

# PalanqueIA: A Retrieval-Augmented Generation Chatbot for Government Plans in the Brazilian Electoral Landscape

## Abstract

**Context:** According to the Brazilian electoral code, candidates for Executive positions are required to present their government plans. This plan outlines proposed actions and policies, though implementation is discretionary. Proposed legislation (Bill 2082/21) requires the electoral courts to provide online access to these plans, aiming to counter electoral demagoguery. **Problem:** In Brazilian politics, electoral campaigns often lack substantive proposals for regional development, prioritizing self-promotion and personal attacks over meaningful debate. Candidates frequently make unrealistic promises beyond legal or budgetary limits, fostering disinformation and public mistrust. **Proposed Solution:** The proposed solution entails the development of a conversational system. PalanqueIA is an advanced virtual assistant designed to support candidates, analysts, and policymakers in the formulation of public policies. Utilizing artificial intelligence and Retrieval-Augmented Generation, it facilitates users' access to, exploration of, and generation of innovative policy proposals based on comprehensive data and contextual insights. **Information Systems Theory:** The work is grounded in the theory of information retrieval in Information Science and Computer Science. **Method:** The methodology employed was a Proof of Concept. The vector database was constructed from official documents, and user interaction with the chatbot was monitored to evaluate the accuracy of the responses. **Summary of Results:** The findings demonstrated that the chatbot is capable of executing complex queries efficiently, providing relevant and contextualized answers. The combination of a large language model and the vector base proved effective in retrieving information. **Contributions and Impact on the Information Systems Area:** This research contributes to the Information Systems field by demonstrating how Information Retrieval techniques can be enhanced by advanced language models in specific search contexts.

## Keywords

Government plans, chatbot, conversational AI, retrieval-augmented generation, elections

## 1 Introdução

Incluído pela Lei nº 12.034 de 2009 [4], a qual estabeleceu novas regras para o processo eleitoral, o pedido de registro de candidatura dos cargos a Prefeito, Governador do Estado e Presidente da República, deve vir acompanhada, obrigatoriamente, das propostas de governo do candidato.

O plano de governo é um documento no qual candidatos a cargos do Executivo apresentam à sociedade suas propostas, planejamentos e ideias de ações e intenções que pretendem implementar caso sejam eleitos durante o mandato. As propostas não são juridicamente vinculativas, permitindo ao agente político eleito discricionariedade quanto à sua implementação.

A divulgação de conteúdos é de particular importância. O Projeto de Lei 2082/21 [6], atualmente em tramitação na Câmara dos Deputados, prevê que o Tribunal Superior Eleitoral (TSE) e os Tribunais Regionais Eleitorais indiquem aos eleitores, em suas páginas oficiais na internet, formas de acesso ao conteúdo do conteúdo supracitado. Segundo o autor do Projeto de lei, a proposta política é o antídoto à demagogia eleitoral.

### 1.1 Problemática

No cenário político brasileiro, as campanhas eleitorais frequentemente são marcadas pela superficialidade e pela falta de compromisso com propostas sólidas visando ao desenvolvimento regional. Em vez de focar em ideias e iniciativas concretas, os debates políticos geralmente priorizam a autopromoção e, frequentemente, ataques pessoais e retórica vazia, desviando a atenção das questões fundamentais que impactam o bem-estar social. [1]

Outro aspecto considerável são as esdrúxulas promessas feitas em campanha, por vezes desvinculada de um plano de execução factível. Ao propor medidas que vão além das competências legais ou sem consideração pelas restrições orçamentárias e legislativas, candidatos não apenas desviam o foco das reais necessidades do público, mas também promovem um ambiente de desinformação, minando o debate político qualificado. Tais promessas, ao não serem cumpridas, contribuem para o descrédito das instituições políticas e perpetuam um ciclo de frustração e desconfiança na sociedade, comprometendo a responsabilização e a eficácia do processo eleitoral. [7]

Esse quadro afeta diretamente a qualidade da informação disponível para os eleitores, fazendo com que a sociedade desconheça e não compreenda as ideias políticas dos candidatos, e desmotivando um debate público mais substancial. Como resultado, o cenário eleitoral acaba favorecendo práticas como captação ilícita de sufrágio e difusão de desinformação, ao invés de gerar discussões informadas sobre o futuro social.

### 1.2 Solução Proposta

Este trabalho apresenta o PalanqueIA, um assistente virtual inteligente projetado para auxiliar os candidatos, analistas, gestores de políticas públicas e outros profissionais, no desenvolvimento de políticas públicas. Construído sob o formato de um *ChatBot*, agente conversacional autônomo, ou simplesmente agente, baseado em inteligência artificial (IA) e técnicas de Recuperação Aumentada de Dados (RAG), o sistema proposto é especialista em recuperação e criação de propostas políticas.

A ferramenta integra uma ampla base de dados com milhares de propostas políticas encaminhadas pelos candidatos na ocasião de sua candidatura, associada a modelos de linguagens amplos (LLMs), possibilitando aos usuários consultar propostas existentes, explorar alternativas e obter sugestões para novas políticas. A arquitetura RAG emprega técnicas avançadas de recuperação e geração de informações, permitindo que o *chatbot* responda a consultas com

precisão e contextualização, oferecendo *insights* baseados em políticas já formuladas e em sugestões geradas pela inteligência artificial.

O presente estudo apresenta uma contribuição à Ciência Política aliada à teoria de Recuperação da Informação (TRI) no sentido de ampliar o acesso a informações sobre planos de governo e incentivar a participação cidadã. O diferencial do projeto está no uso de um banco de dados de propostas políticas inovador, além do desenvolvimento de mecanismo otimizado de consulta inicial baseado no reconhecimento de entidades, que melhora a precisão das informações recuperadas. Além disso, o PalanqueIA gera propostas de governo criativas a partir de planos existentes, fomentando a formulação de políticas públicas originais e fundamentadas em dados. A relevância da pesquisa é destacada pela demanda por ferramentas que facilitem a análise crítica de planos de governo, em um cenário político frequentemente marcado por desinformação. Assim, o PalanqueIA se posiciona como uma solução tecnológica para qualificar os debates públicos e promover um processo eleitoral mais transparente e democrático.

### 1.3 Objetivos

O objetivo principal deste estudo é realizar uma prova de conceito de um sistema inteligente de recuperação e proposição de planos de governo utilizando técnicas RAG e LLMs. O PalanqueIA pretende oferecer um ambiente acessível e interativo, onde candidatos, eleitores e formuladores de políticas públicas possam interagir de maneira qualificada com propostas políticas existentes, facilitando o processo de criação e aprimoramento de planos de governo por meio de consultas em temas específicos.

No sentido de alcançar a implementação do sistema e validar a prova de conceito, o estudo se baseou em três objetivos secundários:

- Construir um banco de dados vetorial de planos de governo de candidatos políticos
- Criar um mecanismo de recuperação da informação
- Desenvolver um sistema de informação em formato de *Chat-Bot*

### 1.4 Abordagem Teórica

Segundo os autores[3], a TRI se concentra em buscar, classificar e apresentar os documentos relevantes de uma coleção, em resposta à consulta de um usuário. Os processos principais envolvidos são: indexação, operações de consulta, busca e classificação e interface do usuário.

Esse campo é fundamentado em modelos matemáticos e computacionais, que analisam a similaridade entre a consulta do usuário e os documentos disponíveis com base em técnicas que avaliam a correspondência semântica, a frequência de termos e a estrutura dos dados.

Os autores [3] também destacam a importância da avaliação na RI, utilizando métricas como revocação, precisão e medidas de classificação para determinar a eficácia do sistema. Segundo os autores, encontrar o equilíbrio ideal entre revocação e precisão é um desafio na RI.

No presente estudo, a teoria de RI é aplicada baseada no modelo vetorial de recuperação. Esse modelo representa documentos e consultas como vetores em um espaço multidimensional, onde cada termo ou palavra ocupa uma dimensão específica. Neste modelo, a

relevância de um documento para uma consulta é calculada com base na similaridade de seus vetores — neste caso, medida pelo cosseno do ângulo entre os vetores. Se o ângulo for pequeno, a similaridade é maior, indicando que o documento é relevante para a consulta.

A RI e o RAG estão fortemente relacionados, uma vez que o RAG depende de técnicas de RI para fornecer informações relevantes que servem de base para a geração de respostas de linguagem natural. Portanto, o RAG pode ser considerado uma integração do modelo vetorial da RI com os LLMs.

## 2 Trabalhos Relacionados

A técnica RAG emergiu como uma promissora abordagem na área de Processamento de Linguagem Natural (PLN), através do trabalho em conjunto dos LLMs, responsáveis por facilitar o entendimento do usuário e gerar respostas em linguagem natural, com a recuperação de dados disponibilizados em inúmeros fontes e domínios. Inúmeros estudos têm explorado as aplicações RAG em vários assuntos de pesquisa, revelando seu potencial em melhorar as respostas em sistemas baseados em diálogos e perguntas-respostas. [17]

O artigo seminal que introduziu os conceitos básicos da técnica RAG foi publicado em 2020. Os autores buscavam solucionar o problema de limitação da precisão das LLMs em tarefas intensivas de conhecimento. Além disso, fornecer proveniência para suas decisões e a atualização do conhecimento ao longo do tempo foram dificuldades encontradas pelos autores. [12]

Uma notável revisão sistemática[10] descreve os rápidos avanços em RAG. Os autores reconhecem as limitações significativas dos LLMs e reconhece que a técnica RAG reduz efetivamente o problema de conteúdo factóide, estabelecendo como tecnologia-chave no avanço dos *chatbots*.

Uma das principais contribuições deste estudo foi a apresentação da Árvore tecnológica da pesquisa RAG. Os autores ainda classificam em três modos o envolvimento da RAG com os LLMs, cada qual em uma abordagem distinta:

- **Inferência:** concentra os primeiros estudos na técnica. A informação é recuperada e enviada à LLM no momento em que ele gera uma resposta ao usuário, acessando dados em tempo real.
- **Ajuste-fino:** Após o aprofundamento da técnica, a mesma foi se integrando ao ajuste fino em LLMs pré-treinadas. O processo de ajuste fino é incorporado com buscas e integração de informações.
- **Pré-treinamento:** Por fim, os grandes modelos de linguagem começaram a ser treinados desde sua fundação utilizando conjunto de dados expandidos através de RAG.

Além desta contribuição, a revisão classifica a pesquisa RAG em três paradigmas: ingênuo, avançado e modular. Segundo os autores, apesar do RAG ingênuo ter custo-benefício e superar o LLM nativo, ele apresenta limitações. O RAG avançado e o modular surge como resposta a essas deficiências.

- **Ingênuo:** representa a metodologia inicial utilizada. Segue uma estrutura tradicional de “recuperação-leitura” em três etapas: indexação, recuperação e geração. Este paradigma

possui problemas com precisão e recall, além de criar alucinações quando o modelo gera conteúdo além dos documentos recuperados.

- **Avançado:** utilizando estratégias de pré e pós-recuperação, o paradigma avançado aprimora o RAG ingênuo. São usadas técnicas de otimização da indexação. A consulta inicial é transformada por refinamento e reescrita, na fase pré-recuperação. Já na fase pós, métodos como reordenação dos fragmentos e compressão do contexto ajudam a selecionar e priorizar informações relevantes.
- **Modular:** é um aprimoramento dos paradigmas anteriores, com adição de módulos de busca e refinamento do recuperador via ajuste fino.

O artigo [13] propõe uma nova abordagem para aprimorar a pesquisa em banco de dados usando LLMs. Em vez de utilizar diretamente a consulta do usuário, os autores introduzem uma etapa de reescrita de consulta, visando preencher a lacuna entre a necessidade de informação do usuário e a maneira como essa necessidade é expressa. O conceito central envolve o emprego de um LLM para gerar uma consulta mais eficaz, que é posteriormente utilizada por um mecanismo de busca para recuperar informações relevantes. Essa técnica se mostra particularmente benéfica em cenários em que os usuários formulam perguntas complexas ou ambíguas. Os autores demonstram que essa reescrita de consulta, seja realizada por um LLM pré-treinado ou um modelo menor treinado especificamente para essa tarefa, leva consistentemente a um melhor desempenho de pesquisa. O artigo postula que a reescrita de consulta facilita uma pesquisa mais precisa, resultando na recuperação de documentos mais relevantes e, conseqüentemente, respostas mais precisas do LLM.

Em um interessante trabalho [11], ao investigar as estratégias de decodificação utilizadas para gerar texto a partir de modelos de linguagem neurais, o autor detalha o impacto do uso do parâmetro *temperature*, que controla a aleatoriedade do texto gerado por uma LLM, e o parâmetro *top-p*, também conhecido como *Nucleus Sampling*, o qual define um limite dinâmico para a amostragem de tokens.

### 3 Metodologia

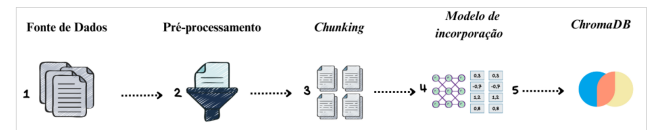
A metodologia deste estudo é estruturada em torno da implementação do PalanqueIA e posterior avaliação. Com o intuito de validar a prova de conceito do sistema, foram delineados três objetivos principais: a construção de um banco de dados vetorial contendo planos de governo dos candidatos, o desenvolvimento de um mecanismo eficiente de recuperação de informações e a criação de uma interface interativa.

O primeiro objetivo envolveu a coleta e organização de dados de propostas políticas, permitindo a formação de um repositório robusto que serve como base para as consultas dos usuários. Em seguida, o mecanismo de recuperação foi projetado para otimizar a busca por informações relevantes, utilizando algoritmos avançados que garantem a precisão e a relevância dos dados apresentados. Por último, o sistema foi desenvolvido em uma plataforma amigável, possibilitando a interação fluida entre candidatos, eleitores e formuladores de políticas públicas.

O código aplicado no desenvolvimento do estudo encontra-se disponível em repositório anônimo [2]. O sistema desenvolvido para a submissão de artigos pode ser acessado [15], onde é possível utilizar o PalanqueIA. A seguir, segue o desenvolvimento detalhado de cada objetivo secundário do estudo:

#### 3.1 Desenvolvimento do banco de dados vetorial

O processo da criação do banco de dados de planos de governo foi projetado em cinco etapas sequenciais, desde a extração dos dados no sítio do TSE até a persistência em um banco de dados específico para fins vetoriais. A Figura 1 ilustra, em detalhes, o processo de construção e indexação do banco utilizado neste projeto, destacando os seus componentes.



**Figure 1: Fluxo das etapas na construção do banco de dados vetorial**

**3.1.1 Etapa 1: Extração dos Dados.** A primeira etapa consiste na coleta dos arquivos disponibilizados em formato .PDF, especificamente planos de governo de candidatos a cargos executivos nas eleições municipais de 2024. Esses documentos foram obtidos no sítio oficial do Tribunal Superior Eleitoral (TSE) [19]. Para a leitura dos arquivos, foi empregada a biblioteca PyPDF2 [16], que permitiu processar e armazenar os documentos como objetos estruturados. Na construção dos metadados, foram utilizados dados das candidaturas disponíveis na mesma plataforma do TSE.

**3.1.2 Etapa 2: Pré-processamento.** Ao longo do processo do pré-processamento foi constatada uma considerável diversidade de formatos, tipos e estruturas nos planos de governo disponíveis. Essa heterogeneidade nos arquivos representa um desafio significativo para a conformidade do banco de dados. Para abordar os arquivos em formato de imagem, foi empregada a biblioteca em Python (madmaze, 2024), que possibilitou a conversão das imagens em strings, na medida do possível. Aproximadamente 8% dos arquivos foram descartados devido a resultados insatisfatórios no tratamento de dados. A seguir, apresenta-se a figura 2 - um exemplo que ilustra a falta de padronização nos documentos mencionados.

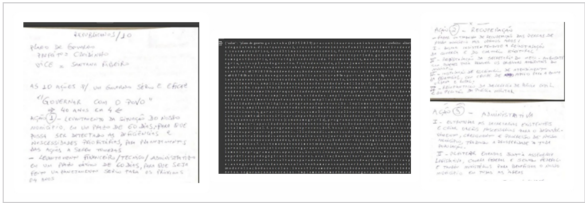


Figure 2: Exemplos de planos de governo disponibilizados

Na continuidade desta etapa, foram aplicadas técnicas de limpeza de dados com o intuito de padronizar e otimizar as informações para a criação de vetores. Entre as técnicas utilizadas, destacam-se a remoção de caracteres indesejados e a conversão do texto para letras minúsculas, entre outras. A granularidade de recuperação foi estabelecida no nível de parágrafo, permitindo que a resposta incluisse tanto o antecessor quanto o sucessor do parágrafo em questão.

**3.1.3 Etapa 3: Fragmentação.** A terceira etapa, conhecida como (*chunking*), consiste na divisão do texto extraído em segmentos menores, os quais são mais adequados para a análise de similaridade entre o texto da consulta e o banco de dados. Para essa divisão, foi estabelecido um tamanho de 256 tokens para cada fragmento, com a aplicação de um *overlay* de 20 tokens. O termo *overlay* refere-se ao método em que uma sobreposição de tokens é incorporada entre os fragmentos consecutivos, garantindo que haja um contexto compartilhado entre as divisões adjacentes, o que pode melhorar a precisão da comparação semântica e a continuidade da informação ao longo da fragmentação.

O framework empregado para o gerenciamento do banco de dados vetorial foi o ChromaDB. Conforme descrito em sua documentação [5], Chroma é um banco de dados vetorial de código aberto, nativo para inteligência artificial, que oferece ferramentas para o armazenamento de *embeddings* e metadados, bem como para a geração, consultas e pesquisa desses *embeddings*.

**3.1.4 Etapa 4: Incorporação.** Para a geração dos *embeddings*, utilizou-se o modelo sentence-transformers/all-MiniLM-L6-v2 [8], disponível na plataforma *Huggingface*. Nesta fase de incorporação dos dados, o modelo é responsável por gerar as representações vetoriais dos textos as quais serão armazenados no banco.

**3.1.5 Etapa 5: Indexação.** A etapa final consiste na persistência dos dados vetoriais gerados no banco. Durante esta fase, ocorre a indexação, onde os fragmentos enriquecidos junto aos metadados, como nome do candidato, partido e outras informações coletadas na etapa 1, são gravados de forma organizada no banco de dados objetivando performance nas consultas.

**3.1.6 Considerações finais do banco de dados.** Ao longo do processo de desenvolvimento do banco de dados, a base registrou

14.197 propostas políticas, fragmentadas em 1.768.177 *chunks*, conforme ilustrado na tabela 1.

Table 1: Quantidade de fragmentos por Unidade da Federação do candidato

Estados	Fragmentos	Estados	Fragmentos
AC	7.754	MG	259.482
AL	26.054	MS	29.789
AM	24.479	MT	43.456
AP	6.949	PA	62.303
BA	173.226	PB	52.504
CE	62.539	PE	57.611
ES	39.795	PI	51.925
GO	62.610	PR	122.548
MA	69.118	RJ	64.371
RN	46.353	RO	16.830
RR	3.864	RS	100.370
SC	72.560	SP	256.654
SE	25.999	TO	29.034
<b>TOTAL</b>	<b>1.768.177</b>		

Conforme se observa nas nuvens de palavras a seguir (figuras 3 e 4), existe a predominância de alguns assuntos em detrimento de outros, tais como educação, saúde e segurança. As nuvens de palavras foram construídas utilizando as bibliotecas *nltk* e *wordcloud*. Além das palavras stopwords padrão do *nltk*, foi acrescentada à lista: a, o, é, em, de, ao, e, da, que, plano, governo, municipal, município, políticas, programa, programas, públicos, público e pública.



Figure 3: Nuvem de palavras por frequência



Figure 4: Nuvem das 200 palavras mais relevantes

Por fim, na figura 5, apresentamos a distribuição de frequência da quantidade de palavras em cada fragmento. O valor médio é de 31,4 palavras e o desvio-padrão de 9,3 palavras. O coeficiente de variação de 29,62% indica que a dispersão dos dados em relação à média é considerável. Com o objetivo de superar essa variação, o mecanismo de recuperação busca fragmentos vizinhos para compor o documento relevante de recuperação final.

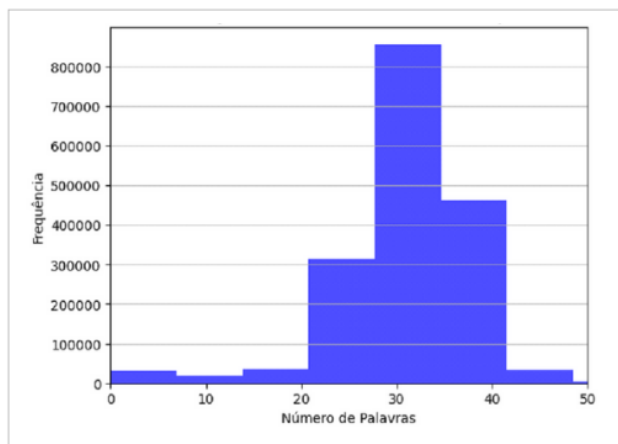


Figure 5: Histograma da quantidade de palavras por fragmento

### 3.2 Mecanismo de Recuperação da Informação

O mecanismo de recuperação da informação abrange todo o processo desde a entrada da consulta pelo usuário na interface até o retorno da resposta a ele. Esse processo pode ser segmentado em seis etapas principais, conforme ilustrado na Figura 6.

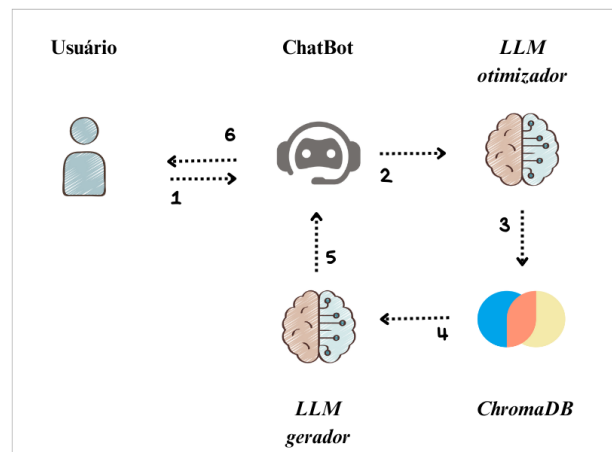


Figure 6: Engenharia de Dados do PalanqueIA

**3.2.1 Etapa 1.** O fluxo de dados se inicia a partir de uma consulta do usuário interessado em acessar propostas de governo acerca de um tema específico ou de algum candidato em especial. A etapa inicial consiste no envio e recebimento da informação da consulta inicial.

**3.2.2 Etapa 2.** O propósito dessa etapa é otimizar a consulta inicial do usuário a fim de recuperar informações realmente relevantes do banco de dados. Um dos principais desafios do RAG é sua dependência com a consulta original do usuário. É comum termos formulações imprecisas, imprudentes e de difícil entendimento, além dos problemas de vocabulários, regionalismos e outras dificuldades linguísticas e semânticas [10].

Para contornar essa problemática, executamos um processo de melhoramento na consulta inicial. Esse processo é realizado em fases conforme figura 7:

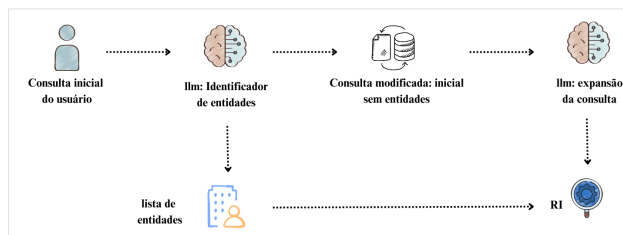


Figure 7: Processo de otimização da consulta inicial

No contexto da otimização da consulta inicial, a primeira fase se resume a identificação de entidades relacionadas. O modelo utilizado foi o **FacebookAI/xlm-roberta-large-finetuned-conll03-english** disponibilizado na plataforma *Huggingface*. O objetivo dessa separação é conferir maior relevância às entidades no processo de recuperação, além de evitar que o modelo de otimização da consulta gere alucinações ou informações baseadas em entidades. Isso se deve ao fato de que, conforme observações de testes locais,

a presença de entidades no prompt aumenta significativamente a probabilidade de alucinações [13].

Após a identificação e separação das entidades, a função retorna dois objetos: a lista de entidades extraídas da consulta e a consulta modificada, na qual as entidades foram removidas. A lista de entidades é armazenada para ser posteriormente anexada à resposta gerada pelo LLM responsável pela expansão da consulta, enquanto a consulta modificada, sem as entidades, é encaminhada ao modelo **Meta Llama 3.1 405B**, utilizando os seguintes parâmetros:

**System content:** "Você é um candidato político." **User content:** "Retorne em poucas palavras 3 propostas diferentes de acordo com o seguinte pedido" + consulta inicial sem entidades.

Por fim, o sistema agrega a resposta da LLM **Meta Llama 3.1 405B**, responsável pela expansão, junto à lista de entidades para compor o texto final a ser realmente encaminhado ao recuperador de informação.

**3.2.3 Etapa 3.** Nesta etapa, a consulta otimizada é enviada para o Chroma com o objetivo de buscar textos similares, isto é, e retornar documentos relevantes para a resposta final. Esta etapa é nevrálgica para o processo de RAG. Pode ser dividida em três fases: geração de *embeddings*, consulta ao ChromaDB e a busca dos fragmentos vizinhos.

- **Geração de embeddings:** Nesta fase, a consulta otimizada é transformada em um vetor de *embeddings* – representações numéricas da consulta que capturam o significado semântico. O modelo de vetorização utilizado é o **sentence-transformers/all-MiniLM-L6-v2**.
- **Consulta ao ChromaDB:** Consiste em calcular as distâncias vetoriais entre os *embeddings* provenientes da consulta otimizada e os vetores disponíveis no banco de dados. Objetivando encontrar documentos semanticamente semelhantes, utilizamos uma busca por similaridade utilizando a distância baseada na similaridade do cosseno. Nesta métrica, A e B são vetores no espaço n-dimensional que representam entidades ou objetos, conforme equação abaixo:

$$d = 1.0 - \frac{\sum(A_i^2) \cdot \sum(B_i^2)}{\sum(A_i \times B_i)} \quad (1)$$

- **Busca dos fragmentos vizinhos:** Após a recuperação dos fragmentos mais similares, retornamos à base para recuperar os fragmentos vizinhos do principal. Para evitar que assuntos de outras propostas sejam anexados ao documento principal recuperado, é realizado um segundo cálculo de similaridade, desta vez entre os vizinhos e o fragmento principal, com o **corte de d < 0,30 para validação**, como critério de inclusão do vizinho ao fragmento principal.

A partir de testes locais de qualidade de resposta, um filtro na distância mínima foi estipulado para o fragmento ser classificado como similar ou não similar. O corte definido foi de 0 a 0,30 para o fragmento ser considerado relevante. Para ser realizado este teste, foram considerados 100 prompts com temas relevantes e 100 prompts com assuntos aleatórios.

**3.2.4 Etapa 4.** Esta etapa consiste em gerenciar o encaminhamento das informações recuperadas aos LLMs objetivando a geração das respostas finais aos usuários. Para a execução do resumo e geração

das informações levantadas, foi utilizado o modelo **Meta Llama 3.1 405B**.

Com o propósito de fornecer ao usuário tanto uma consulta a projetos específicos de candidatos em outras eleições quanto sugestão de novas idéias de projetos, no presente estudo, foram desenvolvidos dois modos de geração de conteúdo ou de funcionamento do sistema: modo "Pesquisa" e o modo "Criação".

O modo "Pesquisa" retorna ao usuário cinco projetos reais de planos de governos de candidatos mais similares à consulta inicial, enquanto o modo "Criação" retorna uma proposta estruturada de um projeto político baseado na consulta inicial e em outros planos de governo semelhantes. Segue as especificações abaixo:

**Table 2: Especificações do modo "Pesquisa"**

Parâmetro	Valor
System content	"Você é um assistente que resume em poucas palavras de maneira factual, com foco em:"
User content	Textos extraídos provenientes da etapa 3 dos cinco (5) documentos mais relevantes extraídos via RAG
Temperature	0.20
top-p	0.70

**Table 3: Especificações do modo "Criação"**

Parâmetro	Valor
System content	"Você é um candidato político e deve criar uma proposta de governo, seja criativo e realista, com foco apenas em: "
User content	Textos extraídos provenientes da etapa 3 dos quinze (15) documentos mais relevantes extraídos via RAG
Temperature	1.50
top-p	0.95

No contexto dos LLMs [11], os parâmetros *top-p* e *temperature* controlam a aleatoriedade e a criatividade da geração de texto. O *top-p* (ou *nucleus sampling*) limita a seleção de tokens àqueles cujas probabilidades acumuladas atingem um valor p, restringindo ou ampliando as opções de palavras que o modelo considera, o que afeta a diversidade das respostas geradas. Já a *temperature* ajusta a distribuição de probabilidades, onde valores mais altos (acima de 1) tornam as escolhas mais aleatórias e criativas, enquanto valores baixos (abaixo de 1) tornam as respostas mais previsíveis e focadas nas palavras de maior probabilidade. Juntos, esses parâmetros permitem controlar o equilíbrio entre criatividade, diversidade e coerência nas respostas geradas pelo modelo.



**3.2.5 Etapa 5.** Essa etapa se resume a coletar a resposta do LLM gerador e encaminhar ao agente conversacional autônomo. Para o modo "Pesquisa", são apresentadas as cinco propostas mais similares recuperadas e processadas. Para o modo "Criação", é apresentada uma proposta inovadora baseada em quinze (15) propostas recuperadas.

**3.2.6 Etapa 6.** Por fim, nessa etapa, o agente prepara a resposta final e a envia ao usuário, e encerra a atividade sem guardar memória.

## 4 Arquitetura Modular do Sistema

A seção a seguir apresenta uma descrição sucinta dos principais componentes do PalanqueIA. O sistema é estruturado em três módulos principais: *Front-End*, que corresponde à interface de interação com o usuário; *Back-End*, responsável pelo processamento e gestão das funcionalidades internas; e os *Endpoints*, que consistem em serviços acessados por meio de requisições, conforme ilustrado na Figura 8.

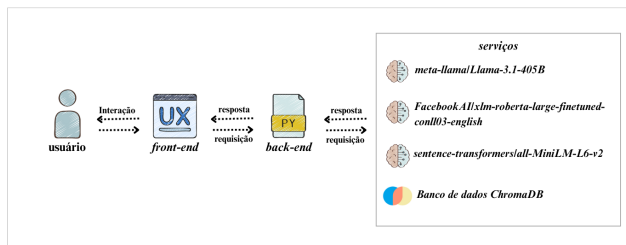


Figure 8: Arquitetura modular do PalanqueIA

### 4.1 Front-end

A camada de *Front-end* corresponde à interface de apresentação do sistema, concebida para proporcionar uma experiência de usuário intuitiva e interativa. Sua implementação foi realizada utilizando as tecnologias HTML, CSS e JavaScript, resultando em um aplicativo de página única com uma interface de bate-papo que permite a interação contínua entre o usuário e o sistema. Esse design promove uma experiência fluida e responsiva, garantindo a simplicidade no uso das funcionalidades do sistema. Este módulo é responsável por capturar as consultas realizadas pelo usuário, se comunicar com o módulo de processamento e gestão, e por fim, exibir as respostas recebidas, facilitando a interação com funcionalidades complexas. O ChatBot inicia a interação com o seguinte texto:

"Olá! Sou sua assistente virtual especializada em busca de propostas políticas. Estou aqui para ajudá-lo a explorar e entender diferentes iniciativas governamentais e políticas públicas, desde planos de incentivo à saúde até estratégias de gestão de recursos públicos. Se desejar buscar por planos de governos ou ver propostas inovadoras de determinado tema, é só pesquisar! Vamos começar?"

A Figura 9 ilustra a interface do usuário do sistema, destacando os principais elementos visuais e funcionalidades disponíveis.



Figure 9: Interface de Usuário do PalanqueIA

A interface do usuário do sistema foi projetada para oferecer funcionalidades que facilitam a interação e personalização das operações realizadas. Entre os recursos disponíveis, destacam-se:

- **Modo de funcionamento:** Botões de opção única para definir o modo de geração de conteúdo do PalanqueIA: **PESQUISA** ou **CRIAÇÃO**;
- **Novo Chat:** abre uma nova janela;
- **Exemplos Práticos:** fornece alguns *prompts* exemplificativos para auxiliar na consulta do usuário.

### 4.2 Back-end

O módulo *Back-end* do PalanqueIA atua como o núcleo de processamento do sistema, intermediando a comunicação entre o *Front-end*, os *Endpoints* e o banco de dados vetorial. Desenvolvido com o framework Python FastAPI, este componente foi projetado para operar de forma assíncrona, assegurando um alto desempenho e respostas eficientes às interações realizadas pelos usuários.

As principais responsabilidades do *Back-end* incluem:

- **Processamento da consulta inicial:** Recebe as consultas realizadas pelo usuário e realiza uma otimização textual com o objetivo de maximizar a precisão das respostas geradas pelos LLMs.
- **Encaminhamento ao Mecanismo de RI:** Encaminhar a consulta otimizada ao mecanismo de RI, que retorna documentos relevantes para o contexto da consulta.
- **Geração de respostas contextualizadas:** Transmite a consulta e os documentos recuperados para um modelo LLM, responsável por gerar respostas precisas e contextualizadas com base no material fornecido.
- **Comunicação com o Front-end:** Envia a resposta processada para ser apresentada ao usuário de forma clara e intuitiva.

### 4.3 Endpoints

Os *endpoints* são componentes fundamentais do sistema, responsáveis por permitir o acesso aos LLMs e ao banco de dados vetorial. Esses *endpoints* suportam uma variedade de tarefas, incluindo geração de texto, resumo, inferência e recuperação de informações. Os modelos utilizados estão hospedados na plataforma *HuggingFace* e são detalhados a seguir:

- **meta-llama/Llama-3.1-405B** [14]: A coleção Meta Llama 3.1 de modelos multilíngues de linguagem grande é uma coleção de modelos generativos pré-treinados e ajustados para instruções nos tamanhos 8B, 70B e 405B (entrada de texto/saída de texto). Os modelos somente de texto ajustado para instruções Llama 3.1 (8B, 70B, 405B) são otimizados para casos de uso de diálogo multilíngue e superam muitos dos modelos de software livre e bate-papo fechado disponíveis em benchmarks comuns do setor.
- **FacebookAI/xlm-roberta-large-finetuned-conll03-english** [9]: O modelo XLM-RoBERTa foi proposto em Unsupervised Cross-lingual Representation Learning at Scale. É baseado no modelo RoBERTa do Facebook lançado em 2019. É um grande modelo de linguagem multilíngue, treinado em 2,5 TB de dados CommonCrawl filtrados. Este modelo é XLM-RoBERTa-large ajustado com o conjunto de dados conll2003 em inglês.
- **Sentence-transformers/all-MiniLM-L6-v2** [18]: O modelo destina-se a ser usado como um codificador de frases e parágrafos curtos. Dado um texto de entrada, ele gera um vetor que captura a informação semântica. O vetor de sentença pode ser usado para tarefas de recuperação de informações, agrupamento ou similaridade de sentenças.
- **ChromaDB**: Banco de dados disponível via API da própria biblioteca Python.

## 5 Resultados e Discussão

Para avaliar a performance do PalanqueIA, foram realizados testes com diferentes consultas, abrangendo os modos "Pesquisa" e "Criação". Os resultados obtidos demonstram a capacidade do sistema em fornecer informações relevantes e úteis sobre planos de governo, além de gerar propostas inovadoras.

### 5.1 Respostas no modo "Pesquisa"

**5.1.1 Consulta do usuário: "Retorne projetos na área de oncologia"**. A tabela 2 apresenta o resultado do PalanqueIA ao fornecer como entrada no chat a consulta acima em contraste com a saída do ChatGPT-4 utilizando a mesma consulta, acrescida do texto "de candidatos políticos da eleição de 2024".

O ChatGPT-4, ao receber a mesma consulta, apresentou uma resposta mais genérica sobre propostas para a área de oncologia em eleições municipais, sem mencionar candidatos ou cidades específicas. O PalanqueIA, por sua vez, forneceu informações precisas sobre projetos de candidatos, incluindo detalhes sobre as ações propostas. Essa diferença destaca a capacidade do PalanqueIA de fornecer respostas personalizadas e focadas no domínio político, proporcionado pelo uso do mecanismo de recuperação da informação implementado.

**5.1.2 Consulta do usuário: "traga iniciativas sobre o tema combate à criminalidade"**. De forma semelhante à tabela 2, a tabela 3 apresenta o resultado do PalanqueIA ao fornecer como entrada no chat a consulta acima em contraste com a saída do ChatGPT-4 utilizando a mesma consulta, acrescida do texto "de candidatos políticos da eleição de 2024".

**Table 4: Respostas do PalanqueIA e ChatGPT-4 sobre propostas na área de oncologia**

Candidato (Cidade/Estado)	Propostas
Sebastião Rogatti (Nova América da Colina-PR)	- Expansão e ampliação do programa de prevenção ao câncer do colo de útero. - Criação do programa de prevenção ao câncer de próstata.
Átila José Pizarro Carvalho (Gurinhata-MG)	- Assistência e manutenção em casa de apoio em Barretos para pessoas em tratamento de câncer e seus acompanhantes.
Saída do ChatGPT-4	- Ampliação de serviços de diagnóstico precoce, como criação de centros especializados e mutirões para exames preventivos. - Investimento em infraestrutura hospitalar, incluindo construção e modernização de unidades de tratamento oncológico. - Apoio a pacientes em tratamento com transporte, medicamentos e suporte psicológico. - Campanhas educativas para conscientização sobre prevenção e detecção precoce do câncer.

Assim como na consulta anterior, o PalanqueIA apresentou respostas específicas, com propostas de candidatos reais. Essa capacidade de recuperar informações detalhadas de planos de governo demonstra a eficácia do sistema em auxiliar eleitores e analistas políticos na busca por propostas concretas.

### 5.2 Respostas no modo "Criação"

**5.2.1 Consulta do usuário: "Forneça uma proposta de governo sobre o assunto saúde do idoso"**.

===== Início da Resposta =====

Segue aqui uma proposta de governo para a saúde do idoso:

Proposta: "Saúde do Idoso: Atenção Integral e Humanizada"

Objetivo: Garantir o acesso a serviços de saúde de qualidade, humanizados e especializados para a população idosa, promovendo a prevenção, o tratamento e a reabilitação de doenças crônicas e condições específicas da idade avançada.

Eixos de Ação:

1. Atenção Primária: Fortalecer a atenção primária à saúde, com equipes de saúde da família treinadas para atender às necessidades específicas da população idosa, incluindo a prevenção e o tratamento de doenças crônicas. 2. Especialidades Médicas: Garantir o acesso a especialidades médicas, como geriatria, nefrologia, cardiologia e neurologia, para atender às necessidades específicas da população idosa. 3. Reabilitação e Fisioterapia: Oferecer serviços de



Table 5: Combate à criminalidade: PalanqueIA e ChatGPT-4

Candidato (Cidade/Estado)	Propostas
Pedro Antônio Pereira Malheiros (Sebastião Laranjeiras-BA)	<ul style="list-style-type: none"><li>- Fortalecer políticas de segurança com mais presença policial e tecnologias de vigilância.</li><li>- Implementar programas comunitários de prevenção ao crime, focando em educação e inclusão social de jovens.</li><li>- Objetivo: melhorar a segurança e a qualidade de vida dos cidadãos.</li></ul>
Patricia Lopes de Assis (Bom Jesus da Lapa-BA)	<ul style="list-style-type: none"><li>- Fortalecimento da segurança pública com mais presença policial e tecnologias de vigilância.</li><li>- Programa de prevenção ao crime com foco em educação e inclusão social dos jovens.</li><li>- Objetivo: Melhorar a segurança e a qualidade de vida dos cidadãos.</li></ul>
Saída do ChatGPT-4	<ul style="list-style-type: none"><li>- Fortalecimento da guarda municipal com aumento do efetivo, capacitação e uso de tecnologia.</li><li>- Monitoramento por câmeras com expansão de sistemas de reconhecimento facial.</li><li>- Melhoria da iluminação pública em áreas com alta incidência criminal.</li><li>- Programas sociais de inclusão para jovens em situação de vulnerabilidade.</li><li>- Parcerias com forças estaduais para operações conjuntas com a polícia militar e civil.</li><li>- Medidas focadas na prevenção e resposta ágil à violência.</li></ul>

reabilitação e fisioterapia para ajudar a população idosa a manter a mobilidade e a independência. 4. Saúde Mental: Promover a saúde mental da população idosa, com ações de prevenção, diagnóstico e tratamento de condições como depressão, ansiedade e doenças neurodegenerativas. 5. Assistência Domiciliar: Oferecer assistência domiciliar para a população idosa que necessita de cuidados especiais, incluindo a entrega de medicamentos e o acompanhamento médico. 6. Educação e Prevenção: Promover a educação e a prevenção sobre doenças crônicas e condições específicas da idade avançada, com ações de saúde pública e campanhas de conscientização. 7. Parcerias e Redes: Estabelecer parcerias com instituições de saúde, hospitais e organizações não governamentais para garantir a articulação e a integração dos serviços de saúde para a população idosa.

Metas:

- Reduzir a taxa de mortalidade por doenças crônicas em 20% em 4 anos.
- Aumentar a cobertura de atenção primária à saúde para 90% da população idosa em 4 anos.
- Garantir o acesso a especialidades médicas para 80% da população idosa em 4 anos.
- Reduzir a taxa de hospitalização por condições evitáveis em 30% em 4 anos.

Orçamento: R\$ 10 milhões anuais, para serem destinados à implementação das ações e metas propostas.

Essa proposta de governo visa garantir a saúde e o bem-estar da população idosa, promovendo a prevenção, o tratamento e a reabilitação de doenças crônicas e condições específicas da idade avançada. Além disso, busca fortalecer a atenção primária à saúde, garantir o acesso a especialidades médicas e promover a saúde mental da população idosa.

===== Fim da Resposta =====

A proposta gerada pelo PalanqueIA no modo "Criação" demonstra a capacidade do sistema em formular propostas inovadoras e abrangentes, com base em informações existentes em sua base de dados. A proposta para a saúde do idoso apresenta eixos de ação, metas e orçamento, evidenciando a capacidade do PalanqueIA em auxiliar candidatos na elaboração de planos de governo completos e bem estruturados.

O tempo de resposta do PalanqueIA, variando entre 18 e 22 segundos para as consultas testadas, demonstra a eficiência do sistema em recuperar informações e gerar propostas em um tempo razoável. Se comparado com a leitura manual dos 14.197 planos de governo, a uma taxa estimada de 10 segundos por fragmento, o PalanqueIA proporciona uma economia significativa de tempo, equivalente a 4.911 horas.

Por meio da leitura das respostas acima entregues pelo sistema, conclui-se que é possível obter informações úteis e satisfatórias, em se tratando de planos de governo a partir de um chatbot, utilizando TRI e LLMs.

Em relação à comparação do resultado obtido entre o PalanqueIA e o ChatGPT-4, é possível observar que, para certos domínios, os chatbots desenvolvidos utilizando RAG são mais específicos em suas respostas, ressaltando o potencial do uso da técnica para assistentes especializados em assuntos de um domínio particular.

6 Conclusão

O estudo demonstrou a viabilidade do uso de técnicas de RAG para criar um sistema inteligente de recuperação e proposição de planos de governo. O PalanqueIA, um chatbot desenvolvido como prova de conceito, demonstrou a capacidade de:

- Aperfeiçoar a consulta inicial do usuário através de um mecanismo inovador de reescrita da consulta inicial, utilizando o reconhecimento de entidades;
- Consultar uma base de dados vetorial com mais de 1,7 milhão de fragmentos de propostas políticas;
- No modo de Pesquisa, fornecer respostas relevantes e contextualizadas às consultas dos usuários;
- No modo de Criação, gerar propostas de Governo criativas a partir de outras pré-existentes.

O estudo evidencia o potencial do RAG para aprimorar o processo eleitoral. O PalanqueIA possui pontencial para auxiliar candidatos na formulação de propostas políticas inovadoras e embasadas em

dados; facilitar a análise comparativa de planos de governo por eleitores e analistas políticos; e promover um debate público mais qualificado e focado em propostas concretas.

A pesquisa contribui para o avanço do campo da Ciência Política Computacional ao aplicar técnicas de Inteligência Artificial para democratizar o acesso à informação e fomentar a participação cidadã.

Este estudo se conecta a múltiplos tópicos de interesse da área de Sistemas de Informação. Primeiramente, ele se encaixa no tópico "Sistemas de informação e o mundo aberto", pois o PalanqueIA, acessível online, visa democratizar o acesso à informação sobre planos de governo. A pesquisa também aborda a "Visão sociotécnica de sistemas de informação", considerando tanto os aspectos técnicos da implementação do PalanqueIA com RAG e LLMs quanto os impactos sociais da ferramenta na participação cidadã e na qualidade do debate público. O processo de desenvolvimento do PalanqueIA, detalhado nas etapas de construção do banco de dados vetorial, do mecanismo de recuperação da informação e da interface do chatbot, ilustra a aplicação de "Paradigmas, modelagem, design, engenharia e avaliação de sistemas de informação".

A pesquisa se destaca pela aplicação de "Tecnologias emergentes aplicadas a sistemas de informação", utilizando RAG e LLMs para a análise e proposição de políticas públicas. As questões éticas relacionadas ao uso de dados de planos de governo são abordadas, garantindo transparência e responsabilidade, o que se relaciona ao tópico "Ética, cultura, legislação e política em sistemas de informação". O PalanqueIA, inserido no ecossistema digital do governo eletrônico, conecta candidatos, eleitores e formuladores de políticas, abordando o tema "Ecossistemas digitais e de negócios". Finalmente, a utilização de LLMs e técnicas de PLN no desenvolvimento do chatbot enquadra a pesquisa no tópico "Sistemas de informação e Inteligência Artificial (Generativa, LLM, PLN, entre outros)".

Por fim, este estudo se alinha com os Grandes Desafios de Pesquisa em Sistemas de Informação no Brasil para a Década de 2016-2026, em particular com os desafios Sistemas de Informação e o Mundo Aberto e Metodologias e Tecnologias para a participação cidadã. O PalanqueIA, ao utilizar tecnologias de ponta como RAG e LLMs para democratizar o acesso à informação e promover a participação cidadã, contribui para a criação de um ecossistema digital mais transparente e colaborativo, em consonância com os princípios do GrandSI-BR.

Sugestões para trabalhos futuros:

- Expandir a base de dados com planos de governo de outros anos e cargos eletivos;
- Avaliar a usabilidade e a satisfação dos usuários com o PalanqueIA;
- Investigar o impacto do sistema no debate público e na qualidade das propostas políticas;
- Criar uma proposta política completa baseada em modelos de gestão de Projetos.

## References

- [1] Ana Paula Albarelli. 2020. *Uma análise da descortesia como estratégia de persuasão em interações polêmicas: o debate político*. Ph.D. Dissertation. Universidade de São Paulo. <https://doi.org/10.11606/T.8.2020.tde-18082020-170840>
- [2] Anon. 2024. Model Study. <https://anonymous.4open.science/r/ModelStudy-C21B/>. Accessed: 2024-11-04.
- [3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology Behind Search* (second ed.). Addison Wesley, New York.
- [4] Brasil. 2009. Lei nº 12.034, de 29 de setembro de 2009. Institui modificações no Código Eleitoral e na Lei dos Partidos Políticos e estabelece normas para o voto eletrônico. Diário Oficial da União: seção 1, Brasília, DF, 30 set. 2009.
- [5] ChromaDB. 2024. Chroma Docs. <https://docs.trychroma.com/>. Acesso em: [data de acesso].
- [6] Câmara dos Deputados. 2024. *Projeto prevê ampla divulgação de programas de governo de candidatos*. <https://www.camara.leg.br/noticias/788631-projeto-preve-ampla-divulgacao-de-programas-de-governo-de-candidatos/> Acesso em: 24 set. 2024.
- [7] Simone Diniz and Lucy Oliveira. 2020. Programas de Governo e Promessas de Campanha. *Revista de Sociologia e Política* 28, 76 (2020), e006. <https://doi.org/10.1590/1678-987320287606>
- [8] Hugging Face. 2024. sentence-transformers/all-MiniLM-L6-v2. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>. Acesso em: [data de acesso].
- [9] Facebook. 2024. XLM-RoBERTa Large Fine-Tuned on CoNLL-03 (English). <https://huggingface.co/FacebookAI/xlm-roberta-large-finetuned-conll03-english> Accessed: 2024-11-01.
- [10] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023). <https://doi.org/10.48550/ARXIV.2312.10997>
- [11] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The Curious Case of Neural Text Degeneration. *arXiv* (2019). <https://doi.org/10.48550/ARXIV.1904.09751>
- [12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401* (2020). <https://doi.org/10.48550/ARXIV.2005.11401>
- [13] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting for Retrieval-Augmented Large Language Models. *arXiv* (2023). <https://doi.org/10.48550/ARXIV.2305.14283>
- [14] Meta. 2024. Llama 3.1 - 405B. <https://huggingface.co/meta-llama/Llama-3.1-405B> Accessed: 2024-11-01.
- [15] Palanqueia. [n.d.]. Palanqueia: O que é e como funciona? <https://www.palanqueia.site/> Accessed: 2024-11-04.
- [16] PyPDF. 2024. Welcome to PyPDF2. <https://pypdf2.readthedocs.io/en/3.x/>. Acesso em: [data de acesso].
- [17] Mujahid Ali Quidwai and Alessandro Lagana. 2024. A RAG Chatbot for Precision Medicine of Multiple Myeloma. <https://doi.org/10.1101/2024.03.14.24304293> Preprint.
- [18] Sentence-Transformers. 2024. all-MiniLM-L6-v2. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2> Accessed: 2024-11-01.
- [19] Tribunal Superior Eleitoral. 2024. Divulgação de Candidaturas e Contas Eleitorais. <https://divulgacandcontas.tse.jus.br/>. Acesso em: 24 set. 2024.