



Universidade Federal do Maranhão - UFMA
Centro de Ciências Exatas e Tecnologia - CCET
Coordenação de Engenharia da Computação -EECP

Documentação do projeto

Integrando Modelagem e IA Generativa

FERNANDO DA SILVA COSTA
GABRYELLA CRUZ SOUSA
VITOR DOS SANTOS DE SOUSA
Discentes

THALES LEVI AZEVEDO VALENTE
Docente Responsável

São Luís - MA, 07 de janeiro de 2025



SUMÁRIO

1. Identificação	3
2. Introdução	3
2.1. Objetivos da documentação	3
2.2. Escopo do projeto	4
3. Levantamento de requisitos	5
3.1. O que são requisitos	5
3.2. Requisitos funcionais	5
3.3. Requisitos não funcionais	6
4. Regras de negócio	6
4.1. O que são regras de negócio	6
[RN1.] Login e cadastro	7
[RN2.] Planejamento de projetos	7
[RN3.] Refinamento de requisitos	7
[RN4.] Análise de impacto	7
[RN5.] Exportação de documento	7
5. Diagramas	7
5.1. Diagramas de caso de uso	7
5.1.1. Diagrama de caso de uso - login e cadastro do usuário	8
5.1.2. Diagrama de caso de uso - planejamento de projetos	10
5.1.3. Diagrama de caso de uso - refinamento de requisitos	11
5.1.4. Diagrama de caso de uso - análise de impacto em casos de uso	12
5.1.5. Diagrama de caso de uso - exportação de documentos	14
5.2. Diagrama de classe	15
5.3. Diagramas de atividade	17
5.3.1. Diagrama de atividade - login e cadastro do usuário	18
5.3.2. Diagrama de atividade - planejamento de projetos	19
5.3.3. Diagrama de atividade - refinamento de requisitos	20
5.3.4. Diagrama de atividade - análise de impacto em caso de uso	21
5.3.5. Diagrama de atividade - exportação de documentos	22
5.4. Diagramas de estado	24
5.4.1. Diagrama de estados - Login e Cadastro	24
5.4.2. Diagrama de estados - Planejamento de Projeto	25
5.4.3. Diagrama de estados - Exportação de Documento	26
5.4.4. Diagrama de estados - Refinamento de Requisitos	28
5.4.5. Diagrama de estados - Análise de Impacto	29
5.5. Diagramas de sequência	30
5.5.1. Diagrama de sequência - login e cadastro do usuário	31
5.5.2. Diagrama de sequência - planejamento de projetos	32
5.5.3. Diagrama de sequência - refinamento de requisitos	33



5.5.4. Diagrama de sequência - análise de impacto de caso de uso	34
5.5.5. Diagrama de sequência - exportação de documentos	35
6. Tecnologias	36
7. Escopo da Prototipação	36
7.1 Tela de Login	37
7.2 Tela de Cadastro	37
7.3 Tela Inicial do Software	38
7.4 Perfil do Usuário	39
7.5 Refinamento de Requisitos	39
7.6 Planejamento de Projetos	40
7.7 Análise de Impactos	41
7.8 Exportação de Documentação	42
Conclusão	42
Referências	43

1. Identificação

Título:

Integrando Modelagem e IA Generativa

Nome dos Discentes:

Fernando da Silva Costa - Matrícula: 2019004026

Gabryella Cruz Sousa - Matrícula: 2022012235

Vitor dos Santos de Sousa - Matrícula:2021061852

Docente Responsável:

Thales Levi Azevedo Valente

Disciplina:

Projeto e Desenvolvimento de Software

2. Introdução

2.1. Objetivos da documentação

Este documento tem como propósito apresentar uma visão clara e organizada da etapa inicial do projeto de desenvolvimento de software, com foco no levantamento de requisitos, definição de regras de negócio e elaboração de diagramas UML (Unified Modeling Language). Ele serve como um guia para alinhar as expectativas e promover a comunicação eficaz entre todos os envolvidos no



projeto, garantindo uma compreensão compartilhada sobre o sistema a ser desenvolvido.

No documento, são registrados os requisitos funcionais e não funcionais, bem como as necessidades e expectativas dos usuários finais, proporcionando uma base sólida para o desenvolvimento do sistema. Além disso, ele detalha as regras de negócio que devem ser respeitadas e utiliza diagramas UML, como casos de uso, classes, atividades, estado e sequência, para representar visualmente tanto a estrutura quanto o comportamento do sistema.

Essa abordagem facilita a prevenção de erros e mal-entendidos, ao mesmo tempo em que apoia a manutenção e futuras evoluções do sistema. O documento também contribui para o planejamento e controle do projeto, possibilitando estimativas mais precisas de esforços, prazos e recursos necessários para a execução das atividades nas diferentes fases do desenvolvimento.

2.2. Escopo do projeto

O escopo deste projeto envolve o desenvolvimento de um sistema web que integra tecnologias de inteligência artificial para transformar e otimizar o processo de modelagem de software. Este sistema será uma ferramenta colaborativa e inteligente, projetada para ajudar analistas, desenvolvedores e outros envolvidos a criar modelos de software de forma mais simples, rápida e eficiente.

O sistema oferecerá login seguro. Entre as principais inovações, está o planejamento de projetos, permitindo a geração automática de diagramas de caso de uso a partir de descrições textuais. A IA interpretará a linguagem natural e converterá requisitos em elementos visuais ajustáveis manualmente.

A ferramenta de refinamento de requisitos analisará ambiguidades e lacunas, oferecendo sugestões inteligentes de requisitos. Também será possível gerar documentação técnica personalizada em PDF, Word e XML.

Outra funcionalidade essencial é a análise de impacto em casos de uso. Ao inserir novos casos de uso ou alterações, o sistema avaliará automaticamente os impactos nas estruturas existentes, gerando relatórios detalhados e sugerindo ajustes para minimizar problemas.

Com essas funcionalidades, o sistema reduzirá o tempo de criação de modelos, melhora a clareza dos requisitos e facilitará a colaboração entre equipes. A identificação de problemas garantirão soluções mais rápidas e eficazes.

3. Levantamento de requisitos

3.1. O que são requisitos

“Requisitos de software são especificações que definem as funcionalidades e restrições de um software.” Os requisitos devem satisfazer as necessidades ou expectativas de seus usuários, clientes ou partes interessadas. Eles são essenciais para o planejamento e desenvolvimento de soluções, garantindo que o resultado final atenda aos objetivos estabelecidos. São especificações detalhadas das funcionalidades e características que um sistema deve possuir para atender às necessidades dos usuários e aos objetivos do negócio, esses requisitos são divididos entre requisitos funcionais e não funcionais.

3.2. Requisitos funcionais

Os requisitos funcionais do sistema incluem:

- Login e Cadastro do Usuário:
 - O sistema deve permitir cadastro via email e senha;
 - O sistema deve permitir login seguro;
 - O sistema deve oferecer a função de manter-se conectado ao software.
- Refinamento de Requisitos:
 - O usuário pode solicitar o refinamento de requisitos inserindo os requisitos textuais;
 - O sistema analisa os requisitos verificando ambiguidades;
 - O sistema apresenta a opção de gerar sugestões de requisitos;
 - O software mostra a opção de ajustar os requisitos enviados.
- Planejamento de Projetos:
 - O usuário fornece os requisitos recebidos em reunião com o cliente;
 - A ferramenta deve gerar diagramas de caso de uso automaticamente a partir de descrições textuais fornecidas pelos usuários;

-
- Os diagramas gerados podem ser ajustáveis manualmente.
 - **Análise de Impacto :**
 - O software recebe os casos de uso de um projeto;
 - O software também recebe as alterações de casos de uso do mesmo projeto;
 - O sistema deve avaliar os impactos de novos casos de uso.;
 - O software pode gerar relatórios mostrando o impacto da inclusão de novos casos de uso.
 - **Exportação de Documento:**
 - O usuário pode solicitar a geração de documentos;
 - Deve ser possível gerar documentos em formatos PDF, Word e XML a partir dos modelos criados.

3.3. Requisitos não funcionais

Os requisitos não funcionais do nosso software são:

- **Desempenho:** O sistema deve ser capaz de processar entradas textuais e gerar diagramas de forma rápida e eficiente;
- **Usabilidade:** A interface deve ser intuitiva, para garantir uma experiência de uso agradável para o usuário;
- **Segurança:** Deve oferecer proteção dos dados e prevenção para garantir a integridade das informações armazenadas no sistema;
- **Manutenibilidade:** O código do sistema deve seguir boas práticas de desenvolvimento, incluindo documentação clara, testes automatizados e separação de responsabilidades.

4. Regras de negócio

4.1. O que são regras de negócio

Segundo Dallavalle e Cazarini regras de negócio são “Uma categoria de requisitos do sistema que representam decisões sobre como executar o negócio.” com isso podemos afirmar que as regras de negócio são as diretrizes e critérios que definem como um sistema ou aplicação deve se comportar para atender às necessidades específicas do negócio ou organização que está sendo atendida.



[RN1.] Login e cadastro

- O sistema deve autenticar usuários exclusivamente por e-mail e senha;
- O cadastro de usuários deve incluir validação de e-mail para ativação da conta.

[RN2.] Planejamento de projetos

- O sistema deve gerar automaticamente diagramas de caso de uso com base em descrições textuais fornecidas;
- Diagramas gerados devem ser editáveis de forma manual pelos usuários.

[RN3.] Refinamento de requisitos

- O sistema deve identificar ambiguidades e lacunas em requisitos textuais;

[RN4.] Análise de impacto

- O sistema deve avaliar o impacto de alterações ou novos casos de uso nos modelos existentes;
- Relatórios de impacto gerados devem incluir detalhes sobre os fluxos e elementos afetados.

[RN5.] Exportação de documento

- A exportação deve incluir opções de formatos PDF, Word e XML;
- Os documentos gerados devem conter os diagramas, fluxos e descrições detalhadas dos casos de uso.

5. Diagramas

5.1. Diagramas de caso de uso

Na UML (Linguagem de Modelagem Unificada), o diagrama de caso de uso é uma representação visual que descreve como os usuários (atores) interagem com um sistema para alcançar determinados objetivos. Ele é uma ferramenta poderosa para capturar os requisitos funcionais de um sistema de maneira simples e compreensível, sendo amplamente utilizado na análise de sistemas e no levantamento de requisitos.

Os diagramas de caso de uso são compostos por três elementos principais: os atores, que representam os usuários ou sistemas externos que interagem com o sistema; os casos de uso, que descrevem as funcionalidades específicas oferecidas

pelo sistema aos atores; e as relações entre atores e casos de uso, que demonstram como os atores utilizam essas funcionalidades para alcançar seus objetivos. Essa representação clara e objetiva das interações entre usuários e sistema é essencial para compreender e comunicar os requisitos do sistema durante o processo de desenvolvimento de software.

Outro tópico que se faz necessário destacar são os fluxos principais e alternativos que na UML, desempenham um papel crucial na complementação dos casos de uso, detalhando os diferentes caminhos que uma interação pode seguir. O fluxo principal representa o cenário ideal ou padrão, descrevendo a sequência de passos necessários para atingir o objetivo principal do caso de uso. Já os fluxos alternativos mapeiam variações ou exceções ao fluxo principal, como condições de erro, caminhos opcionais ou eventos inesperados. Essa distinção é fundamental para capturar a complexidade real do sistema, garantindo que ele atenda tanto aos cenários mais comuns quanto às situações especiais, aumentando a robustez e a previsibilidade do software.

5.1.1. Diagrama de caso de uso - login e cadastro do usuário

O caso de uso login e cadastro de usuário descreve a interação entre o usuário e o sistema para autenticar sua identidade, garantindo acesso seguro às funcionalidades disponíveis.

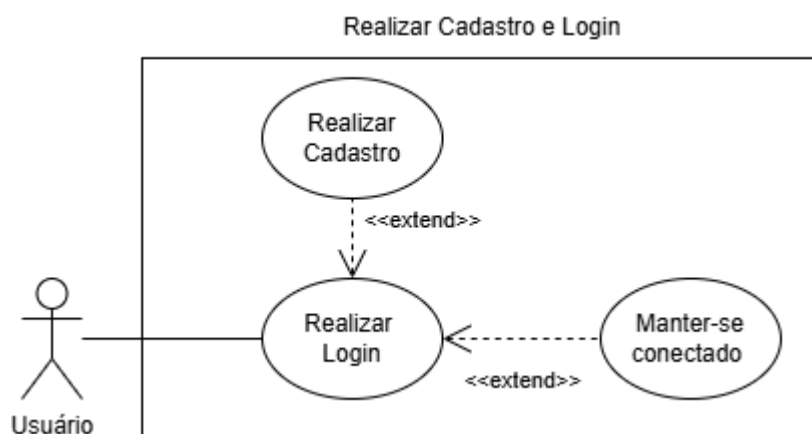


Figura 1: Diagrama de caso de uso, login e cadastro de usuário.



Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none">1. O usuário acessa a interface de cadastro;2. O usuário insere email e senha;3. Se as credenciais forem válidas, o sistema autentica o usuário e concede acesso ao sistema;4. O sistema exibe a tela de login.	<ol style="list-style-type: none">2a. O sistema detecta que e-mail não é válido;2b. Exibe uma mensagem de erro ao usuário;2c. O caso de uso retorna ao passo 2.

Tabela 1: Fluxo principal e alternativo do caso de uso cadastro.

Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none">5. O usuário acessa a interface de login;6. O usuário insere suas credenciais (e-mail e senha);7. O sistema valida as credenciais fornecidas:<ul style="list-style-type: none">• Verifica se o e-mail está registrado;• Confirma se a senha corresponde ao registro no banco de dados.8. Se as credenciais forem válidas, o sistema autentica o usuário e concede acesso ao sistema;9. O sistema exibe a tela inicial.	<ol style="list-style-type: none">2a. O sistema detecta que e-mail não está registrado ou que a senha está incorreta;2b. Exibe uma mensagem de erro ao usuário;2c. O caso de uso retorna ao passo 2.

Tabela 2: Fluxo principal e alternativo do caso de uso login.

5.1.2. Diagrama de caso de uso - planejamento de projetos

O software poderá gerar automaticamente diagramas de caso de uso a partir de uma descrição textual de requisitos fornecidos pela equipe de produto.

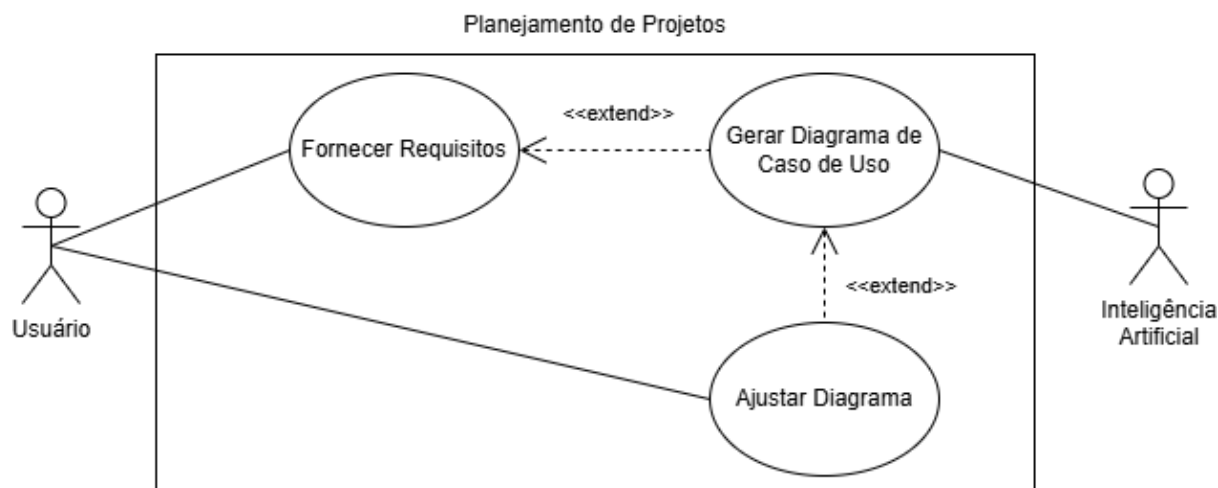


Figura 2: Diagrama de caso de uso, planejamento de projetos.

Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> 1. O usuário acessa a interface do sistema; 2. O usuário insere uma descrição textual dos requisitos do projeto; 3. O sistema processa as informações fornecidas e utiliza IA generativa para interpretar e organizar os dados; 4. O sistema gera automaticamente diagramas caso de uso (representando atores, casos de uso e relacionamentos); 5. O sistema exibe os diagramas 	<ol style="list-style-type: none"> 5a. O sistema permite que o usuário edite diretamente os diagramas ou refine os requisitos textuais; 5b. O sistema processa novamente os dados ajustados; 5c. Novos diagramas são gerados com base nos ajustes realizados; 5d. O fluxo retorna ao passo 5 do fluxo principal.

<p>gerados na interface para revisão;</p> <p>6. O Usuário revisa os diagramas:</p> <ul style="list-style-type: none"> • Caso estejam corretos, o fluxo é encerrado. • Caso sejam necessários ajustes, o fluxo alternativo é acionado. 	
---	--

Tabela 3: Fluxo principal e alternativo do caso de uso automação do planejamento de projetos.

5.1.3. Diagrama de caso de uso - refinamento de requisitos

A IA analisa requisitos ambíguos ou incompletos e sugere melhorias nos requisitos que poderiam atender melhor às necessidades do cliente.

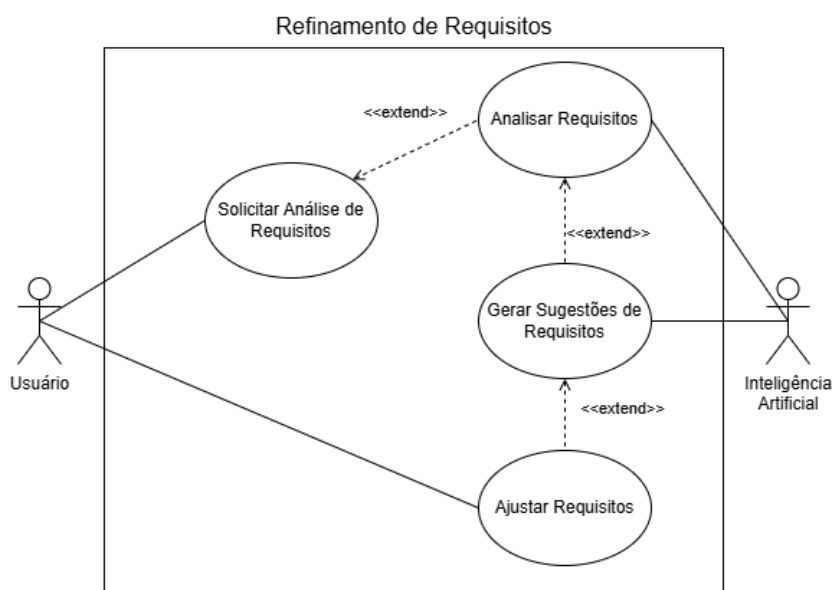


Figura 3: Diagrama de caso de uso, refinamento de requisitos.

Fluxo Principal	Fluxo Alternativo
1. O usuário fornece uma lista de requisitos do projeto no formato especificado	2a. Os requisitos fornecidos contém ambiguidades que não podem ser resolvidas automaticamente;



<p>2. O sistema de IA processa os requisitos fornecidos e identifica ambiguidades ou informações incompletas.</p> <p>3. A IA sugere requisitos específicos baseados nos requisitos analisados, mapeando funcionalidades relevantes.</p> <p>4. O sistema exibe os requisitos sugeridos e suas descrições detalhadas para o usuário.</p> <p>5. O usuário revisa as sugestões, ajusta se necessário e confirma a finalização do processo.</p>	<p>2b. O sistema exibe mensagem de erro e retorna ao passo 1.</p> <p>4a. O usuário não está satisfeito com os casos de uso sugeridos pela IA;</p> <p>4b. O sistema permite que o usuário ajuste os parâmetros de análise ou refine os requisitos fornecidos;</p> <p>4c. Após o ajuste, o sistema reprocessa os requisitos para gerar novas sugestões.</p>
--	---

Tabela 4: Fluxo principal e alternativo do caso de uso assistente de refinamento de requisitos.

5.1.4. Diagrama de caso de uso - análise de impacto em casos de uso

O software pode realizar uma análise de impacto sempre que um novo requisito ou alteração for inserido, gerando uma visão de como isso afetaria os casos de uso existentes e as atividades relacionadas.

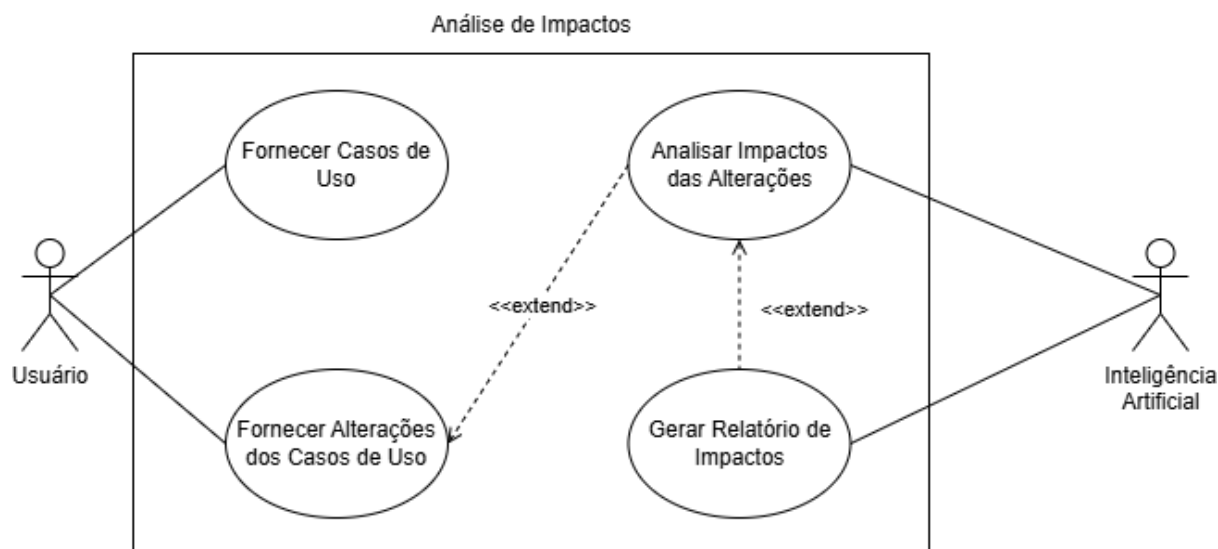


Figura 4: Diagrama de caso de uso, análise de impactos.

Fluxo Principal	Fluxo Alternativo
<ol style="list-style-type: none"> 1. O usuário insere um novo requisito ou solicita alteração em um requisito existente no sistema; 2. O sistema analisa automaticamente os diagramas, requisitos e atividades relacionadas para identificar os elementos que serão impactados; 3. O sistema cria um relatório detalhado com os seguintes dados: <ul style="list-style-type: none"> • Casos de uso e requisitos diretamente afetados. • Dependências indiretas (como outros casos de uso relacionados). 	<ol style="list-style-type: none"> 2a. O novo requisito ou alteração não afeta nenhum caso de uso: <ul style="list-style-type: none"> • O sistema informa ao usuário que não há impacto detectado; • O usuário confirma a ausência de impacto e finaliza a análise; 2b. A alteração gera um impacto muito amplo, afetando múltiplos elementos críticos do sistema: <ul style="list-style-type: none"> • O sistema alerta o usuário sobre a extensão do impacto; • Retorno ao fluxo principal no passo 3. 4a. O usuário não está satisfeito com as soluções sugeridas no relatório:

<ul style="list-style-type: none"> Riscos potenciais associados às mudanças. <p>4. O sistema exibe o relatório para o usuário, destacando as áreas afetadas e sugerindo possíveis soluções ou ajustes;</p> <p>5. O usuário revisa o relatório, ajusta os elementos necessários e aprova ou rejeita a alteração.</p>	<ul style="list-style-type: none"> O sistema permite que o usuário refine os requisitos ou ajuste os parâmetros de análise; Retorno ao fluxo principal no passo 2.
--	--

Tabela 5: Fluxo principal e alternativo do caso de uso análise de impactos.

5.1.5. Diagrama de caso de uso - exportação de documentos

Gera automaticamente documentos (em formato PDF, Word e XML) baseado nos diagramas criados.

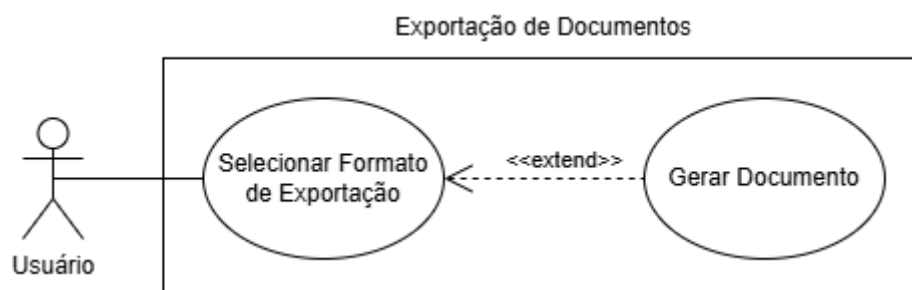


Figura 5: Diagrama de caso de uso, exportação de documentos.

Fluxo Principal	Fluxo Alternativo
<p>1. O usuário acessa a funcionalidade de exportação de documento no sistema;</p> <p>2. O sistema apresenta as opções de formato de exportação: PDF, Word e XML;</p>	<p>2a. O sistema não consegue carregar as opções de exportação devido a um erro de configuração.</p> <p>2b. O sistema notifica o usuário sobre o problema.</p> <p>2c. O processo retorna ao passo 1.</p>



<ol style="list-style-type: none">3. O usuário seleciona o formato desejado (PDF, Word e XML);4. O sistema gera a documento no formato selecionado, baseada nos diagramas e informações disponíveis;5. O sistema apresenta botão para download do arquivo gerado;6. O usuário baixa o arquivo e confirma a finalização do processo.	<p>2d. O processo retorna ao passo 2</p>
--	--

Tabela 6: Fluxo principal e alternativo do caso de uso exportação de documentos.

5.2. Diagrama de classe

Os diagramas de classes são fundamentais na modelagem de sistemas orientados a objetos, servindo como uma representação visual das classes que compõem um sistema, suas propriedades, métodos e as relações entre elas. Eles permitem que desenvolvedores e analistas compreendam a estrutura de um sistema e promovem a comunicação clara entre os membros da equipe. A UML (Unified Modeling Language) é a linguagem padrão para a criação desses diagramas.

No diagrama de classes, cada classe é representada como um retângulo subdividido em três partes: o nome da classe, seus atributos e seus métodos. Os relacionamentos, como associações, heranças e dependências, são representados por diferentes tipos de linhas e conectores. A multiplicidade é um conceito importante nesses diagramas, pois indica quantas instâncias de uma classe podem estar associadas a outra.

Além disso, os diagramas de classes são úteis não apenas na fase de design do sistema, mas também durante o desenvolvimento e a manutenção, ajudando na documentação e no entendimento contínuo da arquitetura do software.

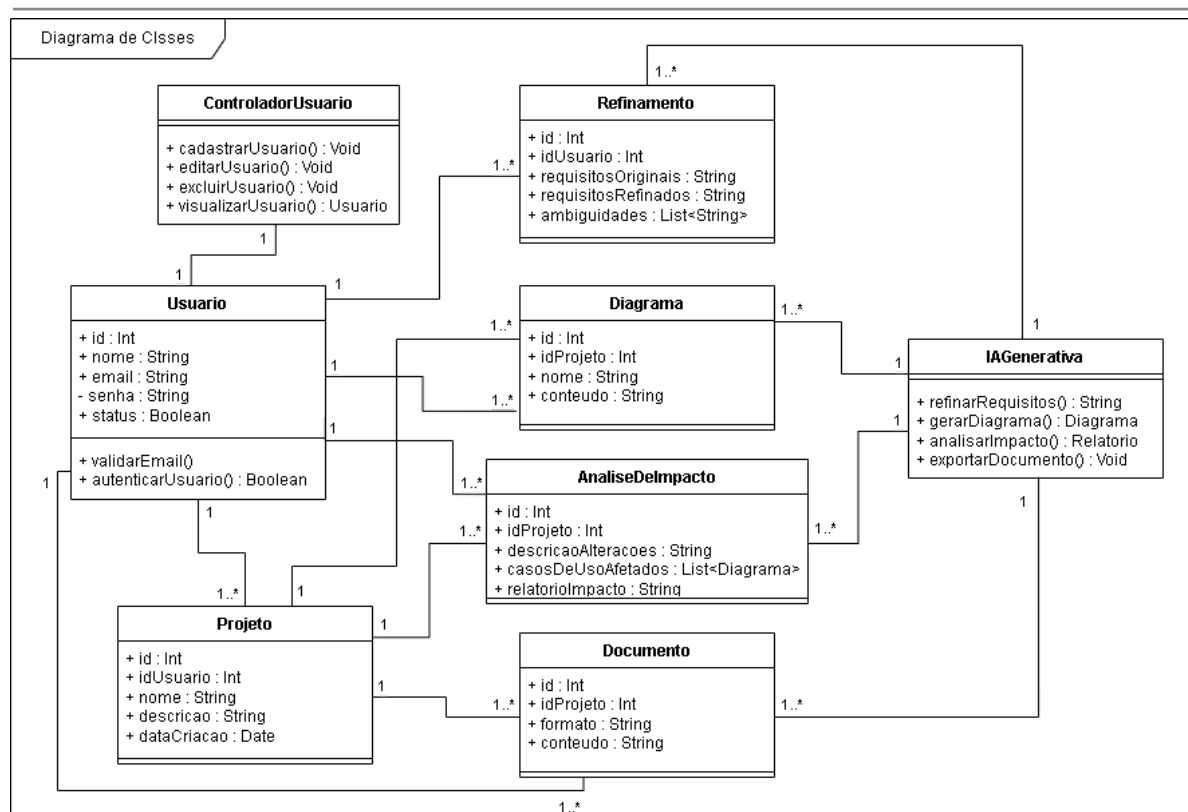


Figura 6: Diagrama de classe do sistema.

Este sistema tem como objetivo principal facilitar a criação e manutenção de diagramas e documentação técnica, além de aprimorar os processos de análise de requisitos e mudanças.

A classe "Usuario" representa os indivíduos que utilizam o software, sendo responsável por gerenciar as informações de identificação, como nome e email, além de manter o controle de acesso aos projetos criados por eles. Essa classe se relaciona diretamente com a classe "Projeto", que centraliza os elementos organizados pelo usuário, incluindo diagramas, documentos, refinamentos e análises de impacto.

A classe "Projeto" é essencial para estruturar o sistema, pois organiza todos os componentes relacionados a um trabalho específico, permitindo que cada usuário tenha controle sobre os diferentes aspectos do projeto. Os diagramas gerados pelo sistema são representados pela classe "Diagrama", que armazena as informações sobre os casos de uso e serve como base para análises e exportações.



Da mesma forma, os documentos gerados pelo sistema são descritos pela classe "Documento", que registra relatórios técnicos e exportações em diversos formatos, como PDF, Word e XML.

A inteligência artificial desempenha um papel central no sistema, representada pela classe "IAGenerativa". Essa classe é responsável por realizar todas as operações complexas, como o refinamento de requisitos, a análise de impacto, a geração de diagramas e a exportação de documentos. A "IAGenerativa" atua como o núcleo de processamento inteligente, recebendo as entradas das outras classes e retornando os resultados de forma automatizada.

Para lidar com os processos específicos de refinamento de requisitos e análise de impacto, o sistema conta com as classes "Refinamento" e "AnaliseDeImpacto". A classe "Refinamento" gerencia o processo de identificar ambiguidades nos requisitos textuais e sugerir melhorias. Já a classe "AnaliseDeImpacto" é responsável por avaliar as alterações feitas nos diagramas ou requisitos, gerando relatórios detalhados sobre os impactos nos elementos afetados.

Os relacionamentos entre as classes foram cuidadosamente definidos para garantir a consistência e a funcionalidade do sistema. Um usuário pode estar associado a múltiplos projetos, e cada projeto pode conter diversos diagramas, documentos, refinamentos e análises de impacto. A classe "IAGenerativa" interage com as demais classes para executar suas operações e fornecer os resultados esperados, garantindo a integração total entre as partes do sistema.

5.3. Diagramas de atividade

O diagrama de atividades é uma ferramenta visual da UML que representa, de forma clara, o passo a passo de um processo ou sistema. Ele ajuda a entender a sequência de tarefas, identificar quem realiza cada etapa e visualizar fluxos complexos. Seus principais elementos incluem atividades (ações representadas por retângulos arredondados), fluxos de controle (setas que indicam a sequência), pontos de decisão (losangos para indicar caminhos alternativos) e swimlanes (faixas que mostram os responsáveis por cada etapa).

Por exemplo, no fluxo de um pedido online: o cliente faz o pedido, o sistema verifica o pagamento e, dependendo da aprovação, o pedido é enviado ou o cliente é notificado. Os benefícios incluem facilitar a comunicação entre equipes, identificar gargalos e otimizar processos, tornando-os mais compreensíveis e eficientes.

5.3.1. Diagrama de atividade - login e cadastro do usuário

O diagrama descreve o processo de autenticação e cadastro. No login, o usuário insere e-mail e senha, o sistema valida as credenciais e, se forem válidas, concede acesso. Caso sejam inválidas, o sistema exibe um erro, orienta o usuário a revisar os dados ou o redireciona para o cadastro. No cadastro, o usuário insere seus dados, o sistema os valida e, se forem corretos, finaliza o registro; caso contrário, o ciclo recomeça.

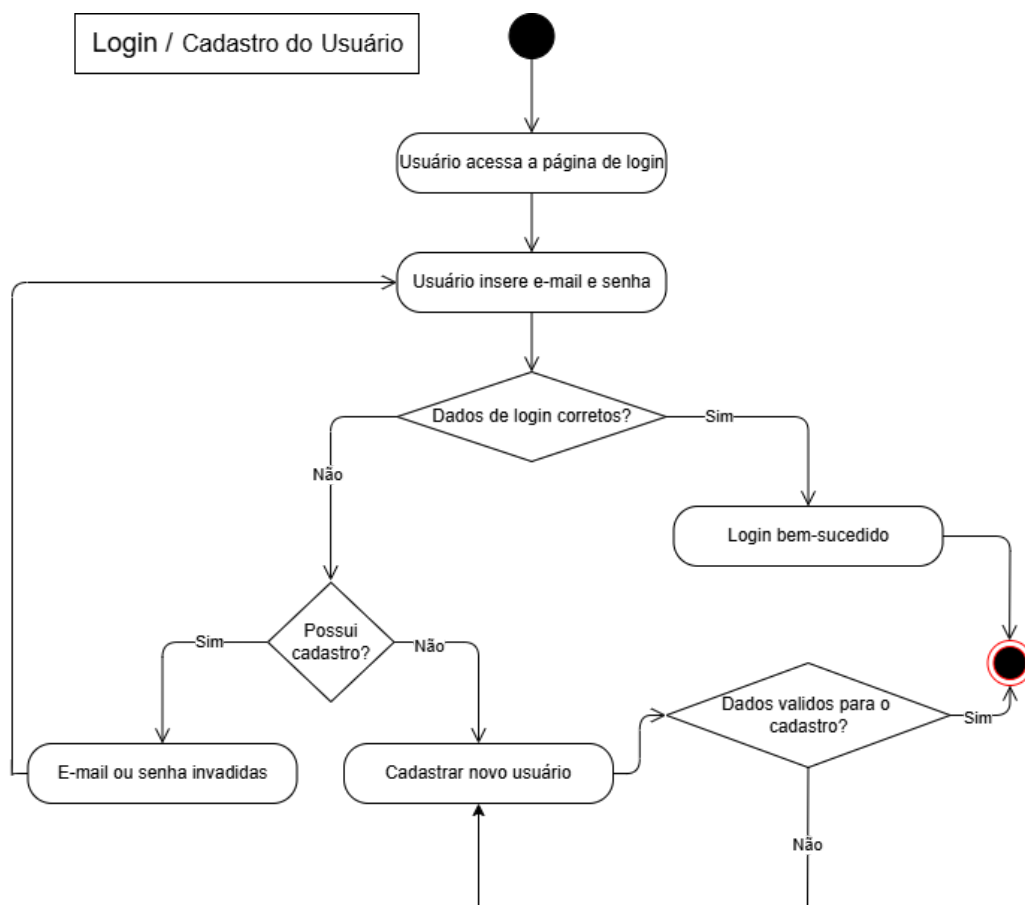


Figura 7: Diagrama de atividade, login e cadastro do usuário.

5.3.2. Diagrama de atividade - planejamento de projetos

O diagrama de atividade representa o processo de geração automatizada de diagramas de caso de uso. O fluxo inicia com o usuário acessando o sistema e inserindo os requisitos do projeto em formato textual. O sistema utiliza IA para processar os dados e gerar automaticamente os diagramas. Em seguida, os diagramas são exibidos para o usuário revisar. Se aprovados, o processo é encerrado; caso contrário, um fluxo alternativo para ajustes é acionado e de revisado e ajustado ele retorna novamente para o sistema de requisitos e assim o ciclo é iniciado novamente.

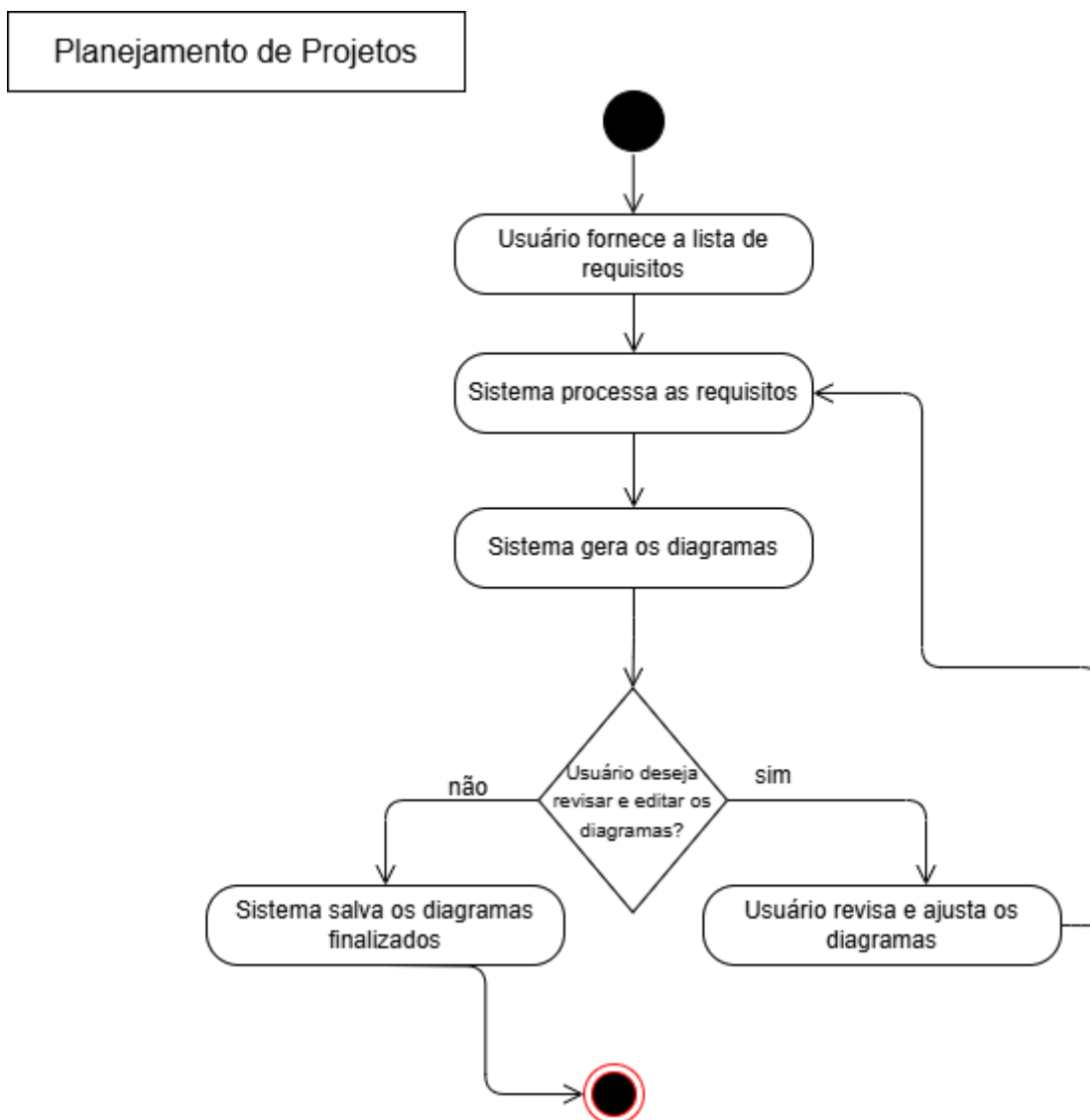


Figura 8: Diagrama de atividade, planejamento de projetos.

5.3.3. Diagrama de atividade - refinamento de requisitos

O usuário fornece os requisitos do projeto, a IA analisa e identifica possíveis ambiguidades, sugere melhorias e funcionalidades relevantes. O sistema exibe as sugestões detalhadas, o usuário revisa e ajusta, se necessário, confirmando a finalização. Se houver ambiguidades não resolvíveis, o sistema exibe um erro e retorna ao início. Caso o usuário não esteja satisfeito com as sugestões, pode ajustar os requisitos ou parâmetros, e o sistema reprocessa para gerar novas sugestões.

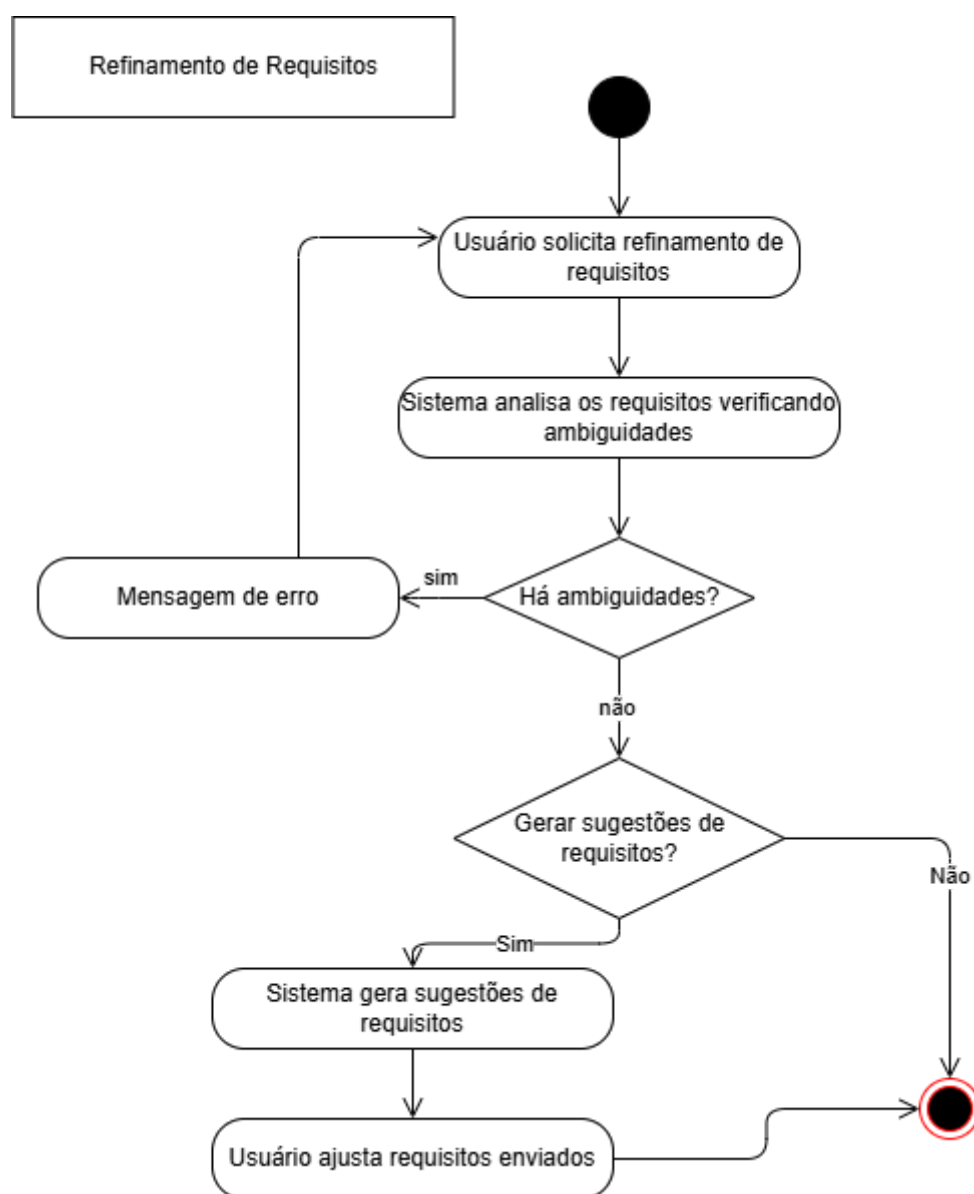




Figura 9: Diagrama de atividade, refinamento de requisitos.

5.3.4. Diagrama de atividade - análise de impacto em caso de uso

O diagrama descreve a gestão de alterações em requisitos. O usuário insere ou modifica um requisito, o sistema analisa impactos nos diagramas e atividades relacionadas, gera um relatório com áreas afetadas, dependências e riscos, e

sugere ajustes.

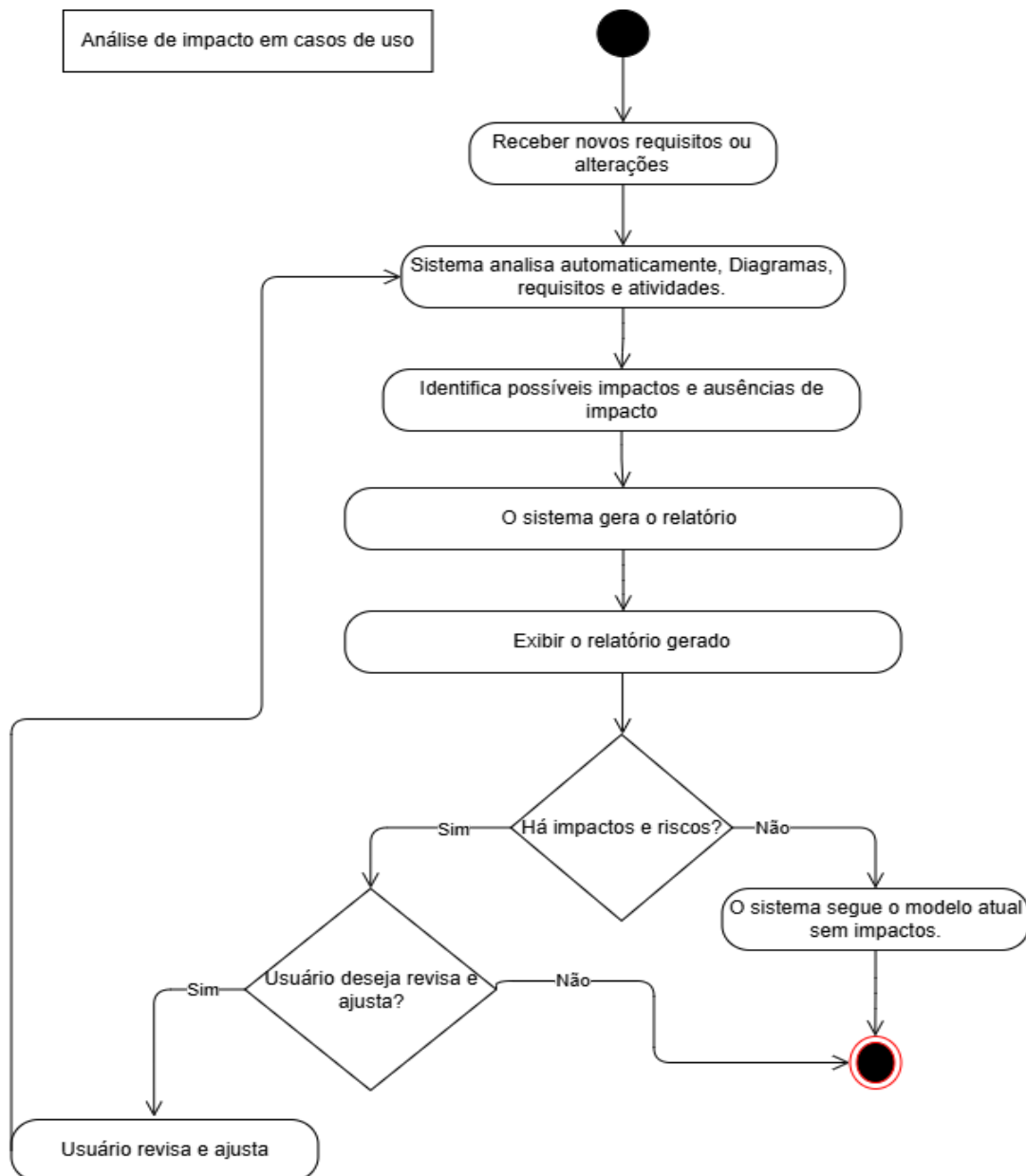


Figura 10: Diagrama de atividade, análise de impacto em caso de uso.

5.3.5. Diagrama de atividade - exportação de documentos

O diagrama descreve o processo de exportação de documentos. O usuário acessa a funcionalidade, escolhe o formato de exportação (PDF, Word ou XML), o

sistema gera o documento com base nas informações disponíveis e disponibiliza o botão de download. O usuário baixa o arquivo e conclui o processo.

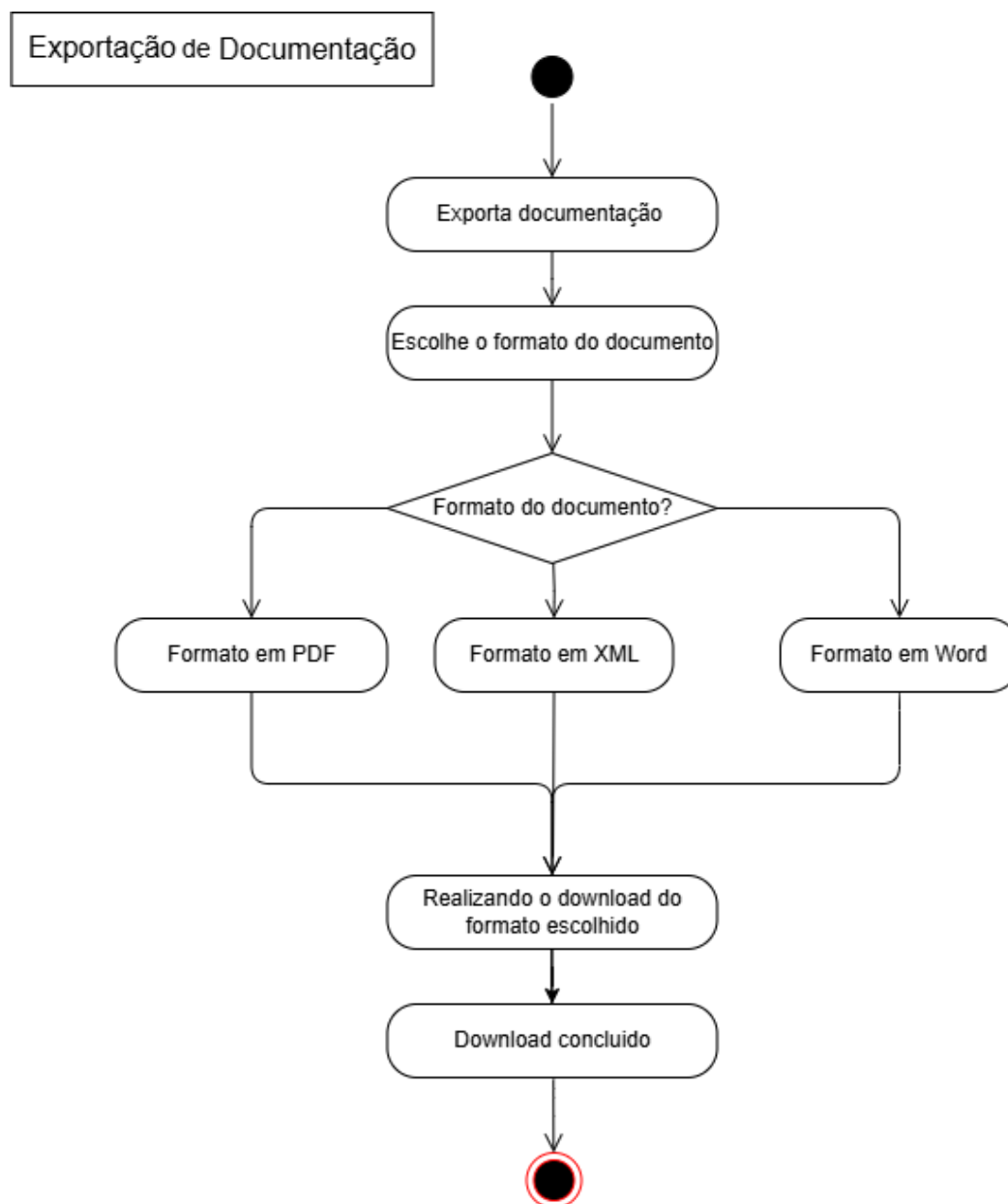


Figura 11: Diagrama de atividade, exportação de documentos.

5.4. Diagramas de estado

Os diagramas de estados são ferramentas fundamentais para modelar o comportamento dinâmico de sistemas. Eles representam de forma gráfica os estados que um objeto pode assumir durante seu ciclo de vida, bem como as transições entre esses estados, frequentemente acionadas por eventos ou condições específicas. Estes diagramas permitem uma compreensão clara de como um sistema responde a eventos internos e externos, além de auxiliar na identificação de comportamentos críticos e no planejamento de soluções robustas.

Os elementos principais de um diagrama de estados incluem os estados, as transições, os pontos de início e fim, bem como as atividades internas, que podem ser categorizadas como entry, para ações executadas ao entrar no estado; do, para atividades realizadas enquanto o objeto permanece no estado; e exit, para ações realizadas ao sair do estado. Os diagramas de estados são amplamente utilizados em sistemas complexos que requerem controle preciso de eventos, como o sistema descrito neste documento, composto por cinco componentes principais: login e cadastro, planejamento de projetos, exportação documento, refinamento de requisitos e análise de impacto.

5.4.1. Diagrama de estados - Login e Cadastro

O diagrama de estados do Login e Cadastro representa o fluxo de interação do usuário com o sistema durante o processo de autenticação. Ele descreve como o sistema gerencia as etapas de cadastro e login, incluindo a validação das credenciais inseridas pelo usuário. O fluxo principal abrange o acesso às interfaces, a entrada de e-mail e senha, e a validação no banco de dados. Em caso de erros, como e-mails ou senhas inválidas, o sistema notifica o usuário e permite novas tentativas, garantindo que o processo seja robusto e de fácil uso.

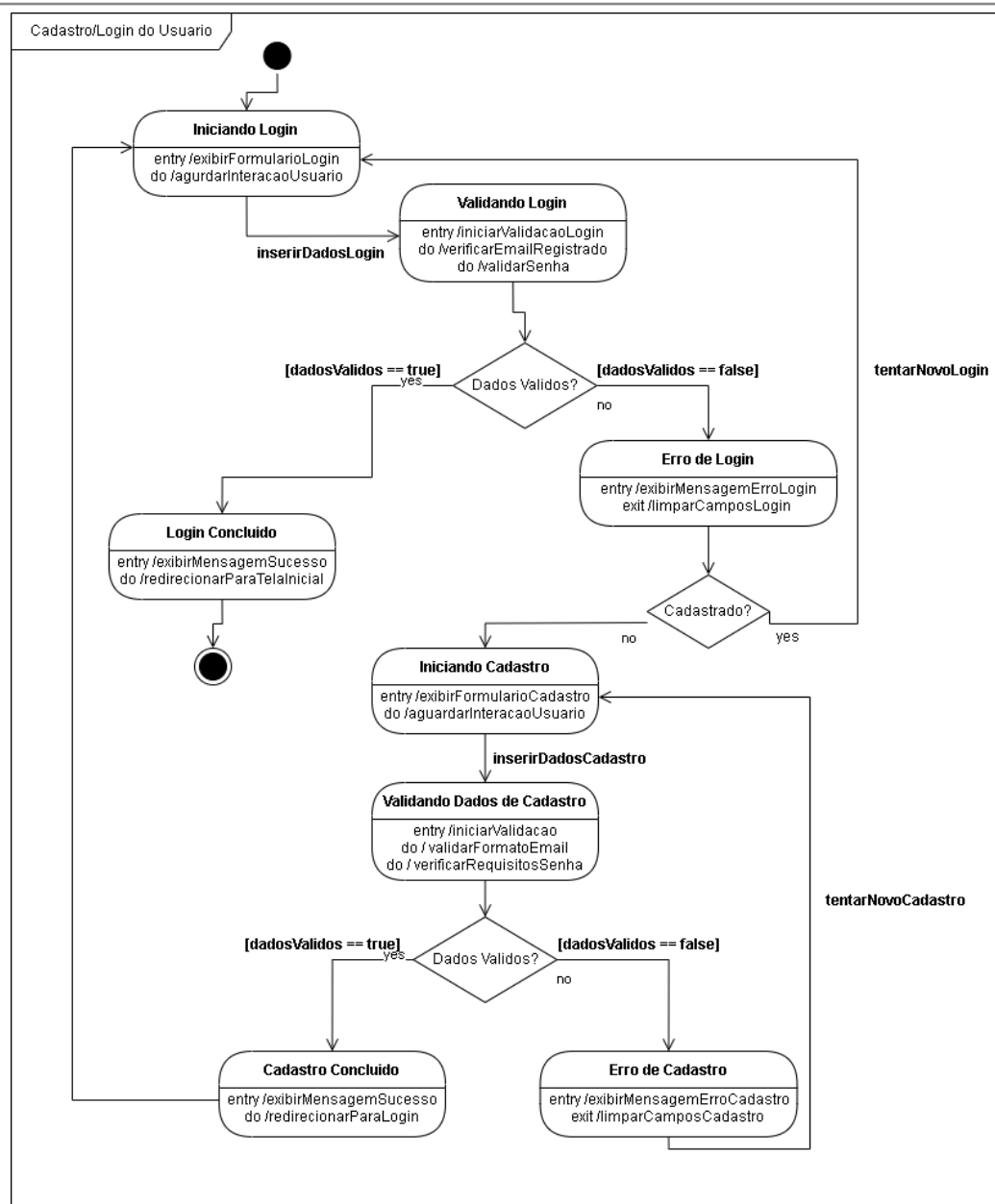


Figura 12: Diagrama de estados de Login e Cadastro.

5.4.2. Diagrama de estados - Planejamento de Projeto

O diagrama de estados de Planejamento de Projeto demonstra como o sistema utiliza inteligência artificial para interpretar requisitos textuais e gerar diagramas de caso de uso automaticamente. O fluxo principal inclui a entrada dos requisitos, o processamento das informações pela IA e a geração dos diagramas.

Os diagramas gerados são exibidos ao usuário para revisão. Caso sejam necessários ajustes, o sistema permite alterações e reprocessa os dados para gerar novos diagramas.

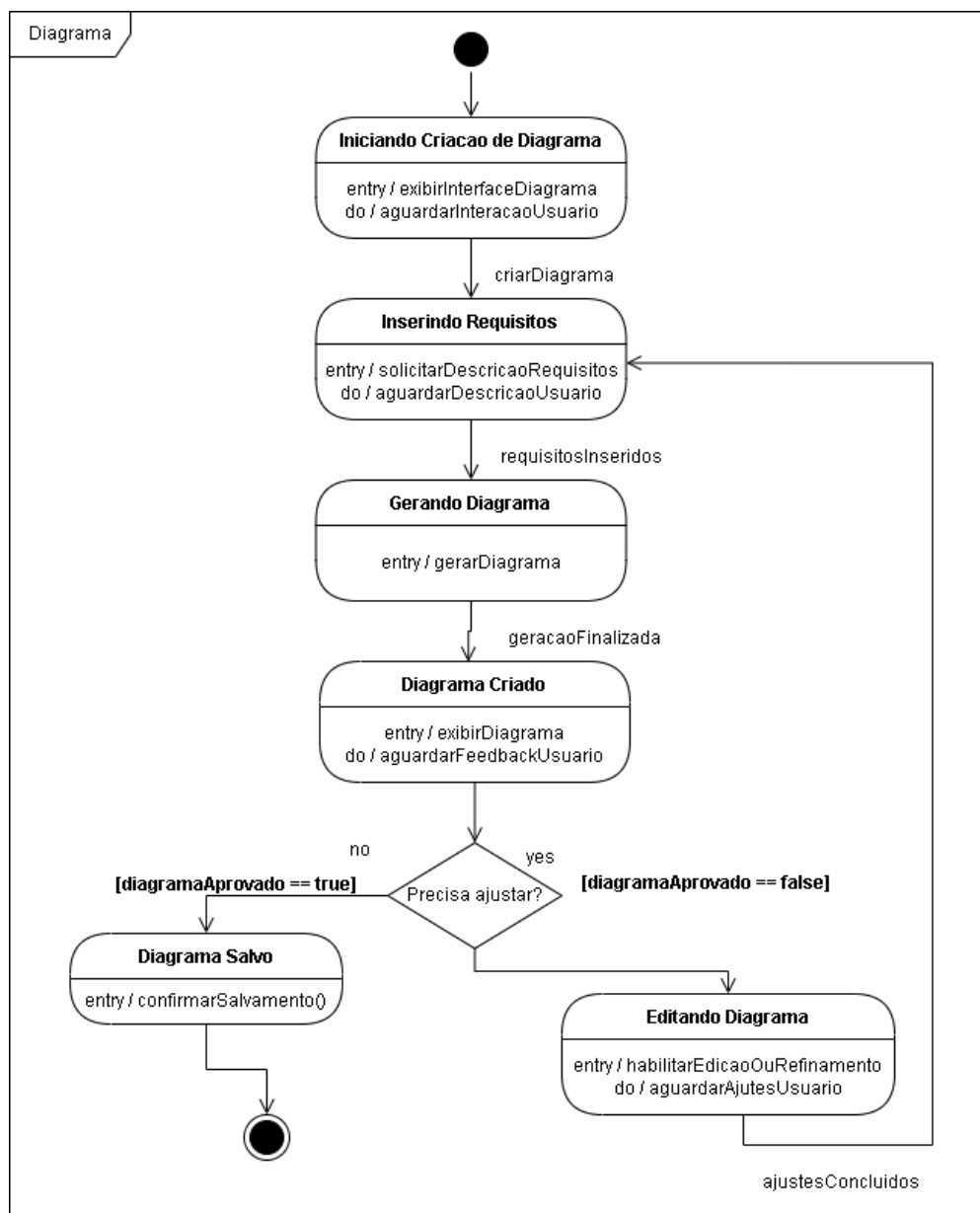


Figura 13: Diagrama de estados de Planejamento de Projeto.

5.4.3. Diagrama de estados - Exportação de Documento

O diagrama de estados da Exportação de Documento detalha como o sistema lida com o processo de exportação. O fluxo principal começa com o acesso à funcionalidade de exportação, seguido pela escolha do formato do documento,

que pode ser PDF, Word ou XML. Após a escolha, o sistema gera o arquivo e exibe um botão para download, finalizando o processo. Cenários alternativos, como falhas de configuração que impedem a apresentação das opções, também são considerados, com o sistema notificando o usuário e oferecendo a possibilidade de reiniciar o fluxo.

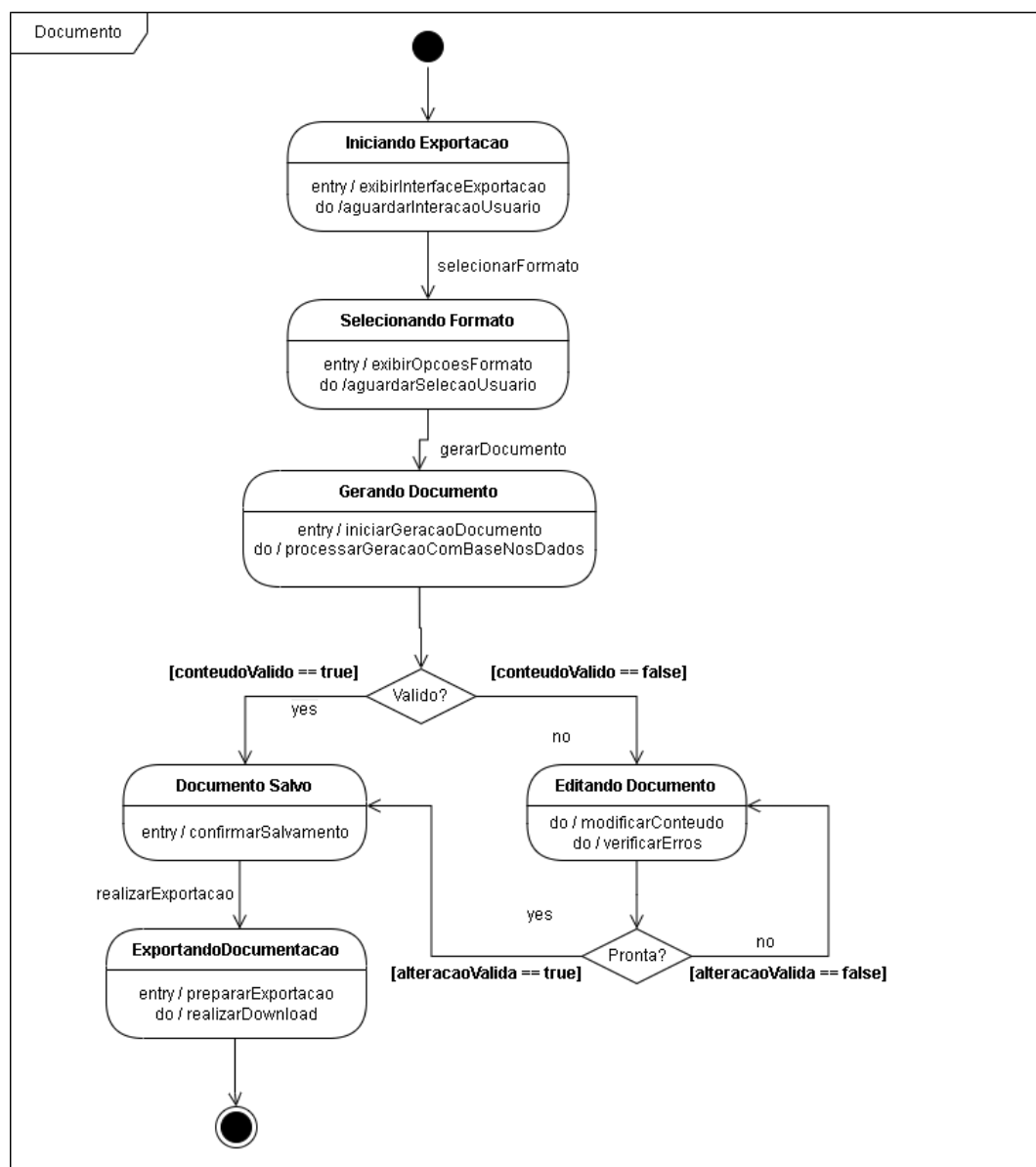


Figura 14: Diagrama de estados da Exportação de Documento



5.4.4. Diagrama de estados - Refinamento de Requisitos

No diagrama de estados do Refinamento de Requisitos, é descrito o processo pelo qual o sistema analisa os requisitos textuais fornecidos pelo usuário e utiliza IA para sugerir melhorias e requisitos complementares. O fluxo principal cobre desde a entrada dos requisitos até a exibição das sugestões detalhadas, que o usuário pode revisar e confirmar. Caso os requisitos sejam ambíguos ou as sugestões da IA não atendam às expectativas, o usuário pode ajustar os parâmetros ou refinar os requisitos para obter novas sugestões.

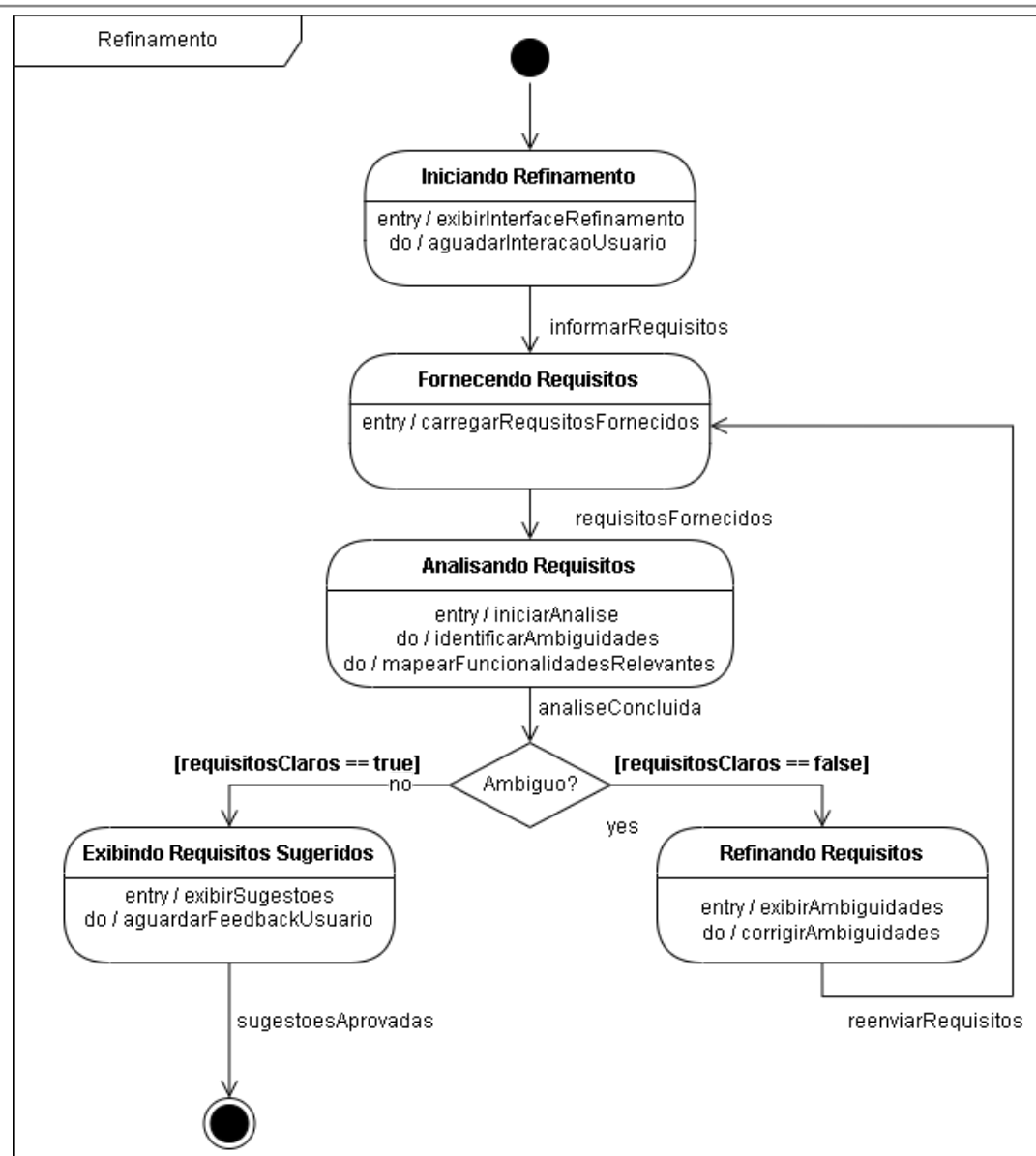


Figura 15: Diagrama de estados de Refinamento de Requisitos.

5.4.5. Diagrama de estados - Análise de Impacto

O diagrama de estados da Análise de Impacto ilustra como o sistema avalia as implicações de novos requisitos ou alterações em requisitos existentes. O sistema analisa casos de uso, dependências e riscos, gerando um relatório detalhado que informa ao usuário as áreas afetadas e possíveis ajustes. Se a alteração proposta não tiver impacto, o sistema notifica o usuário e conclui o

processo. Nos casos em que o impacto é muito extenso, o sistema alerta o usuário, permitindo ajustes antes de continuar.

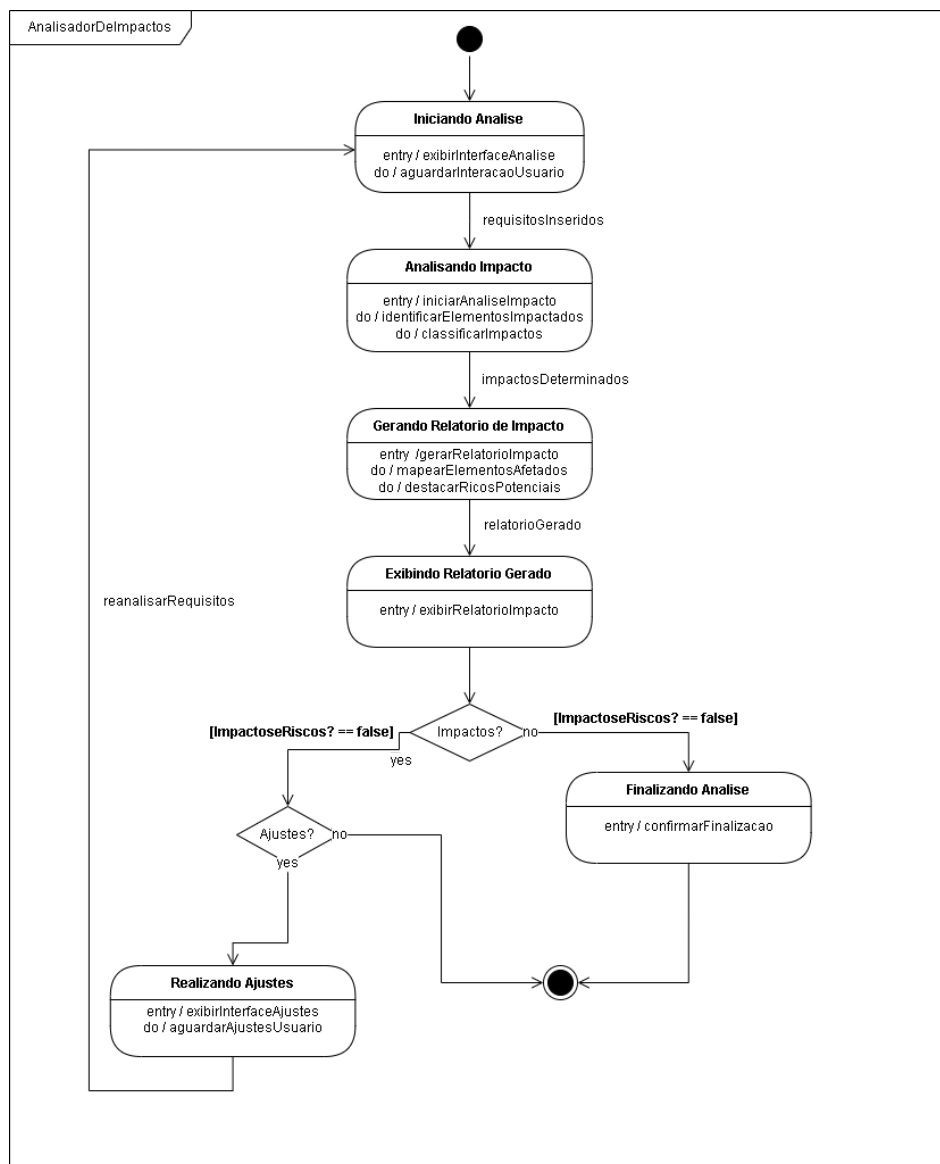


Figura 16 : Diagrama de estados de Análise de Impacto.

5.5. Diagramas de sequência

Um diagrama de sequência é um dos diagramas da UML (Unified Modeling Language) usado para modelar a interação entre diferentes partes de um sistema ao longo do tempo. É um grupo de objetos representados por linhas de vida e as mensagens que eles trocam durante a interação.



Os diagramas de sequência são fundamentais na modelagem de sistemas, pois permitem visualizar a interação temporal entre os objetos envolvidos em um processo. Estes diagramas representam a troca de mensagens em ordem cronológica, mostrando como os componentes de um sistema colaboram para atingir objetivos específicos. As principais partes de um diagrama de sequência incluem os participantes (objetos ou atores), as mensagens trocadas e os momentos em que essas interações ocorrem. A importância desse tipo de diagrama reside em sua capacidade de detalhar fluxos de trabalho, identificar dependências e melhorar a comunicação entre equipes de desenvolvimento e partes interessadas.

5.5.1. Diagrama de sequência - login e cadastro do usuário

O diagrama de sequência para login e cadastro ilustra a interação do usuário com o sistema durante a realização dessas tarefas. Ele também apresenta os fluxos alternativos, destacando os cenários em que o usuário já está registrado ou ainda não consta no banco de dados.

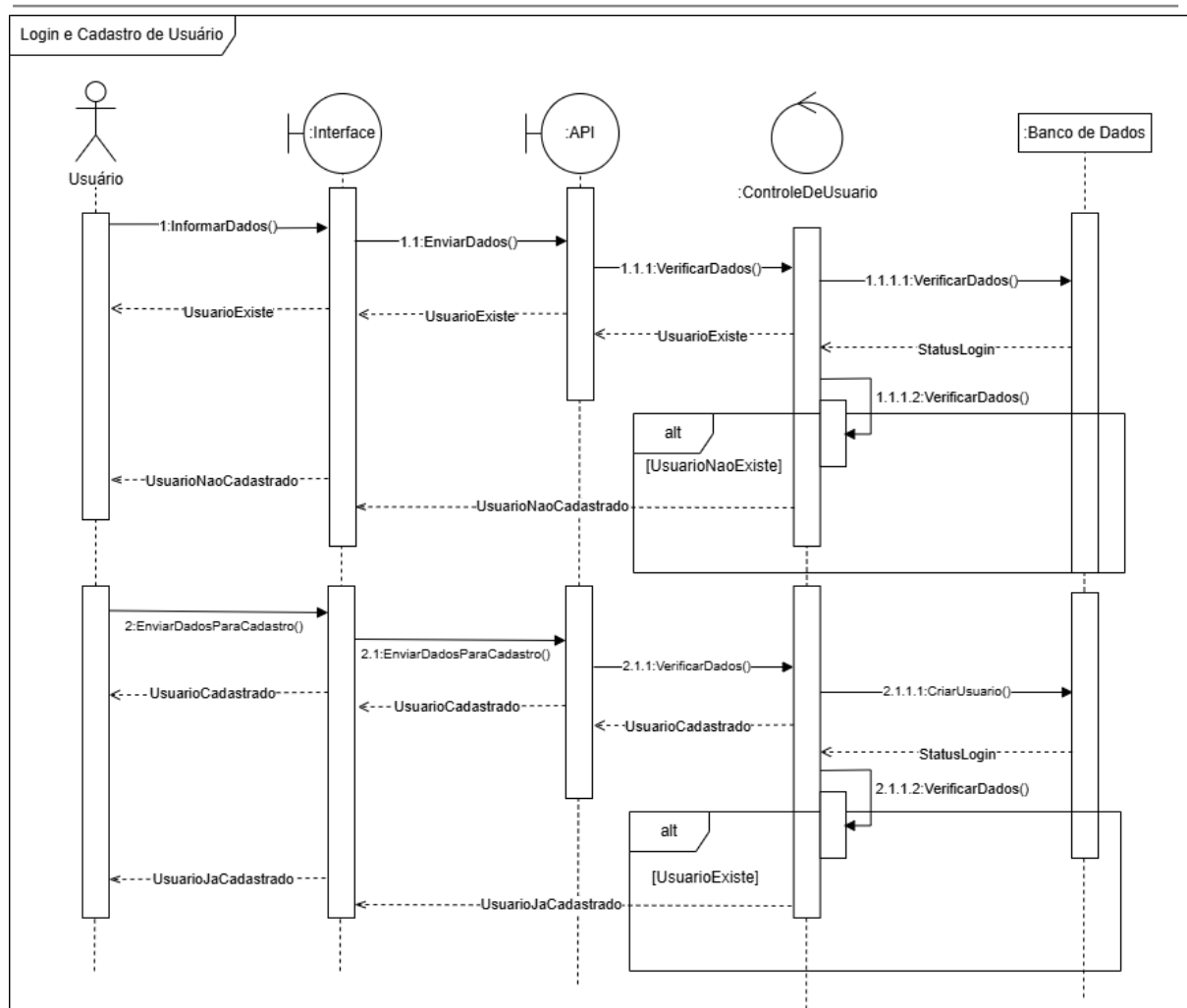


Figura 17 : Diagrama de sequência login e cadastro.

5.5.2. Diagrama de sequência - planejamento de projetos

O diagrama de sequência do planejamento de projetos demonstra o fluxo ideal de execução, bem como o cenário em que o usuário, ao não estar satisfeito com o diagrama gerado, pode realizar alterações para ajustá-lo conforme necessário.

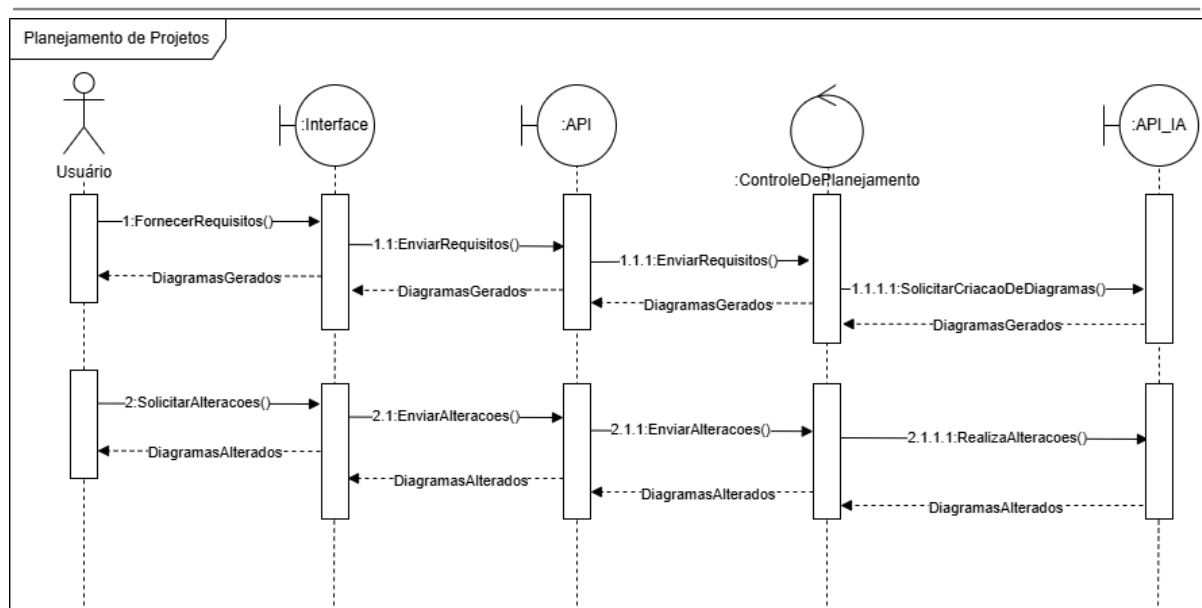


Figura 18: Diagrama de sequência planejamento de projetos.

5.5.3. Diagrama de sequência - refinamento de requisitos

O diagrama de sequência de refinamento de requisitos ilustra o processo em que o usuário insere os requisitos coletados durante a conversa com o cliente. A IA então refina esses requisitos, gerando versões adicionais e aprimoradas. O fluxo alternativo é acionado caso os requisitos inseridos sejam excessivamente ambíguos, exibindo uma mensagem de erro. Por fim, o diagrama também representa o momento em que o usuário reenvia os requisitos corrigidos, permitindo que a IA gere e entregue a versão final refinada.

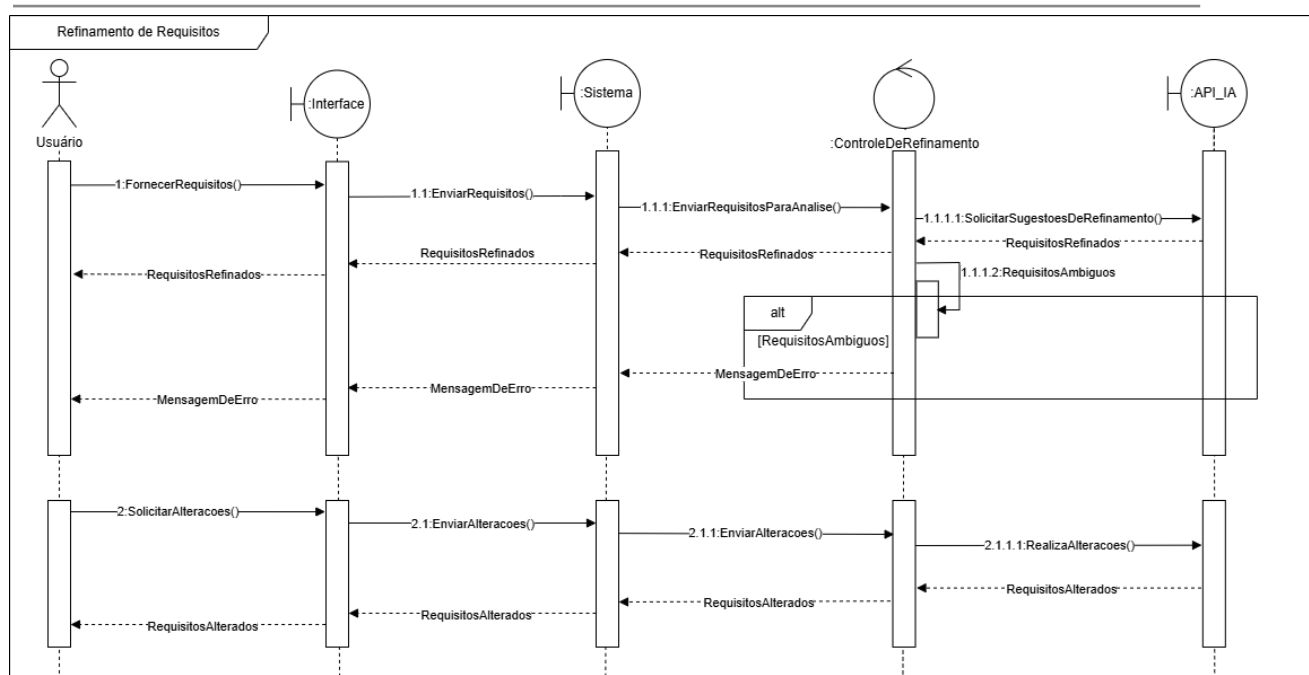


Figura 19 : Diagrama de sequência, refinamento de requisitos.

5.5.4. Diagrama de sequência - análise de impacto de caso de uso

O diagrama de sequência de análise de impacto de caso de uso ilustra o processo em que o usuário insere os casos de uso junto com as novas alterações. No fluxo principal, a IA gera um relatório de impacto. Caso os casos de uso sejam ambíguos, o fluxo alternativo é acionado, exibindo uma mensagem de erro. Por fim, o usuário pode enviar novas alterações, permitindo que a IA gere o relatório de impacto final.

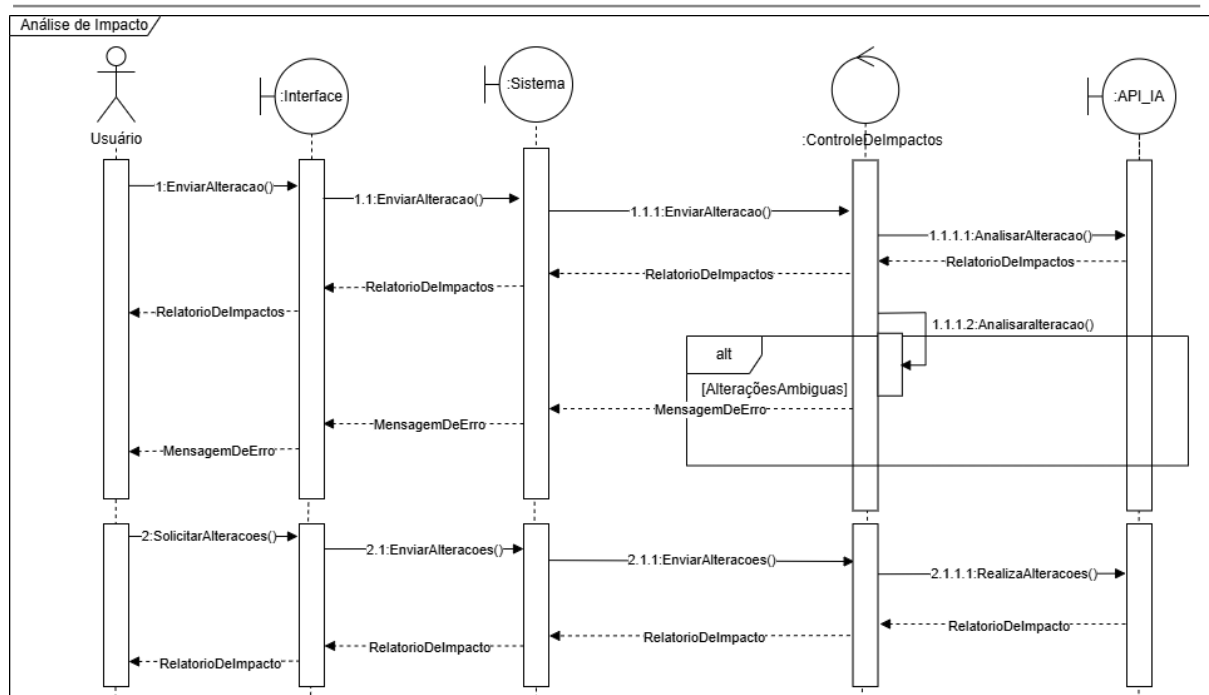


Figura 20 : Diagrama de sequência análise de impacto de caso de uso.

5.5.5. Diagrama de sequência - exportação de documentos

O diagrama de sequência de exportação de documentos representa o processo em que o usuário seleciona o formato desejado para a exportação. Se tudo ocorrer corretamente, o sistema de arquivos prepara o documento para download no formato solicitado. Caso ocorra alguma falha, o fluxo alternativo é acionado, exibindo uma mensagem de erro.

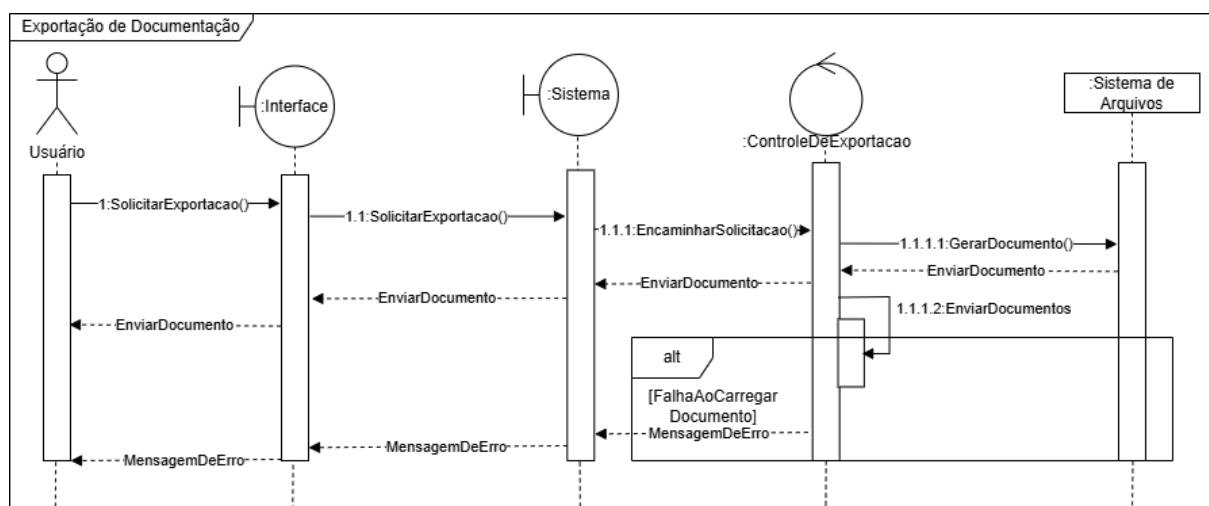


Figura 21 : Diagrama de sequência, exportação de documentos.



6. Tecnologias

Embora o sistema não tenha sido totalmente desenvolvido, realizamos a prototipação completa das interfaces utilizando Figma e CorelDRAW. O Figma foi usado para criar wireframes e protótipos interativos, permitindo a validação da experiência do usuário antes do desenvolvimento. Já o CorelDRAW foi utilizado para refinamento visual e criação de elementos gráficos detalhados, garantindo um design mais profissional para as telas do sistema.

Se o projeto fosse desenvolvido, a tecnologia escolhida para o backend seria Python com Django, devido à sua robustez e facilidade na criação de aplicações escaláveis. O Django oferece um conjunto completo de funcionalidades, incluindo segurança integrada, autenticação de usuários e uma arquitetura baseada no padrão MVC (Model-View-Controller), facilitando a organização do código. Além disso, sua compatibilidade com bibliotecas de inteligência artificial permitiria a análise automática de requisitos e a geração de sugestões aprimoradas.

Para o armazenamento de dados, utilizaríamos o PostgreSQL, um banco de dados relacional confiável e escalável, adequado para lidar com grande volume de informações de requisitos, diagramas e usuários. O PostgreSQL oferece suporte a transações seguras, integridade dos dados e excelente desempenho em consultas complexas, garantindo eficiência e confiabilidade no processamento das informações do sistema.

7. Escopo da Prototipação

A prototipação do sistema incluirá as seguintes telas referentes às funcionalidades presentes no software:



7.1 Tela de Login

Onde ocorre a autenticação dos usuários com suas credenciais.

The image shows a login interface for a system called DevCore. On the left, there is a purple vertical panel with a white circuit-like logo at the top. Below the logo, the word "DevCore" is written in a large, white, bold font. Underneath, in a smaller white font, it says "Seja bem-vindo!" followed by "O coração do seu projeto começa aqui!". To the right of this panel is a white area containing the login form. At the top of this white area is the heading "Realizar login" in a purple font. Below the heading are two input fields: the first is labeled "E-mail" with an envelope icon, and the second is labeled "Senha" with a padlock icon. Below these fields is a checkbox labeled "Mantenha-me conectado". At the bottom of the form are two purple buttons: "Login" and "Cadastre-se".

Figura 22 : Tela de login do sistema.

7.2 Tela de Cadastro

Onde o usuário pode fazer seu cadastro no sistema.

DevCore

Realizar cadastro

 Nome completo

 E-mail

 Senha

Cadastre-se

Figura 23 : Tela de cadastro.

7.3 Tela Inicial do Software

Tela onde contém todas as funcionalidades do sistema e uma breve explicação sobre elas.

DevCore



O que a Thalia pode fazer por você?



Refinamento de requisitos

Analisar e fazer melhorias nos casos de uso coletados durante entrevistas.



Planejamento de projetos

Gerar diagramas de casos de uso com base na inserção de requisitos do sistema.



Análise de impacto

Gerar relatórios de impacto para as mudanças em casos de uso do software.

 Falar com Thalia 

Figura 24 : Tela Inicial do Software.

7.4 Perfil do Usuário

Local onde contém todas as informações do usuário.

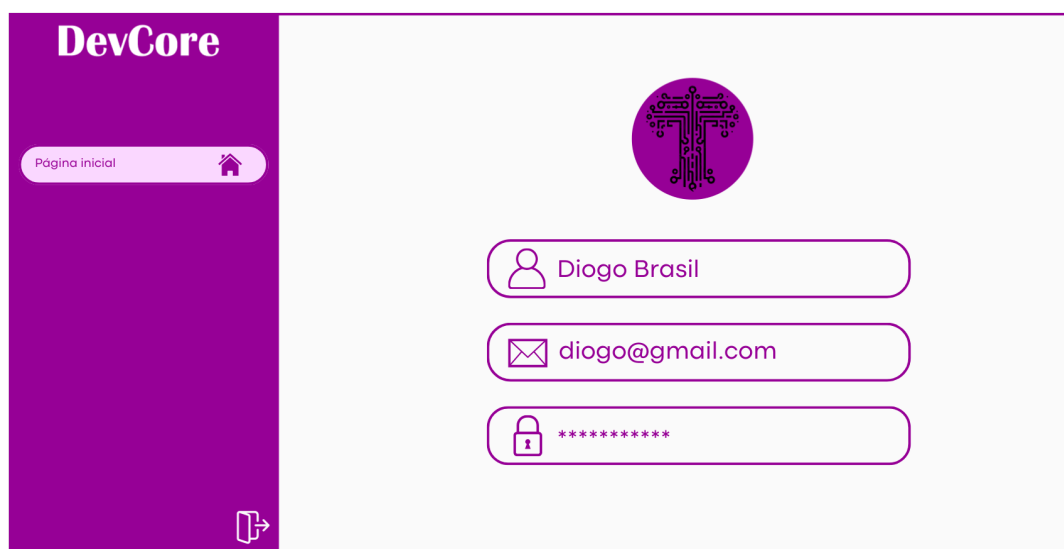


Figura 25 : Perfil do Usuário.

7.5 Refinamento de Requisitos

Funcionalidade onde o usuário realiza ajustes nos requisitos passados.

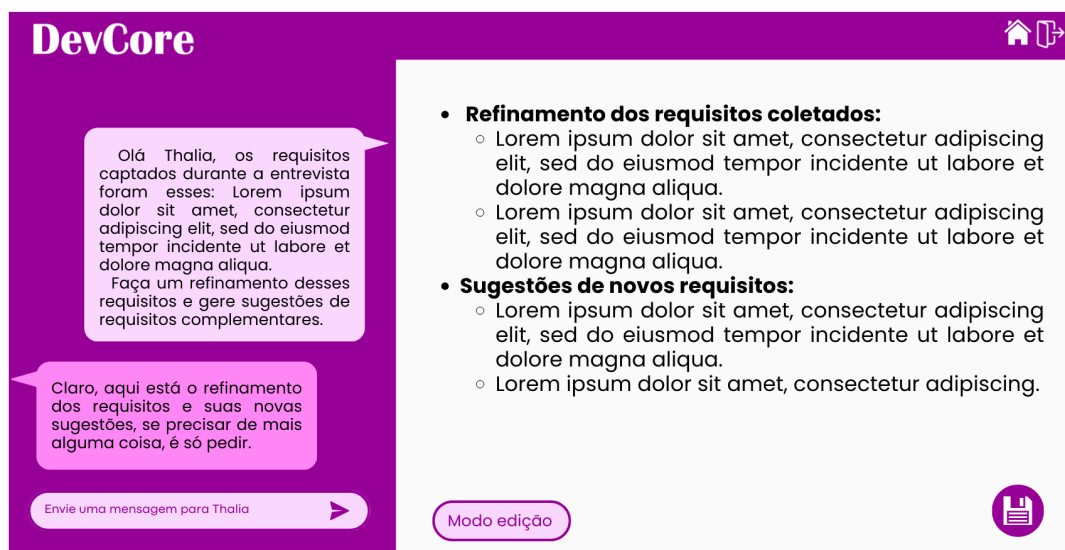


Figura 26 : Refinamento de Requisitos.

7.6 Planejamento de Projetos

Funcionalidade em que o usuário pode gerar diagramas e editá-los caso necessário.

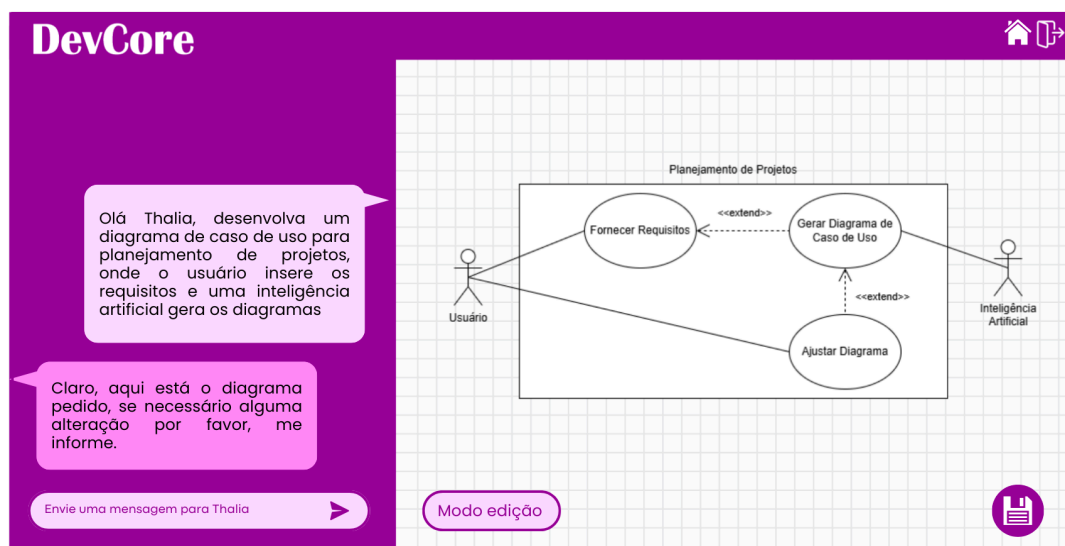


Figura 27 : Planejamento de Projetos - Geração de Diagramas.

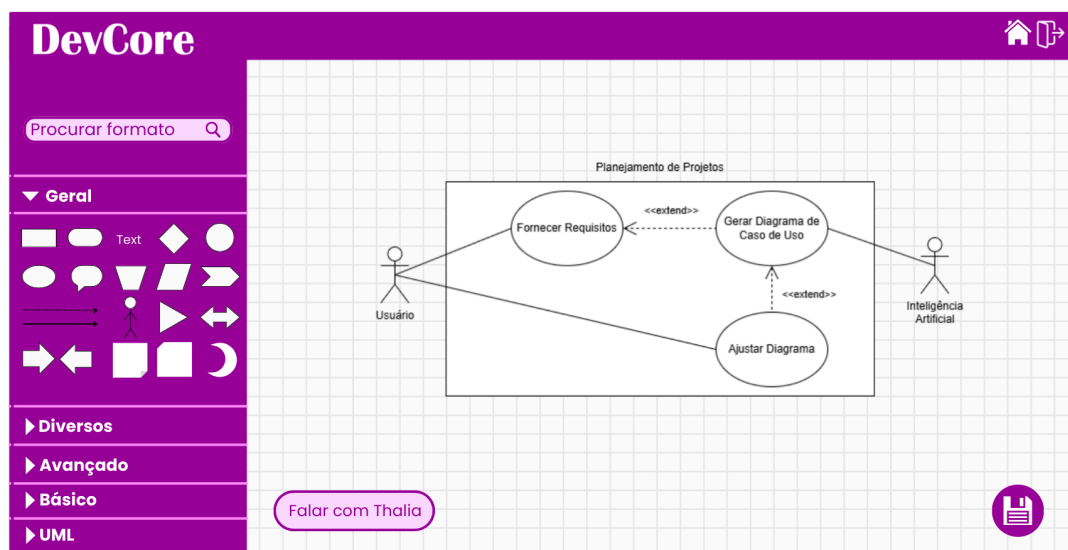


Figura 28 : Planejamento de Projetos - Modo Edição de Diagrama.

7.7 Análise de Impactos

Onde pode ser feito a verificação das possíveis mudanças e impactos que elas ocasionam no sistema.

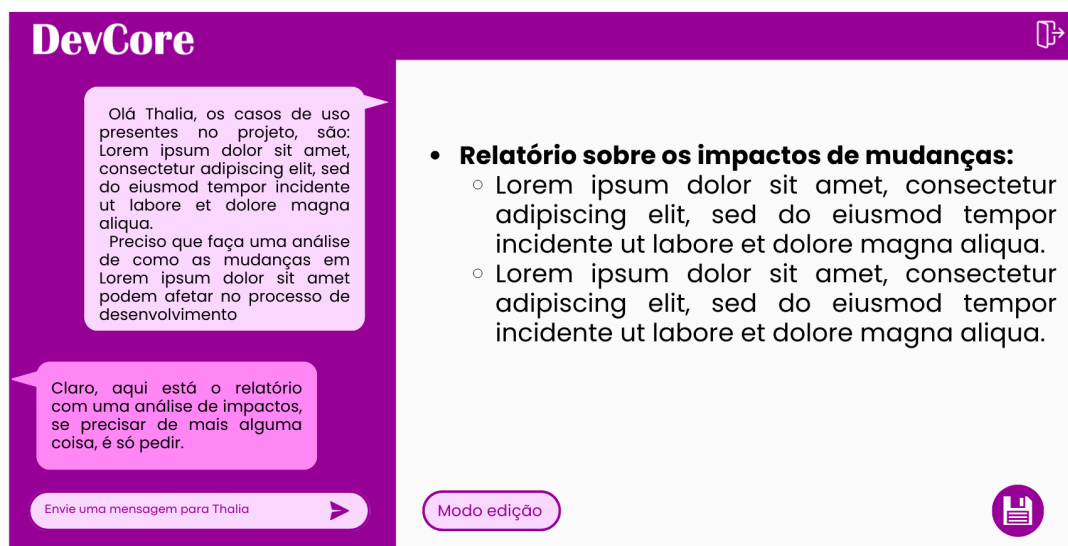


Figura 29: Análise de Impactos.

7.8 Exportação de Documentação

Local onde o usuário pode solicitar a geração de um documento daquilo que ele está fazendo, escolhendo um dos formatos disponíveis.

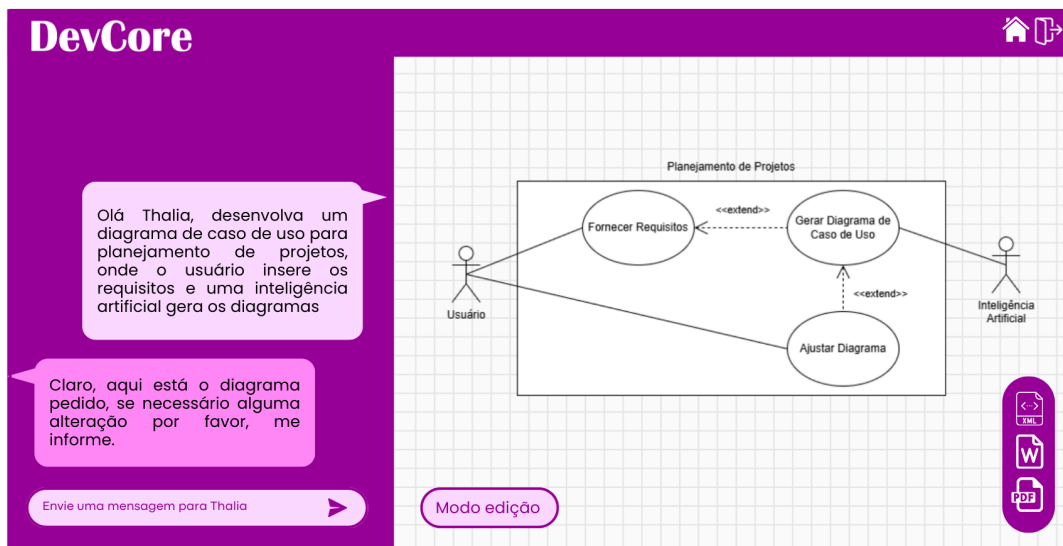


Figura 30 : Exportação de Documentação.

Conclusão

A conclusão deste projeto destaca a importância da automatização no processo de desenvolvimento de software, principalmente na análise de requisitos e geração de diagramas, proporcionando maior eficiência e precisão na hora de projetar e construir sistemas. A proposta desenvolvida integra inteligência artificial para facilitar a identificação de ambiguidades e a criação de diagramas, tornando o processo mais ágil e reduzindo erros manuais.

Embora o sistema não tenha sido totalmente implementado, sua concepção foi cuidadosamente planejada, incluindo a definição das tecnologias, a modelagem dos fluxos de funcionamento e a criação de protótipos interativos no Figma e CorelDRAW. Essa abordagem permitiu validar a viabilidade do projeto e oferecer uma visão clara de como a ferramenta poderia beneficiar empresas e profissionais que lidam com engenharia de software.

Dessa forma, o projeto se apresenta como uma solução inovadora para otimizar os processos para criação de sistemas, trazendo mais produtividade e confiabilidade



ao desenvolvimento de software. Caso venha a ser desenvolvido no futuro, ele poderá impactar significativamente a forma como equipes gerenciam e estruturam seus projetos, tornando os processos mais eficientes e organizados.

Referências

1. Thales. Teaching - Software Development Process. GitHub, Disponível em: <https://github.com/thalesvalente/teaching/tree/main/software-development-process>.
2. ROSA, Ângela - Requisitos de software funcionais e não funcionais: o que são?, 2024. Disponível em: <https://softdesign.com.br/blog/requisitos-de-software-funcionais-e-nao-funcionais/>. Acessado em: 22/12/2024
3. Diagrama de caso de uso UML: O que é, como fazer e exemplos, 2024. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acessado em: 22/12/2024
4. Diagramas de Caso de Uso, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>. Acessado em 08/01/2025
5. Requisitos funcionais e não funcionais: o que são?, 2024. Disponível em: <https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acessado em: 02/01/2025
6. Diagramas de Sequência, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>. Acessado em: 02/01/2025.
7. DALLAVALLE, Silvia Inês; CAZARINI, Edson Walmir. Regras do Negócio, um fator chave de sucesso no processo de desenvolvimento de Sistemas de Informação. Anais do XX ENEGEP-Encontro Nacional de Engenharia de Produção. São Paulo, 2000.



-
8. GUEDES, Gilleanes T. A. UML2: Uma Abordagem Prática. 3ª ed. São Paulo: Novatec, 201
 9. O que é um diagrama de classe UML? Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>>. Acesso em: 9 jan. 2025.
 10. Rational Software Architect Standard Edition 7.5.5. Disponível em: <<https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>>. Acesso em: 9 jan. 2025.
 11. TYBEL, D. Diagrama de classes (UML): Orientações básicas na elaboração. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 9 jan. 2025.
 12. O que é um diagrama de máquina de estados? Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>>. Acesso em: 9 jan. 2025.
 13. Rational Software Architect RealTime Edition 9.5.0. Disponível em: <<https://www.ibm.com/docs/pt-br/dmrt/9.5?topic=diagrams-state-machines>>. Acesso em: 9 jan. 2025.



Reconhecimentos e Direitos Autorais

@autor:Fernando da Silva Costa, Gabryella Cruz Sousa e Vitor dos Santos Sousa

@contato:gabryellacruzsousa@gmail.com, fernandodasilvacosta22@gmail.com e

@data última versão: 27/01/2025

@versão: 1.0

@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", sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2024.2, 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, conforme descrito abaixo, e, portanto, é compatível com a 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.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:



Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE 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 INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

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