



UNIVERSIDADE FEDERAL DO MARANHÃO
Bacharelado Interdisciplinar em Ciência e Tecnologia
Projeto e Desenvolvimento de Software

Bianca Milena Da Silva Laranjeira
Jamilly Vitória Ferreira Barbosa
Marcos Vinícius Jansem Oliveira
Paulo Arthur Lima Brito
Yhago Luis Maciel De Matos

Sistema de Correção de Scripts Usando IA

São Luís, 2025.

Sumário

1 Introdução	2
2 Descrição Geral do Software	2
2.1 Situação de Atuação	2
2.2 Atores	3
3 Levantamento de Requisitos	3
3.1 Requisitos Funcionais	4
3.2 Requisitos Não Funcionais	5
4 Diagramas	6
4.1 Diagramas de Caso de Uso	6
4.2 Diagrama de Classe	9
4.3 Diagrama de Atividade	11
4.4 Diagrama de Estado	18
4.5 Diagrama de Sequência	21

1 Introdução

A resolução de problemas em scripts de programação é um desafio constante para alunos de tecnologia, especialmente para aqueles que estão em níveis iniciante e intermediário. Dúvidas sobre sintaxe, lógica e boas práticas de codificação são frequentes e, muitas vezes, demandam grande esforço por parte dos professores para serem sanadas. Essa sobrecarga não apenas consome tempo, mas também pode impactar negativamente o processo de aprendizado.

Para atender a essa demanda, foi desenvolvido o Sistema de Correção de Scripts, uma solução que utiliza inteligência artificial para analisar e corrigir códigos de forma eficiente e didática. O sistema é projetado para detectar erros de sintaxe e lógica, sugerir correções automáticas e fornecer explicações claras sobre os problemas encontrados. Além disso, oferece recomendações baseadas em boas práticas de programação, contribuindo para o desenvolvimento técnico dos alunos.

A plataforma apresenta uma interface web intuitiva e acessível, permitindo que os alunos submetam seus códigos e recebam feedback imediato. A integração com APIs de IA possibilita um processamento robusto e preciso dos scripts, enquanto o feedback detalhado promove um aprendizado autônomo e contínuo.

Esse sistema foi solicitado por um professor de programação que busca centralizar as dúvidas recorrentes em um ambiente dedicado, reduzindo a necessidade de atendimentos individuais via outros meios, como o WhatsApp. A expectativa é que a ferramenta não apenas alivie a carga de trabalho do professor, mas também motive os alunos a serem mais independentes na resolução de problemas, melhorando sua confiança e habilidades técnicas.

Com a implementação do Sistema de Correção de Scripts, espera-se criar um ambiente de aprendizado mais eficiente, interativo e escalável, no qual tanto professores quanto alunos possam maximizar seu potencial e alcançar resultados mais expressivos no ensino e prática de programação.

2 Descrição Geral do Software

O Sistema de Correção de Scripts é desenvolvido para otimizar o processo de aprendizado e desenvolvimento de software. Projetado para atender às necessidades de alunos e professores de tecnologia, o sistema oferece uma solução abrangente para a análise, correção e validação de scripts de código.

2.1 Situação de Atuação

A criação do Sistema de Correção de Scripts foi motivada por desafios recorrentes enfrentados por alunos de tecnologia, especialmente aqueles em níveis iniciante e intermediário. Esses alunos frequentemente enfrentam dificuldades com sintaxe, lógica e boas práticas de codificação, o que demanda grande esforço por parte dos professores para serem

sanadas. Essa sobrecarga não apenas consome tempo, mas também pode impactar negativamente o processo de aprendizado.

2.2 Atores

A aplicação conta com três atores principais, cada um com suas definições e privilégios específicos, conforme apresentado na Tabela 1.

Tabela 1: Definição dos Atores

Ator	Definição e Privilégios de Acesso
Usuário	Representa o papel geral que engloba as interações básicas tanto do aluno quanto do professor: <ul style="list-style-type: none">● Realizar login no sistema.● Realizar cadastro.● Redefinir a senha.
Professor	Possui funções adicionais para o gerenciamento de turmas e suporte aos alunos: <ul style="list-style-type: none">● Gerenciar turmas: Criar, editar e visualizar turmas cadastradas.● Visualizar o painel de dúvidas: Acompanhar as principais dúvidas enviadas pelos alunos.● Responder dúvidas no painel: Fornecer explicações adicionais para dúvidas que a IA não conseguiu esclarecer.● Criar e gerenciar turmas: Organizar os grupos de alunos de acordo com as necessidades pedagógicas.
Aluno	Focado no uso das ferramentas de correção e aprendizado de programação: <ul style="list-style-type: none">● Enviar códigos para correção: Submeter scripts à análise da IA.● Consultar correções: Visualizar os erros identificados e conforme explicado pela IA.● Validar correções: Avaliar se a explicação dada pela IA foi satisfatória ou se precisa de mais detalhamento.

Fonte: Autor

3 Levantamento de Requisitos

O processo de levantamento de requisitos teve como objetivo identificar e documentar as funcionalidades, necessidades e expectativas do sistema, assegurando que ele atendesse às demandas dos usuários e aos objetivos do cliente. Para isso, foram realizadas entrevistas com o professor solicitante, com o intuito de compreender suas necessidades e os desafios relacionados ao ensino de programação. Esse processo permitiu à equipe identificar as

principais funcionalidades que deveriam estar presentes no sistema para atender aos requisitos levantados de forma eficaz.

3.1 Requisitos Funcionais

Os requisitos funcionais correspondem às principais funcionalidades do sistema, determinando como o software deve operar para atender às necessidades dos usuários. Esses requisitos estão detalhados na Tabela 2, que lista cada funcionalidade essencial identificada no levantamento de requisitos.

Tabela 2: Requisitos funcionais

Código	Nome	Descrição
RF01	Análise de erros	O sistema deve ser capaz de analisar os scripts submetidos pelos usuários, identificando erros de sintaxe , semântica e lógica , a fim de garantir maior precisão e qualidade nos.
RF02	Explicação dos Erros	Além de identificar os erros, o sistema deve fornecer explicações claras e detalhadas, permitindo que os usuários compreendam as falhas em suas falhas.
RF03	Correção de erros identificados	O sistema deve realizar a correção automática dos erros detectados, aplicando as melhores práticas de programação para garantir que o código corrigido seja funcional e eficiente.
RF04	Encaminhar dúvida professor	O sistema deve permitir o envio do primeiro prompt de correção de script para o painel de dúvidas do professor. O objetivo é que o professor possa fornecer explicações adicionais ou compartilhar materiais de apoio para auxiliar o aluno.
RF05	Geração de relatórios	O sistema deve ser capaz de identificar as principais dúvidas dos alunos e apresentar essas informações de forma gráfica, facilitando a análise do professor sobre os pontos fracos mais recorrentes entre os alunos.
RF06	Controle de Acesso	O sistema deve implementar um controle de acesso eficiente, garantindo que cada usuário tenha especificações específicas em seu perfil.
RF07	Diferença de acesso entre usuárias	Deve-se garantir que cada tipo de usuário (como administradores, professores e alunos) tenha acesso apenas às funcionalidades pertinentes ao seu papel, preservando a segurança e a integridade

		do sistema.
--	--	-------------

Fonte: Autor

Embora sete requisitos funcionais tenham sido mapeados, será possível entregar apenas três deles (RF01, RF02 e RF03). O protótipo será desenvolvido como um exemplo prático de como o software funcionará quando estiver completamente finalizado, oferecendo um panorama claro das funcionalidades esperadas. O foco será principalmente na funcionalidade de suporte ao aluno, que permitirá o uso de uma API integrada com IA generativa para realizar a correção de scripts. A decisão de limitar a entrega a esses três requisitos considera o cronograma já estabelecido e os recursos disponíveis, assegurando que a entrega seja viável e cumpra os objetivos dentro das restrições do projeto.

3.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as qualidades que o sistema deve atender, garantindo eficiência e proporcionando uma experiência satisfatória aos usuários. Esses requisitos estão detalhados na Tabela 3, que apresenta as características essenciais identificadas no levantamento de requisitos.

Tabela 3: Requisitos não funcionais

Código	Nome	Descrição
RNF01	Interface Intuitiva	O sistema deve fornecer uma interface amigável, de fácil navegação e acessível a todos os usuários, permitindo uma experiência ágil e sem complexidade.
RNF02	Velocidade de resposta	O sistema deve apresentar tempos de resposta rápidos para as ações realizadas pelos usuários, minimizando o tempo de espera e melhorando a eficiência.
RNF03	Segurança	O sistema precisa garantir a proteção dos dados dos usuários, utilizando medidas de segurança como criptografia, autenticação, autorização e proteção contra acessos não autorizados.
RNF04	Manutenibilidade	O sistema deve ser fácil de manter, permitindo a rápida identificação e correção de erros, além da escalabilidade para futuras expansões.
RNF05	Compatibilidade	O sistema precisa ser compatível com diferentes navegadores, dispositivos (móveis e desktops) e sistemas operacionais, garantindo um funcionamento eficiente em diversas plataformas.

Fonte: Autor

Embora tenham sido identificados cinco requisitos não funcionais, apenas dois deles (RNF01 e RNF04) serão atendidos no escopo atual do projeto.

4 Diagramas

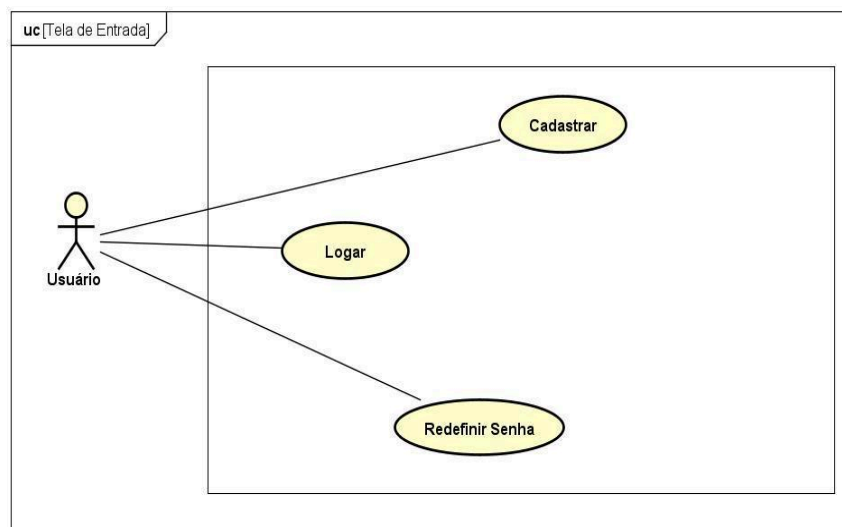
Nesta seção, irá apresentar o funcionamento do sistema utilizando diferentes tipos de diagramas: caso de uso, classe, atividade, sequência e estado.

4.1 Diagramas de Caso de Uso

O diagrama de caso de uso é uma representação visual que descreve as interações entre os atores do sistema e suas funcionalidades principais. Ele é utilizado para identificar, organizar e ilustrar como os usuários interagem com o sistema, permitindo uma visão clara das atividades que o software suporta. No contexto deste projeto, o diagrama foi elaborado para mapear os casos de uso relacionados aos quatro atores principais (Aluno, Professor, Sistema Interno e API), detalhando as ações que cada um pode realizar, bem como as relações entre essas ações e o sistema.

O primeiro caso de uso, ilustrado na Imagem 1, apresenta a relação do usuário com o sistema, representando as atividades comuns tanto para o professor quanto para o aluno. Essas atividades incluem: realizar cadastro, efetuar login e redefinir senha.

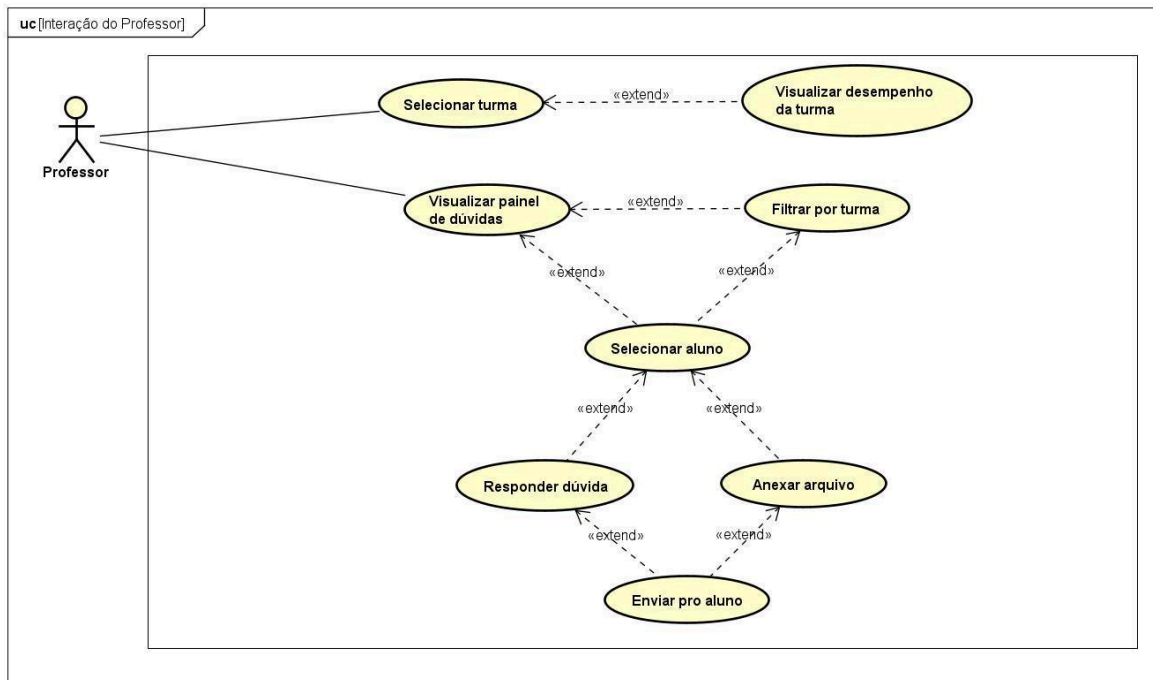
Imagem 1: Interação do Usuário



Fonte: Autor

O segundo caso de uso, ilustrado na Imagem 2, apresenta a interação do professor com o sistema. Nesse contexto, o professor pode filtrar por turma para visualizar o desempenho dos alunos ou acessar o painel de dúvidas, onde estão listadas as questões enviadas por alunos que não compreenderam as explicações fornecidas pela IA generativa.

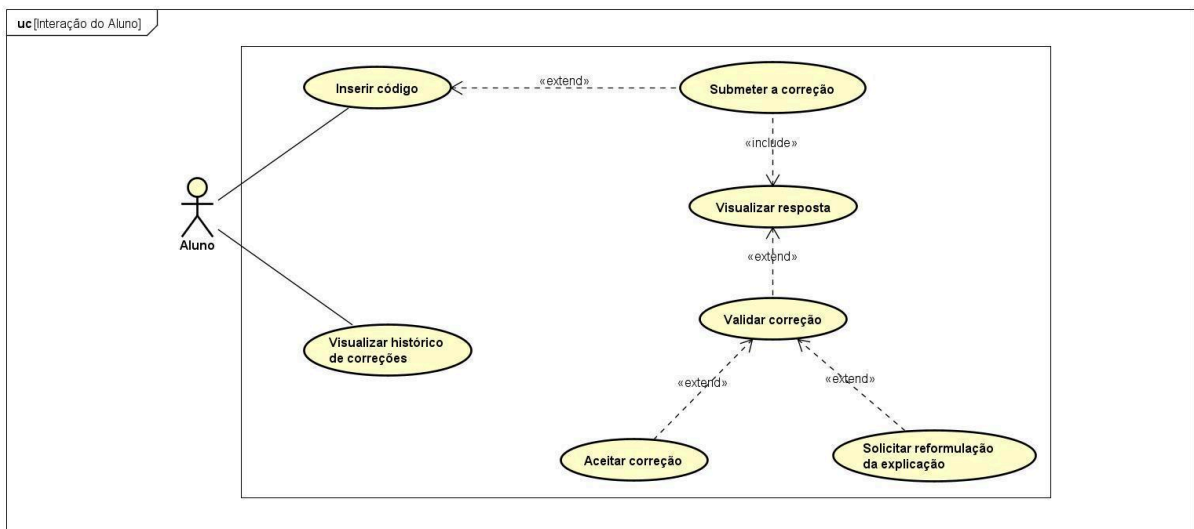
Imagem 2: Interação do Professor



Fonte: Autor

O terceiro caso de uso, ilustrado na Imagem 3, representa a interação do aluno com o sistema. Nessa interação, o aluno pode visualizar seu desempenho, com base nos temas em que encontrou maior dificuldade, ou submeter seu código para correção.

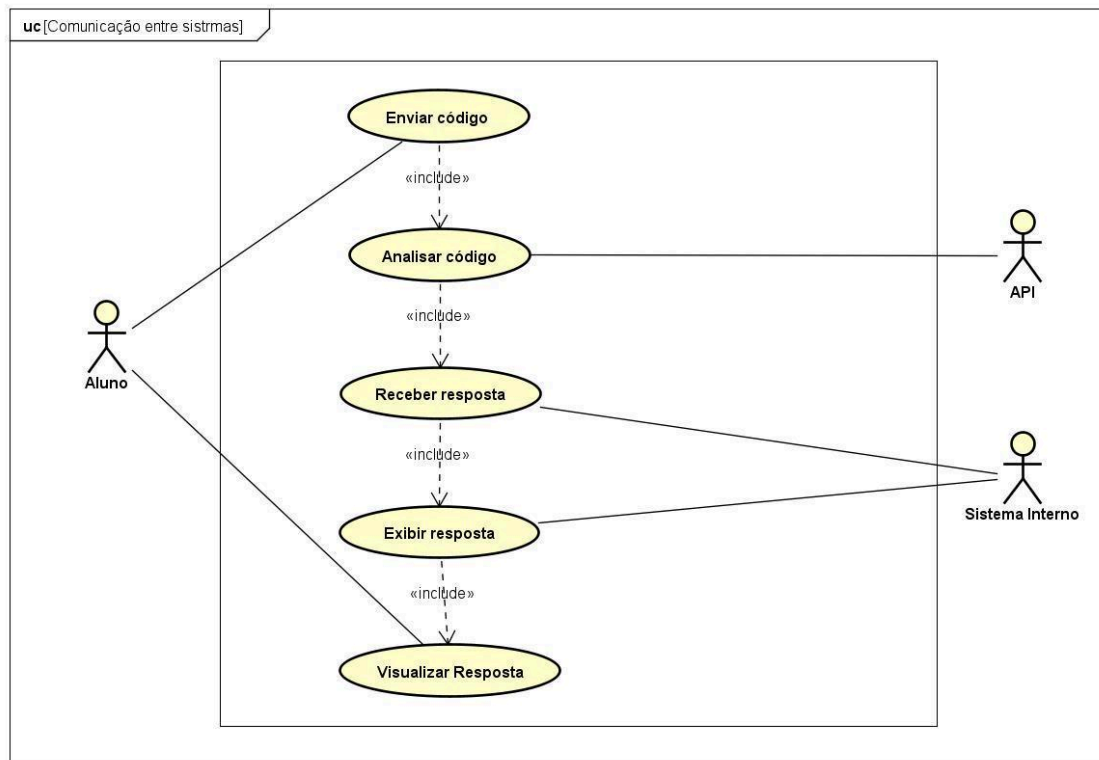
Imagem 3: Interação do Aluno



Fonte: Autor

O quarto diagrama de caso de uso, ilustrado na Imagem 4, demonstra o funcionamento da comunicação entre o sistema e a API durante o processo de correção do script enviado pelo aluno.

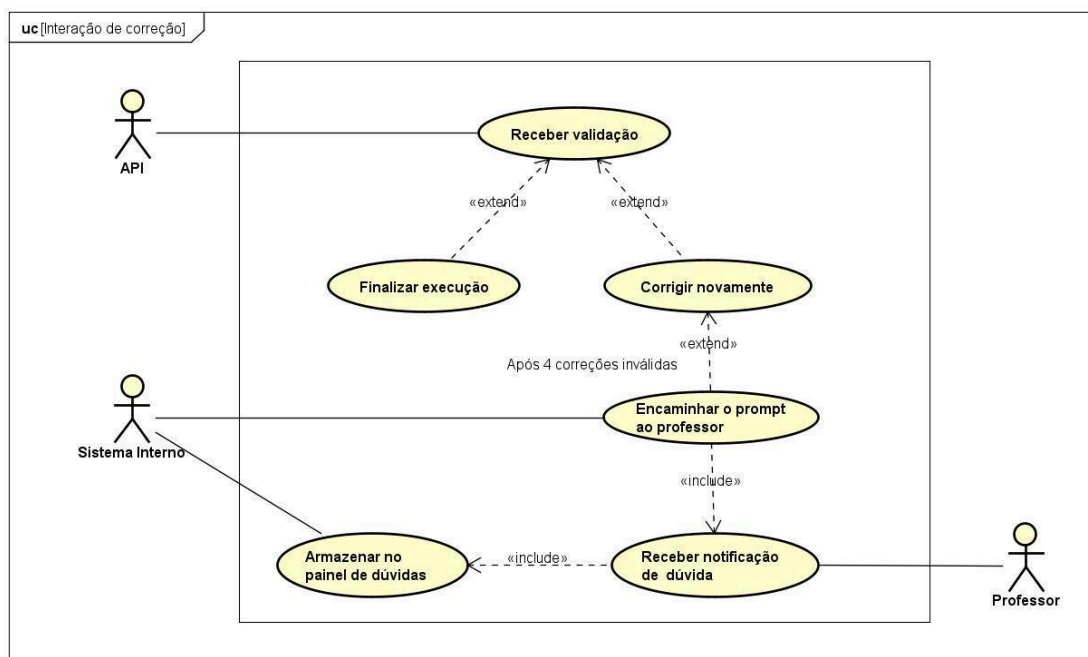
Imagem 4: Interação entre Sistemas



Fonte: Autor

O quinto caso de uso, ilustrado na Imagem 5, apresenta o funcionamento do mecanismo de validação da correção fornecida pela API. Ele demonstra o processo desde a avaliação realizada pelo aluno até o envio da dúvida para o painel do professor, caso o aluno não compreenda a explicação fornecida.

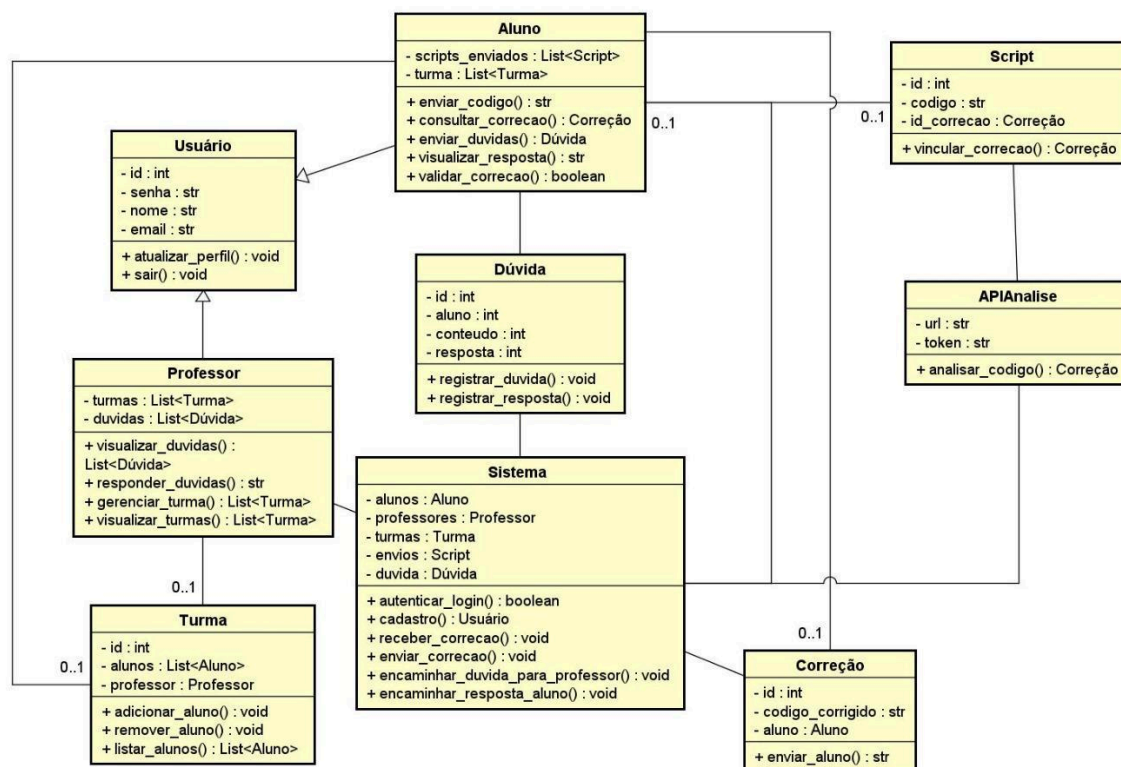
Imagem 5: Fluxo de Validação e Envio de Dúvidas ao Professor



4.2 Diagrama de Classe

O diagrama de classe é uma representação visual que descreve a estrutura estática do sistema, detalhando as classes, seus atributos, métodos e os relacionamentos entre elas. Ele é utilizado para modelar os elementos principais do sistema e suas interações, proporcionando uma visão clara e organizada da arquitetura e lógica do software. No contexto deste projeto, o diagrama de classe foi elaborado para representar as principais entidades do sistema, destacando suas responsabilidades, métodos associados e os vínculos que garantem a comunicação entre as partes.

Imagem 6: Diagrama de Classe



Fonte: Autor

O diagrama de classes, ilustrado na Imagem 6, descreve a arquitetura do sistema e as interações entre suas classes principais. De forma visual, ele apresenta as seguintes classes:

Usuário

○ Atributos:

- id: Identificador único do usuário.
- senha: Credenciais do usuário para login no sistema.
- nome: Nome completo do usuário.
- email: Endereço de e-mail do usuário.

○ Métodos:

- atualizar_perfil(): Permite ao usuário atualizar suas informações pessoais.
 - sair(): Método para realizar o logout do sistema.
- **Aluno**
- **Atributos:**
 - scripts_enviados: Lista de scripts submetidos pelo aluno.
 - turma: Lista de turmas associadas ao aluno.
- **Métodos:**
 - enviar_codigo(): Permite ao aluno submeter um script para análise.
 - consultar_correcao(): Retorna as correções feitas em um script submetido.
 - enviar_duvidas(): Registra dúvidas relacionadas aos scripts.
 - visualizar_resposta(): Exibe a resposta de uma dúvida enviada.
 - validar_correcao(): Método para confirmar a validade de uma correção recebida.

Professor

- **Atributos:**
 - turmas: Lista de turmas gerenciadas pelo professor.
 - duvidas: Lista de dúvidas enviadas pelos alunos.
- **Métodos:**
 - visualizar_duvidas(): Exibe as dúvidas pendentes para o professor responder.
 - responder_duvidas(): Método para responder dúvidas dos alunos.
 - gerenciar_turma(): Permite ao professor criar, editar ou excluir turmas.
 - visualizar_turmas(): Exibe as turmas associadas ao professor.

Sistema

- **Atributos:**
 - alunos: Lista de todos os alunos cadastrados no sistema.
 - professores: Lista de professores registrados.
 - turmas: Lista de turmas disponíveis.
 - envios: Histórico de scripts submetidos.
 - duvidas: Registro de todas as dúvidas enviadas.
- **Métodos:**
 - autenticar_login(): Método para autenticação de usuários.
 - cadastro(): Permite a inclusão de novos usuários no sistema.
 - receber_correcao(): Salva correções enviadas por professores ou pela API.
 - encaminhar_duvida_para_professor(): Encaminha dúvidas enviadas por alunos para os professores.

- `encaminhar_resposta_aluno()`: Retorna as respostas das dúvidas aos alunos.

APIAnalise

- **Atributos:**
 - `url`: URL da API para análise automatizada de scripts.
 - `token`: Chave de autenticação para uso da API.
- **Métodos:**
 - `analisar_codigo()`: Envia um script para análise automatizada e retorna as correções.

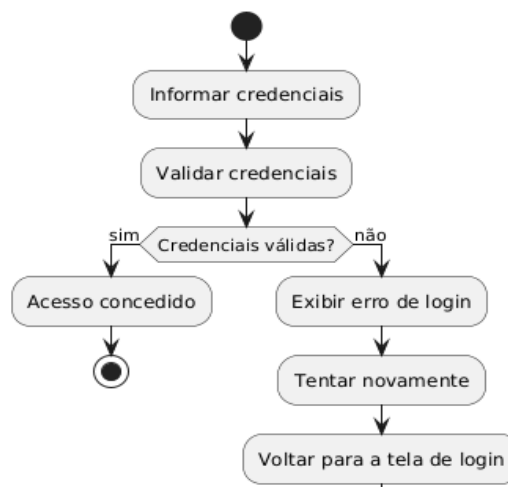
Correção

- **Atributos:**
 - `id`: Identificador único da correção.
 - `codigo_corrigido`: Código do script corrigido.
 - `aluno`: Referência ao aluno associado à correção.
- **Métodos:**
 - `enviar_aluno()`: Envia a correção para o aluno.

4.3 Diagrama de Atividade

O diagrama de atividade é, essencialmente, um fluxograma que ilustra o fluxo de controle entre diferentes atividades, sendo utilizado para modelar os aspectos dinâmicos do sistema. Serão apresentadas cinco atividades principais.

Imagem 7: Atividade de Login



Fonte: Autor

O fluxo de atividades relacionadas ao processo de entrada no sistema, conforme representado na Imagem 7. Esse processo abrange desde o início da interação do usuário até a validação das credenciais e o redirecionamento ou notificação, dependendo do resultado:

1. **Início:** Representa o início da atividade, marcando o ponto de partida do processo de login.
2. **Usuário preenche credenciais:** O usuário insere seus dados de acesso previamente cadastrados (ID e senha).
3. **Submissão de credenciais:** Após preencher os campos, o usuário submete as credenciais para validação pelo sistema.
4. **Validação das credenciais:** O sistema verifica se as credenciais fornecidas estão corretas.
 - **Sim:** Se as credenciais forem válidas, o sistema redireciona o usuário para a página principal.
 - **Não:** Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro, notificando o usuário para corrigir as informações e tentar novamente.
5. **Fim:** Indica a conclusão do fluxo de atividades, seja com o acesso à página principal ou após a notificação de erro.

Imagem 8: Atividade de Cadastro

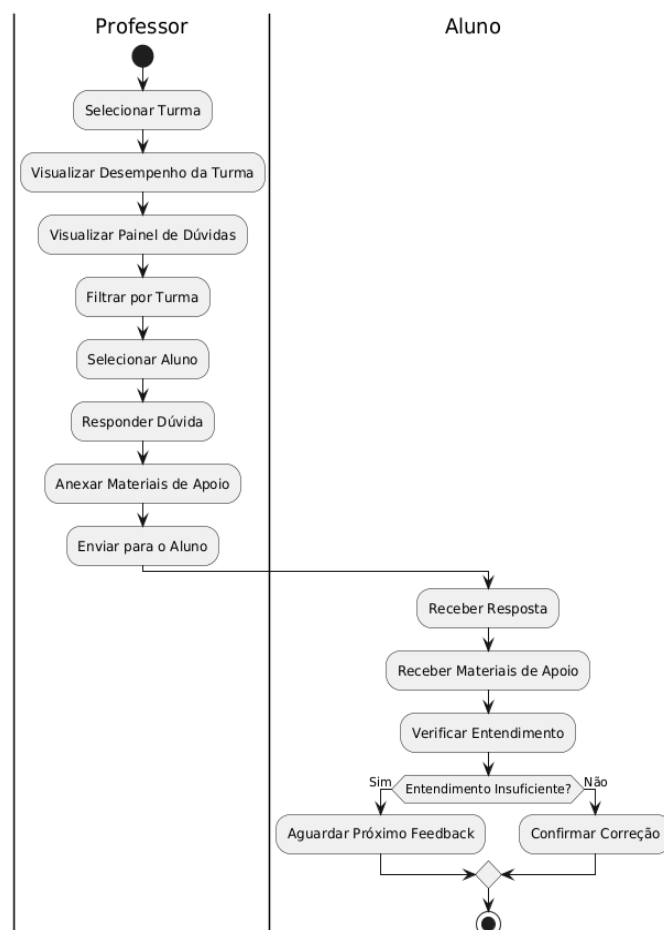


Fonte: Autor

O diagrama apresentado na Imagem 8 ilustra o fluxo de atividades relacionado ao processo de cadastro de novos usuários no sistema. Esse processo detalha cada etapa desde o início do cadastro até a validação:

1. **Início:** Representa o ponto de partida do processo de cadastro.
2. **Usuário acessa formulário de registro:** O usuário escolhe entre as opções de cadastro como aluno ou professor.
3. **Usuário preenche dados:** O usuário insere informações como nome, e-mail e senha.
4. **Submissão de dados:** Após preencher, o usuário envia os dados para validação pelo sistema.
5. **Validação dos dados:** O sistema verifica a consistência e a validade das informações fornecidas.
 - **Sim:** Se os dados forem válidos, o sistema os salva no banco de dados e exibe uma mensagem confirmando o sucesso do cadastro.
 - **Não:** Se os dados forem inválidos, o sistema notifica o erro ao usuário, que poderá corrigir as informações e reenviá-las.
6. **Fim:** Indica a conclusão do processo, seja com o cadastro concluído ou após a notificação de erro.

Imagem 9: Atividades do Professor



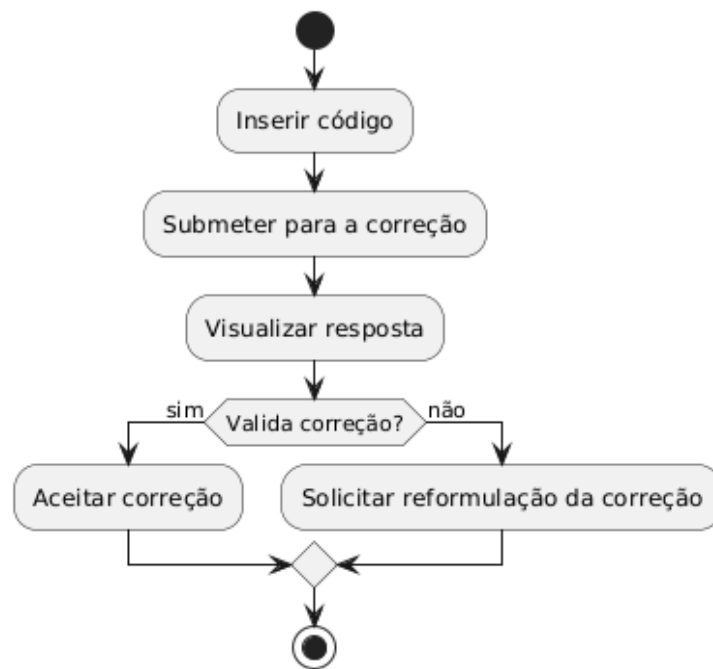
Fonte: Autor

O diagrama apresentado na Imagem 9 ilustra o fluxo de atividades relacionadas à interação do professor com o sistema, destacando como ele gerencia as dúvidas dos alunos e

acessa informações relevantes. O fluxo detalha a interface do professor e sua interação com outros processos integrados ao sistema:

1. **Início:** Representa o ponto de partida da interação do professor com o sistema.
2. **Selecionar turma:** O professor escolhe a turma que deseja visualizar.
3. **Visualizar desempenho da turma:** Permite ao professor analisar o desempenho geral dos alunos de uma turma específica.
4. **Painel de dúvidas:** O professor acessa o painel onde são listadas todas as dúvidas encaminhadas pelo sistema. Essas dúvidas são originadas de alunos que tiveram mais de quatro correções consideradas insuficientes pela IA.
5. **Filtrar por turma:** O professor organiza as dúvidas do painel, exibindo apenas as relacionadas a uma turma específica.
6. **Selecionar aluno:** O professor refina ainda mais a busca, selecionando dúvidas específicas de um aluno filtrado.
7. **Responder dúvida:** O professor resolve a dúvida do aluno selecionado, fornecendo explicações detalhadas.
8. **Anexar materiais:** Caso necessário, o professor pode enviar materiais de apoio junto à resposta.
9. **Receber resposta:** O aluno recebe a resposta fornecida pelo professor.
10. **Receber materiais de apoio:** Se o professor anexou materiais, o aluno os acessa por meio do sistema.
11. **Verificar entendimento:** O aluno verifica se compreendeu a resposta e/ou os materiais enviados.
12. **Entendimento insuficiente:** O sistema registra se o aluno não compreendeu a resposta:
 - **Sim:** Caso o aluno ainda tenha dúvidas, ele aguardará um novo feedback.
 - **Não:** O aluno confirma que a dúvida foi resolvida.
13. **Fim:** Representa a conclusão do processo, seja com a dúvida resolvida ou com a necessidade de um novo feedback.

Imagem 10: Atividades do Aluno



Fonte: Autor

O diagrama apresentado na Imagem 10 descreve o processo de correção de scripts submetidos pelo aluno no sistema, detalhando o fluxo de atividades desde o envio do código até a validação das correções. Esse processo ilustra o comportamento do sistema e as interações do aluno durante a análise e validação:

1. **Início:** Representa o início da atividade, marcando o ponto de partida do processo de correção.
2. **Sistema recebe o script:** O aluno submete o script, e o sistema o recebe para análise.
3. **Analisar script:** O sistema processa o script submetido, identificando erros e aplicando correções.
4. **Visualizar resposta:** O sistema exibe a correção realizada, incluindo explicações sobre os erros encontrados.
5. **Validação da correção:** O aluno avalia se aceita ou não as correções fornecidas pelo sistema.
 - **Sim:** Caso o aluno aceite as correções, ele submete a versão final, e o script corrigido é considerado um sucesso.
 - **Não:** Caso o aluno não aceite as correções, o sistema reformula a explicação e apresenta uma nova versão para o aluno.
6. **Fim:** Representa a conclusão do processo, seja com o script corrigido com sucesso ou com o aluno revisitando as explicações fornecidas.

Imagem 11:Atividades entre Sistemas

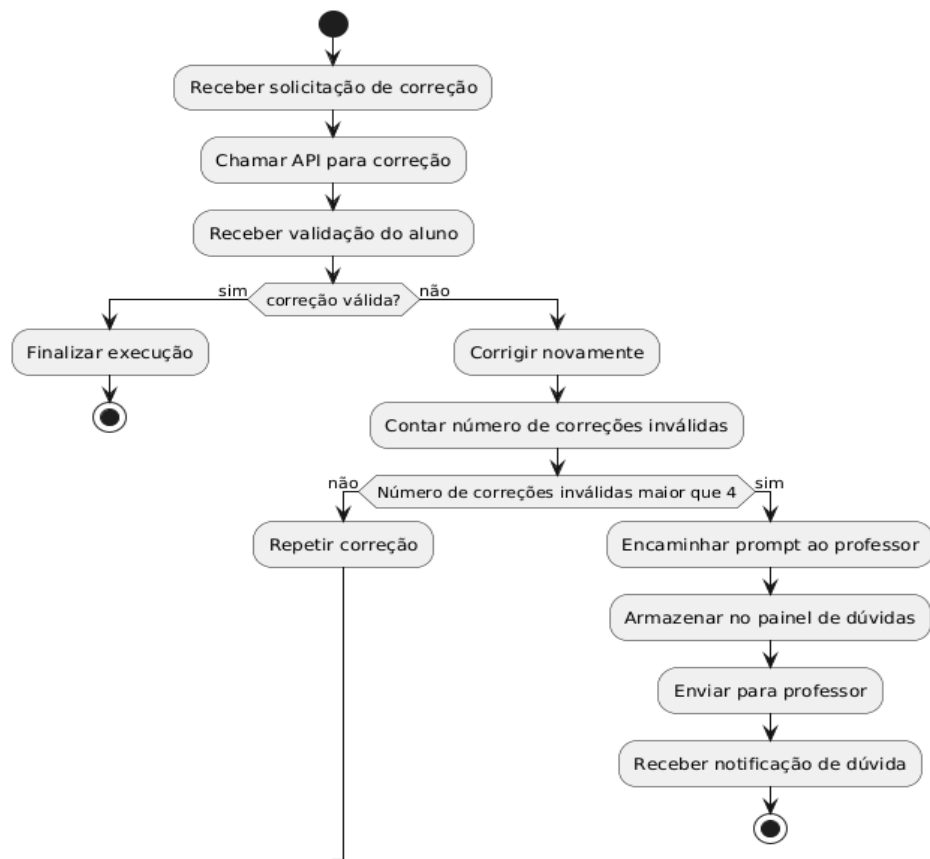


Fonte: Autor

O diagrama apresentado na Imagem 11 ilustra a interação entre o sistema interno, a API de correção e o usuário aluno. Ele detalha o fluxo de atividades que ocorre desde a submissão do script até a exibição dos resultados ao aluno, destacando como os sistemas trabalham de forma integrada para realizar a correção:

1. **Submeter código:** O aluno submete o script a ser corrigido por meio da interface do sistema.
2. **Analisar código:** A API processa o código enviado, identifica e corrige os erros encontrados.
3. **Receber resposta:** O sistema interno recebe os resultados da análise e correção realizados pela API.
4. **Exibir resposta:** O sistema interno apresenta ao usuário as alterações realizadas no script.
5. **Visualizar resposta:** No painel do aluno, ele visualiza as correções feitas, com detalhes sobre os erros corrigidos e melhorias aplicadas.

Imagem 12: Atividades de Correção



Fonte: Autor

O processo de interação entre o aluno, o sistema e a API é ilustrado na Imagem 12, detalhando o fluxo desde a submissão do código até a validação ou o encaminhamento de dúvidas ao professor:

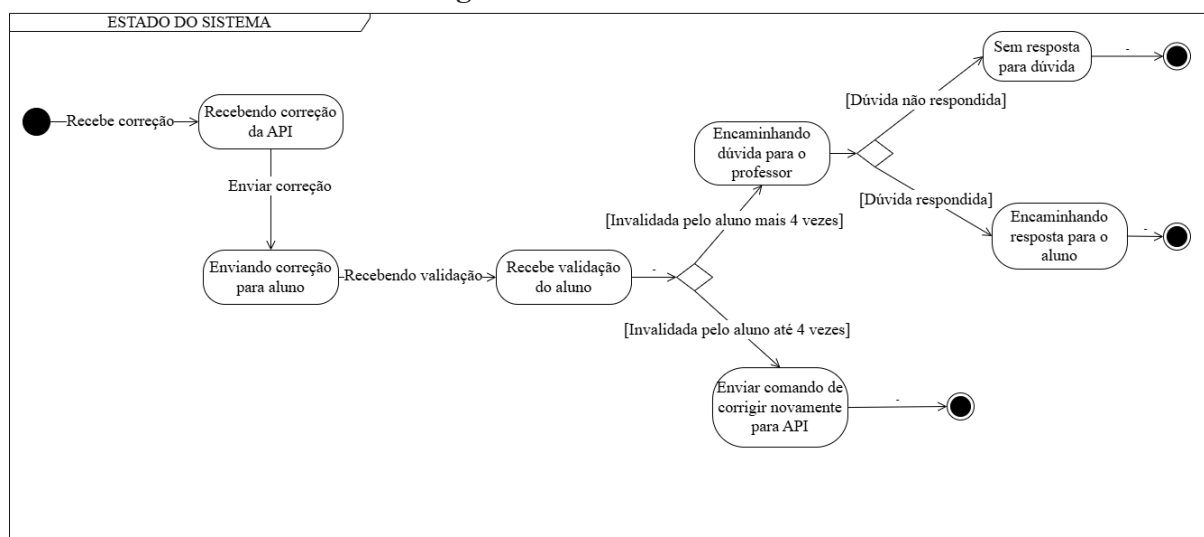
1. **Início:** A atividade é iniciada.
2. **Receber solicitação de correção:** O script é submetido pelo aluno, e o sistema recebe a solicitação para análise.
3. **Chamar API para correção:** O sistema utiliza a API para analisar e corrigir o script submetido.
4. **Receber validação do aluno:** Após visualizar a correção, o aluno escolhe entre as opções fornecidas pelo sistema.
5. **Correções válidas?:** O aluno decide se aceita ou não as correções realizadas.
 - **Sim:** Se as correções forem aceitas, o aluno submete a versão final, e o script corrigido é considerado um sucesso.
 - **Não:** Se as correções não forem aceitas, o sistema reformula a explicação para o aluno.
6. **Contar número de correções inválidas:** O sistema realiza a contagem de correções consideradas inválidas com base nos feedbacks do aluno, até um limite de quatro tentativas.

7. **Número de correções inválidas maior que 4?:** O sistema verifica o total de tentativas inválidas para definir o próximo passo.
 - **Sim:** O sistema encaminha o prompt ao professor e armazena a dúvida no painel de dúvidas.
 - **Não:** O sistema reformula a explicação novamente para o aluno.
8. **Fim:** O processo é concluído.

4.4 Diagrama de Estado

O diagrama de estado é uma representação visual que descreve os diferentes estados pelos quais um objeto ou componente do sistema pode passar durante o seu ciclo de vida. Ele é utilizado para modelar as transições de estado com base em eventos ou condições específicas.

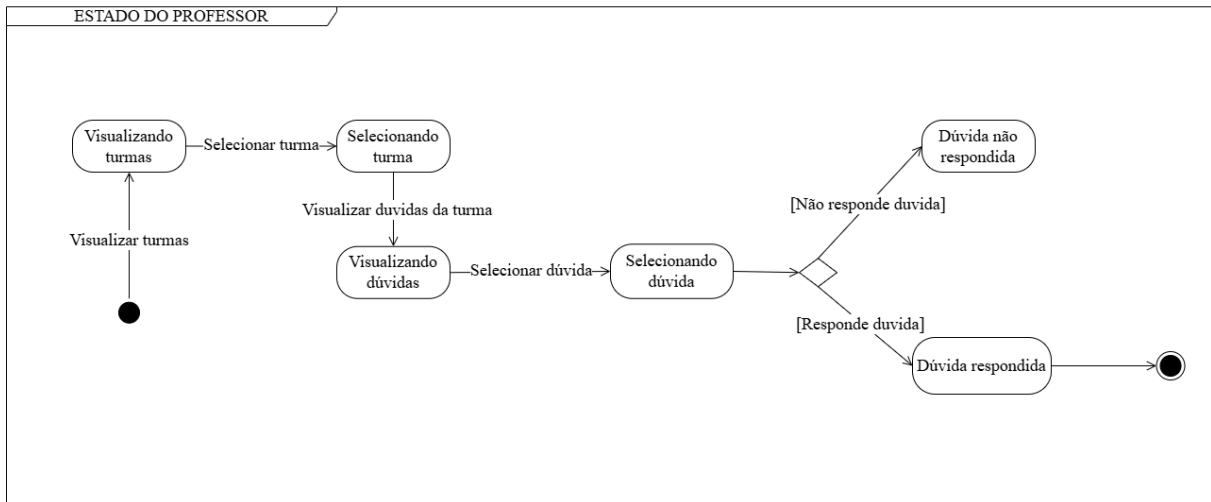
Imagem 13: Estado do Sistemas



Fonte: Autor

O diagrama apresentado na Imagem 13 ilustra os estados do objeto sistema a partir do momento em que ele recebe a correção realizada pela API. O fluxo inclui a exibição da correção na interface do aluno, a validação feita pelo aluno e uma condição que verifica se, após quatro tentativas, o aluno ainda não compreendeu a explicação. Caso essa condição seja atendida, o sistema encaminha o primeiro prompt para o painel de dúvidas do professor. No painel, o professor pode responder à dúvida e, se necessário, anexar materiais de apoio para auxiliar o aluno.

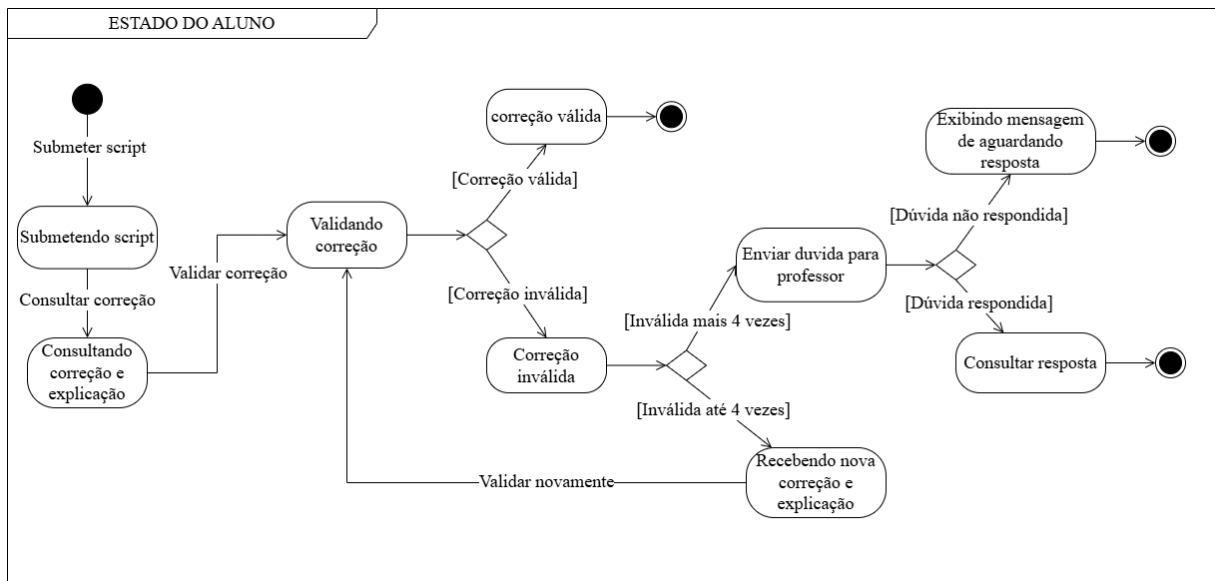
Imagem 14: Estado do Professor



Fonte: Autor

O diagrama apresentado na Imagem 14 representa os estados das atividades que podem ser realizadas pelo perfil do professor. Ele abrange desde o momento em que o professor utiliza o filtro por turmas, no painel de dúvidas, até a etapa em que seleciona a dúvida específica que deseja responder.

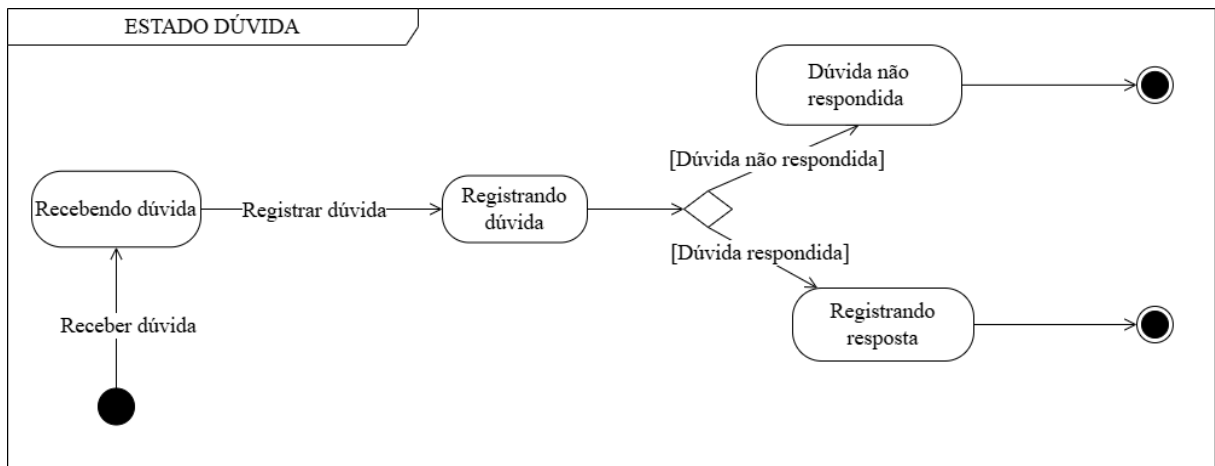
Imagem 15: Estado do Aluno



Fonte: Autor

O diagrama apresentado na Imagem 15 representa os estados das atividades realizadas pelo aluno, iniciando no momento em que ele submete seu código para correção. O fluxo segue até a etapa em que o sistema avalia, por meio de uma condicional, se o aluno compreendeu a explicação fornecida. Caso contrário, o sistema encaminha a dúvida para o professor ou oferece uma nova tentativa de explicação.

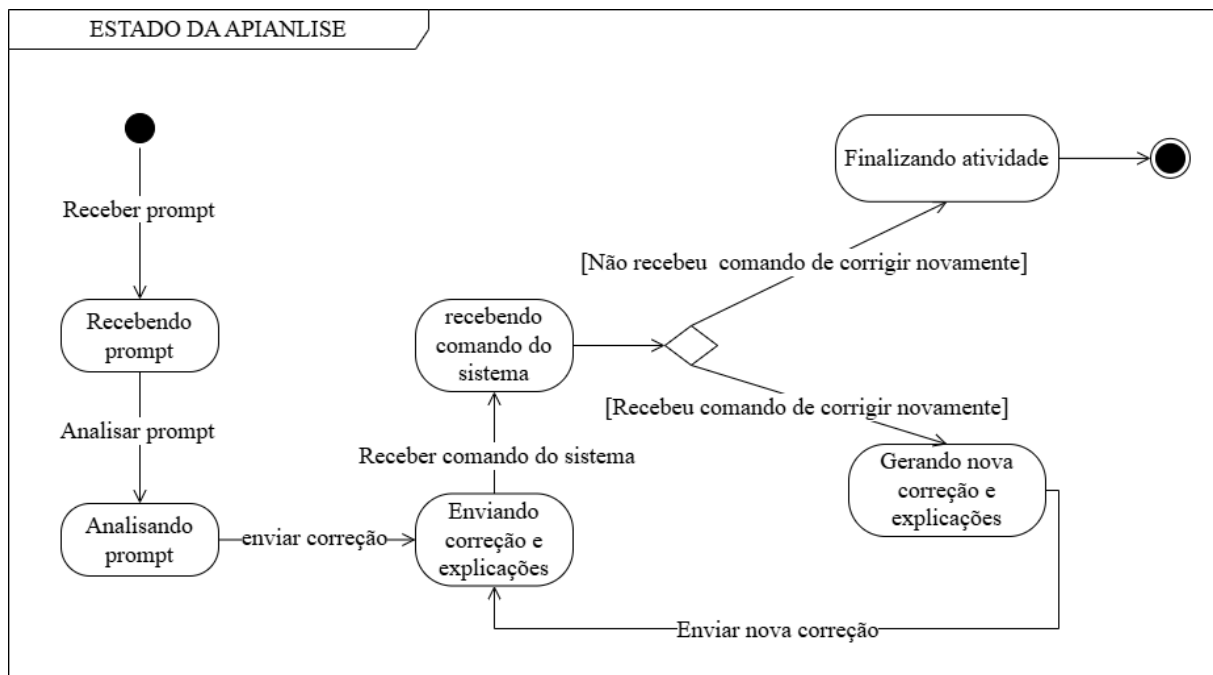
Imagem 16: Estado da Dúvida



Fonte: Autor

O diagrama apresentado na Imagem 16 representa os estados do objeto "dúvida", abrangendo desde o momento em que ela é recebida pelo sistema até as etapas em que pode ser registrada, respondida ou permanecer sem resposta.

Imagem 17: Estado da API



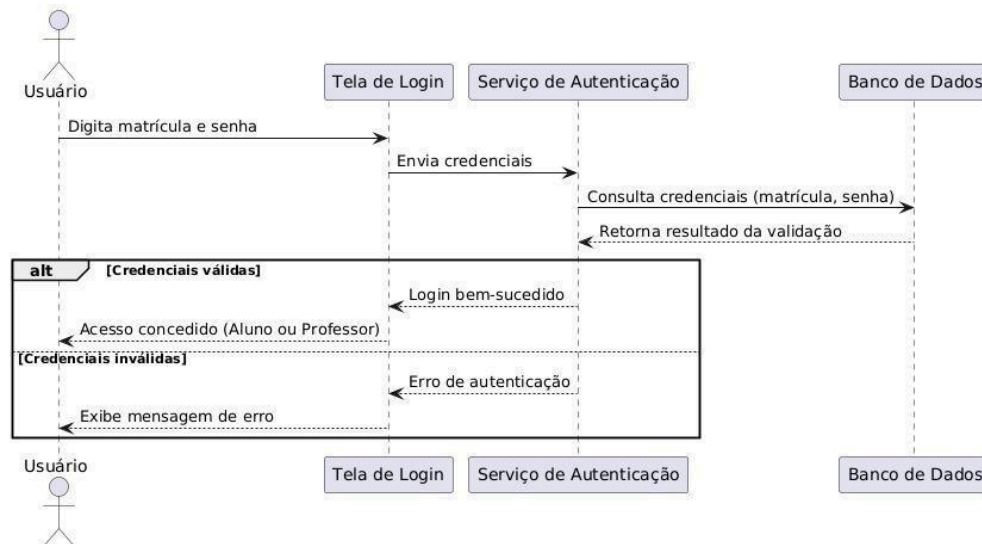
Fonte: Autor

O diagrama apresentado na Imagem 17 ilustra os estados das atividades realizadas pela API, que conecta o sistema interno à IA Generativa. Ele abrange desde o momento em que a API recebe o prompt enviado pelo sistema, analisa e processa uma possível resposta, envia o resultado de volta ao sistema e aguarda o próximo comando. Esse próximo comando depende de uma condicional que verifica se o aluno compreendeu ou não a correção fornecida.

4.5 Diagrama de Sequência

O diagrama de sequência é uma representação visual que ilustra a interação entre os objetos do sistema ao longo do tempo, destacando a troca de mensagens e a ordem em que ocorrem.

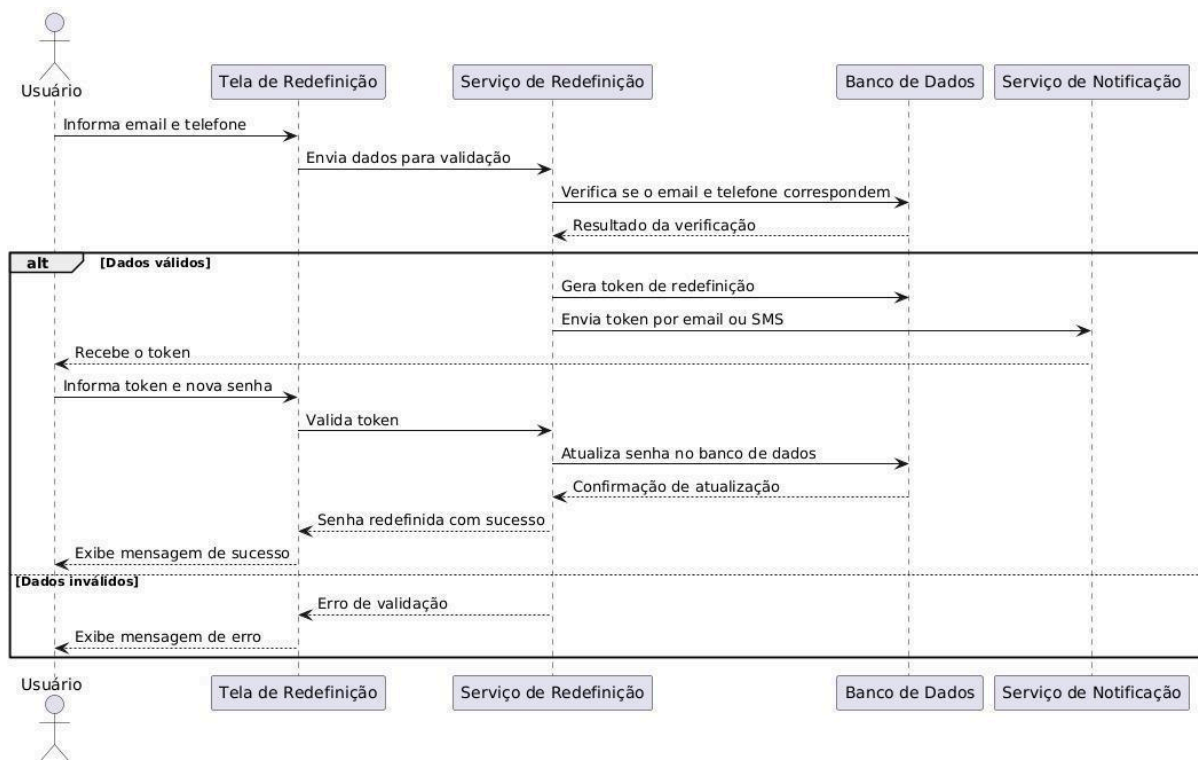
Imagem 18: Sequência da Atividade de Login



Fonte: Autor

O diagrama da imagem 18 ilustra o fluxo da atividade de login para usuários que já possuem cadastro.

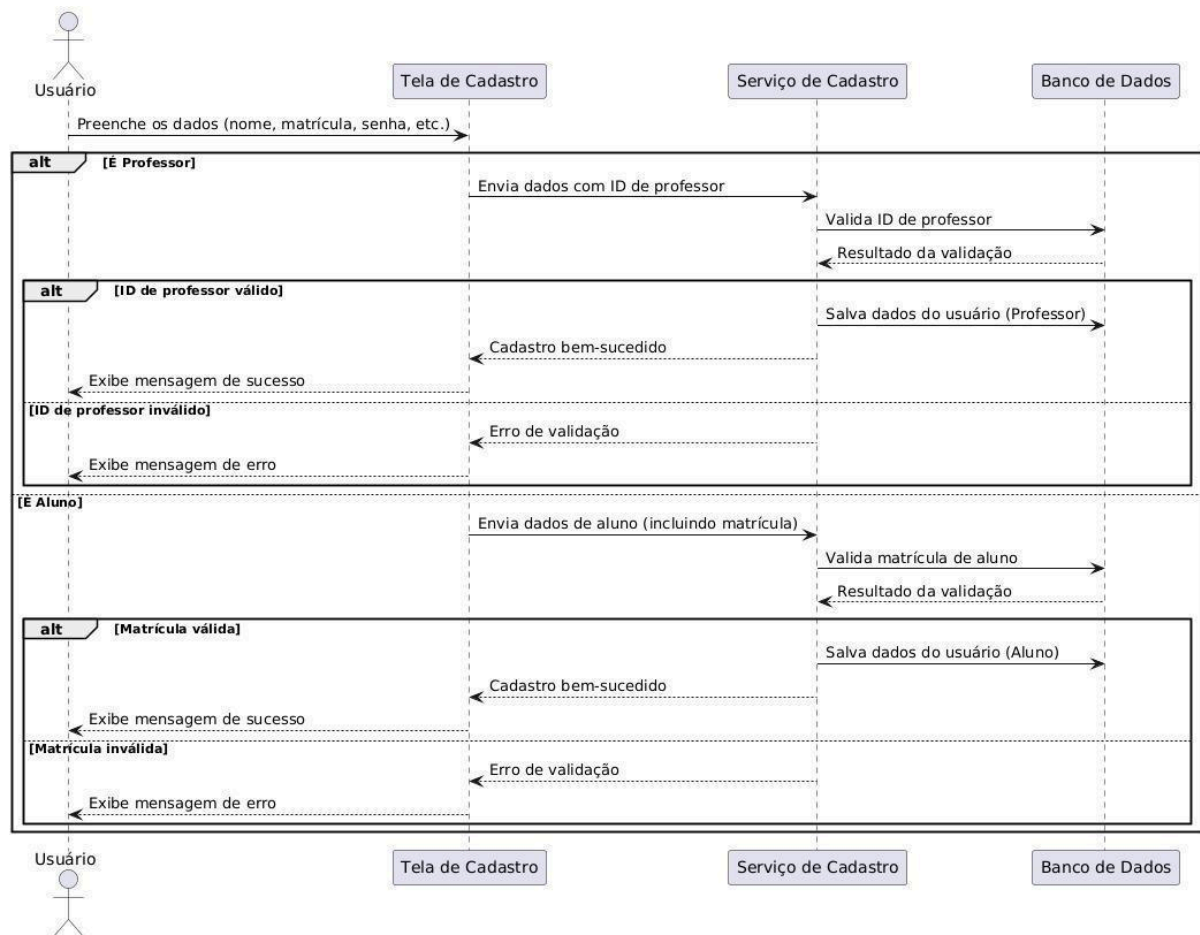
Imagem 19: Sequência da atividade de Redefinição de senha



Fonte: Autor

O diagrama da imagem 19 ilustra a sequência da atividade de redefinição de senha para usuários já cadastrados, destacando também o processo de validação dos dados do usuário.

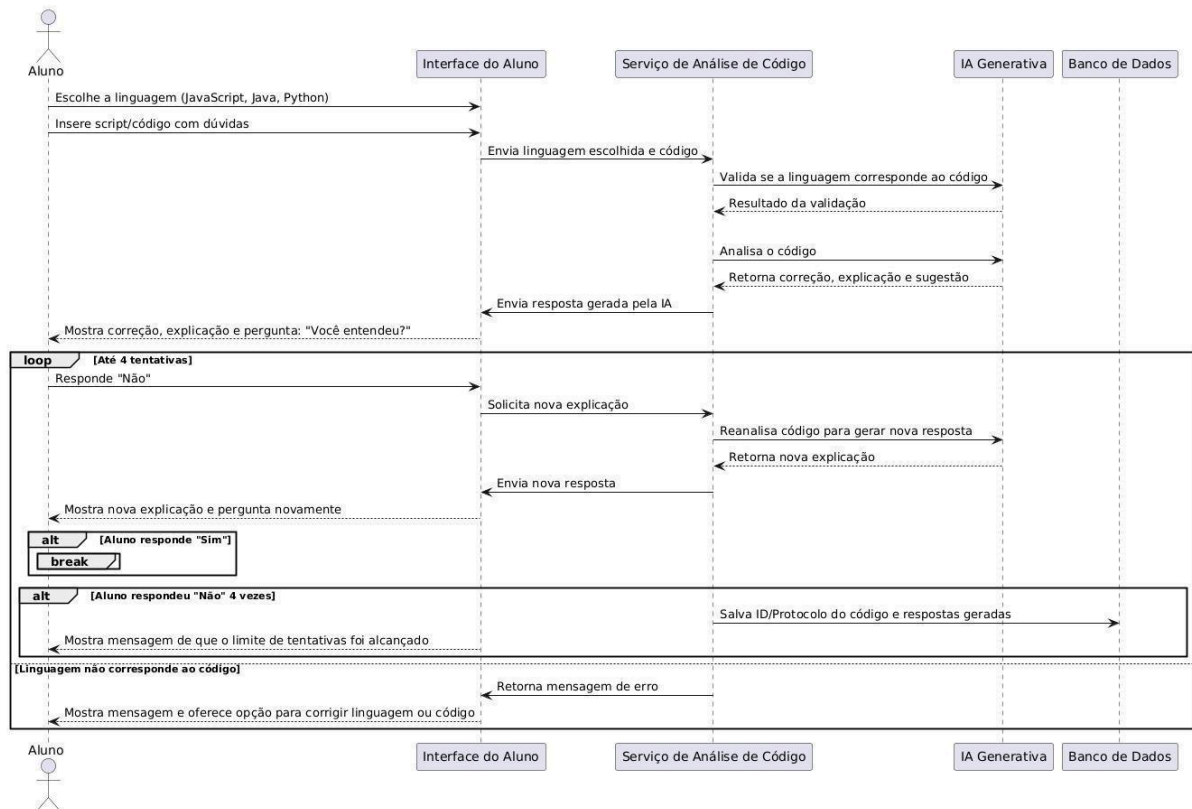
Imagem 20: Sequência da Atividade de Cadastro



Fonte: Autor

O diagrama da imagem 20 ilustra o fluxo da atividade de cadastro para novos usuários, explicando o processo de validação de credenciais e a diferenciação de perfis, seja aluno ou professor.

Imagem 21: Sequência das Atividades dos Alunos



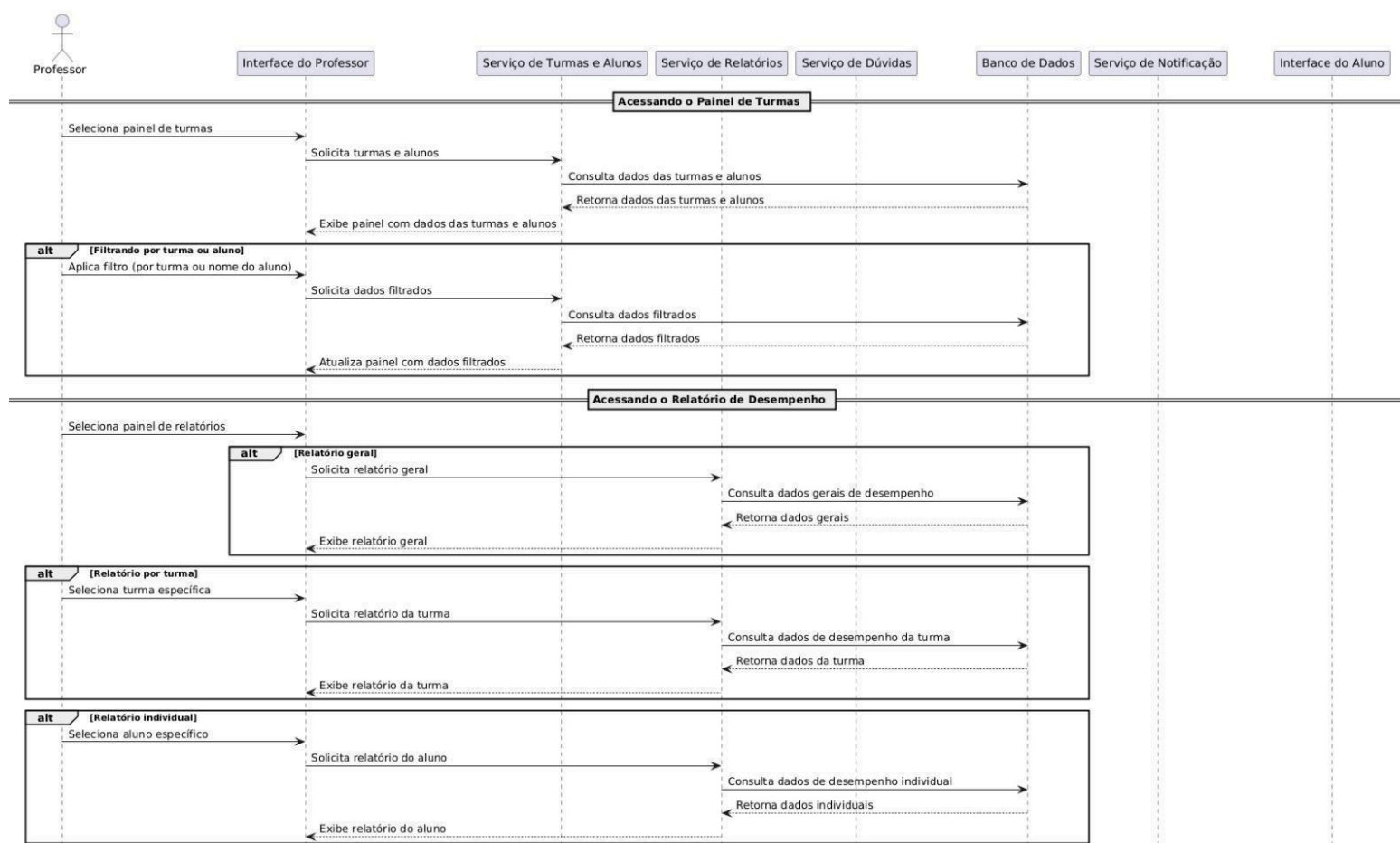
Fonte: Autor

O diagrama ilustra as interações entre o aluno, a interface do sistema e a API. Ele representa o fluxo de atividades desde o momento em que o aluno insere o código para correção até a etapa de validação da explicação.

O processo inicia com o envio do código pelo aluno, que é analisado pela IA Generativa. A correção e a explicação são retornadas ao sistema e exibidas ao aluno, acompanhado da pergunta: "Você entendeu?". Caso o aluno responda "Não", o código é reanalisado para fornecer uma nova explicação. Esse ciclo se repete até o aluno confirmar que entendeu ou até que o limite de quatro respostas negativas seja atingido.

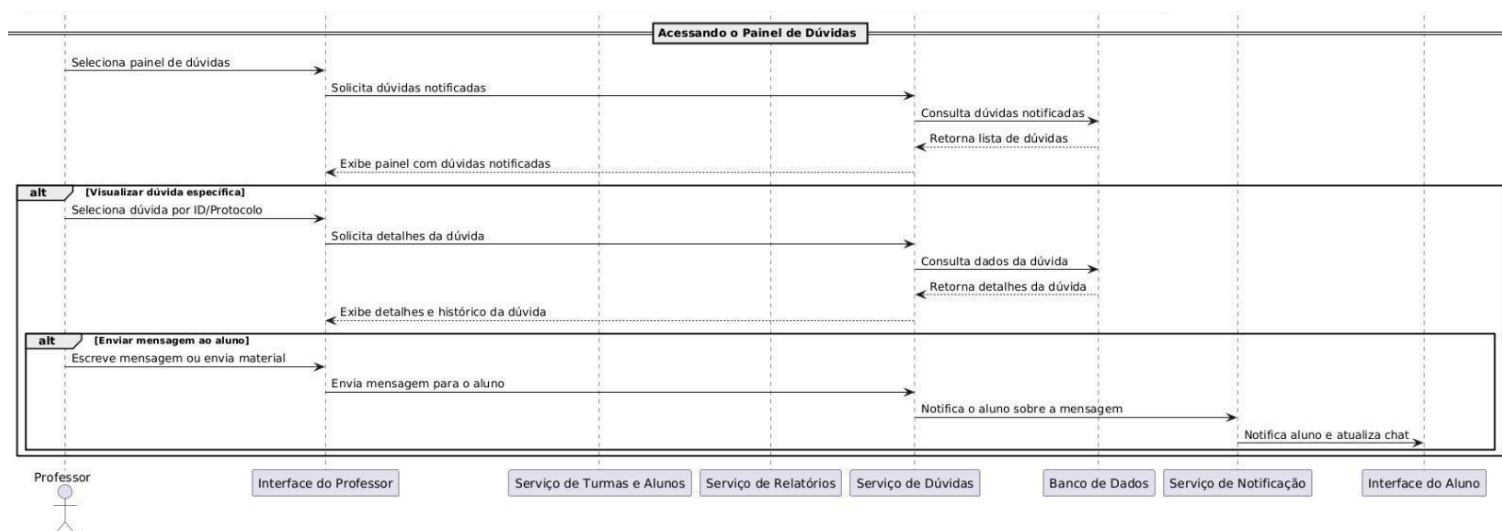
Se o aluno atingir o limite, o sistema gera e salva um ID/protocolo relacionado à dúvida e direciona a dúvida ao painel de dúvidas do professor.

Imagem 22: Sequência das atividades no Painel de Turmas



Fonte: Autor

Imagem 23: Sequência das atividades no Painel de Dúvidas



Fonte: Autor

Os diagramas das imagens 22 e 23 representam as interações realizadas pelo professor com o sistema, detalhando os fluxos de comunicação entre a interface do professor, serviços internos (turmas e alunos, relatórios, dúvidas e notificações) e o banco de dados. Ele ilustra, de forma sequencial, as ações realizadas desde o acesso ao painel de turmas e relatórios, até a navegação pelo painel de dúvidas. Cada etapa descreve como as solicitações feitas pelo professor são processadas pelo sistema, incluindo filtros por turma ou aluno, consulta de relatórios e envio de explicações para os alunos.

@autor: Bianca Milena Da Silva Laranjeira, Jamilly Vitória Ferreira Barbosa, Marcos Vinícius Jansem Oliveira, Paulo Arthur Lima Brito Yhago Luis Maciel De Matos.

@contato:

@data última versão: 28/ 01 / 2025

@versão: 1.1

@outros repositórios:

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

@Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina PROJETO E DESENVOLVIMENTO DE SOFTWARE, sobre a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2024.1, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da licença MIT reproduzida abaixo e, portanto, é compatível com GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Copyright © 2025 Educational Material

Este material está licenciado sob a Licença MIT. É permitido o uso, cópia, modificação, e distribuição deste material para qualquer fim, desde que acompanhado deste aviso de direitos autorais.

O MATERIAL É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO VIOLAÇÃO. EM HIPÓTESE ALGUMA OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM UMA AÇÃO DE CONTRATO, ATO ILÍCITO OU DE OUTRA FORMA, DECORRENTE DE, OU EM CONEXÃO COM O MATERIAL OU O USO OU OUTRAS NEGOCIAÇÕES NO MATERIAL.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>.