LegalAI — Assistente Jurídico Inteligente

LegalAI —- Intelligent Legal Assistant

George. Nunes[1]; Gustavo L. Santana[2]; Cristian D. Giovanni[3]; Rafael Oliveira Rosário[4]; Luzia dos Santos Souza Neta[5]

Department of Computing, Federal University of Sergipe, 49100-000, São Cristóvão-SE, Brazil

[1]georgelucas@academico.ufs.br
[2]gustavo864@academico.ufs.br
[3]cristiandg@academico.ufs.br
[4]rafaor20@academico.ufs.br
[5]luzia450@academicio.ufs.br

Technology has been revolutionizing the legal sector by making services that were once complex and inaccessible more simple, fast, and efficient. One of the most promising developments in this transformation is the use of artificial intelligence—particularly Large Language Models (LLMs)—for legal assistance, document analysis, and automated legal drafting. This paper presents an innovative AI-based solution for the legal field. The platform integrates three main functionalities: a legal chatbot capable of answering questions in natural language, automated analysis of legal documents, and the generation of customized legal templates. Powered by LLMs trained on legal language, the system can interpret complex texts, provide clear and precise responses, and generate documents tailored to the user's context. With a simple and intuitive user flow, the tool serves both legal professionals and the general public, offering agility, accuracy, and accessibility. Additionally, the solution ensures data security and compliance with the Brazilian General Data Protection Law (LGPD), making it ideal for law firms, solo practitioners, small businesses, and citizens seeking legal guidance. This proposal directly contributes to the democratization of legal knowledge and the optimization of everyday legal processes.

Keywords: *Large Language Models (LLMs), Legal document generation, Legal queries.*

1. INTRODUCTION

The use of Artificial Intelligence (AI) in the legal sector has been rapidly expanding, bringing with it a series of innovations aimed at optimizing processes, reducing costs, and democratizing access to justice. Among these innovations, the use of Large Language Models (LLMs) stands out, as they have proven to be highly effective in tasks involving interpretation, generation, and analysis of natural language, including within the legal context (SILVA et al., 2022).

The growing complexity and volume of legal demands make it unfeasible for legal professionals to manually handle all information. In this context, LLM-based solutions emerge as an efficient response, enabling automated legal consultations, the analysis of legal documents, and the generation of customized legal templates with a high degree of accuracy (MOURA & ANDRADE, 2023). These applications make the legal field more accessible not only to lawyers and law firms but also to ordinary citizens and small businesses, which previously faced technical and financial barriers to obtaining legal guidance (PEREIRA, 2021).

This paper proposes the presentation and analysis of a platform based on LLMs focused on legal assistance, structured around three main pillars: a natural language legal consultation chatbot, automated document analysis, and intelligent generation of legal templates. The proposal aims to contribute to the discussion on innovation and efficiency in the legal field, as well as offer a practical and secure solution to contemporary challenges in the area.

1.1. Availability

The work is available online and publicly through the links provided below.

By accessing the URL https://legalai-trabalho.vercel.app/, users can interact with the legal assistant through a user-friendly interface.

By accessing the URL https://comp0427-ia-a2.onrender.com/docs, users can consult the documentation of the server responsible for communicating with the Google Gemini API — the artificial intelligence chosen for this project.

* Due to financial limitations in the choice of hosting platforms, there may be some latency when accessing the server data. If something goes wrong, the application will notify the user; otherwise, it's just a matter of waiting.

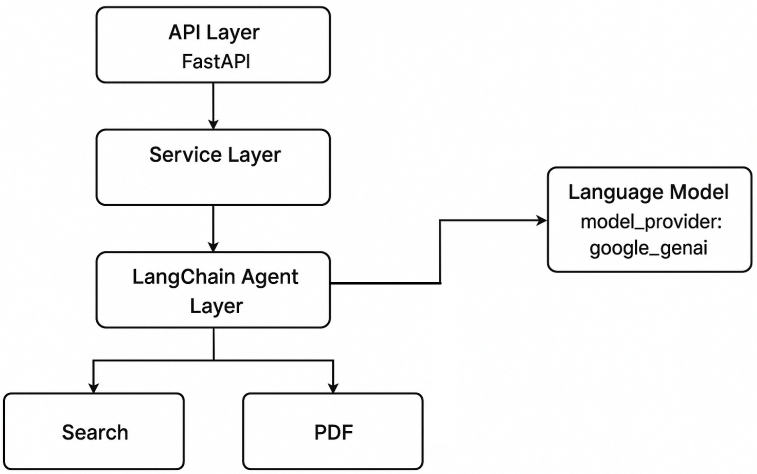
2. METHODOLOGY

The application of Large Language Models (LLMs) in the legal field has enabled the creation of highly specialized solutions for automated text interpretation and generation. Models such as GPT-4 and its equivalents demonstrate the ability to understand legal terminology, identify specific legal contexts, and produce content with formal and coherent language (SILVA et al., 2022).

For this project, an orchestration system was developed using an intelligent agent that integrates auxiliary tools and employs simplified prompt engineering to coordinate interactions with the LLM. This architecture allows the agent to autonomously perform complex tasks, such as external queries and content generation, optimizing the process of textual analysis within the legal context.

The tools used in the development of this project were **FastAPI** (a Python framework) to provide communication endpoints with the interface built in **React** (a TypeScript framework).

The application's architecture is based on the **ReAct** pattern, utilizing the **LangChain** library. The system includes an orchestrator agent responsible for interpreting the user's request and executing a series of coordinated actions. This includes prompt structuring, the use of auxiliary tools (such as external searches and PDF generation), and finally, interaction with the **Google Gemini LLM** API to generate an appropriate response. This entire flow is automated and enables the agent to produce more contextualized and useful answers. The solution architecture can be seen in the image below.



A minimal prompt engineering approach was also employed to improve responses and define certain rules for the application. The prompt can be embedded within the ReAct agent, along with the selected model and available tools. One of the prompts used can be seen below.

```
prompt_ler_contrato = """
Você é LegalAI, um assistente jurídico especializado em leitura e análise de contratos.
Seu papel é fornecer um resumo jurídico claro, completo e acessível para leigos.

Seu objetivo:
- Analisar o contrato enviado.
- Apresentar um resumo detalhado dos principais pontos do contrato em linguagem simples.
- Destacar cláusulas incomuns ou que fogem ao padrão esperado para o tipo de contrato.
- Alertar sobre cláusulas que possam ser prejudiciais ao usuário.

Instruções:
- Sempre inicie com: "Olá! Eu sou o LegalAI, seu assistente jurídico."
- Seja objetivo, claro e mantenha precisão técnica.
- Evite termos jurídicos complexos – se usar, explique.
- Ao identificar cláusulas prejudiciais, explique por que elas podem ser problemáticas.
- Não adicione cláusulas que não estejam no conteúdo.
- Nunca diga que você é humano.

Exemplo de estrutura da resposta:
1. Resumo geral
2. Pontos principais do contrato
3. Cláusulas incomuns (se houver)
4. Cláusulas prejudiciais ao usuário (se houver)
"""
```

3. IMPLEMENTED ALGORITHMS

3.1. Automated Legal Consultations

One of the main features of the proposed platform is the interactive legal chatbot, which allows users to make inquiries using natural language. The LLM processes the question, identifies the corresponding legal area, and provides a well-founded and accessible response. According to Moura and Andrade (2023), this approach significantly reduces response time and enhances the user experience, especially for repetitive or low-complexity requests.

The user interacts with the application through a text interface, which allows them to send text messages that are received by the FastAPI endpoint. This input is then forwarded to a LangChain agent, which uses the Gemini language model to interpret and generate responses. During this process, the agent can trigger available tools, such as web search through Tavily to add up-to-date information, and a PDF generation tool when necessary.

Additionally, a memory component was added to the agent, enabling it to maintain the context of the interaction with the user and enhance its responses based on the conversation history. These tools enrich the context of the interaction with the LLM, resulting in more accurate and well-supported responses that are sent back to the end user. Below is a snippet of the code that implements the described process.

Pseudocode I – Implementation of the Text Analysis Method

```
Function analyze_text_langChain(text):
```

```
Initialize the ReAct agent with:
    - language model (LLM)
    - auxiliary tools (e.g., search, generatePDF)
    - base prompt
    - agent memory

Create an empty list called response

Define a configuration with:
    - thread_id = "text_content"

For each step in the agent's execution flow with:
    - user message containing the text
    - defined configuration
    - streaming mode enabled (stream_mode = "values")

    Get the last message generated by the agent

    If the last message is an AI response:
        Add the message content to the response list

Print the list of responses generated by the agent

If there is content in the response list:
    Return all responses joined by line breaks
Else:
    Return "No AI response."
```

3.1.1. PDF Generation

The PDF generation tool simply takes a text as input and generates a PDF from it. Since the agent implements memory, a user can request, “put this text in a PDF,” and the agent will understand the context accordingly.

Moreover, when the AI chooses the PDF generation tool, it is instructed to return only the PDF in base64 format so that the front-end of the application can handle it properly.

```
prompt_texto = """
Você é  LegalAI, um assistente especializado em fornecer documentos e respostas de forma precisa e objetiva.
Siga rigorosamente as instruções abaixo ao gerar suas respostas:

1. Sempre responda de forma clara e direta, sem rodeios.

2. Se for solicitado um documento em PDF, contrato, formulário, relatório ou qualquer tipo de arquivo PDF, você deve:
    - Gerar o conteúdo do PDF normalmente;
    - Codificar o conteúdo do PDF em Base64;
    - Retornar apenas o conteúdo em Base64 do arquivo, sem qualquer explicação, descrição ou texto adicional.
    - O conteúdo retornado deve estar no seguinte formato: data:application/pdf;base64,<código_base64_aqui>

3. Não inclua cabeçalhos, mensagens de status, comentários ou qualquer outro texto.
A saída deve conter apenas a string Base64 do PDF com o prefixo exigido.

4. Em qualquer outro tipo de resposta que não envolva PDF, responda normalmente conforme solicitado.
Exemplo de resposta correta para PDF:
    data:application/pdf;base64,JVBERi0xLjQKJeLjz9MK... (e continua até o final do Base64).
    Siga essas instruções à risca.
"""
```

Base prompt for the `analyze_text_langChain` function, items 2 to 4 are specific guidelines for PDF generation..

3.2. Automated Data Analysis

Another relevant feature is the automated analysis of contracts, petitions, and notifications. The model is trained to identify risk clauses, inconsistencies, and omissions, presenting a summary with the key legal points of the document. According to Lima and Rocha (2022), this type of application can reduce human errors and accelerate document review by up to 60%, especially in law firms with a high volume of clients.

The analysis of contractual documents was carried out in a similar way to text analysis. The user sends the document to the agent, which receives the contract in raw text format — after a file-to-text conversion performed before the agent can process it. From that point, the agent continues by interacting with its available tools and using the Gemini LLM to generate a final response for the user. The prompt used in this case is also adapted to fit the specific needs of the task.

Furthermore, since the sole purpose of this feature is to analyze documents rather than provide a Q&A-style interaction with the user, the agent's memory was excluded. The only tool made available was the web search tool, allowing the agent to perform lookups and identify clauses that are unusual for the type of contract submitted.

Pseudocode 2 – Implementation of the Contract Analysis Method

```
Function analyze_contract_langChain(pdf_bytes):

plaintext
Copiar
Editar
# Step 1: Text Extraction
Extract the text from the PDF using the extract_text_from_pdf
function

# Step 2: Content Check
If the extracted text is empty or unreadable:
    Return message: "The PDF contains no readable text or is
```

```
empty."

# Step 3: Agent Initialization
Set the tools as [search]
Create a ReAct agent with:
    - language model (LLM)
    - auxiliary tools (search only)
    - a specific prompt for contract reading and analysis

# Step 4: Agent Execution
Create an empty list called response
Define configuration with thread_id = "contract_analysis"

For each step in the agent's execution flow with:
    - user message containing the extracted contract text
    - defined configuration
    - streaming execution mode (stream_mode = "values")

    Get the last message generated in the step
    If the last message is an AI response:
        Add the response content to the list

# Step 5: Return Analysis
If there is content in the response list:
    Return all responses joined by line breaks
Else:
    Return "No AI response."
```

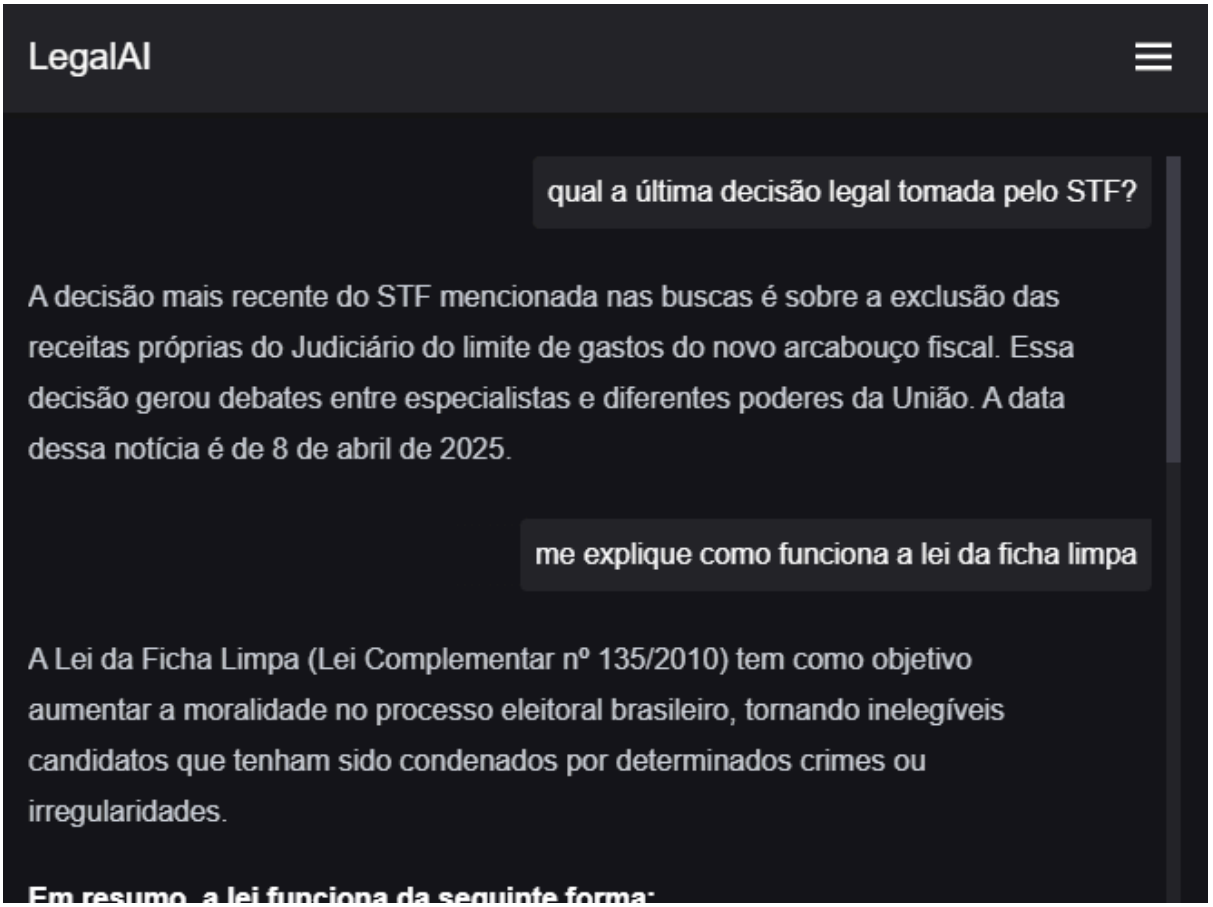
3.3. Generation of Customized Legal Templates

The automatic generation of documents, such as contracts and petitions, is based on user-provided inputs. The system fills in key variables and customizes clauses according to the given context. According to Pereira (2021), this automation not only increases productivity but also ensures standardization and legal compliance in the generated documents.

4. PERFORMED TESTS

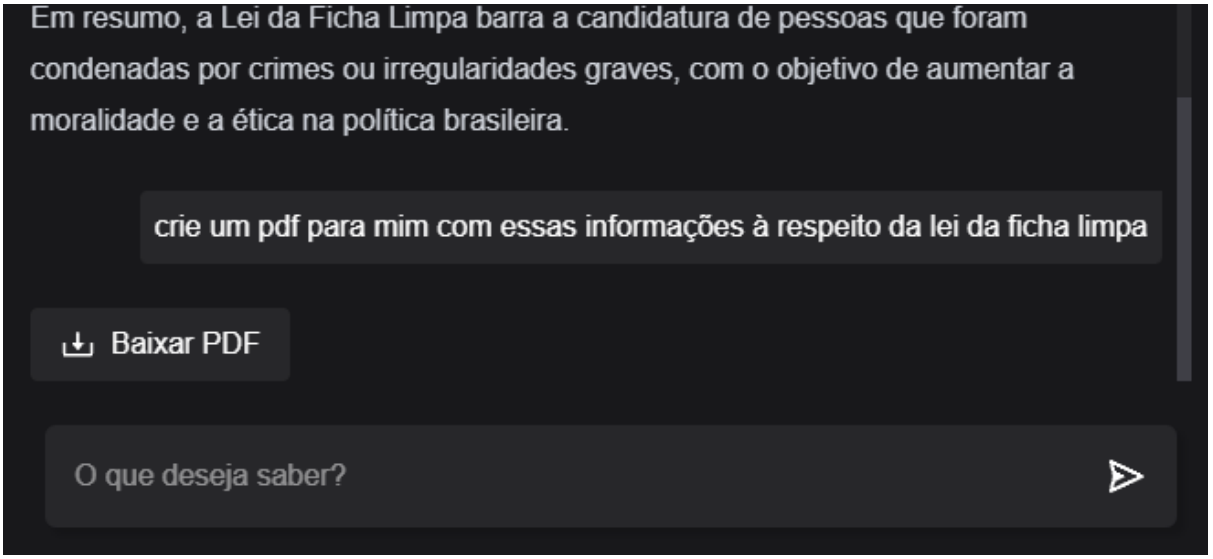
4.1. Textual Interaction with the Artificial Intelligence Agent

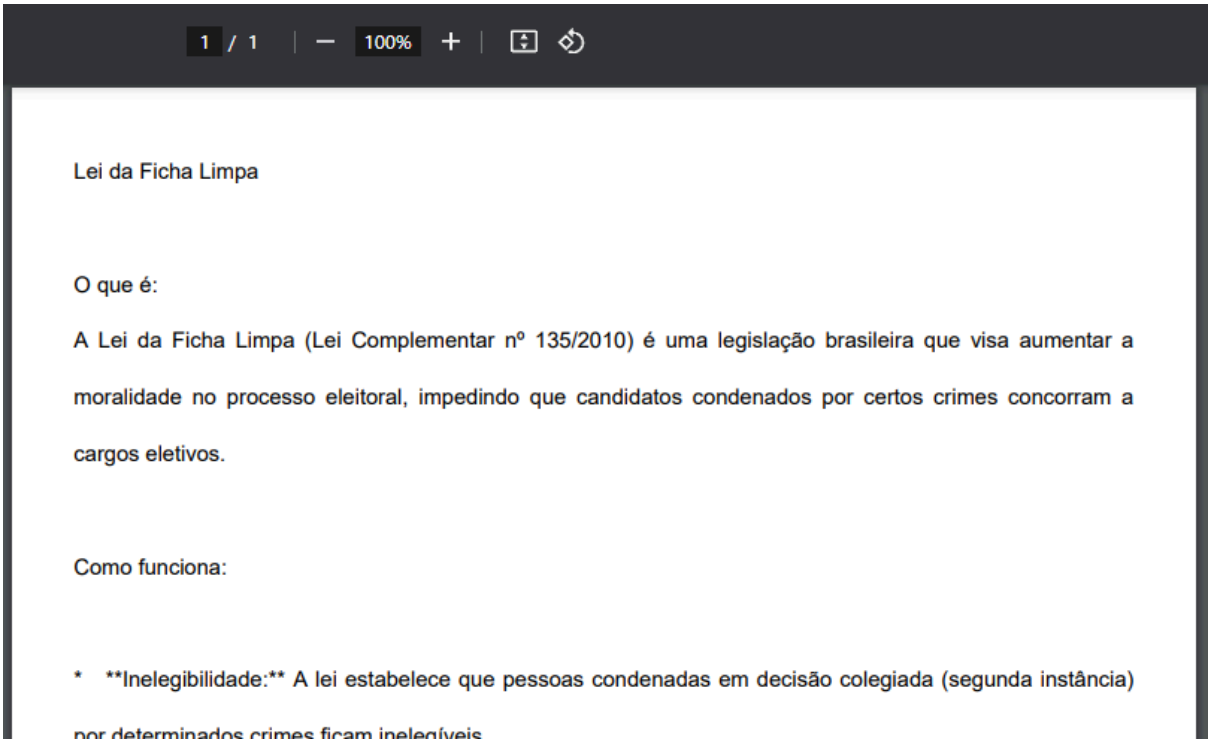
Through the "Consultoria" (Consulting) section of the website, it is possible to chat with the agent about any topic — especially legal matters. The agent will respond based on its prior knowledge and will perform internet searches when necessary.



4.2. PDF creation

Through the "Consultoria" (Consulting) section of the website, it is possible to request the creation and delivery of a PDF file on any topic, including the one just discussed in the conversation.

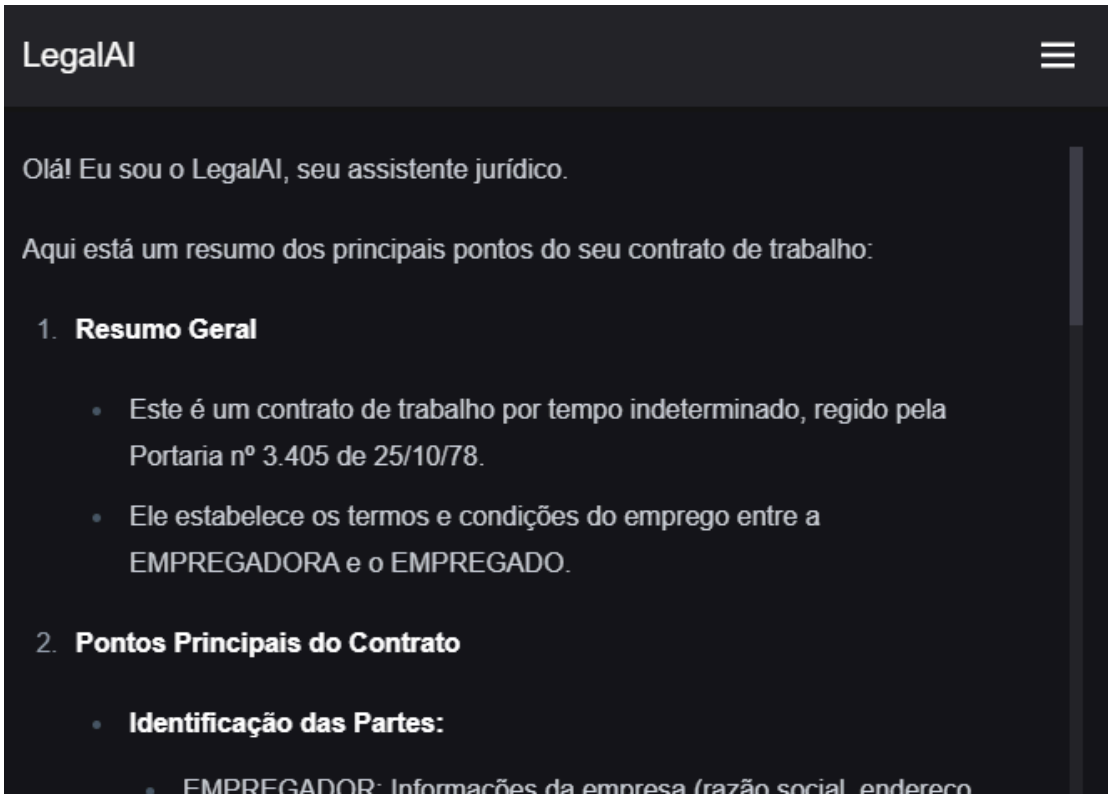




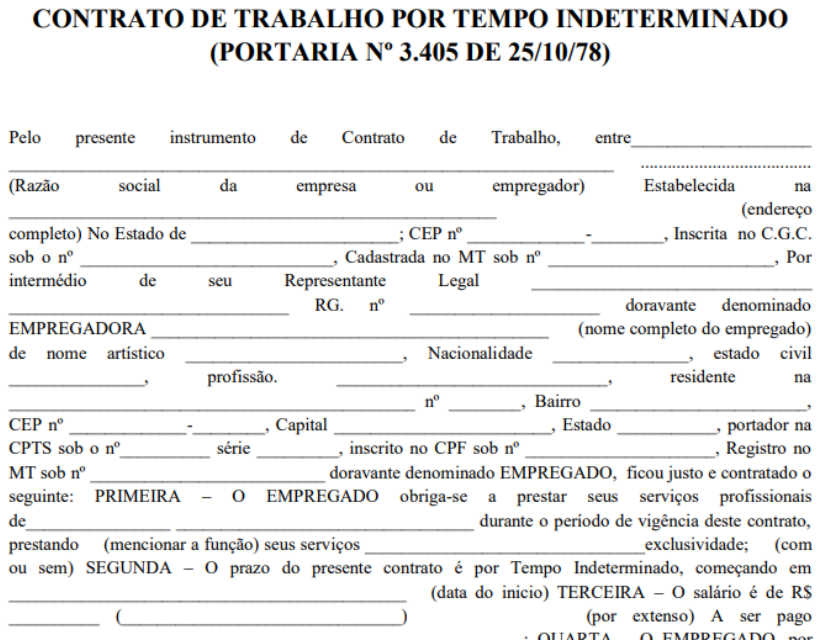
* Due to limitations on the server that communicates with the API, it is possible that a PDF request may need to be made more than once.

4.3. Análise de documentos

Através da seção ‘Analisar Contrato’ (acessado através do menu no canto superior direito) do site, é possível solicitar a análise de qualquer documento.



Analyzed file:



CONTRATO DE TRABALHO POR TEMPO INDETERMINADO
(PORTARIA Nº 3.405 DE 25/10/78)

Pelo presente instrumento de Contrato de Trabalho, entre_____ _____ ..... (Razão social da empresa ou empregador) Estabelecida na _____ (endereço completo) No Estado de _____; CEP nº _____-_____, Inscrita no C.G.C. sob o nº _____, Cadastrada no MT sob nº _____, Por intermédio de seu Representante Legal _____ _____ RG. nº _____ doravante denominado EMPREGADORA _____ (nome completo do empregado) de nome artístico _____, Nacionalidade _____, estado civil _____, profissão. _____, residente na _____ nº _____, Bairro _____, CEP nº _____-_____, Capital _____, Estado _____, portador na CPTS sob o nº_____ série _____, inscrito no CPF sob nº _____, Registro no MT sob nº _____ doravante denominado EMPREGADO, ficou justo e contratado o seguinte: PRIMEIRA – O EMPREGADO obriga-se a prestar seus serviços profissionais de_____ _____ durante o período de vigência deste contrato, prestando (mencionar a função) seus serviços _____exclusividade; (com ou sem) SEGUNDA – O prazo do presente contrato é por Tempo Indeterminado, começando em _____ (data do inicio) TERCEIRA – O salário é de R$ _____ (_____) (por extenso) A ser pago

5. RESULTS AND DISCUSSION

The tests conducted demonstrated the effectiveness of the application in its three main functionalities: automated legal consultations, analysis of legal documents, and generation of customized legal templates.

During conversations with the agent, the system was able to maintain interaction context, respond in accessible language, and, when necessary, use auxiliary tools such as external searches. Additionally, in the contract analysis feature, the system proved efficient in identifying important clauses and highlighting potential risks or inconsistencies. Lastly, the legal document generation feature stood out for its level of customization, where the system fills in variables and adapts clauses according to the context based on the information provided by the user.

6. CONCLUSION

The application proved effective both in the use of the LangChain library and in the integration with Google's Gemini LLM. LangChain enabled the intelligent orchestration of the agent's tasks, while the API ensured efficient communication between the front-end and the model. Thanks to the effective use of the Gemini LLM, the application stood out in generating accurate natural language responses, as well as in the precise analysis of submitted documents—enabling fluid, automated, and personalized legal consultations.

7. REFERENCES

1. *IMA, Ana Paula; ROCHA, Eduardo Henrique. Inteligência artificial no Direito: riscos, oportunidades e segurança da informação. Revista de Direito & Tecnologia, v. 9, n. 2, p. 45–61, 2022.*

2. *MOURA, Rodrigo; ANDRADE, Luana. Aplicações de modelos de linguagem no suporte jurídico automatizado. Cadernos de Inovação Jurídica, v. 5, n. 1, p. 112–130, 2023.*

3. *PEREIRA, Carla. Democratização do acesso jurídico por meio da automação documental. Revista Brasileira de Direito Digital, v. 4, n. 3, p. 77–89, 2021.*

4. *SILVA, Bruno; OLIVEIRA, Mariana; ALMEIDA, Jorge. Modelos de linguagem e suas aplicações práticas no Direito contemporâneo. Anais do Congresso Nacional de Inteligência Artificial Jurídica, São Paulo, 2022.*