JurisMediar ou ExtraJud ou

**Mediatrix**

ou outro a escolher

**1. Título do Projeto:**

**Mediatrix: Plataforma de Mediação e Conciliação Online com Assistência de Modelos de Linguagem (LLM)**

**2. Introdução**

**2.1. Contextualização e Justificativa**

Com a crescente demanda por métodos ágeis e acessíveis de resolução de conflitos, o Conselho Nacional de Justiça (CNJ), por meio da Resolução 125, estabelece diretrizes para mediação e conciliação, incentivando métodos alternativos para promover a paz social e desafogar o sistema judiciário. A adoção de uma plataforma de conciliação e mediação online que utiliza LLMs pode ampliar o acesso a métodos extrajudiciais, oferecendo um ambiente seguro e orientado por inteligência artificial para que as partes possam resolver seus conflitos de forma rápida, efetiva e acessível.

**2.2. Problema**

Os métodos tradicionais de resolução de conflitos enfrentam desafios, como o tempo elevado de resposta e a necessidade de mediadores treinados e disponíveis em larga escala. A ausência de mediadores e conciliadores capacitados em localidades remotas ou com alta demanda compromete o acesso rápido e eficiente à justiça.

**2.3. Objetivo Geral**  
Desenvolver uma plataforma de mediação e conciliação online, assistida por LLMs, que atenda às diretrizes da Resolução 125 do CNJ, ampliando o acesso à resolução de conflitos extrajudiciais.

**2.4. Objetivos Específicos**

* Automatizar a detecção de linguagem inflamatória e propor reformulações mais neutras para favorecer o diálogo.
* Auxiliar mediadores ao sugerir intervenções baseadas no histórico da conversa entre as partes.
* Promover acordos amigáveis por meio de uma interface que guia as partes em direção a uma solução.
* Disponibilizar um modelo que intervenha de forma autônoma em disputas de baixa complexidade, promovendo soluções amigáveis onde a intervenção humana é limitada.

Inflamatória – podemos trocar essa palavra ou deixar

**3. Referencial Teórico**

**3.1. Resolução 125 do CNJ**

Contextualização da Resolução 125, destacando seu foco na implementação de políticas públicas para mediação e conciliação, com base na promoção do acesso à justiça e na resolução pacífica de conflitos.

**3.2. Modelos de Linguagem e Inteligência Artificial no Direito**

Discussão sobre o uso de LLMs (como Meta Llama) em cenários de mediação e conciliação, ressaltando estudos de casos e plataformas já existentes, como o LLMediator, que utilizam IA para apoiar a resolução de disputas extrajudiciais.

**3.3. Mediação e Conciliação no Ambiente Digital**

Exame das práticas e limitações da mediação e conciliação em ambientes digitais, enfatizando como a tecnologia pode facilitar o processo e tornar a resolução de conflitos mais acessível.

**3.4. Lei nº 13.140/2015 – Lei da Mediação**

A Lei nº 13.140/2015 dispõe sobre a mediação entre particulares e sobre a autocomposição de conflitos no âmbito da administração pública. A legislação estabelece regras e princípios fundamentais para o processo de mediação,

**3.5 Código de Processo Civil (CPC) de 2015**

O Código de Processo Civil brasileiro de 2015 também fortalece os métodos de autocomposição ao instituir a audiência de conciliação e mediação como etapa obrigatória nos processos judiciais cíveis, exceto quando houver manifestação contrária de ambas as partes.

**4. Metodologia**

**4.1. Arquitetura da Plataforma**  
Descrição da estrutura da plataforma, que incluirá:

* **Módulo de Mediação e Conciliação Assistida por IA**: onde o LLM analisa o conteúdo das conversas, identifica linguagem ofensivas e propõe reformulações para manter o tom cordial e neutro.
* **Módulo de Assistência ao Mediador**: sugere respostas baseadas no contexto da discussão, que o mediador pode adaptar para guiar as partes.
* **Módulo de Intervenção Autônoma**: para disputas de baixo impacto financeiro e complexidade, onde o modelo poderá propor soluções sem intervenção humana direta.

**4.2. Desenvolvimento e Treinamento do LLM**

* Adaptação e treinamento de um modelo de linguagem baseado em dados de mediação e conciliação, respeitando a ética e a privacidade.
* Utilização de técnicas de processamento de linguagem natural para interpretar a comunicação entre as partes e identificar conflitos potenciais.

**4.3. Procedimentos de Avaliação**

* **Testes de Usabilidade**: aplicação de testes com usuários simulando casos de conciliação, avaliando a eficácia do sistema em promover acordos amigáveis.
* **Avaliação de Conformidade com a Resolução 125 do CNJ**: garantir que o fluxo de mediação e conciliação esteja de acordo com as diretrizes da Resolução.

**4.4. Requisitos Legais e Éticos**

* Garantir que a plataforma esteja em conformidade com a Lei Geral de Proteção de Dados Pessoais (LGPD).
* Implementar princípios éticos, com transparência sobre o uso de IA e respeito ao direito das partes em tomar decisões informadas.

**5. Resultados Esperados**

**5.1. Aumento no Acesso à Mediação e Conciliação**

Espera-se que a plataforma facilite o acesso à mediação e conciliação extrajudicial, especialmente para pessoas que residem em áreas com poucos recursos judiciais.

**5.2. Eficiência e Redução de Custos**

Com a assistência dos LLMs, é esperado que os casos sejam resolvidos com maior rapidez e com menos necessidade de intervenção de mediadores humanos.

**5.3. Acordos Mais Sustentáveis**

A reformulação de mensagens inflamadas e o apoio em comunicações neutras deve resultar em acordos mais amigáveis e sustentáveis, com menores chances de reincidência de conflitos.

**6. Cronograma**

| **Etapa** | **Mês 1** | **Mês 2** | **Mês 3** | **Mês 4** | **Mês 5** | **Mês 6** |
| --- | --- | --- | --- | --- | --- | --- |
| Pesquisa e Revisão de Literatura | X |  |  |  |  |  |
| Planejamento da Plataforma | X | X |  |  |  |  |
| Desenvolvimento e Testes |  | X | X | X |  |  |
| Treinamento do Modelo |  |  | X | X |  |  |
| Testes de Usabilidade |  |  |  | X | X |  |
| Avaliação e Ajustes Finais |  |  |  |  | X | X |
| Finalização e Documentação |  |  |  |  |  | X |

**7. Orçamento (Estimativa)**

Desenvolvimento de Plataforma e Integração do LLM: R$

Treinamento do Modelo e Custos de Computação: R$

Testes de Usabilidade e Avaliações: R$

Custos com Licenciamento e Manutenção: R$

Total Estimado: R$

**8. Conclusão**

A criação de uma plataforma de mediação e conciliação online assistida por LLM representa uma evolução importante no acesso à justiça e na eficiência de resolução de conflitos extrajudiciais. Esse projeto pode democratizar o acesso à mediação e ajudar a descongestionar o sistema judiciário, em consonância com as diretrizes da Resolução 125 do CNJ, promovendo a paz social de maneira eficiente e acessível.

Aceito sugestões de Títulos para a projeto.

**Silvana:**

JurisMediador

JurisDigital

JurisMediar

Chama o Juris

**Aurélio:**

ExtraJud

**ChatGPT:**

JustiçaFácil

Mediar+

Conciliar+

O Conciliador

Mediatrix

**Atenção, Turma:**

Eu pedi ajuda ao ChatGPT para criar um exemplo de código com base apenas no título do projeto. Minha solicitação inicial foi simplesmente um exemplo de como poderíamos estruturar o título no código, mas o ChatGPT forneceu um protótipo completo, com uma sugestão de estrutura para a plataforma de mediação e conciliação. Como essa não é minha área de domínio, não tenho como validar se essa estrutura atende totalmente às nossas necessidades.

É importante destacar que o protótipo abaixo foi criado pelo ChatGPT e contém algumas partes específicas, como o uso de "GPT" para o modelo de linguagem. Como optamos por utilizar o **Meta Llama**, algumas adaptações serão necessárias para ajustar o código ao modelo escolhido.

Caso o ChatGPT tenha se excedido em alguma parte ou alucinado alguma funcionalidade, só saberemos com a implementação prática do projeto. Além disso, as estruturas específicas que o Meta Llama vai exigir para integrar ao nosso sistema ainda precisam ser revisadas e confirmadas e provavelmente só saberemos no dia do treinamento, quando eles passagem as coordenadas. Acredito, que assim será. Mas até lá vamos trocando figurinhas

**Para o projeto**

Para o projeto **Mediatrix: Plataforma de Mediação e Conciliação Online com Assistência de Modelos de Linguagem (LLM)**, aqui está um exemplo inicial de código que implementa funcionalidades básicas para uma plataforma de mediação e conciliação assistida por modelos de linguagem. Esse código inclui a criação de uma interface para o envio de mensagens entre as partes, o uso de um modelo de linguagem para reformular mensagens e uma função básica de análise de tom para garantir um ambiente de mediação mais amigável.

**Exemplo de Código**

Para este exemplo, vamos criar uma aplicação web usando **Python** e **Flask** para a interface de mediação e o processamento de linguagem natural com Meta Llama (ou outro modelo de linguagem, como o BERT para análise de tom).

**Estrutura Básica do Projeto**

app.py: Arquivo principal do Flask para a aplicação.

templates/index.html: Página da interface do usuário.

model.py: Lógica para integração Meta Llama e análise de tom.

static/style.css: Estilos básicos para a interface.

Protótipo inicial

1. Código do Arquivo Principal (app.py)

from flask import Flask, render\_template, request, jsonify

from model import reformulate\_message, analyze\_tone

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/send\_message', methods=['POST'])

def send\_message():

user\_message = request.json['message']

# Analisar o tom da mensagem

tone = analyze\_tone(user\_message)

# Verificar se a mensagem tem tom agressivo e, se necessário, reformular

if tone == "agressivo":

reformulated\_message = reformulate\_message(user\_message)

return jsonify({'status': 'reformulated', 'message': reformulated\_message})

else:

return jsonify({'status': 'ok', 'message': user\_message})

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

**2. Lógica de Processamento (model.py)**

Este arquivo contém as funções para integração com o modelo de linguagem Meta Llama e a análise de tom da mensagem.

import openai

# Defina sua chave de API do OpenAI

openai.api\_key = "sua\_chave\_de\_api"

def reformulate\_message(message):

"""Função para reformular uma mensagem usando o GPT-4"""

prompt = f"Reformule a seguinte mensagem para um tom mais neutro e amigável: '{message}'"

response = openai.Completion.create(

engine="text-davinci-004", # ou substitua pelo modelo adequado

prompt=prompt,

max\_tokens=60

)

reformulated\_text = response.choices[0].text.strip()

return reformulated\_text

def analyze\_tone(message):

"""Simples análise de tom usando palavras-chave para detecção de tom agressivo."""

aggressive\_keywords = ["idiota", "incompetente", "nunca", "culpa"]

if any(word in message.lower() for word in aggressive\_keywords):

return "agressivo"

return "neutro"

**3. Interface do Usuário (index.html)**

Aqui está uma interface simples em HTML para que o usuário envie mensagens e receba feedback.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Mediatrix - Plataforma de Mediação</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<h1>Mediatrix: Plataforma de Mediação e Conciliação Online</h1>

<div id="chatbox">

<div id="messages"></div>

<input type="text" id="user-input" placeholder="Digite sua mensagem...">

<button onclick="sendMessage()">Enviar</button>

</div>

<script>

async function sendMessage() {

const userInput = document.getElementById('user-input').value;

const response = await fetch('/send\_message', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ message: userInput })

});

const data = await response.json();

const messageBox = document.getElementById('messages');

if (data.status === 'reformulated') {

messageBox.innerHTML += `<p><strong>Mediatrix:</strong> ${data.message} (Mensagem reformulada para um tom neutro)</p>`;

} else {

messageBox.innerHTML += `<p><strong>Usuário:</strong> ${data.message}</p>`;

}

document.getElementById('user-input').value = '';

}

</script>

</body>

</html>

4. CSS Básico (style.css)

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

text-align: center;

}

#chatbox {

width: 500px;

margin: 0 auto;

padding: 10px;

background-color: #ffffff;

border: 1px solid #ddd;

border-radius: 8px;

}

#messages {

height: 300px;

overflow-y: scroll;

margin-bottom: 10px;

padding: 10px;

border: 1px solid #ddd;

background-color: #f9f9f9;

border-radius: 8px;

}

#user-input {

width: 80%;

padding: 8px;

border-radius: 4px;

border: 1px solid #ddd;

}

button {

padding: 8px 12px;

border: none;

border-radius: 4px;

background-color: #007bff;

color: #fff;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

**Explicação do Código**

* **Interface (HTML)**: A interface permite que o usuário insira uma mensagem e receba uma resposta, exibindo se a mensagem foi reformulada pelo LLM.
* **Funções no Backend (Python)**:
  + send\_message(): Recebe a mensagem do usuário e analisa o tom. Se o tom for identificado como agressivo, a mensagem é reformulada antes de ser enviada ao chat.
  + reformulate\_message(): Usa o modelo GPT-4 para reformular a mensagem para um tom mais neutro.
  + analyze\_tone(): Verifica se a mensagem contém palavras potencialmente agressivas.

**Considerações**

Esse é um protótipo inicial. Para uma aplicação real:

* **Expandir a análise de tom**: Utilizar APIs de NLP mais avançadas ou treinar um modelo específico.
* **Aprimorar o processamento de linguagem**: Integrar uma camada de segurança para monitorar o uso de linguagem inadequada.
* **Implementar autenticação e segurança**: Para proteger as mensagens e informações dos usuários.

Este código pode servir como base para a implementação de funcionalidades mais complexas na plataforma **Mediatrix**.