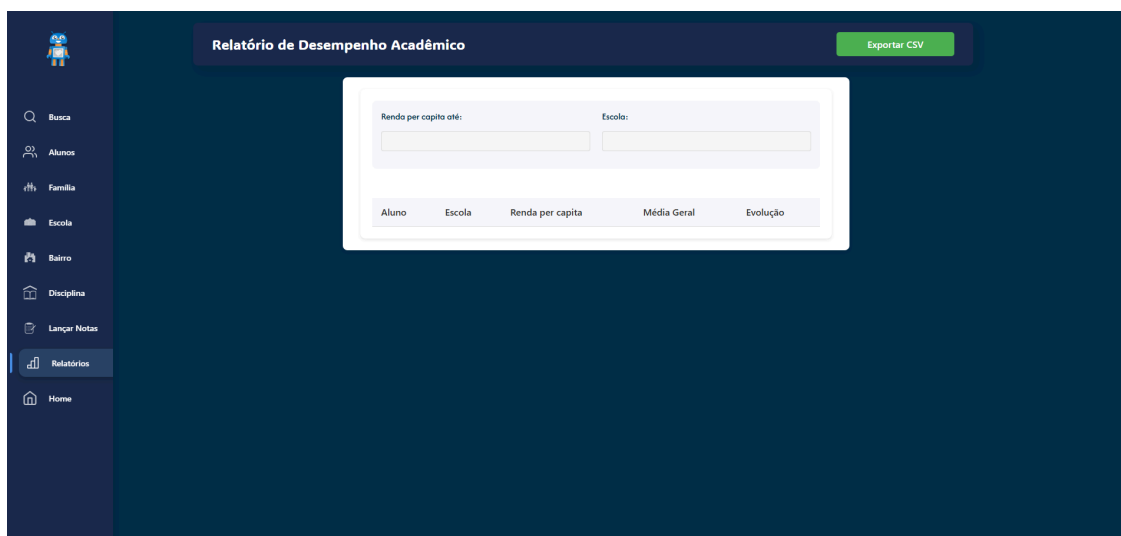
Tela 1: Menu principal

A tela inicial, apresentada logo após o login, oferece uma visão geral das principais funcionalidades do sistema. Nela, o usuário tem acesso rápido a todas as ferramentas essenciais, com uma navegação intuitiva e organizada.



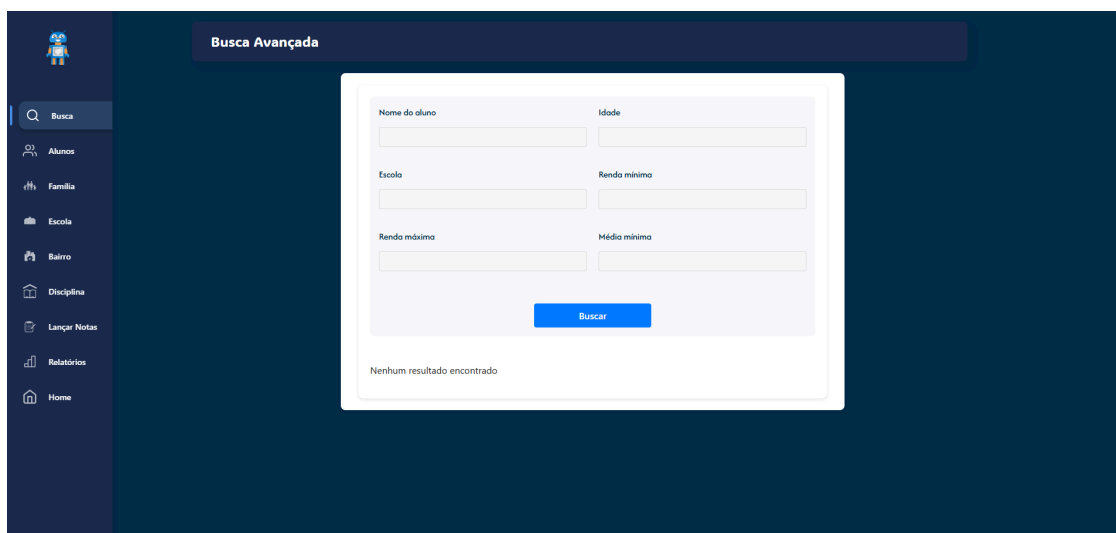
Tela 2: Lançamento de Notas

A imagem mostra a tela de lançamento de notas, o usuário deve inserir uma disciplina e aluno já cadastrados. Após a seleção o usuário pode inserir notas correspondentes de cada bimestre.



Tela 3: Relatório

A tela Relatório de Desempenho Acadêmico exibe todos os alunos, com informações detalhadas como nome da escola, renda per capita, média geral e a evolução geral das notas por bimestre. A evolução é calculada comparando as notas de cada bimestre com o anterior, mostrando o progresso de forma acumulada. O usuário tem a opção de filtrar os alunos por uma renda per capita máxima e também pode selecionar escolas específicas para análise



Tela 4: Busca

A tela de Busca Avançada permite que o usuário busque alunos com uma base de filtros detalhados, como nome, idade, escola, faixa de renda mínima e máxima, e média de nota mínima. Além disso, ao realizar a busca o usuário pode selecionar cada aluno para ver todos os dados cadastrados.

7 Considerações Finais

Um dos maiores desafios durante o desenvolvimento do projeto foi a dificuldade em manter uma frequência alta de reuniões, especialmente devido ao período de fim de ano, com festas e viagens.

Quanto às melhorias, o sistema pode ser aprimorado com a implementação de novos filtros na busca, proporcionando uma experiência mais personalizada. Além disso, a possibilidade de abrir telas de cadastro sobre outras, como permitir o cadastro de uma família diretamente na tela de cadastro de alunos, tornaria o processo mais prático e eficiente.