



**Universidade Católica do Salvador
Bacharelado em Engenharia de Software**

**Jailton da Cruz Mocitaiba Filho
João Antônio Luz dos Santos
Nicolas Raimundo Meneses de Araújo
Silas Coutinho Passos
Vitor Gabriel Fontes Santos**

**NoRisk: Sistema Baseado em Inteligência Artificial para Análise
de Riscos Jurídicos em Contratos de Estágio**

Salvador

2025

Jailton da Cruz Mocitaiba Filho
João Antônio Luz dos Santos
Nicolas Raimundo Meneses de Araújo
Silas Coutinho Passos
Vitor Gabriel Fontes Santos

**NoRisk: Sistema Baseado em Inteligência Artificial
para Análise de Riscos Jurídicos em Contratos de
Estágio**

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia de Software.
Orientadora: Prof.^a Msc. Semíramis Ribeiro de Assis

Universidade Católica do Salvador

Salvador
2025

Jailton da Cruz Mocitaiba Filho
João Antônio Luz dos Santos
Nicolas Raimundo Meneses de Araújo
Silas Coutinho Passos
Vitor Gabriel Fontes Santos

NoRisk: Sistema Baseado em Inteligência Artificial para Análise de Riscos Jurídicos em Contratos de Estágio

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Salvador, 14 de janeiro de 2026

Banca Examinadora:

Prof.^a Msc. Semíramis Ribeiro de Assis
Universidade Católica do Salvador
Orientadora

Prof. Dr. Flávio Dusse
Universidade Católica do Salvador

Prof.^a Sheila Tirony de Almeida Silva
Universidade Católica do Salvador

Dedicamos este trabalho a todos os calouros do curso de Direito que iniciam sua jornada em busca de uma vaga de estágio digna, com boas condições de aprendizado e respeito às leis.

Agradecimentos

Agradecemos, em especial, à nossa orientadora Prof.^a Semiramis Assis, por sua dedicação, paciência e disponibilidade em nos orientar. Mesmo com a agenda cheia, sempre encontrava tempo para esclarecer dúvidas, reunir-se conosco e analisar o desenvolvimento do projeto. Seu apoio e confiança foram fundamentais para a conclusão deste trabalho, mesmo diante do curto prazo.

Estendemos nossos agradecimentos a todos os professores que fizeram parte de nossa trajetória acadêmica, cujos ensinamentos e orientações contribuíram significativamente para nossa formação pessoal e profissional.

Agradecemos também à nossa equipe, que colaborou de forma comprometida e eficaz durante todas as etapas do projeto, e à família e amigos, cujo apoio, incentivo e compreensão tornaram possível a dedicação necessária para a realização deste TCC.

"I believe in human-centered AI to benefit people in positive and benevolent ways."
(Fei-Fei Li)

Resumo

Este trabalho propõe o desenvolvimento de uma ferramenta baseada em Inteligência Artificial (IA) para a análise e avaliação de riscos legais em contratos de estágio. A solução tem como objetivo democratizar o acesso à informação jurídica, permitindo que estudantes e profissionais não especialistas compreendam cláusulas contratuais e identifiquem possíveis inconformidades legais. A ferramenta utiliza técnicas de Processamento de Linguagem Natural (PLN) e modelos de linguagem baseados em aprendizado profundo para analisar o texto dos contratos, verificar a conformidade com a legislação vigente e apontar cláusulas obrigatórias ausentes ou inadequadas. A análise de literatura evidenciou o potencial da IA como instrumento de apoio à análise contratual, e os testes mostraram que a ferramenta desenvolvida identificou corretamente todos os documentos em experimentos limitados a contratos preenchidos com dados fictícios. O sistema alcançou 100% de acerto na identificação de documentos íntegros e não íntegros, com percentual de erros igual a 0%.

Palavras-Chave: 1. Inteligência Artificial. 2. Cláusulas 3. Processamento de Linguagem Natural. 4. Contratos. 5. Avaliação de Riscos.

Abstract

This work proposes the development of an Artificial Intelligence (AI)-based tool for the analysis and evaluation of legal risks in internship contracts. The solution aims to democratize access to legal information, allowing students and non-specialist professionals to understand contractual clauses and identify potential legal non-conformities. The tool uses Natural Language Processing (NLP) techniques and deep learning-based language models to analyze the text of contracts, verify compliance with current legislation, and point out missing or inadequate mandatory clauses. Literature analysis highlighted the potential of AI as a tool to support contractual analysis, and tests showed that the developed tool correctly identified all documents in experiments limited to contracts filled with fictitious data. The system achieved 100% accuracy in identifying complete and non-complete documents, with a 0% error rate.

Keywords: 1. Artificial Intelligence. 2. Clauses. 3. Natural Language Processing. 4. Contracts. 5. Risk Assessment.

Lista de figuras

Figura 1 – Exemplo de quadro Kanban	26
Figura 2 – Diagrama de Casos de Uso	27
Figura 3 – Diagrama de Sequência	28
Figura 4 – Diagrama de Classes	29
Figura 5 – Exemplo de aplicação da Escala Likert	29
Figura 6 – Quadro Trello – Setembro: pendências de documentação e início do protótipo	33
Figura 7 – Quadro Trello – Outubro: finalização do desenvolvimento do sistema	34
Figura 8 – Quadro Trello – Novembro: ajustes finais e preparação da apresentação	35
Figura 9 – Distribuição dos tipos de respondentes	37
Figura 10 – Distribuição dos alunos quanto ao curso prévio da disciplina de Direito do Trabalho	37
Figura 11 – Distribuição da essencialidade por requisito	39
Figura 12 – Fluxo BPMN do processo de análise do sistema NoRisk	49
Figura 13 – Diagrama UML de Casos de Uso do sistema NoRisk.	50
Figura 14 – Fluxo do Módulo 1	51
Figura 15 – Fluxo do Módulo 2	52
Figura 16 – Fluxo do Módulo 3	53
Figura 17 – Tela inicial do sistema NoRisk	56
Figura 18 – Tela de <i>upload</i> do arquivo – NoRisk	56
Figura 19 – Tela de análise automática do contrato	57
Figura 20 – Tela de resultado indicando que o contrato é íntegro	58
Figura 21 – Tela de resultado indicando que o contrato não é íntegro	58
Figura A1 – Formulário de Pesquisa – Página 1	66
Figura A2 – Formulário de Pesquisa – Página 2	67
Figura A3 – Formulário de Pesquisa – Página 3	68
Figura A4 – Formulário de Pesquisa – Página 4	69
Figura A5 – Formulário de Pesquisa – Página 5	70
Figura B1 – Respostas da questão 1 do Forms	71
Figura B2 – Respostas da questão 2 do Forms (aplicada somente aos alunos) .	71
Figura B3 – Respostas da questão 3 do Forms	72
Figura B4 – Respostas da questão 4 do Forms	72
Figura B5 – Respostas da questão 5 do Forms	73
Figura B6 – Respostas da questão 6 do Forms	73

Figura B7 – Respostas da questão 7 do Forms	74
Figura B8 – Respostas da questão 8 do Forms	74
Figura B9 – Respostas da questão 9 do Forms	75
Figura B10–Respostas da questão 10 do Forms	75
Figura B11–Respostas da questão 11 do Forms	76
Figura B12–Respostas da questão 12 do Forms	76
Figura C1 – Trecho do código com definição do <i>prompt</i> de análise de contratos .	77
Figura D1 – Essa parte de <code>analyze.py</code> o código inicializa a API com FastAPI. . .	79
Figura D2 – Continuação do arquivo <code>analyze.py</code> mostrando como funciona a rota da API.	81
Figura D3 – Trecho inicial do código <code>App.jsx</code>	82
Figura D4 – Funções de controle e envio de arquivo no <code>App.jsx</code>	83
Figura D5 – Tratamento de eventos de upload e arrastar-e-soltar no <code>App.jsx</code> . .	85
Figura D6 – Definição de estilos visuais da interface no <code>App.jsx</code>	87
Figura D7 – Função de renderização condicional do conteúdo da interface . . .	88
Figura D8 – Estrutura principal da aplicação e integração dos componentes visuais	89

Lista de Siglas e Abreviaturas

AI	<i>Artificial Intelligence</i>
API	<i>Application Programming Interface</i>
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
BES	Bacharelado em Engenharia de Software
BPMN	<i>Business Process Modeling and Notation</i>
CORS	<i>Cross-Origin Resource Sharing</i>
CSS	<i>Cascading Style Sheets</i>
CVS	<i>Concurrent Versions System</i>
GPT	<i>Generative Pre-trained Transformer</i>
HTML	<i>HyperText Markup Language</i>
IA	Inteligência Artificial
IDE	<i>Integrated Development Environment</i>
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
LLMs	<i>Large Language Model</i>
PDF	<i>Portable Document Format</i>
PLN	Processamento de Linguagem Natural
ML	<i>Machine Learning</i>
MVP	<i>Minimum Viable Product</i>
NLP	<i>Natural Language Processing</i>
REST	<i>Representational State Transfer</i>
RN	Regras de Negócio
SVN	<i>Subversion</i>
UML	<i>Unified Modeling Language</i>
URLs	<i>Uniform Resource Locator</i>
UX	<i>User Experience</i>

Sumário

1	INTRODUÇÃO	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Projetos Correlatos	20
2.2	Inteligência Artificial e Processamento de Linguagem Natural	20
2.3	Modelos Generativos e a API Gemini	21
2.4	Base Legal	21
2.4.1	Legislação Aplicada aos Contratos de Estágio	21
2.4.2	Legislação de Proteção de Dados: GDPR e LGPD	21
2.5	Tecnologias e Ferramentas Utilizadas	22
2.5.1	<i>Frontend</i>	22
2.5.2	<i>HyperText Markup Language (HTML)</i>	22
2.5.3	JavaScript	22
2.5.4	Vite	23
2.5.5	React	23
2.5.6	<i>Cascading Style Sheets (CSS)</i>	23
2.5.7	<i>Backend</i>	23
2.5.8	Python	23
2.5.9	<i>Application Programming Interface (API)</i>	24
2.5.10	<i>JavaScript Object Notation (JSON)</i>	24
2.5.11	<i>Representational State Transfer (REST)</i>	24
2.5.12	<i>Cross-Origin Resource Sharing (CORS)</i>	24
2.5.13	<i>Integrated Development Environment (IDE)</i>	24
2.5.14	Visual Studio Code	25
2.5.15	Repositório de Código Fonte	25
2.5.16	GitHub	25
2.5.17	Kanban	25
2.5.18	<i>Unified Modeling Language (UML)</i>	26
2.5.19	Escala Likert	29
2.5.20	<i>Minimum Viable Product (MVP)</i>	30
2.5.21	<i>Business Process Model and Notation (BPMN)</i>	30
3	METODOLOGIA	31
3.1	Classificação da Pesquisa	31
3.2	Etapas da Metodologia	32

3.3	Gestão do Projeto	33
3.3.1	Evolução do Projeto no Trello	33
3.4	Aspectos Éticos da Pesquisa	35
3.5	Apresentação dos Resultados	36
3.6	Coleta e Análise de Dados	36
4	NORISK	40
4.1	Requisitos do Sistema	40
4.1.1	Requisitos Funcionais	40
4.1.1.0.1	Módulo: <i>Upload</i> e Processamento	40
4.1.1.0.2	Módulo: Análise de Cláusulas	40
4.1.1.0.3	Módulo: Relatório e Visualização	41
4.1.2	Requisitos Não Funcionais	41
4.1.2.0.1	Desempenho e Eficiência	41
4.1.2.0.2	Escalabilidade	42
4.1.2.0.3	Disponibilidade	42
4.1.2.0.4	Usabilidade e Experiência do Usuário	42
4.1.2.0.5	Precisão e Qualidade da IA	42
4.2	Regras de Negócio (RN)	42
4.2.0.0.1	RN: Conformidade Legal (Core)	42
4.2.0.0.2	RN: Análise de Risco (IA/ML)	43
4.2.0.0.3	RN: Interface e Usuário (UX/Usabilidade)	43
4.2.0.0.4	RN: Sistema (Geral)	43
4.3	Ferramentas e Tecnologias Utilizadas	44
4.3.1	Controle de Versão e Gestão de Projeto	45
4.3.2	Frontend: Interface do Usuário	46
4.3.3	Backend: Lógica e Processamento	47
4.3.4	Módulo de Inteligência Artificial	47
4.3.4.0.1	Arquitetura do Módulo	47
4.3.4.0.2	Funcionamento do <i>prompt</i> e Avaliação de Conformidade	48
4.3.4.0.3	Integração com o Sistema	48
4.3.4.0.4	Benefícios e Limitações	48
4.4	Modelagem do Processo – Fluxo BPMN	49
4.5	Arquitetura do Sistema - Diagrama UML	50
4.6	Fluxo de Operação	51
4.6.1	Etapas Executadas pelo Módulo de IA	54
4.6.2	Interação Final com o Usuário	55
4.6.3	Síntese Arquitetural	55
4.7	Descrição do Arquivo <code>analyze.py</code>	55

4.8	Tela Inicial – Sistema Inativo	55
4.9	Tela de <i>Upload</i> – Seleção do Arquivo	56
4.10	Tela de Processamento – Análise Automática	57
4.11	Tela de Resultado – Contrato Íntegro	57
4.12	Tela de Resultado – Contrato Não Íntegro	58
4.13	Resultados dos Testes	59
5	CONCLUSÃO E TRABALHOS FUTUROS	60
	REFERÊNCIAS	62
	APÊNDICE A – FORMULÁRIO DE PESQUISA	65
	APÊNDICE B – RESULTADOS DA PESQUISA	71
	APÊNDICE C – <i>PROMPTS</i> ELABORADOS	77
C.1	Descrição do <i>Prompt</i> de Análise	77
C.2	Estrutura Geral do Código	77
	APÊNDICE D – CÓDIGO DA FERRAMENTA	79
D.1	Descrição do Código	79
D.2	Funcionamento Detalhado	80
D.3	Rota da API (/analyze):	81
D.4	Endpoint da API	81
D.5	Descrição do Código – Inicialização da Aplicação	82
D.6	Descrição do Código – Envio do Arquivo e Controle de Estado	83
D.7	Descrição do Código – Interação com o Usuário e Estilização da Interface	85
D.8	Renderização Condicional da Interface e Estrutura Final da Aplicação	88

1 Introdução

A celebração de um contrato de estágio representa um marco fundamental na trajetória acadêmica e profissional de milhões de estudantes brasileiros. Contudo, a complexidade estrutural e a linguagem técnica característica desses documentos jurídicos frequentemente transformam essa etapa em uma fonte de incerteza e vulnerabilidade. Para grande parte dos estagiários, que não possuem formação jurídica, a leitura e a interpretação adequada das cláusulas contratuais tornam-se um desafio significativo.

Essa dificuldade é especialmente sensível no contexto da Lei nº 11.788/2008¹, conhecida como Lei do Estágio (BRASIL, 2008), que estabelece um conjunto de direitos, deveres e limites legais destinados a preservar o caráter educativo da atividade. A assimetria informacional entre as partes contratantes pode resultar na omissão de cláusulas obrigatórias ou na inclusão de disposições inadequadas, comprometendo tanto a segurança jurídica quanto a finalidade formativa do estágio. Como apontam Lima e Ciciliati (LIMA; CICILIATI, 2008), a ausência de padronização e de acessibilidade nos documentos jurídicos dificulta a análise sistemática e transparente de contratos, ampliando os riscos para a parte mais vulnerável da relação.

Nesse cenário, os avanços recentes da Inteligência Artificial (IA) têm demonstrado potencial para transformar a forma como documentos jurídicos são analisados, interpretados e validados. O uso de técnicas de Processamento de Linguagem Natural (PLN) permite tratar textos jurídicos de maneira automatizada, extraindo informações relevantes, identificando padrões e auxiliando na verificação de conformidade legal. Essas inovações têm impulsionado o crescimento da área de *Legal Tech*, que busca aplicar soluções tecnológicas para aumentar a eficiência, a precisão e a transparência na prática jurídica (REMUS; LEVY, 2016; SURDEN, 2019).

Inserido nesse contexto, este trabalho propõe o desenvolvimento de uma aplicação baseada em IA denominada NoRisk, voltada à análise automatizada de contratos de estágio no Brasil. A ferramenta atua como um assistente jurídico virtual, permitindo que estudantes compreendam, de forma mais autônoma e segura, os direitos e deveres estabelecidos em seus contratos. Para isso, o sistema utiliza técnicas de PLN e modelos de linguagem baseados em aprendizado profundo para identificar e classificar cláusulas relevantes, como jornada de trabalho, duração do contrato, concessão de benefícios, recesso remunerado e seguro contra acidentes pessoais.

Além da classificação estrutural, o sistema realiza a verificação da conformidade legal do contrato, comparando automaticamente as cláusulas identificadas com os requisitos obrigatórios previstos na legislação vigente. Inconsistências como cargas ho-

¹ <https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11788.htm>

rárias incompatíveis, ausência de cláusulas compulsórias ou extrapolação do prazo máximo de vigência são apontadas de forma objetiva, auxiliando na identificação de riscos jurídicos (DIKMEN et al., 2025). O sistema também é capaz de sinalizar cláusulas potencialmente abusivas ou atípicas, identificando padrões anômalos mesmo quando não há violação literal da lei, conforme discutido por Kazemi et al. (KAZEMI; ALVANCHI, 2025).

Sendo assim, no escopo deste trabalho, risco jurídico em contratos de estágio é definido como a probabilidade de ocorrência de problemas legais decorrentes da violação da Lei do Estágio, do não cumprimento dos requisitos identificados na pesquisa detalhada no capítulo 3 desta obra e de termos contratuais que contenham cláusulas que se afastam do padrão esperado.

Um dos diferenciais do NoRisk reside na apresentação dos resultados em linguagem clara e acessível. O uso da IA permite traduzir o jargão jurídico para uma forma compreensível ao público leigo, explicando o significado das cláusulas analisadas e suas implicações práticas para o estagiário. Dessa forma, a ferramenta não se limita à identificação de riscos, mas também desempenha um papel educativo, promovendo maior conscientização jurídica.

O desenvolvimento do NoRisk justifica-se pela recorrente dificuldade enfrentada por estudantes ao firmar contratos sem pleno entendimento de seus termos. Essa lacuna de informação pode resultar em prejuízos legais e pedagógicos, como descumprimento de carga horária, ausência de benefícios obrigatórios e situações de exploração laboral. Pesquisas recentes demonstram a eficácia de modelos de aprendizado profundo na análise contratual, alcançando elevados índices de acurácia na identificação de cláusulas de risco, o que reforça a viabilidade técnica da proposta (MOON; CHI; IM, 2022; DIKMEN et al., 2025).

Dessa forma, o presente trabalho insere-se no contexto interdisciplinar entre Direito e Engenharia de Software, contribuindo tanto do ponto de vista científico quanto social. Ao propor uma solução acessível, automática e transparente, o NoRisk busca reduzir a assimetria informacional em relações de estágio, ampliar o acesso ao conhecimento jurídico e evidenciar o papel da Inteligência Artificial como instrumento de apoio à análise e à educação jurídica.

Este trabalho está estruturado em 5 capítulos. O Capítulo 1 (Introdução), apresenta o problema de pesquisa, os objetivos, a justificativa e a contextualização inicial do estudo. O Capítulo 2 (Fundamentação Teórica) aborda todas as tecnologias utilizadas para a elaboração do atual trabalho. O Capítulo 3 (Metodologia) descreve os procedimentos e etapas utilizados para o desenvolvimento do estudo. O Capítulo 4 (NoRisk) apresenta a solução desenvolvida, detalhando a implementação e os principais resultados. Por fim, o Capítulo 5 (Conclusão e Trabalhos Futuros) traz as considerações finais e perspectivas futuras.

2 Fundamentação Teórica

Este tópico apresenta os principais projetos correlatos, conceitos teóricos, tecnológicos e normativos que fundamentam o desenvolvimento do sistema NoRisk. São abordados os temas relacionados à Inteligência Artificial e ao Processamento de Linguagem Natural, o uso de modelos generativos baseados na API Gemini, a legislação aplicada à validação de contratos de estágio, os aspectos legais de proteção de dados pessoais e, por fim, as tecnologias e ferramentas utilizadas na implementação do sistema.

2.1 Projetos Correlatos

Alguns trabalhos importantes foram encontrados e serviram de base para a elaboração deste projeto de pesquisa, os quais exploram o potencial do *Natural Language Processing* (NLP) e do *Machine Learning* na análise automatizada de contratos jurídicos, especialmente em contextos de risco e responsabilidade.

Conforme apresentado por (KAZEMI; ALVANCHI, 2025), técnicas de classificação de sentenças e aprendizado de máquina são aplicadas para detectar declarações potencialmente arriscadas, alcançando alta acurácia na identificação de riscos contratuais.

(DIKMEN et al., 2025) abordam um sistema automatizado que analisa cláusulas contratuais complexas, cujos resultados indicam acurácia de até 89%, demonstrando que a abordagem pode reduzir erros humanos e agilizar a revisão de contratos.

(MOON; CHI; IM, 2022) tratam da utilização de técnicas avançadas de aprendizado profundo para a classificação de sentenças contratuais de forma precisa, destacando o excelente desempenho do modelo BERT em relação a métodos tradicionais de NLP na detecção de riscos legais.

2.2 Inteligência Artificial e Processamento de Linguagem Natural

A IA compreende um conjunto de técnicas computacionais capazes de realizar tarefas que, tradicionalmente, exigiriam inteligência humana, como interpretação de texto, reconhecimento de padrões e tomada de decisão. No contexto jurídico, a IA tem sido amplamente explorada para apoiar atividades como análise documental, classificação de cláusulas e detecção de riscos legais.

O PLN é uma subárea da IA que se dedica à interação entre computadores e linguagem humana. Por meio de técnicas de PLN, é possível extrair informações relevantes, identificar entidades, classificar trechos textuais e interpretar o significado semântico de documentos extensos e complexos, como contratos jurídicos. Essa capacidade é essencial para lidar com textos caracterizados por vocabulário técnico, estrutura rígida e formalidade, comuns no domínio do Direito.

2.3 Modelos Generativos e a API Gemini

Os modelos generativos de linguagem representam um avanço significativo no campo da IA, pois são capazes de compreender contextos complexos e produzir respostas coerentes em linguagem natural. Esses modelos são treinados com grandes volumes de dados textuais, utilizando redes neurais profundas baseadas em arquiteturas do tipo *Transformer*.

A API Gemini, disponibilizada pelo Google, especificamente o modelo `gemini-2.5-flash`¹, é caracterizado por elevada eficiência computacional e rapidez na geração de respostas, tornando-o adequado para aplicações web que demandam processamento quase instantâneo.

Além da API Gemini, outro modelo generativo relevante é o ChatGPT², desenvolvido pela OpenAI, destacando-se como um modelo de linguagem que compreende e gera texto em diferentes contextos. Assim, observa-se o papel dos modelos generativos na evolução da inteligência artificial.

2.4 Base Legal

2.4.1 Legislação Aplicada aos Contratos de Estágio

A Lei nº 11.788, de 25 de setembro de 2008, conhecida como Lei do Estágio (BRASIL, 2008), estabelece os direitos e deveres das partes envolvidas, bem como os requisitos mínimos para que um contrato de estágio seja considerado regular e preserve seu caráter educativo.

2.4.2 Legislação de Proteção de Dados: GDPR e LGPD

O desenvolvimento de sistemas baseados em IA que manipulam documentos jurídicos exige atenção especial às normas de proteção de dados pessoais. Nesse contexto, destacam-se o Regulamento Geral sobre a Proteção de Dados (*General Data*

¹ <<https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash?hl=pt-br>>

² <<https://chatgpt.com/>>

Protection Regulation, ou GDPR) (União Europeia, 2016), da União Europeia, e a Lei Geral de Proteção de Dados (LGPD) (BRASIL, 2018), Lei nº 13.709/2018³, no Brasil.

Ambas as legislações compartilham princípios semelhantes, como a finalidade específica do tratamento de dados, a necessidade, a transparência e a segurança da informação. A LGPD foi fortemente inspirada no GDPR, adaptando seus conceitos à realidade jurídica brasileira. Entre suas diferenças, destaca-se o escopo territorial mais amplo do GDPR e a existência de autoridades reguladoras específicas em cada contexto.

2.5 Tecnologias e Ferramentas Utilizadas

O desenvolvimento do sistema utilizou um conjunto de tecnologias modernas voltadas à criação de aplicações web eficientes e escaláveis.

2.5.1 *Frontend*

O *frontend* é a camada visual e interativa de um site ou aplicativo responsável por estruturar os elementos na tela do usuário, como botões, menus, imagens, cores e formulários, também possibilitando que o design seja funcional, intuitivo e responsivo em diferentes dispositivos, como celulares ou computadores (TOTVS, 2023).

2.5.2 *HyperText Markup Language (HTML)*

O HTML é a linguagem responsável por definir a estrutura de uma página web. HTML não executa lógicas complexas como cálculos, mas organiza todo o conteúdo através de *tags* ou "etiquetas", instruindo ao navegador exatamente o que é cada elemento (onde está um título, um parágrafo, uma imagem ou um link), servindo como a base estrutural sobre a qual o design e a interatividade são aplicados depois (Mozilla Developer Network, 2025b). A sua versão atual é a HTML5.

2.5.3 *JavaScript*

O JavaScript é uma linguagem de programação fundamental para o desenvolvimento web (Mozilla Developer Network, 2024). Enquanto o HTML define a estrutura e o CSS estabelece a aparência, o JavaScript controla o comportamento das páginas, possibilitando a atualização de conteúdos sem recarregar, exibição de mapas interativos, animação de elementos gráficos e validação de formulários. Por possuir suporte nativo nos principais navegadores, o JavaScript desempenha papel essencial no desenvolvimento *frontend*.

³ <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>

2.5.4 Vite

O Vite é uma ferramenta moderna de build utilizada para inicializar, configurar e executar o projeto frontend de forma rápida e eficiente (VITE, 2025). Ele oferece um ambiente de desenvolvimento com recarregamento instantâneo (*hot reload*) e baixo tempo de inicialização. Além disso, o Vite facilita a configuração de variáveis de ambiente, como a URL da API do backend, permitindo que aplicações frontend se comuniquem corretamente com servidores em diferentes ambientes, como local e produção.

2.5.5 React

React é uma biblioteca JavaScript voltada para a criação de interfaces de usuário (REACT, 2025). No React, interfaces são construídas a partir de componentes, e, com suas lógicas separadas, são combinados para criar a interface. Os componentes são reutilizáveis e atualizam automaticamente quando os dados são alterados, proporcionando melhor organização do código e experiência do usuário.

2.5.6 Cascading Style Sheets (CSS)

CSS é a linguagem de design responsável pela apresentação visual e estética de uma página web. O CSS atua sobre a estrutura do HTML para definir como os elementos devem parecer, controlando cores, fontes, espaçamentos, posicionamento e layout, além de permitir que o site se adapte automaticamente a telas de celulares e computadores, seguindo o conceito de responsividade (World Wide Web Consortium, 2024).

2.5.7 Backend

O *backend* é responsável pela lógica, regras de negócio e armazenamento de dados que sustentam o funcionamento do sistema. Enquanto o *frontend* é responsável pela interface e experiência visual do usuário, o *backend* é executado nos servidores, processando as informações, comunicando-se com o banco de dados e assegurando que, ao interagir com a aplicação (por exemplo, ao clicar em um botão), a ação correta seja executada e retornada à interface (Amazon Web Services, 2025a).

2.5.8 Python

O Python é uma linguagem de programação de alto nível, reconhecida por sua versatilidade e por possuir uma sintaxe simples e próxima ao inglês, o que facilita a leitura e o aprendizado (Python Software Foundation, 2024). É amplamente utilizada no de-

envolvimento de *backend* de sites e APIs, além de ser uma das principais linguagens em áreas como Ciência de Dados, Inteligência Artificial e automação de tarefas.

2.5.9 *Application Programming Interface (API)*

Uma *Application Programming Interface (API)* é um conjunto de regras e padrões que permite a comunicação entre diferentes sistemas ou componentes de software, de forma organizada e padronizada (Amazon Web Services, 2024). Por meio da API, um sistema pode solicitar serviços ou dados de outro sistema sem precisar conhecer como ele foi implementado internamente. Entre os usos possíveis de uma API está a interação em que o frontend envia requisições HTTP, como o envio de arquivos ou dados, e o backend as processa, executando a lógica necessária e retornando as respostas em formato JSON.

2.5.10 *JavaScript Object Notation (JSON)*

JSON é um formato de dados baseado em texto, derivado da sintaxe do JavaScript. É amplamente utilizado para transmitir informações em aplicações web (Mozilla Developer Network, 2025c).

2.5.11 *Representational State Transfer (REST)*

REST é uma arquitetura que define uma API com um design desacoplado entre cliente e servidor, e comunicação "*stateless*", ou seja, sem estado entre as requisições. Uma API REST presume que o cliente e servidor não estão diretamente conectados, e que as requisições podem passar por diferentes camadas. Diz-se que uma API é "*RESTful*" se ela segue a arquitetura REST (Amazon Web Services, 2025b; IBM, 2025b).

2.5.12 *Cross-Origin Resource Sharing (CORS)*

CORS é um mecanismo de segurança implementado em navegadores que restringe a maneira em que websites podem fazer requisições a recursos que existem em outros websites. Um dos seus objetivos é evitar que sites maliciosos obtenham dados sigilosos em servidores diferentes do website que faz a requisição (Mozilla Developer Network, 2025a).

2.5.13 *Integrated Development Environment (IDE)*

IDE é um programa de computador voltado ao desenvolvimento de software, reunindo ferramentas como editor de código fonte, compilador, depurador e integração

com sistemas de controle de versão, como o Git, facilitando o desenvolvimento de software.

2.5.14 Visual Studio Code

O Visual Studio Code⁴ ou VS Code é uma IDE desenvolvida pela Microsoft. Ele oferece suporte para várias linguagens de programação, ferramentas integradas de depuração e uma vasta loja de extensões que permitem personalizar a ferramenta.

2.5.15 Repositório de Código Fonte

Um repositório de código fonte é um ambiente digital destinado ao armazenamento, organização e versionamento dos arquivos que compõem um projeto de software, permitindo o acompanhamento das alterações ao longo do tempo, a colaboração entre desenvolvedores e a recuperação de versões anteriores.

Diversos sistemas de controle de versão foram desenvolvidos com esse objetivo, destacando-se o Git, amplamente utilizado por adotar um modelo distribuído, além de soluções mais antigas ou centralizadas, como o CVS (Concurrent Versions System) e o SVN (Subversion), que também permitem o gerenciamento de versões, histórico de alterações e trabalho colaborativo em projetos de software.

2.5.16 GitHub

O GitHub⁵ é uma plataforma de colaboração de projetos de software baseada no Git. O GitHub também inclui ferramentas de *Continuous Deployment* (implantação de código automática em produção depois dos testes) e *issue tracking* (rastreamento de problemas.)

2.5.17 Kanban

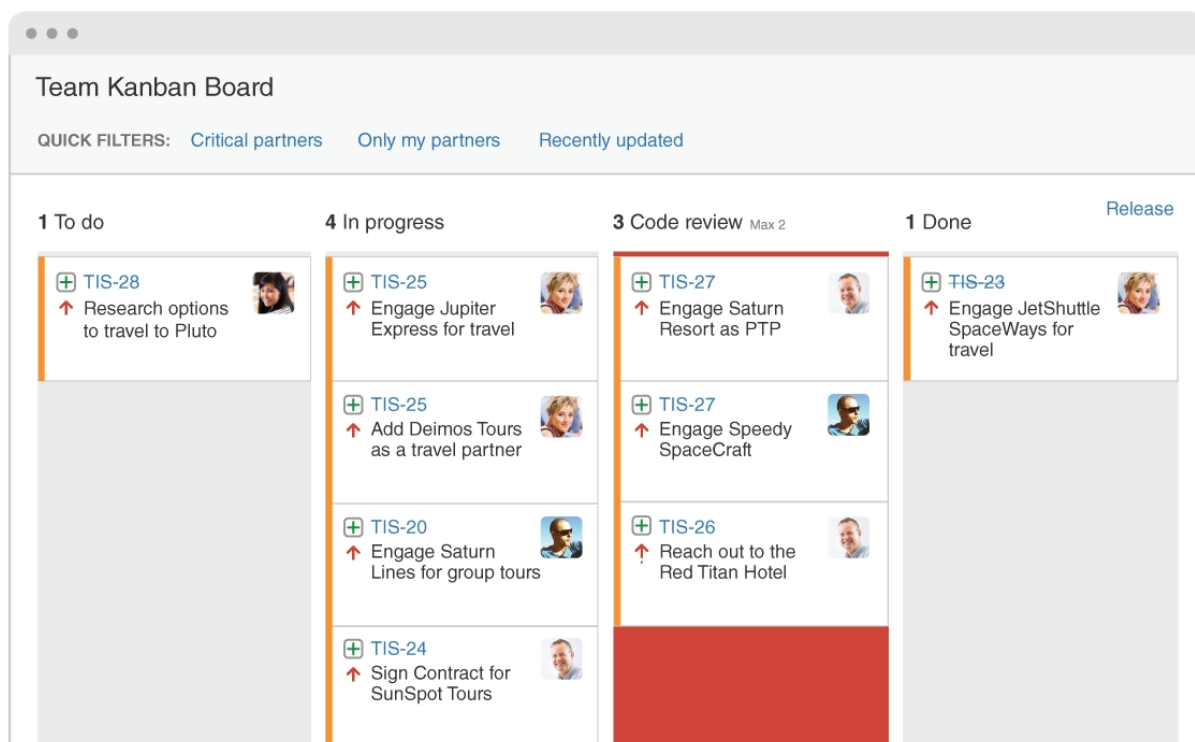
Kanban é uma metodologia ágil de gestão de trabalho voltada para o controle e a melhoria contínua do fluxo de atividades. Sua principal característica é a visualização do processo, permitindo que equipes acompanhem o andamento das tarefas, identifiquem gargalos e melhorem a eficiência operacional de forma incremental. É importante diferenciar a **metodologia Kanban** (com K maiúsculo) do **quadro kanban** (com k minúsculo). A metodologia Kanban consiste em um conjunto de princípios, práticas e regras que orientam a gestão do fluxo de trabalho, como a limitação do trabalho em progresso, a melhoria contínua e a gestão baseada em métricas. Já o quadro kanban é

⁴ <<https://code.visualstudio.com/>>

⁵ <<https://github.com/>>

apenas uma ferramenta visual utilizada para representar esse fluxo, normalmente dividida em colunas como “A Fazer”, “Em Andamento” e “Concluído”. A origem do Kanban remonta à década de 1940, quando foi desenvolvido na Toyota, no Japão, como parte do Sistema Toyota de Produção. Seu objetivo inicial era otimizar processos industriais, reduzindo desperdícios e equilibrando a produção conforme a demanda. Com o tempo, seus princípios foram adaptados para áreas como desenvolvimento de software, gestão de projetos e serviços. Dentro do contexto das metodologias ágeis, o Kanban pode ser utilizado de forma independente ou combinado com outras abordagens, como o Scrum. Enquanto o Scrum trabalha com ciclos de tempo fixos (sprints), o Kanban foca no fluxo contínuo de trabalho, oferecendo maior flexibilidade e adaptação às mudanças ao longo do projeto (Atlassian, 2024). A Figura 1 ilustra um exemplo de quadro kanban, demonstrando visualmente a movimentação das tarefas entre as etapas do processo, facilitando o acompanhamento do progresso e a identificação de possíveis gargalos.

Figura 1 – Exemplo de quadro Kanban



Fonte: (Atlassian, 2024)

2.5.18 Unified Modeling Language (UML)

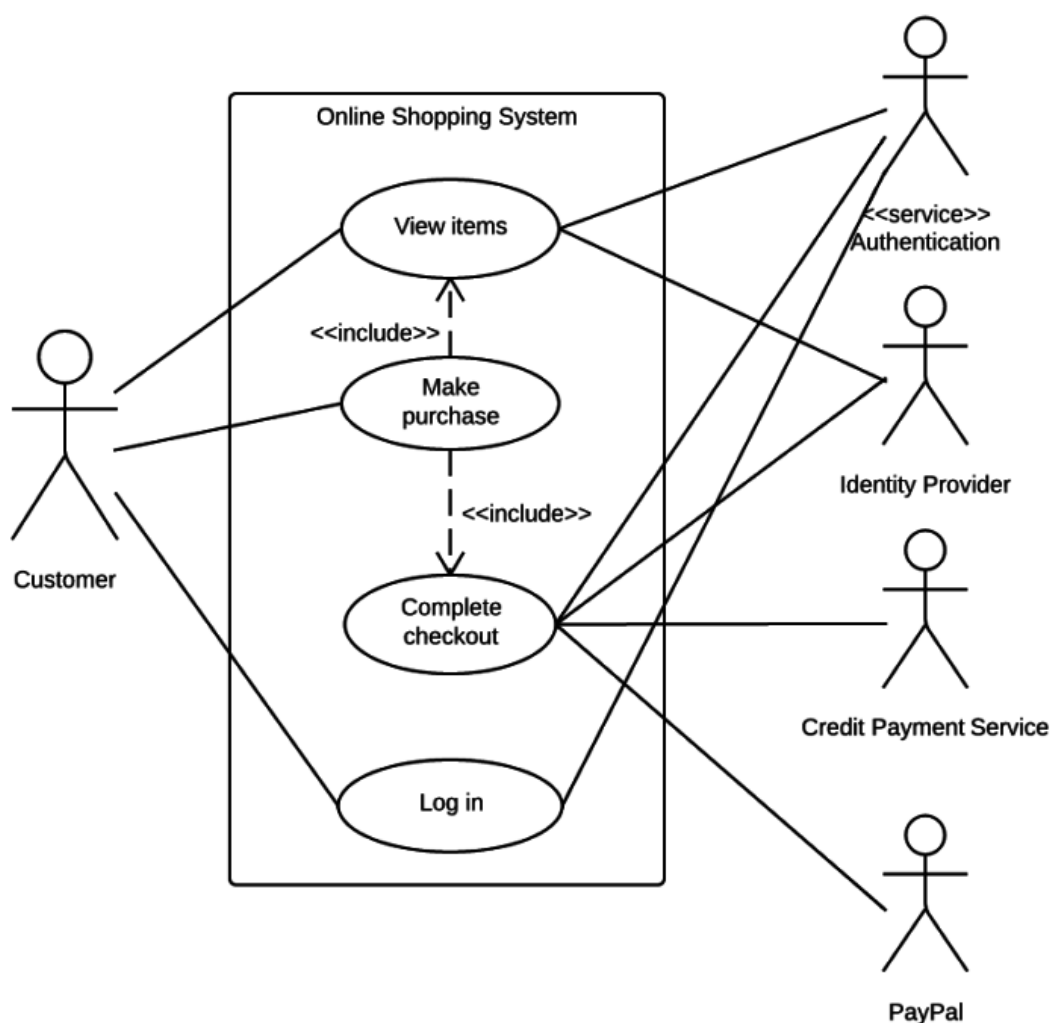
A *Unified Modeling Language* (UML) é uma linguagem padrão de modelagem visual utilizada na Engenharia de Software para especificar, visualizar e documentar sistemas computacionais de forma padronizada (BOOCH; RUMBAUGH; JACOBSON,

2005). Por meio de diagramas, a UML auxilia na compreensão da estrutura, do comportamento e das interações entre os componentes de um sistema.

Entre os principais diagramas da UML, destacam-se:

- **Diagrama de Casos de Uso:** representa as principais funcionalidades do sistema sob a perspectiva do usuário, ilustrando de forma clara e intuitiva as interações entre os atores e o sistema. Esse diagrama facilita a compreensão dos requisitos funcionais, define os limites do sistema e serve como base para o desenvolvimento, validação e alinhamento das funcionalidades propostas.

Figura 2 – Diagrama de Casos de Uso

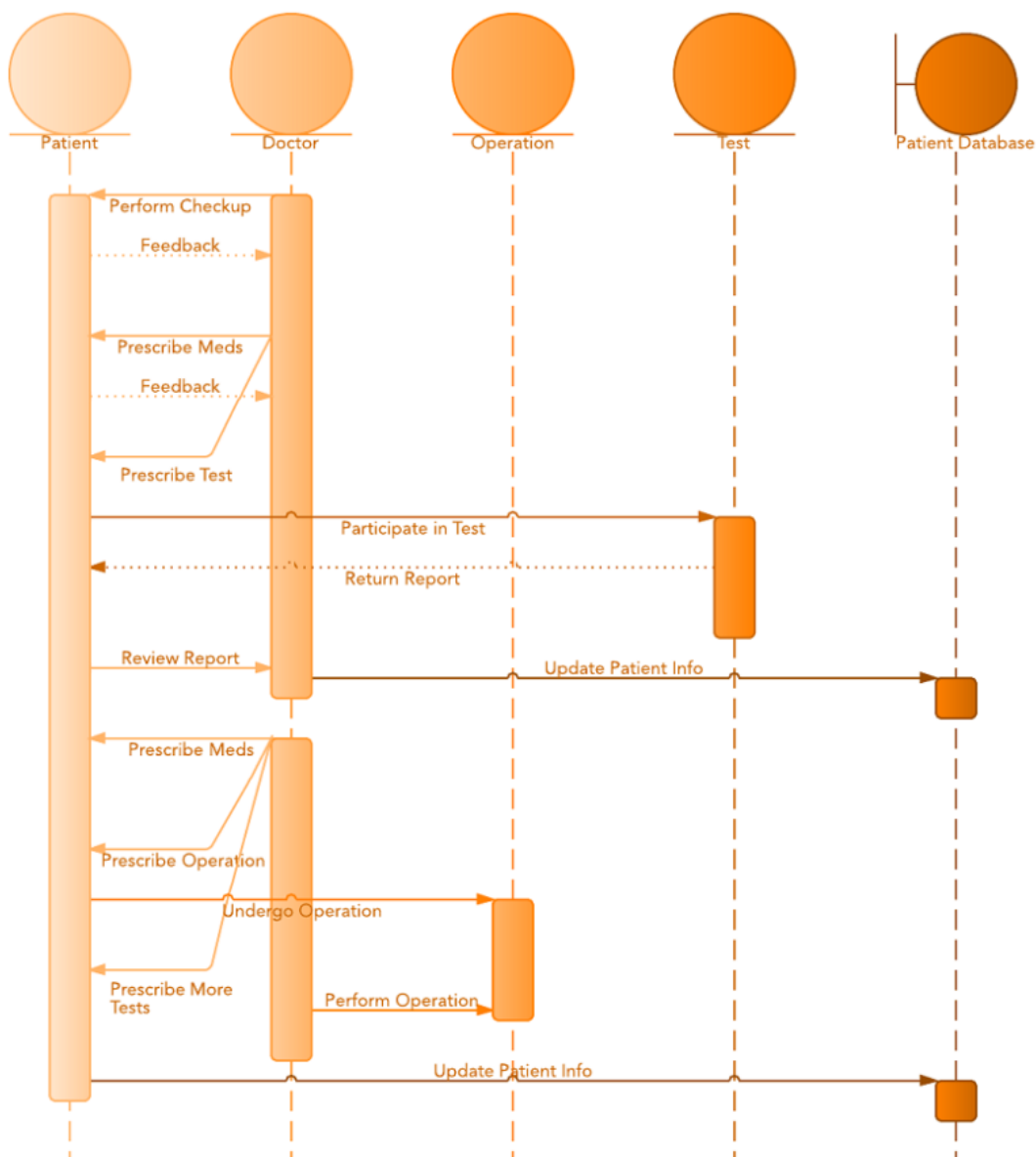


Fonte: (LUCIDCHART, 2025)

- **Diagrama de Sequência:** representa a ordem temporal das trocas de mensagens entre os componentes do sistema, evidenciando como os objetos interagem ao longo do tempo. Permite visualizar o fluxo de execução de cada funcionalidade, detalhando chamadas, respostas e dependências. Auxilia na identificação

de falhas lógicas e gargalos de comunicação. Serve como apoio essencial para o desenvolvimento e a validação do comportamento do sistema.

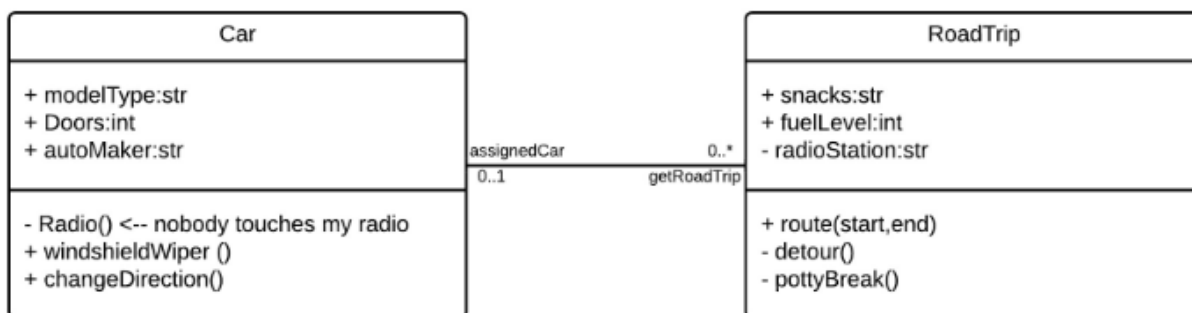
Figura 3 – Diagrama de Sequência



Fonte: (LUCIDCHART, 2025)

- **Diagrama de Classes:** modela a estrutura estática do sistema ao evidenciar as classes, seus atributos e métodos, bem como os relacionamentos entre elas. Permite compreender a organização interna do sistema e a distribuição das responsabilidades. Facilita a manutenção e a evolução do software. Serve como referência fundamental para a implementação orientada a objetos.

Figura 4 – Diagrama de Classes



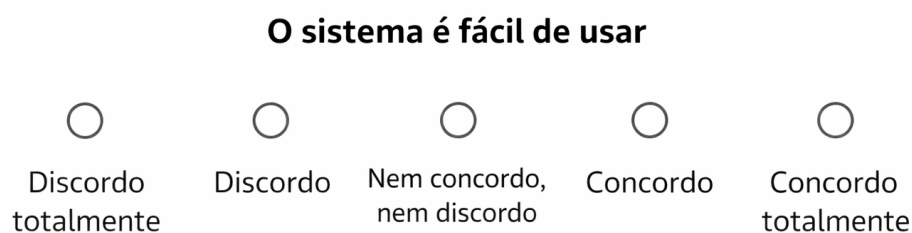
Fonte: (LUCIDCHART, 2025)

2.5.19 Escala Likert

Escala Likert é uma técnica muito utilizada em pesquisas para medir a intensidade de opiniões, atitudes ou comportamentos. Diferente de respostas simples, como "sim" ou "não", ela oferece uma gradação de opções, geralmente entre 5 ou 7 pontos, permitindo que o respondente indique o quanto concorda ou discorda de uma afirmação. Por exemplo, as alternativas podem variar de "Discordo totalmente" até "Concordo totalmente" (BERMUDES et al., 2016).

A Figura 5 ilustra um exemplo clássico de utilização da Escala Likert, variando de "Discordo totalmente" a "Concordo totalmente".

Figura 5 – Exemplo de aplicação da Escala Likert



Fonte: Autoria própria

2.5.20 *Minimum Viable Product (MVP)*

Um MVP, ou Produto Mínimo Viável é a versão mais básica de um produto, contendo apenas as funcionalidades essenciais. O seu propósito principal é testar a proposta de valor do produto e atrair clientes, além de coletar opiniões dos usuários (ATLASSIAN, 2025).

2.5.21 *Business Process Model and Notation (BPMN)*

BPMN é um padrão internacional de modelagem de processos de negócios que facilita a visualização e a otimização dos fluxos de trabalho, além de simplificar a análise e o aprimoramento das operações ao oferecer uma linguagem comum entre desenvolvedores, analistas de negócios e stakeholders (IBM, 2025a).

3 Metodologia

Neste capítulo, apresentam-se os procedimentos metodológicos adotados na execução desta pesquisa e no desenvolvimento do sistema NoRisk. A metodologia compreende etapas de planejamento, coleta de dados, construção do protótipo, testes e gestão do projeto, garantindo a validade, confiabilidade e reprodutibilidade dos resultados.

Segundo Gil (GIL, 2008), a metodologia é o conjunto de técnicas e métodos que orientam o pesquisador desde a formulação do problema até a apresentação das conclusões, assegurando coerência e rigor científico ao estudo.

3.1 Classificação da Pesquisa

Esta pesquisa foi conduzida com base em metodologias consolidadas na literatura de pesquisa científica aplicada à Engenharia de Software e Ciências Sociais, conforme Gil (GIL, 2008) e Creswell (CRESWELL, 2014).

De acordo com esses autores, a pesquisa pode ser classificada da seguinte forma:

- **Quanto aos objetivos:** exploratória, pois busca compreender e analisar a aplicação de técnicas de IA e PLN na análise de contratos jurídicos, identificando padrões de risco e oportunidades de melhoria. A escolha por pesquisa exploratória se justifica pela ausência de estudos consolidados sobre ferramentas de IA voltadas especificamente para contratos de estágio, o que exige investigação inicial e detalhada do problema (GIL, 2008).
- **Quanto à abordagem:** mista, combinando qualitativa e quantitativa. A abordagem qualitativa foi aplicada na interpretação das cláusulas contratuais e no estudo do comportamento do modelo de IA durante a análise de contratos, enquanto a abordagem quantitativa foi utilizada na coleta de dados por meio de questionários estruturados com estudantes, bacharéis e profissionais do Direito. Esta combinação permite avaliar tanto a precisão técnica do sistema quanto a percepção dos usuários sobre sua utilidade (CRESWELL, 2014).
- **Quanto à natureza:** aplicada, pois a pesquisa tem caráter prático, com objetivo de desenvolver uma solução tecnológica funcional — o sistema NoRisk — capaz de automatizar a análise de contratos de estágio e verificar conformidade com a legislação vigente, especialmente a Lei nº 11.788/2008 (BRASIL, 2008). A pesquisa aplicada busca gerar resultados imediatos para solução de proble-

mas reais, diferentemente da pesquisa teórica, que se concentra na produção de conhecimento abstrato (GIL, 2008).

Dessa forma, a classificação da pesquisa justifica-se tanto pela necessidade de exploração do problema quanto pela aplicação prática de tecnologias inovadoras aliadas a métodos científicos consolidados.

3.2 Etapas da Metodologia

O desenvolvimento do projeto seguiu um conjunto estruturado de etapas metodológicas, descritas a seguir:

- **Revisão Bibliográfica:** Realizou-se levantamento de estudos, artigos científicos e legislações relacionadas ao uso de IA e PLN no campo jurídico, à área de *Legal Tech* e às normas que regem contratos de estágio, incluindo a Lei nº 11.788/2008. Essa etapa permitiu fundamentar teoricamente o desenvolvimento do sistema e definir critérios de análise contratual.
- **Coleta de Dados:** Os dados foram obtidos por meio de um questionário estruturado criado na plataforma *Google Forms*, aplicado a estudantes universitários, bacharéis e profissionais da área jurídica. O objetivo foi identificar dificuldades na leitura e interpretação de contratos de estágio e avaliar o interesse no uso de ferramentas digitais baseadas em IA (CRESWELL, 2014). Os resultados serviram de base para definição de requisitos funcionais e não funcionais do sistema.
- **Desenvolvimento do Protótipo:** O protótipo NoRisk foi concebido em etapas interligadas:
 1. **Arquitetura do Sistema:** definição dos módulos principais: *upload* de arquivos, processamento de contratos, análise automatizada e geração de relatórios de integridade.
 2. **Modelagem de IA:** integração de modelos de linguagem generativos (API Gemini 2.5 Flash) e técnicas de PLN para identificar cláusulas obrigatórias e desejáveis, avaliar conformidade legal e gerar relatórios claros e estruturados.
 3. Implementação do **Frontend e Backend:** desenvolvimento da interface web com *React*, *Vite*, *HTML*, *CSS* e *JavaScript*, comunicando-se com o backend em *Python* utilizando *FastAPI* e *middleware CORS*. Essa integração possibilitou *upload* de arquivos PDF e DOCX, envio ao modelo de IA e retorno do relatório de análise ao usuário.

4. **Testes e Validação:** O sistema passou por testes unitários, de integração e de usabilidade, incluindo:
- Avaliação da precisão do modelo de IA na detecção de cláusulas e conformidade legal;
 - Testes de performance e escalabilidade da API;
 - Validação da interface com usuários simulando *upload* de contratos e conferindo a apresentação dos relatórios.

3.3 Gestão do Projeto

A gestão do projeto foi conduzida de forma colaborativa entre os membros do grupo, com reuniões semanais para acompanhamento do progresso e definição de metas. Para o gerenciamento das atividades e controle do cronograma, foi utilizada a ferramenta **Trello**, estruturada com base na metodologia ágil *Kanban*.

O quadro foi dividido nas colunas: “A Fazer”, “Em Andamento” e “Concluído”, permitindo a visualização clara das tarefas e responsabilidades de cada integrante. O progresso do projeto foi registrado mês a mês, conforme ilustrado nas figuras a seguir.

3.3.1 Evolução do Projeto no Trello

A Figura 6 representa o quadro de tarefas do mês de setembro.

Figura 6 – Quadro Trello – Setembro: pendências de documentação e início do protótipo

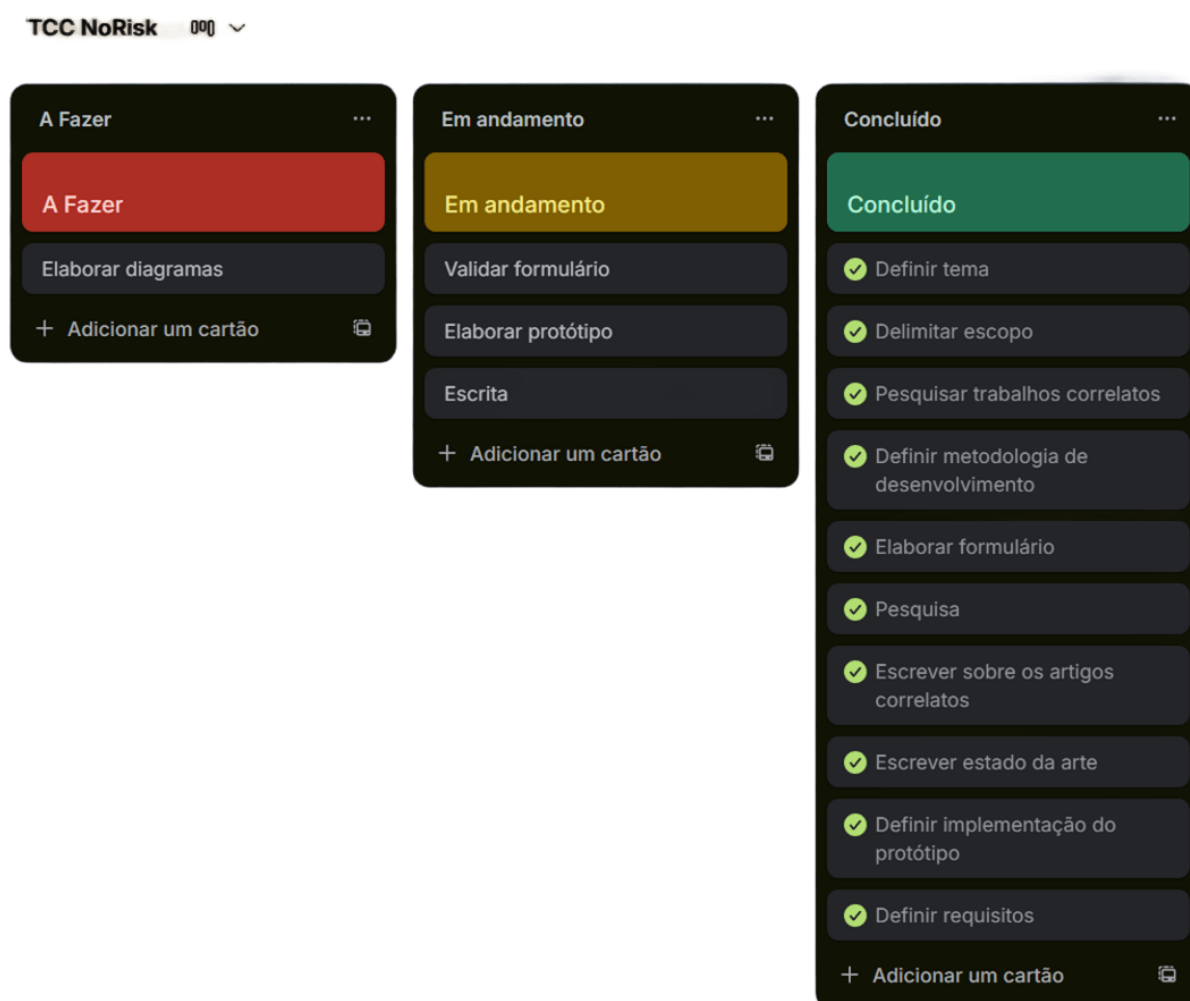


Fonte: Autoria própria

A partir da figura 6, percebe-se que estavam pendentes as atividades de documentação, desenvolvimento inicial do protótipo e elaboração dos primeiros diagramas do sistema.

A figura 7 representa o quadro de tarefas do mês de outubro.

Figura 7 – Quadro Trello – Outubro: finalização do desenvolvimento do sistema

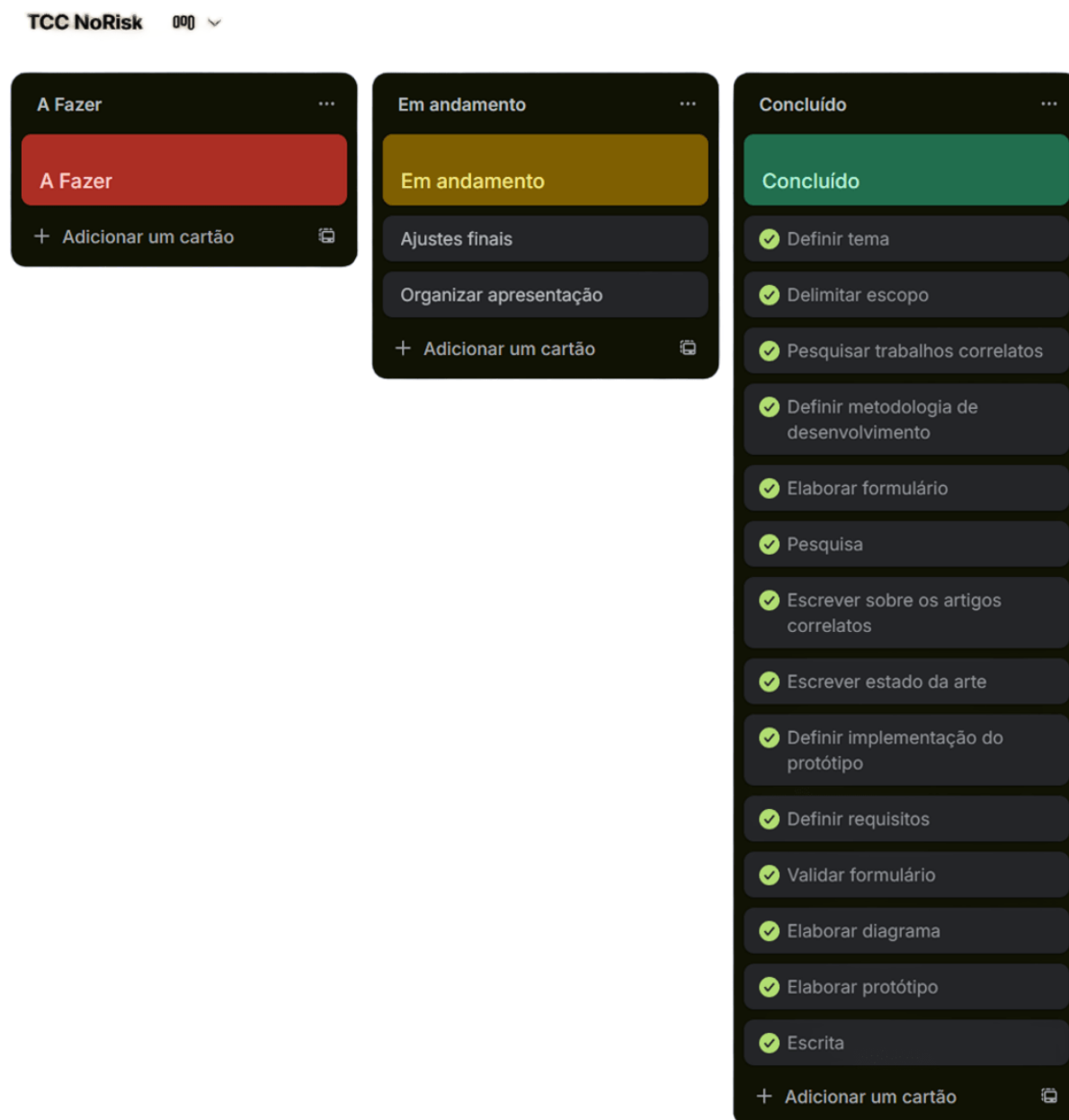


Fonte: Autoria própria

A partir da figura 7, nota-se que a equipe concentrou esforços na finalização do sistema e na integração dos módulos de análise automática, consolidando a parte técnica do projeto.

A figura 8 representa o quadro de tarefas do mês de novembro.

Figura 8 – Quadro Trello – Novembro: ajustes finais e preparação da apresentação



Fonte: Autoria própria

A partir da figura 8, observa-se que o projeto estava praticamente concluído, restando apenas ajustes finais e a definição da estrutura de apresentação e documentação final do trabalho.

3.4 Aspectos Éticos da Pesquisa

A pesquisa respeitou todos os princípios éticos aplicáveis a estudos com seres humanos, conforme a Resolução nº 510/2016 do Conselho Nacional de Saúde ((BRASIL), 2016). A participação foi voluntária e anônima, e os dados foram utilizados exclu-

sivamente para fins acadêmicos, garantindo o cumprimento das diretrizes da Lei Geral de Proteção de Dados Pessoais (LGPD) (BRASIL, 2018).

3.5 Apresentação dos Resultados

Os dados obtidos por meio do formulário foram analisados de forma descritiva e interpretativa, subsidiando o desenvolvimento do sistema NoRisk. As informações coletadas permitiram identificar o nível de compreensão dos participantes sobre contratos de estágio, validar o problema de pesquisa e direcionar a modelagem das funcionalidades da ferramenta. Os resultados detalhados estão disponíveis no Apêndice B.

3.6 Coleta e Análise de Dados

Para a coleta de requisitos e validação do trabalho, um formulário foi criado na plataforma Google Forms¹. Foram elaboradas 10 questões em formato de Escala Likert e com base na Lei do Estágio. As respostas foram organizadas em uma escala de 1 a 5, com o valor 1 representando discordância total e o 5 correspondendo à concordância total. Os valores intermediários (de 2 a 4) equivalem a “pouco importante”, “neutro” e “importante”, respectivamente. O questionário foi aberto em 11 de outubro de 2025, e divulgado para profissionais, bacharéis e estudantes, todos da área de Direito. No dia 13 de novembro de 2025, a coleta de respostas foi encerrada sem continuidade e, ao todo, 22 participantes responderam à pesquisa.

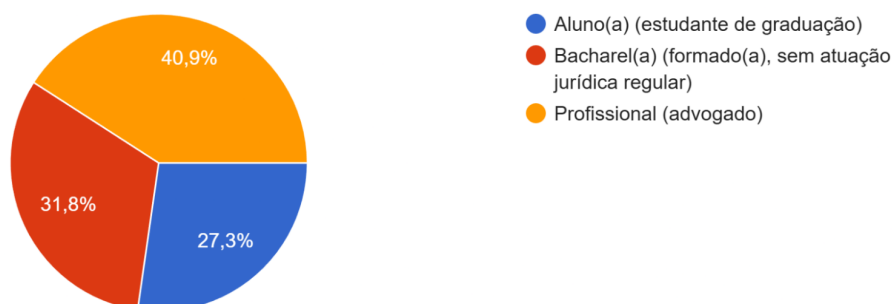
Em conformidade com a Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018), todos os participantes foram explicitamente informados, através do próprio formulário de coleta, sobre o caráter anônimo da participação e a finalidade acadêmica do uso das informações, manifestando o consentimento ao enviar as respostas.

A figura 9 mostra a divisão dos respondentes da pesquisa.

¹ <<https://forms.google.com/>>

Figura 9 – Distribuição dos tipos de respondentes

Tipo de respondente
22 respostas

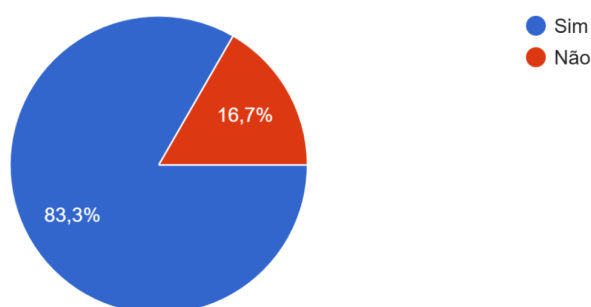


Fonte: Autoria própria

A figura 9 permite observar que 40,9% dos respondentes se identificaram como profissionais, 31,8% como bacharéis e 27,3% como alunos, sendo assim, apesar do baixo número de respondentes, a relevância da pesquisa é sustentada pela maioria dos participantes ser composta por advogados que atuam na área. Além disso, também para fins de validação da amostra, a Figura 10 ilustra como os alunos estão distribuídos de acordo com o curso prévio da disciplina de Direito do Trabalho.

Figura 10 – Distribuição dos alunos quanto ao curso prévio da disciplina de Direito do Trabalho

Você já cursou a disciplina de Direito do Trabalho?
6 respostas



Fonte: Autoria própria

Conforme a figura 10, nota-se que apenas um aluno não cursou a disciplina de Direito do Trabalho, assim sendo, sua resposta foi descartada, totalizando 21 respostas para as futuras questões.

Para a análise dos resultados, os requisitos foram enumerados de maneira sequencial:

R1: Verificar a existência de um Termo de Compromisso formalizado entre o estudante, a empresa e a instituição de ensino.

R2: Checar se o contrato atesta a matrícula e frequência regular do estudante.

R3: Identificar a cláusula que estabelece a contratação de seguro contra acidentes pessoais para o estagiário.

R4: Checar se os nomes do supervisor da parte concedente e do professor orientador da instituição de ensino estão designados no termo.

R5: Verificar se a carga horária descrita no contrato é compatível com o nível de ensino do estudante, respeitando os limites legais específicos (ex: 4h/dia e 20h/semana para educação especial; 6h/dia e 30h/semana para ensino superior; ou até 40h/semana para cursos que alternam teoria e prática).

R6: Calcular o período entre a data de início e a de término do contrato para garantir que não exceda o limite de 2 anos, exceto quando se tratar de estagiário portador de deficiência.

R7: Verificar se há uma seção ou anexo denominado "Plano de Atividades" com a descrição das tarefas que serão realizadas pelo estagiário.

R8: Verificar, para os estágios do tipo não obrigatório, a inclusão das cláusulas compulsórias de Bolsa, Auxílio-Transporte e a previsão do Recesso Remunerado.

R9: Verificar a existência da cláusula que assegura o direito ao recesso de 30 dias (ou proporcional), benefício obrigatório para todos os tipos de estágio.

R10: Buscar por uma cláusula que mencione a obrigação do estagiário de entregar relatórios de atividades em um período não superior a 6 meses.

Após a análise das respostas obtidas, realizou-se a classificação dos requisitos em duas categorias: obrigatórios e desejáveis. Essa etapa teve como objetivo organizar os resultados de acordo com o grau de importância atribuído pelos participantes, facilitando a priorização das funcionalidades. Para essa categorização, considerou-se a distribuição das respostas na Escala Likert, adotando-se como critério o percentual de concordância máxima. Os itens que obtiveram mais de 50% das respostas na categoria 5, correspondente à opção "essencial", foram classificados como obrigatórios, por representarem funcionalidades consideradas indispensáveis para o funcionamento adequado do sistema. Por outro lado, os itens que não atingiram esse percentual foram classificados como desejáveis, indicando funcionalidades relevantes, porém não críticas, que podem ser implementadas em etapas posteriores, conforme a evolução do sistema. A figura 11 ilustra a porcentagem de essencialidade de cada requisito.

Figura 11 – Distribuição da essencialidade por requisito



Fonte: Autoria própria

A figura 11 evidencia que é obrigatório abranger a formalização do Termo de Compromisso (66,70%), a comprovação de matrícula e frequência (52,40%), a adequação da carga horária aos limites legais (71,40%), o limite de duração do contrato (57,10%), as cláusulas de bolsa e auxílio-transporte (52,40%) e a garantia do recesso remunerado (66,70%).

Ademais, os dados apresentados indicam que é apenas desejável verificar a cláusula de seguro contra acidentes pessoais (47,60%), a designação nominal do supervisor e do orientador (47,60%), a existência do Plano de Atividades (33,30%) e a obrigação de entrega dos relatórios (23,80%).

Consequentemente, para o desenvolvimento do protótipo, estabeleceu-se cobertura total dos requisitos obrigatórios. Os requisitos desejáveis foram considerados na modelagem do sistema, mas a sua implementação foi tratada como complementar, de modo que a ausência ou eventuais instabilidades na execução destes itens não invalidam o MVP.

4 NoRisk

O sistema denominado NoRisk foi desenvolvido como uma ferramenta baseada em Inteligência Artificial voltada à análise de contratos de estágio, com foco na identificação de riscos jurídicos e verificação de integridade documental.

4.1 Requisitos do Sistema

Os requisitos do sistema NoRisk foram definidos com base nos objetivos do projeto e nas necessidades observadas durante o levantamento de dados. Eles foram divididos em duas categorias principais: requisitos funcionais e não funcionais. Segundo Sommerville (SOMMERVILLE, 2011), os requisitos representam o conjunto de propriedades e restrições que descrevem o comportamento esperado de um sistema de software, servindo como guia para seu desenvolvimento, validação e manutenção.

4.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, ou seja, as funcionalidades esperadas e os comportamentos observáveis do sistema durante sua operação.

4.1.1.0.1 Módulo: *Upload* e Processamento

- **RF001:** O sistema deve permitir que o usuário faça o *upload* de um documento de contrato de estágio para análise.
- **RF002:** O sistema deve ser capaz de extrair o texto do documento enviado para processamento, aplicando técnicas de Processamento de Linguagem Natural (PLN) (JURAFSKY; MARTIN, 2023).

4.1.1.0.2 Módulo: Análise de Cláusulas

- **RF003:** O sistema deve identificar e classificar automaticamente as seções contratuais.
- **RF004:** O sistema deve categorizar, no mínimo, as cláusulas de:
 - a) Jornada de trabalho;
 - b) Valor da bolsa-auxílio;

- c) Recesso remunerado;
- d) Seguro contra acidentes pessoais.
- **RF005:** O sistema deve comparar as cláusulas com os requisitos da Lei nº 11.788/2008 (Lei do Estágio) (BRASIL, 2008).
- **RF006:** O sistema deve identificar e sinalizar a ausência de cláusulas obrigatórias.
- **RF007:** O sistema deve sinalizar inconsistências legais claras (exemplo: carga horária superior à permitida).
- **RF008:** O sistema deve utilizar modelos de aprendizado de máquina para detectar anomalias e cláusulas de risco, conforme (DIKMEN et al., 2025; KAZEMI; ALVANCHI, 2025).
- **RF009:** O sistema deve gerar uma tradução em linguagem simples das cláusulas jurídicas analisadas, com foco na acessibilidade do conteúdo.

4.1.1.0.3 Módulo: Relatório e Visualização

- **RF010:** O sistema deve gerar um relatório final consolidando todos os achados da análise.
- **RF011:** O relatório deve associar as explicações em linguagem simples (RF009) a cada cláusula.
- **RF012:** O relatório deve destacar visualmente inconsistências e ausências de cláusulas obrigatórias.
- **RF013:** O relatório deve apresentar os potenciais riscos detectados (RF008).

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais determinam as qualidades do sistema, como desempenho, segurança e usabilidade (PRESSMAN, 2014).

4.1.2.0.1 Desempenho e Eficiência

- O sistema deve processar a análise de contratos em tempo adequado para aplicações web, garantindo resposta em poucos segundos após o envio do documento.
- O backend deve realizar o processamento de forma assíncrona, evitando bloqueios na interface do usuário.

4.1.2.0.2 Escalabilidade

- A arquitetura baseada em API REST deve possibilitar a ampliação futura da capacidade de processamento, inclusive mediante integração com serviços em nuvem.

4.1.2.0.3 Disponibilidade

- O sistema deve estar disponível para acesso contínuo por meio de um navegador web.
- Análises devem ser idempotentes, produzindo resultados consistentes.

4.1.2.0.4 Usabilidade e Experiência do Usuário

- Deve ser compatível com arquivos PDF e DOCX.
- A interface deve ser responsiva, intuitiva e de fácil navegação, permitindo o uso por usuários leigos.
- O processo de envio e análise do contrato deve ocorrer em poucos passos, reduzindo a complexidade da interação.

4.1.2.0.5 Precisão e Qualidade da IA

- O sistema deve fornecer análises coerentes e alinhadas à legislação vigente aplicada aos contratos de estágio.
- As respostas geradas pelo modelo de IA devem apresentar linguagem clara, objetiva e apropriada ao público não especialista.

4.2 Regras de Negócio (RN)

As regras de negócio definem as políticas, restrições e lógicas que guiam o comportamento do sistema, garantindo conformidade com o domínio jurídico e coerência na interpretação dos resultados.

4.2.0.0.1 RN: Conformidade Legal (Core)

- **RN001:** A principal base de verificação é a Lei nº 11.788/2008.
- **RN002:** O sistema pode incorporar novas normativas à medida que a base de conhecimento for atualizada.

- **RN003:** Uma cláusula será inconsistente se exceder os limites de carga horária previstos em lei.
- **RN004:** A ausência de cláusula de seguro contra acidentes deve ser tratada como não conformidade grave.
- **RN005:** A ausência de recesso remunerado deve ser considerada não conformidade.

4.2.0.0.2 RN: Análise de Risco (IA/ML)

- **RN006:** As detecções de risco devem ser baseadas em modelos de aprendizado supervisionado (BROWN; MANN; RYDER, 2020).
- **RN007:** Padrões fora do comum devem ser classificados como “pontos de atenção”.
- **RN008:** O escore de risco deve refletir a gravidade e quantidade de anomalias detectadas.

4.2.0.0.3 RN: Interface e Usuário (UX/Usabilidade)

- **RN009:** A tradução de cláusulas deve explicar direitos e deveres de forma acessível.
- **RN010:** O relatório deve ser compreensível para usuários não especialistas.
- **RN011:** O aviso legal deve ser apresentado antes da visualização do relatório.

4.2.0.0.4 RN: Sistema (Geral)

- **RN012:** O sistema não deve aprovar ou reprovar contratos, apenas sinalizar riscos.
- **RN013:** O sistema deve funcionar como protótipo validado comparativamente com análises humanas.
- **RN014:** O escopo da análise é restrito a contratos de estágio brasileiros.

4.3 Ferramentas e Tecnologias Utilizadas

O desenvolvimento do sistema NoRisk foi realizado com base em uma arquitetura web moderna e modular, empregando tecnologias amplamente utilizadas no mercado de software e em projetos de Inteligência Artificial. A escolha das ferramentas considerou aspectos como acessibilidade, escalabilidade, compatibilidade com a IA e facilidade de integração entre componentes.

No projeto NoRisk, o PLN é empregado para analisar contratos de estágio em formato textual, permitindo a identificação de cláusulas obrigatórias, a verificação de conformidade legal e a apresentação de resultados de forma compreensível para usuários não especialistas.

A API Gemini foi integrada ao *backend* do sistema para realizar a análise semântica dos contratos. A partir de um *prompt* cuidadosamente estruturado, o modelo recebe o texto do contrato e avalia sua conformidade com os requisitos legais, retornando um relatório que classifica o contrato como íntegro ou não íntegro, além de indicar pontos críticos, observações e requisitos atendidos. O uso de um modelo generativo elimina a necessidade de treinamento local de algoritmos, possibilitando maior escalabilidade e redução de custos computacionais. A escolha pelo Gemini levou em conta sua capacidade técnica e a viabilidade de aplicação em ambiente acadêmico, visto que, diferentemente da API do ChatGPT, o Gemini pôde ser explorado em caráter experimental graças aos créditos gratuitos disponibilizados oficialmente pela plataforma Google AI Studio¹, o que viabilizou a realização dos testes sem custos adicionais.

A validação jurídica realizada pelo sistema NoRisk baseia-se principalmente na Lei nº 11.788, de 25 de setembro de 2008, conhecida como Lei do Estágio (BRASIL, 2008). Essa legislação estabelece os direitos e deveres das partes envolvidas, bem como os requisitos mínimos para que um contrato de estágio seja considerado regular e preserve seu caráter educativo.

Entre os principais elementos verificados pelo sistema estão a existência do termo de compromisso entre estagiário, instituição de ensino e concedente, a definição da carga horária compatível com o nível de ensino, a duração máxima do estágio, a concessão de bolsa e auxílio-transporte nos casos de estágio não obrigatório, o direito ao recesso remunerado e a previsão de supervisão e acompanhamento pedagógico.

A análise automatizada do contrato consiste na comparação do conteúdo textual com essas exigências legais, permitindo a identificação de cláusulas ausentes ou inadequadas. Dessa forma, o sistema contribui para a redução de riscos jurídicos e para a proteção dos direitos dos estudantes.

A IDE adotada foi o Visual Studio Code, escolhida por sua ampla compatibilidade com as linguagens utilizadas, extensibilidade e suporte a ferramentas de versiona-

¹ <<https://aistudio.google.com/>>

mento e depuração.

No *frontend*, foram empregadas as linguagens HTML e JavaScript, com o uso da biblioteca React para construção da interface do usuário. Para a inicialização e configuração do projeto *frontend*, foi utilizado o Vite, uma ferramenta moderna que oferece um ambiente de desenvolvimento rápido e eficiente, além de suporte nativo ao React.

No *backend*, o sistema foi implementado utilizando a linguagem Python e o framework FastAPI². O FastAPI foi escolhido por seu alto desempenho, tipagem explícita e facilidade de integração com APIs REST. Para permitir a comunicação entre o *frontend* React e o *backend* Python, foi configurado um middleware de CORS, garantindo que requisições feitas a partir do navegador sejam aceitas de forma segura.

4.3.1 Controle de Versão e Gestão de Projeto

A gestão do desenvolvimento do sistema NoRisk foi realizada com o apoio da plataforma **Trello**, com base no quadro kanban. Essa abordagem permitiu a organização das atividades em quadros visuais, facilitando a definição de tarefas, o acompanhamento do progresso e o estabelecimento de prazos realistas ao longo das etapas do projeto.

Adicionalmente, o uso do Kanban possibilitou uma adaptação contínua das atividades, permitindo ajustes conforme novas demandas surgiam durante o desenvolvimento, característica fundamental em projetos de software (BECK et al., 2001).

Como ferramenta de controle de versão, foi utilizado o **Git**, que possibilitou o versionamento do código-fonte e o registro histórico de todas as alterações realizadas. A utilização do Git permitiu acompanhar a evolução incremental do sistema, reduzir riscos de perda de código e facilitar a correção de erros por meio da recuperação de versões anteriores.

Além disso, o repositório remoto no GitHub serviu como base central de armazenamento e colaboração do projeto.

O backend, desenvolvido em *Python* com *FastAPI*, foi implantado na plataforma **Render**. Essa escolha permitiu a configuração de servidores e execução da API de forma escalável e automatizada.

Através do uso de *Continuous Deployment* (CD), qualquer atualização no código-fonte enviada ao repositório GitHub disparava automaticamente o processo de build e deploy no Render, garantindo que novas funcionalidades, correções de bugs ou ajustes de configuração fossem refletidos rapidamente no ambiente de produção sem intervenção manual.

Paralelamente, o frontend foi desenvolvido com React e estruturado com o Vite, que foi utilizado para criar, configurar e construir a aplicação de forma rápida e efici-

² <<https://fastapi.tiangolo.com/>>

ente. O Vite facilitou o desenvolvimento local e a geração dos arquivos finais do site, além de permitir o uso de variáveis de ambiente para definir a URL da API do backend, garantindo a comunicação correta entre as partes do sistema. Esse frontend foi publicado na plataforma Vercel, que possui integração direta com o GitHub, fazendo com que cada commit na branch principal gerasse automaticamente uma nova versão do site online. Dessa forma, a interface web consegue consumir a API REST hospedada no Render de maneira transparente, sem a necessidade de ajustes manuais de URLs ou servidores locais.

Assim sendo, o uso combinado de ferramentas de gestão ágil e controle de versão contribuiu para um desenvolvimento mais organizado, rastreável e confiável, além de possibilitar que o sistema fosse disponibilizado gratuitamente de forma online, ampliando seu alcance e aplicabilidade.

O objetivo principal da implementação de um Front-end para este projeto foi traduzir a complexidade dos dados e da lógica do sistema (Back-end) em uma interface visual que seja intuitiva, agradável e funcional para o ser humano.

Sem um Front-end, o usuário teria que lidar com linhas de código ou terminais de comando. Portanto, o objetivo é garantir a Usabilidade (facilidade de uso) e a Acessibilidade (funcionar bem em celulares, tablets e computadores).

4.3.2 Frontend: Interface do Usuário

A camada de interface foi desenvolvida utilizando um conjunto de tecnologias modernas da web, garantindo interatividade, responsividade e integração com o backend:

- **HTML5:** utilizado para estruturar o conteúdo da página, definindo elementos semânticos e garantindo compatibilidade entre navegadores;
- **CSS3:** empregado para estilização, cores, tipografia, layout responsivo e efeitos visuais, assegurando boa experiência de uso em diferentes dispositivos;
- **JavaScript (JS):** responsável pelas interações dinâmicas, validações de *upload*, manipulação do DOM e lógica de exibição de resultados;
- **React.js:** biblioteca para construção de interfaces reativas e componentizadas, facilitando o gerenciamento do estado da aplicação e a reutilização de componentes;
- **JSON:** formato de dados baseado em texto, utilizado para estruturar informações e permitir comunicação entre sistemas e aplicações.
- **Particles-bg:** biblioteca para animações de partículas no fundo da interface, melhorando a estética visual da aplicação.

4.3.3 Backend: Lógica e Processamento

O backend foi projetado para gerenciar o fluxo de dados, processar documentos e realizar análises automatizadas com auxílio de Inteligência Artificial. As tecnologias utilizadas incluem:

- **Python 3:** linguagem principal para desenvolvimento da lógica de negócio, manipulação de arquivos e integração com a IA;
- **FastAPI:** framework web moderno e assíncrono, utilizado para criação de endpoints RESTful de alta performance;
- **Uvicorn:** servidor ASGI utilizado para executar a aplicação FastAPI em ambiente de desenvolvimento e produção;
- **Google Generative AI API:** modelo *Gemini-2.5-flash* integrado para análise de contratos, identificação de cláusulas obrigatórias e desejáveis, detecção de riscos e geração de relatórios;
- **PyPDF2 / pypdf e python-docx:** bibliotecas utilizadas para leitura e extração de texto de arquivos PDF e DOCX;
- **FastAPI Middleware (CORS):** configurado para liberar o acesso ao backend a partir do frontend hospedado em outro domínio, permitindo integração segura e controlada;

4.3.4 Módulo de Inteligência Artificial

O módulo de Inteligência Artificial (IA) do sistema NoRisk é responsável pela análise automatizada dos contratos de estágio, identificando cláusulas obrigatórias e desejáveis, avaliando a conformidade legal e gerando relatórios estruturados. Este módulo combina técnicas de Processamento de Linguagem Natural (PLN) com modelos generativos avançados, possibilitando interpretação contextual do texto jurídico presente nos contratos.

4.3.4.0.1 Arquitetura do Módulo

O módulo foi projetado em camadas, contemplando:

- **Leitura de Documentos:** extração de conteúdo textual de arquivos PDF e DOCX utilizando as bibliotecas `pypdf` e `python-docx`.
- **Pré-processamento de Texto:** limpeza, normalização e segmentação do texto em parágrafos e sentenças, facilitando a análise subsequente pelo modelo de IA.

- **Análise com Modelo Generativo:** integração com a API Gemini 2.5 Flash, que recebe o texto do contrato e um *prompt* estruturado contendo os requisitos obrigatórios e desejáveis definidos a partir da pesquisa e da legislação vigente.
- **Geração de Relatórios:** o modelo produz um relatório textual categorizando as cláusulas como “Atendidas”, “Não Atendidas” ou “Observações”, permitindo que o usuário visualize rapidamente pontos críticos e recomendações.

4.3.4.0.2 Funcionamento do *prompt* e Avaliação de Conformidade

O *prompt* enviado ao modelo generativo é construído dinamicamente a partir dos requisitos identificados na pesquisa, incluindo:

- Requisitos obrigatórios (ex.: termo de compromisso, matrícula regular, carga horária compatível, recesso remunerado).
- Requisitos desejáveis (ex.: seguro contra acidentes, plano de atividades, relatórios periódicos).

O modelo analisa cada item e retorna se o contrato está **ÍNTEGRO** ou **NÃO ÍNTEGRO**, listando pontos críticos e observações separadamente. Esta abordagem garante que a avaliação seja consistente, reproduzível e baseada em critérios objetivos definidos pela legislação e pela pesquisa realizada.

4.3.4.0.3 Integração com o Sistema

O módulo de IA é acessado através do backend em *Python* com *FastAPI*, permitindo:

- Recepção de arquivos do frontend (React).
- Armazenamento temporário do documento para análise.
- Processamento do texto e envio ao modelo generativo.
- Retorno do relatório de análise em formato JSON, exibido de forma estruturada na interface do usuário.

4.3.4.0.4 Benefícios e Limitações

O módulo permite automatizar parte do processo de verificação contratual, reduzindo tempo e esforço humano. Entretanto, como se trata de um protótipo baseado em IA, recomenda-se que o resultado final seja sempre validado por um profissional do

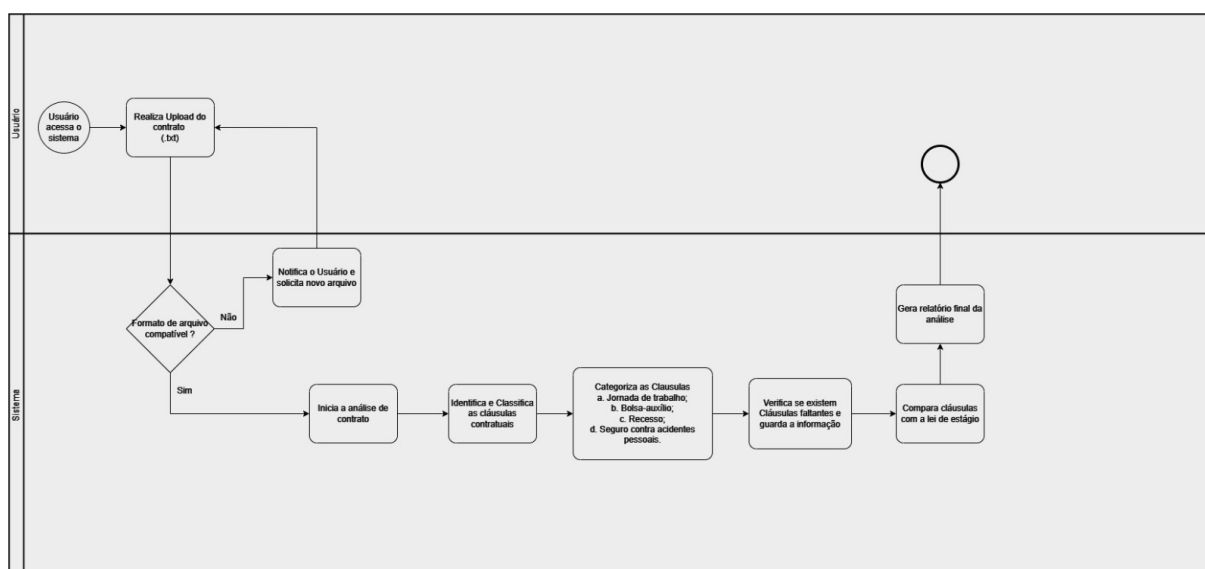
Direito, garantindo segurança jurídica e mitigando possíveis interpretações incorretas do modelo.

4.4 Modelagem do Processo – Fluxo BPMN

A modelagem do processo de análise de contratos foi realizada utilizando a notação **BPMN (Business Process Model and Notation)**, que permite representar graficamente o fluxo de atividades do sistema. De acordo com Weske (WESKE, 2019), a modelagem de processos é uma etapa essencial no desenvolvimento de sistemas, pois traduz requisitos técnicos em uma visão de fluxo compreensível por diferentes stakeholders.

O processo principal do sistema NoRisk está ilustrado na Figura 12.

Figura 12 – Fluxo BPMN do processo de análise do sistema NoRisk



Fonte: Autoria própria

Conforme ilustrado na figura 12, o fluxo compreende as seguintes etapas:

1. Realização do *upload* de arquivo (PDF ou DOCX);
2. Verificação de compatibilidade e leitura do texto;
3. Inicialização da análise automática do contrato;
4. Identificação e classificação das cláusulas contratuais;
5. Categorização das cláusulas em: jornada de trabalho, bolsa-auxílio, recesso remunerado e seguro contra acidentes;

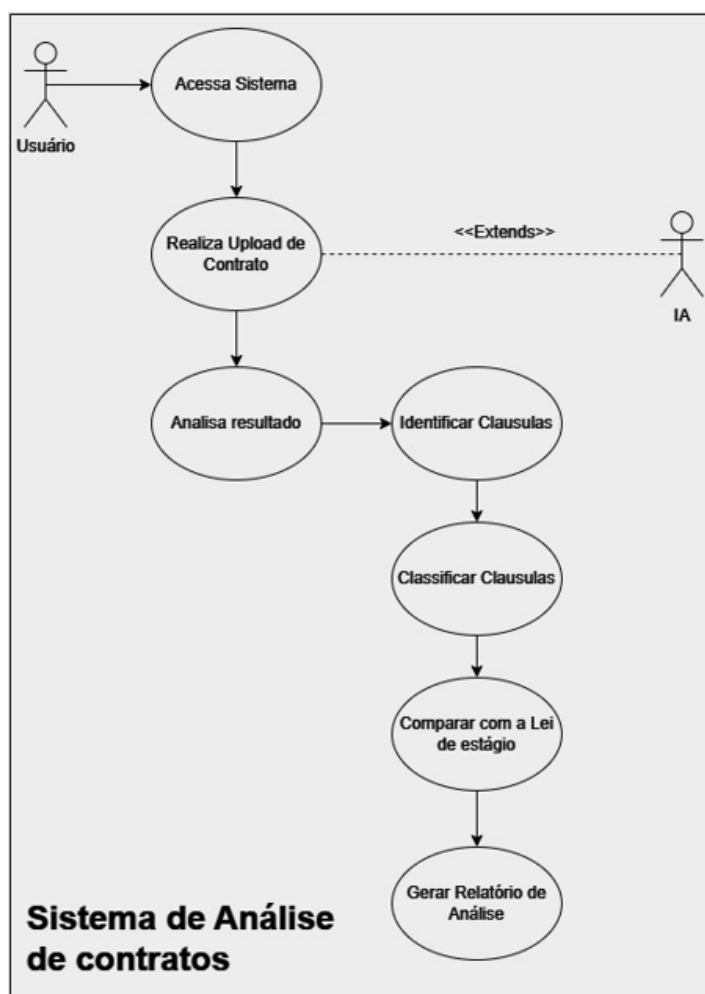
6. Verificação de ausência de cláusulas obrigatórias;
7. Comparação das cláusulas com os dispositivos da Lei do Estágio;
8. Geração do relatório final, indicando se o contrato é íntegro ou não.

Essa modelagem serviu como base para a implementação e validação do protótipo, garantindo que o fluxo operacional refletisse fielmente os requisitos definidos.

4.5 Arquitetura do Sistema - Diagrama UML

A figura 13 representa o Diagrama UML de Casos de Uso do sistema NoRisk. A UML (Unified Modeling Language) é uma linguagem padronizada amplamente utilizada na Engenharia de Software para modelar, visualizar e documentar sistemas de maneira estruturada. Os diagramas UML permitem compreender, de forma gráfica, a organização, os comportamentos e as interações entre os diversos componentes de um sistema.

Figura 13 – Diagrama UML de Casos de Uso do sistema NoRisk.



Fonte: Autoria própria

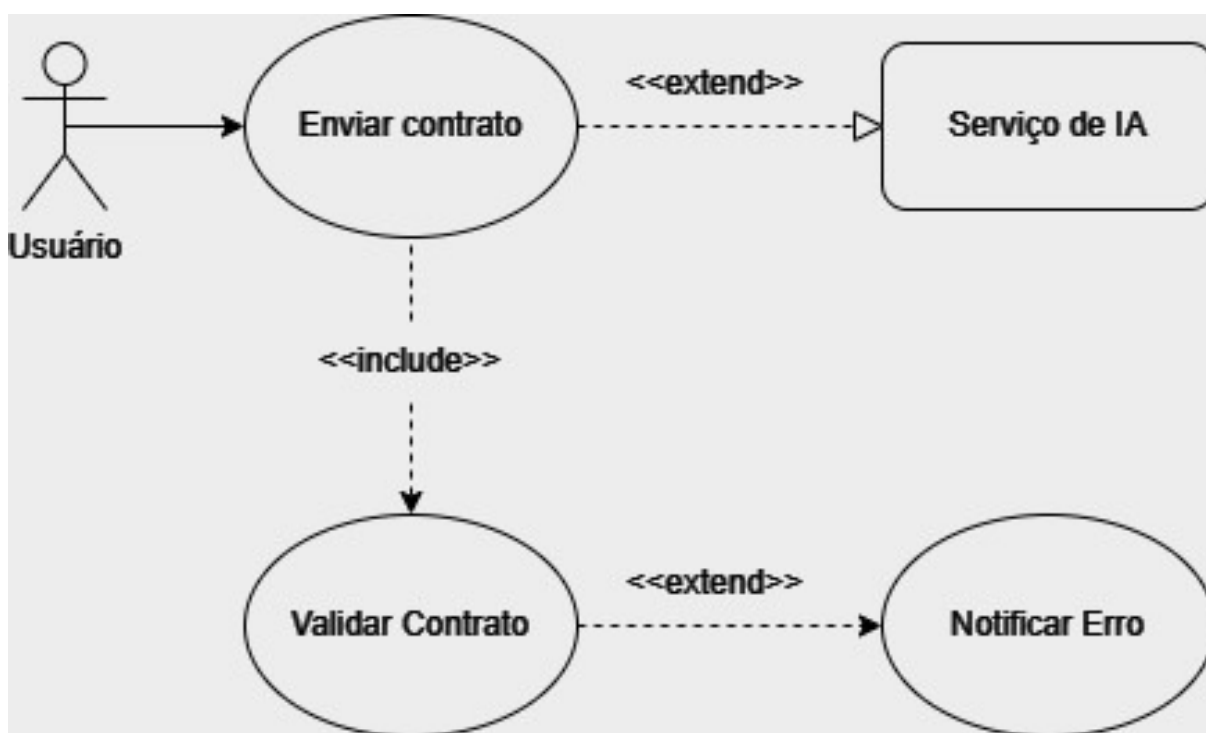
A figura 13 demonstra que o processo inicia quando o usuário acessa o sistema e realiza o *upload* do contrato a ser analisado. A partir desta ação, é ativado um conjunto de funcionalidades internas que pertencem ao módulo de IA, representado como um ator externo devido à sua autonomia e papel especializado na análise textual.

4.6 Fluxo de Operação

Módulo 1

Este módulo corresponde à etapa inicial da interação do usuário com o sistema, a qual permite o envio de contratos para análise.

Figura 14 – Fluxo do Módulo 1



Fonte: Autoria própria

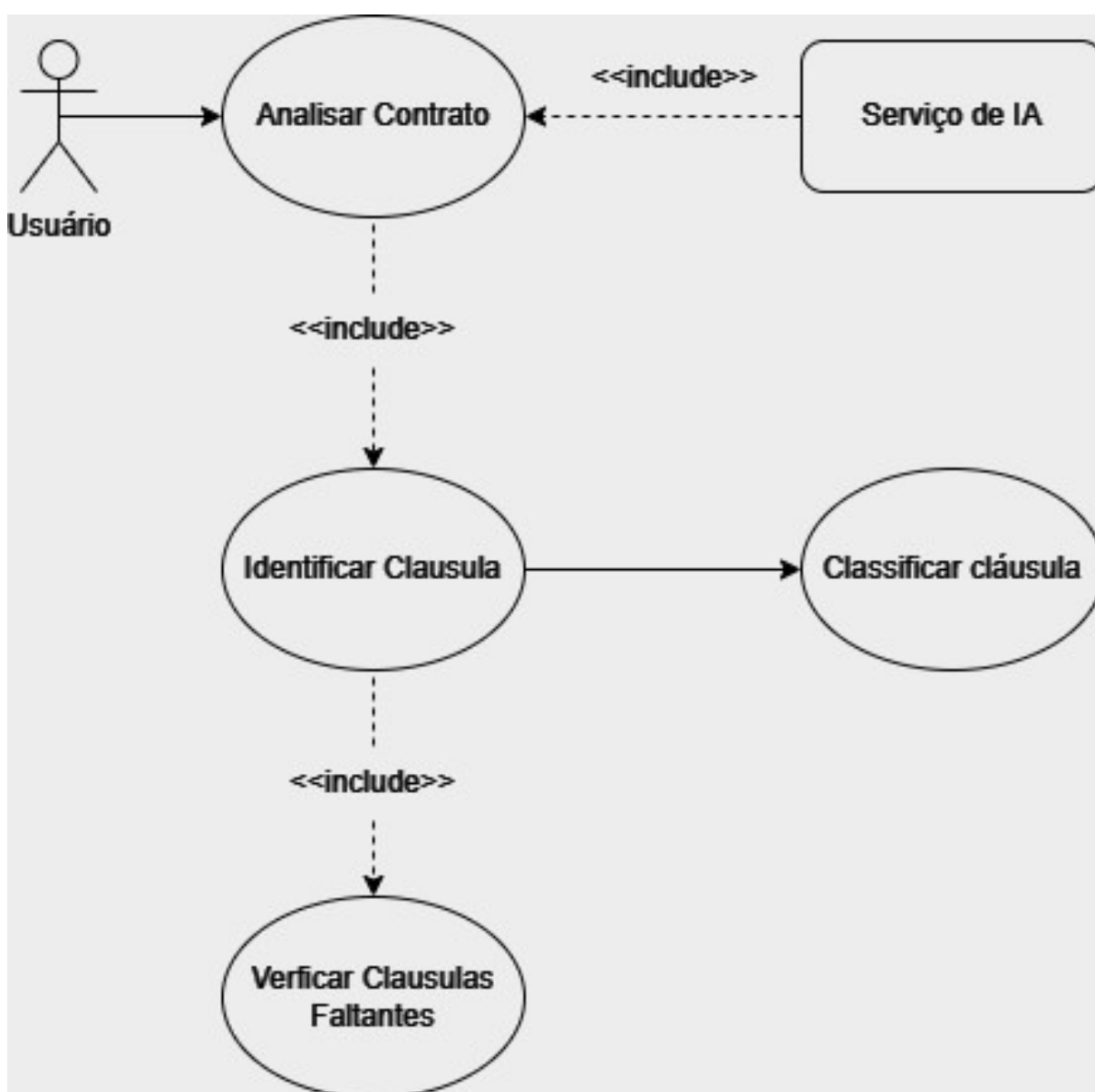
- **Acesso ao sistema:** O usuário acessa o sistema.
- **Seleção de Arquivo:** Após acessar o sistema, o usuário seleciona o arquivo para *upload*.
- **Validação de Formato:** Sistema realiza a verificação automática da compatibilidade do arquivo.
- **Notificação de Erro:** Caso o arquivo enviado seja incompatível com os formatos aceitos pelo sistema (PDF ou DOCX).

- **Encaminhamento para Análise:** Caso o arquivo seja considerado compatível, o sistema armazena o documento e encaminha ao módulo de análise baseado em IA.

Módulo 2

Este módulo descreve o fluxo automático realizado pelo sistema e pelo serviço de IA responsável pela análise das cláusulas contratuais.

Figura 15 – Fluxo do Módulo 2



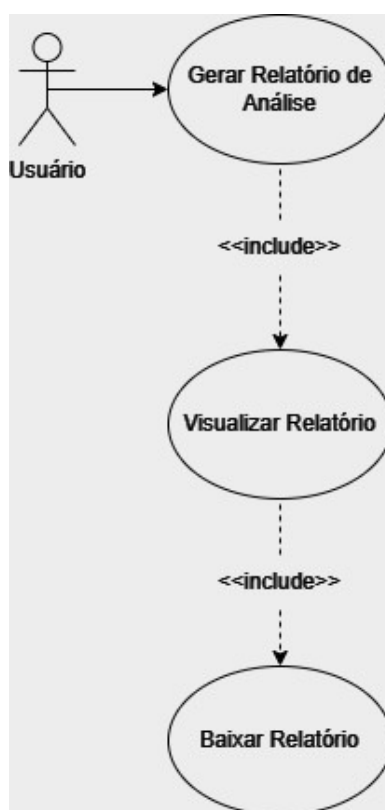
Fonte: Autoria própria

- **Processamento Inicial:** O Sistema envia o documento para o serviço de IA.
- **Identificação de Cláusulas:** A IA realiza a segmentação do texto do contrato e identifica as cláusulas contratuais.
- **Verificação de Ausência de Cláusulas:** A IA verifica se existem cláusulas ausentes de acordo com a lei do estágio.
- **Categorização de Cláusulas:** A IA separa as cláusulas contratuais dos demais textos presentes no arquivo.
- **Análise de Riscos:** A IA avalia os potenciais riscos e desvios contratuais de acordo com a lei de estágio.
- **Consolidação Interna:** O sistema recebe os resultados da IA, organiza os dados e prepara um resumo.

Módulo 3

Este módulo descreve a forma como o sistema disponibiliza ao usuário o resultado da análise.

Figura 16 – Fluxo do Módulo 3



Fonte: Autoria própria

- **Geração do Relatório:** O sistema compila todas as informações processadas e gera um relatório.
- **Exibição ao Usuário:** O usuário pode visualizar a análise diretamente na interface do sistema, a qual dirá se o contrato analisado é ou não considerado Íntegro.

Resumo do Fluxo Geral de Operação

1. O usuário acessa o sistema e realiza login.
2. O usuário envia um arquivo de contrato.
3. O sistema valida o arquivo.
4. Em caso de erro, o usuário é notificado.
5. Se válido, o sistema encaminha o contrato para a IA.
6. A IA identifica, classifica e verifica cláusulas.
7. O sistema consolida os resultados.
8. É gerado um relatório que é exibido em tela.

4.6.1 Etapas Executadas pelo Módulo de IA

O caso de uso “Realiza *Upload* de Contrato” estende o conjunto de funcionalidades do ator IA, que passa a executar automaticamente uma sequência de etapas fundamentais:

1. **Identificação de Cláusulas:** O texto do contrato é processado e dividido em cláusulas ou segmentos semânticos relevantes.
2. **Classificação das Cláusulas:** Cada cláusula é classificada conforme sua natureza jurídica (por exemplo: remuneração, atividades, jornada, rescisão, direitos e deveres, supervisão etc.).
3. **Comparação com a Lei de Estágio:** O sistema utiliza regras jurídicas e modelos de PLN para verificar a conformidade das cláusulas com os requisitos previstos na legislação aplicável.
4. **Geração do Relatório de Análise:** Após o processamento, o sistema sintetiza o resultado em um relatório estruturado, destacando riscos, inconsistências, pontos de atenção e conformidades.

4.6.2 Interação Final com o Usuário

Após a conclusão das etapas realizadas pela IA, o usuário acessa o caso de uso “Analisa resultado”, onde tem acesso ao relatório final gerado automaticamente. Esse fluxo evidencia a separação de responsabilidades no sistema:

- o usuário apenas fornece o documento e recebe o resultado;
- a IA executa todo o processamento técnico;
- o sistema atua como intermediário, integrando as interações.

4.6.3 Síntese Arquitetural

O diagrama reforça a estratégia arquitetural adotada pelo sistema *NoRisk*, baseada em automação inteligente, modularidade e clareza nas responsabilidades de cada componente.

4.7 Descrição do Arquivo `analyze.py`

O arquivo `analyze.py` constitui o núcleo do backend do sistema. Nele é definida a aplicação FastAPI responsável por receber os arquivos enviados pelo usuário, processar seu conteúdo e interagir com a API Gemini.

Inicialmente, o arquivo configura o ambiente, importa as bibliotecas necessárias e define o middleware de CORS. Em seguida, são implementadas funções auxiliares para leitura de arquivos PDF e DOCX, que extraem o texto bruto do contrato. Esses métodos garantem compatibilidade com os formatos mais comuns de documentos jurídicos.

A função principal de análise constrói dinamicamente um *prompt* contendo as regras legais aplicáveis aos contratos de estágio e insere o texto extraído do documento. Esse *prompt* é enviado ao modelo `gemini-2.5-flash` por meio da API Gemini, que retorna uma resposta textual estruturada em formato de relatório.

Por fim, o endpoint `/analyze` recebe o arquivo enviado pelo frontend, identifica seu formato, extrai o texto, executa a análise e retorna o resultado ao usuário.

4.8 Tela Inicial – Sistema Inativo

A Figura 17 apresenta a interface inicial do *NoRisk*.

Figura 17 – Tela inicial do sistema NoRisk

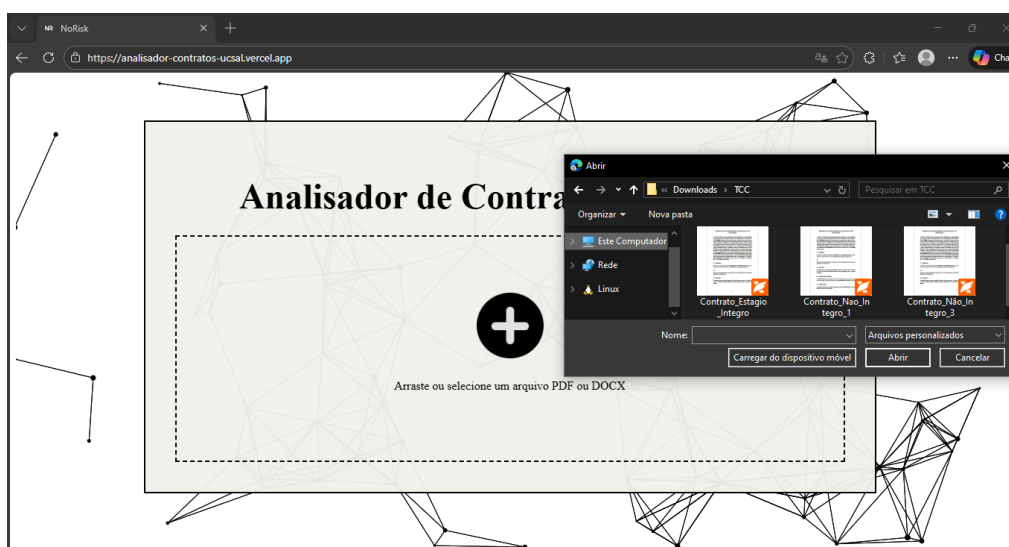


Fonte: Autoria própria

Na figura 17, observa-se que o usuário tem acesso às instruções básicas de utilização e à opção para realizar o *upload* do contrato a ser analisado. Essa tela foi projetada para oferecer uma experiência simples e intuitiva, facilitando o uso por pessoas sem conhecimento técnico.

4.9 Tela de *Upload* – Seleção do Arquivo

Na segunda etapa (Figura 18), o usuário pode selecionar o contrato armazenado em seu computador para envio.

Figura 18 – Tela de *upload* do arquivo – NoRisk

Fonte: Autoria própria

Conforme ilustrado na figura 18, o sistema suporta arquivos em formato PDF e DOCX, garantindo compatibilidade com os principais modelos de contrato utilizados no meio acadêmico e corporativo.

4.10 Tela de Processamento – Análise Automática

Após o envio, o sistema inicia automaticamente a análise do documento, conforme ilustrado na Figura 19.

Figura 19 – Tela de análise automática do contrato



Fonte: Autoria própria

A figura 19 evidencia a tela de carregamento ao analisar o contrato. Durante esse processo, o módulo de Inteligência Artificial aplica técnicas de (PLN) para identificar cláusulas obrigatórias, verificar conformidade com a legislação vigente e detectar possíveis anomalias contratuais.

4.11 Tela de Resultado – Contrato Íntegro

Quando o contrato analisado está em conformidade com os critérios estabelecidos pela Lei do Estágio e não apresenta indícios de irregularidades, o sistema retorna o resultado ilustrado na Figura 20.

Figura 20 – Tela de resultado indicando que o contrato é íntegro



Fonte: Autoria própria

Na figura 20, percebe-se que o selo “**Contrato Íntegro**” é exibido, indicando que o documento atende aos requisitos mínimos de validade e integridade jurídica.

4.12 Tela de Resultado – Contrato Não Íntegro

Por outro lado, quando o sistema identifica inconsistências, cláusulas omissas ou potenciais riscos jurídicos, é exibida a tela mostrada na Figura 21.

Figura 21 – Tela de resultado indicando que o contrato não é íntegro



Fonte: Autoria própria

Conforme ilustrado na figura 21, o resultado “**Contrato Não Íntegro**” vem acompanhado de uma lista de observações explicativas, orientando o usuário sobre os pontos

que necessitam de revisão ou correção.

Dessa forma, o NoRisk se apresenta como uma ferramenta de apoio à análise de contratos de estágio, oferecendo uma forma prática de verificar cláusulas obrigatórias e identificar possíveis inconsistências, promovendo maior clareza e agilidade na avaliação documental.

4.13 Resultados dos Testes

Para validar o funcionamento do sistema, foi utilizado como referência um modelo de contrato disponibilizado pela Universidade de São Paulo (USP) (GEOCIÊNCIAS, 2018). O contrato foi preenchido com dados fictícios e, a partir dele, duas cópias íntegras foram geradas: uma em formato PDF e outra em formato DOCX.

Em seguida, foram produzidas oito cópias adicionais do contrato, sendo sete em PDF e uma em DOCX. Em cada uma dessas cópias, uma cláusula diferente foi modificada, de modo a torná-las não íntegras.

Dessa forma, totalizaram-se **duas cópias íntegras e oito cópias não íntegras**.

Os testes realizados demonstraram que o sistema conseguiu identificar corretamente 100% dos documentos íntegros e não íntegros, sem ocorrência de erros (0%). Portanto, o funcionamento esperado foi comprovado em todos os cenários testados. Apesar da alta precisão obtida nos testes, ressalta-se que os resultados não são suficientes para garantir a eficácia do sistema em situações reais, uma vez que os testes foram realizados apenas com um modelo de contrato modificado com dados fictícios, sem validação com documentos autênticos ou em um conjunto mais amplo de amostras.

5 Conclusão e Trabalhos Futuros

Este capítulo apresenta os principais resultados obtidos ao longo do projeto, e também as dificuldades.

O presente trabalho apresentou o desenvolvimento de uma ferramenta baseada em Inteligência Artificial voltada para a análise e avaliação de riscos jurídicos em contratos de estágio. A aplicação de técnicas de *Processamento de Linguagem Natural* (PLN) e aprendizado de máquina mostrou-se promissora para a automatização da leitura e interpretação de cláusulas contratuais, oferecendo suporte à tomada de decisão de estudantes e profissionais não especializados.

Durante o desenvolvimento do projeto, foram constatadas dificuldades na utilização da API do ChatGPT, disponível apenas em formato pago. Esse fator motivou a escolha do Gemini, que possibilitou a realização dos experimentos sem custos adicionais por meio de créditos gratuitos limitados.

Os testes demonstraram que o NoRisk identificou corretamente 100% dos documentos íntegros e não íntegros. Contudo, por terem sido realizados apenas com modelos fictícios, os resultados não garantem a eficácia em situações reais.

Os resultados e a revisão teórica indicam que modelos como o BERT e o GPT alcançam alto desempenho na detecção de cláusulas de risco, conforme observado por Moon, Chi e Im (MOON; CHI; IM, 2022) e Dikmen et al. (DIKMEN et al., 2025). Tais abordagens, aliadas à padronização de documentos jurídicos proposta por Lima e Ciciliati (LIMA; CICILIATI, 2008), demonstram que a tecnologia pode contribuir de forma significativa para a democratização do acesso à informação jurídica e para a redução de vulnerabilidades contratuais.

O uso da IA no contexto jurídico, no entanto, traz consigo desafios éticos e técnicos. Entre eles estão a necessidade de transparência dos modelos, a proteção dos dados pessoais dos usuários e a interpretabilidade das decisões algorítmicas. Kazemi e Alvanchi (KAZEMI; ALVANCHI, 2025) destacam que, embora os modelos de aprendizado profundo apresentem excelente desempenho, sua natureza de “caixa-preta” dificulta a explicação dos resultados ao usuário final — o que reforça a importância de desenvolver soluções explicáveis e auditáveis.

Como perspectiva futura, recomenda-se a ampliação do conjunto de dados para incluir contratos de diferentes áreas do Direito, o aprimoramento da interface da ferramenta e a implementação de módulos de explicabilidade da IA. Tais aprimoramentos poderão fortalecer a confiança dos usuários e permitir a adoção da tecnologia em ambientes institucionais.

Conclui-se que a integração entre IA, Direito e Engenharia de Software pode repre-

sentar um marco na forma como os contratos são compreendidos e geridos. Quando aplicada com responsabilidade e foco na ética, a Inteligência Artificial se torna uma ferramenta de empoderamento, capaz de promover a autonomia informacional e a equidade nas relações jurídicas.

Referências

Amazon Web Services. *O que é uma API?* 2024. Acesso em: 19 out. 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/api/>>. 24

Amazon Web Services. *Qual é a diferença entre frontend e backend?* 2025. Acesso em: 18 out. 2025. Disponível em: <<https://aws.amazon.com/pt/compare/the-difference-between-frontend-and-backend/>>. 23

Amazon Web Services. *What is a RESTful API?* 2025. Acesso em: 24 out. 2025. Disponível em: <<https://aws.amazon.com/what-is/restful-api/>>. 24

Atlassian. *Kanban*. 2024. Acesso em: 19 out. 2025. Disponível em: <<https://www.atlassian.com/br/agile/kanban>>. 26

ATLASSIAN. *Minimum Viable Product (MVP)*. 2025. Acesso em: 26 out. 2025. Disponível em: <<https://www.atlassian.com/agile/product-management/minimum-viable-product>>. 30

BECK, K. et al. *Manifesto for Agile Software Development*. 2001. Acesso em: 19 out. 2025. Disponível em: <<https://agilemanifesto.org/>>. 45

BERMUDES, W. L. et al. Tipos de escalas utilizadas em pesquisas e suas aplicações. *VÉRTICES*, v. 18, n. 2, p. 7–20, May/Aug 2016. Acesso em 26 out. 2025. Disponível em: <<https://editoraessentia.iff.edu.br/index.php/vertices/article/download/1809-2667.v18n216-01/5242/20548>>. 29

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modeling Language User Guide*. 2. ed. [S.l.]: Addison-Wesley, 2005. 27

BRASIL. *Lei nº 11.788, de 25 de setembro de 2008 — Dispõe sobre o estágio de estudantes*. 2008. Acesso em: 09 out. 2025. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11788.htm>. 18, 21, 31, 41, 44

BRASIL. *Lei nº 13.709, de 14 de agosto de 2018 – Lei Geral de Proteção de Dados Pessoais (LGPD)*. 2018. Diário Oficial da União. Acesso em: 13 nov. 2025. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. 22, 36

(BRASIL), C. N. de S. *Resolução nº 510, de 7 de abril de 2016 – Dispõe sobre as normas aplicáveis a pesquisas em Ciências Humanas e Sociais*. 2016. Diário Oficial da União. Acesso em: 13 nov. 2025. Disponível em: <<https://conselho.saude.gov.br/resolucoes/2016/Reso510.pdf>>. 35

BROWN, T. B.; MANN, B.; RYDER, N. e. a. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, v. 33, p. 1877–1901, 2020. Acesso em: 19 out. 2025. 43

CRESWELL, J. W. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 4th. ed. [S.l.]: SAGE Publications, 2014. 31, 32

- DIKMEN, I. et al. Automated construction contract analysis for risk and responsibility assessment using natural language processing and machine learning. *Computers in Industry*, v. 166, p. 104251, 2025. ISSN 0166-3615. Acesso em: 11 out. 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0166361525000168>>. 19, 20, 41, 60
- GEOCIÊNCIAS, U. de São Paulo. Instituto de. *Termo de Compromisso de Estágio*. São Paulo: [s.n.], 2018. Disponível em: <https://igc.usp.br/wp-content/uploads/2018/10/Termo_de_Compromisso_de_Estagio.pdf>. Acesso em: 08 out. 2025. 59
- GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. 6. ed. [S.l.]: Atlas, 2008. 31, 32
- IBM. *O que é BPMN?* 2025. Acesso em: 20 dez. 2025. Disponível em: <Disponível em: <https://www.ibm.com/br-pt/think/topics/bpmn>>. 30
- IBM. *REST APIs*. 2025. Acesso em: 24 out. 2025. Disponível em: <<https://www.ibm.com/br-pt/think/topics/rest-apis>>. 24
- JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*. 3. ed. [S.l.]: Stanford University, 2023. 40
- KAZEMI, M. H.; ALVANCHI, A. Application of nlp-based models in automated detection of risky contract statements written in complex script system. *Expert Systems with Applications*, v. 259, p. 125296, 2025. ISSN 0957-4174. Acesso em: 19 out. 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417424021638>>. 19, 20, 41, 60
- LIMA, J. A. de O.; CICILIATI, F. *LexML - Esquema XML para Normas Jurídicas*. 2008. Acesso em: 10 out. 2025. Disponível em: <<https://projeto.lexml.gov.br/documentacao/Parte-3-XML-Schema.pdf>>. 18, 60
- LUCIDCHART. *O que é UML*. 2025. <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 20 dez. 2025. 27, 28, 29
- MOON, S.; CHI, S.; IM, S.-B. Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (bert). *Automation in Construction*, v. 142, p. 104465, 2022. ISSN 0926-5805. Acesso em: 09 out. 2025. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0926580522003387>>. 19, 20, 60
- Mozilla Developer Network. *JavaScript*. 2024. Acesso em: 19 out. 2025. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. 22
- Mozilla Developer Network. *CORS — Cross-Origin Resource Sharing*. 2025. Acesso em: 22 out. 2025. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CORS>>. 24
- Mozilla Developer Network. *HTML: HyperText Markup Language*. 2025. Acesso em: 26 out. 2025. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. 22
- Mozilla Developer Network. *Trabalhando com JSON*. 2025. Acesso em: 25 out. 2025. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Scripting/JSON>. 24

PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. [S.I.]: AMGH Editora, 2014. 41

Python Software Foundation. *Python Programming Language — Executive Summary*. 2024. Acesso em: 19 out. 2025. Disponível em: <<https://www.python.org/doc/essays/blurb/>>. 23

REACT. *Documentação oficial do React*. 2025. Acesso em: 25 out. 2025. Disponível em: <<https://pt-br.legacy.reactjs.org/>>. 23

REMUS, D.; LEVY, F. S. Can robots be lawyers? computers, lawyers, and the practice of law. *Georgetown Journal of Legal Ethics*, v. 30, p. 501–558, 2016. Acesso em: 19 out. 2025. 18

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. [S.I.]: Pearson Education, 2011. 40

SURDEN, H. Artificial intelligence and law: An overview. *Georgia State University Law Review*, v. 35, n. 4, p. 1305–1336, 2019. Acesso em: 21 out. 2025. 18

TOTVS. *O que é Front-end*. 2023. Acesso em: 18 out. 2025. Disponível em: <<https://www.totvs.com/blog/developers/front-end/>>. 22

União Europeia. *Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho, de 27 de abril de 2016 — Regulamento Geral sobre a Proteção de Dados (GDPR)*. 2016. Jornal Oficial da União Europeia, L 119, 4 maio 2016. Acesso em: 09 out. 2025. Disponível em: <<https://eur-lex.europa.eu/eli/reg/2016/679/oj>>. 22

VITE. *Documentação Oficial do Vite*. 2025. Acesso em: 29 out. 2025. Disponível em: <<https://pt.vite.dev/guide/>>. 23

WESKE, M. *Business Process Management: Concepts, Languages, Architectures*. [S.I.]: Springer, 2019. 49

World Wide Web Consortium. *Cascading Style Sheets (CSS)*. 2024. Acesso em: 19 out. 2025. Disponível em: <<https://www.w3.org/Style/CSS/Overview.en.html>>. 23

APÊNDICE A – Formulário de Pesquisa

Este apêndice apresenta o formulário utilizado na coleta de dados para o desenvolvimento deste Trabalho de Conclusão de Curso. O instrumento foi elaborado por meio da plataforma *Google Forms* e teve como objetivo identificar as principais dificuldades e percepções dos estudantes, bacharéis e profissionais da área de Direito em relação à análise e compreensão de contratos de estágio, bem como avaliar o interesse em ferramentas baseadas em Inteligência Artificial para apoio jurídico.

O questionário foi composto por perguntas objetivas e discursivas, abrangendo aspectos como:

- compreensão das cláusulas contratuais e direitos previstos na Lei nº 11.788/2008 (Lei do Estágio);
- percepção de riscos e abusos contratuais;
- familiaridade com a linguagem jurídica;
- abertura ao uso de tecnologias digitais aplicadas ao Direito.

As Figuras A1 a A5 apresentam as páginas do formulário de pesquisa aplicado.

Figura A1 – Formulário de Pesquisa – Página 1

Formulário para Implementação de Ferramenta de Análise de Contratos de Estágio para TCC

Este formulário faz parte de um Trabalho de Conclusão de Curso (TCC) em Engenharia de Software.

O objetivo do projeto é desenvolver uma ferramenta para avaliação de aspectos fundamentais em contratos de estágio, com base na **Lei nº 11.788/2008 (Lei do Estágio)**

Sua contribuição como profissional ou estudante da área jurídica é essencial para compreendermos os principais pontos de atenção nesses contratos e validarmos a aplicabilidade da ferramenta.

As respostas serão utilizadas exclusivamente para fins acadêmicos e, respeitando a LGPD, não coletaremos nenhum dado pessoal sensível neste formulário, sendo totalmente anônimo.

Agradecemos muito pela sua colaboração!

** Indica uma pergunta obrigatória*

1. Tipo de respondente *

Marcar apenas uma oval.

- Aluno(a) (estudante de graduação) *Pular para a pergunta 2*
- Bacharel(a) (formado(a), sem atuação jurídica regular)
Pular para a pergunta 3
- Profissional (advogado) *Pular para a pergunta 3*

Fonte: Autoria própria

Figura A2 – Formulário de Pesquisa – Página 2

2. Você já cursou a disciplina de Direito do Trabalho? *

Marcar apenas uma oval.

- Sim Pular para a pergunta 3
 Não Pular para a seção 3 (Participação Encerrada)

Participação Encerrada

Agradecemos o seu interesse e disposição em participar da nossa pesquisa. As perguntas a seguir exigem conhecimento em Direito do Trabalho, por isso encerraremos sua participação aqui.

Itens

Leia cada item e atribua uma prioridade de **1 a 5**, conforme indicado na legenda, para que nossa ferramenta avalie os **aspectos fundamentais**.

1 - Totalmente irrelevante

2 - Pouco importante

3 - Neutro

4 - Importante

5 - Essencial

3. Verificar a existência de um Termo de Compromisso formalizado entre o estudante, a empresa e a instituição de ensino *

Marcar apenas uma oval.

1 2 3 4 5

Fonte: Autoria própria

Figura A3 – Formulário de Pesquisa – Página 3

4. Checar se o contrato atesta a matrícula e frequência regular do estudante *

Marcar apenas uma oval.

1 2 3 4 5

5. Identificar a cláusula que estabelece a contratação de seguro contra acidentes pessoais para o estagiário *

Marcar apenas uma oval.

1 2 3 4 5

6. Checar se os nomes do supervisor da parte concedente e do professor orientador da instituição de ensino estão designados no termo *

Marcar apenas uma oval.

1 2 3 4 5

7. Verificar se a carga horária descrita no contrato é compatível com o nível de ensino do estudante, respeitando os limites legais específicos (ex: 4h/dia e 20h/semana para educação especial; 6h/dia e 30h/semana para ensino superior; ou até 40h/semana para cursos que alternam teoria e prática) *

Marcar apenas uma oval.

1 2 3 4 5

Fonte: Autoria própria

Figura A4 – Formulário de Pesquisa – Página 4

8. Calcular o período entre a data de início e a de término do contrato para garantir que não exceda o limite de 2 anos, exceto quando se tratar de estagiário portador de deficiência *

Marcar apenas uma oval.

1 2 3 4 5

9. Verificar se há uma seção ou anexo denominado "Plano de Atividades" com a descrição das tarefas que serão realizadas pelo estagiário *

Marcar apenas uma oval.

1 2 3 4 5

10. Verificar, para os estágios do tipo não obrigatório, a inclusão das cláusulas compulsórias de Bolsa, Auxílio-Transporte e a previsão do Recesso Remunerado *

Marcar apenas uma oval.

1 2 3 4 5

11. Verificar a existência da cláusula que assegura o direito ao recesso de 30 dias (ou proporcional), benefício obrigatório para todos os tipos de estágio *

Marcar apenas uma oval.

1 2 3 4 5

Fonte: Autoria própria

Figura A5 – Formulário de Pesquisa – Página 5

12. Buscar por uma cláusula que mencione a obrigação do estagiário de entregar *
relatórios de atividades em um período não superior a 6 meses

Marcar apenas uma oval.

1 2 3 4 5

Fonte: Autoria própria

As respostas obtidas por meio desse instrumento serviram de base para a análise dos requisitos do sistema proposto e para a validação da relevância do uso de Inteligência Artificial na análise de contratos de estágio.

APÊNDICE B – Resultados da Pesquisa

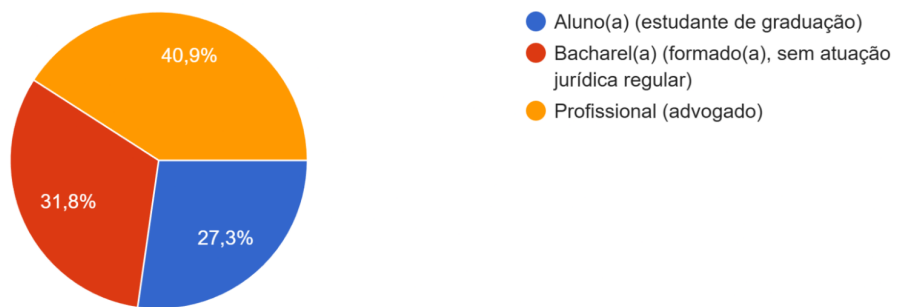
Neste apêndice, disponibiliza-se o acesso ao documento contendo os dados brutos coletados por meio do questionário aplicado a **22 voluntários**.

O relatório completo, com a consolidação e detalhamento dos resultados da pesquisa, pode ser consultado abaixo:

Figura B1 – Respostas da questão 1 do Forms

Tipo de respondente

22 respostas

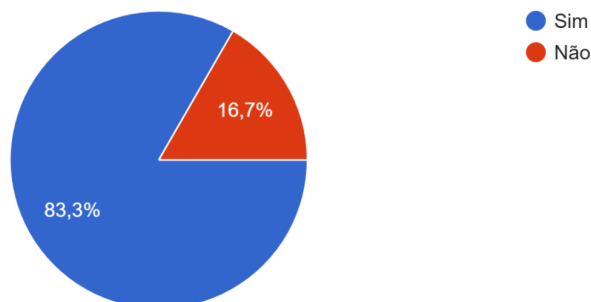


Fonte: Autoria própria

Figura B2 – Respostas da questão 2 do Forms (aplicada somente aos alunos)

Você já cursou a disciplina de Direito do Trabalho?

6 respostas

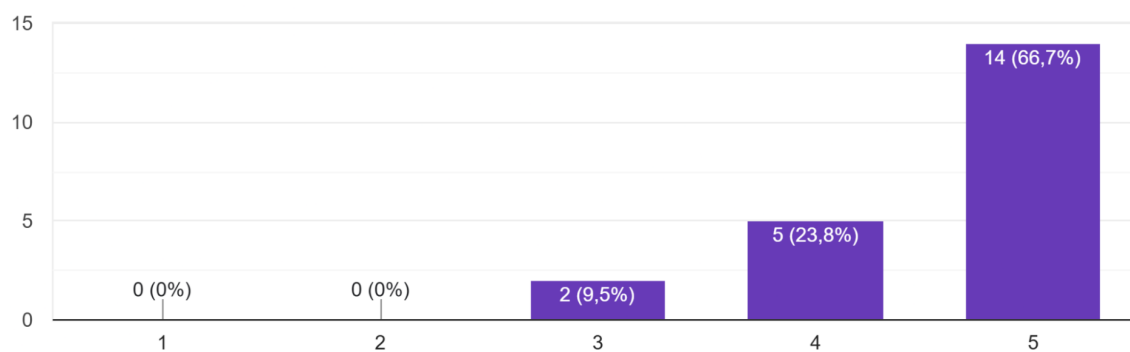


Fonte: Autoria própria

Figura B3 – Respostas da questão 3 do Forms

Verificar a existência de um Termo de Compromisso formalizado entre o estudante, a empresa e a instituição de ensino

21 respostas

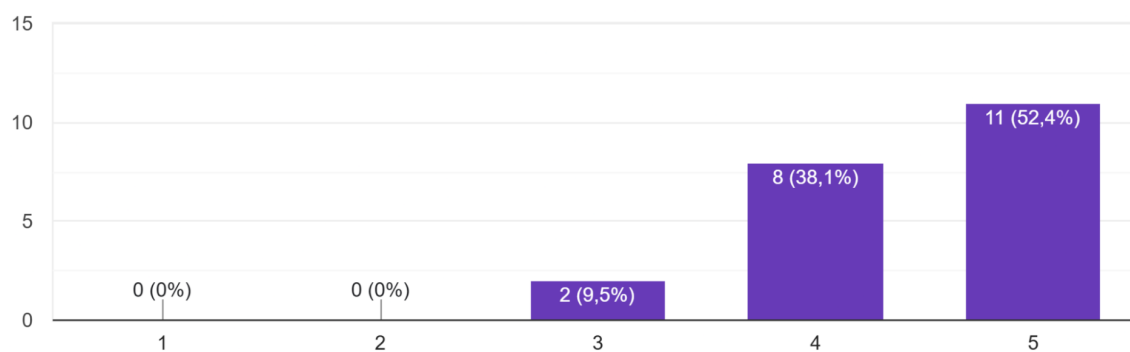


Fonte: Autoria própria

Figura B4 – Respostas da questão 4 do Forms

Checar se o contrato atesta a matrícula e frequência regular do estudante

21 respostas

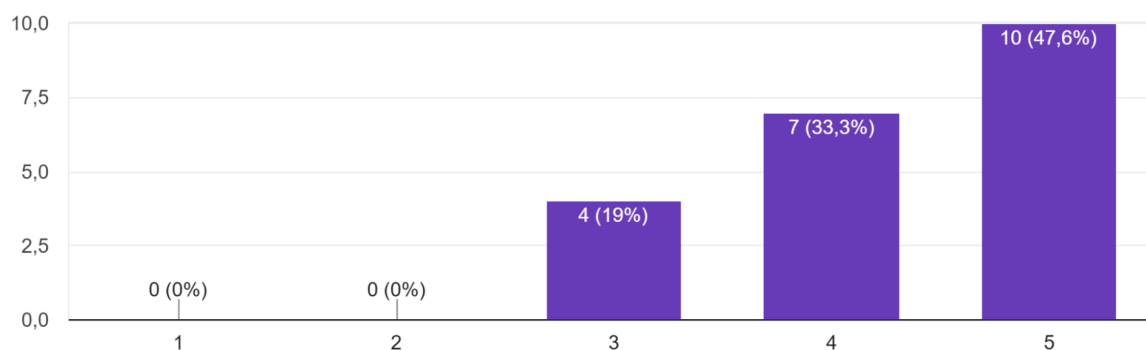


Fonte: Autoria própria

Figura B5 – Respostas da questão 5 do Forms

Identificar a cláusula que estabelece a contratação de seguro contra acidentes pessoais para o estagiário

21 respostas

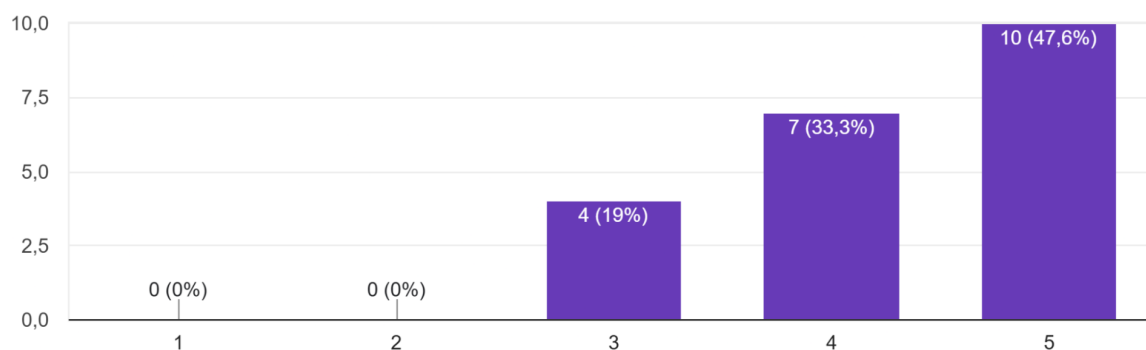


Fonte: Autoria própria

Figura B6 – Respostas da questão 6 do Forms

Checar se os nomes do supervisor da parte concedente e do professor orientador da instituição de ensino estão designados no termo

21 respostas

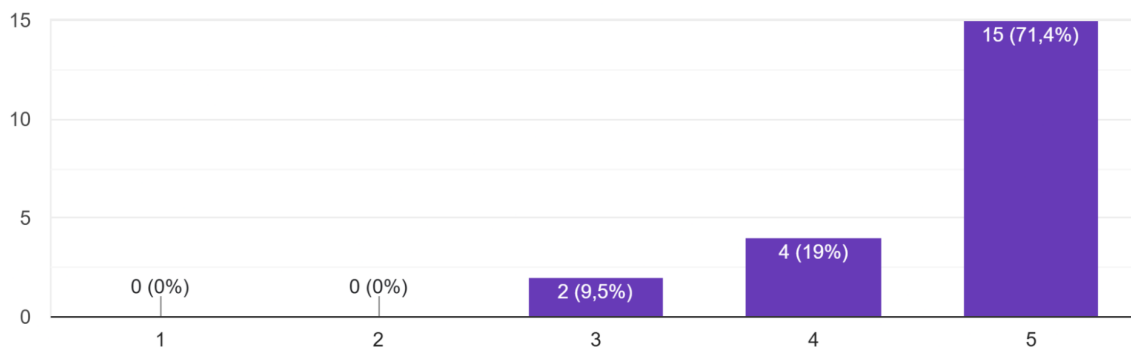


Fonte: Autoria própria

Figura B7 – Respostas da questão 7 do Forms

Verificar se a carga horária descrita no contrato é compatível com o nível de ensino do estudante, respeitando os limites legais específicos (ex: 4h/.../semana para cursos que alternam teoria e prática)

21 respostas

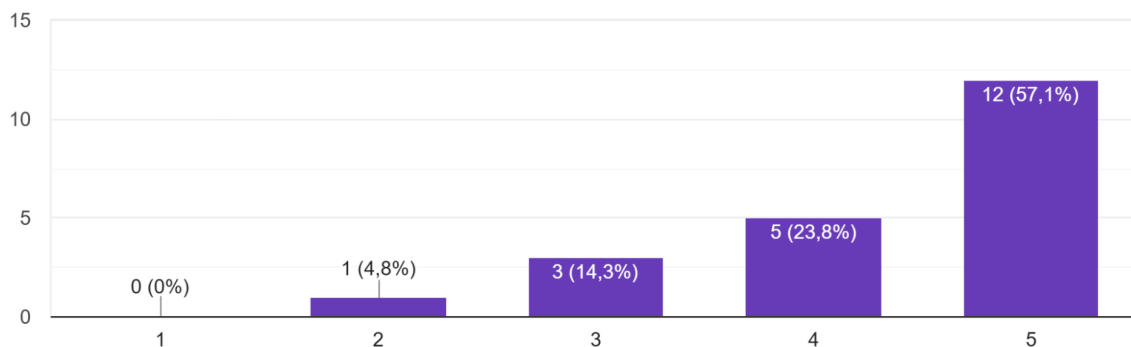


Fonte: Autoria própria

Figura B8 – Respostas da questão 8 do Forms

Calcular o período entre a data de início e a de término do contrato para garantir que não exceda o limite de 2 anos, exceto quando se tratar de estagiário portador de deficiência

21 respostas

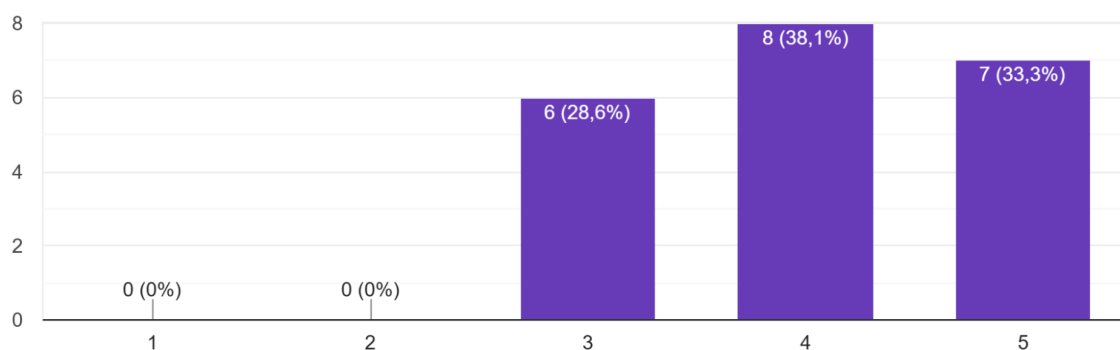


Fonte: Autoria própria

Figura B9 – Respostas da questão 9 do Forms

Verificar se há uma seção ou anexo denominado "Plano de Atividades" com a descrição das tarefas que serão realizadas pelo estagiário

21 respostas

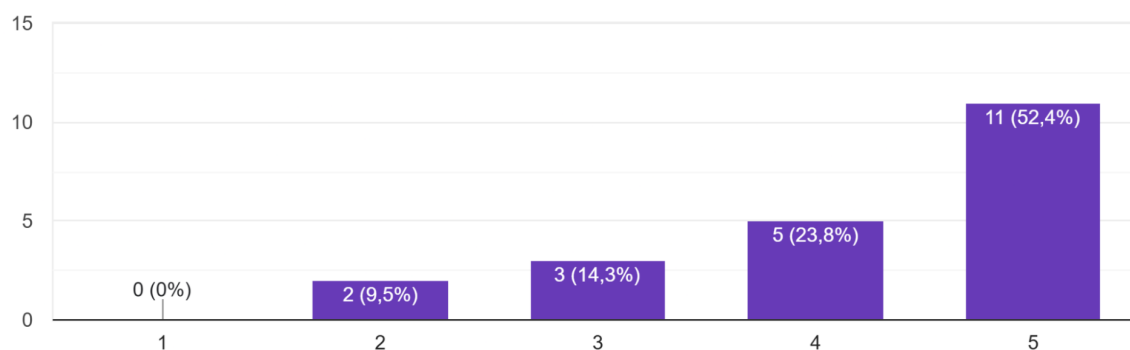


Fonte: Autoria própria

Figura B10 – Respostas da questão 10 do Forms

Verificar, para os estágios do tipo não obrigatório, a inclusão das cláusulas compulsórias de Bolsa, Auxílio-Transporte e a previsão do Recesso Remunerado

21 respostas

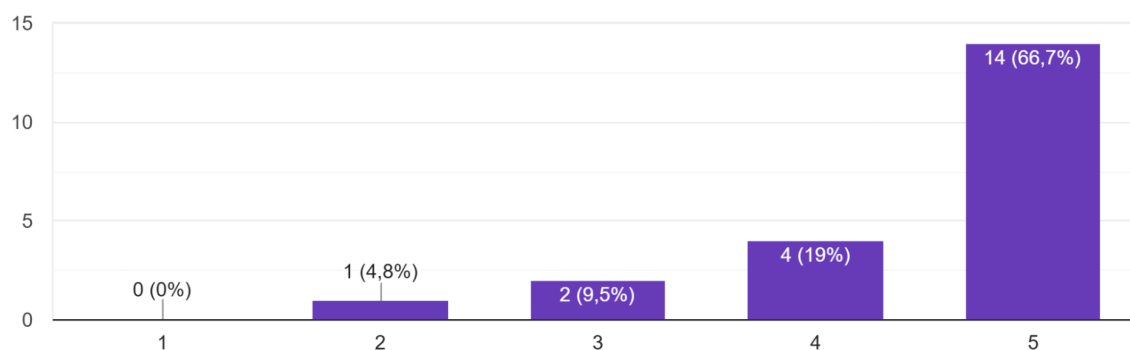


Fonte: Autoria própria

Figura B11 – Respostas da questão 11 do Forms

Verificar a existência da cláusula que assegura o direito ao recesso de 30 dias (ou proporcional), benefício obrigatório para todos os tipos de estágio

21 respostas

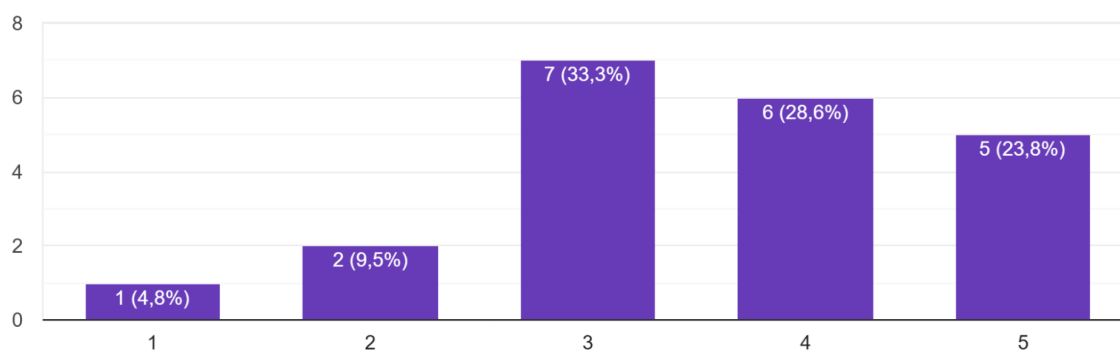


Fonte: Autoria própria

Figura B12 – Respostas da questão 12 do Forms

Buscar por uma cláusula que mencione a obrigação do estagiário de entregar relatórios de atividades em um período não superior a 6 meses

21 respostas



Fonte: Autoria própria

APÊNDICE C – Prompts Elaborados

Este apêndice apresenta o *prompt* utilizado no módulo de Inteligência Artificial do sistema **NoRisk**, responsável pela análise automatizada de contratos de estágio. O *prompt* define as regras, critérios jurídicos e a estrutura de saída empregadas pelo modelo de linguagem para interpretar o conteúdo contratual e gerar um relatório descritivo quanto à sua integridade e conformidade legal.

C.1 Descrição do *Prompt* de Análise

A Figura C1 ilustra o trecho do código-fonte responsável pela construção do *prompt* enviado ao modelo de Inteligência Artificial, implementado no arquivo `analyze.py`. Esse *prompt* é a principal interface de comunicação entre o sistema e o modelo de linguagem, sendo responsável por orientar todo o processo de análise do contrato.

Figura C1 – Trecho do código com definição do *prompt* de análise de contratos

```

analyze.py x
backend > analyze.py > analyze_contract
31 def analyze_contract(text):
32     prompt = f"""
33     Você é uma ferramenta de análise de contratos de estágio. Verifique se o contrato está INTEGRADO ou NÃO INTEGRADO com base nos requisitos abaixo.
34
35     Requisitos OBRIGATORIOS:
36
37     - Existe um Termo de Compromisso formalizado entre o estudante, a empresa e a instituição de ensino?
38     - O contrato atesta a matrícula e frequência regular do estudante?
39     - A carga horária descrita no contrato é compatível com o nível de ensino do estudante, respeitando os limites legais específicos (ex: 4h/dia e 20h/semana para educação especial; 6h/dia e 30
40     - O período entre a data de início e a de término do contrato não excede o limite de 2 anos? (exceto caso esteja tratando de estagiário portador de deficiência)?
41     - Em caso de estágio do tipo não obrigatório, as cláusulas compulsórias de Bolsa, Auxílio-Transporte estão inclusas e há presença de previsão do Recurso Remunerado?
42     - Existe a cláusula que assegura o direito ao recesso de 30 dias (ou proporcional), benefício obrigatório para todos os tipos de estágio?
43
44     Requisitos DESEJAVEIS:
45
46     - A cláusula que estabelece a contratação de seguro contra acidentes pessoais para o estagiário foi identificada?
47     - Os nomes do supervisor da parte concedente e do professor orientador da instituição de ensino estão designados no termo?
48     - Há uma seção ou anexo denominado "Plano de Atividades" com a descrição das tarefas que serão realizadas pelo estagiário?
49     - Existe a cláusula que menciona a obrigação do estagiário de entregar relatórios de atividades em um período não superior a 6 meses?
50
51     Regras:
52     - Se todos os requisitos obrigatórios estiverem conformes, o contrato é INTEGRADO.
53     - Se algum requisito obrigatório falhar, o contrato é NÃO INTEGRADO.
54     - Requisitos desejáveis não conformes geram apenas observações.
55
56     Instruções de saída:
57     - Use apenas texto simples.
58     - Comece com o título: Relatório de Análise de Contrato de Estágio.
59     - Em seguida, informe o resultado como: "Resultado: CONTRATO INTEGRADO" ou "Resultado: CONTRATO NÃO INTEGRADO" de acordo com a sua análise.
60     - Crie três seções chamadas ▲ Pontos Críticos, 📌 Observações e ✅ Requisitos Atendidos.
61     - Em Pontos Críticos, você colocará apenas os requisitos obrigatórios não atendidos.
62     - Em Observações, você colocará apenas os requisitos desejáveis não atendidos.
63     - Em Requisitos Atendidos, você colocará apenas os requisitos obrigatórios/desejáveis que foram atendidos.
64
65     Contrato: {text}
66     """
67     model = genai.GenerativeModel("gemini-2.5-flash")
68     response = model.generate_content(prompt)
69     return response.text
70
  
```

Fonte: Autoria própria

C.2 Estrutura Geral do Código

A estrutura do módulo de análise do sistema **NoRisk** é composta por funções responsáveis pela interpretação do contrato, comunicação com a API de Inteligência Artificial e retorno estruturado dos resultados. Os principais componentes são descritos a seguir:

- **Função de análise de contratos** (`analyze_contract(text)`): Essa função é responsável por construir um *prompt* detalhado, que é enviado à API generativa **Gemini 2.5 Flash**. O *prompt* orienta o modelo a avaliar se o contrato de estágio é considerado íntegro ou não, com base em requisitos obrigatórios e desejáveis definidos previamente.

O modelo é instruído a:

- Verificar a presença de cláusulas obrigatórias, como Termo de Compromisso, comprovação de matrícula regular, carga horária compatível com o nível de ensino e garantia do recesso remunerado;
- Identificar cláusulas desejáveis, como seguro contra acidentes pessoais e plano de atividades;
- Gerar uma resposta em texto simples, organizada nas seções *Pontos Críticos*, *Observações* e *Requisitos Atendidos*.

APÊNDICE D – Código da Ferramenta

Este apêndice detalha o funcionamento do módulo de *prompts* utilizados no sistema NoRisk para análise automatizada de contratos de estágio. Ele apresenta trechos de código, fluxos de execução e instruções enviadas à Inteligência Artificial, mostrando como os contratos são processados, interpretados e classificados quanto à integridade e conformidade legal. O funcionamento do módulo `analyze.py`, responsável pela análise automatizada de contratos de estágio no sistema NoRisk, ilustrado na Figura D1.

Figura D1 – Essa parte de `analyze.py` o código inicializa a API com FastAPI.



```

analyze.py X
backend > analyze.py > read_pdf
1 import os
2 import tempfile
3 from fastapi import FastAPI, UploadFile, File
4 from fastapi.middleware.cors import CORSMiddleware
5 import google.generativeai as genai
6 from docx import Document
7 from pypdf import PdfReader
8
9 genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
10
11 app = FastAPI()
12
13 # libera acesso do React
14 app.add_middleware(
15     CORSMiddleware,
16     allow_origins=["*"],
17     allow_methods=["*"],
18     allow_headers=["*"],
19 )
20
21 def read_pdf(path):
22     reader = PdfReader(path)
23     return "\n\n".join(
24         page.extract_text() or "" for page in reader.pages
25     )
26
27 def read_docx(path):
28     doc = Document(path)
29     return "\n\n".join(p.text for p in doc.paragraphs if p.text)
30
  
```

Fonte: Autoria própria

D.1 Descrição do Código

O módulo `analyze.py` foi desenvolvido em *Python 3* utilizando o framework *FastAPI* para construção de APIs REST. Abaixo são detalhadas as principais funcionalidades:

- **Importação de bibliotecas:** o arquivo importa módulos para manipulação de arquivos PDF e DOCX (`pypdf` e `python-docx`), gerenciamento de arquivos temporários (`tempfile`), integração com a API generativa da Google (`google.generativeai`) e construção do servidor web (`FastAPI` e `CORS Middleware`).
- **Configuração da API Generativa:** a chave da API é configurada por meio da variável de ambiente `GOOGLE_API_KEY` utilizando:

```
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
```

- **Criação da instância FastAPI:** o objeto `app` é instanciado e configurado com middleware CORS para permitir requisições do frontend em React, possibilitando comunicação segura entre cliente e servidor:

```
app.add_middleware(  
    CORSMiddleware,  
    allow_origins=["*"],  
    allow_methods=["*"],  
    allow_headers=["*"],  
)
```

- **Funções de leitura de arquivos:** duas funções são responsáveis por extrair o texto de contratos:
 - `read_pdf(path)`: percorre cada página de um PDF e retorna o texto concatenado.
 - `read_docx(path)`: percorre os parágrafos de um arquivo DOCX e retorna o texto.

D.2 Funcionamento Detalhado

O fluxo de execução ocorre da seguinte forma:

1. O usuário envia um contrato (PDF ou DOCX) através da interface React.
2. O backend recebe o arquivo via endpoint `/analyze`.
3. O arquivo é armazenado temporariamente e lido, extraindo seu conteúdo textual.
4. O texto é passado para a função `analyze_contract`, que constrói um *prompt* completo e envia à API generativa Gemini 2.5 Flash.
5. O modelo retorna o relatório de análise, indicando se o contrato está íntegro ou não, listando pontos críticos e observações.
6. O backend retorna o resultado ao frontend, que atualiza a interface exibindo o relatório ao usuário.

Na segunda parte do `analyze.py`, D2.

Figura D2 – Continuação do arquivo `analyze.py` mostrando como funciona a rota da API.

```
71 @app.post("/analyze")
72 async def analyze(file: UploadFile = File(...)):
73     suffix = os.path.splitext(file.filename)[1]
74
75     with tempfile.NamedTemporaryFile(delete=False, suffix=suffix) as tmp:
76         tmp.write(await file.read())
77         tmp_path = tmp.name
78
79     if suffix == ".pdf":
80         text = read_pdf(tmp_path)
81     elif suffix == ".docx":
82         text = read_docx(tmp_path)
83     else:
84         return {"error": "Formato não suportado"}
85
86     result = analyze_contract(text)
87     return {"result": result}
88
```

Fonte: Autoria própria

D.3 Rota da API (/analyze):

A rota `analyze` recebe um arquivo enviado via método `POST`, cria um arquivo temporário para leitura, extrai o texto conforme a extensão do documento (PDF ou DOCX) e chama a função `analyze_contract`. O resultado da análise é retornado ao frontend no formato JSON, conforme exemplificado no trecho a seguir:

```
@app.post("/analyze")
async def analyze(file: UploadFile = File(...)):
    # leitura e processamento do contrato
    result = analyze_contract(text)
    return {"result": result}
```

D.4 Endpoint da API

O endpoint `/analyze` é responsável por receber arquivos PDF ou DOCX, extrair o texto e enviar para análise pelo modelo Gemini 2.5 Flash:

- O arquivo enviado é armazenado temporariamente usando a biblioteca `tempfile`;
- Dependendo da extensão, o texto é extraído com `read_pdf(path)` ou `read_docx(path)`;
- A função `analyze_contract` gera o relatório e o resultado é retornado em formato JSON.

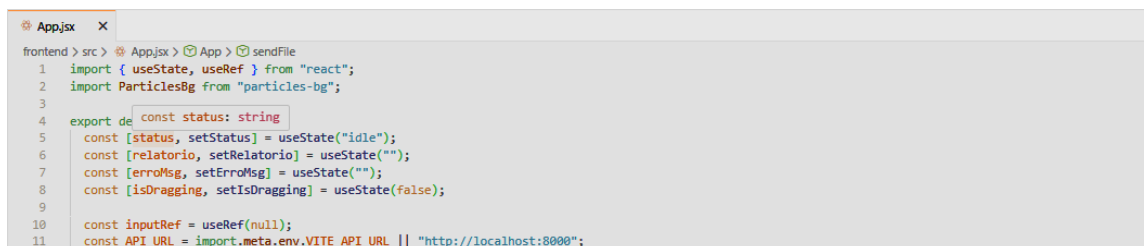
Essa organização garante que o fluxo de análise seja automatizado, seguro e que a interface frontend receba dados estruturados para exibição ao usuário.

Nesta seção, apresenta-se a implementação da interface de usuário do sistema NoRisk, responsável pelo upload de contratos e comunicação com o backend.

D.5 Descrição do Código – Inicialização da Aplicação

A Figura D3 apresenta a primeira parte do arquivo `App.jsx`, responsável pela definição da estrutura inicial da aplicação frontend do sistema NoRisk. Esse trecho compreende a importação de dependências, a configuração de estados internos da aplicação e a definição de parâmetros essenciais para a comunicação com o backend.

Figura D3 – Trecho inicial do código `App.jsx`



```
App.jsx X
frontend > src > App.jsx > App > sendFile
1 import { useState, useRef } from "react";
2 import ParticlesBg from "particles-bg";
3
4 export de const status: string
5 const [status, setStatus] = useState("idle");
6 const [relatorio, setRelatorio] = useState("");
7 const [errorMsg, setErrorMsg] = useState("");
8 const [isDragging, setIsDragging] = useState(false);
9
10 const inputRef = useRef(null);
11 const API_URL = import.meta.env.VITE_API_URL || "http://localhost:8000";
```

Fonte: Autoria própria

Inicialmente, o código importa os *hooks* `useState` e `useRef` da biblioteca *React*, além do componente `ParticlesBg`, utilizado para renderização do fundo animado da interface:

- `useState`: permite o gerenciamento de estados internos da aplicação, possibilitando reações dinâmicas às interações do usuário;
- `useRef`: cria uma referência direta a elementos do DOM, sendo empregado para controlar o campo de upload de arquivos;
- `ParticlesBg`: biblioteca responsável pelo efeito visual de partículas no plano de fundo da aplicação, contribuindo para a experiência do usuário.

Em seguida, é definido o componente principal da aplicação por meio da função `App()`, que encapsula toda a lógica da interface. Dentro dessa função, são declaradas variáveis de estado que controlam o comportamento do sistema:

- `status`: armazena o estado atual da análise do contrato, podendo assumir valores como *idle*, *loading*, *ok* ou *error*;
- `relatorio`: guarda o texto do relatório gerado automaticamente pela Inteligência Artificial após o processamento do contrato;
- `errorMsg`: armazena mensagens de erro retornadas pela API ou geradas durante o processamento;
- `isDragging`: controla o comportamento visual da área de upload durante a ação de arrastar arquivos.

O uso desses estados permite que a interface se atualize de forma responsiva, refletindo visualmente cada etapa do processo de análise contratual.

O objeto `inputRef` é criado por meio do `useRef`, sendo utilizado para acessar e manipular diretamente o elemento HTML do tipo `file input`. Essa abordagem é necessária para disparar ações programáticas, como abrir o seletor de arquivos ou limpar o campo após o envio.

Por fim, o código define a variável `API_URL`, responsável por armazenar o endereço do backend. Esse valor é obtido preferencialmente a partir de uma variável de ambiente (`VITE_API_URL`), garantindo flexibilidade entre ambientes de desenvolvimento e produção. Caso a variável não esteja definida, o sistema utiliza como padrão o endereço local `http://localhost:8000`.

Esse trecho inicial estabelece a base funcional da aplicação frontend, permitindo a correta comunicação com o backend em *FastAPI*, o controle do fluxo de execução da aplicação e a organização da experiência do usuário durante a análise dos contratos.

D.6 Descrição do Código – Envio do Arquivo e Controle de Estado

A Figura D4 apresenta a segunda parte do arquivo `App.jsx`, responsável pelo controle do fluxo da aplicação durante o envio do arquivo para análise, comunicação com o backend e gerenciamento dos estados que refletem o resultado da verificação contratual.

Figura D4 – Funções de controle e envio de arquivo no `App.jsx`

```
App.jsx X
frontend > src > App.jsx > App > resetApp
13 function resetApp() {
14   setStatus("idle");
15   setRelatorio("");
16   setErrMsg("");
17   setIsDragging(false);
18 }
19
20 async function sendFile(file) {
21   if (!file) return;
22
23   setStatus("loading");
24   setRelatorio("");
25   setErrMsg("");
26
27   const formData = new FormData();
28   formData.append("file", file);
29
30   try {
31     const response = await fetch(`${API_URL}/analyze`, {
32       method: "POST",
33       body: formData,
34     });
35
36     if (!response.ok) {
37       throw new Error(`Erro ${response.status} ao conectar ao servidor`);
38     }
39
40     const data = await response.json();
41     setRelatorio(data.result || "Relatório não retornado.");
42
43     if (data.result && data.result.includes("CONTRATO ÍNTEGRO")) {
44       setStatus("ok");
45     } else {
46       setStatus("error");
47     }
48   } catch (err) {
49     console.error(err);
50     setStatus("error");
51     setErrMsg("Falha ao analisar o contrato.");
```

Fonte: Autoria própria

O primeiro bloco define a função `resetApp()`, cuja finalidade é reinicializar o estado da aplicação. Essa função é utilizada sempre que o usuário deseja realizar uma nova análise, garantindo que dados antigos não interfiram na execução atual. Para isso, são redefinidos os seguintes estados:

- `status`: retorna para o estado inicial `idle`, indicando que nenhuma análise está em andamento;
- `relatorio`: é limpo para remover o relatório da análise anterior;
- `errorMsg`: é redefinido para uma string vazia, eliminando mensagens de erro prévias;
- `isDragging`: retorna para `false`, normalizando o comportamento visual da área de upload.

Em seguida, é declarada a função assíncrona `sendFile(file)`, responsável por enviar o contrato selecionado para o backend e processar o retorno da Inteligência Artificial. A função inicia verificando se o arquivo foi fornecido; caso contrário, a execução é interrompida, evitando erros desnecessários.

Antes do envio, o estado da aplicação é atualizado para refletir o início da análise, definindo:

- `status` como `loading`, indicando ao usuário que o processamento está em andamento;
- limpeza dos estados `relatorio` e `errorMsg`, assegurando que apenas informações da análise atual sejam exibidas.

Na sequência, é criado um objeto `FormData`, utilizado para encapsular o arquivo enviado no formato adequado para requisições HTTP do tipo `multipart/form-data`. O arquivo é adicionado ao formulário por meio da chave `"file"`, compatível com a API desenvolvida no backend.

A comunicação com o servidor é realizada por meio da função `fetch`, que envia uma requisição HTTP do tipo `POST` para o endpoint `/analyze`. Caso a resposta do servidor não seja bem-sucedida, é lançada uma exceção contendo o código de erro HTTP retornado, permitindo o tratamento adequado da falha.

Após uma resposta válida, o conteúdo retornado é convertido para o formato JSON. O relatório produzido pela Inteligência Artificial é então armazenado no estado `relatorio`. Caso o backend não retorne um resultado válido, é exibida uma mensagem padrão indicando a ausência do relatório.

Por fim, o código avalia o conteúdo textual do relatório. Caso seja identificado o marcador "CONTRATO ÍNTEGRO", o estado da aplicação é atualizado para *ok*, representando conformidade contratual. Caso contrário, o estado é definido como *error*, indicando a presença de inconsistências ou riscos jurídicos.

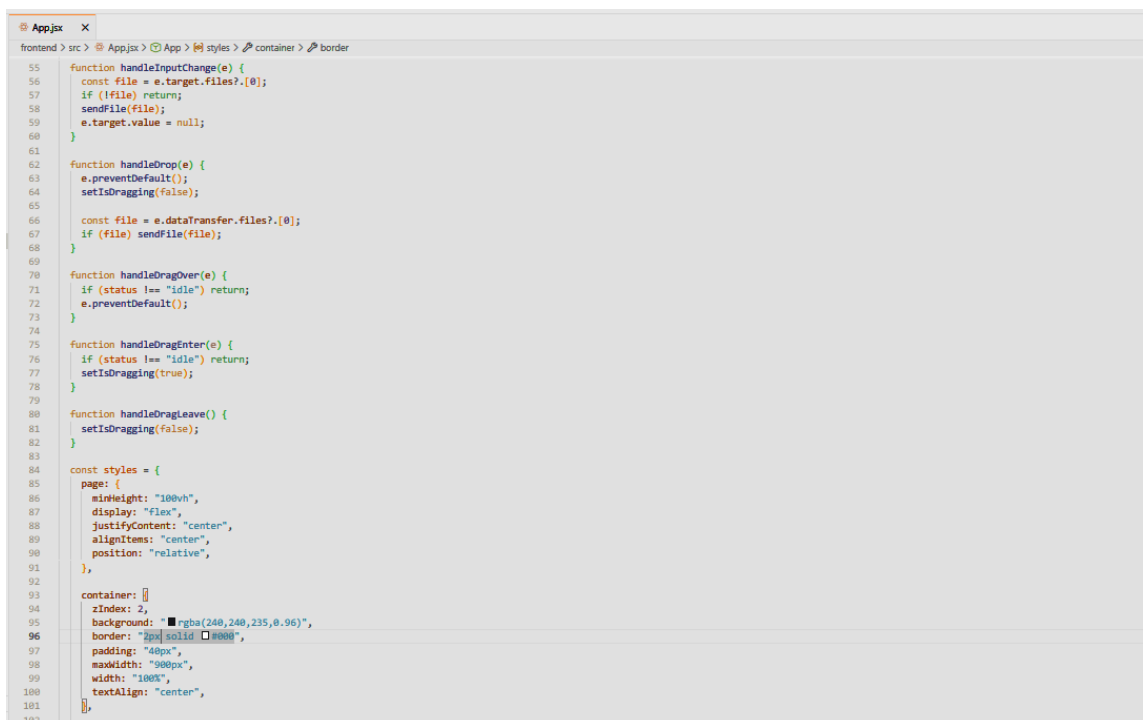
O bloco `try-catch` garante robustez à aplicação, permitindo o tratamento de exceções decorrentes de falhas de comunicação com o servidor ou erros inesperados durante o processamento. Em situações de erro, o sistema registra o ocorrido no console, altera o estado para *error* e exibe ao usuário uma mensagem informativa, assegurando clareza e usabilidade.

Esse conjunto de funções é essencial para a integração entre frontend e backend, possibilitando o envio seguro dos contratos, o processamento automatizado pela Inteligência Artificial e a apresentação clara dos resultados ao usuário.

D.7 Descrição do Código – Interação com o Usuário e Estilização da Interface

As Figuras D5 e D6 apresentam, respectivamente, os trechos do arquivo `App.jsx` responsáveis pelo tratamento das interações do usuário durante o envio do contrato e pela definição dos estilos visuais da interface do sistema NoRisk.

Figura D5 – Tratamento de eventos de upload e arrastar-e-soltar no `App.jsx`



```
App.jsx X
frontend > src > App.jsx > App > styles > container > border
55 function handleInputChange(e) {
56   const file = e.target.files?.[0];
57   if (!file) return;
58   sendFile(file);
59   e.target.value = null;
60 }
61
62 function handleDrop(e) {
63   e.preventDefault();
64   setIsDragging(false);
65
66   const file = e.dataTransfer.files?.[0];
67   if (file) sendFile(file);
68 }
69
70 function handleDragOver(e) {
71   if (status !== "idle") return;
72   e.preventDefault();
73 }
74
75 function handleDragEnter(e) {
76   if (status !== "idle") return;
77   setIsDragging(true);
78 }
79
80 function handleDragLeave() {
81   setIsDragging(false);
82 }
83
84 const styles = {
85   page: {
86     minHeight: "100vh",
87     display: "flex",
88     justifyContent: "center",
89     alignItems: "center",
90     position: "relative",
91   },
92
93   container: [
94     zIndex: 2,
95     background: "rgba(240,240,235,0.96)",
96     border: "2px solid #000",
97     padding: "40px",
98     maxWidth: "900px",
99     width: "100%",
100    textAlign: "center",
101  ],
102 }
```

Fonte: Autoria própria

O primeiro bloco de código apresentado na Figura D5 é responsável pelo controle dos eventos de entrada de arquivos, contemplando tanto o método tradicional de seleção via campo de upload quanto a funcionalidade de *drag and drop*.

A função `handleInputChange(e)` é acionada sempre que o usuário seleciona um arquivo através do elemento `input[type=file]`. O código extrai o primeiro arquivo selecionado, valida sua existência e, em caso positivo, dispara a função `sendFile(file)`, responsável pelo envio do documento ao backend. Por fim, o valor do campo de entrada é resetado, permitindo que o mesmo arquivo possa ser selecionado novamente, caso necessário.

A função `handleDrop(e)` implementa o comportamento de arrastar e soltar arquivos sobre a área de upload. O evento padrão do navegador é cancelado para evitar comportamentos indesejados, o estado visual de arraste é normalizado e, caso um arquivo seja identificado, este é imediatamente enviado para análise por meio da função `sendFile(file)`.

O controle visual da área de upload é reforçado pelas funções `handleDragOver(e)`, `handleDragEnter(e)` e `handleDragLeave()`. Essas funções governam o comportamento da interface quando o usuário interage arrastando um arquivo sobre a aplicação, garantindo que:

- interações só ocorram quando o sistema estiver no estado *idle*;
- o estado `isDragging` seja atualizado corretamente, permitindo feedback visual imediato;
- a experiência do usuário seja intuitiva e consistente.

Figura D6 – Definição de estilos visuais da interface no App.jsx

```
App.jsx x
frontend > src > App.jsx > App > styles > container > border
103 title: {
104   fontFamily: "Times New Roman, serif",
105   fontSize: "clamp(1.8rem, 4vw, 3rem)",
106   marginBottom: "25px",
107 },
108
109
110 box: {
111   minHeight: "260px",
112   border: "2px dashed #000",
113   display: "flex",
114   flexDirection: "column",
115   alignItems: "center",
116   justify-content: "center",
117   padding: "20px",
118   cursor: status === "idle" ? "pointer" : "default",
119   background: isDragging ? "rgba(0,0,0,0.05)" : "transparent",
120   userSelect: status === "idle" ? "none" : "text",
121 },
122
123 img: {
124   width: "90px",
125   marginBottom: "12px",
126 },
127
128 report: {
129   textAlign: "left",
130   whiteSpace: "pre-wrap",
131   maxHeight: "300px",
132   overflow: "auto",
133   border: "1px solid #000",
134   padding: "12px",
135   margin: "12px",
136   userSelect: "text",
137 },
138
139 button: {
140   margin: "20px",
141   padding: "10px 18px",
142   border: "1px solid #000",
143   background: "#fff",
144   cursor: "pointer",
145   fontWeight: "bold",
146 },
147
148 errorText: {
149   color: "#900",
150   margin: "10px",
151   fontWeight: "bold",
152 }
```

Fonte: Autoria própria

A Figura D6 ilustra o objeto `styles`, responsável pela estilização da interface do sistema utilizando estilos inline. Essa abordagem permite maior controle visual e facilita a adaptação dinâmica da interface conforme o estado da aplicação.

O estilo `page` define a estrutura base da aplicação, garantindo centralização vertical e horizontal do conteúdo, ocupação total da altura da tela e posicionamento relativo para suportar camadas visuais, como o fundo animado.

O estilo `container` delimita a área principal do sistema, aplicando bordas, espaçamento interno e uma coloração suave, assegurando contraste adequado e boa legibilidade do conteúdo exibido.

O estilo `title` configura a identidade visual do título principal, utilizando fonte serifada compatível com o contexto acadêmico, tamanhos responsivos e espaçamento adequado.

A área de upload e exibição de resultados é definida pelo estilo `box`, que utiliza bordas tracejadas para indicar a área interativa, alinhamento flexível dos elementos internos e alterações visuais condicionadas ao estado de arraste (`isDragging`) e ao estado atual da aplicação.

Os estilos `img` e `report` são responsáveis, respectivamente, pela padronização dos ícones apresentados ao usuário e pela formatação do relatório retornado pela Inteligência Artificial, garantindo rolagem vertical, preservação de quebras de linha e legibilidade do conteúdo textual.

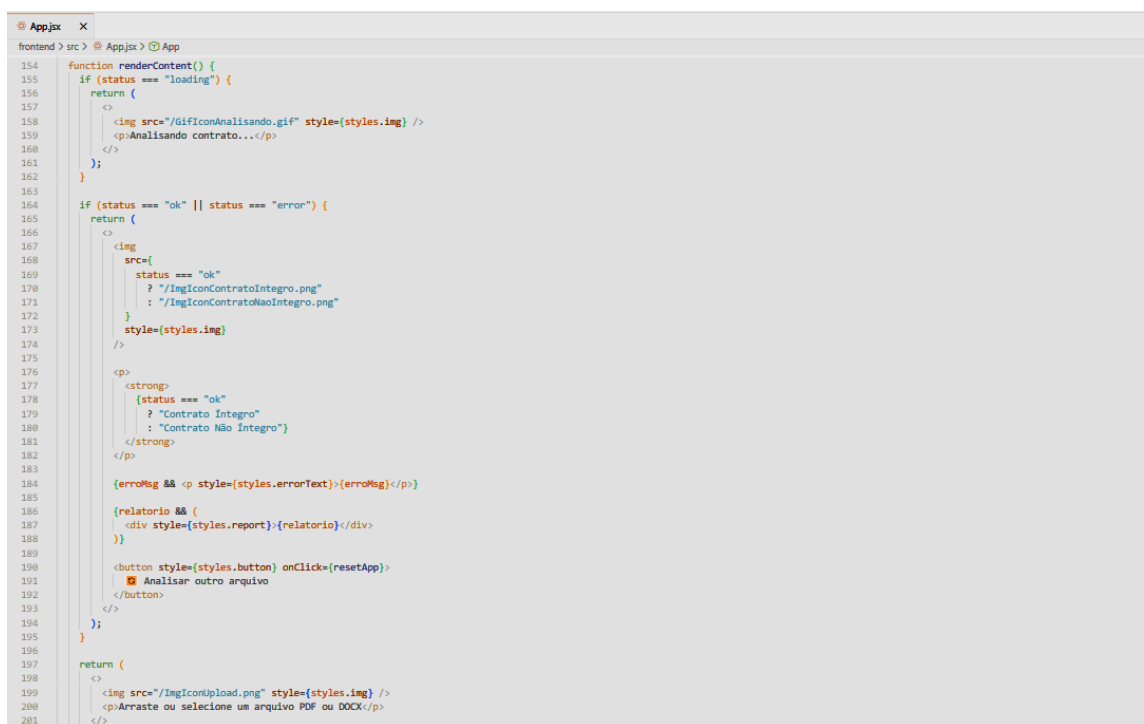
O estilo `button` define a aparência dos botões de ação, enquanto `errorText` destaca mensagens de erro de forma visualmente perceptível, utilizando cor contrastante e ênfase tipográfica.

Em conjunto, os trechos apresentados nas Figuras D5 e D6 demonstram como o sistema NoRisk alia recursos de interação intuitiva, feedback visual imediato e organização estética para proporcionar uma experiência de uso clara, acessível e funcional, alinhada aos objetivos de apoio à análise de contratos de estágio.

D.8 Renderização Condicional da Interface e Estrutura Final da Aplicação

As Figuras D7 e D8 apresentam os trechos finais do arquivo `App.jsx`, responsáveis pela renderização dinâmica do conteúdo exibido ao usuário e pela organização estrutural da interface do sistema NoRisk.

Figura D7 – Função de renderização condicional do conteúdo da interface



```
154 function renderContent() {
155   if (status === "loading") {
156     return (
157       <>
158         
159         <p>Analisando contrato...</p>
160       </>
161     );
162   }
163
164   if (status === "ok" || status === "error") {
165     return (
166       <>
167         <img
168           src={
169             status === "ok"
170               ? "/imgIconContratoIntegro.png"
171               : "/imgIconContratoNaoIntegro.png"
172           }
173           style={styles.img}
174         />
175
176         <p>
177           <strong>
178             {status === "ok"
179               ? "Contrato Integro"
180               : "Contrato Não Integro"}
181           </strong>
182         </p>
183
184         {errorMsg && <p style={styles.errorText}>{errorMsg}</p>}
185
186         {relatorio && (
187           <div style={styles.report}>{relatorio}</div>
188         )}
189
190         <button style={styles.button} onClick={resetApp}>
191           🗑 Analisar outro arquivo
192         </button>
193       </>
194     );
195   }
196
197   return (
198     <>
199       
200       <p>Arraste ou selecione um arquivo PDF ou DOCX</p>
201     </>
  
```

Fonte: Autoria própria

A função `renderContent()` é responsável por determinar, de forma condicional, quais elementos visuais e informacionais serão exibidos na interface, conforme o estado atual da aplicação armazenado na variável `status`. Essa estratégia permite atualizar a interface de maneira automática e coerente com o fluxo de análise do contrato.

Quando o estado é igual a `loading`, o sistema exibe uma imagem animada acompanhada do texto “Analisando contrato...”, indicando ao usuário que o arquivo foi recebido e está sendo processado pela Inteligência Artificial. Esse feedback visual evita ambiguidades e melhora a experiência de uso.

Nos casos em que o estado assume os valores `ok` ou `error`, o sistema apresenta um ícone correspondente ao resultado da análise — contrato íntegro ou não íntegro — além de um título textual que reforça o diagnóstico obtido. Caso exista alguma mensagem de erro armazenada na variável `erroMsg`, esta é exibida em destaque, permitindo ao usuário compreender falhas de processamento ou comunicação.

O relatório textual retornado pela API de análise é apresentado em um contêiner próprio, com formatação adequada para leitura contínua e preservação do conteúdo gerado pelo modelo de Inteligência Artificial. Adicionalmente, um botão de ação permite reiniciar o fluxo da aplicação por meio da função `resetApp()`, possibilitando a análise de um novo contrato sem a necessidade de recarregar a página.

Quando o sistema se encontra no estado inicial (`idle`), a função retorna uma interface simplificada, exibindo apenas o ícone de upload e a instrução para que o usuário arraste ou selecione um arquivo nos formatos PDF ou DOCX.

Figura D8 – Estrutura principal da aplicação e integração dos componentes visuais



```
205 return (  
206   <div style={styles.page}>  
207     <ParticlesBg type="cobweb" color="#8080" bg />  
208   </div>  
209   <div style={styles.container}>  
210     <h1 style={styles.title}>Analisador de Contratos de Estágio</h1>  
211   </div>  
212   <div style={styles.box}>  
213     style={styles.box}  
214     onClick={() => status === "idle" && inputRef.current.click()}  
215     onDrop={handleDrop}  
216     onDragOver={handleDragOver}  
217     onDragEnter={handleDragEnter}  
218     onDragLeave={handleDragLeave}  
219   >  
220     {renderContent()}  
221   </div>  
222   <input  
223     ref={inputRef}  
224     type="file"  
225     hidden  
226     accept=".pdf, .docx"  
227     onChange={handleInputChange}  
228   />  
229 </div>  
230 </div>  
231 </div>  
232 </div>  
233 );
```

Fonte: Autoria própria

A Figura D8 ilustra o retorno principal do componente `App`, que define a estrutura visual completa da aplicação. O elemento externo utiliza o estilo `page`, garantindo centralização do conteúdo e posicionamento adequado dos elementos gráficos.

Como plano de fundo, é utilizado o componente `ParticlesBg`, configurado com o efeito `cobweb`, que adiciona um efeito visual sutil sem comprometer a legibilidade ou a usabilidade da interface. Esse recurso contribui para uma experiência visual mais agradável, mantendo o foco no conteúdo funcional.

No contêiner central, o sistema apresenta o título “Analisador de Contratos de Estágio”, reforçando o propósito da ferramenta. A área principal de interação é composta por um bloco (`div`) estilizado que captura os eventos de clique, arraste e soltura de arquivos, integrando as funções de tratamento de eventos previamente definidas.

Dentro desse contêiner, o conteúdo exibido é totalmente controlado pela função `renderContent()`, centralizando a lógica de apresentação da interface. O campo de upload de arquivos permanece oculto, sendo acionado programaticamente quando o usuário clica na área indicada, garantindo uma interface mais limpa e intuitiva.

Em conjunto, os trechos apresentados nas Figuras D7 e D8 demonstram como o sistema NoRisk organiza seu fluxo de interação, fornece feedback contínuo ao usuário e apresenta os resultados da análise de forma clara e acessível, reforçando seu papel como ferramenta de apoio à verificação de contratos de estágio.