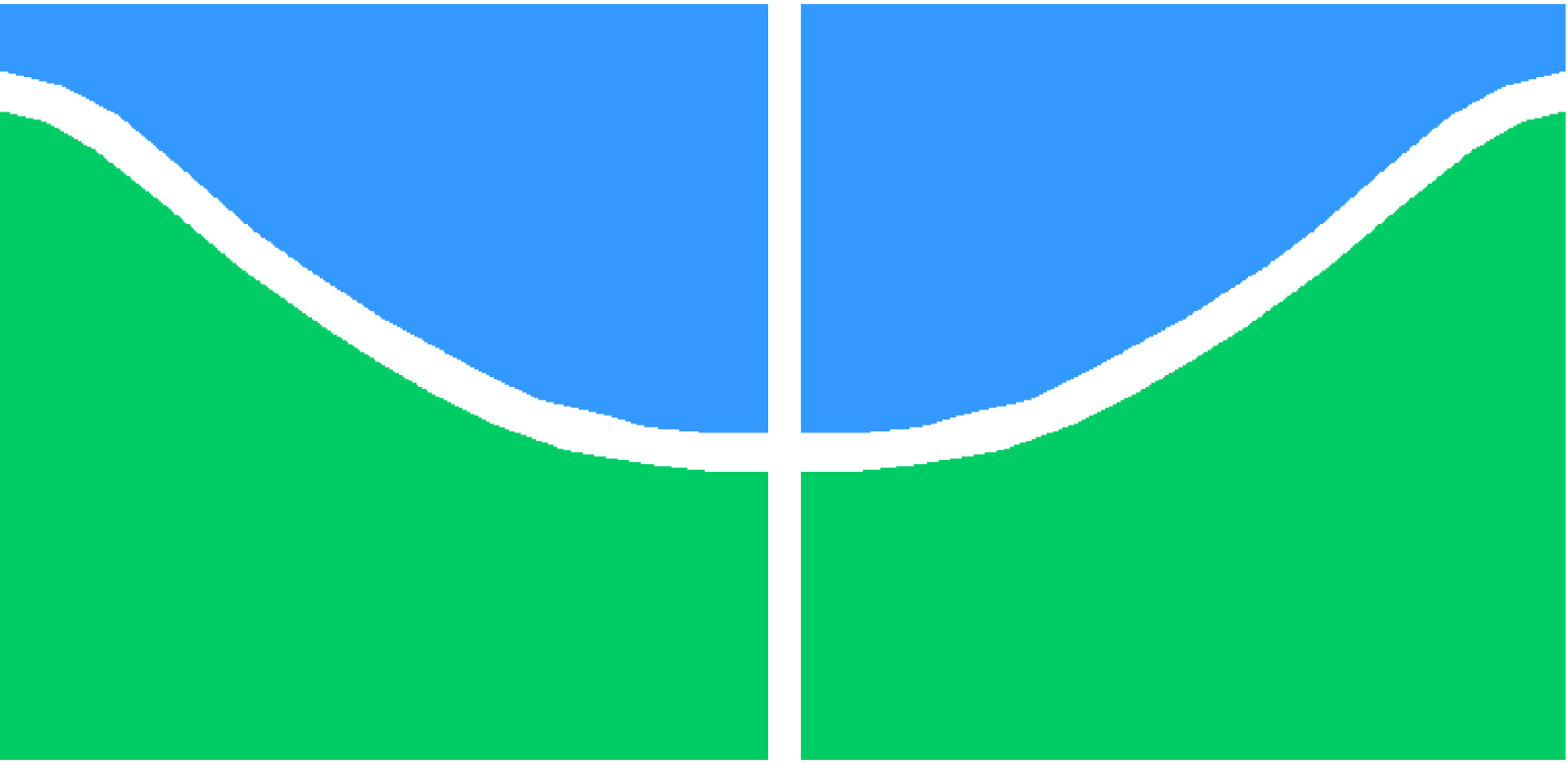

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

Utilização de *Large Language Models* no desenvolvimento de um *chatbot* multimodal para consultoria jurídico-trabalhista

Autor: Álvaro Henrique de S. Gouvea, Luís Guilherme G. Lins

Orientador: Prof. Dr. Sergio Antonio Andrade De Freitas

Brasília, DF

2023

Álvaro Henrique de S. Gouvea, Luís Guilherme G. Lins

Utilização de *Large Language Models* no desenvolvimento de um *chatbot* multimodal para consultoria jurídico-trabalhista

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Sergio Antonio Andrade De Freitas

Coorientador: Prof. Dr. Giovanni Almeida Santos

Brasília, DF

2023

1 Introdução

No cenário contemporâneo, marcado pela incessante evolução tecnológica, a integração de inovações como os *Large Language Models* (LLMs) assume um papel proeminente nas mais diversas esferas profissionais. No universo jurídico, especificamente na área trabalhista, a complexidade e dinamismo das leis e regulamentações demandam soluções eficientes para oferecer suporte a profissionais e indivíduos na compreensão e aplicação correta desses preceitos legais. Nesse contexto, a utilização de *chatbots* impulsionados por LLMs surge como uma ferramenta promissora para facilitar o acesso à consultoria jurídico-trabalhista, oferecendo respostas rápidas, precisas e contextualmente relevantes de forma mais acessível.

1.1 Motivação

A motivação subjacente a este estudo reside na necessidade de aprimorar a acessibilidade e eficácia dos serviços de consultoria jurídica no âmbito trabalhista. A abordagem tradicional, caracterizada por custos elevados e limitações temporais, pode ser superada pela implementação de soluções tecnológicas inovadoras. A utilização de LLMs no desenvolvimento de *chatbots* para consultoria jurídico-trabalhista propicia uma resposta ágil e eficiente às dúvidas legais, contribuindo para a democratização do acesso à informação jurídica de qualidade.

1.2 Objetivo

O objetivo principal desse projeto é o desenvolvimento de um *chatbot* multimodal especializado em consultoria jurídico-trabalhista via aplicação web.

1.3 Metodologia

A metodologia adotada neste projeto seguiu uma abordagem estruturada em quatro etapas principais, cada uma desempenhando um papel crucial no desenvolvimento e na implementação do chatbot multimodal para consultoria jurídico-trabalhista. As etapas são descritas a seguir:

Levantamento de Referências e Dados: Inicialmente, foi realizado um levantamento extensivo de referências bibliográficas e dados relacionados ao tema do trabalho. Esta etapa envolveu a revisão de artigos acadêmicos, livros, relatórios técnicos e estudos de caso sobre o uso de LLMs e *chatbots* no contexto jurídico e trabalhista. Além disso,

dados relevantes foram coletados de fontes confiáveis para garantir uma base sólida de informações que sustentasse o treinamento do modelo.

Implementação do LLM: Com os dados devidamente coletados e tratados, procedeu-se ao treinamento do *Large Language Model* (LLM). Inicialmente, a proposta consistia em utilizar um LLM pré-treinado, realizando *fine-tuning* para adequá-lo ao contexto específico do projeto. No entanto, devido a limitações de recursos computacionais e financeiros, optou-se por uma abordagem alternativa, empregando um sistema multi-modelo com LLMs já treinados. Esses modelos receberam instruções específicas para se comportarem conforme o esperado, evitando assim a necessidade de personalização extensa e reduzindo os custos sem comprometer os objetivos do projeto.

Desenvolvimento da Aplicação Web: Paralelamente à implementação dos modelos, foi desenvolvida uma aplicação web que serve como interface para interação com o *chatbot*. Esta etapa envolveu a escolha das tecnologias de *front-end* e *back-end* adequadas, o design da interface do usuário e a integração do LLM com a plataforma web. A aplicação foi projetada para ser intuitiva e acessível, permitindo que usuários finais interajam de maneira eficiente e eficaz com o *chatbot*.

Documentação e Escrita dos Resultados: Por fim, todos os resultados, descobertas obtidas ao longo do projeto foram sistematicamente documentados e apresentados nesta monografia. A documentação incluiu uma análise dos dados coletados, a metodologia de desenvolvimento, as dificuldades enfrentadas e as soluções implementadas, além de uma discussão detalhada sobre os resultados alcançados e as contribuições do projeto para a área de consultoria jurídico-trabalhista.

1.4 Estrutura da Monografia

Esta monografia está organizada em seis tópicos principais, cada um desempenhando um papel fundamental na construção do argumento central e na apresentação dos resultados obtidos ao longo da pesquisa. A estrutura é delineada da seguinte maneira:

Introdução: Este capítulo inicial fornece uma visão geral do tema abordado, estabelecendo o contexto e a relevância da pesquisa. Apresenta os objetivos gerais e específicos do estudo, além das questões de pesquisa que orientaram o desenvolvimento do trabalho. A introdução também destaca a justificativa para a escolha do tema e a importância do estudo para a área acadêmica e prática.

Revisão Bibliográfica: Neste capítulo, são exploradas as principais teorias, conceitos e estudos relacionados ao tema da monografia. A revisão da literatura oferece uma base sólida para o entendimento do contexto acadêmico e tecnológico em que a pesquisa se insere. Discute as diferentes abordagens existentes e identifica lacunas no conhecimento

atual que justificam a realização do estudo.

Estudo Comparativo entre Diferentes LLMs: Este capítulo é dedicado a uma análise detalhada e comparativa dos modelos de linguagem de grande escala (LLMs) utilizados no estudo. São apresentados os critérios de seleção dos modelos, as metodologias empregadas para a comparação e os resultados obtidos. Essa análise crítica visa destacar as diferenças, vantagens e limitações de cada modelo, contribuindo para uma compreensão mais profunda de suas capacidades e aplicações.

Desenvolvimento do Projeto: Neste segmento, a monografia detalha a proposta do projeto desenvolvido durante a pesquisa assim como o seu processo prático de desenvolvimento. São descritos os objetivos específicos do projeto, as estratégias adotadas para sua implementação e as etapas de desenvolvimento. O capítulo também inclui uma discussão sobre as escolhas metodológicas e técnicas que guiaram a execução do projeto, bem como os desafios enfrentados e as soluções encontradas.

O Sistema: Este capítulo apresenta os resultados obtidos com o desenvolvimento e aplicação do projeto proposto. São analisados os dados coletados, destacando as descobertas principais e as implicações dos achados em relação aos objetivos iniciais da pesquisa. A apresentação dos resultados visa fornecer uma visão clara e objetiva do que foi alcançado, utilizando gráficos e tabelas, quando necessário, para facilitar a compreensão.

Considerações Finais: O capítulo final sintetiza as principais contribuições do estudo, refletindo sobre os resultados obtidos e suas implicações teóricas e práticas. São discutidas as limitações da pesquisa e apresentadas sugestões para trabalhos futuros, destacando o impacto potencial da pesquisa na área em questão. As considerações finais encerram a monografia com uma reflexão crítica sobre o percurso investigativo e as perspectivas de continuidade do trabalho.

2 Revisão Bibliográfica

A revolução tecnológica nas últimas décadas tem trazido inúmeras inovações para diversos campos, e a área jurídico-trabalhista não é exceção. A utilização de tecnologias de inteligência artificial (IA) tem se mostrado uma ferramenta promissora na busca por soluções eficientes e eficazes. Neste contexto, esta revisão bibliográfica abordará os conceitos fundamentais de *chatbots* e *Large Language Models* (LLMs), bem como um estudo comparativo e possíveis aplicações na área jurídico-trabalhista, considerando também as expectativas e atividades desse domínio.

2.1 Definição de *Chatbots*

Em 1950, Alan Turing introduziu o teste que levaria a uma revolução na interação entre humanos e máquinas: o Teste de Turing, uma avaliação fundamental da capacidade de uma máquina reproduzir pensamento inteligente de maneira indistinguível do humano. O surgimento desse teste marcou o início da popularização do conceito de *chatbots*, atualmente também referidos como assistentes virtuais ou agentes de conversação. O Teste de Turing incentivou a criação de Eliza em 1966 por Joseph Weizenbaum, amplamente reconhecida como o primeiro *chatbot*. Eliza adotou uma estratégia singular, simulando o papel de terapeuta e respondendo às entradas dos usuários com perguntas para dificultar seu reconhecimento como uma máquina (ADAMOPOULOU; MOUSSIADES, 2020).

As diferenças entre *chatbots* orientados a tarefas, como Eliza, e assistentes virtuais mais avançados, como o ChatGPT, estão no escopo de suas interações e a sua relação com dados. Eliza foi desenvolvida em 1966 com escopo e proposta limitadas, sua função era agir como uma terapeuta e questionar o usuário com o objetivo de não ser facilmente identificada como uma máquina. Já alguns *chatbots* mais modernos como o ChatGPT, Siri e Alexa fazem parte de outro grupo de *chatbots*, os orientados por dados e preditivos. Essas inteligências são muito mais sofisticadas e interativas, utilizando de tecnologias como *Natural Language Processing* (NLP) e *Machine Learning* (ML) para melhor compreender seus contextos e adaptar suas respostas de acordo, aplicando diferentes técnicas de análise para permitir personalização de acordo com o perfil do usuário e seu histórico de interações. (ORACLE, 2023)

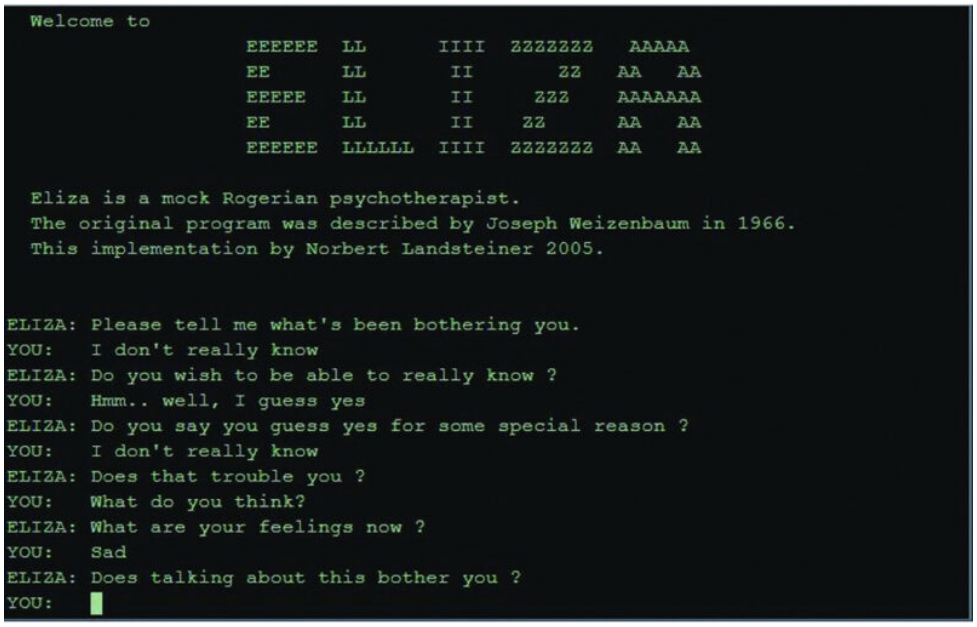

Figura 1 – Conversa com Eliza em sua implementação de 2005 por Norbert Landsteiner (RODRIGUES, 2023)

Eliza baseou suas interações em identificação de padrões simples e mecanismos de resposta modelados. De acordo com as observações da Oracle em seu artigo "O que é um chatbot", tais características posicionam Eliza como um pioneiro entre os assistentes virtuais orientados a tarefas ou declarativos (ORACLE, 2023).

2.1.1 Chatbots Declarativos

Os *chatbots* declarativos seguem um conjunto pré-definido de instruções e respostas, utilizando pouco ou nenhum aprendizado de máquina e atualmente se apoiando mais na utilização de tecnologias como NLP, que é possível concluir que é um avanço tecnológico não utilizado por Eliza. Essas características gerais fazem com que os *chatbots* orientados a tarefas sejam mais adequados para tarefas simples e específicas. Por exemplo, um *chatbot* declarativo pode ser usado para atendimento ao cliente, suporte técnico e melhor direcionamento do usuário (MICROSOFT, 2023). Eles funcionam bem em cenários onde as interações são padronizadas e previsíveis.

2.1.2 Chatbots Conversacionais

Por outro lado, os *chatbots* conversacionais, como os baseados no Llama 2 (META, 2023), por exemplo, utilizam algoritmos de aprendizado de máquina e NLP para entender contextos complexos e fornecer respostas mais sofisticadas e personalizadas. Isso significa que eles podem lidar com perguntas mais abertas e variadas, e então se adaptarem à

qualquer escopo (ORACLE, 2023). Por esse motivo podem ser adotados em diversos setores, inclusive o jurídico-trabalhista.

Além de sua capacidade de aprendizado contínuo, os *chatbots* conversacionais também apresentam maior flexibilidade no que diz respeito ao ajuste fino do escopo, devido a não necessidade de rotulação dos dados de treinamento, visto que os modelos usam técnicas de aprendizado não-supervisionadas (NVIDIA, 2023).

Característica	**Chatbots descritivos**	**Chatbots conversacionais**
Natureza da Interação	Fornecem informações específicas e diretas.	Engajam em diálogos mais naturais e interativos.
Escopo da Conversa	Geralmente focados em tarefas específicas.	Capazes de lidar com uma variedade de assuntos.
Interatividade	Limitada; responde a comandos específicos.	Maior interação, entendendo contextos e nuances.
Aprendizado de Máquina	Menos dependência de técnicas de aprendizado.	Pode envolver técnicas avançadas de aprendizado.
Personalização	Menos personalização; respostas predefinidas.	Adapta-se às preferências e histórico do usuário.
Complexidade da Lógica	Lógica mais simples e direta.	Pode envolver lógica mais complexa e contextual.
Aplicações Comuns	FAQs, assistentes de ajuda específicos.	Atendimento ao cliente, assistentes virtuais.
Desenvolvimento Inicial	Mais rápido, com foco em respostas específicas.	Pode exigir mais tempo devido à complexidade.
Experiência do Usuário	Adequado para interações transacionais.	Oferece uma experiência mais próxima de uma conversa.

Tabela 1 – Comparação entre Chatbots Descritivos e Conversacionais (ORACLE, 2023), (NVIDIA, 2023)

2.2 Definição de *Large Language Models*

Os *Large Language Models*, ou modelos de linguagem de grande escala, representam uma conquista significativa na área de inteligência artificial, destacando-se em diversas tarefas relacionadas à geração de texto e compreensão da linguagem natural. Esses modelos, como o GPT-3 e seu sucessor, o GPT-4 (OPENAI, 2023), são treinados em enormes volumes de dados textuais, abrangendo uma variedade de fontes e contextos linguísticos. Os LLMs tem como seu diferencial a capacidade de compreender características semânticas, sintáticas e contextuais da linguagem humana e consequentemente não apenas reproduzir informações coerentes mas agir de acordo com o contexto dado no processo de geração de novos textos.

Os *Large Language Models* possuem essas características pois são parte de uma classe de arquiteturas de *deep learning* chamadas de redes transformadoras. Uma rede neural transformadora consiste de um modelo de rede que é capaz de aprender um contexto por meio da identificação de relacionamentos de dados sequenciais, como as palavras em uma frase. Essas redes são compostas de diversas camadas, como camadas de atenção, que buscam simular a forma como os humanos prestam atenção em diferentes palavras, camadas de normalização, entre outras (NVIDIA, 2023).

Esses modelos são altamente flexíveis e podem ser usados para uma ampla variedade de tarefas, desde tradução de idiomas até geração de texto em estilo literário. Eles também são capazes de realizar tarefas de processamento de linguagem natural, como resumir documentos, responder a perguntas e manter diálogos naturais. Devido à sua

capacidade de compreender e gerar texto de alta qualidade, os LLMs podem se mostrar valiosos em aplicações jurídicas, particularmente na consultoria jurídico-trabalhista.

A aplicação de *Large Language Models* no contexto jurídico pode incluir a geração de documentos legais, a revisão automática de contratos e consulta de jurisprudências. Esses modelos são capazes de analisar grandes volumes de textos de forma rápida e precisa, identificando precedentes relevantes e fornecendo resumos legais detalhados. Isso economiza um tempo considerável para os profissionais do direito, permitindo que eles se concentrem em tarefas que requerem interpretação e estratégia.

2.3 Relação entre *Chatbots* e *Large Language Models*

A interconexão existente entre as tecnologias de *chatbots* e LLMs pode ser compreendida a partir das categorias de *chatbots* previamente discutidas. A visão convencional dessas ferramentas muitas vezes as associa predominantemente a assistentes virtuais em setores específicos, como bancos ou empresas, direcionando clientes para canais de comunicação apropriados. No entanto, essa percepção é limitada e não abrange completamente o vasto potencial de aplicação dessas tecnologias, que estão se tornando cada vez mais entrelaçadas no cenário tecnológico contemporâneo.

Os *chatbots* com IA, como o ChatGPT, podem ser vistos como uma categoria de *chatbots* que se beneficiam dos recursos dos LLMs, combinando interações conversacionais com poderosas capacidades de processamento de linguagem natural. Isso significa que, enquanto um *chatbot* tradicional poderia responder a perguntas básicas sobre direitos trabalhistas, um *chatbot* com IA baseado em *Large Language Models* pode oferecer orientações mais detalhadas e específicas, analisar contratos complexos e fornecer análises jurídicas mais avançadas.

Essa combinação de *chatbots* conversacionais e LLMs abre novas possibilidades para a consultoria jurídico-trabalhista, pois permite o usuário à consultar suas questões sem necessariamente envolver e pagar pelos serviços de advogados, como também a automação de alguns serviços por parte dos profissionais da área do Direito, tal como pesquisa e análise de casos.

2.4 Licenças *Open Source*

Outro conceito importante para ser analisado nesse projeto é o conceito de licenças ou tecnologias *open source*, que refere-se a um tipo de licença de software que permite aos usuários visualizar, modificar e distribuir o código-fonte de um programa. Diferentemente das licenças proprietárias, que restringem o acesso ao código-fonte e impõem limitações rigorosas sobre o uso e a distribuição de software, as licenças *open source* promovem a

transparência e a colaboração. Exemplos populares dessas licenças incluem a GNU *General Public License* (GPL), a MIT *License* e a Apache *License*.

A utilização de produtos com licença *open source* no desenvolvimento de um projeto pode ter um impacto significativo. Primeiramente, o acesso ao código-fonte desses produtos permite uma maior compreensão e controle sobre as tecnologias utilizadas. Além disso, a escolha de produtos com essas licenças permite uma maior facilidade de desenvolvimento, já que não existe custo adicional para utilização do produto.

Dessa a forma, a equipe do projeto considera de alta relevância a escolha de tecnologias *open source* no desenvolvimento do projeto proposto, visando maior transparência, facilidade de desenvolvimento e menor custo de desenvolvimento. Os alunos também creem que o conceito por trás dessa licenças se alinha com a visão do produto final, que busca trazer maior acesso e informação ao público geral, sem tantas barreiras de conhecimento ou recursos financeiros.

2.5 Atividades e expectativas da área jurídico-trabalhista

A área jurídico-trabalhista envolve um amplo espectro de atividades, incluindo aconselhamento jurídico, elaboração de contratos, pesquisa de jurisprudência, entre outras (TIRADENTES, 2021). As expectativas nesse campo incluem a eficiência, precisão e economia de tempo. Os *chatbots* e LLMs podem desempenhar um papel fundamental na automação de tarefas rotineiras, na pesquisa legal e na revisão de documentos, permitindo que profissionais do direito se concentrem em tarefas mais estratégicas e de alto valor agregado.

Por exemplo, um *chatbot* conversacional pode ser usado para responder a perguntas comuns sobre direitos trabalhistas, como o cálculo de férias ou a legislação sobre horas extras. Isso economiza tempo para os advogados, permitindo que eles se concentrem em casos mais complexos que requerem análise especializada. Além de possuir a capacidade de aprender com interações passadas, aprimorando suas respostas ao longo do tempo. Esses *chatbots* podem ser treinados em um escopo específico da área jurídica e incorporar esse conhecimento para fornecer orientações mais precisas e detalhadas aos usuários. Essa flexibilidade os torna ideais para a consultoria jurídico-trabalhista, onde questões podem variar amplamente em complexidade e contexto.

Além disso, os *Large Language Models* podem ser empregados na revisão de contratos de trabalho e outros documentos legais. Eles podem identificar cláusulas potencialmente problemáticas, destacar áreas de risco e oferecer sugestões para aprimorar o texto do contrato. Isso não apenas economiza tempo, mas também ajuda a garantir que os contratos sejam redigidos de forma precisa e em conformidade com a legislação vigente.

Os benefícios dos *chatbots* e *Large Language Models* na área jurídico-trabalhista vão além da eficiência. Eles também ajudam a reduzir o risco de erros humanos, o que é particularmente crítico em questões legais. Um erro na interpretação da lei pode ter sérias consequências, e as ferramentas de IA estão se tornando uma camada adicional de verificação e precisão nas operações jurídicas (PROMAD, 2022).

Além das expectativas de eficiência e precisão, os avanços contínuos em IA estão moldando as atividades da área jurídico-trabalhista. A capacidade de acesso rápido a informações atualizadas, análises de risco mais precisas e aprimoramentos na tomada de decisões são aspectos cruciais que a tecnologia está trazendo para a prática jurídica. Isso não apenas beneficia os profissionais do direito, mas também oferece aos clientes serviços mais transparentes e confiáveis.

Essas tecnologias podem ser uma ferramenta de ampla utilidade para os trabalhadores comuns. Frequentemente, indivíduos leigos desconhecem seus direitos e, por vezes, os limites da autoridade de seus empregadores. Nesse sentido, as tecnologias de IA desempenham um papel crucial ao fornecer uma fonte confiável de informações, tornando-as mais acessíveis. Elas não apenas esclarecem dúvidas específicas sobre a legislação vigente, mas também desempenham um papel essencial na interpretação de textos jurídicos, como contratos e artigos constitucionais, apresentando-os de maneira mais compreensível para a população geral.

Somado à capacidade de facilitar a compreensão de textos jurídicos essas tecnologias podem também auxiliar o trabalhador a identificar possíveis clausulas abusivas em seus contratos, ou a omissão de certos direitos trabalhistas. Essas ferramentas podem oferecer ainda apoio simples como por exemplo avaliar se o cálculo da rescisão trabalhista está correto.

Em outros casos de maior complexidade, as tecnologias de IA podem ser um apoio ao trabalhador, permitindo que ele possua uma capacidade independente de verificar se está sendo devidamente retribuído por seu trabalho de acordo com a atividade desenvolvida, considerando por exemplo turno de trabalho, atividades de risco e tipos de hora extra.

Ao facilitar o acesso às informações sobre direitos trabalhistas, a IA contribui para nivelar o entendimento e comunicação entre empregados e empregadores. Essa democratização do conhecimento jurídico auxilia os trabalhadores, permitindo que compreendam melhor seus direitos e, consequentemente, tomem decisões informadas em relação ao seu ambiente de trabalho. A capacidade das tecnologias de IA de adotarem uma linguagem mais simples é um elemento-chave nesse processo, tornando a legislação menos intimidadora e mais compreensível para o público, que normalmente não é acostumado com os padrões textuais jurídicos que utilizam de diversos termos específicos, além de uma linguagem altamente intelectualizada.

2.6 Tecnologias de Inteligência Artificial no contexto jurídico

A aplicação de tecnologias de IA no contexto jurídico tem se expandido rapidamente no decorrer dos últimos anos, a ideia da utilização de *chatbots* com IA já tem sido abordada há anos como podemos ver no artigo "Inteligência artificial: uma realidade no Poder Judiciário"publicado em 2019 pelo TJDFT (MELO, 2019). Esse debate sobre a utilização de IA no contexto jurídico brasileiro ganhou força com a aprovação da resolução 322/2020 de agosto de 2020 que instituiu o SINAPSES, um produto desenvolvido pelo conselho nacional de justiça, como a plataforma nacional de armazenamento, treinamento supervisionado, controle de versionamento, distribuição e auditoria de modelos de inteligência artificial.

Mas a utilização de tecnologias de IA no meio jurídico não se limitam ao uso de *chatbots* e LLMs. Outras tecnologias, como análise preditiva e mineração de dados, têm encontrado aplicações valiosas nesse contexto. A IA pode, por exemplo, auxiliar na identificação de tendências legais, no monitoramento de mudanças na legislação e na previsão de desfechos de casos, contribuindo para uma tomada de decisão mais informada.

A análise preditiva, por exemplo, pode ser uma ferramenta poderosa para a estipulação do resultado de um caso judicial, considerando dados históricos assim, como informações de maior relevância. Essa previsão pode ser de extrema importância tanto para advogados quanto para clientes no que se trata da avaliação de riscos legais e elaboração de estratégias eficazes para o caso. Já a mineração de dados, pode ser uma tecnologia de auxilio ao processo judicial por meio da otimização de pesquisas, permitindo a extração de dados de um grande volume de informações. Dessa forma, a pesquisa jurídica se torna mais rápida e objetiva.

Em resumo, a aplicação de tecnologias de IA no contexto jurídico-trabalhista está revolucionando a forma como os advogados prestam serviços e como as empresas lidam com questões jurídicas. A combinação de *chatbots*, *large language models* e outras ferramentas de IA oferece a promessa de um atendimento mais eficiente, preciso e acessível, ao mesmo tempo em que permite uma análise mais profunda e estratégica no campo jurídico.

3 Estudo comparativo entre diferentes LLMs

Neste capítulo, será apresetnado um estudo comparativo entre três tecnologias populares de desenvolvimento de *chatbots*: a plataforma RASA e os modelo Llama2 e Mistral v0.1 para melhor compreender a adequação de cada uma para o projeto.

Este estudo busca caracteriza-las e analisar as dificuldades e qualidades de cada uma para melhor definir a tecnologia a ser utilizada do projeto, compreendendo os recursos materiais e humanos disponíveis.

3.1 RASA

O Rasa Open Source, atualmente, é a maior plataforma de código aberto voltada para o desenvolvimento de *chatbots* e assistentes virtuais tanto de texto como de voz. O desenvolvimento desses é feito utilizando aprendizado de máquina e processamento de linguagem natural para a compreensão das entradas do usuário. Seu objetivo é fornecer ferramentas para desenvolvedores criarem seus próprios *chatbots* personalizados para os casos de uso imaginados por eles (RASA, 2023).

A utilização dessa plataforma é dada pela especificação de "intenções"do usuário e "respostas"para essas intenções, sendo também possível definir fluxos específicos de conversa, chamados de "histórias", demonstrando relações entre diferentes intenções. Isso faz com que essa plataforma seja extremamente versátil e de fácil compreensão para o desenvolvedor, porém com projetos maiores os fluxos de conversa podem se tornar complexos e estabelecer todas as intenções do usuário juntamente com as repostas possíveis para cada uma torna-se uma tarefa trabalhosa.

```
{
  "intent": "search_restaurant",
  "entities": {
    "cuisine": "French",
    "location": "center"
  }
}
```

Figura 2 – Exemplo de código de intenção do Rasa (RASA, 2023)

Apesar disso, o Rasa é uma ferramenta poderosa e versátil que pode ser de extrema utilidade dentro do escopo correto, permitindo um controle maior das informações geradas e das interações possíveis com o usuário, mas que dentro de uma equipe com recursos e tempo limitados pode gerar pontos restritivos no projeto. Além disso, por essa plataforma

tratar apenas de desenvolvimento de *chatbots* orientados a tarefas, seu escopo e funções são limitados pelas definições manuais dos desenvolvedores, dificultando a expansão do projeto e sendo melhor adequado a fluxos mais objetivos, como de atendimento ao cliente.

```
stories:
- story: migrate from IBM Watson
  steps:
    - intent: migration
      entities:
      - product
    - slot_was_set:
      - product: Watson
    - action: utter_watson_migration

- story: migrate from Dialogflow
  steps:
    - intent: migration
      entities:
      - product
    - slot_was_set:
      - product: Dialogflow
    - action: utter_dialogflow_migration

- story: migrate from unspecified
  steps:
    - intent: migration
    - action: utter_ask_migration_product
```

Figura 3 – Exemplo de código de histórias do Rasa (RASA, 2023)

3.2 Llama 2

O Llama 2, diferentemente do Rasa não se trata de uma plataforma de *chatbot*, mas de um *Large Language Model* (LLM) que pode ser utilizado na implementação de um *chatbot* com IA, especificamente IA Generativa. Isso quer dizer que ao invés dos fluxos de conversa com o usuário serem pré-definidos pelos desenvolvedores, eles não são inicialmente estabelecidos, e o treinamento prévio exercido no modelo irá direcionar como o *chatbot* responderá às questões especificas. Ou seja, de acordo com o artigo "O que é um chatbot"da Oracle (ORACLE, 2023), o Llama 2 se trata de uma tecnologia base para o desenvolvimento de *chatbots* orientados a dados e preditivos.

Esse modelo foi desenvolvido pela empresa Meta, proprietária de produtos como Instagram, Facebook e Whatsapp, e possui variações de modelos que se baseiam na quantidade de parâmetros existentes dentro das redes neurais de cada modelo, variando de 7 bilhões a 70 bilhões de parâmetros. Como anteriormente citado, esse modelo não permite o mesmo nível de controle que uma plataforma de *chatbot* como o Rasa, em vez disso é

necessário organizar o conjunto de dados de treinamento do modelo para que ele possa melhor se comportar dentro dos casos de uso esperados, tendo em vista que o processo de treinamento desses modelos é não supervisionado e portanto a ação humana no processo de treinamento se limita à organização dos dados disponibilizados (NVIDIA, 2023). Nesse contexto os desenvolvedores acabam por lidar menos com o código do modelo em si e tratam mais no escopo de engenharia de dados.

Como informado, esse modelo trata de uma IA Generativa, o que quer dizer que ele tem capacidade de criar, ou produzir dados de forma autônoma de acordo com as informações disponibilizadas para ele e a compreensão que extraiu das exigências do usuário. Isso permite que a IA crie exemplos de dados como imagens e textos, ou que reinterprete dados já existentes permitindo por exemplo que auxilie na reescrita de textos para melhor compreensão.

3.3 Mistral

O Mistral, assim como o Llama2 se trata de um modelo de *Large Language Model* desenvolvido pela empresa Mistral AI. Por ser um LLM suas características são muito semelhantes ao Llama2, sendo uma Inteligência Artificial Generativa, treinada em quantidades grande de dados e capaz de gerar novos conteúdos de acordo com as informações do usuário além de simular uma linguagem mais natural.

O aspecto distintivo do Mistral AI reside na sua abordagem integralmente de código aberto, caracterizando-se como um projeto *Open Source*. Essa decisão estratégica proporciona uma notável transparência em sua arquitetura e confere maior independência de uso ao público em geral. Ao disponibilizar seu código fonte abertamente, o Mistral AI promove uma atmosfera de colaboração e inovação.

A documentação oficial da MistralAI destaca a competitividade de seu produto em relação a outras tecnologias de ponta, como o Llama2. Afirmando que o modelos de 7 bilhões de parâmetros do Mistral AI estão em paridade com os equivalentes, e superiores, do Llama2, incluindo uma comparação com o modelo Llama 1, que possui 34 bilhões de parâmetros. Essas comparações são fundamentadas em métricas de acurácia, abrangendo categorias diversas, como conhecimentos gerais nas áreas de ciência, tecnologia, engenharia e matemática (MMLU), conforme fornecido pela MistralAI (MISTRALAI, 2023).

3.4 Experimento

Essa seção descreve a experimentação de diversos modelos de inteligência artificial no contexto do projeto, a fim de identificar qual modelo será utilizado para a execução da solução de software.

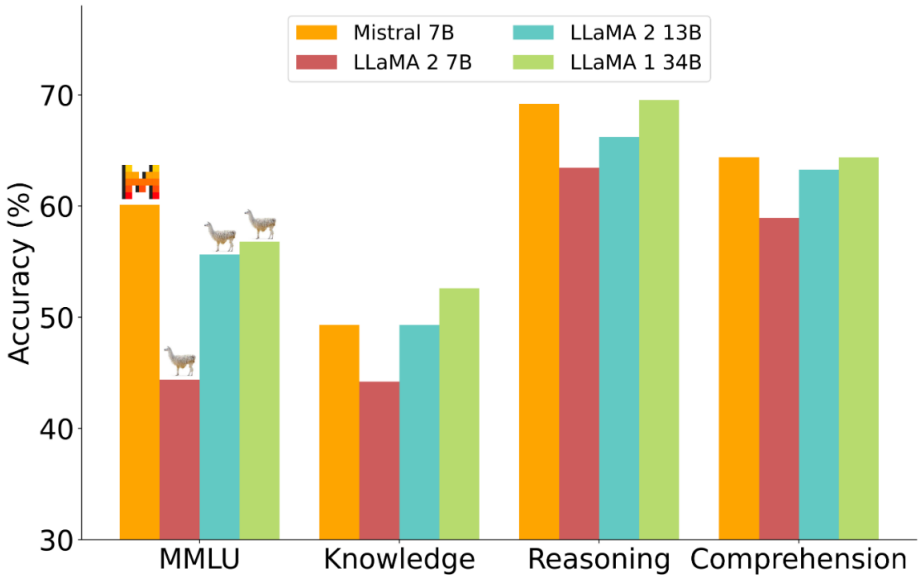

Figura 4 – Gráfico Comparativo Mistralv0.1 - métricas 1(MISTRALAI, 2023)

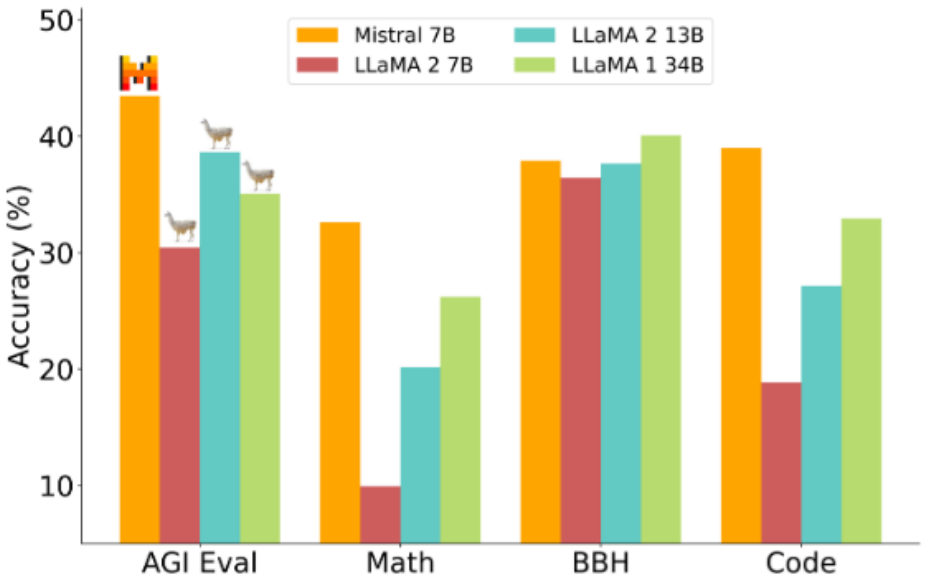

Figura 5 – Gráfico Comparativo Mistralv0.1 métricas 2(MISTRALAI, 2023)

3.4.1 Descrição

Neste experimento, foi realizada uma comparação entre a plataforma de desenvolvimento de *chatbots* RASA e os LLM LLAMA 2 e Mistral AI. O objetivo foi avaliar o desempenho, a facilidade de uso e a adequação dessas plataformas para um projeto de consultoria jurídico-trabalhista.

3.4.1.1 Parâmetros e Métricas do Experimento

Plataformas Avaliadas:

- RASA (Versão 3.0)
- LLAMA 2 7B(Versão 1.5)
- Mistral AI (Versão 0.1)

Pontos de Análise:

- Esforço manual
- Estimativa de tempo de treino em *dataset* selecionado
- Tempo e qualidade de resposta

3.4.2 Objetivos

O objetivo deste experimento é avaliar a melhor tecnologia para o projeto proposto dentro das limitações existentes de recursos, conhecimento e tempo.

3.4.3 Metodologia

3.4.3.1 Esforço Manual

A avaliação do esforço manual foi conduzida de forma experimental, considerando as características inerentes a cada uma das tecnologias em análise. Nesse processo, foram examinados aspectos como a complexidade operacional, a flexibilidade de configuração e as demandas específicas de personalização.

Considerando os tópicos supracitados foi possível averiguar que o Rasa demonstrou a necessidade de maior esforço manual envolvido em sua utilização. Enquanto o Mistral e o Llama são modelos que fazem uso de treinamento não supervisionado (NVIDIA, 2023) o Rasa trata de um *chatbot* orientado a tarefas, necessitando de uma definição mais minuciosa não apenas dos fluxos de interação e possíveis perguntas a serem feitas pelo o usuário do produto, mas também das respostas para cada fluxo, o que exigiria tempo de trabalho e conhecimento técnico jurídico que os desenvolvedores deste projeto não possuem.

Concomitantemente à conclusão de que o esforço exigido para desenvolver o projeto proposto no Rasa seria maior que do Mistral ou do Llama, também foi observado que o esforço esperado para desenvolver a proposta elaborada em qualquer uma dessas tecnologias seria similar, partindo dos mesmos princípios de elaboração de um *dataset* coerente com a proposta para treinar o modelo de maneira mais eficaz à alcançar o objetivo definido, podendo haver variação apenas no período de treino de cada um desses modelos de acordo com seu tamanho.

3.4.3.2 Estimativa de tempo de treino em *dataset* selecionado

Compreendendo que o esforço exigido para a utilização do Rasa não seria o mais adequado para a elaboração do *chatbot* jurídico proposto, desse ponto em diante serão analisados somente as tecnologias do Mistral e Llama 2. Para isso, buscando escolher o modelo mais adequado para a situação concreta do desenvolvimento desse assistente virtual, foram comparados o tempo de treino de cada um desses modelos, utilizando um mesmo *dataset* e máquina. É importante expressar que por limitações de tempo e *hardware* os testes foram feitos em um servidor em nuvem e foi considerado o a estimativa total de tempo de treino de cada modelo para 5 épocas após 20 minutos de execução do programa. O objetivo dessa análise era a verificação do tempo necessário para o desenvolvimento do modelo em cima desses modelos pré-treinados, já que esse processo muito provavelmente se repetiria diversas vezes durante o desenvolvimento.

Considerando essas informações, foi encontrado uma variação mínima de tempo de treino entre os modelos do Mistral AI e do Llama 2, com o anterior prevendo um tempo estimado de treinamento de 33 horas e o seguinte com uma expectativa de 31h de treino. Assim, compreende-se que apesar do tempo de treino menor para o modelo Llama 2 7B é possível concluir que a utilização de qualquer um desses modelos utilizaria quantidades similares de tempo e seria um fator de influência mínima na execução do projeto.

3.4.3.3 Tempo e qualidade de resposta

Para essa etapa de comparação foram encontrados versões em chat já disponíveis de cada um dos modelos e foi preparado um mesmo texto de entrada para cada um deles, de forma a visualizar o tempo total necessário para que a requisição seja atendida além da verificação de características da reposta para verificar como se adéquam à solicitação. A medição foi feita com cronômetro manual com objetivo apenas de averiguar uma faixa aproximada de tempo de resposta de cada um. Cada um dos modelos recebeu como entrada uma solicitação de elaboração de um texto de quatro parágrafos abordando os principais materiais de comércio no Brasil. A meta dessa solicitação era verificar o tempo necessário para que cada um dos modelos testados produzisse uma saída mais extensa além da coerência da resposta com o que foi solicitado, como por exemplo respondendo na mesma língua da solicitação.

Os resultados encontrados foram os seguintes, para o modelo do Llama 2 7B o tempo final medido entre a confirmação da entrada do usuário e a resposta finalizada foi de 13 segundos. Quanto à qualidade da respostam, o texto gerado não atendeu a solicitação de 4 parágrafos de texto, gerando um total de 6 parágrafos sendo 4 de corpo além de introdução e conclusão. Abordou, porém, o tema solicitado, além de fornecer título ao texto e subtítulo a cada um dos parágrafos. Falhou porém, ao gerar o texto completo em inglês mesmo para uma entrada em português.

Já para o modelo do Mistral 7B o tempo total entre entrada do usuário e finalização da resposta foi de 27 segundos, mais que dobrando o tempo exigido pelo Llama 2, porém ainda dentro de uma estimativa de tempo considerada aceitável para os fins do projeto. A resposta gerada não atendeu à solicitação de 4 parágrafos, escrevendo apenas três, e não possuindo conclusão alguma. Porém, apesar de algumas falhas gramaticais, como acentuação, o texto inteiro foi gerado em português conforme a língua da entrada do usuário.

3.4.4 Conclusão

Em conclusão, como informado ao começo dessa análise, foi de entendimento dos membros da equipe que o esforço manual relativo à implementação de um *chatbot* utilizando Rasa seria demasiada para a disponibilidade da equipe, pela sua necessidade de definições minuciosas, restando assim a comparação entre o Llama 2 e o Mistral AI.

Como já informado, ambos modelos foram julgados possuírem o mesmo esforço manual de organização de dados para treinamento, compreendendo que será necessário a elaboração de um *dataset* específico para os objetivos desse trabalho. Já, no que tange à estimativa de tempo necessário para treino, considerando ambos os modelos em versão de 7 bilhões de parâmetros, é possível concluir que o tempo necessário para qualquer um deles terá uma variação miníma que pode decorrer de diversas outras variáveis que não o modelo em si, os colocando em patamares comparáveis de exigência de recursos físicos.

Já quanto ao tempo e qualidade das respostas, quando apresentados com a mesma questão, o modelo do Llama 2 desempenhou a tarefa de maneira mais rápida, levando próximo de 50% do tempo necessário para o Mistral realizar a mesma solicitação. Porém cada um dos modelos falhou em diferentes características da resposta, falhando em compreensão da solicitação ou do contexto ao errarem a quantidade de parágrafos ou língua da resposta em si.

Tecnologia	**Esforço Manual**	**Tempo de Treino**	**Tempo de Resposta**	**Nº de paragrafos**	**Linguagem da Resposta**	**Tecnologia Selecionada**
Rasa	Altissimo	Não Avaliado	Não Avaliado	Não Avaliado	Não Avaliado	**Não**
Llama 2	Alto	Aprox. 31h	Aprox. 13s	6	Inglês	**Não**
Mistral	Alto	Apox. 33h	Aprox. 27s	3	Português	**Sim**

Tabela 2 – Comparativo de tecnologias do projeto de acordo com experimento

Por tanto, em conclusão aos experimentos realizados, foi decidido pelo uso da tecnologia do Mistral AI para desenvolvimento o projeto por uma série de fatores. Apesar de seu desempenho ter se mostrado abaixo do que foi capaz de ser atingido pelo Llama 2, levando uma quantidade considerável a mais de tempo para realização da tarefa, seu desempenho ainda esteve presente dentro de um tempo considerado, para os fins desse projeto, aceitável para uso cotidiano. Além disso demonstrou compreensão da entrada em par com o Llama 2 ao errar na quantidade de parágrafos gerados, assim como demonstrou

melhor compreensão de contexto ao fornecer o texto inteiro na língua da questão, sem necessidade de explicitar essa informação. Por fim, outro ponto diferencial que o Mistral AI trouxe para os fins do projeto, foi sua característica totalmente *Open Source*, que se mostra muito mais alinhada com os objetivos maiores do projeto, que busca se tornar uma ferramenta de apoio ao público geral, sem necessidade de compras de licenças e similares.

4 Desenvolvimento do projeto

Esse capítulo se propõe a apresentar o projeto que foi desenvolvido abordando aspectos que abrangem desde a concepção do sistema até sua implementação de fato

4.1 Introdução

O desenvolvimento do projeto, no contexto de engenharia de software, é o processo de planejamento, organização e criação de uma solução de software. Envolve a definição dos requisitos, a identificação do *público-alvo*, a definição dos objetivos, a elaboração da arquitetura, o design, a implementação e a realização de testes para garantir a qualidade do software. Todos esses processos tem como objetivo final atender a necessidade dos *clientes*. Por fim, analisaremos o produto final considerando os requisitos elicitados e o cumprimento destes.

4.2 Definição do Problema

O problema abordado nesta pesquisa centra-se na inacessibilidade dos direitos do cidadão brasileiro em linguagem simplificada para indivíduos de baixa ou nenhuma alfabetização. Esta dificuldade é agravada pela baixa probabilidade de cidadãos de baixa renda terem acesso a consultorias jurídicas devido ao alto custo financeiro associado a esses serviços. Como resultado, tanto o contratado quanto o contratante estão expostos a cláusulas inconstitucionais ou fraudulentas devido à falta de conhecimento, levando a consequências legais para ambas as partes. Além disso, o atraso ou cancelamento de procedimentos jurídicos devido à inconformidade com os requisitos durante a elaboração, bem como a dificuldade na pesquisa de jurisprudências, contribuem para a perpetuação desse problema.

Afetando diretamente contratados, advogados e contratantes, o impacto dessa inacessibilidade manifesta-se na negação ilegal de direitos, na perda de clientes pelos advogados, na elaboração de contratos prejudiciais e em disputas legais evitáveis.

Uma solução de sucesso envolveria a democratização do acesso à legislação, tornando-a mais acessível à população de baixa renda e alfabetização, possibilitando a revisão automática de textos por advogados e contratantes, e permitindo que os contratados consultem seus direitos e responsabilidades de forma mais eficaz.

4.2.1 Análise SWOT

A análise SWOT revela as forças, fraquezas, oportunidades e ameaças relacionadas à implementação de uma solução para o problema abordado (NAKAGAWA, 2023).

4.2.1.1 Forças *(Strengths)*

- A capacidade das Linguagens de Modelagem de Linguagem (LLMs) atuarem como revisores de texto eficientes.
- Acesso gratuito pela internet.
- Utilização de linguagem acessível.
- Alta disponibilidade da solução.

4.2.1.2 Fraquezas *(Weaknesses)*

- As respostas geradas pelas LLMs podem não ser 100% confiáveis.
- A solução não substitui completamente a necessidade de um advogado.

4.2.1.3 Oportunidades *(Opportunities)*

- A legislação brasileira é aberta ao público, mas ainda não possui um chatbot especializado em jurisdição trabalhista.

4.2.1.4 Ameaças *(Threats)*

- Mudanças na legislação podem afetar a eficácia da solução.
- Possível surgimento de concorrentes no mercado.

4.3 Público-Alvo

O público-alvo da solução proposta é composto por três segmentos principais: contratados, advogados e contratantes. Cada grupo enfrenta desafios específicos relacionados à inacessibilidade dos direitos legais, e a implementação da solução visa atender às necessidades distintas de cada um.

Contratados: Indivíduos de baixa renda ou baixa alfabetização que frequentemente assinam contratos sem compreensão adequada dos termos legais, ou ainda cidadãos que simplesmente não possuem conhecimento jurídico. A solução visa auxiliá-los a entender seus direitos e responsabilidades perante a Lei.

Advogados: Profissionais que enfrentam a perda de clientes devido a processos infrutíferos resultantes de pesquisa insuficiente de jurisprudências ou falta de revisão de suas peças. A solução proporcionará suporte na revisão automática de textos legais.

Contratantes: Entidades que elaboram contratos, mas podem inadvertidamente incluir cláusulas inconstitucionais. A solução ajudará a evitar litígios desnecessários, oferecendo uma revisão automatizada de contratos antes da assinatura.

4.4 Objetivos

O principal objetivo do projeto era desenvolver uma plataforma que permita o acesso do público geral à consultoria jurídico-trabalhista, tal como oferecer suporte na pesquisa e na elaboração de textos jurídicos. Dessa forma, foi necessário a elaboração de uma solução de software, por meio da utilização de LLMs com dados recentes da legislação trabalhista e jurisprudências, juntamente com o desenvolvimento de uma plataforma web para permitir o acesso ao modelo.

Com a finalidade de acompanhar o andamento do projeto, até a sua finalização após um semestre letivo, foi mensurada a quantidade de requisitos atendidos por quantidade de tempo, métrica essa que será melhor explicada nas seções subsequentes.

4.5 Requisitos

Dada as características do projeto, a definição dos requisitos se deu mediante um *brainstorm*, seguindo os conceitos de engenharia de software para o levantamento de requisitos funcionais e não-funcionais. Como resultado, foram elaboradas as tabelas 3 e 4 com a descrição dos requisitos levantados.

4.5.1 Requisitos Funcionais

Nesta seção, serão delineados os requisitos funcionais do sistema, descrevendo as principais funcionalidades que o sistema deve oferecer para atender aos objetivos do projeto. Os requisitos estão divididos entre requisitos de sistema e requisitos do usuário. O processo de levantamento de requisitos funcionais foi iniciado por meio de uma sessão de *brainstorming*, uma técnica colaborativa que estimula a geração espontânea e livre de ideias entre os membros da equipe de desenvolvimento. O objetivo principal era identificar as funcionalidades essenciais do sistema, tais como as necessidades dos seus usuários.

A análise das ideias geradas resultou na identificação dos requisitos funcionais cruciais para atender aos objetivos do projeto. Esses requisitos foram identificados e refinados para garantir clareza e compreensão. Veja a tabela 3.

#	Descrição
RF01	O usuário deve conseguir se cadastrar
RF02	O usuário deve conseguir fazer *login* e *logout*
RF03	O usuário deve conseguir excluir o seu cadastro
RF04	O usuário deve conseguir criar conversas e elas devem ser independentes
RF05	O sistema deve armazenar as conversas anteriores dos usuários
RF06	O usuário deve conseguir acessar as suas conversas anteriores
RF07	O usuário deve conseguir excluir conversas
RF08	O usuário deve conseguir escrever uma mensagem
RF09	O sistema deve conseguir compreender as mensagens do usuário
RF10	O sistema deve responder às mensagens do usuário considerando o contexto necessário
RF11	O usuário deve conseguir selecionar de qual modelo ele deseja obter a resposta
RF12	O sistema deve permitir que os usuários forneçam *feedback*, seja positivo ou negativo, para melhorias contínuas

Tabela 3 – Requisitos funcionais

4.5.2 Requisitos Não-Funcionais

Nesta seção, serão apresentados os requisitos não-funcionais, que descrevem atributos de qualidade e restrições do sistema, complementando os requisitos funcionais delineados na seção anterior. Diferentemente dos requisitos funcionais, que se concentram nas funcionalidades específicas do sistema, os requisitos não-funcionais abordam características transversais que impactam a eficiência, desempenho e experiência geral do usuário.

Assim como no caso dos requisitos funcionais, a identificação dos requisitos não-funcionais foi conduzida por meio de uma abordagem colaborativa, destacando a importância de considerar não apenas o "o que"o sistema deve fazer, mas também "como"ele deve desempenhar suas funções. Foram realizadas discussões entre os membros da equipe através de *brainstorming*, que teve como resultado os requisitos não-funcionais apresentados na tabela 4.

#	Descrição
RNF01	Deve conseguir responder em linguagem acessível, isso é, de fácil compeensão para a maior parte dos usuários
RNF02	Deve ser uma aplicação web
RNF03	Deve ser compatível com os navegadores mais populares
RNF04	Deve ser responsiva a tamanhos de telas diferentes
RNF05	Deve ser compatível com mobile
RNF06	Deve usar linguagem apropriada ao cunho jurídico
RNF07	Deve estar atualizado com as modificações legislativas da CLT até 2023

Tabela 4 – Requisitos não-funcionais

4.6 *Backlog* do Produto

O conceito de *backlog* do produto é um conceito fundamental do desenvolvimento de software ágil, e se refere à uma lista dinâmica e priorizada de tarefas de trabalho que precisam ser efetuadas no decorrer do desenvolvimento do projeto. Na elaboração de um *backlog*, é comum organizá-lo de forma a melhor compreender às áreas e relações entre as tarefas por meio do agrupamento dessas em épicos, *feature*, requisitos ou histórias de usuário, entre outros.

Dessa forma, com o objetivo de melhor organizar o projeto e melhor se adequar à metodologia de desenvolvimento ágil, foi construído o *backlog* do produto com a seguinte estrutura: Épicos, Requisitos, Importância e Priorização. A sua elaboração se deu através de reuniões estruturadas para agrupar os requisitos em diferentes épicos. Por conta da definição inicial dos requisitos já estar em maior nível de granularidade, foi tomada a decisão de não estender o *backlog* para o nível de *features* ou tarefas. Posteriormente foram definidos níveis de importância para cada um dos requisitos listados, seguindo a técnica MOSCOW (NAKAGAWA, 2023), sendo 1 a maior importância e 3 a menor.

Épico	Requisito	Importância	Priorização
Gerenciamento de Conta	RF01	2	5
	RF02	2	6
	RF03	2	10
Gerenciamento de Conversas	RF04	1	1
	RF05	2	7
	RF06	2	8
	RF07	2	9
Chat	RF08	1	2
	RF09	1	3
	RF10	1	4
Feedback	RF11	1	11
	RF12	3	12

Tabela 5 – Backlog do Produto

4.7 Produto Mínimo Viável

O desenvolvimento do sistema foi iniciado visando a implementação do *Minimum Viable Product* (MVP), uma versão inicial que visa atender aos requisitos fundamentais identificados anteriormente (ALLIANCE, 2017). O MVP foi concebido considerando a viabilidade técnica, o tempo disponível e os recursos disponíveis para garantir uma entrega inicial de alto valor agregado.

4.7.1 Escopo do MVP

O MVP deve deve incluir as seguintes funcionalidades principais, priorizadas com base na sua importância para o sucesso inicial do projeto, tal como foram definidas na tabela 5:

- **Especificação de um modelo de LLM para dados da área:** A solução inicial deve utilizar o modelo já treinado com as principais fontes de dados da área jurídico-trabalhista, isso é, a Legislação sobre a CLT, de forma que as respostas do *chatbot* tenham embasamento correto.

- **Criação de conversas:** O usuário, mesmo ainda sem cadastro, deve conseguir criar diferentes conversas com o *chatbot* e elas devem funcionar independentemente umas das outras. Além disso, as conversas ficarão salvas durante a sessão do navegador, caso o usuário feche a janela ou atualize as páginas, os dados se perderão.

- **Conversação de texto do usuário com o *chatbot*:** O usuário deve conseguir ter um diálogo somente de texto com o modelo, mandando e recebendo mensagens, de forma que ele tire suas dúvidas sobre a área jurídico-trabalhista. Ademais, o *chatbot* deve se portar como um humano, respeitando os requisitos não-funcionais levantados na tabela 4.

4.7.2 Justificativa do Escopo do MVP

A escolha dessas funcionalidades específicas para o MVP foi orientada pela necessidade de oferecer uma experiência mínima e valiosa aos usuários desde o início. Cada funcionalidade foi selecionada para fornecer uma base sólida para o desenvolvimento futuro, permitindo a validação das hipóteses do projeto e a coleta de *feedbacks*. Além disso, foi definido que o maior esforço inicial do projeto seria no treinamento do modelo de LLM com dados pertinentes da área jurídico-trabalhista, o que envolve a recuperação, tratamento e organização desses dados para incluir no algoritmo de treinamento.

4.8 Protótipos

Para visualizar e validar as principais características da aplicação proposta, foi desenvolvido um protótipo de média fidelidade. Este protótipo serve como uma representação visual e interativa do design da interface do usuário, proporcionando uma visão preliminar das funcionalidades principais do sistema.

O protótipo foi construído utilizando ferramentas de design de interfaces, como o Figma (FIGMA, 2023), que permite a criação rápida e iterativa de telas e fluxos de

interação. A metodologia adotada enfatizou a facilidade de interação, permitindo ajustes contínuos com base no *feedback* dos *stakeholders* e nas mudanças dos requisitos.

4.8.1 Componentes e Funcionalidades Representados no Protótipo

O protótipo de média fidelidade incorpora elementos visuais e funcionais essenciais para a compreensão da experiência do usuário. Entre os componentes principais representados estão:

4.8.1.1 Página de Chat

A interface de chat, conforme apresentada nas figuras 6, 7, 8, 9 representa a principal funcionalidade da aplicação, que permite que o usuário envie mensagens para o modelo. Ela é composta pela pela área das mensagens e por um menu lateral à direita com informações sobre o projeto que pode ser estendido ou ocultado, sendo a simplicidade a característica visual de destaque da página. A seção de mensagens é organizada verticalmente, apresentando diferentes elementos dependendo do contexto. Em uma nova conversa, são exibidos botões contendo assuntos comuns para iniciar a interação e abaixo uma caixa de texto que permite ao usuário formular novas questões. Em conversas em andamento, a mesma área exibe as mensagens enviadas e também apresenta uma caixa de texto para o usuário elaborar novas perguntas.

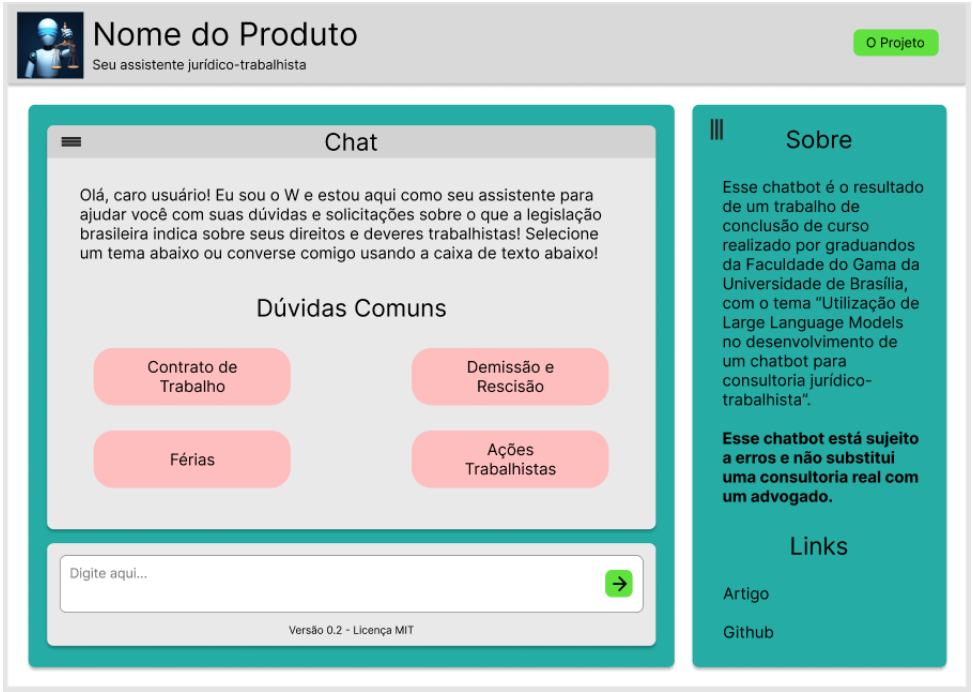

Figura 6 – Página de chat com menu sobre

Figura 7 – Página de chat sem menu sobre

Figura 8 – Página de chat inicializada

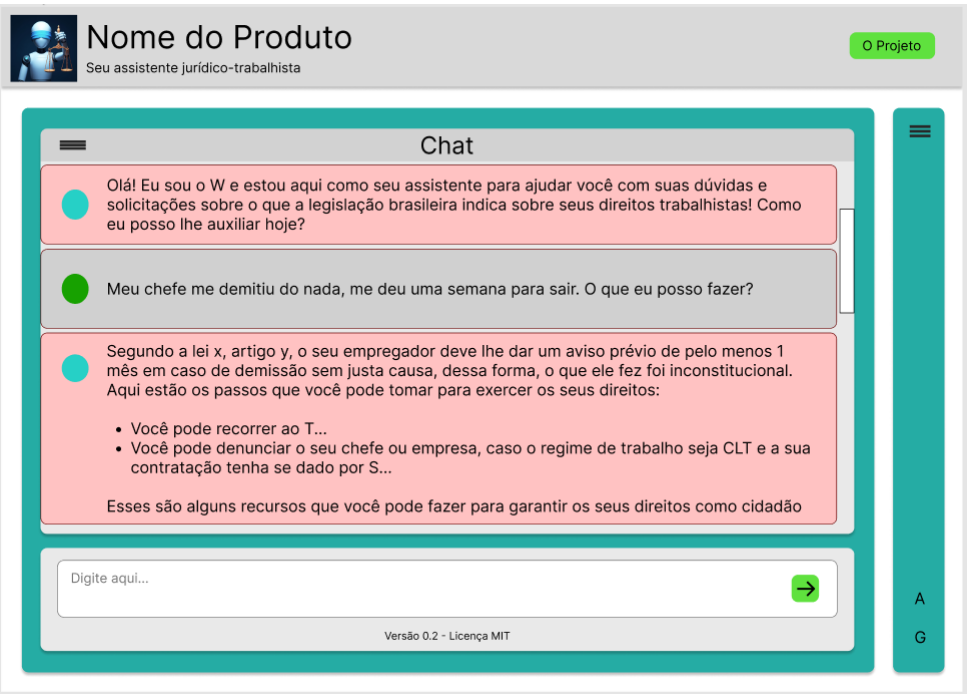

Figura 9 – Página de chat com conversa em andamento

Ainda na página de chat, temos outras ações a serem realizadas mediante botões, as quais são gerenciamento de conversas e navegação para a página de informações sobre o projeto:

- **Abrir o menu de conversas:** Ao clicar no ícone de menu superior esquerdo, um menu em forma *overlay* é apresentado, o qual lista todas as conversas do usuário logado, com a opção de acessá-las ao clicar e também de criar novas conversas ao clicar no botão correspondente. Por último, também existe a opção de apagar todas as conversas, além do botão de fechar o menu, que também pode ser fechado ao clicar em qualquer região externa a ele. Ver figura 10.

4.8.1.2 Página de Informações sobre o Projeto

A página de informações do projeto, apresentada na figura 11, desempenha um papel crucial na apresentação e contextualização do sistema proposto. Essa seção tem como objetivo fornecer uma visão abrangente do projeto, seus objetivos, funcionalidades-chave e links pertinentes.

A página de informações do projeto foi estruturada de forma bem simples a fim de facilitar a assimilação de informações pelos usuários. Seções de texto distintas foram criadas para abordar temas como:

Figura 10 – Página de chat com menu de conversas

Figura 11 – Página de informações sobre o projeto

- **Descrição do Projeto:** Explicação da origem e definição do *chatbot* em questão.
- **Objetivos do Projeto:** Descrição clara e concisa dos objetivos principais do sistema.

- **Funcionalidades-Chave:** Destaque para as principais funcionalidades que distinguem o projeto.

- **Aviso sobre utilização:** Um aviso acerca da insubstituibilidade de advogados e profissionais da área de direito para tratar sobre questões legais.

- **Links:** Lista de links pertinentes do projeto, tais como para o endereço do código fonte e do artigo.

4.9 Metodologia de Desenvolvimento

A abordagem metodológica para o desenvolvimento do projeto foi fundamentada em uma combinação estratégica de práticas ágeis, *Lean Start Up* e Scrum buscando fazer continuamente entregas com valor agregado, como estabelecido pelo manifesto ágil. Também baseado no manifesto ágil, foi valorizado a flexibilidade do projeto, compreendendo que com os recursos e tempo limitados, podem ocorrer mudanças no processo de desenvolvimento (BEEDLE, 2001).

Como citado, também foram utilizados conceitos da metodologia *Lean Start Up*, de forma mais específica foi utilizado o conceito de MVP, ou o mínimo produto viável, que se refere à uma versão simplificada do produto, contendo apenas as funcionalidades mais fundamentais para a resolução do problema abordado (RIES, 2011).

Por fim, para a organização diária do projeto foi utilizada uma adaptação da metodologia Scrum para a implementação do método ágil. A adaptação foi feita para melhor se adequar às rotinas e horários dos estudantes, abrindo mão das reuniões diárias, as *dailies*, mas ainda mantendo as *sprint reviews* e *sprint plannings*. As *sprint reviews* são reuniões que ocorreram ao final de cada *sprint*, no caso do projeto em questão, ao final de cada semana, onde as entregas feitas foram analisadas pela equipe e definir possíveis mudanças que possam ocorrer. Já as *sprint plannings* são reuniões feitas no inicio de todas as *sprints* com objetivo de definir as tarefas a serem executadas, considerando possíveis débitos técnicos das *sprints* anteriores (SCHWABER, 2023).

4.9.1 RoadMap

Para melhor organização do desenvolvimento do projeto, o grupo elaborou um *RoadMap* baseado nos conceitos anteriormente abordados sobre metodologia ágil e Scrum, assim como considerando o *Backlog* do produto e a consequente priorização de requisitos. O conceito de *RoadMap* se refere à uma representação visual do caminho, plano estratégico e prazos necessários para atingir um objetivo, é uma forma intuitiva e simples de se estabelecer um cronograma de projeto.

Dessa forma, foram estipuladas 19 sprints totais para a elaboração do projeto, iniciando em fevereiro de 2024 e finalizando em junho do mesmo ano. O objetivo deste *RoadMap* é permitir aos alunos o acompanhamento do andar do projeto visando compreender as tarefas ainda necessárias a serem desenvolvidas. O tempo estimado para a execução de cada conjunto de requisitos alocado por *sprint* foi alcançado pelos estudantes por meio de um processo de *brainstorming* em conjunto e a definição das tarefas a serem executadas a cada semana foi baseada no *backlog* apresentado, considerando a priorização de requisitos e possíveis dependências de requisitos.

Sprints	**Data**	**Atividades**
1	05/02 - 11/02	Inicialização do Projeto
2	12/02 - 18/02	
3	19/02 - 25/02	RF04, RF08
4	26/02 - 03/03	RF09, RF10
5	04/03 - 10/03	
6	11/03 - 17/03	Teste com usuários
7	18/03 - 24/03	Treinamento LLM
8	25/03 - 31/03	RF01, RF02
9	01/04 - 07/04	RF05, RF06
10	08/04 - 14/04	RF11, RF12
11	15/04 - 21/04	Teste com usuários
12	22/04 - 28/04	Refinamento do treinamento
13	29/04 - 05/05	RF13, RF14
14	06/05 - 12/05	RF07, RF03
15	13/05 - 19/05	RF15
16	20/05 - 26/05	Teste Final
17	27/05 - 02/06	
18	03/06 - 09/06	Escrita e Correções
19	10/06 - 16/06	

Tabela 6 – Roadmap do Produto

4.10 Arquitetura de Software

Para a elaboração do projeto proposto neste documento, foram selecionadas uma série de tecnologias para desenvolvimento e auxilio desse processo, com o objetivo de melhor alcançar o produto final desejado.

Para o desenvolvimento da plataforma web foi utilizado um *backend* desenvolvido em Python utilizando o *framework* Django em comunicação com um banco de dados PostgreSQL para o manuseio e controle dos dados de usuários e históricos de conversas. Esse está em comunicação não apenas com os LLMs escolhidos por meio de uma conexão HTTP com a plataforma AWS (onde foram hospedados os modelos Mistral AI v3 e Llama 3) mas também com um *frontend* desenvolvido em Javascript utilizando a biblioteca

ReactJS.

Já para auxiliar no processo de desenvolvimento do projeto, foi utilizada a plataforma GitHub para o versionamento do código, organização de tarefas a serem executadas em cada *sprint*, atribuição das tarefas a cada membro da equipe, e organização geral do código do projeto.

Muitas dessas tecnologias específicas foram selecionadas por atenderem as necessidades do produto proposto, além de já serem tecnologias familiares aos alunos envolvidos nesse trabalho. Dessa forma, não foi necessário um período extenso de estudo das tecnologias por parte dos estudantes, permitindo um foco maior no desenvolvimento e no treinamento do LLM.

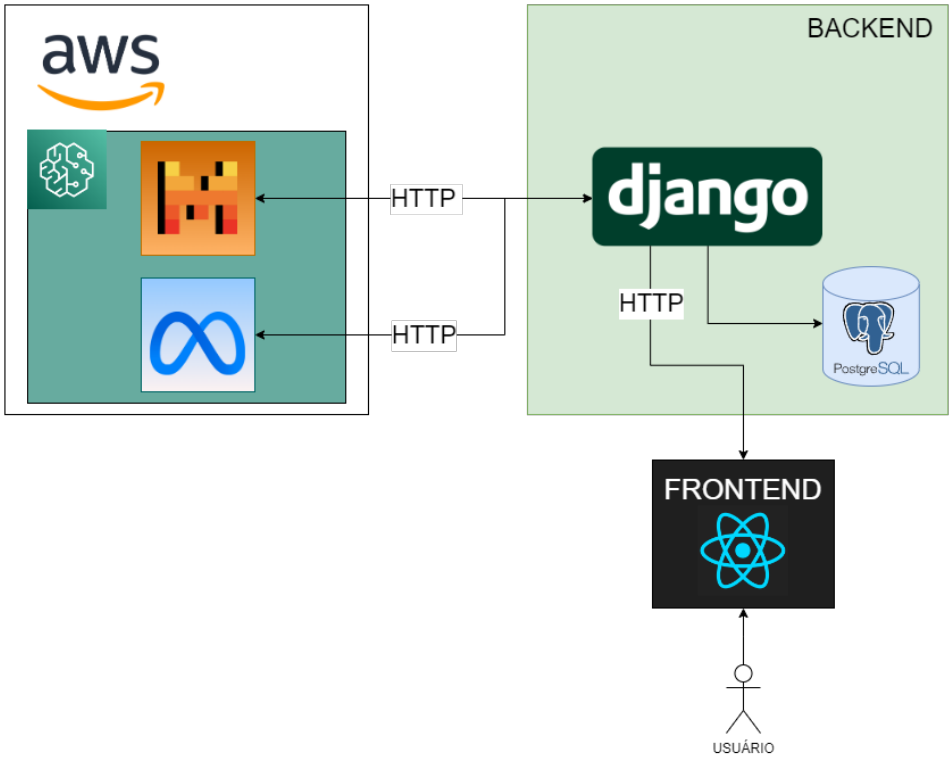

Figura 12 – Diagrama de Arquitetura

4.11 Gestão de Riscos

A gestão de riscos é uma parte fundamental do desenvolvimento do projeto, especialmente ao lidar com tecnologias inovadoras, como um *chatbot*. Nesta seção, serão identificados possíveis riscos, apresentado um plano de ações preventivas, bem como estratégias para mitigar os impactos adversos.

4.11.1 Identificação e Plano de Ação dos Riscos

A identificação de riscos é um processo crucial para antecipar possíveis desafios. Os riscos potenciais associados ao desenvolvimento e implementação do *chatbot* incluem:

- **Risco Tecnológico:** Possíveis falhas no treinamento do modelo de LLM, tal como incompatibilidades com as tecnologias utilizadas. A ação preventiva para isso será o monitoramento contínuo dos resultados dos treinamentos, além de testes experimentais contínuos da integração das tecnologias.
- **Risco de Desempenho:** Possível baixo tempo de resposta e desafios na escalabilidade do *chatbot* para lidar com um aumento inesperado no número de usuários simultâneos. A ação preventiva será a implementação de estratégias de escalabilidade e otimização de código mediante revisões periódicas durante o desenvolvimento.
- **Risco de Segurança:** Vulnerabilidades na proteção de dados e privacidade dos usuários durante as interações com o *chatbot*. A ação preventiva será realizar auditorias regulares de segurança e conformidade com padrões de proteção de dados.
- **Risco de Aceitação do Usuário:** Resistência ou desinteresse dos usuários em adotar o *chatbot* como canal de comunicação. A ação preventiva envolverá estratégias de marketing para promover os benefícios e a utilidade do *chatbot*.

Este plano de gestão de riscos será revisado regularmente ao longo do ciclo de vida do projeto. A cada marco significativo ou evento inesperado, será realizada uma análise de riscos para garantir a eficácia contínua do plano e a adaptação a novas circunstâncias.

Essa abordagem sistemática na gestão de riscos proporcionará uma estrutura robusta para lidar com desafios potenciais, permitindo que o projeto se adapte eficientemente às mudanças e mantenha sua integridade ao longo do tempo.

4.12 Desenvolvimento

Como citado, o projeto se baseou na metodologia Scrum para sua organização de forma que foram organizadas *sprints* de 1 semana com o foco em desenvolvimento e entrega contínua do projeto. Abaixo será apresentado de forma mais concisa os objetivos e ocorrências das *sprints*.

4.12.1 *Sprints* 01 e 02

Durante os *sprints* 01 e 02, o foco foi na configuração inicial do projeto. As atividades incluíram a configuração do ambiente de desenvolvimento local e do repositório, a instalação das ferramentas necessárias e a construção da infraestrutura do código.

4.12.2 *Sprints* 03 e 04

As *sprints* 03 e 04 foram dedicadas ao refinamento da identidade visual do projeto, buscando incrementar nos protótipos elaborados durante a idealização do sistema, assim como as primeiras funcionalidades de gerenciamento de conta e estudos de formas e ferramentas de treinamento do modelo.

4.12.3 *Sprints* 05 a 10

No período das *sprints* 05 a 10 o foco foi voltado para os requisitos de gerenciamento de conversas, maneiras de permitir o feedback do usuário e os primeiros testes locais com o treinamento e carregamento do modelo. Nessas *sprints* foi quando a equipe se deparou com as limitações computacionais para rodar o modelo localmente e buscou soluções em plataformas em nuvem para continuar o desenvolvimento dentro da modelagem idealizada.

4.12.4 *Sprints* 11 a 13

As *sprints* 11 a 13 foram focadas no fluxo de chat co o bot, renovação visual do sistema e a responsividade do mesmo. Essas *sprints* continuaram com o desenvolvimento de formas de permitir o treinamento do bot com dados especializados, porém sem muito avanço. No final da *sprint* 13 foi tomada a decisão de modificar a arquitetura do projeto, agora não mais fazendo o treinamento do modelo, mas utilizando a especificação por *prompts*, e assim permitindo o uso da plataforma Amazon Sage Maker assim como a abordagem multi-modelo.

4.12.5 *Sprints* 14 a 17

Durante as *sprints* 14 a 17 o objetivo foi na utilização da plataforma Amazon Sage Maker, o tratamento dos dados recebidos dos *endpoints* de cada modelo, a implementação da memória de contexto nas conversas e na interação multi-modelo assim como os testes internos do produto.

4.12.6 *Sprints* 18 e 19

por fim, nas *sprints* 18 e 19 a equipe se voltou para a finalização da escrita do trabalho e o refinamento do código e documentação, corrigindo possíveis *bugs* e melhor organizando o tratamento de dados.

4.13 Avaliação da funcionalidade

Nesta seção, apresentamos os resultados da avaliação da funcionalidade do sistema desenvolvido, estruturada em três subseções: Requisitos Funcionais, Requisitos Não Funcionais e Melhoria da Interface Visual.

Nas subseção de Requisitos Funcionais e não Funcionais, analisaremos cada requisito definido, justificando seu cumprimento ou não com as devidas evidências. Por fim, na subseção de Interface Visual, discutimos as alterações realizadas em relação ao proposto inicialmente, justificando cada mudança e seus benefícios para a usabilidade do sistema.

4.13.1 Requisitos Funcionais

Nessa seção serão abordados os requisitos funcionais estabelecidos no planejamento do projeto, assim como o nível de cumprimento do requisito separado em 3 (três) categorias, Cumprido Totalmente, Cumprido Parcialmente, e Não Cumprido. Também será incluído uma justificativa para o nível de cumprimento definido.

- RF01 - O Usuário deve conseguir se Cadastrar

  O requisito funcional 01 foi cumprido completamente pela equipe de forma que o produto permita o cadastro do usuário utilizando os campos de usuário, e-mail e senha.

- RF02 - O usuário deve conseguir fazer login e logout

  O requisito funcional 02 também foi completamente cumprido, permitindo que um usuário já cadastrado no sistema consiga acessar sua conta por meio do e-mail ou nome de usuário juntamente com sua senha, assim como fazer logout de sua conta.

- RF03 - O usuário deve conseguir excluir o seu cadastro

  O requisito funcional 03 foi completamente cumprido pela equipe, permitindo que um usuário logado possa solicitar a remoção da sua conta, assim apagando seus dados do banco de dados e não mais o permitindo logar-se.

- RF04 - O usuário deve conseguir criar conversas e elas devem ser independentes

  O requisito funcional 04 foi completamente cumprido pela equipe, permitindo que um usuário possua mais de uma conversa e havendo total independência entre as conversas registradas.

- RF05 - O sistema deve armazenar as conversas anteriores dos usuário

  O requisito funcional 05 foi completamente cumprido pela equipe, registrando o histórico completo das mensagens de uma conversa iniciada pelo usuário.

- RF06 - O usuário deve conseguir acessar as suas conversas anteriores

  O requisito funcional 06 foi cumprido completamente pela equipe, permitindo que um usuário cadastrado acesse suas conversas anteriores e visualize todas as mensagens.

- RF07 - O usuário deve conseguir excluir conversas

  O requisito funcional 07 foi completamente cumprido pela equipe no desenvolvimento do projeto, permitindo que um usuário visualize a lista de conversas iniciadas por ele e possa excluir as conversas desejadas.

- RF08 - O usuário deve conseguir escrever uma mensagem

  O requisito funcional 08 foi completamente cumprido, permitindo ao usuário escrever e visualizar a escrita da sua mensagem dentro de uma conversa.

- RF09 - O sistema deve conseguir compreender as mensagens do usuário

  O requisito funcional 09 foi completamente cumprido, de forma que a mensagem do usuário é encaminhada ao modelo hospedado na nuvem e seu treinamento garante uma compreensão da mensagem escrita.

- RF10 - O sistema deve responder às mensagens do usuário considerando o contexto necessário

  O requisito funcional 10 foi cumprido parcialmente. A mensagem escrita pelo usuário é enviada ao modelo hospedado na nuvem acompanhada de uma instrução de comportamento, porém a mudança entre modelos dentro de uma mesma conversa pode gerar erros de comunicação fazendo com que o *bot* encontre dificuldade de compreensão completa do contexto podendo gerar resposta repetitivas ou de pouca relevância.

- RF11 - O usuário deve conseguir selecionar de qual modelo ele deseja obter a resposta

  Esse requisito foi completamente cumprido, permitindo que qualquer mensagem possa ser facilmente encaminhada a qualquer um dos modelos listados de forma que é possível uma incrementação do número de modelos sem grandes dificuldades.

- RF12 - O sistema deve permitir que os usuários forneçam *feedback*, seja positivo ou negativo, para melhorias contínuas

  O requisito funcional 12 foi completamente cumprido, de forma que o usuário tem uma página com um formulário para registrar seu *feedback*, que pode ser visualizada na própria aplicação por usuários com permissão suficiente.

5 O Sistema

5.1 Introdução

Nesse capítulo, será apresentado o produto final e suas características e alterações de projeto, assim como as intercorrências do Projeto, onde é apresentado e explicado cada tópico que não ocorreu como o planejado, e como foi realizado; e por fim, a seção Cronograma, discorre sobre as alterações no cronograma de desenvolvimento enfrentadas ao longo do trabalho.

5.1.1 Melhoria da Interface Visual

Durante a análise do protótipo de alta fidelidade e desenvolvimento da aplicação, foram percebidas possibilidades de aprimoramento da aplicação, tanto visuais quanto de usabilidades, os quais estão dispostos abaixo:

Em relação à identidade visual, optamos por cores mais claras, degradês e menos divisões, a fim de simplificar a visualização do usuário e não poluir visualmente a tela com linhas desnecessárias, tal como transmitir modernidade.

Ademais, um nome foi dado ao projeto: AI.dvogado, como uma junção da sigla AI (*Artificial Inteligence*) com a profissão "advogado". Foi concordado entre a equipe que um nome chamativo e quase cômico pode alavancar a adoção pela plataforma.

- **Página Home**

  Foi criada uma página de apresentação do projeto, com o objetivo de atrair o usuário. Foi organizada em painéis horizontais e com rolagem de tela, cada painel sendo um tópico relevante sobre o projeto: O que é, Principais funcionalidades e importância contextual.

Figura 13 – Página Home - topo

Figura 14 – Página Home - meio

Figura 15 – Página Home - baixo

- **Página Sobre**

Já a antiga página home, foi substituída por um breve informativo sobre a natureza acadêmica do projeto e para reforçar de que esse produto é somente informativo e não substitui uma consultoria com um profissional do Direto. Os links para o artigo e para o repositório continuam sendo botões nesta tela.

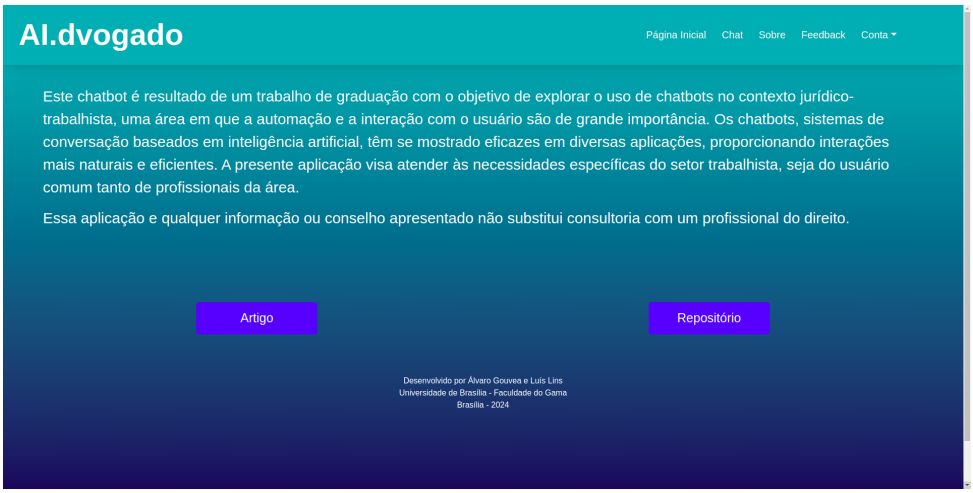

Figura 16 – Página Sobre

- **Página Feedback**

  Além da reforma visual e da criação da tela, também foi adicionado um texto para incentivar o usuário à compartilhar a sua opinião sobre o projeto, composta por um pequeno formulário e o botão de submissão.

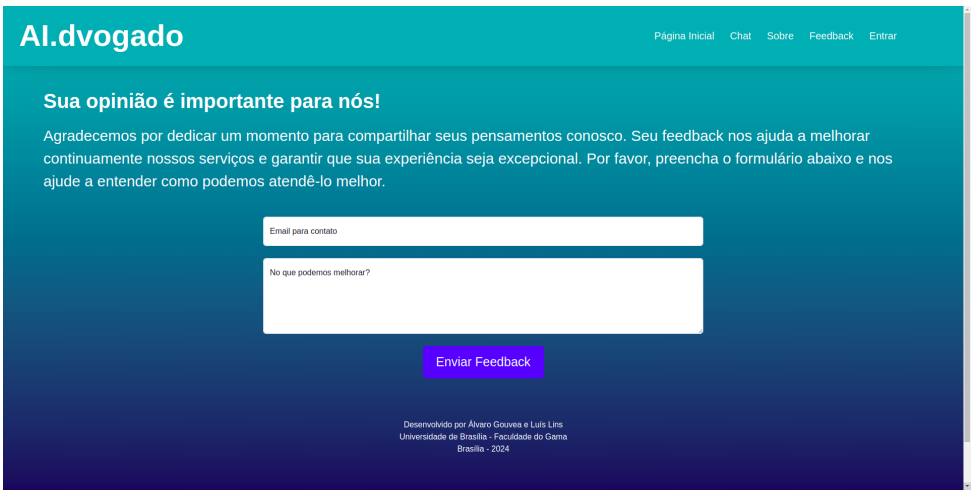

Figura 17 – Página Feedback

- **Páginas de Acesso**

  Conforme os requisitos funcionais, o projeto possui telas de acesso, a saber, de entrar e de criar conta. Elas são compostas por formulários com os campos necessários

para autenticação, respeitando a renovação da identidade visual e preservando a simplicidade. As páginas de Entrar e de Criar conta são navegáveis entre si mediante os botões cinzas, sendo os botões azuis os de ação.

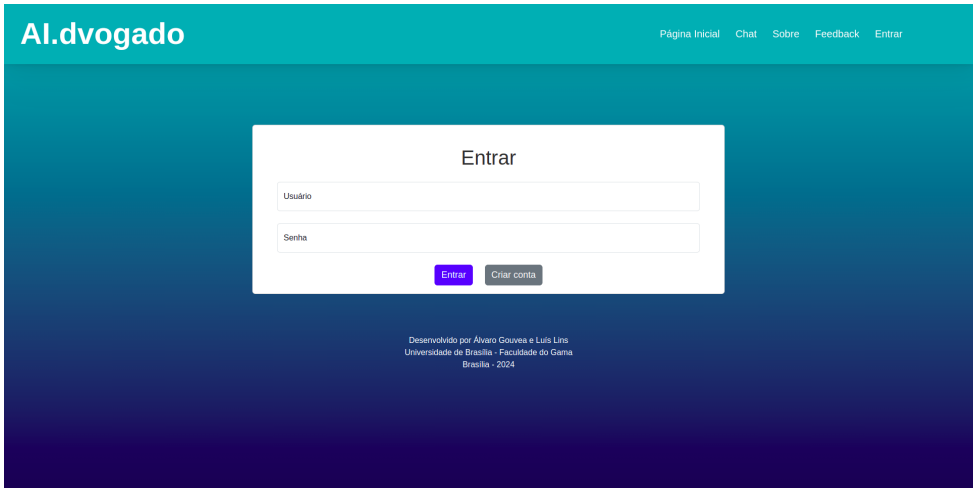

Figura 18 – Página de Entrar

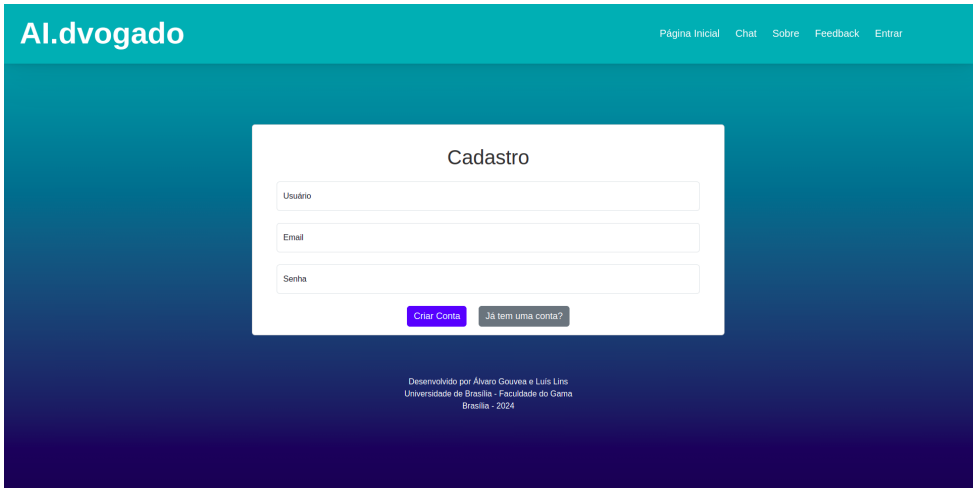

Figura 19 – Página de Criar Conta

- **Página de Chat**

  Além da reforma visual, foi adicionado o campo para seleção do modelo que irá responder a próxima mensagem do usuário, no topo direto ao lado do nome do chat, tal como a funcionalidade do usuário não autenticado usar o chat de forma limitada, isso é, sem possibilidade de salvar as conversas. Por outro lado, o usuário autenticado continua tendo acesso à lista de conversas salvas, com possibilidade de criar e excluir conversas. A seção lateral direita destaca a informação de que esse produto não substitui um advogado e pode ser escondida, caso deseje.

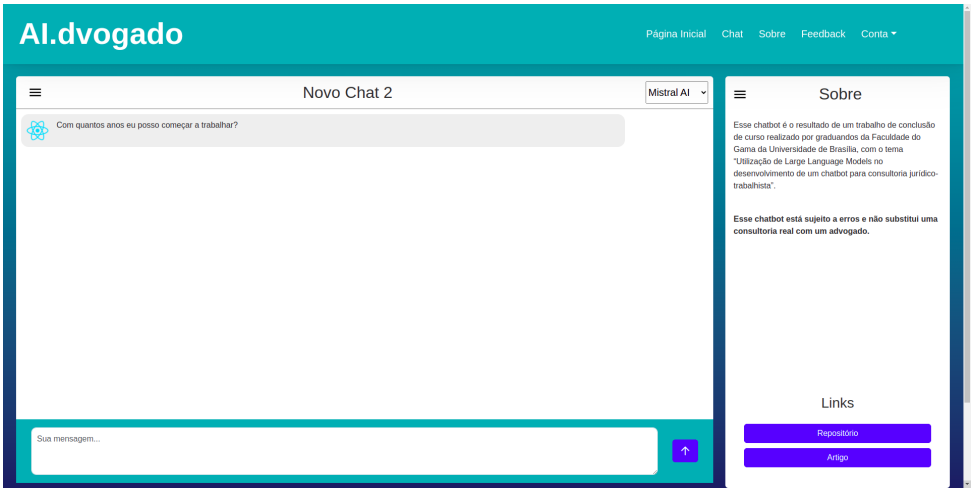

Figura 20 – Página de Chat 1

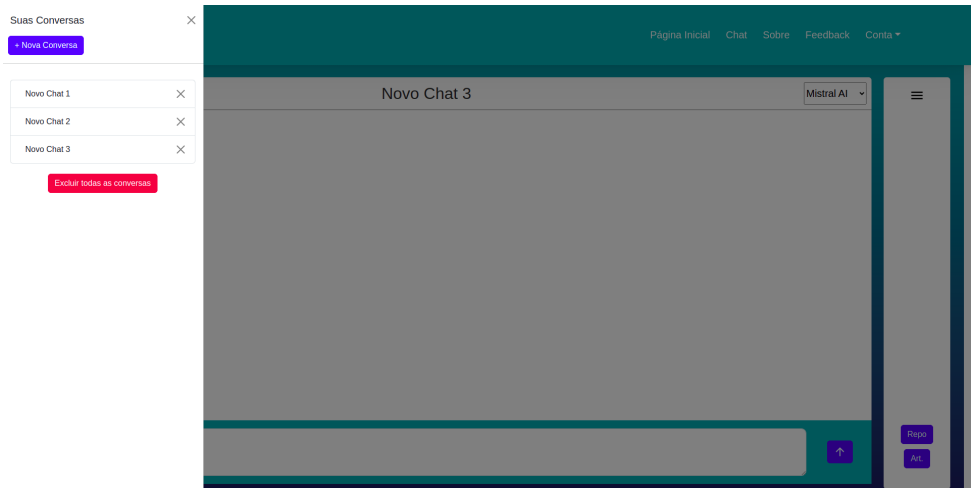

Figura 21 – Menu da Página de Chat

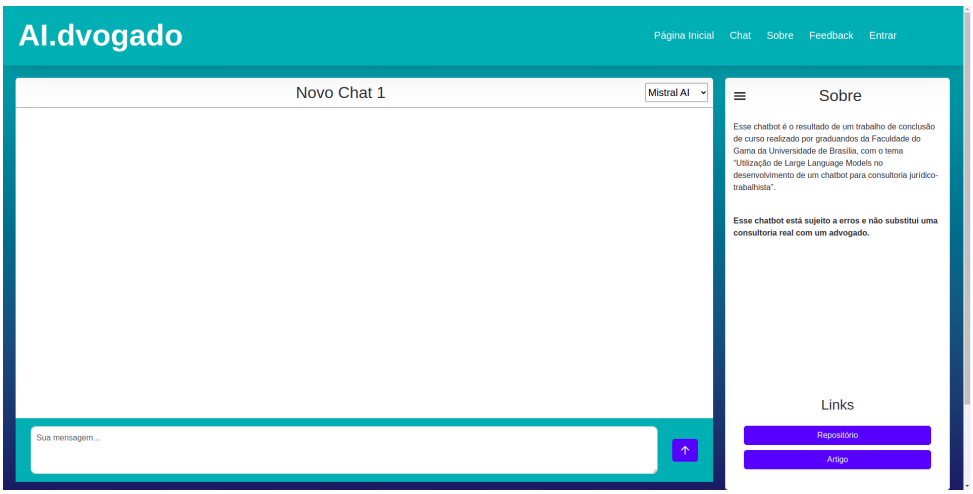

Figura 22 – Página de Chat do usuário não autenticado

- **Página de Informações da Conta**

  Por fim, o usuário logado tem acesso à tela de informações da sua conta, onde pode ver o nome de usuário e email cadastrados, tal como mudar a sua senha. Os botões de ação incluem o de salvar alterações e o de excluir a conta.

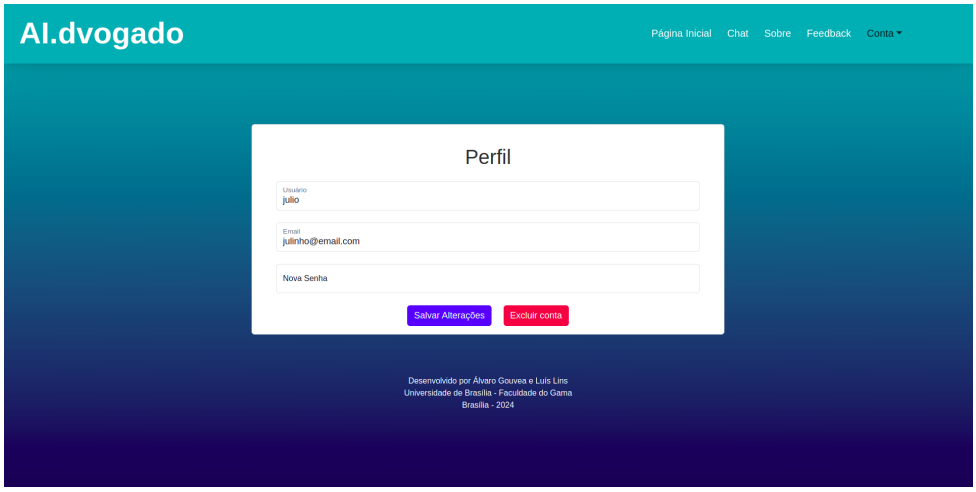

Figura 23 – Página de Informações da Conta

5.2 Intercorrências do Projeto

Durante o desenvolvimento do projeto, foram enfrentados diversos desafios e imprevistos que impactaram a execução das atividades planejadas. As intercorrências foram divididas em dois principais subtópicos: Integração dos Modelos, Disponibilização da plataforma e Arquitetura da Solução, e Cronograma. Esses desafios exigiram da equipe adaptações e replanejamentos que foram cruciais para o andamento e a conclusão do projeto. A análise dessas intercorrências proporciona um entendimento detalhado dos obstáculos enfrentados e das soluções implementadas para superá-los.

No subtópico de Integração e Arquitetura, foram discutidas as dificuldades técnicas encontradas na implementação da comunicação entre os diferentes componentes do sistema, bem como a necessidade de adaptação da abordagem inicial devido a limitações de recursos computacionais e operacionais assim como novas propostas apresentadas no período. Já no subtópico de Cronograma, são apresentadas as dificuldades em manter o cronograma original devido a imprevistos técnicos e organizacionais, e como essas questões foram abordadas para garantir a entrega dos resultados esperados dentro dos prazos ajustados.

5.2.1 Integração dos Modelos e Arquitetura da Solução

A construção do projeto havia sido planejada utilizando um *backend* em Python, mais especificamente com o *framework* Django, que se comunicaria tanto com o *frontend* em ReactJS quanto com o modelo treinado especializado do Mistral AI. No entanto, ao longo do desenvolvimento, encontramos adversidades que dificultaram a execução dessa arquitetura planejada. A proposta de treinamento do modelo para especialização mostrou-se mais complexa do que o esperado devido a limitações de recursos computacionais. Inicialmente, a equipe acreditava que o treinamento poderia ser realizado localmente na máquina de um dos integrantes. Contudo, logo no início do desenvolvimento, foi constatada a inviabilidade dessa abordagem, além das dificuldades em realizar o treinamento do modelo em plataformas de computação em nuvem, tanto por dificuldades de encontrar ferramentas com recursos computacionais suficientes quanto pelas questões da própria execução do treinamento e custos da plataforma.

Após discussões com o professor, surgiu a proposta de readaptar o projeto, não mais focando no treinamento especializado do modelo, mas utilizando instruções de *prompt* para orientar o comportamento do modelo conforme as expectativas do projeto. Dessa forma, alteramos a arquitetura da comunicação do *backend* com o modelo, de modo que ele não mais se comunicasse diretamente com um modelo treinado especificamente, mas sim com um modelo hospedado na plataforma Amazon AWS SageMaker. Essa mudança permitiu que o sistema mantivesse a funcionalidade desejada, contornando as limitações de recursos computacionais e simplificando a integração dos componentes do sistema, nos permitindo ainda uma maior flexibilidade de uso de diferentes modelos com menor carga de esforço envolvida. Além disso também foi proposto pelo professor o desenvolvimento de um sistema multi-modelo, permitindo o usuário comunicar-se com mais de um *LLM* de forma que a arquitetura final do projeto executado é a representada na figura abaixo.

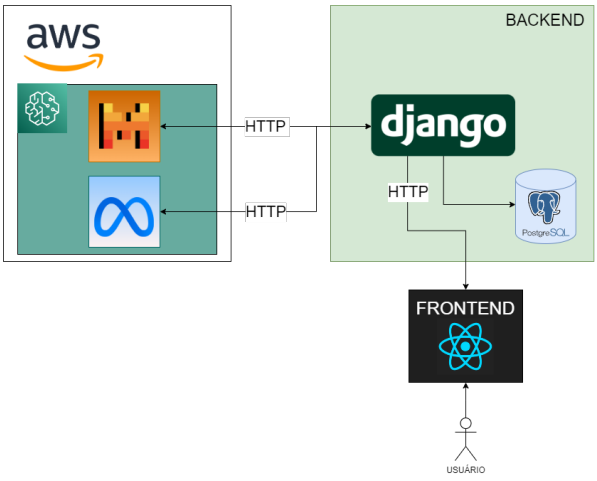

Figura 24 – Diagrama de Arquitetura

5.2.2 Disponibilização da Plataforma

Como já foi abordado anteriormente, a utilização de tecnologias de grande proporção como LLMs gerou para a equipe diversas dificuldades por conta de limitações de recursos computacionais e financeiros de forma que levou à alteração da arquitetura final da solução. Outra barreira criada pela limitação desses recursos foi a da possibilidade de disponibilização da plataforma para o público geral. Foram encontradas plataformas que permitissem a hospedagem do *frontend* e do próprio *backend* da solução, porém manter os modelos disponíveis se demonstrou extremamente custoso financeiramente, dessa forma inviabilizando que a equipe pudesse manter a plataforma amplamente disponível.

5.2.3 Cronograma

O cronograma proposto não pôde ser seguido a risca pelo membros da equipe, por questões pessoais diante da greve da Universidade de Brasília no primeiro semestre de 2024, de família e outros. Além disso, por imprecisão em estimar tarefas que nunca haviam sido realizadas por nenhum dos membros, tal como treinar e fazer *deploy* de LLMs, e a relação de tempo entre o trabalho e o curso.

Dessa forma, o cronograma foi atrasado aproximadamente 1 mês, sendo este especialmente importante para a elaboração do trabalho escrito, conforme explicitado na seção 4.12.

5.3 Conclusão

Neste capítulo, foi realizada uma análise detalhada dos resultados obtidos ao longo do desenvolvimento do projeto. A avaliação da funcionalidade demonstrou que a maioria dos requisitos funcionais e não funcionais foram completamente cumpridos, garantindo um sistema robusto e alinhado aos objetivos iniciais. Apenas o requisito relacionado à compreensão contextual das mensagens pelo sistema apresentou cumprimento parcial, evidenciando a necessidade de melhorias futuras para otimizar a interação com o usuário.

As melhorias na interface visual contribuíram significativamente para a usabilidade e a atratividade do sistema, simplificando a experiência do usuário e reforçando a identidade do projeto. A adição de funcionalidades como a possibilidade de *feedback* e a escolha do modelo de resposta também enriqueceram a aplicação.

As intercorrências enfrentadas durante o projeto proporcionaram aprendizados valiosos, destacando a importância de uma abordagem flexível e adaptável frente aos desafios. As dificuldades na integração dos modelos e na manutenção do cronograma foram superadas com soluções e ajustes estratégicos, garantindo a entrega de um produto de qualidade.

Em suma, os resultados alcançados refletem o comprometimento em superar obstáculos e entregar um sistema funcional e inovador. As lições aprendidas ao longo do processo servirão como base para futuras melhorias e evoluções do projeto, com foco na satisfação e na experiência do usuário.

6 Conclusão

A decisão de realizar o trabalho em dupla mostrou-se vantajosa, proporcionando apoio mútuo diante de imprevistos. Quando um membro enfrentava problemas ou compromissos externos que afetavam o andamento do projeto, o outro assumia mais responsabilidades, assegurando o progresso contínuo. Essa colaboração flexível não apenas superou obstáculos, mas também potencializou as habilidades individuais, enriquecendo o trabalho e garantindo a participação integral de ambos os membros em todas as etapas. Durante o desenvolvimento prático, a separação do projeto em duplas novamente se revelou benéfica, permitindo que cada membro focasse nos aspectos do trabalho em que tinha mais afinidade, embora ainda houvesse compartilhamento de tarefas e aprendizado entre ambos os estudantes.

Ao final da primeira etapa do trabalho, consideramos positivamente a viabilidade do projeto, levando em conta nossa habilidade de desenvolvimento, sua estrutura cronológica e os requisitos definidos. Contudo, antecipamos que o principal desafio residiria no processo ETL (extração, transformação e carga) dos dados provenientes de diversas fontes. Além disso, a variação na jurisprudência, especialmente pelo Tribunal Superior do Trabalho, conforme apontado por uma profissional experiente na área, emergiu como um desafio a ser enfrentado nos próximos passos do desenvolvimento do projeto.

Ao longo do desenvolvimento, enfrentamos desafios significativos relacionados ao uso de tecnologias externas para o treinamento e configuração do sistema. A complexidade dessas ferramentas exigiu um esforço adicional para entender suas funcionalidades e integrar suas capacidades com os requisitos específicos do nosso projeto. Este processo envolveu uma curva de aprendizado acentuada e demandou tempo e recursos para garantir que o sistema operasse de maneira coerente com os requisitos estipulados na etapa inicial do trabalho. Para superar essas dificuldades foram necessários ajustes no projeto de para mantê-lo relevante com as constantes tecnologias que surgem.

A escolha do tema deste trabalho se justificou principalmente pela crescente popularidade dos *chatbots* conversacionais e o reconhecimento da demanda por soluções inovadoras no campo jurídico. A interação natural e intuitiva oferecida por essas interfaces automatizadas tem se mostrado uma ferramenta valiosa em diversas áreas, e a aplicação específica no contexto jurídico-trabalhista representa uma evolução no modo como as informações são acessadas, analisadas e compartilhadas.

Além disso, observa-se que o processo a ser desenvolvido ao longo deste trabalho possui potencial de generalização. A adaptação dos dados de treinamento permite a expansão do modelo para lidar com diferentes domínios ou áreas específicas do direito,

ampliando seu alcance e utilidade. Essa flexibilidade representa não apenas uma conquista no contexto deste projeto, mas também sinaliza para futuras aplicações e aprimoramentos em sistemas similares.

Em relação às tecnologias utilizadas, inicialmente nos foi proposto utilizar a Plataforma Rasa para o desenvolvimento do trabalho, entretanto, nós tivemos o interesse de explorar soluções relativamente novas no contexto acadêmico da UnB/FGA no que diz respeito às ferramentas para a construção de um *chatbot*. Dessa forma, nós estudamos a utilização do Llama 2 e posteriormente do MistralAI, sendo o último a nossa escolha após a realização dos testes e por causa de sua licença de software mais liberal.

6.1 Trabalhos Futuros

Um dos desafios encontrados durante o desenvolvimento do *chatbot* foi a implementação da leitura de documentos e a geração de respostas multimodais simultaneamente. O processo de interpretar documentos extensos, como contratos ou pareceres jurídicos, exigiu um tratamento mais profundo das informações, o que tornou inviável a integração com respostas multimodais (textuais, visuais, auditivas) em tempo real com a infraestrutura que tivemos. Como trabalho futuro, sugere-se a investigação de métodos mais eficientes de processamento e paralelismo, que permitam ao modelo lidar com múltiplas formas de entrada e saída de maneira sincronizada, melhorando a usabilidade e flexibilidade do *chatbot*.

Outro ponto a ser explorado é o treinamento mais específico do *chatbot* com base na Consolidação das Leis do Trabalho (CLT) e em jurisprudências trabalhistas. Devido à complexidade e à quantidade de dados, o treinamento adequado nessas fontes não foi completamente implementado. No futuro, a utilização de modelos específicos para linguagem jurídica ou o desenvolvimento de técnicas de aprendizado transferível pode possibilitar ao *chatbot* fornecer respostas mais precisas e contextualizadas, considerando o histórico e as decisões recentes dos tribunais.

Referências

ADAMOPOULOU, E.; MOUSSIADES, L. An overview of chatbot technology. In: *IFIP Advances in Information and Communication Technology*. [s.n.], 2020. (IFIPAICT, v. 584). Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-50516-5_1>. Citado na página 7.

ALLIANCE, A. 2017. Disponível em: <https://www.agilealliance.org/glossary/mvp/>. Citado na página 27.

BEEDLE, M. *Manifesto Ágil*. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/principles.html>. Citado na página 33.

FIGMA. 2023. Disponível em: <https://www.figma.com/>. Citado na página 28.

MELO, J. *Inteligência artificial: uma realidade no Poder Judiciário*. 2019. Disponível em: <https://www.tjdft.jus.br/institucional/imprensa/campanhas-e-produtos/artigos-discursos-e-entrevistas/artigos/2020/inteligencia-artificial>. Citado na página 13.

META. *Documentação Llama2*. 2023. Acessado em: 16/11/23. Disponível em: <https://ai.meta.com/llama/>. Citado na página 8.

MICROSOFT. *O que é um chatbot? | Microsoft Azure*. 2023. Acessado em: 16/11/23. Disponível em: <https://powervirtualagents.microsoft.com/pt-br/what-is-a-chatbot/>. Citado na página 8.

MISTRALAI. *Documentação Mistral*. 2023. Acessado em: 27/11/23. Disponível em: <https://mistral.ai/news/announcing-mistral-7b/>. Citado 2 vezes nas páginas 17 e 18.

NAKAGAWA, M. 2023. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/>. Citado 2 vezes nas páginas 24 e 27.

NVIDIA. 2023. Acessado em: 16/11/23. Disponível em: <https://www.nvidia.com/en-us/glossary/data-science/large-language-models/>. Citado 3 vezes nas páginas 9, 17 e 19.

OPENAI. 2023. Acessado em: 16/11/23. Disponível em: <https://openai.com/blog/chatgpt>. Citado na página 9.

ORACLE. *O que é Chatbot*. 2023. Acessado em: 16/11/23. Disponível em: <https://www.oracle.com/br/chatbots/what-is-a-chatbot/>. Citado 4 vezes nas páginas 7, 8, 9 e 16.

PROMAD, E. *Erros humanos que podem ser evitados com o uso de Inteligência Artificial no Direito*. 2022. Disponível em: <https://www.promad.adv.br/blog/erros-humanos-que-podem-ser-evitados-com-o-uso-de-inteligencia-artificial-no-direito/>. Citado na página 12.

RASA. 2023. Acessado em: 16/11/23. Disponível em: <https://rasa.com/docs/rasa/>. Citado 2 vezes nas páginas 15 e 16.

RIES, E. *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. 1st ed. ed. New York: Crown Business, 2011. ISBN 9780307887894. Citado na página 33.

RODRIGUES, J. *Quais os melhores chatbots para usar em 2023?* 2023. Disponível em: <https://blog.culte.com.br/quais-os-melhores-chatbots-para-usar-em-2023/>. Citado na página 8.

SCHWABER, K. *What Is Scrum*. 2023. Disponível em: <https://www.scrum.org/learning-series/what-is-scrum>. Citado na página 33.

TIRADENTES, G. 2021. Disponível em: <https://portal.unit.br/blog/noticias/direito-trabalhista-uma-das-areas-mais-dinamicas-do-sistema-juridico/>. Citado na página 11.