

**CENTRO FEDERAL DE EDUCACAO TECNOLÓGICA
CELSO SUCKOW DA FONSECA**

**FlexRelat: Plataforma Web para criação de
relatórios comerciais, apoiada por Inteligência
Artificial Generativa**

Marcus Vinicius Gomes de Oliveira

Prof. Orientador:

Marcelo Areas Rodrigues da Silva, D.Sc.

**Rio de Janeiro,
20 de Janeiro de 2026**

**CENTRO FEDERAL DE EDUCACÃO TECNOLÓGICA
CELSE SUCOW DA FONSECA**

**FlexRelat: Plataforma Web para criação de
relatórios comerciais, apoiada por Inteligência
Artificial Generativa**

Marcus Vinicius Gomes de Oliveira

Projeto final apresentado em cumprimento às
normas do Departamento de Educação
Superior do Centro Federal de Educação
Tecnológica Celso Suckow da Fonseca,
CEFET/RJ, como parte dos requisitos para
obtenção do título de Bacharel em Ciência da
Computação.

Prof. Orientador:
Marcelo Areas Rodrigues da Silva, D.Sc.

**Rio de Janeiro,
20 de Janeiro de 2026**

DEDICATÓRIA

Dedico este trabalho à causa do povo palestino,
cuja luta por liberdade, dignidade e justiça
inspira a todos que acreditam em um futuro de
paz e igualdade. Palestina livre já!

AGRADECIMENTOS

Agradeço aos meus pais, por me darem a oportunidade de ser o primeiro da minha família a entrar em uma universidade federal.

Agradeço ao meu namorado pelo apoio dado até aqui, sou muito grato por sua vida e por toda a sua ajuda.

Agradeço também ao Marcelo Areas, meu professor orientador, por ter me ensinado com excelência e com muito comprometimento.

E por fim, agradeço aos amigos, em especial os camaradas do DCE do CEFET-RJ, que me ajudaram a ser quem eu sou, que me apoiaram em um dos momentos mais difíceis da minha vida, e que fizeram de tudo para me ajudar a superar os desafios da graduação e da vida. Só a luta muda o mundo.

RESUMO

A crescente necessidade de agilidade e precisão na execução de atividades complexas e repetitivas tem impulsionado o uso de tecnologias baseadas em Inteligência Artificial Generativa no cotidiano corporativo. Processos tradicionalmente onerosos, como a elaboração manual de relatórios detalhados, frequentemente enfrentam desafios significativos, incluindo a ocorrência de erros, atrasos na entrega e a exigência de elevado conhecimento técnico especializado. Além disso, a variabilidade na qualidade dos documentos produzidos pode comprometer a tomada de decisões estratégicas e operacionais, evidenciando a necessidade de soluções tecnológicas que simplifiquem e otimizem esses processos.

Propõe-se, portanto, a plataforma web FlexRelat, uma abordagem baseada em Inteligência Artificial Generativa (GenAI), que utiliza Modelos de Linguagem de Larga Escala (LLM) para gerar e editar relatórios personalizados a partir de entradas em linguagem natural, exigindo o mínimo de conhecimento técnico por parte do usuário. A plataforma foi validada por meio de testes funcionais e da coleta de feedbacks de usuários reais, permitindo avaliar desempenho, usabilidade e qualidade das funcionalidades assistidas por inteligência artificial. Com este projeto, busca-se não apenas apresentar uma solução viável para a geração de relatórios assistida por Inteligência Artificial Generativa, mas também contribuir para a compreensão do potencial dessas tecnologias no contexto corporativo, especialmente no que se refere à automação de processos documentais e ao aumento da eficiência operacional.

Palavras-chave: Inteligência Artificial Generativa; Relatórios inteligentes; Geração de relatórios; Plataforma Web; Aplicações com IA; Assistente virtual;

ABSTRACT

The growing need for agility and accuracy in the execution of complex and repetitive activities has driven the adoption of technologies based on Generative Artificial Intelligence in the corporate environment. Traditionally time-consuming processes, such as the manual preparation of detailed reports, often face significant challenges, including errors, delivery delays, and the requirement for a high level of specialized technical knowledge. In addition, variability in the quality of the documents produced can compromise strategic and operational decision-making, highlighting the need for technological solutions that simplify and optimize these processes.

Therefore, the FlexRelat web platform is proposed, an approach based on GenAI that uses LLM to generate and edit personalized reports from natural language inputs, requiring minimal technical knowledge from the user. The platform was validated through functional tests and the collection of feedback from real users, enabling the evaluation of performance, usability, and the quality of artificial intelligence–assisted functionalities. This project seeks not only to present a viable solution for AI-assisted report generation, but also to contribute to the understanding of the potential of these technologies in the corporate context, particularly with regard to the automation of document-related processes and the enhancement of operational efficiency.

Keywords: Generative Artificial Intelligence; Smart Reports; Report generation; Web platform; AI applications; Virtual assistance;

SUMÁRIO

1	Introdução	1
2	Fundamentação Teórica	3
2.1	Linguagem natural	3
2.2	Inteligência Artificial Generativa	4
2.3	Arquitetura em Camadas	6
2.4	API RESTful	8
3	Análise e Projeto	10
3.1	Arquitetura do sistema	10
3.2	Modelagem	11
3.2.1	Modelagem do Banco de dados	11
3.2.2	Diagrama de classes UML	15
3.2.3	Diagrama de casos de uso	15
3.3	Tecnologias utilizadas	18
3.4	Modelos de Inteligência Artificial e custos	19
3.5	Interface de usuário	20
3.5.1	Login	20
3.5.2	Documentos	21
3.5.3	Modelos	22
3.5.4	Edição de documento	23
3.5.5	Edição de modelo	23
3.5.6	Sugestão de alteração por chat	24
3.5.7	Menu de opções de alterações por IA	25
3.5.8	Menu de alteração de zoom e contagem de páginas	26
3.5.9	Modal de alteração do gráfico	27
3.5.10	Modal de dados	28
3.5.11	Modal de criação de modelo	29
4	Validação	31
4.1	Testes Funcionais	31

4.1.1	Teste de edição de documentos e modelos	32
4.1.2	Testes das funcionalidades com uso de Inteligência Artificial	34
4.2	Coleta de Feedbacks	36
4.2.1	Análise dos feedbacks coletados	38
5	Conclusões	40
	Referências Bibliográficas	42
	Apêndice A Requisitos do sistema	46
A.1	Requisitos funcionais	46
A.2	Requisitos não-funcionais	47
A.3	Regras de Negócio	49
	Apêndice B Especificação dos casos de Uso	53
	Apêndice C Diagrama de Classe UML	65

LISTA DE FIGURAS

FIGURA 1:	Representação visual da arquitetura em três camadas.	8
FIGURA 2:	Modelagem do banco de dados do sistema.	12
FIGURA 3:	Diagrama de classes UML simplificado do sistema.	15
FIGURA 4:	Primeira parte do diagrama de casos de uso do sistema.	16
FIGURA 5:	Segunda parte do diagrama de casos de uso do sistema.	17
FIGURA 6:	Tela de login.	21
FIGURA 7:	Tela de documentos.	22
FIGURA 8:	Tela de modelos.	22
FIGURA 9:	Tela de edição de documento.	23
FIGURA 10:	Tela de edição de modelo - Modo Visualização.	24
FIGURA 11:	Sugestão de alteração por chat.	25
FIGURA 12:	Menu de opções de alterações por IA.	26
FIGURA 13:	Menu de alteração de zoom.	27
FIGURA 14:	Modal de alteração de um gráfico.	28
FIGURA 15:	Modal de dados.	29
FIGURA 16:	Modal de dados.	30
FIGURA 17:	Diagrama de classes UML completo do sistema.	65

LISTA DE TABELAS

TABELA 1:	Descrição da entidade Users	12
TABELA 2:	Descrição da entidade Models	13
TABELA 3:	Descrição da entidade Documents	13
TABELA 4:	Descrição da entidade Messages	14
TABELA 5:	Descrição da entidade AI_Changes	14
TABELA 6:	Custos dos modelos da OpenAI utilizados no FlexRelat	19
TABELA 7:	Casos de testes para edição de documentos e modelos.	33
TABELA 8:	Resultados dos testes para edição de documentos e modelos.	33
TABELA 9:	Casos de teste das funcionalidades com uso de IA	35
TABELA 10:	Resultados dos testes das funcionalidades com IA	35
TABELA 11:	Perguntas do questionário de coleta de feedbacks	38
TABELA 12:	Requisitos Funcionais	46
TABELA 13:	Requisitos Não Funcionais	47
TABELA 14:	Regras de Negócio	49

LISTA DE ABREVIACÕES

GENAI	Inteligência Artificial Generativa	v, vi, 1–3, 5
HTTP	Protocolo De Transferência De Hipertexto	9
HTTPS	Protocolo De Transferência De Hipertexto Seguro	10
IA	Inteligência Artificial	1–5, 11, 28, 34, 36, 39–41, 47, 51, 52, 63, 64
LLM	Modelos De Linguagem De Larga Escala	v, vi, 1, 3, 5, 6
MVC	Modelo-Visão-Controlador	18
NLG	Geração De Linguagem Natural	4
NLU	Interpretação De Linguagem Natural	4
PLN	Processamento De Linguagem Natural	4
RAG	Geração Aumentada Por Recuperação	3
REST	Transferência De Estado Representacional	8, 9
URI	Identificador Uniforme De Recurso	9
UUID	Identificador Único Universal	12–14, 47

Capítulo 1

Introdução

O avanço da tecnologia proporcionou diversas mudanças na sociedade, que atualmente se organiza como uma civilização baseada quase exclusivamente na industrialização e na tecnologia [Monego et al., 2021]. Essas mudanças, segundo Pereira [2021], resultam em um mundo completamente volátil, com muitas exigências, principalmente em áreas correlatas ao gerenciamento de organizações e empresas.

As pessoas físicas e jurídicas têm enfrentado desafios cada vez mais complexos e incertos, demandando maior agilidade e precisão na tomada de decisões para a resolução ou atenuação dos problemas [Silva, 2023]. As tarefas atuais sofrem com alto grau de monotonia e repetição e, ao mesmo tempo, podem exigir elevada qualificação técnica para a resolução de problemas corriqueiros [Pereira, 2021].

Existem múltiplas soluções na área da computação desenvolvidas para atenuar os problemas mencionados, desde *hardwares*, como calculadoras avançadas capazes de realizar cálculos com maior rapidez e precisão do que humanos, até *softwares* que auxiliam tanto na automação de tarefas monótonas e repetitivas, como o processamento automático de planilhas, geração de relatórios e classificação de dados, quanto na execução de atividades mais complexas, como organizadores automáticos de agenda e editores de vídeo e imagem, solucionando problemas práticos do cotidiano.

Nos últimos anos, o mercado de *softwares* sofreu um crescimento de soluções baseados em Inteligência Artificial, principalmente utilizando a tecnologia de Inteligência Artificial Generativa (GenAI). Esse crescimento se deve ao fato do lançamento, em novembro de 2022, do modelo de Inteligência Artificial (IA) Generativa da OpenAI, o ChatGPT [Holmström and Carroll, 2024]. Após o lançamento dessa ferramenta, que podia apresentar conversas semelhantes à linguagem natural de um ser humano, os desenvolvedores de *software* viram isso como uma oportunidade para explorar os limites desse sistema, utilizando os modelos de linguagem de larga escala (LLM, do inglês, *Large Language Model*) para construir *softwares* que resolviam problemas a partir de uma entrada semelhante a uma conversa entre dois humanos, reduzindo ainda mais a complexidade na solução de problemas.

O uso da IA no dia a dia dos brasileiros já é uma realidade. Segundo o estudo *Nossa Vida com IA: Da inovação à aplicação*, divulgado pela Google Brasil em janeiro de 2025, cerca de 78% dos brasileiros utilizam IA no trabalho e 88% a consideram essencial para lidar com problemas complexos e inovação. Essa mesma pesquisa também indica que a maioria dos brasileiros tem a percepção de que a inteligência artificial transformará diversos setores da sociedade [Google and Ipsos, 2025].

A crescente demanda por geração de relatórios em setores como saúde, educação e negócios tem evidenciado um gargalo na eficiência operacional das organizações. A elaboração manual desses documentos não apenas consome recursos financeiros consideráveis, mas também exige um tempo excessivo de profissionais que poderiam estar dedicados a atividades de caráter estratégico [Davenport, 2014].

Além dos custos diretos, a manipulação manual de grandes volumes de dados para a criação de tabelas, gráficos e modelos complexos aumenta a possibilidade de erros humanos e atrasos na entrega de informações essenciais, gerando custos ocultos que impactam diretamente a tomada de decisão [Chugh and Grandhi, 2013].

Nesse contexto, o presente trabalho tem como objetivo corrigir essa problemática por meio do desenvolvimento de uma plataforma web para geração e edição de relatórios, fundamentada em Inteligência Artificial Generativa. O sistema atuará como uma interface de abstração que permite a criação de conteúdos personalizados com o suporte de GenAI, exigindo o mínimo de conhecimento técnico do usuário e tornando a produção de documentos complexos acessível a diversos níveis de especialização profissional.

O restante deste trabalho está estruturado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica, que aborda conceitos essenciais para a compreensão do trabalho. O Capítulo 3 descreve os aspectos de análise e projeto do sistema desenvolvido, reunindo as principais decisões de implementação, a modelagem e os diagramas de funcionamento do sistema, além de apresentar a interface de usuário do sistema. O Capítulo 4 discute o processo de validação do sistema desenvolvido e os resultados obtidos. O Capítulo 5 apresenta as conclusões do projeto, sintetizando os principais resultados alcançados, as limitações identificadas ao longo do desenvolvimento e as perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

A crescente monotonia, necessidade de agilidade e complexidade das tarefas atuais reflete na alta presença de tecnologias baseadas em GenAI no cotidiano, proporcionando transformações significativas na forma como pessoas e organizações resolvem problemas complexos [Microsoft, 2025]. Um desses problemas, que este trabalho busca resolver, é a criação de relatórios, tarefa com alto grau de complexidade para equipes de diversos contextos. Este capítulo abordará o referencial teórico essencial para o entendimento do estudo, como conceitos de inteligência artificial e sua interação com linguagens naturais, IA Generativa com o uso de LLM, Geração Aumentada por Recuperação (RAG), visando o aumento da confiabilidade, arquitetura em camadas, adotada como a principal do sistema, e conceitos de API RESTful fundamentais para compreender a estrutura da camada de aplicação.

2.1 Linguagem natural

Para Gomes [2010], o ser humano possui uma capacidade exclusiva de raciocínio e por muito tempo busca compreender de que forma conseguem entender, perceber, prever e manipular o mundo à sua volta, com o objetivo de solucionar problemas mais complexos do que podem compreender. Portanto, uma possível solução para os problemas complexos que existem na humanidade é a construção de soluções computacionais capazes de lidar com processos difíceis, compreendendo, prevendo e manipulando dados para gerar respostas inteligentes, como faz a mente humana [Hunt, 2014].

McCarthy [1959], criou a ideia de um sistema de computação ideal, que seria um “tomador de conselhos”, que receberia frases em linguagens formais e resolveria problemas por meio da manipulação dessas sentenças. Esse programa, portanto, seria um marco na produção de soluções modernas, pois resolver problemas complexos exige “compreensão”, habilidade que máquinas não possuem [Hunt, 2014].

Essa ideia de sistema se destaca ao considerar que, embora os registros humanos estejam sendo transformados cada vez mais em linguagem compatível com sistemas computacionais,

grande parte das nossas transações são compartilhadas em linguagem natural, por meio de conversas, documentos, imagens, publicações, entre outras formas. Essas transações geram um enorme volume de dados, resultando em registros detalhados sobre o funcionamento da nossa sociedade, exigindo formas automatizadas para a manipulação desses dados, pois ultrapassam a capacidade humana de processamento [Hunt, 2014][Gonzalez and Lima, 2003].

Para Hunt [2014], o sucesso na resolução de problemas de registro, resumo e análise de dados, principalmente em áreas onde o volume de dados é muito grande, dependerá da criação de sistemas capazes de compreender as complexidades das linguagens humanas naturais.

O conceito de “linguagem natural” se refere as línguas faladas pelos humanos, que são distintas de tipos de linguagens como as matemáticas, visuais, gestuais e de programação. Para a tarefa de fazer computadores entenderem a linguagem natural, foram desenvolvidos estudos em Processamento de Linguagem Natural (PLN), que é um subcampo da Inteligência Artificial, que estuda a interação entre computadores e línguas naturais humanas. O PLN visa permitir que os computadores entendam, interpretem e gerem a linguagem humana de forma assertiva [Islam and Moushi, 2024].

O PLN tem duas subáreas principais que são a Interpretação de Linguagem Natural (NLU) e Geração de Linguagem Natural (NLG). NLU é o estudo do processamento com o fim de analisar e interpretar a língua, segmentando e classificando os componentes linguísticos e realizando tentativas de compreensão de significados feitos pelos seres humanos. O NLG, etapa após o NLU, possui o objetivo de simular linguagem natural, gerando textos parecidos aos feitos por seres humanos [Islam and Moushi, 2024].

2.2 Inteligência Artificial Generativa

Desde 1940, com a invenção da bomba eletromecânica de Turing, até os dias de hoje, os estudos sobre Inteligência Artificial objetivam a criação de sistemas capazes de realizar tarefas humanas cansativas, de alto esforço e também com elevados níveis de complexidade [Pereira, 2021].

Muitas pessoas já perceberam o poder da inteligência artificial para mudar o rumo da sociedade em que vivemos. Para Agrawal et al. [2022], a IA tem o "potencial de transformar economias", essa narrativa vai de encontro com as ideias do CEO da Google, Sundar Pichai, que em 2016 afirmou que a Inteligência Artificial terá um impacto no mundo, maior que inovações que mudaram o rumo da história no passado, segundo ele “A IA é uma das coisas mais

importantes em que a humanidade está trabalhando, é mais profunda do que eletricidade ou fogo” [Pichai, Sundar, 2018].

Segundo Feuerriegel et al. [2024], a Inteligência Artificial Generativa é um conjunto de processos computacionais que podem criar conteúdos inéditos, como imagens, textos, áudios, a partir do treinamento em uma base de dados estabelecida.

Diferente do pensamento comum, os sistemas baseados em IA Generativa não precisam ser exclusivamente usados para fins criativos, em produções de arte no geral, mas possuem uma ampla área de possíveis atuações, como a geração de respostas para perguntas complexas, tradução de textos, assistentes ou automatizadores de programação, automatização de processos complexos e análise de dados estatísticos, fornecendo *insights* em cima de dados pré-estabelecidos, resumindo textos equiparáveis a profissionais de alto nível, dentre outros exemplos [Feuerriegel et al., 2024] [Wan et al., 2025] [Fountaine et al., 2019] [Figueiredo et al., 2025].

No geral, a Inteligência Artificial Generativa pode ser usada para a redução de tempo despendido em tarefas repetitivas e complexas, porque, a depender da base de dados em que a Inteligência Artificial foi treinada, ela pode possuir alto grau de assertividade e de forma mais rápida que um ser humano [Wan et al., 2025] [Figueiredo et al., 2025] [Kaplan, 2024]. Com isso, a GenAI pode ajudar empresas a cortarem custos, economizando tempo e recursos de pessoas, permitindo que os funcionários usem melhor o seu tempo para atividades críticas do negócio [Holmström and Carroll, 2024] [Sadeghpour et al., 2025].

Para explorar todo esse potencial da GenAI, é comum o uso de modelos pré-treinados e de uso geral. Um modelo de IA Generativa usa os conceitos de aprendizado de máquina para gerar informações novas baseadas em padrões e relações obtidas durante o treinamento do modelo [Feuerriegel et al., 2024].

Um dos tipos de modelos mais comuns da atualidade são os chamados LLM, ou Modelo de Linguagem de Grande Escala. Uma LLM é um tipo de modelo de GenAI, baseado em redes neurais, que é capaz de tratar de uma grande variedade de processos de Linguagem Natural, como compreensão, classificação e geração de texto [Feuerriegel et al., 2024] [Wang et al., 2019]. Esse tipo de modelo é treinado com uma grande quantidade de dados textuais, fazendo com que o modelo possa entender padrões de linguagem e os mais diversos contextos onde aparecem diversas combinações de palavras [Ray, 2023].

As LLM são conhecidas principalmente pelo seu uso em geração de textos, por causa da alta qualidade da criação textual, com coesão e coerência semelhantes às de um humano. Sendo

assim, elas são fundamentais na criação de sistemas como *ChatBots* e sistemas de criação de imagem através de entradas de textos [Wan et al., 2025] [Wang et al., 2019].

Os LLM tiveram avanços significativos, tendo exemplos marcantes, como GPT-4.1 da OpenAI, Gemini do Google e DeepSeek-R1 da DeepSeek [OpenAI, 2024] [DeepMind, n.d.] [DeepSeek, n.d.]. Estes modelos são baseados na arquitetura *transformer* e utilizam mecanismos de atenção para otimizar a compreensão do contexto. Ao priorizar as partes mais relevantes do texto de entrada, eles conseguem gerar respostas mais fluidas e precisas, posicionando-se como pilares fundamentais para a criação de sistemas modernos. Modelos como o DeepSeek e o Gemini estão disponíveis em código aberto, fazendo com que os usuários possam baixar o seu conteúdo e executar localmente, caso tenham um equipamento compatível com o poder de processamento necessário. Isso permite uma maior flexibilidade para uma maior quantidade de casos de uso, além de facilitar a criação de sistemas mais seguros.

As LLM ainda possuem limitações inerentes ao tipo de modelo, como geração de informações incorretas, imprecisas ou totalmente inventadas, que são chamadas de alucinações, e também de conhecimento interno desatualizado [Fan et al., 2024]. Para Dahl et al. [2024], esses problemas são perceptíveis principalmente quando há uma busca sobre campos de conhecimentos específicos, como na medicina e no direito, por exemplo, pesquisas sobre legislação possuem uma taxa de alucinação alta, que pode variar de 69% a 88% nas LLM de última geração, ou seja, os modelos possuem taxas de erro maiores do que de acertos, levando a um problema de falta de confiabilidade nos modelos já existentes.

2.3 Arquitetura em Camadas

Segundo Valente [2020], a arquitetura em camadas é um dos padrões de arquitetura de software mais utilizados do mundo desde o desenvolvimento de programas de grande porte nas décadas de 1960 e 1970. Nessa arquitetura, os sistemas são estruturados de forma modular em camadas lógicas, geralmente apresentação, aplicação e dados, que possuem uma hierarquia. Em sistemas desse tipo, uma camada depende apenas dos serviços da camada imediatamente inferior a ela. Por exemplo, a camada de apresentação deve interagir apenas com a camada de aplicação, que por sua vez se comunica com a camada de dados. Essa estrutura evita acessos cruzados e garante que alterações em uma camada não impactem diretamente nas demais. Dessa forma, esse tipo arquitetural promove modularidade, isolamento e baixo acoplamento [Valente, 2020] [Thomas and Webb, 2023].

Para viabilizar notificações e mensagens assíncronas entre camadas sem violar a independência entre elas, são utilizados padrões como *Observer*, padrão de projeto comportamental que permite que você defina um mecanismo de assinatura para notificar múltiplos objetos sobre quaisquer eventos que aconteçam com o objeto que eles estão observando, e mecanismos como *callbacks* e eventos. A comunicação de camadas inferiores com superiores deve ocorrer por interfaces genéricas, sem dependência de detalhes específicos [Thomas and Webb, 2023].

A arquitetura em camadas pode ser implementada em diferentes formatos, sendo a arquitetura em três camadas uma das mais comuns em sistemas corporativos. Nessa estrutura, a separação entre camada de apresentação, aplicação e banco de dados permite distribuir responsabilidades de forma clara e organizada. Além disso, essa divisão facilita a manutenção e a escalabilidade do sistema, uma vez que cada camada pode ser atualizada ou substituída independentemente, desde que as interfaces entre elas sejam preservadas. Em arquiteturas distribuídas, cada camada pode ser executada em uma máquina distinta, otimizando o desempenho e o uso de recursos [Valente, 2020].

A camada de aplicação, que centraliza as regras de negócio, costuma conter submódulos, como fachadas e mecanismos de persistência, reforçando os princípios da arquitetura em camadas. Isso garante que os dados sejam validados e acessados de forma segura. Por isso, a arquitetura em três camadas é amplamente adotada como padrão em sistemas web e corporativos modernos, oferecendo maior robustez e flexibilidade [Valente, 2020] [Tu, 2023].

A Figura 1 ilustra a organização típica da arquitetura em três camadas, destacando a separação entre a interface com o usuário, a lógica de negócio e o banco de dados, bem como a distribuição dessas camadas em diferentes ambientes computacionais.

Fonte: Elaborado pelo autor.

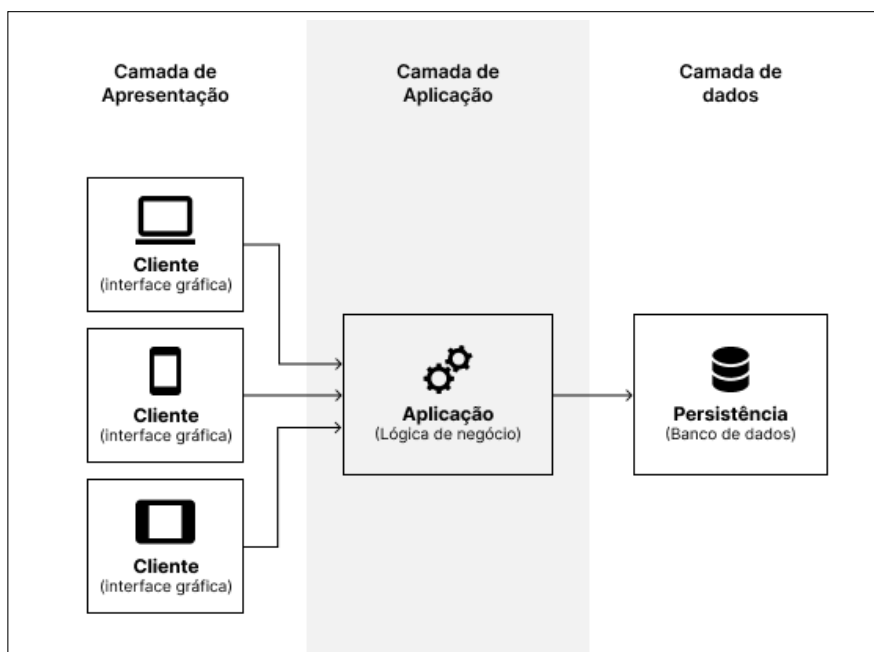


Figura 1: Representação visual da arquitetura em três camadas.

No contexto do projeto, a arquitetura em camadas permite evoluir facilmente cada parte do sistema, adicionar novas funcionalidades sem grandes impactos e garantir que testes e *deployments* ocorram de forma independente e confiáveis.

2.4 API RESTful

O termo Transferência de Estado Representacional (REST), foi proposto por Roy Fielding em sua tese de doutorado no ano de 2000, na qual definiu os princípios desse estilo arquitetural [Fielding and Taylor, 2000]. A dissertação de Fielding se deu em contrapartida a problemas decorrentes do grande sucesso das redes de computadores e a um aumento expressivo no consumo de serviços web [Dos Santos, 2013]. Segundo Fielding and Taylor [2000], a arquitetura REST surgiu como uma solução para os principais problemas do cotidiano do desenvolvimento de soluções para essa nova web de grande escala, como:

- Guiar o desenvolvimento da arquitetura web, com princípios claros.
- Padronizar a comunicação entre sistemas, com foco em simplicidade e escalabilidade.
- Separar cliente e servidor, para permitir a evolução dos sistemas de forma independente.

- Usar a semântica do Protocolo de Transferência de Hipertexto (HTTP) (GET, POST, PUT, DELETE) de forma mais correta.
- Permitir cache e comunicação sem estado, para assim otimizar o desempenho.

A arquitetura REST é definida por cinco restrições que garantem simplicidade, escalabilidade e padronização. A primeira é a separação entre cliente e servidor, onde o cliente lida com a interface e o servidor com a lógica de negócio e persistência dos dados, permitindo evolução independente. A segunda é o princípio stateless, ou seja, cada requisição deve conter todas as informações necessárias para ser processada, sem depender de sessões armazenadas no servidor. A terceira restrição exige que as respostas do servidor sejam cacheáveis, informando se podem ser armazenadas para reutilização. A quarta é o uso de uma interface uniforme, com métodos HTTP padronizados, Identificador Uniforme de Recurso (URI) para identificação de recursos e mensagens autoexplicativas, que garante consistência na comunicação. Por fim, a quinta restrição é que a arquitetura deve ser em camadas, permitindo o uso de servidores intermediários, sem que o cliente precise saber da estrutura completa entre ele e o servidor final.

No contexto do atual trabalho, a arquitetura RESTful é usada na camada de aplicação para expor a API que conecta o frontend à lógica de negócio e à inteligência artificial. Essa escolha objetiva a simplicidade, flexibilidade e escalabilidade, além de facilitar a manutenção e futuras integrações.

Capítulo 3

Análise e Projeto

3.1 Arquitetura do sistema

A arquitetura do sistema desenvolvido neste trabalho foi planejada para ser simples, modular e de fácil manutenção, com o objetivo de atender usuários com diferentes níveis de conhecimento técnico e permitir futuras evoluções da plataforma. A abordagem adotada foi a de uma arquitetura em três camadas: camada de apresentação (*frontend*), camada de aplicação (*backend*) e a camada de persistência de dados (banco relacional), implementada utilizando tecnologias modernas e compatíveis com aplicações web interativas e responsivas.

A interface do usuário foi desenvolvida utilizando o *framework* Next.js, permitindo renderização híbrida com *Server-side Rendering* (SSR) e *Client-side Rendering* (CSR), navegação fluida e interação com a API. Esta camada é responsável pela comunicação com o usuário, oferecendo recursos para criação, edição e visualização de relatórios.

A camada de aplicação foi construída com Node.js e expõe uma API RESTful que centraliza a lógica de negócio e o gerenciamento das entidades principais: usuários, modelos, documentos e mensagens. Essa API também é responsável pela orquestração das chamadas à inteligência artificial generativa (por meio da API da OpenAI), garantindo que os conteúdos gerados sejam integrados aos relatórios conforme as ações dos usuários.

O banco de dados utilizado é o PostgreSQL, escolhido por sua robustez e suporte a operações relacionais complexas. Ele armazena as informações estruturadas do sistema, incluindo usuários, documentos, mensagens enviadas e os modelos de relatórios disponíveis.

A comunicação entre as camadas ocorre via protocolo Protocolo de Transferência de Hipertexto Seguro (HTTPS), utilizando JSON como formato de troca de dados. Para autenticação e controle de acesso, foi implementado um mecanismo baseado em *tokens* JWT, garantindo segurança na identificação dos usuários e proteção das rotas sensíveis da aplicação.

A arquitetura adotada facilita a manutenção, detém clara separação de responsabilidades e uma boa escalabilidade horizontal.

3.2 Modelagem

A modelagem do sistema foi feita tendo em vista a proposta principal do trabalho que é facilitar a geração e edição de relatórios personalizados para usuários de diferentes níveis de conhecimento técnico, usando Inteligência Artificial Generativa. O objetivo principal é garantir que a estrutura do sistema seja simples, porém flexível para possíveis futuras atualizações e adaptações a novos contextos de uso.

A etapa de modelagem foi feita baseando-se nas condições de software obtidas através de um processo de mapeamento de todos os requisitos funcionais, não funcionais, regras de negócios e casos de uso, na etapa de elicitação de requisitos do software, que estão disponibilizados no Apêndice A.

Nesse processo, buscou-se garantir que a estrutura do sistema fosse simples, porém flexível o suficiente para futuras expansões e adaptações a novos contextos de uso, utilizando os conceitos de arquitetura em três camadas e arquitetura RESTful para o servidor do projeto.

O sistema foi idealizado com base em quatro entidades centrais: Users, Models, Documents e Messages. De forma simplificada, Users representa os usuários que vão interagir com o sistema, podendo visualizar, cadastrar, atualizar e remover uma série de dados. Models é a representação dos modelos de arquivos disponíveis na plataforma, criados pelos usuários da plataforma e inseridos de forma estática no banco de dados durante o desenvolvimento inicial. Documents são os documentos criados pelos usuários, que representam os arquivos dos relatórios que o usuário deseja gerenciar. Messages são as mensagens trocadas entre o usuário e a IA da plataforma em um documento ou modelo.

A partir dessas entidades, foram definidos os relacionamentos e estruturas necessárias para garantir integridade, desempenho e usabilidade. A modelagem também considerou aspectos de acessibilidade e validação, assegurando que o sistema seja inclusivo e confiável em diferentes contextos.

Os tópicos a seguir explicitam modelagens mais visuais e essenciais para a identificação das necessidades do projeto como a modelagem do banco de dados e diagramas.

3.2.1 Modelagem do Banco de dados

A modelagem do banco de dados foi estruturada de forma a refletir diretamente as cinco entidades principais do sistema. Cada entidade foi mapeada em tabelas relacionais com atributos

específicos que garantem a integridade e coerência dos dados. A Figura 2 apresenta o modelo lógico do banco de dados.

Fonte: Elaborado pelo autor.

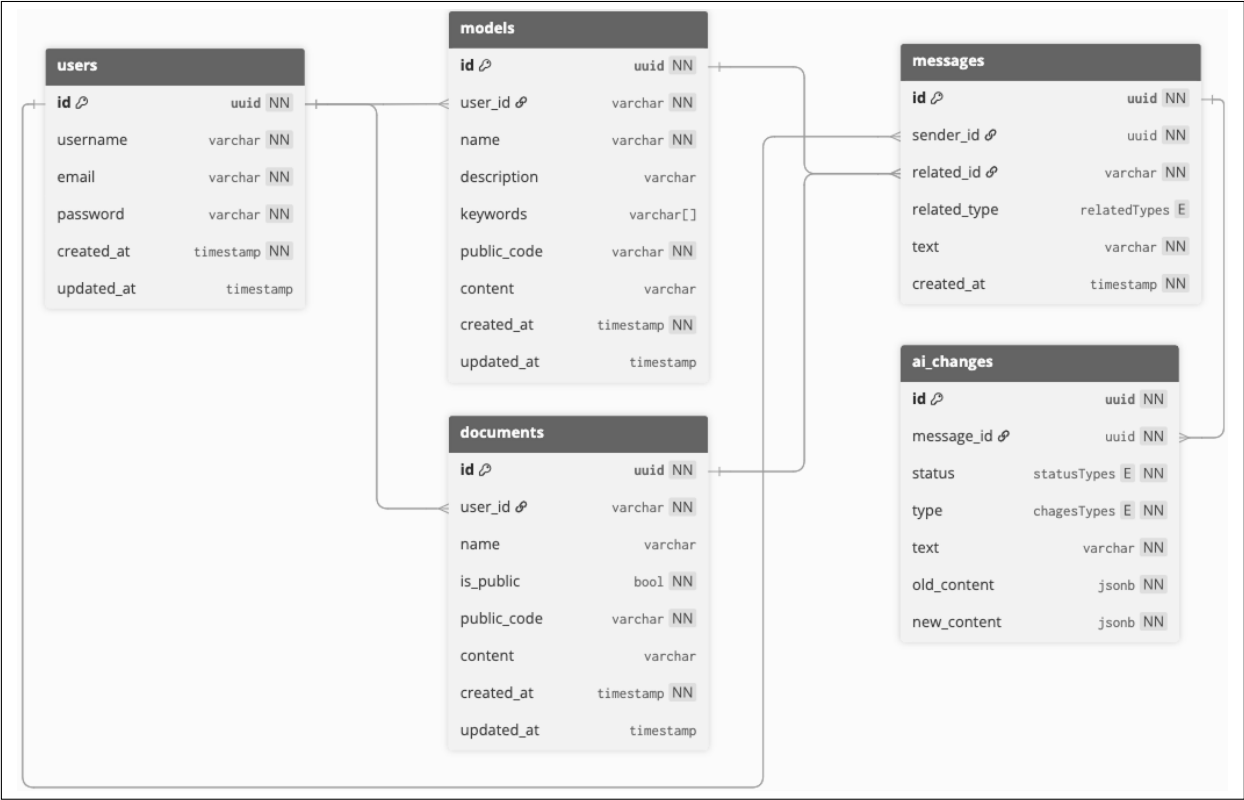


Figura 2: Modelagem do banco de dados do sistema.

Abaixo, apresenta-se a especificação de cada uma das tabelas que compõem o modelo de dados do sistema, com seus respectivos atributos e descrições.

Tabela 1: Descrição da entidade Users

Atributo	Descrição
id	Identificador único do usuário no formato Identificador Único Universal (UUID).
email	E-mail do usuário cadastrado.
password	Hash da senha do usuário cadastrado.
created_at	Timestamp do momento de criação do usuário.
updated_at	Timestamp do momento de atualização dos dados do usuário.

Tabela 2: Descrição da entidade Models

Atributo	Descrição
id [PK]	Identificador único do modelo no formato UUID.
user_id [FK]	Identificador do usuário que criou o modelo, chave estrangeira da relação com a tabela Users.
name	Nome do modelo.
keywords	Palavras chaves associadas ao modelo.
public_code	Código público do modelo usado para gerar o <i>link</i> de compartilhamento público do modelo.
content	Conteúdo do modelo com todos os componentes em formato de texto.
created_at	Timestamp do momento de criação do modelo.
updated_at	Timestamp do momento de atualização dos dados do modelo.

Tabela 3: Descrição da entidade Documents

Atributo	Descrição
id [PK]	Identificador único do documento no formato UUID.
user_id [FK]	Identificador do usuário que criou o documento, chave estrangeira da relação com a tabela Users.
name	Nome do documento.
is_public	Booleano que define se o documento poderá ser acessado por outras pessoas, ou somente pelo usuário criador do documento.
public_code	Código público do modelo usado para gerar o <i>link</i> de compartilhamento público do documento.
content	Conteúdo do documento com todos os componentes em formato de texto.
created_at	Timestamp do momento de criação do documento.
updated_at	Timestamp do momento de atualização dos dados do documento.

Tabela 4: Descrição da entidade Messages

Atributo	Descrição
id [PK]	Identificador único do chat no formato UUID.
sender_id [FK]	Identificador de quem enviou a mensagem, chave estrangeira da relação com a tabela Users.
related_id [FK]	Identificador do documento ou modelo que a mensagem pertence, chave estrangeira da relação com as tabelas Documents e Models.
related_type	Enumerador com os valores “model” e “document” que identificam se o chat está relacionado com a tabela Models ou a tabela Documents.
text	Texto da mensagem que foi enviada.
created_at	Timestamp do momento da criação da mensagem.

Tabela 5: Descrição da entidade AI_Changes

Atributo	Descrição
id [PK]	Identificador único do chat no formato UUID.
message_id [FK]	Identificador da mensagem que a alteração da IA foi feita, chave estrangeira da relação com a tabela Messages.
status	Enumerador com os valores “pending”, “approved” e “reproved” que identificam o estado atual da proposta de alteração.
type	Enumerador com os valores “add”, “update” e “remove” que identificam o tipo de alteração feita no documento ou modelo.
text	Texto com o resumo das alterações feitas pela IA.
old_content	Conteúdo do componente antes de ser alterado.
new_content	Conteúdo do componente após ser alterado.

3.2.2 Diagrama de classes UML

O diagrama de classes é uma forma de representar a estrutura de um sistema, disponibilizando suas classes, métodos, atributos e seus relacionamentos. É essencial para a compreensão da organização do sistema, facilitando a identificação de responsabilidades e dependências entre componentes. Na Figura 3, é apresentado o diagrama de classes UML simplificado do projeto. O diagrama completo está disponível no Apêndice C.

Fonte: Elaborado pelo autor.

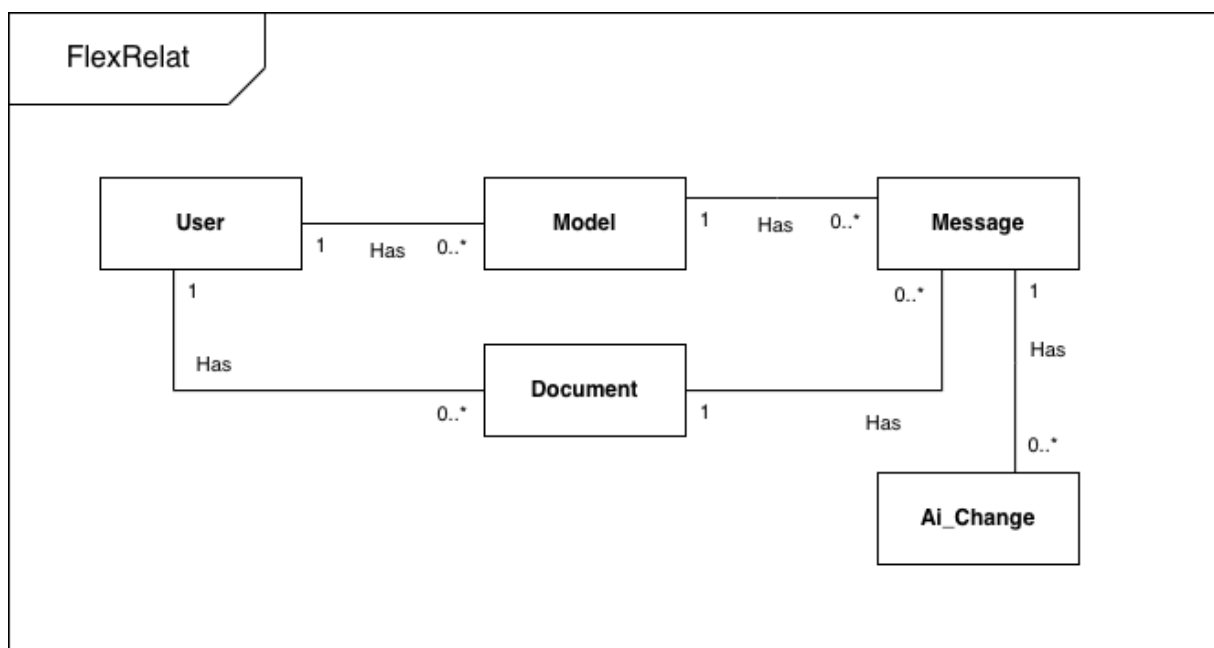


Figura 3: Diagrama de classes UML simplificado do sistema.

3.2.3 Diagrama de casos de uso

O diagrama de casos de uso é uma representação visual que descreve as principais funcionalidades oferecidas por um sistema e como os atores interagem com essas funcionalidades. Ele permite identificar os diferentes papéis envolvidos e os serviços que cada um pode acessar. Nas Figura 4 e Figura 5 é possível observar o diagrama de casos de uso do projeto, que detalha essas interações e fornece uma visão geral do comportamento esperado do sistema do ponto de vista dos usuários. Para a descrição completa dos casos de uso, consultar o Apêndice B.

Fonte: Elaborado pelo autor.

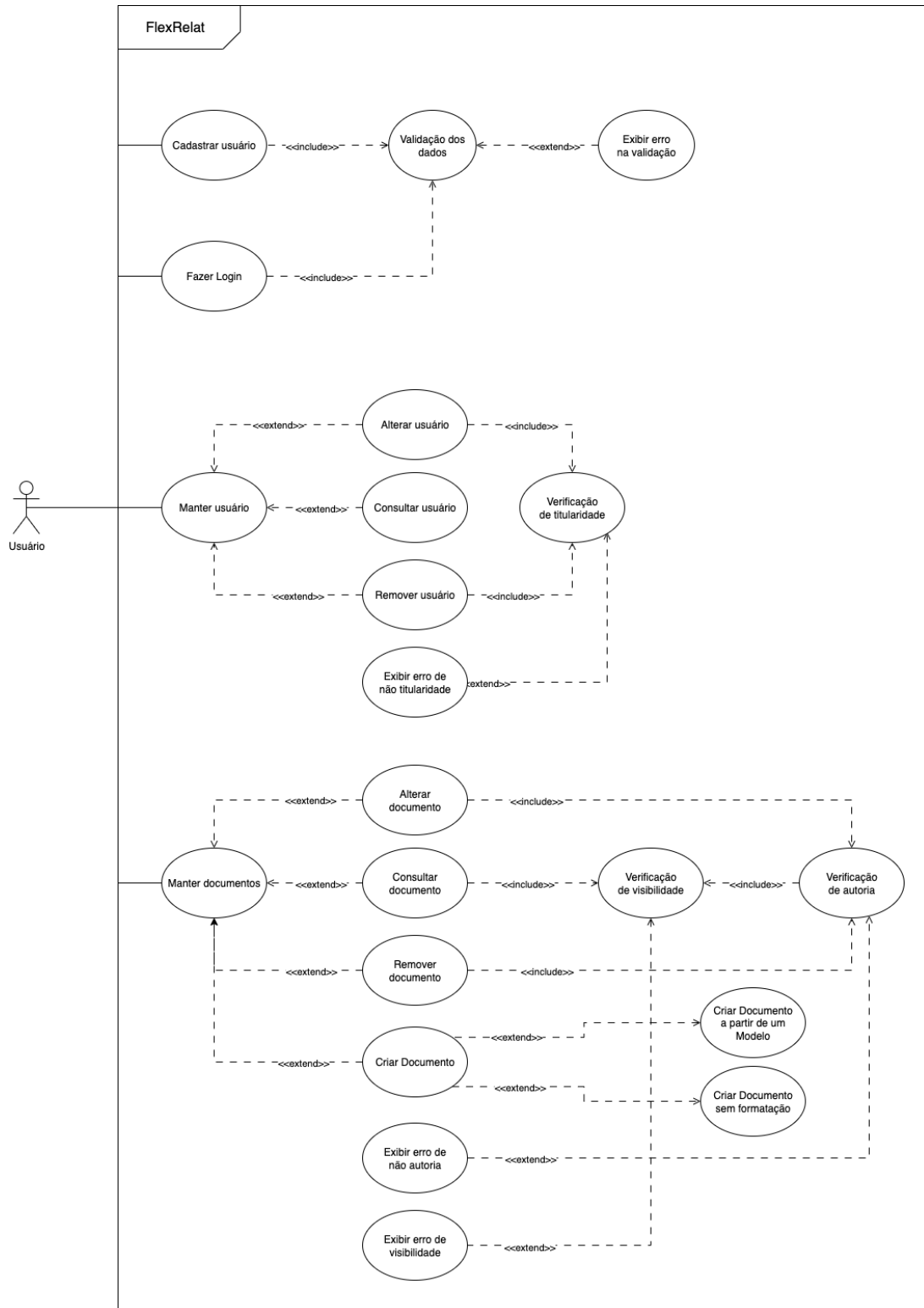


Figura 4: Primeira parte do diagrama de casos de uso do sistema.

Fonte: Elaborado pelo autor.

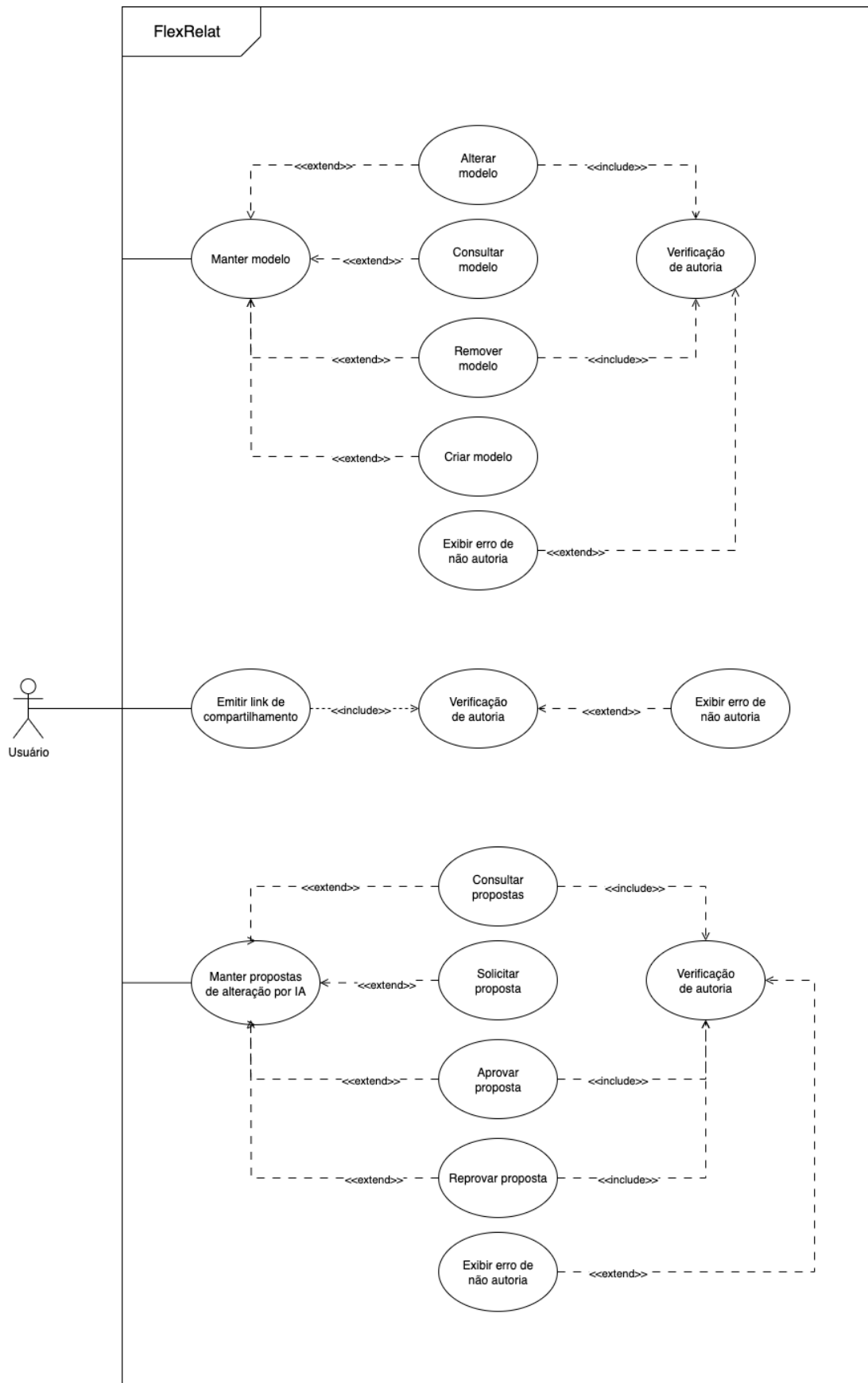


Figura 5: Segunda parte do diagrama de casos de uso do sistema.

3.3 Tecnologias utilizadas

Este tópico apresenta as principais tecnologias adotadas no desenvolvimento da plataforma, divididas por camadas para maior facilidade no entendimento das tecnologias por segmento do sistema. Cada ferramenta foi escolhida priorizando produtividade, escalabilidade, facilidade de manutenção e experiência prévia, sempre atentando-se às necessidades previstas nos requisitos do sistema, disponível no Apêndice A.

- **Front-end:** A camada de apresentação foi desenvolvida com Next.js na versão 15.3. A escolha se deu porque o framework oferece suporte nativo a rotas modernas, excelente performance e integração fluida com bibliotecas do mercado. Para a edição de documentos, foi implementada a biblioteca Tiptap, um *wrapper* sobre o ProseMirror, que permitiu a criação de um editor de texto que simula documentos em formato A4 e que poderá ser adaptado para múltiplos formatos. Para a construção dessa solução, foram utilizadas extensões próprias e da comunidade, que possuíam código aberto. Para comunicação com o backend, foram utilizados TanStack React Query e Axios, garantindo controle eficiente do estado assíncrono, cache, refetch e sincronização com a API. Na validação de dados, foi adotada a biblioteca Zod, pela sua integração nativa com TypeScript, permitindo validação segura e tipada.
- **Back-end:** A camada de aplicação foi implementada em NestJS. A escolha do NestJS está relacionada à sua estrutura modular baseada no padrão Modelo-Visão-Controlador (MVC), que organiza a aplicação em três camadas principais: o Model representa os dados e regras de negócio, o Controller recebe e trata as requisições, coordenando a lógica da aplicação, e a View é responsável pela interface com o usuário ou pela resposta retornada (como JSON em APIs). Essa separação torna o sistema mais modular, facilitando a manutenção, testes e evolução do código. A plataforma se comunica com a API da OpenAI para geração dos textos e componentes dos relatórios, integrando modelos de linguagem de ponta ao fluxo de criação do usuário. Para autenticação, foi adotado o Firebase Authentication, serviço de identidade externo, para facilitar o fluxo de autenticação e garantir segurança avançada sem precisar de implementação própria. Essa abordagem reduz a complexidade e aumenta a confiabilidade do gerenciamento de autenticação.
- **Banco de Dados:** O sistema utiliza o PostgreSQL como banco relacional. A escolha foi

motivada por sua robustez, confiabilidade e ampla adoção no mercado. Como a complexidade do modelo de dados do FlexRelat é baixa, optou-se por uma solução simples, com suporte a relacionamentos claros e bom desempenho em consultas estruturadas.

- **Gerenciamento:** O controle de versão é realizado com Git, utilizando a estratégia de ramificação Git Flow, que organiza o desenvolvimento em etapas bem definidas (como feature, release e hotfix) e facilita a entrega contínua. O repositório é hospedado no GitHub, permitindo colaboração e versionamento remoto.

3.4 Modelos de Inteligência Artificial e custos

A plataforma FlexRelat utiliza modelos de linguagem disponibilizados pela API da OpenAI para a execução das funcionalidades assistidas por Inteligência Artificial Generativa. Durante o desenvolvimento e a validação do sistema, foram empregados dois modelos distintos, selecionados de acordo com o tipo de operação realizada.

O modelo *gpt-5-mini* foi utilizado nas rotas da API de criação, atualização e remoção de conteúdo, por demandarem maior capacidade de compreensão de contexto e geração de texto em larga escala. Já o modelo *gpt-4.1-mini* foi adotado para pedidos pré-definidos de menor complexidade, como correção ortográfica, encurtamento e aprimoramento textual, em razão de seu tempo de resposta reduzido.

Os valores da Tabela 6 correspondem aos preços de utilização dos modelos, divulgados pela OpenAI [OpenAI, 2025], no período de desenvolvimento e validação do FlexRelat, sendo utilizados como referência para análise do custo e de viabilidade econômica. Ressalta-se que tais custos podem sofrer alterações ao longo do tempo, não sendo objetivo deste trabalho a realização de uma análise financeira aprofundada, precificação do sistema e a avaliação de custos em cenários de uso em larga escala com múltiplos usuários.

Tabela 6: Custos dos modelos da OpenAI utilizados no FlexRelat

Modelo	Entrada (US\$ / 1M tokens)	Entrada em cache (US\$ / 1M tokens)	Saída (US\$ / 1M tokens)
<i>gpt-5-mini</i>	0,25	0,025	2,00
<i>gpt-4.1-mini</i>	0,40	0,10	1,60

A escolha desses modelos buscou equilibrar qualidade das respostas, desempenho e custo monetário, considerando que a utilização de modelos mais avançados implica maior custo por

volume de tokens processados. Os testes foram realizados utilizando os planos padrão disponibilizados pela OpenAI, não sendo avaliado, neste trabalho, o impacto financeiro em cenários de uso intensivo ou com múltiplos usuários simultâneos.

Essa abordagem permite que, em trabalhos futuros, a plataforma seja ajustada para utilizar diferentes modelos, incluindo soluções de outras empresas, ou estratégias de roteamento dinâmico, de acordo com requisitos de custo, desempenho e qualidade exigidos pelo contexto de uso.

3.5 Interface de usuário

Essa sessão apresenta as interfaces de usuário da plataforma, que foi projetada com foco na simplicidade, acessibilidade e usabilidade, buscando atender usuários com diferentes níveis de afinidade com tecnologia. A proposta visual adota uma estrutura limpa e intuitiva. As interfaces aqui apresentadas correspondem às telas construídas até o presente momento, sendo importante destacar que o sistema ainda se encontra em fase de desenvolvimento.

3.5.1 Login

A Figura 6 apresenta a tela de login da plataforma. Nela, o usuário informa suas credenciais (e-mail e senha) para se autenticar e acessar as funcionalidades privadas do sistema. Após a validação, a aplicação estabelece a sessão do usuário e redireciona para a área principal, onde é possível gerenciar documentos e modelos. Além disso, a tela concentra mensagens de erro em casos de credenciais inválidas, garantindo um fluxo inicial simples e seguro.

Fonte: Elaborado pelo autor.

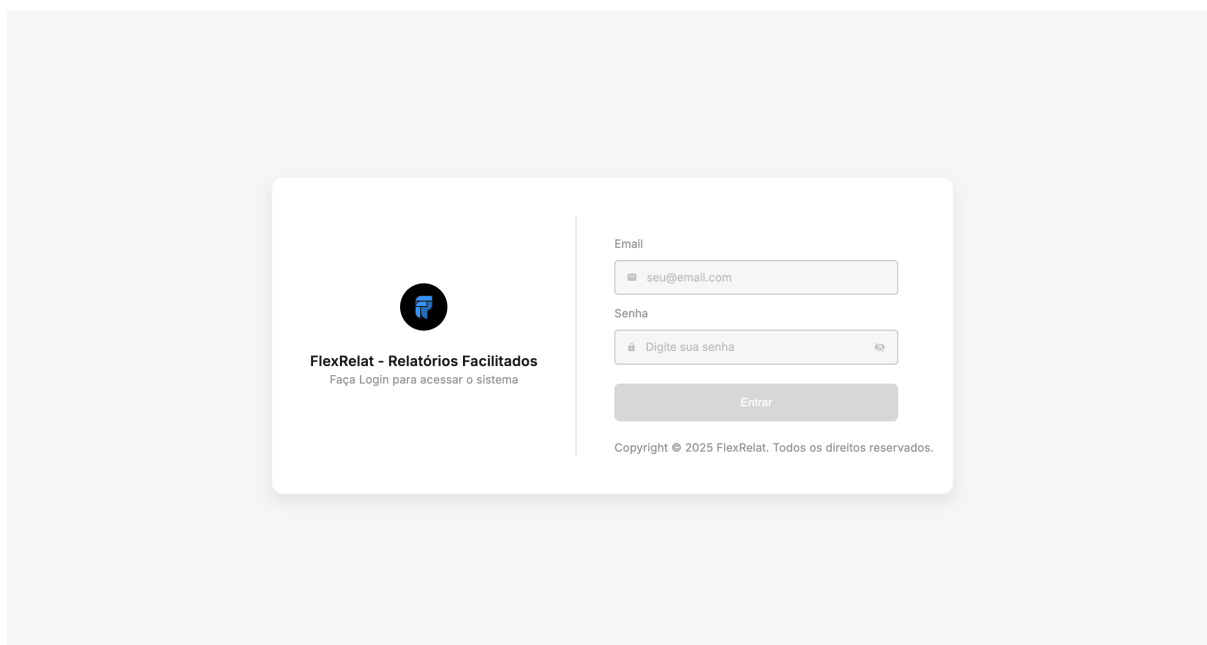


Figura 6: Tela de login.

3.5.2 Documentos

A Figura 7 apresenta a tela de documentos, acessível assim que é feito o login e também pela abertura ao clicar na barra de navegação da esquerda. Esta tela é o local de visualização dos documentos criados pelo usuário, funcionando como o ambiente principal onde o usuário pode criar um novo documento e abrir um documento para edição.

Fonte: Elaborado pelo autor.

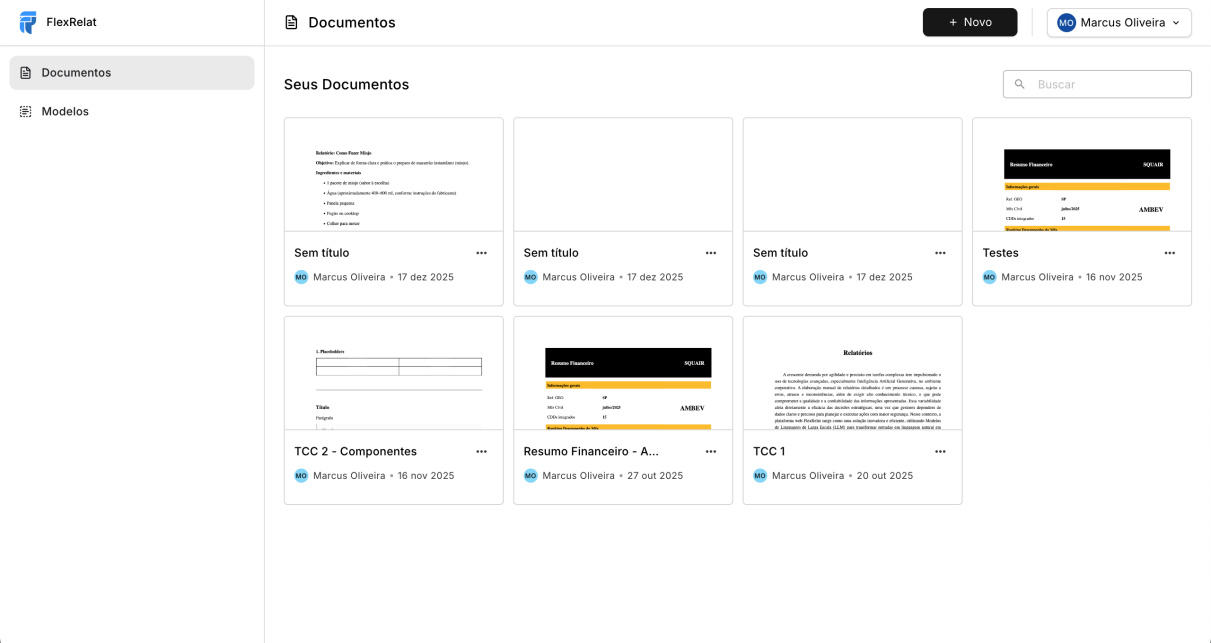


Figura 7: Tela de documentos.

3.5.3 Modelos

Fonte: Elaborado pelo autor.

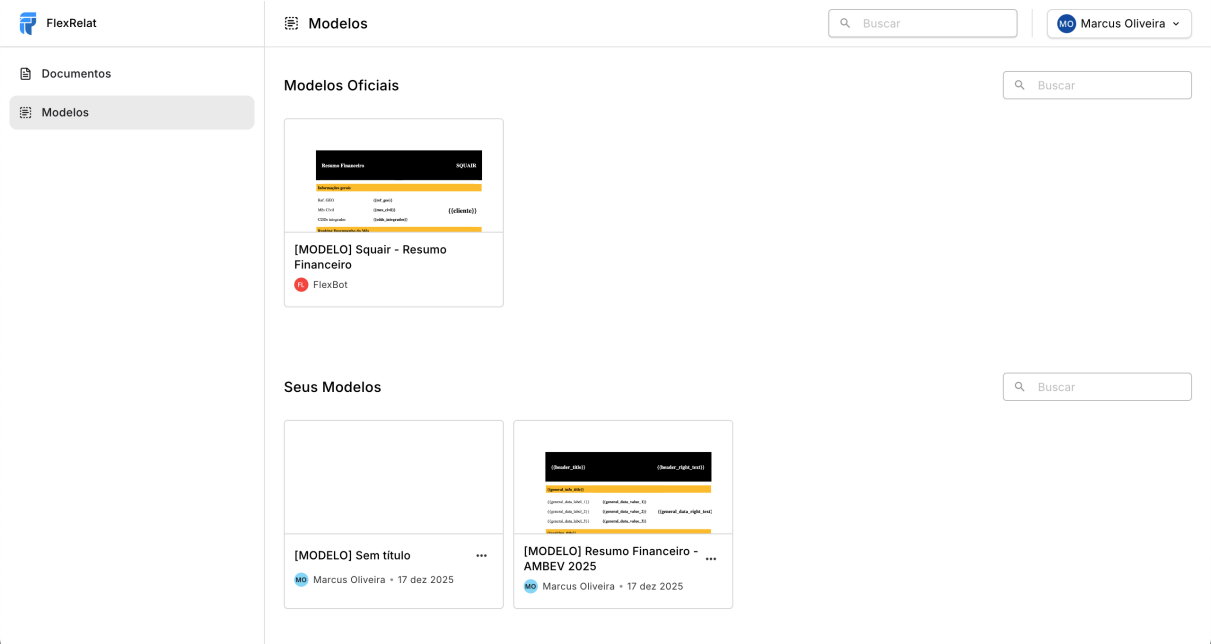


Figura 8: Tela de modelos.

A Figura 8 apresenta a tela de modelos, acessível ao clicar em “Modelos” na barra de navegação lateral. Essa interface reúne os modelos de relatórios disponíveis na plataforma, incluindo modelos oficiais e modelos criados pelo próprio usuário, e funciona como um catálogo para organização e reutilização de estruturas de relatórios.

3.5.4 Edição de documento

A Figura 9 apresenta a tela de edição de um documento, acessível tanto pela abertura direta na página de documentos quanto por meio de um link de compartilhamento. Esta tela é o núcleo central da aplicação, funcionando como o ambiente principal onde estão disponíveis as funcionalidades essenciais da plataforma. Nela, o usuário pode realizar diversas operações fundamentais para o gerenciamento, modificação e aprimoramento dos documentos, com a opção de utilizar a Inteligência Artificial do sistema para aumentar a eficiência do processo de edição.

Fonte: Elaborado pelo autor.

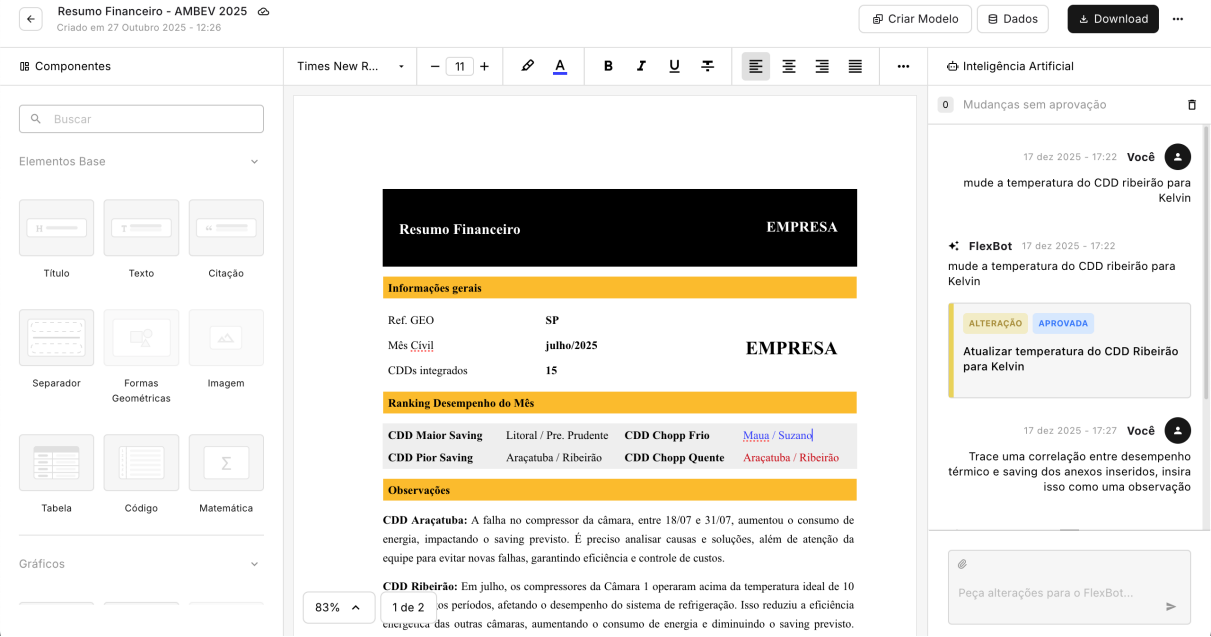


Figura 9: Tela de edição de documento.

3.5.5 Edição de modelo

A Figura 10 apresenta a tela de edição de um modelo no modo de visualização, que ocorre quando o modelo não pertence ao usuário autenticado. Nesse modo, o conteúdo do modelo pode ser consultado e usado para criar um documento a partir dele, porém, as ações de edição

ficam indisponíveis, garantindo a integridade dos modelos públicos e de terceiros. Quando o modelo pertence ao usuário, a interface funciona de forma semelhante à edição de documentos: o conteúdo é exibido em páginas no formato A4 e pode ser modificado com as mesmas ferramentas de formatação, organização e gerenciamento de componentes, além do suporte da Inteligência Artificial para auxiliar na criação e refinamento do conteúdo.

Fonte: Elaborado pelo autor.

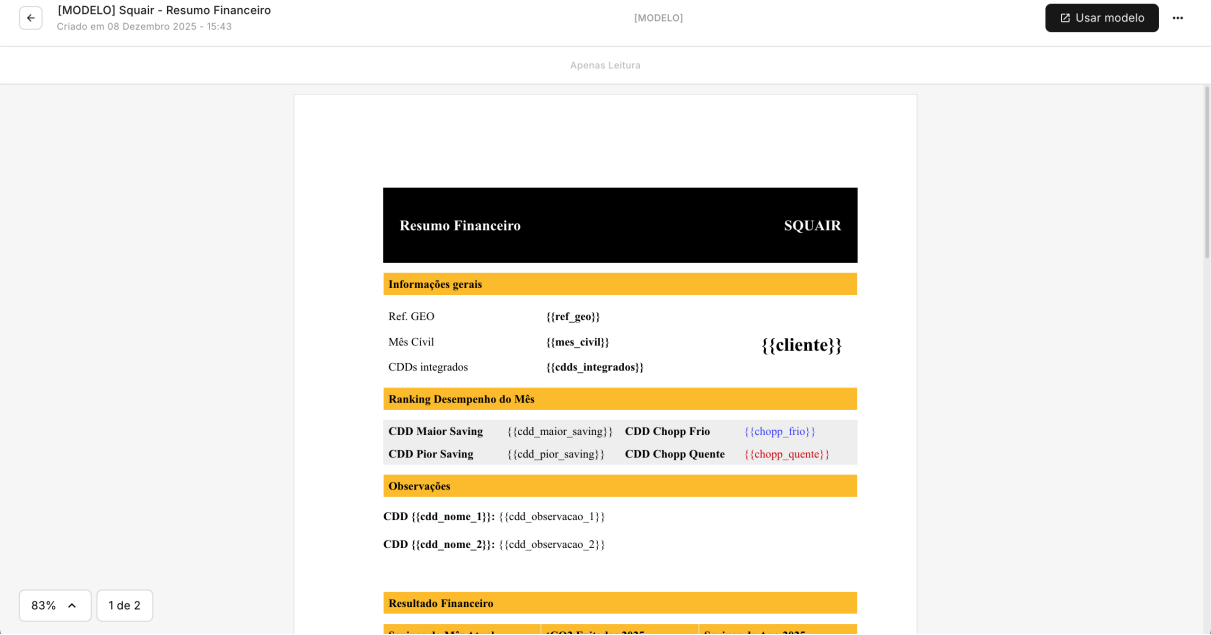


Figura 10: Tela de edição de modelo - Modo Visualização.

3.5.6 Sugestão de alteração por chat

Na Figura 11 pode-se observar uma sugestão de edição do conteúdo de um componente de Título, localizado na tela de edição de um documento. A sugestão é exibida ao clicar em uma sugestão de alteração, criação ou remoção, gerada automaticamente a partir de uma entrada ou várias entradas de texto fornecidas pelo usuário no chat, localizado na aba "Inteligência Artificial".

Fonte: Elaborado pelo autor.

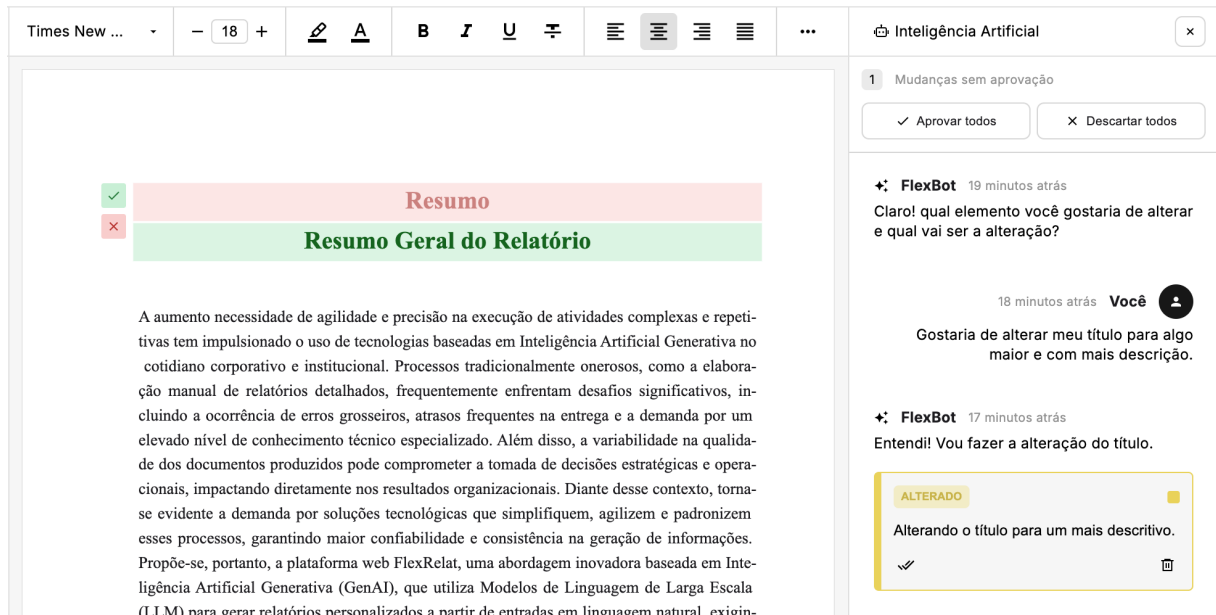


Figura 11: Sugestão de alteração por chat.

3.5.7 Menu de opções de alterações por IA

O menu de opções de alteração, apresentado na Figura 12, é um menu flutuante que possibilita edições rápidas em um componente, fica localizado na tela de edição de um documento e é ativado quando o usuário seleciona um componente pressionando o botão esquerdo do mouse em cima dele + COMMAND (Macbook) ou CTRL (Windows ou Linux). Essas opções são personalizadas para cada tipo de componente.

Fonte: Elaborado pelo autor.

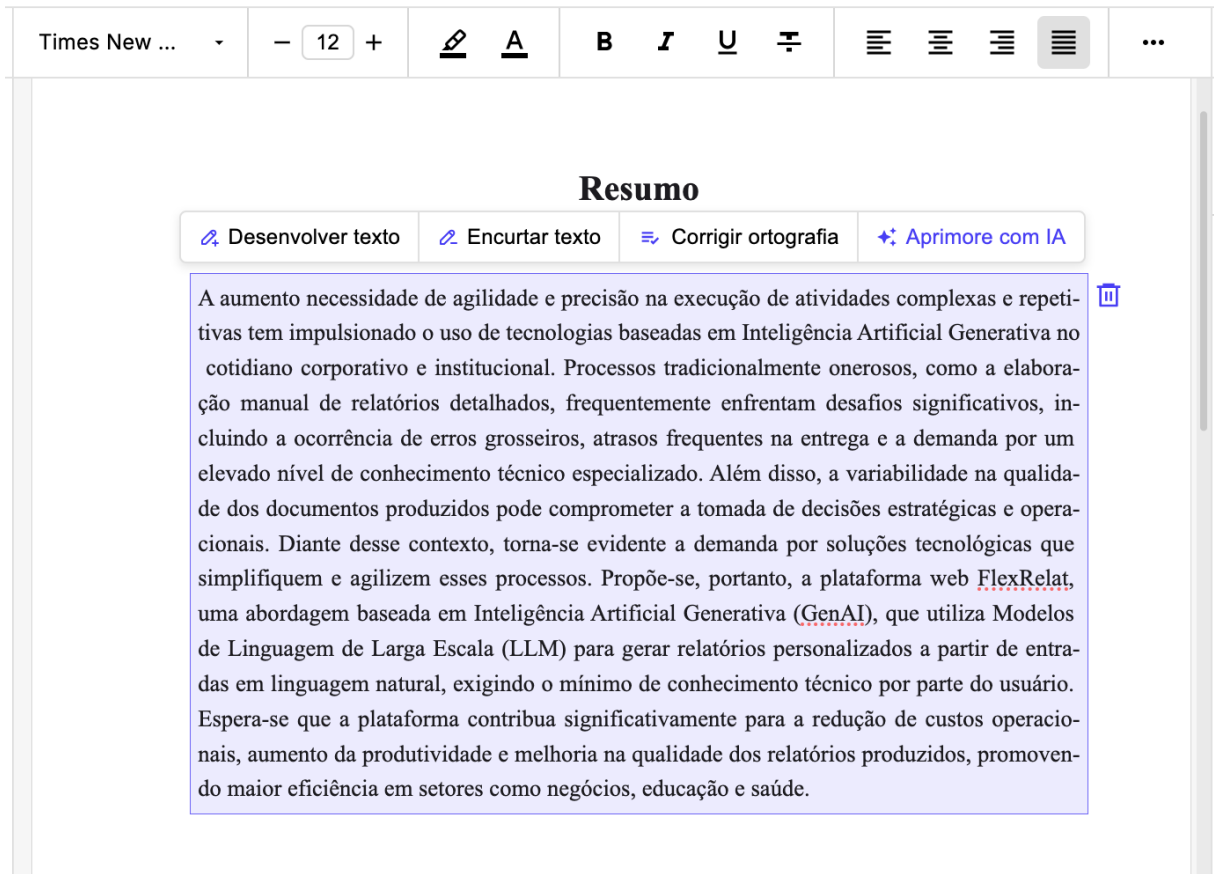


Figura 12: Menu de opções de alterações por IA.

3.5.8 Menu de alteração de zoom e contagem de páginas

A Figura 13 representa o menu de alteração de zoom e de contagem de páginas. Os menus são flutuantes, podem ser acessados em um documento ou modelo e servem para melhorar a navegação dentro do documento.

Fonte: Elaborado pelo autor.

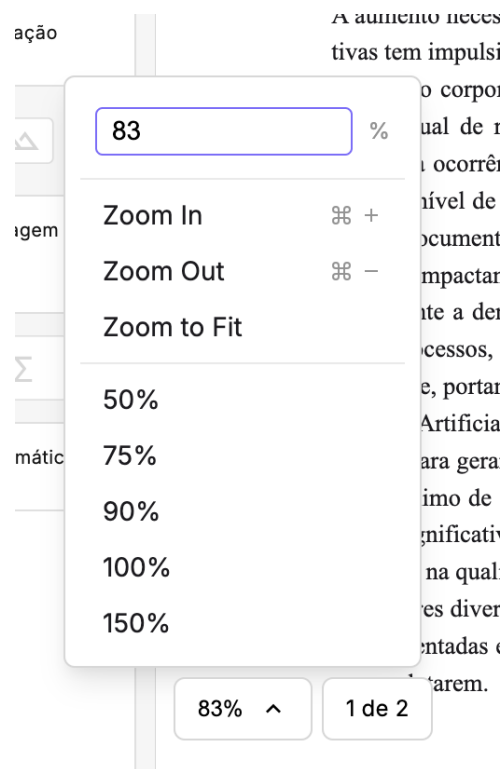


Figura 13: Menu de alteração de zoom.

3.5.9 Modal de alteração do gráfico

A Figura 14 representa o modal de manipulação das informações apresentadas do gráfico, como a inserção e alterações dos dados e modificações visuais dos dados e de estilo do gráfico. esse modal pode ser acessado ao clicar em um gráfico com o botão esquerdo do mouse.

Fonte: Elaborado pelo autor.

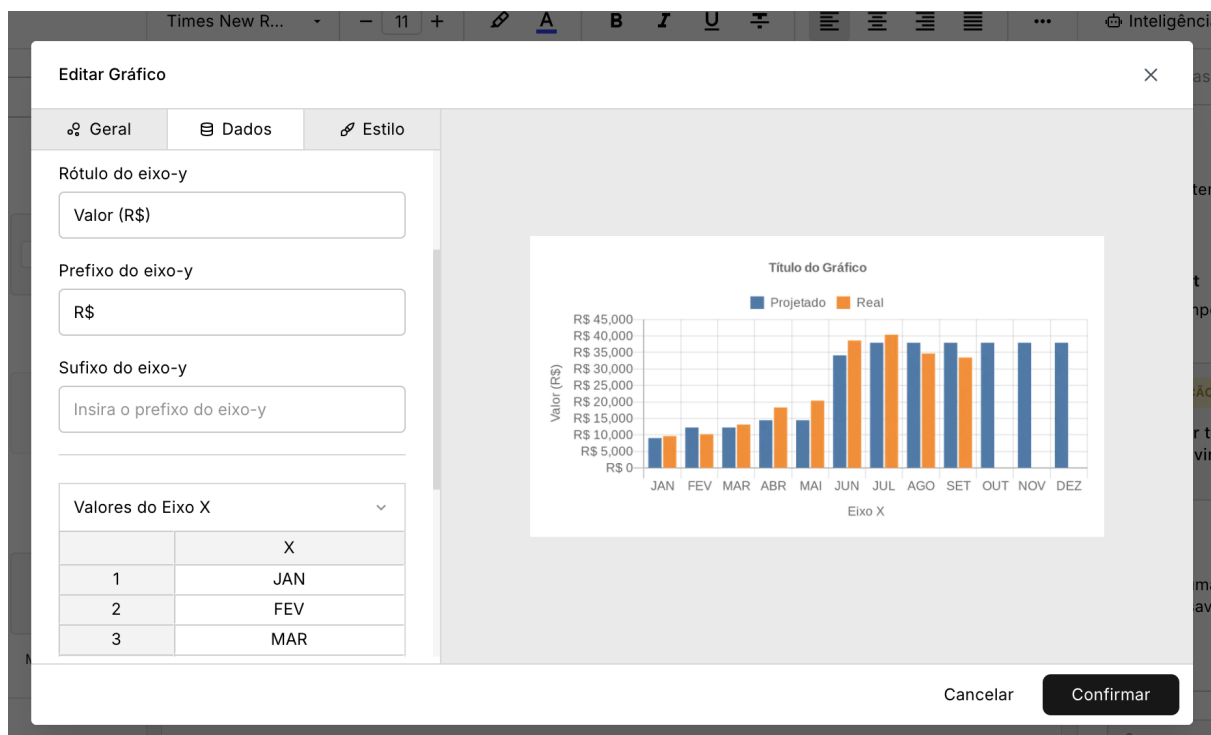


Figura 14: Modal de alteração de um gráfico.

3.5.10 Modal de dados

A Figura 15 representa o modal de inserção de dados de contexto do documento. Nesse modal, o usuário pode adicionar novos conjuntos de dados, editar valores existentes e organizar a estrutura das informações que serão consumidas pelo FlexBot, IA de assistência a geração e edição dos conteúdos do relatório, como tabelas, gráficos e textos, garantindo que o documento final esteja alinhado ao contexto dos dados disponíveis para o relatório. O modal pode ser acessado ao clicar no botão de "Dados" no menu superior, da área de edição do documento, abrindo uma interface dedicada para atualização das informações de contexto.

Fonte: Elaborado pelo autor.

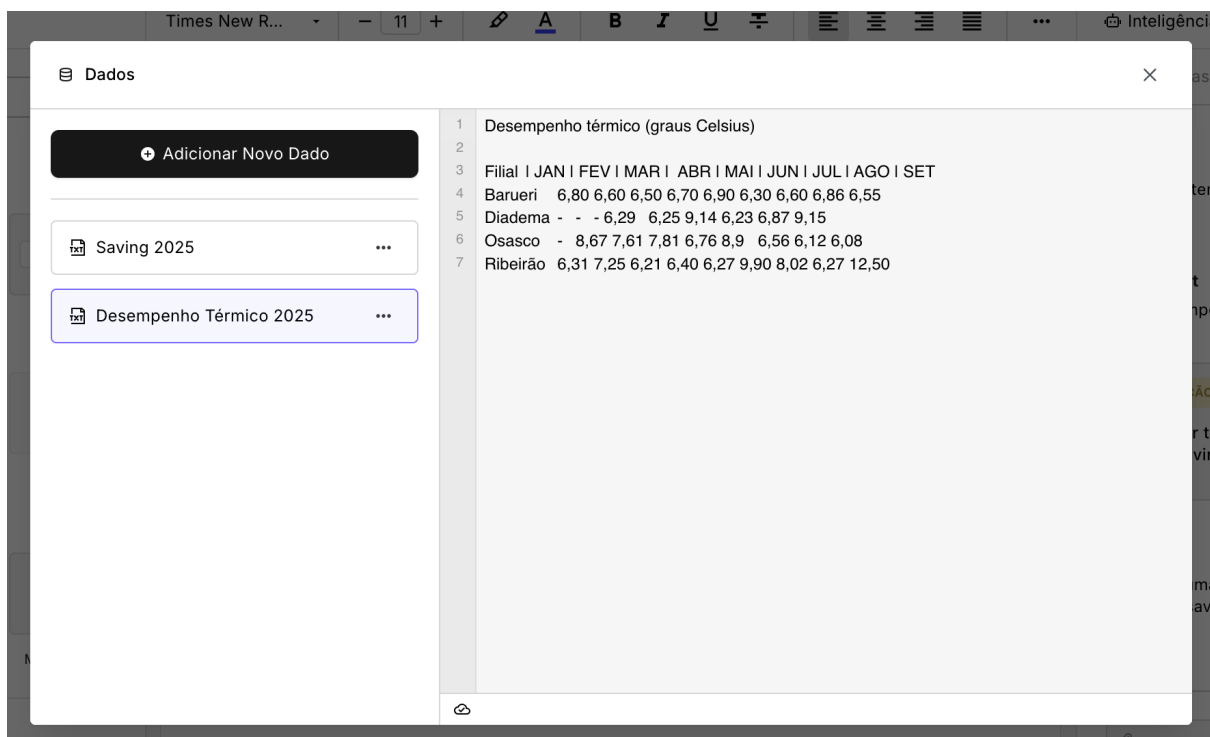


Figura 15: Modal de dados.

3.5.11 Modal de criação de modelo

A Figura 16 representa o modal de criação de um modelo a partir de um conteúdo base. Nesse fluxo, o usuário pode escolher entre duas opções: **“Manter o conteúdo original”**, que cria o modelo preservando integralmente o conteúdo fornecido, e **“Transformar conteúdo com IA”**, que aciona o FlexBot para reestruturar e adaptar o texto automaticamente para o formato de modelo (com variáveis no lugar dos dados, mas mantendo a estrutura e estilo), propondo ajustes de organização, seções e padronização do documento.

Esse modal é exibido após clicar no botão "Criar Modelo", localizado no menu superior da tela de edição de um documento, permitindo que o usuário defina rapidamente se deseja reutilizar o conteúdo como está ou otimizar sua estrutura com auxílio de Inteligência Artificial.

Fonte: Elaborado pelo autor.



Figura 16: Modal de dados.

Capítulo 4

Validação

Este capítulo tem como objetivo apresentar a metodologia e os resultados dos testes realizados para validação da plataforma desenvolvida. Para isso, serão apresentadas métricas de natureza quantitativas e qualitativas, obtidas por meio da execução de testes das funcionalidades e da coleta de *feedback* em questionários respondidos por usuários de teste de diversas empresas.

4.1 Testes Funcionais

Para validar a usabilidade e comportamento funcional do FlexRelat, foram realizados testes focados na execução correta das funcionalidades que compõem o núcleo central da aplicação. Esses testes tiveram como objetivo avaliar métricas relacionadas ao tempo de execução das operações, taxas de sucesso e falha, severidade das falhas e cobertura dos requisitos funcionais.

Para classificar o impacto das falhas, a severidade foi dividida em três níveis distintos. A severidade mínima engloba falhas de caráter cosmético, facilmente corrigíveis ou pequenas inconsistências na interface que não interferem na capacidade de uso, como erros de grafia ou desalinhamentos visuais. O nível de severidade médio refere-se a problemas que afetam funcionalidades específicas ou dificultam a experiência do usuário, mas que permitem a conclusão da tarefa principal por meio de caminhos alternativos. Já a severidade alta caracteriza erros críticos que resultam no bloqueio total de funções vitais da plataforma, impedindo que o usuário atinja o objetivo central da aplicação.

Ressalta-se que não houve a intenção de mensurar desempenho sob concorrência ou múltiplos usuários simultâneos, uma vez que os testes concentraram-se exclusivamente na verificação do funcionamento adequado da plataforma em um ambiente controlado.

Para manter o controle sobre o ambiente de teste, foi utilizado um ambiente de desenvolvimento local, permitindo maior controle sobre as variáveis externas que poderiam influenciar os resultados. A escolha desse ambiente teve como objetivo isolar o comportamento do sistema, evitando interferências relacionadas à latência de rede, variações de conectividade com a inter-

net ou limitações de infraestrutura externa.

Dessa forma, os testes concentraram-se exclusivamente na validação das funcionalidades internas do FlexRelat, assegurando que os resultados obtidos refletissem o desempenho lógico e funcional da plataforma. Essa abordagem possibilitou uma análise mais precisa das métricas definidas, bem como a identificação de falhas relacionadas à implementação, fluxo de uso e atendimento aos requisitos funcionais estabelecidos.

O escopo dos testes funcionais foi delimitado às funcionalidades consideradas mais críticas e com maior potencial de causar falhas na aplicação. Funcionalidades mais simples e de baixo risco, como criação, exclusão ou renomeação de documentos e modelos, aplicação de modelos e autenticação de usuários, não foram incluídas nos testes, uma vez que apresentam menor complexidade, baixo potencial de falhas e reduzida probabilidade de comprometer o funcionamento do núcleo do sistema. Dessa forma, os testes concentraram-se em funcionalidades como a edição de documentos e modelos, bem como todas as funcionalidades relacionadas ao uso de Inteligência Artificial, garantindo a validação dos componentes mais sensíveis e críticos do FlexRelat.

4.1.1 Teste de edição de documentos e modelos

A Tabela 7 apresenta os principais casos de teste funcionais definidos para a validação das funcionalidades críticas de edição de documentos e modelos no FlexRelat. Esses casos de teste foram elaborados com o objetivo de verificar o correto funcionamento do mecanismo de salvamento, a persistência dos dados e a consistência entre *frontend* e *backend*, assegurando que as operações mais sensíveis da plataforma atendam aos requisitos funcionais estabelecidos. Cada caso de teste foi executado vinte vezes, em um documento e um modelo com sessenta páginas, e os resultados consolidados dessas execuções, juntamente com as métricas coletadas, são apresentados na Tabela 8.

Tabela 7: Casos de testes para edição de documentos e modelos.

ID	Funcionalidade	Descrição do caso de teste	Entrada/Ação
CT-01	Edição de documento	Salvamento após edição	Editar texto e aguardar o armazenamento
CT-02	Edição de documento	Persistência após recarregar	Editar e recarregar a página
CT-03	Edição de modelo	Salvamento após edição	Editar modelo e aguardar o armazenamento
CT-04	Edição de modelo	Persistência do modelo	Editar e recarregar a aplicação

Tabela 8: Resultados dos testes para edição de documentos e modelos.

ID	Tempo mín.	Tempo méd.	Tempo máx.	Sucesso	Severidade falhas	Requisitos atendidos
CT-01	636 ms	844 ms	1,06 s	95%	Mínima	100%
CT-02	310 ms	381 ms	478 ms	100%	—	100%
CT-03	640 ms	832 ms	1,28 s	100%	—	100%
CT-04	331 ms	460 ms	487 ms	100%	—	100%

Podemos observar alguns pontos relevantes a partir dos resultados apresentados:

1. Desempenho consistente: O FlexRelat apresentou tempos de execução mínimos, médios e máximos estáveis entre os casos de teste avaliados, indicando um comportamento previsível do mecanismo de edição e salvamento, mesmo diante de múltiplas operações consecutivas.
2. Alta taxa de sucesso: Todos os testes executados obtiveram altas taxas de sucesso, com erros pontuais e que são notificados ao usuário, evidenciando a robustez das funcionalidades críticas avaliadas e o controle de dados sobre as falhas durante os cenários testados.
3. Conformidade com os requisitos: Os resultados demonstram que 100% dos requisitos funcionais associados aos casos de teste foram atendidos, confirmando que as funcionalidades analisadas estão alinhadas às especificações definidas para a plataforma.

4. Ausência de falhas críticas: Não foram identificadas falhas com severidade classificada como média, alta ou crítica, reforçando a estabilidade das operações centrais do FlexRelat no ambiente de testes adotado.

4.1.2 Testes das funcionalidades com uso de Inteligência Artificial

As funcionalidades que utilizam Inteligência Artificial representam um dos componentes mais críticos do FlexRelat, uma vez que envolvem processamento externo, interpretação de linguagem natural e geração dinâmica de conteúdo. Dessa forma, foram definidos testes funcionais específicos para validar os pedidos de criação, atualização e remoção de conteúdo assistidos pelo FlexBot, bem como as opções de pedidos pré-definidos disponibilizados pela plataforma.

Os testes relacionados às operações de criação, atualização, remoção de conteúdo com IA foram realizados utilizando o modelo *gpt-5-mini* da OpenAI. Já os testes referentes aos pedidos pré-definidos, "desenvolver texto", "encurtar texto", "corrigir ortografia" e "aprimorar conteúdo", foram executados com o modelo *gpt-4.1-mini*, escolhido por apresentar respostas mais rápidas e adequadas para tarefas pontuais de refinamento textual.

Além disso, os testes foram separados entre cenários com e sem dados anexos, a fim de avaliar o impacto da contextualização adicional no comportamento do modelo e na consistência dos resultados gerados. Cada caso de teste foi executado vinte vezes e também apenas em cima de componentes de texto, permitindo a coleta de métricas representativas sobre o comportamento do modelo e a consistência dos resultados gerados. A Tabela 9 apresenta os principais casos de teste definidos para essas funcionalidades, enquanto a Tabela 10 reúne as métricas coletadas durante sua execução.

Tabela 9: Casos de teste das funcionalidades com uso de IA

ID	Funcionalidade	Descrição do caso de teste	Entrada/Ação
CT-01	Criação com IA	Criação de conteúdo sem dados anexos	Enviar prompt de criação
CT-02	Criação com IA	Criação de conteúdo com dados anexos	Enviar prompt com anexos
CT-03	Atualização com IA	Atualização de conteúdo sem dados anexos	Enviar prompt de atualização
CT-04	Atualização com IA	Atualização de conteúdo com dados anexos	Enviar prompt com anexos
CT-05	Remoção com IA	Remoção de conteúdo sem dados anexos	Enviar prompt de remoção
CT-06	Pedido pré-definido	Desenvolver texto	Acionar opção de desenvolvimento
CT-07	Pedido pré-definido	Encurtar texto	Acionar opção de encurtamento
CT-08	Pedido pré-definido	Correção ortográfica	Acionar opção de correção
CT-09	Pedido pré-definido	Aprimoramento de texto	Acionar opção de aprimoramento

Tabela 10: Resultados dos testes das funcionalidades com IA

ID	Tempo mín.	Tempo méd.	Tempo máx.	Sucesso	Severidade falhas	Requisitos atendidos
CT-01	19,84 s	22,67 s	27,24 s	100%	—	100%
CT-02	25,31 s	33,8 s	45,36 s	88,23%	Mínima	100%
CT-03	27,64 s	43,83 s	79,25 s	84,6%	Mínima	100%
CT-04	22,33 s	37,37 s ms	65 s	85,7%	Mínima	100%
CT-05	12,28 s	26,55 s	48,90 s	100%	—	100%
CT-06	2,43 s	4,0 s	6,85 s	100%	—	100%
CT-07	2,81 s	3,08 s	3,23 s	100%	—	100%
CT-08	2,20 s	2,71 s	3,01 s	100%	—	100%
CT-09	2,48 s	2,75 s	5,36 s	100%	—	100%

Podemos observar alguns pontos relevantes a partir dos resultados apresentados nas tabelas:

1. Alta variação no tempo de execução: Os casos de teste CT-01, CT-02, CT-03, CT-04 e CT-05 apresentaram tempos de execução superiores aos demais, esse comportamento acontece por que essas operações utilizam todo o conteúdo do arquivo como entrada para a IA. Dessa forma, o volume de contexto processado é maior, impactando diretamente no tempo necessário para geração das respostas.
2. Estabilidade dos resultados: Apesar do aumento do tempo de execução em cenários com maior contexto, os tempos mínimo, médio e máximo mantiveram-se, em geral, consistentes entre as execuções, indicando estabilidade e previsibilidade do sistema. Variações mais elevadas entre os tempos mínimo e máximo foram observadas em casos como o CT-03 e o CT-04, decorrentes do processamento do modelo de IA generativa da OpenAI. Ainda assim, os tempos médios permaneceram mais próximos dos valores mínimos, reforçando a consistência do desempenho obtido.
3. Qualidade e cobertura funcional: As elevadas taxas de sucesso e os altos percentuais de requisitos atendidos indicam que o sistema manteve desempenho adequado mesmo em cenários de maior complexidade, sem a ocorrência de falhas críticas que comprometessem as funcionalidades avaliadas. As falhas observadas nos casos CT-02, CT-03 e CT-04 apresentaram severidade mínima, uma vez que se limitaram ao não atendimento pontual de solicitações de criação ou alteração, sem impactar o fluxo principal de uso do sistema. Ademais, tais ocorrências puderam ser facilmente contornadas por meio de novas solicitações por parte do usuário, evidenciando a maturidade funcional da aplicação no contexto analisado.
4. Impacto controlado do contexto adicional: A separação entre cenários com e sem dados anexos evidenciou que o aumento do contexto influencia o tempo de execução, mas mantendo as taxas de sucesso acima dos 80%, e a cobertura dos requisitos completa, indicando boa robustez do modelo frente a variações no volume de entrada.

4.2 Coleta de Feedbacks

Com o objetivo de complementar a validação funcional do FlexRelat, foram aplicados questionários para a coleta de feedbacks, de natureza quantitativa e qualitativa, junto a usuários de

teste. Essa etapa teve como foco avaliar aspectos relacionados à facilidade de uso, utilidade das funcionalidades, intuitividade do sistema, satisfação geral e percepção de impacto na produtividade dos usuários. A análise dessas informações permite identificar pontos fortes da plataforma, bem como oportunidades de melhoria, servindo de base para ajustes na versão final do sistema e para direcionar trabalhos futuros.

A pesquisa contou com a participação de cinco usuários provenientes de diferentes contextos organizacionais, proporcionando uma visão diversificada sobre o uso do FlexRelat em cenários reais. Dois participantes pertenciam à empresa SQUAIR, atuante no setor de monitoramento IoT. Os demais participantes eram oriundos da CEDAE (Companhia Estadual de Águas e Esgoto do Rio de Janeiro), da PipeImob Tecnologia, empresa do setor imobiliário, e do Zé Delivery, plataforma de entrega de bebidas pertencente ao grupo AmBev. Essa diversidade de perfis contribuiu para uma avaliação mais abrangente da aplicabilidade da plataforma em distintos domínios de negócio. A Tabela 11 apresenta as perguntas aplicadas no questionário.

Tabela 11: Perguntas do questionário de coleta de feedbacks

ID	Pergunta
Q-01	Em uma escala de 1 a 5, o quanto o serviço é útil para você?
Q-02	Quais funcionalidades do FlexRelat você acha útil?
Q-03	Quais funcionalidades do FlexRelat você NÃO acha útil?
Q-04	Quais funcionalidades novas ajudariam no seu trabalho?
Q-05	Qual o nível de facilidade que você teve com o sistema?
Q-06	Quais foram as dificuldades que você teve com o sistema?
Q-07	Quanto você acredita que o sistema é intuitivo?
Q-08	O que você acredita que pode ser melhorado?
Q-09	Como você avalia o sistema de exportação do relatório em PDF?
Q-10	O sistema de download em PDF apresentou alguma falha?
Q-11	Quais falhas ele apresentou? Qual a frequência?
Q-12	Como você avalia o sistema de IA da plataforma?
Q-13	O sistema de IA Generativo apresentou alguma falha?
Q-14	Quais falhas ele apresentou? Qual a frequência?
Q-15	Você utilizaria a plataforma novamente?
Q-16	Para quais tipos de relatórios você utilizaria?
Q-17	Por que você não utilizaria a plataforma?
Q-18	A plataforma melhoraria sua produtividade?
Q-19	Por que ela melhoraria sua produtividade?
Q-20	Qual sua satisfação geral com o sistema?

4.2.1 Análise dos feedbacks coletados

Os resultados das avaliação dos usuários indicam utilidade percebida do FlexRelat. A satisfação geral do sistema alcançou a média de 9 pontos de 10. Quando perguntados sobre a utilidade do serviço, dos cinco participantes, três atribuíram nota 5 e dois atribuíram nota 4, demonstrando uma avaliação positiva.

Entre as funcionalidades mais úteis, destacaram-se o uso da Inteligência Artificial para automação de tarefas repetitivas, a edição direta de conteúdos, a criação de *templates* e a geração facilitada de documentos técnicos. Também foram mencionadas a simplicidade na edição de

textos sem a necessidade de *prompts* complexos e a conversão automática de arquivos. Não foram identificadas funcionalidades consideradas não úteis pelos participantes.

Quanto às sugestões de novas funcionalidades, os usuários apontaram principalmente demandas relacionadas à automação e integração, como conexão com bancos de dados e planilhas, geração automática de relatórios a partir de tabelas de dados e envio periódico de documentos. Também foram sugeridos recursos como *autocomplete* de texto, busca semântica, novos formatos de exportação, mais tipos de gráficos e histórico de *prompts*.

A usabilidade do sistema foi bem avaliada, com três participantes atribuindo nota 4 e dois nota 5. As dificuldades relatadas estiveram relacionadas principalmente à compreensão inicial da estrutura por seções, à definição de *templates* por usuários não técnicos e a inconsistências pontuais na interação com a IA, mas não foram consideradas impeditivas.

A avaliação da intuitividade também apresentou resultados positivos, com três notas 5 e duas notas 4. As sugestões de melhoria concentraram-se em aprimoramentos de usabilidade, como melhor identificação visual das seções, maior clareza na definição de variáveis e suporte explicativo às funcionalidades, além da possibilidade de sobrepor modelos a documentos em desenvolvimento.

O sistema de exportação em PDF foi avaliado de forma satisfatória, com duas notas 5, uma nota 4 e duas notas 3. Embora a maioria não tenha identificado falhas, alguns usuários relataram problemas pontuais, como corte de texto e inconsistências de formatação em *downloads* repetidos, indicando necessidade de novos testes e ajustes da funcionalidade.

A Inteligência Artificial do FlexRelat recebeu avaliação positiva, com quatro notas 4 e uma nota 5. Apesar disso, três usuários relataram falhas ocasionais, como duplicação de conteúdo, ausência de alterações solicitadas, criação incorreta de citações e lentidão nas respostas, sugerindo a necessidade de refinamentos no controle de contexto e na consistência das respostas.

Todos os participantes afirmaram que utilizariam a plataforma novamente. O uso foi considerado adequado para relatórios técnicos e corporativos, incluindo análises energéticas, relatórios operacionais, documentação técnica e acompanhamento de desempenho de equipes, especialmente com futura integração a dados externos.

Por fim, a totalidade dos participantes afirmou que a plataforma melhora a produtividade, principalmente pela redução do trabalho manual, apoio à organização do conteúdo, facilidade na formatação e redação dos textos e otimização de rotinas repetitivas.

Capítulo 5

Conclusões

Este trabalho apresentou a análise, modelagem, implementação e validação da plataforma web FlexRelat, projetada para apoiar a criação e edição de relatórios personalizados por meio do uso de Inteligência Artificial Generativa. A solução foi concebida com foco na redução da complexidade técnica, no aumento da produtividade e na melhoria da qualidade dos documentos produzidos, atendendo usuários com diferentes níveis de conhecimento.

Os resultados obtidos na etapa de validação demonstram que a plataforma apresenta desempenho consistente, elevada taxa de sucesso e total conformidade com os requisitos funcionais definidos, sem a ocorrência de falhas críticas.

A avaliação realizada por usuários de diferentes contextos organizacionais evidenciou alta satisfação geral da plataforma, utilidade e possível melhoria na produtividade, reforçando que a plataforma poderá ser aplicada em diversos cenários corporativos reais. As sugestões coletadas indicam oportunidades de evolução, especialmente relacionadas à automação, integração com fontes externas de dados e aprimoramentos de usabilidade.

A utilização de tecnologias modernas como Next.js, NestJS e PostgreSQL, aliadas à integração com a API da OpenAI, mostrou-se adequada para sustentar uma plataforma robusta, modular e escalável.

Entretanto, o desenvolvimento do FlexRelat também apresentou limitações. A dependência de serviços externos de IA implica variações no tempo de resposta e eventuais inconsistências nos resultados gerados, especialmente em cenários com grande volume de contexto. Além disso, alguns aspectos de usabilidade, como a compreensão inicial da estrutura por seções, definição de templates e refinamento da exportação em PDF, mostraram-se pontos que necessitam de melhorias.

No que tange à validação experimental, o estudo apresentou limitações quanto à abrangência e ao ambiente de teste. Embora as funcionalidades centrais tenham sido validadas, os testes foram realizados com um grupo restrito de usuários, o que pode não refletir toda a diversidade de perfis e níveis de literacia digital do público-alvo final.

Além disso, cabe ressaltar as limitações inerentes à natureza deste projeto. O caráter in-

dividual do desenvolvimento, aliado ao cronograma acadêmico restrito, impôs delimitações ao escopo da plataforma. O FlexRelat foi concebido como um protótipo funcional para prova de conceito. Consequentemente, prioridades foram estabelecidas na validação da arquitetura central e da integração com a IA, em detrimento de uma robustez do sistema.

Diante dos resultados obtidos, observa-se que, apesar da plataforma FlexRelat atender aos objetivos propostos, ainda existem limitações e oportunidades de aprimoramento, que incluem:

1. Aprimoramento da IA: melhorar os mecanismos de controle de contexto e consistência das respostas de IA.
2. Novas integrações: inserir novas integrações com fontes externas de dados, como bancos de dados, planilhas e APIs.
3. Maior automatização: adicionar geração automática de relatórios a partir de dados estruturados e tabelas.
4. Envio automatizado: implementação de envio periódico e agendamento de relatórios.
5. Formatos de exportação: expansão dos formatos de exportação, como DOC e DOCX.
6. Melhoria na exportação: resolução dos erros no processo de exportação em PDF.
7. Funcionalidades de texto: Inclusão de recursos de autocomplete e busca semântica.
8. Melhorias no *Chat* com IA: Implementação de histórico e gerenciamento de prompts e novas interações com o FlexBot.

Conclui-se, portanto, que o FlexRelat se mostra uma solução viável e promissora para apoio à geração de relatórios assistida por Inteligência Artificial Generativa, contribuindo para a redução de custos operacionais, aumento da eficiência e melhoria da qualidade dos documentos produzidos, ao mesmo tempo em que abre espaço para pesquisas e evoluções futuras na área.

Referências Bibliográficas

- Ajay Agrawal, Joshua Gans, and Avi Goldfarb. *Power and prediction: The disruptive economics of artificial intelligence*. Harvard Business Press, 2022.
- Ritesh Chugh and Sriman Grandhi. Why business intelligence? *International Journal of E-Entrepreneurship and Innovation*, 4:1–14, 08 2013. doi: 10.4018/ijeei.2013040101.
- Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93, 06 2024. ISSN 2161-7201. doi: 10.1093/jla/laae003. URL <https://doi.org/10.1093/jla/laae003>.
- Thomas Davenport. *Big data at work: dispelling the myths, uncovering the opportunities*. Harvard Business Review Press, 2014.
- DeepMind. Gemini. <https://deepmind.google/technologies/gemini/>, n.d. Acesso em 26 de Abril de 2025.
- DeepSeek. Deepseek. <https://www.deepseek.com/>, n.d. Acesso em 26 de Abril de 2025.
- Alexandre Ferreira Dos Santos. Metodologia de desenvolvimento de uma api de serviço restful integrado com uma aplicação móvel android para o setor imobiliário. Trabalho de conclusão de curso (bacharelado em sistemas de informação), Universidade Estadual de Maringá (Univem), Maringá, PR, Brasil, December 2013. URL <http://hdl.handle.net/11077/984>. Defendido em 05 de dezembro de 2013.
- Wenqi Fan, Yajuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671470. URL <https://doi.org/10.1145/3637528.3671470>.
- Stefan Feuerriegel, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. Generative ai. *Business & Information Systems Engineering*, 66(1):111–126, 2024. ISSN 1867-0202. doi: 10.1007/s12599-023-00834-7. URL <https://doi.org/10.1007/s12599-023-00834-7>.

- Roy Thomas Fielding and Richard N. Taylor. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000. AAI9980887.
- Francisco José Gonçalves Figueiredo, Davi Feitoza Pinto Marinho, Guilherme Cavalcante Holanda, Natália Bitu Pinto, Rafaelle Cavalcante de Lira, Elisangela Vilar de Assis, and Marco Antônio Prado Nunes. Impacto da ia na produtividade científica: Uma revisão sistematizada. *Revista Foco*, 18(1), jan. 2025. doi: 10.54751/revistafoco.v18n1-015. URL <https://ojs.focopublicacoes.com.br/foco/article/view/7398>.
- Tim Fountaine, Brian McCarthy, and Tamim Saleh. Building the ai-powered organization. *Harvard business review*, 97(4):62–73, 2019.
- Dennis dos Santos Gomes. Inteligência artificial: conceitos e aplicações. *Revista Olhar Científico*, 1(2):234, 2010.
- Marco Gonzalez and Vera Lúcia Strube Lima. Recuperação de informação e processamento da linguagem natural. In *XXIII Congresso da Sociedade Brasileira de Computação*, volume 3, pages 347–395. sn, 2003.
- Google and Ipsos. Our life with ai: From innovation to application. Disponível em: https://publicpolicy.google/resources/our_life_with_ai_google_ipsos_report.pdf, 2025. Acesso em 26 de abril de 2025.
- Jonny Holmström and Noel Carroll. How organizations can innovate with generative ai. *Business Horizons*, 2024. ISSN 0007-6813. doi: <https://doi.org/10.1016/j.bushor.2024.02.010>. URL <https://www.sciencedirect.com/science/article/pii/S0007681324000247>.
- Earl B Hunt. *Artificial intelligence*. Academic Press, 2014.
- Raisa Islam and Owana Marzia Moushi. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*, 2024.
- Jerry Kaplan. *Generative artificial intelligence: What everyone needs to know*. Oxford University Press, 2024.

John McCarthy. Programs with common sense. Tech. report no. irs-rr-514, Stanford Artificial Intelligence Laboratory, March 1959.

Microsoft. 2025 work trend index annual report: The year the frontier firm is born. Technical report, Microsoft, 4 2025. URL <https://www.microsoft.com/en-us/worklab/work-trend-index/2025-the-year-the-frontier-firm-is-born>. Acessado em: 19 jan. 2026.

Emilia Monego, Fabio Lopes Schwertz, Fernanda dos Santos Medeiros, Juliano Carvalho Barros, Maíke Stredr Ferreira Machado, and Rodrigo Donizeti da Silva. Teorias da administração e das relações humanas. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 7(8):254–261, ago. 2021. doi: 10.51891/rease.v7i8.1882. URL <https://periodicorease.pro.br/rease/article/view/1882>.

OpenAI. Gpt-4.1. Disponível em: <https://openai.com/index/gpt-4-1/>, 2024. Acesso em 26 de Abril de 2025.

OpenAI. Api pricing. Disponível em: <https://platform.openai.com/docs/pricing>, 2025. Acesso em 18 de Dezembro de 2025.

Daniel Henrique Pereira. Inteligência artificial: usos e aplicações em pequenos negócios. Monografia (bacharelado em administração), Pontifícia Universidade Católica de Minas Gerais, Contagem, MG, January 2021. URL https://www.researchgate.net/publication/358009360_Inteligencia_Artificial_Usos_e_Aplicacoes_em_Pequenos_Negocios.

Pichai, Sundar. Google ceo: A.i. is more important than fire or electricity. Online article, CNBC via CNN Business, February 2018. URL <https://www.cnbc.com/2018/02/01/google-ceo-sundar-pichai-ai-is-more-important-than-fire-electricity.html>. Acesso em 27 de abril de 2025.

Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154, 2023. ISSN 2667-3452. doi: <https://doi.org/10.1016/j.iotcps.2023.04.003>. URL <https://www.sciencedirect.com/science/article/pii/S266734522300024X>.

- Mahshid Sadeghpour, Arathi Arakala, and Asha Rao. Generative ai in collaborative academic report writing: Advantages, disadvantages, and ethical considerations, 2025. URL <https://arxiv.org/abs/2504.08832>.
- Diego Moura Gomes da Silva. A inteligência artificial na tomada de decisão estratégica em empresas. Trabalho de conclusão de curso (bacharelado em administração), Centro Universitário Christus, Fortaleza, December 2023. URL <https://repositorio.unichristus.edu.br/jspui/handle/123456789/1679>. Orientador: SOUZA, Elnivan Moreira de.
- Richard Thomas and Brae Webb. Layered architecture. Handout, University of Queensland, February 2023. URL <https://csse6400.uqcloud.net/handouts/layered.pdf>. Acesso em 7 de julho de 2025.
- Zhenan Tu. Research on the application of layered architecture in computer software development. *Journal of Computing and Electronic Information Management*, 11:34–38, 2023. doi: 10.54097/jceim.v11i3.08.
- Marco Tulio de Oliveira Valente. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, 2020. ISBN 978-65-00-01950-6. URL <http://hdl.handle.net/1843/37905>. Acesso em 27 de abril de 2025.
- Hanlong Wan, Jian Zhang, Yan Chen, Weili Xu, and Fan Feng. Exploring gen-ai applications in building research and industry: A review. *Building Simulation*, 18(6):1251–1273, May 2025. ISSN 1996-8744. doi: 10.1007/s12273-025-1279-x. URL <http://dx.doi.org/10.1007/s12273-025-1279-x>.
- Chenguang Wang, Mu Li, and Alexander J. Smola. Language models with transformers, 2019. URL <https://arxiv.org/abs/1904.09408>.

Apêndice A

Requisitos do sistema

A.1 Requisitos funcionais

Tabela 12: Requisitos Funcionais

Identificador	Nome	Sumário	Ator
RF01	Manter usuário	O usuário usa essa funcionalidade para consultar, incluir, alterar e excluir o próprio cadastro.	Usuário
RF02	Fazer Login	O usuário usa essa funcionalidade para ganhar acesso aos recursos privados do sistema.	Usuário
RF03	Manter documento	O usuário usa essa funcionalidade para consultar, incluir, alterar e excluir um documento.	Usuário
RF04	Manter modelo	O usuário usa essa funcionalidade para consultar, incluir, alterar e excluir um modelo.	Usuário
RF05	Emitir link de compartilhamento	O usuário usa essa funcionalidade para emitir um link de acesso a um documento ou modelo.	Usuário

Identificador	Nome	Sumário	Ator
RF06	Manter componente	O usuário usa essa funcionalidade para consultar, incluir, alterar e excluir um componente dentro de um documento ou modelo.	Usuário
RF07	Manter propostas de alteração por IA	O usuário usa essa funcionalidade para consultar, solicitar, aprovar e reprovar uma proposta de alteração por IA.	Usuário

A.2 Requisitos não-funcionais

Tabela 13: Requisitos Não Funcionais

Identificador	Classificação	Tipo	Descrição
RNF01	Desejável	Usabilidade	Usuários devem ser capazes de usar todas as funcionalidades da aplicação sem nenhum treinamento.
RNF02	Essencial	Implementação	As entidades do Banco de Dados devem utilizar UUID como padrão para os identificadores.
RNF03	Essencial	Implementação	A autenticação para requisição de endpoints privados da API deverá ser feita através de cookies seguros (HttpOnly).
RNF04	Desejável	Implementação	O sistema deverá ser composto por uma API RESTful e um sistema de software para web.

Identificador	Classificação	Tipo	Descrição
RNF05	Desejável	Implementação	O projeto deverá ser feito em Node.js, utilizando o framework Express para a API e Typescript com NextJs para o sistema de software para web.
RNF06	Essencial	Implementação	O sistema deverá armazenar dados em um Sistema Gerenciador de Banco de Dados Relacional.
RNF07	Desejável	Implementação	O sistema não deverá permitir a utilização de funcionalidades de forma offline.
RNF08	Essencial	Implementação	Usuários com acesso adequado deverão poder emitir, adicionar, alterar e excluir dados através de rotas na API do sistema que deverá utilizar o banco de dados relacional para armazenamento de dados.
RNF09	Desejável	Implementação	Exportação de relatórios deverá ser apenas para PDF, mas deve ser possível implementar novos formatos de forma simplificada.
RNF10	Desejável	Padrões	Os arquivos do projeto deverão ser versionados por Git e publicadas por meio do Github.

Identificador	Classificação	Tipo	Descrição
RNF11	Essencial	Segurança	O acesso às funcionalidades privadas do sistema devem ser apenas por usuário autorizados por meio de login e senha.
RNF12	Essencial	Segurança	A aplicação deverá possuir recursos de criptografia entre o servidor e a base de dados, para armazenamento de informações sigilosas, utilizando recursos próprios e do SGBD.

A.3 Regras de Negócio

Tabela 14: Regras de Negócio

Identificador	Descrição	Requisito Funcional
RN01	Para cadastrar um novo usuário é necessário informar nome de usuário, email e senha.	RF01
RN02	O administrador do sistema poderá remover o registro dos usuário cadastrados.	RF01
RN03	administrador do sistema poderá alterar os dados dos usuários cadastrados.	RF01
RN04	Para fazer login o usuário deverá informar o seu email e senha.	RF02
RN05	Quando um documento for criado o nome deverá ser “Sem título” e poderá ser atualizado após a criação.	RF03

Identificador	Descrição	Requisito Funcional
RN06	Quando um documento for criado ele será privado.	RF03
RN07	O usuário poderá editar apenas documentos que pertencem a ele.	RF03
RN08	O usuário poderá editar apenas modelos que pertencem a ele.	RF04
RN09	O usuário poderá excluir apenas documentos que pertencem a ele.	RF03
RN10	O usuário poderá excluir apenas modelos que pertencem a ele.	RF04
RN11	O link de compartilhamento deve conter o código público associado ao documento ou modelo.	RF05
RN12	O usuário poderá visualizar documentos de outras pessoas por um link de compartilhamento público.	RF03
RN13	Documentos privados só podem ser acessados por quem os criou.	RF03, RF05
RN14	Todos os modelos são públicos.	RF04
RN15	Deverá existir uma opção para fazer um documento a partir de um modelo.	RF03, RF04
RN16	Os componentes devem ser visualizados na aba “Componentes” dentro de um documento ou modelo editáveis.	RF06
RN17	As propriedades de um componente poderão ser alteradas, caso o usuário selecione um componente.	RF06

Identificador	Descrição	Requisito Funcional
RN18	Um componente deve ser editado ao ser clicado, com opções de estilos e opção de remoção de componente.	RF06
RN19	Para fazer um pedido de alteração para a IA, o usuário poderá fazer através de texto em linguagem natural ou com opções pré-definidas.	RF07
RN20	Toda proposta de alteração feita pela Inteligência artificial deve ser informada ao usuário para aprovação.	RF07
RN21	Toda proposta de alteração feita por IA poderá ser visualizada nas páginas do documento ou modelo.	RN07
RN22	Quando o usuário aprovar uma alteração proposta pela IA, essa alteração deve ser feita no componente da página especificada e também deve ser salvo um resumo do que foi aprovado.	RF07
RN23	Quando o usuário reprovar uma alteração proposta pela IA, essa alteração deve ser removida das páginas do documento ou modelo.	RF07
RN24	O sistema deverá emitir as propostas de alterações por IA somente para o usuário que solicitou.	RF07
RN25	Somente o usuário que solicitou poderá aprovar a proposta de alteração feita pela IA.	RF07

Identificador	Descrição	Requisito Funcional
RN26	Somente o usuário que solicitou poderá reprovar a proposta de alteração feita pela IA.	RF07

Apêndice B

Especificação dos casos de Uso

UC-01: Cadastrar Usuários

Nome: Cadastrar usuários.

Identificador: UC-01.

Sumário: O sistema deverá permitir cadastrar novos usuários no sistema, para fazê-los obter acesso a funcionalidades do sistema.

Ator primário: Usuário.

Pré-condição: O usuário não deve estar autenticado pelo sistema.

Fluxo principal:

1. O sistema oferece uma opção de cadastro de um novo usuário.
2. O usuário escolhe a opção de cadastro.
3. O sistema apresenta um formulário para preenchimento dos dados pessoais do novo usuário.
4. O usuário preenche os dados e prossegue com o cadastro. (FE1)
5. O sistema valida os dados.
6. O sistema salva os dados do novo usuário.
7. O caso de uso retorna para o passo 1.

Fluxo de exceção - FE1: Violação da RN01.

1. O sistema informa que os campos nome de usuário, email e senha são obrigatórios.
2. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um novo usuário foi cadastrado pelo sistema.

Regras de negócio: RN01.

UC-02: Manter usuários

Nome: Manter usuários.

Identificador: UC-02.

Sumário: O sistema deverá permitir atualizar, remover e ler dados de usuários no sistema.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema;

Fluxo principal:

1. O sistema disponibiliza os dados do usuário.
2. O sistema verifica se o perfil pertence ao usuário que iniciou o caso de uso.
3. O sistema oferece uma opção de atualização e remoção do usuário. (FE1) (FE2)
4. O usuário escolhe a opção desejada.
5. O caso de uso retorna para o passo 1.

Fluxo alternativo (3): Atualização do usuário.

1. O sistema disponibiliza um formulário para preenchimento com os novos dados pessoais do usuário.
2. O usuário preenche com as novas informações.
3. O sistema valida as informações.
4. O sistema altera os dados do usuário.

Fluxo alternativo (3): Remoção do usuário.

1. O sistema disponibiliza um formulário de confirmação de remoção.
2. O usuário confirma a remoção do seu usuário.

3. O sistema remove os dados do usuário.
4. O sistema revoga o acesso do usuário ao sistema.

Fluxo de exceção - FE1: Violação da RN02.

1. O sistema não disponibiliza a opção de remoção do usuário.
2. O sistema retorna mensagem de erro dizendo que não é possível remover um perfil de usuário que não o pertence.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Fluxo de exceção - FE2: Violação da RN03.

1. O sistema não disponibiliza a opção de atualização dos dados do usuário.
2. O sistema retorna mensagem de erro dizendo que não é possível atualizar um perfil de usuário que não o pertence.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um usuário foi atualizado ou removido pelo sistema.

Regras de negócio: RN02, RN03.

UC-03: Realizar login

Nome: Realizar login.

Identificador: UC-03.

Sumário: O sistema deverá permitir que o usuário se autentique no sistema para acessar a recursos privados.

Ator primário: Usuário.

Pré-condição: O usuário não deve estar autenticado pelo sistema;

Fluxo principal:

1. O sistema disponibiliza um formulário para que o usuário preencha com as informações necessárias para o login.
2. O usuário preenche as informações necessárias.
3. O sistema valida se as informações atendem aos critérios para a autenticação (FE1).
4. O sistema realiza a autenticação do usuário no sistema.
5. O caso de uso retorna para o passo 1.

Fluxo de exceção - FE1: Violação da RN04.

1. O sistema informa que os campos Email e Senha são obrigatórios.
2. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um usuário foi autenticado no sistema.

Regras de negócio: RN04.

UC-04: Manter Documento

Nome: Manter Documento.

Identificador: UC-04.

Sumário: O usuário do sistema utiliza essa funcionalidade para manter atualizado o cadastro de documentos.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema.

Fluxo principal:

1. O sistema oferece uma lista de documentos criados pelo usuário e uma lista com os documentos acessados recentemente por ele com as opções de cadastrar, consultar, alterar e excluir um documento.
2. O usuário escolhe a opção desejada.
3. O caso de uso retorna para o passo 1.

Fluxo alternativo (2): Consultar documentos.

1. O sistema apresenta os dados do documento selecionado. (FE1)
2. O usuário confirma a visualização.

Fluxo alternativo (2): Cadastrar documento.

1. O sistema cria um novo documento com o nome “Sem título”.
2. O sistema redireciona o usuário para a página de edição do documento.

Fluxo alternativo (2): Cadastrar documento a partir de um modelo.

1. O sistema oferece uma lista de modelos criados pelo usuário e uma lista de modelos da comunidade.
2. O usuário seleciona um modelo.
3. O sistema cria um novo documento com o nome “Sem título”.
4. O sistema redireciona o usuário para a página de edição do documento.

Fluxo alternativo (2): Alterar conteúdo do documento.

1. O sistema apresenta o conteúdo do documento selecionado, em páginas com formato de folha A4 editável. (FE1)
2. O usuário altera o conteúdo. (FE2)
3. O sistema salva as alterações do conteúdo.

Fluxo alternativo (2): Alterar o nome do documento.

1. O sistema apresenta o nome do documento selecionado (FE1).
2. O usuário altera o nome do documento. (FE2)
3. O sistema salva a alteração do nome do documento.

Fluxo alternativo (2): Excluir documento.

1. O sistema apresenta os dados do documento selecionado. (FE1)
2. O sistema solicita a confirmação da exclusão. (FE2)
3. O usuário confirma a exclusão.
4. O sistema realiza a exclusão.

Fluxo de exceção - FE1: Violação da RN13.

1. O sistema informa que o documento é privado e que está disponível para visualização apenas do autor do documento.
2. O caso de uso retorna ao passo 1.

Fluxo de exceção - FE2: Violação da RN09.

1. O sistema informa que o documento é somente para visualização.
2. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um documento foi incluído, consultado, alterado ou excluído.

Regras de negócio: RN05, RN06, RN07, RN09, RN12, RN13, RN15.

UC-05: Manter Modelo

Nome: Manter Modelo.

Identificador: UC-05.

Sumário: O usuário do sistema utiliza essa funcionalidade para manter atualizado o cadastro de modelos.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema.

Fluxo principal:

1. O sistema oferece uma lista de modelos criados pelo usuário e uma lista com os modelos da comunidade com as opções de criar, consultar, alterar e excluir um modelo.
2. O usuário escolhe a opção desejada.
3. O caso de uso retorna para o passo 1.

Fluxo alternativo (2): Consultar conteúdo do modelo.

1. O sistema apresenta o conteúdo do documento selecionado, em páginas com formato de folha A4.
2. O usuário confirma a visualização.

Fluxo alternativo (2): Criar modelo.

1. O sistema solicita o nome do modelo.
2. O usuário preenche o nome e confirma.
3. O sistema cria um novo modelo.
4. O sistema redireciona o usuário para a página de edição do modelo.

Fluxo alternativo (2): Alterar conteúdo do modelo.

1. O sistema apresenta o conteúdo do documento selecionado, em páginas com formato de folha A4 editável.
2. O usuário altera o conteúdo. (FE1)
3. O sistema salva as alterações do conteúdo.

Fluxo alternativo (2): Alterar o nome do modelo.

1. O sistema apresenta o nome do modelo selecionado.
2. O usuário altera a informação. (FE1)

3. O sistema salva a alteração.

Fluxo alternativo (2): Excluir modelo.

1. O sistema apresenta os dados do modelo selecionado. (FE2)
2. O sistema solicita a confirmação da exclusão.
3. O usuário confirma a exclusão.
4. O sistema realiza a exclusão.

Fluxo de exceção - FE1: Violação da RN08.

1. O sistema não permite nenhuma alteração.
2. O sistema informa que o modelo é somente para visualização.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Fluxo de exceção - FE2: Violação da RN10.

1. O sistema não disponibiliza a opção de exclusão.
2. O caso de uso retorna ao passo 1.

Pós-condição: Um modelo foi incluído, consultado, alterado ou excluído.

Regras de negócio: RN08, RN10.

UC-06: Emitir Link de Compartilhamento

Nome: Emitir link de compartilhamento.

Identificador: UC-06.

Sumário: O usuário do sistema utiliza essa funcionalidade para emitir um link de acesso a um documento ou modelo.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema; O usuário deve selecionar o documento ou modelo.

Fluxo principal:

1. O sistema oferece a opção de geração de link de compartilhamento. (FE1)
2. O usuário seleciona a opção.
3. O sistema gera o link de compartilhamento do documento ou modelo e disponibiliza uma opção de copiar.

Fluxo de exceção - FE1: Violação da RN13.

1. O sistema não disponibiliza opção de geração de link de compartilhamento.
2. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um link de compartilhamento foi gerado.

Regras de negócio: RN13.

UC-07: Manter Componente

Nome: Manter Componente.

Identificador: UC-07.

Sumário: O usuário do sistema utiliza essa funcionalidade para manter atualizado o cadastro de componentes em um documento ou modelo.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema; O usuário deve estar dentro de um documento ou modelo editável.

Fluxo principal:

1. O sistema oferece uma lista de componentes inseridos no documento ou modelo, com as opções de criar, consultar, alterar e excluir um componente.
2. O usuário escolhe a opção desejada.

3. O caso de uso retorna para o passo 1.

Fluxo alternativo (2): Criar componente.

1. O usuário seleciona o componente escolhido.
2. O usuário escolhe a localização desejada. (FE1)
3. O sistema insere o componente na área escolhida.

Fluxo alternativo (2): Consultar componentes disponíveis.

1. O sistema apresenta os componentes disponíveis para serem adicionados ao documento ou modelo.
2. O usuário confirma a visualização.

Fluxo alternativo (2): Alterar um componente.

1. O sistema apresenta as informações do componente inserido dentro de um documento ou modelo.
2. O usuário altera as informações. (FE1)
3. O sistema salva as alterações.

Fluxo alternativo (2): Remover um componente.

1. O usuário seleciona o componente dentro de um documento ou modelo.
2. O usuário seleciona a opção de remover o componente. (FE1)
3. O sistema remove o componente do documento ou modelo.

Fluxo de exceção - FE1: Violação da RN09.

1. O sistema informa que o documento é somente para visualização.
2. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Um componente foi criado, consultado, alterado ou excluído.

Regras de negócio: RN09, RN17, RN18.

UC-08: Manter propostas de alteração por IA

Nome: Manter propostas de alteração por IA.

Identificador: UC-08.

Sumário: O usuário do sistema utiliza essa funcionalidade para manter atualizado às propostas de alteração de componentes em um documento ou modelo feitas por Inteligência artificial.

Ator primário: Usuário.

Pré-condição: O usuário deve estar autenticado pelo sistema; O usuário deve estar dentro de um documento ou modelo editável.

Fluxo principal:

1. O sistema oferece uma lista de solicitações de alteração por IA, com opções para aprovação, reprovação e solicitação de alterações por IA. (FE1)
2. O usuário escolhe a opção desejada.
3. O caso de uso retorna para o passo 1.

Fluxo alternativo (2): Solicitar alteração.

1. O usuário faz a requisição da alteração que deseja.
2. O sistema disponibiliza as propostas feitas pela IA para adição, alteração ou exclusão de um componente ou mais.

Fluxo alternativo (2): Aprovar alteração.

1. O usuário aceita a alteração proposta pela IA. (FE2)
2. O sistema realiza a alteração.
3. O sistema disponibiliza um resumo das alterações aceitas.
4. O sistema salva as mudanças feitas no documento ou modelo.

Fluxo alternativo (2): Reprovar alteração.

1. O usuário não aceita a alteração proposta pela IA. (FE3)
2. O sistema não realiza a alteração.

Fluxo de exceção - FE1: Violação da RN24.

1. O sistema informa que não foi o usuário que solicitou aquela alteração.
2. O sistema não retorna os dados da proposta de alteração.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Fluxo de exceção - FE2: Violação da RN25.

1. O sistema informa que não foi o usuário que solicitou aquela alteração.
2. O sistema não aceita a aprovação da proposta de alteração.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Fluxo de exceção - FE3: Violação da RN26.

1. O sistema informa que não foi o usuário que solicitou aquela alteração.
2. O sistema não aceita a reprovação da proposta de alteração.
3. O caso de uso retorna ao início do fluxo que fez a chamada.

Pós-condição: Uma alteração por IA foi consultada, solicitada, aprovada ou reprovada.

Regras de negócio: RN19, RN20, RN21, RN22, RN23, RN24, RN25, RN26.

Apêndice C

Diagrama de Classe UML

Este apêndice apresenta na Figura 17 o diagrama de classes UML completo da plataforma FlexRelat. Diferentemente da versão simplificada apresentada na Figura 3, este diagrama detalha todas as classes com seus atributos e relacionamentos definidos na modelagem do sistema, oferecendo uma visão abrangente da estrutura interna da aplicação. O objetivo é fornecer uma representação mais precisa da arquitetura do sistema para fins de documentação técnica e consulta futura.

Fonte: Elaborado pelo autor.

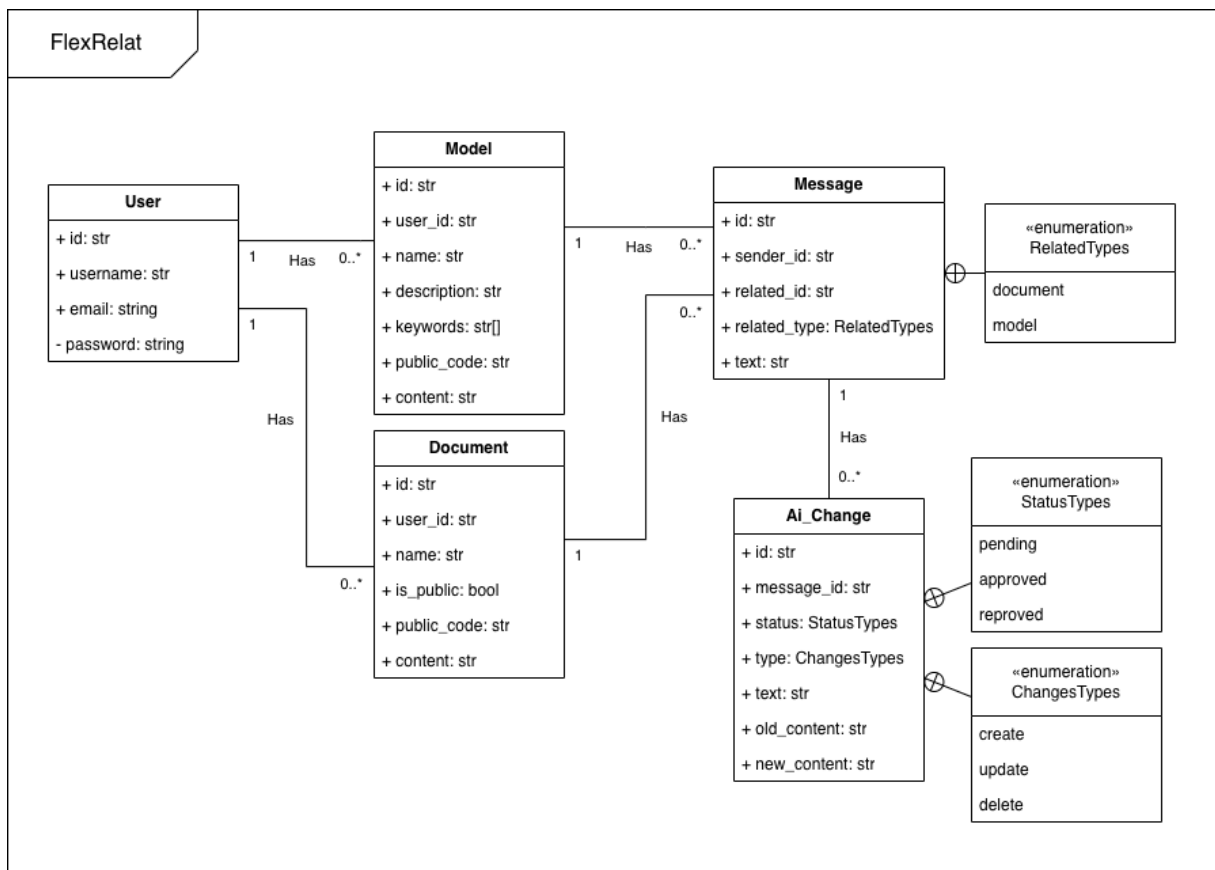


Figura 17: Diagrama de classes UML completo do sistema.