SBC Computing Reviews, 202X, XX:1
ISSN: 2966-3938 • doi: 10.5753/rocs.202X.XXXXXX
This work is licensed under a Creative Commons Attribution 4.0 International License.

Análise de Hiperparâmetros em Sistemas de Geração Aumentada por Recuperação Aplicado ao Acervo do MPES

Hyperparameter Analysis in Retrieval-Augmented Generation (RAG) Systems Applied to the MPES Corpus

Heitor Coitinho Quartezani [Departamento de Estatística |Universidade Federal do Espírito Santo (UFES)]
Diego Roberto Colombo Dias [Departamento de Estatística |Universidade Federal do Espírito Santo (UFES)]

Departamento de Estatística, Centro de Ciências Exatas, UFES, Vitória, ES, Brasil.

Resumo. Este projeto de pesquisa propõe uma metodologia fundamentada em *Design of Experiments* (DoE) para analisar e otimizar os hiperparâmetros de um sistema de *Retrieval-Augmented Generation* (RAG) aplicado ao acervo normativo do Ministério Público do Estado do Espírito Santo (MPES). O objetivo central é identificar o impacto estatisticamente significativo de fatores críticos, como a estratégia de segmentação de documentos (*chunking*), o volume de contexto recuperado (k) e a escolha do *Large Language Model* (LLM) — na qualidade das respostas geradas. A avaliação utiliza o paradigma *LLM-as-a-Judge* com o suporte do *framework Ragas*. Devido à natureza não-paramétrica dos dados, a análise estatística é conduzida via Transformação de Postos Alinhados (*Aligned Rank Transform* - ART). Os resultados revelam que, embora o modelo de linguagem seja o fator de maior peso na precisão, a interação entre o tamanho do *chunk* e a técnica de busca desempenha um papel crucial na mitigação de alucinações, indicando que configurações maiores de k nem sempre resultam em ganhos marginais de fidelidade.

Abstract. This research project proposes a *Design of Experiments* (DoE) methodology to analyze and optimize the hyperparameters of a *Retrieval-Augmented Generation* (RAG) system applied to the document corpus of the Public Prosecutor's Office of the State of Espírito Santo (MPES). The main objective is to identify which factors, such as the document *chunking* strategy, the number of retrieved documents (k), and the *Large Language Model* (LLM) used, have a statistically significant impact on the system's response quality. Quality will be measured automatically through the *LLM-as-a-Judge* paradigm, using metrics from the *Ragas* framework. The statistical analysis will be conducted using the *Aligned Rank Transform* (ART) technique, a robust non-parametric approach for factorial designs, to provide a quantitative guide for configuring more efficient RAG systems.

Palavras-chave: Geração Aumentada por Recuperação, Desenho de Experimentos, Otimização de Hiperparâmetros, LLM-as-a-Judge, Análise de Variância.

Keywords: Retrieval-Augmented Generation, Design of Experiments, Hyperparameter Optimization, LLM-as-a-Judge, Analysis of Variance.

Recebido/Received: DD Month YYYY • **Aceito/Accepted:** DD Month YYYY • **Publicado/Published:** DD Month YYYY

1 Introdução

Nos últimos anos, os *Large Language Models* (LLMs), ou Grandes Modelos de Linguagem, revolucionaram a interação humano-computador ao oferecer interfaces em linguagem natural capazes de sintetizar grandes volumes de informação [Gao *et al*., 2024] e [Wu *et al*., 2025]. No entanto, a utilidade dessas ferramentas é controversa, devido à tendência dos modelos em inventar informação que parece ser verdadeira [Ji *et al*., 2023].

O cenário que motivou este estudo originou-se de uma limitação operacional crítica no Ministério Público do Estado do Espírito Santo (MPES). Embora a instituição possuísse um vasto acervo de atos normativos e portarias digitalizados [Ministério Público do Estado do Espírito Santo, 2026], esse *corpus* carecia de qualquer mecanismo estruturado de indexação ou recuperação. Inexistia, até o momento do projeto, sequer um sistema básico de busca por palavras-chave. Consequentemente, o acesso à informação dependia exclusivamente do conhecimento tácito prévio, pois o usuário precisava saber antecipadamente em qual documento específico a norma ou fato residia para encontrá-la.

Diante dessa barreira de acessibilidade, onde o acervo existia mas era invisível para fins de consulta exploratória, a implementação de um sistema inteligente não surgiu apenas como uma modernização, mas como uma necessidade para desbloquear o valor informacional desse acervo latente. Para endereçar esse problema, optou-se pela arquitetura *Retrieval-Augmented Generation* (RAG), ou Geração Aumentada por Recuperação.

A relevância e a rápida evolução desta arquitetura são evidenciadas por revisões abrangentes da literatura, que a posicionam como a solução padrão para conectar LLMs a dados privados [Gupta *et al*., 2024; Wu *et al*., 2025]. Contudo, o desempenho de um sistema RAG é altamente sensível a hiperparâmetros como a estratégia de *chunking*, ou segmentação, e o número de documentos recuperados (k). A busca por configurações ótimas é um campo de pesquisa ativo, pois escolhas subótimas podem levar a respostas irrelevantes ou alucinações [Wang *et al*., 2024].

Este projeto propõe uma abordagem sistemática para configurar esse *pipeline* no MPES ao utilizar uma metodologia estatística robusta para transformar um acervo antes

inacessível em uma base de conhecimento consultável e confiável.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver e aplicar uma metodologia de *Design of Experiments* (DoE), ou Desenho de Experimentos, para analisar sistematicamente e otimizar os hiperparâmetros de um sistema RAG aplicado ao acervo do MPES. Busca-se identificar os fatores que exercem influência estatisticamente significativa na qualidade das respostas geradas.

Para alcançar o objetivo geral, os seguintes objetivos específicos serão perseguidos:

- Avaliar o impacto de quatro estratégias de *chunking* ao comparar abordagens recursivas e semânticas.
- Comparar a eficácia de métodos de recuperação vetorial, lexical via *Best Matching 25* (BM25) e híbrida via *Reciprocal Rank Fusion* (RRF), ou Fusão de Classificação Recíproca.
- Analisar o comportamento do sistema ao variar o número de documentos recuperados (k) e buscar o ponto de equilíbrio entre contexto e ruído.
- Quantificar o ganho de desempenho entre modelos de diferentes classes de complexidade, especificamente GPT-4o mini e GPT-4o.

1.2 Justificativa

A escolha da arquitetura RAG para resolver o problema de acessibilidade do MPES fundamenta-se na insuficiência das abordagens tradicionais. Dada a inexistência prévia de qualquer mecanismo de busca, a implementação de uma busca puramente lexical por palavras-chave resolveria apenas parcialmente o problema, pois falharia em capturar consultas conceituais ou semânticas típicas do domínio jurídico. Por outro lado, o uso direto de LLMs seria inviável devido à impossibilidade de treinar o modelo com dados privados em tempo real e ao risco inaceitável de alucinações [Lewis *et al.*, 2020]. O RAG oferece o equilíbrio necessário ao ancorar a geração em documentos reais do MPES e garantir rastreabilidade e precisão factual.

Entretanto, a literatura indica que a configuração desses sistemas é frequentemente realizada de maneira empírica, por tentativa e erro, carecendo de um guia sistemático para a escolha de hiperparâmetros. A ausência de uma análise quantitativa rigorosa dificulta a compreensão de quais hiperparâmetros realmente impactam a performance [Wang *et al.*, 2024]. A aplicação do DoE justifica-se por oferecer uma abordagem formal para substituir o empirismo por evidência estatística, o que permite identificar a configuração ótima que maximiza a fidelidade da informação entregue ao usuário [Montgomery, 2017].

Institucionalmente, o trabalho justifica-se por transformar um acervo documental estático e inacessível em uma ferramenta dinâmica de suporte à decisão.

2 Fundamentação Teórica

Este capítulo estabelece o arcabouço teórico do estudo e ancora-se na literatura clássica de *Natural Language Processing* (NLP) Jurafsky and Martin [2024] e de *Information Retrieval* (IR) Manning *et al.* [2008]. Para conferir rigor à análise dos hiperparâmetros, detalha-se a natureza probabilística dos modelos de linguagem, os algoritmos matemáticos de IR e o formalismo estatístico do DoE Montgomery [2017].

2.1 Modelos de Linguagem e a Arquitetura Transformer

A base de qualquer sistema RAG é o LLM. Conforme definido por Jurafsky and Martin [2024], para compreender as limitações que motivam este estudo, como a tendência à "alucinação"factual, é fundamental desmistificar o modelo como um agente cognitivo e analisá-lo sob a ótica de um processo estocástico de previsão de sequência suportado pela arquitetura *Transformer*.

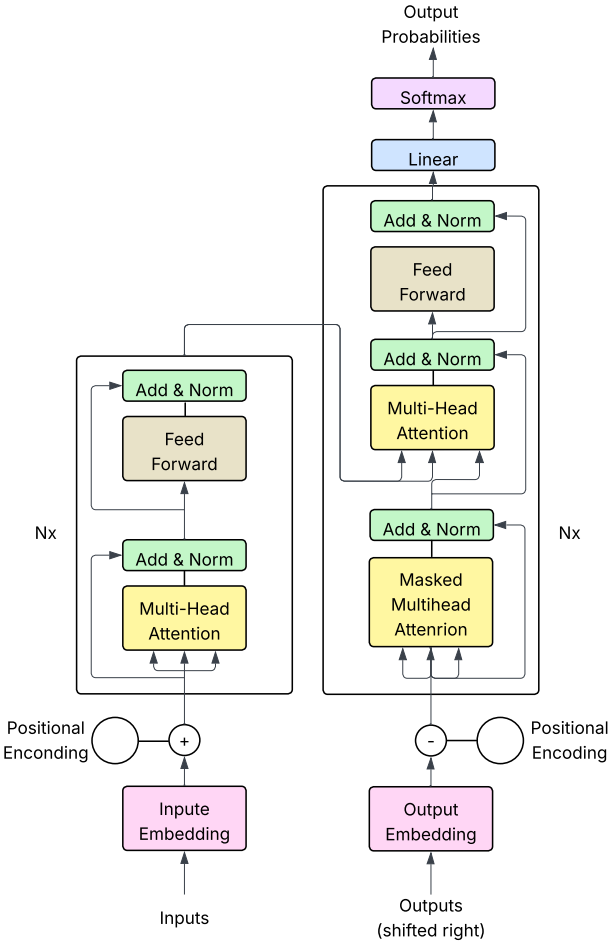

Figura 1. Arquitetura do Modelo Transformer baseado na figura de [Vaswani *et al.*, 2017].

A arquitetura do modelo *Transformer* fundamenta-se em uma estrutura de codificador (encoder) e decodificador (decoder), conforme ilustrado na Figura 1. O processo inicia-se com o tratamento das entradas por meio de camadas de Input e Output Embeddings , acompanhadas de Positional Encoding para preservar a informação sequencial. O núcleo do modelo é composto por múltiplos blocos repetidos (N×) que utilizam mecanismos de atenção de múltiplas cabeças (Multi-Head Attention) , incluindo uma versão com máscara no decodificador para prever tokens subsequentes. Cada subcamada é seguida por operações de normalização e conexão residual (Add & Norm) e redes neurais feed forward , culminando em camadas lineares e softmax para a geração das probabilidades

de saída.

2.1.1 O LLM como Estimador de Probabilidade

Essencialmente, um LLM é uma função paramétrica complexa P_θ treinada para estimar a distribuição de probabilidade conjunta de uma sequência de texto [Bengio *et al*., 2003]. O processo de geração, ou inferência, é uma aplicação iterativa da regra da cadeia da probabilidade, onde o modelo prevê o próximo *token* w_t (unidade mínima de texto, como uma sílaba ou palavra) condicionado a todo o histórico anterior $C = \{w_1, \ldots, w_{t-1}\}$:

$$w_t \sim P(w_t|C;\theta) \tag{1}$$

Onde θ representa os bilhões de parâmetros, ou pesos, da rede neural ajustados via gradiente descendente para maximizar a verossimilhança dos dados de treinamento.

Neste contexto, o fenômeno popularmente chamado de "alucinação"possui uma explicação probabilística: ocorre quando o modelo atribui massa de probabilidade elevada a uma sequência que é linguisticamente fluida e coerente com o contexto, ou seja, possui alta probabilidade sintática, mas que não corresponde a um fato real [Ji *et al*., 2023]. O sistema RAG atua justamente ao alterar o contexto C e injetar evidências externas para condicionar a probabilidade em direção a fatos recuperados.

2.1.2 O Mecanismo de Atenção

A capacidade dos LLMs modernos de processar contextos longos e entender nuances jurídicas deve-se à arquitetura *Transformer* introduzida por Vaswani *et al*. [2017]. Modelos anteriores, como as *Recurrent Neural Networks* (RNNs), processavam o texto sequencialmente, o que causava dificuldades na manutenção de dependências de longo prazo em parágrafos extensos, um problema inerente à natureza recursiva dessas redes [Jurafsky and Martin, 2024].

O *Transformer* resolve isso através do mecanismo de *Self-Attention*, ou Autoatenção. Este mecanismo permite que o modelo processe todos os *tokens* da entrada simultaneamente e aprenda a ponderar a importância de cada palavra em relação às outras. Para cada *token*, o modelo aprende três vetores:

- ***Query*** **(Q):** A "pergunta"que o *token* faz (ex: um verbo procurando seu sujeito).
- ***Key*** **(K):** A "etiqueta"que o *token* oferece para ser encontrado.
- ***Value*** **(V):** O conteúdo informacional do *token*.

O peso da atenção é calculado pela similaridade entre a Pergunta (Q) e as Chaves (K) de todos os outros *tokens*. Matematicamente, aplica-se o produto escalar normalizado seguido de uma função *Softmax*:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

O resultado é uma média ponderada dos valores V. Isso permite, por exemplo, que ao processar a palavra "Artigo"em um texto jurídico, o modelo atribua peso quase total ao número da lei subsequente e ignore ruídos intervenientes.

2.1.3 Controle de Entropia via Temperatura

A etapa final de um LLM converte os valores brutos da rede, os *logits* (z_i), em probabilidades normalizadas. Isso é feito através da função *Softmax* parametrizada por um hiperparâmetro crítico para este estudo: a **Temperatura (T)**.

$$P(w_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{3}$$

Segundo Holtzman *et al*. [2019], a temperatura atua como um regulador de entropia da distribuição:

- **Se $T \to 0$:** A distribuição torna-se um *argmax* determinístico. O modelo sempre escolhe o *token* mais provável. No contexto jurídico deste trabalho, fixou-se $T = 0.1$ para garantir reprodutibilidade e minimizar a criatividade, ou alucinação.
- **Se $T \to 1$:** A distribuição achata-se e permite que o modelo amostre *tokens* menos prováveis, o que gera textos mais diversos mas menos confiáveis.

2.2 Sistemas de Recuperação de Informação

A arquitetura RAG, formalizada por Lewis *et al*. [2020], depende intrinsecamente da capacidade de recuperar um subconjunto de documentos $Z \subset D$ que seja semanticamente relevante para a consulta do usuário q. A eficácia da resposta final do LLM é limitada superiormente pela qualidade desses documentos recuperados, conforme o princípio *Garbage In, Garbage Out* [Zhao *et al*., 2025]. Este trabalho avalia e combina dois paradigmas distintos de recuperação: a busca vetorial densa e a busca lexical esparsa.

2.2.1 Recuperação Vetorial e Embeddings

A recuperação vetorial fundamenta-se na *Distributional Hypothesis*, ou Hipótese Distribucional, de Harris [1954] ao mapear unidades textuais para vetores em um espaço latente contínuo $\mathbb{R}^n$. Para este estudo, utiliza-se o modelo *text-embedding-3-small* [OpenAI, 2024], que projeta textos em $n = 1536$ dimensões.

Geometricamente, o treinamento desses modelos busca otimizar a disposição espacial de tal forma que a proximidade angular reflita a similaridade semântica. Diferente da distância Euclidiana, que é sensível à magnitude e portanto ao tamanho do documento, utiliza-se a **Similaridade de Cosseno**, métrica padrão em IR [Manning *et al*., 2008] que mensura apenas o alinhamento de orientação entre os vetores:

$$sim(\vec{q}, \vec{d}) = \cos(\theta) = \frac{\vec{q} \cdot \vec{d}}{||\vec{q}|| \cdot ||\vec{d}||} \tag{4}$$

Esta abordagem captura sinonímia e relações conceituais, como aproximar "sanção"de "pena", e supera a necessidade de correspondência exata de palavras.

2.2.2 Recuperação Lexical Probabilística

Apesar da robustez semântica dos vetores, eles podem falhar em capturar termos exatos cruciais no domínio jurídico, como o número de uma lei ("8.666") ou um nome próprio. Para isso, utiliza-se a recuperação lexical.

A base teórica é a estatística de frequência. Uma abordagem ingênua seria contar a Frequência do Termo (TF).

Contudo, palavras funcionais como "o", "de"ou "para"são extremamente frequentes e carregam pouca informação discriminativa. Para corrigir isso, a Teoria da Informação introduz a *Inverse Document Frequency* (IDF), ou Frequência Inversa de Documento, proposta originalmente por Sparck Jones [1972] para penalizar termos comuns e valorizar termos raros:

$$IDF(t) = \log\left(\frac{N - n(t) + 0.5}{n(t) + 0.5}\right) \tag{5}$$

Onde N é o total de documentos no *corpus* e $n(t)$ é o número de documentos contendo o termo t.

O algoritmo adotado, o *Okapi BM25* [Robertson *et al.*, 1995], é uma evolução robusta do TF-IDF que introduz saturação. Diferente da contagem simples onde a relevância cresce linearmente com a repetição da palavra, o BM25 aplica uma curva assintótica controlada pelo parâmetro k_1. Além disso, normaliza o *score* pelo tamanho do documento ($|d|$) em relação à média do *corpus* ($avgdl$) via parâmetro b:

$$Score(d,q) = \sum_{t \in q} IDF(t) \cdot \frac{f(t,d) \cdot (k_1 + 1)}{f(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{6}$$

Essa formulação probabilística justifica o uso do BM25 como o componente de precisão cirúrgica do sistema.

2.2.3 Fusão Híbrida de Rankings

A combinação das buscas Vetorial e Lexical não é trivial, pois suas distribuições de pontuação são incompatíveis. O Cosseno varia em $[-1, 1]$, enquanto o BM25 varia em $[0, \infty)$ sem limite superior definido.

Para unificar essas abordagens sem depender de normalizações sensíveis a *outliers*, utiliza-se o algoritmo RRF proposto por Cormack *et al.* [2009]. Este método é não-paramétrico e ignora os valores absolutos dos *scores* ao basear-se apenas na posição, ou posto, do documento em cada lista ordenada. O *score* final é dado pela soma harmônica dos inversos dos *rankings*:

$$RRFscore(d) = \sum_{j \in \mathcal{S}} \frac{1}{\eta + r_j(d)} \tag{7}$$

Onde $\mathcal{S}$ é o conjunto de sistemas (Vetorial, Lexical), $r_j(d)$ é a posição ordinal do documento d no sistema j, e η é uma constante de suavização definida como 60 neste trabalho. O RRF penaliza documentos que aparecem no topo de apenas uma lista e favorece fortemente aqueles consensualmente relevantes em ambos os métodos [Yu *et al.*, 2024].

2.3 Estratégias de Segmentação e Continuidade Semântica

A indexação vetorial exige a discretização de documentos longos em unidades menores, denominadas *chunks*. A definição da granularidade dessa discretização não é trivial e estabelece um *trade-off*, ou compromisso, entre a densidade de informação (sinal) e o ruído contextual.

Conforme Zhang *et al.* [2024a], a segmentação impacta a probabilidade de recuperação $P(Z|q)$ de duas formas antagônicas:

- **Granularidade Fina (*Small Chunks*):** Maximiza a similaridade de cosseno para fatos específicos, mas fragmenta unidades de raciocínio e rompe correferências e elipses necessárias para a compreensão do modelo.
- **Granularidade Grossa (*Large Chunks*):** Preserva a coerência global, mas dilui a representação vetorial e reduz a distinção entre tópicos vizinhos, pois o vetor torna-se uma média do documento e perde especificidade.

2.3.1 Segmentação Recursiva e Janelas Deslizantes

A abordagem heurística padrão, denominada Segmentação Recursiva, discretiza o texto baseando-se em delimitadores sintáticos como parágrafos ou quebras de linha e um limite fixo de caracteres L.

A principal limitação teórica deste método é a imposição de fronteiras arbitrárias que podem seccionar sentenças ou argumentos lógicos. Para mitigar a perda de informação nas bordas (*boundary loss*), aplica-se a técnica de *Overlap*, ou Janela Deslizante com Sobreposição. Seja C_i o i-ésimo *chunk* de comprimento L e O o tamanho da sobreposição, o início do *chunk* C_{i+1} é deslocado por $L - O$ em relação a C_i.

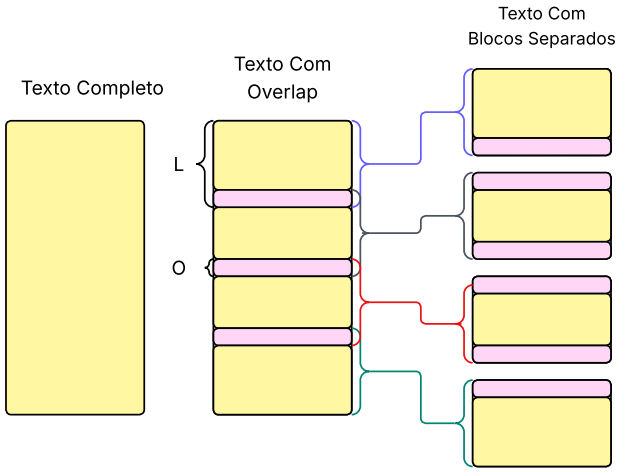

Figura 2. Segmentação com Janela Deslizante

A Figura 2 ilustra o processo de segmentação recursiva usando *Overlap*. O texto completo é separado em segmentos de tamanho L, havendo sobreposição entre os segmentos de tamanho O.

Essa redundância intencional ($O > 0$) garante a continuidade semântica e assegura que termos localizados no final de um segmento possuam contexto à direita na janela subsequente. Neste trabalho, avaliam-se janelas com O proporcional a 20% de L.

2.3.2 Segmentação Semântica

Em contraste com a heurística sintática, a Segmentação Semântica propõe uma abordagem orientada aos dados. O método baseia-se na premissa de que mudanças de tópico no texto correspondem a grandes distâncias angulares no espaço vetorial.

O algoritmo opera calculando a similaridade de cosseno $sim(s_t, s_{t+1})$ entre sentenças sequenciais. Formalmente, define-se um ponto de corte, ou fronteira do *chunk*, no índice t se a dissimilaridade exceder um limiar crítico τ:

$$1 - sim(s_t, s_{t+1}) > \tau \tag{8}$$

Onde τ é tipicamente definido como um percentil (ex: 95) da distribuição de distâncias do documento. Esta abordagem agrupa sentenças semanticamente coesas, minimiza a variância intra-chunk e maximiza a variância inter-chunk.

2.4 Avaliação Automatizada e Métricas

Para superar as limitações das métricas baseadas em n-gramas, como BLEU [Papineni *et al*., 2002] e ROUGE [Lin, 2004], este trabalho adota o paradigma *LLM-as-a-Judge* [Zheng *et al*., 2023] utilizando o *framework Ragas*. A avaliação combina métricas sem referência (*reference-free*), fundamentadas teoricamente por Es *et al*. [2023], e métricas baseadas em referência (*ground truth*), implementadas conforme a documentação técnica da biblioteca [Gradients, 2024].

2.4.1 Faithfulness (Fidelidade)

A métrica de *Faithfulness* mensura a consistência factual da resposta gerada em relação ao contexto recuperado. O objetivo é garantir que a resposta não contenha "alucinações", ou seja, informações que não podem ser deduzidas dos documentos fornecidos.

Conforme detalhado na documentação do *Ragas* [Exploding Gradients, 2024d], o cálculo é realizado em duas etapas sequenciais utilizando um LLM:

1. **Geração de Afirmações:** A resposta gerada (A) é decomposta em um conjunto de afirmações atômicas ($S = \{s_1, s_2, \ldots, s_n\}$). Isso isola cada proposição factual contida no texto.
2. **Verificação de Suporte:** Para cada afirmação s_i, o modelo verifica se ela pode ser inferida lógica e exclusivamente a partir do contexto recuperado (C).

O *score* de fidelidade é calculado pela razão entre o número de afirmações suportadas e o número total de afirmações extraídas:

$$Faithfulness = \frac{|V|}{|S|} \tag{9}$$

Onde $|V|$ representa a contagem de afirmações verificadas como verdadeiras com base no contexto e $|S|$ é o total de afirmações geradas.

2.4.2 Answer Relevancy (Relevância da Resposta)

A métrica de *Answer Relevancy* avalia a pertinência da resposta gerada em relação ao *input* do usuário. Conforme a documentação oficial do *Ragas* [Exploding Gradients, 2025], esta é uma métrica *reference-free* (sem gabarito) que não julga a veracidade factual, mas sim se a resposta endereça a pergunta de forma direta e apropriada, penalizando respostas incompletas ou que contenham detalhes redundantes.

O algoritmo opera sob a premissa de que, se uma resposta for verdadeiramente relevante, deve ser possível reconstruir a pergunta original a partir dela. O cálculo segue três etapas:

1. **Geração de Perguntas Artificiais:** Um LLM analisa a resposta gerada (A) e cria um conjunto de N perguntas prováveis (Q_{g_i}) que poderiam ter motivado aquela resposta. Por padrão, $N = 3$.
2. ***Embeddings*:** O sistema converte tanto a pergunta original do usuário (Q_o) quanto as perguntas geradas artificialmente (Q_{g_i}) em vetores de *embedding*.
3. **Cálculo de Similaridade:** Calcula-se a similaridade de cosseno entre o vetor da pergunta original (E_o) e os vetores das perguntas geradas (E_{g_i}).

O *score* final é a média aritmética dessas similaridades:

$$Answer\ Relevancy = \frac{1}{N}\sum_{i=1}^{N}\cos(E_{g_i}, E_o) \tag{10}$$

Valores próximos de 1 indicam alto alinhamento entre a intenção da pergunta e o conteúdo da resposta, enquanto valores baixos sugerem que a resposta desviou do tópico ou é excessivamente genérica.

2.4.3 Context Precision (Precisão do Contexto)

A métrica de *Context Precision* avalia a qualidade do ordenamento (*ranking*) realizado pelo sistema de recuperação. Diferente de métricas estáticas que apenas verificam a presença da informação, esta métrica mensura se os documentos mais relevantes foram priorizados no topo da lista, o que é crítico para o desempenho de LLMs devido ao viés de primazia.

Conforme a definição formal do *framework Ragas* [Exploding Gradients, 2024b], o *score* é calculado através da média da precisão em k ($Precision@k$) ponderada pela relevância posicional:

$$Context\ Precision@K = \frac{\sum_{k=1}^{K}(Precision@k \times v_k)}{\text{Total de itens relevantes no Top K}} \tag{11}$$

Onde:

- K é o número total de documentos recuperados.
- $Precision@k$ é a proporção de documentos relevantes até a posição k.
- $v_k \in \{0, 1\}$ é o indicador de relevância na posição k, avaliado por um LLM comparando o documento com o gabarito oficial (*reference*).

Valores próximos de 1 indicam que todos os documentos relevantes para responder à pergunta foram apresentados nas primeiras posições, minimizando o ruído contextual que antecede a informação útil.

2.4.4 Context Recall (Revocação do Contexto)

A métrica de *Context Recall* avalia a extensão em que o contexto recuperado (C) alinha-se com a resposta esperada ou *Ground Truth* (G). Conforme definido na documentação do *framework Ragas* [Exploding Gradients, 2024c], esta métrica verifica se toda a informação necessária para responder à pergunta está presente nos documentos recuperados.

O algoritmo de cálculo utiliza um LLM para realizar a auditoria sentença a sentença:

1. **Decomposição:** O gabarito oficial (G) é fragmentado em uma lista de sentenças ou afirmações atômicas individuais.
2. **Verificação de Atribuição:** Para cada sentença s_i do gabarito, o modelo verifica se ela pode ser atribuída ou deduzida a partir do contexto recuperado (C).

O *score* final é a razão entre as sentenças do gabarito que são suportadas pelo contexto e o total de sentenças do gabarito:

$$Context\ Recall = \frac{|\{s \in G \mid C \text{ suporta } s\}|}{|G|} \quad (12)$$

Valores próximos de 1 indicam que o sistema de recuperação trouxe todas as informações relevantes contidas na resposta ideal. Valores baixos indicam que o contexto recuperado está incompleto, obrigando o LLM a usar conhecimento paramétrico ou alucinar para responder corretamente.

2.4.5 Answer Correctness (Corretude da Resposta)

A métrica de *Answer Correctness* avalia a precisão da resposta gerada (A) em comparação direta com o gabarito oficial ou *Ground Truth* (G). Conforme a documentação oficial do *framework Ragas* [Exploding Gradients, 2024a], esta métrica emprega uma abordagem híbrida que combina avaliação semântica e factual através de uma média ponderada.

O cálculo ocorre em duas etapas distintas:

1. **Corretude Factual (*Factual Correctness*):** Utiliza-se um LLM para decompor tanto a resposta gerada quanto o gabarito em um conjunto de afirmações atômicas. A partir dessa decomposição, o algoritmo classifica as afirmações em três categorias:
   - **TP (Verdadeiros Positivos):** Fatos presentes tanto na resposta gerada quanto no gabarito.
   - **FP (Falsos Positivos):** Fatos presentes na resposta gerada, mas ausentes no gabarito (alucinação ou informação extra não solicitada).
   - **FN (Falsos Negativos):** Fatos presentes no gabarito, mas omitidos na resposta gerada.

   O *score* factual é calculado utilizando a fórmula do *F1-Score* modificada para conjuntos:

$$Score_{factual} = \frac{TP}{TP + 0.5 \times (FP + FN)} \quad (13)$$

2. **Similaridade Semântica (*Semantic Similarity*):** Calcula-se a similaridade de cosseno entre os vetores de *embedding* da resposta gerada e do gabarito. Este componente serve como um "fator de amortecimento"para capturar a equivalência de significado global, mesmo que a fraseologia exata difira.

O *score* final de *Answer Correctness* é a média ponderada desses dois componentes, tipicamente atribuindo maior peso à corretude factual (ex: 0.75) em detrimento da similaridade puramente vetorial (ex: 0.25).

2.5 Delineamento e Modelagem

A principal contribuição metodológica deste trabalho reside na formalização do processo de avaliação de sistemas de IA. Enquanto a prática comum na Ciência da Computação envolve ajustes empíricos de hiperparâmetros via tentativa e erro, este estudo fundamenta-se na teoria do DoE para inferir causalidade e quantificar incertezas.

2.5.1 O Experimento Fatorial Completo

Para investigar como diferentes configurações afetam a qualidade do RAG, optou-se por um Delineamento Fatorial Completo (n^k). Esta abordagem testa exaustivamente todas as combinações possíveis entre os níveis dos fatores, varrendo integralmente o espaço de busca definido para os hiperparâmetros.

A escolha por este delineamento, em detrimento de métodos fracionados ou testes "um-fator-por-vez"(OFAT), justifica-se pela necessidade de robustez. Ao avaliar cada fator sob diversas condições de contorno (por exemplo, testar a Busca Híbrida sob quatro tipos diferentes de *chunking*), assegura-se que a estimativa dos efeitos principais não esteja enviesada por um cenário específico, garantindo que a configuração recomendada tenha validade geral [Montgomery, 2017].

Os fatores controlados (X) e a variável resposta (Y) definem a estrutura do experimento:

- **Fatores (Variáveis Independentes):**
  1. *Estratégia de Chunking* (4 níveis): Recursiva (500:100 e 1000:200 *tokens*) e Semântica (Percentil 75 e 95).
  2. *Tipo de Busca* (3 níveis): Vetorial, Textual (BM25) e Híbrida (RRF).
  3. *Top-K* (4 níveis): 5, 10, 15, 20 documentos recuperados.
  4. *Modelo LLM* (2 níveis): GPT-4o e GPT-4o-mini.

- **Variável Resposta (Dependente):** O *score* contínuo $Y \in [0, 1]$ atribuído pelo oráculo para as métricas de qualidade (*Faithfulness*, *Relevancy*, *Precision*, *Recall*, *Correctness*).

O número total de tratamentos (combinações únicas) é $4 \times 3 \times 4 \times 2 = 96$. Como cada tratamento foi replicado para 40 perguntas distintas (bloqueagem), o tamanho total da amostra é $N = 3.840$ observações experimentais.

2.5.2 Transformação de Postos Alinhados (ART)

Dada a natureza das métricas de avaliação, que são limitadas ao intervalo $[0, 1]$ e podem apresentar distribuições assimétricas, a análise estatística foi conduzida utilizando a técnica de *Aligned Rank Transform* (ART), ou Transformação de Postos Alinhados, proposta por Wobbrock *et al*. [2011].

O ART é um método não-paramétrico desenhado especificamente para a análise de experimentos fatoriais com múltiplos fatores. O procedimento metodológico consiste em três etapas sequenciais para cada termo do modelo (seja um efeito principal ou uma interação):

1. **Alinhamento:** Os dados brutos são transformados para isolar o efeito de interesse. Subtraem-se da resposta os efeitos estimados de todos os outros fatores (efeitos "ruído"para aquele teste específico), resultando nos resíduos alinhados.
2. **Ranqueamento:** Aplica-se a transformação de postos (*ranking*) aos dados alinhados.
3. **Análise de Variância:** Realiza-se a Análise de Variância (F-test) sobre os postos alinhados para testar a significância estatística.

Essa abordagem possibilita o uso de procedimentos robustos de inferência estatística em dados que não atendem aos pressupostos de normalidade ou homocedasticidade, mantendo o poder de detectar nuances complexas no comportamento dos hiperparâmetros.

2.5.3 Comparações Múltiplas

A análise de variância via ART indica apenas a existência de diferenças significativas globais, mas não especifica entre quais níveis elas ocorrem. Para identificar essas diferenças pontuais (ex: se a *Busca Híbrida* é estatisticamente superior à *Busca Vetorial*), utilizou-se o método de *Aligned Rank Transform Contrasts*, ou Contrastes de Postos Alinhados, conforme proposto por Elkin *et al*. [2021].

Diferente de testes *post-hoc* paramétricos tradicionais (como Tukey [Montgomery, 2017]), este método computa as diferenças par-a-par diretamente sobre os postos alinhados, preservando a robustez não-paramétrica do delineamento fatorial.

Para assegurar o rigor estatístico e controlar a taxa de erro familiar (FWER) decorrente dos múltiplos testes simultâneos, os p-valores resultantes foram penalizados utilizando a correção de **Bonferroni** [Dunn, 1961]. Neste protocolo conservador, uma diferença entre configurações é considerada estatisticamente significativa apenas se o p-valor ajustado for inferior ao nível de significância $\alpha = 0.05$.

3 Trabalhos Relacionados

A rápida evolução da arquitetura RAG estimulou uma vasta produção acadêmica focada na engenharia de seus componentes e na aplicação em domínios de alta precisão. A literatura recente evidencia uma lacuna entre estudos empíricos de "melhores práticas"e a validação estatística rigorosa dessas configurações em línguas e domínios específicos.

No contexto global de otimização, Wang *et al*. [2024] conduziram uma investigação extensiva sobre a sensibilidade dos sistemas RAG a escolhas de *design* e analisaram empiricamente o impacto de estratégias de segmentação e recuperação. Embora estabeleçam diretrizes valiosas para domínios gerais, a análise baseia-se majoritariamente em estudos de ablação, ou remoção de componentes, e observação direta de métricas, sem a aplicação de um formalismo como o DoE para quantificar interações complexas entre fatores. Nosso trabalho preenche essa lacuna metodológica ao aplicar a Transformação de Postos Alinhados (ART) para validar estatisticamente essas práticas.

A granularidade da informação recuperada é outro ponto crítico. Zhang *et al*. [2024b] demonstraram que a estratégia de *chunking* afeta drasticamente a coerência semântica e sugerem que quebras baseadas em significado superam cortes recursivos simples. Estendemos essa investigação ao testar essa hipótese em textos jurídicos brasileiros, caracterizados por sentenças longas e vocabulário arcaico, um cenário onde a preservação do contexto é ainda mais desafiadora do que nos *corpora* de língua inglesa tipicamente avaliados.

Quanto à aplicação em domínios sensíveis, Johnson *et al*. [2023] consolidaram o uso de RAG na área da saúde e evidenciaram sua capacidade de reduzir alucinações em contextos críticos. No cenário nacional, de Aquino *et al*. [2024] iniciaram a exploração de RAG em documentos jurídicos brasileiros com foco na arquitetura de extração de informações. Contudo, este estudo anterior priorizou a viabilidade técnica e a validação qualitativa. A presente pesquisa avança ao focar na otimização quantitativa dos parâmetros de recuperação e utiliza o paradigma *LLM-as-a-Judge* para auditar a fidelidade do sistema frente a um acervo legislativo não estruturado.

Por fim, no que tange às técnicas de refinamento de busca, propostas recentes como o RankRAG [Yu *et al*., 2024] utilizam modelos neurais avançados, os *Cross-Encoders*, para reordenar documentos. Embora eficazes, esses modelos introduzem latência significativa na inferência. Em contraste, nossa pesquisa investiga a eficácia do algoritmo RRF como uma alternativa computacionalmente eficiente para fundir buscas vetoriais e lexicais, buscando um equilíbrio entre precisão e custo computacional viável para instituições públicas.

A revisão da literatura revela que, embora existam diretrizes gerais para a construção de sistemas RAG, a maioria dos estudos adota uma abordagem de engenharia empírica, ou "melhores práticas", baseada na observação direta de métricas de desempenho sem o suporte de testes de hipóteses formais. Além disso, as soluções desenvolvidas para o contexto jurídico brasileiro focam predominantemente na viabilidade arquitetural e negligenciam a otimização matemática dos parâmetros de recuperação. A Tabela 1 sintetiza essas limitações identificadas nos trabalhos correlatos e demonstra como a presente pesquisa preenche essas lacunas através da aplicação rigorosa do DoE.

Tabela 1. Síntese das lacunas e contribuições da pesquisa.

Estudo	Lacuna / Abordagem	Solução Proposta
Wang *et al*. [2024]	Baseia-se em empirismo e ablação. Sem validação estatística formal.	Uso de **DoE e ART** para validar interações entre fatores.
de Aquino *et al*. [2024]	Foco arquitetural (extração). Validação apenas qualitativa.	Otimização **quantitativa** de parâmetros via *LLM-as-a-Judge*.
Zhang *et al*. [2024b]	Testes de *chunking* focados na língua inglesa.	Validação de *chunking* semântico para **Jurídico PT-BR**.
Yu *et al*. [2024]	Re-ranking neural (pesado). Alta latência.	Uso de **RRF**, alternativa estatística eficiente.

4 Metodologia

Este capítulo descreve os procedimentos computacionais e estatísticos adotados para a construção e avaliação do sistema de RAG aplicado ao domínio jurídico do MPES. A metodologia segue uma abordagem experimental quantitativa, estruturada sobre uma ou *tech stack* ou pilha tecnológica, moderna que integra engenharia de dados, inferência via LLMs e análise estatística inferencial.

A implementação do sistema foi desenvolvida predominantemente em linguagem **Python 3.11**, utilizando o *framework LangChain* [Chase, 2023] para a orquestração dos fluxos de raciocínio e a biblioteca *FastAPI* [Ramírez, 2018] para a exposição de *endpoints* de inferência controlados. Para a persistência e busca semântica, adotou-se o banco de dados vetorial *ChromaDB* [Team, 2023], enquanto os modelos generativos e de *embedding* foram consumidos via API do *Azure OpenAI Service* [Microsoft, 2024].

A validação dos resultados foi conduzida no ambiente estatístico **R** [R Core Team, 2023], especificamente através do pacote `ARTool` [Kay *et al*., 2021], para a condução de testes de hipóteses não-paramétricos.

4.1 Pipeline de Extração e Sincronização

Diferente de abordagens estáticas que processam uma carga de dados única (*batch*), implementou-se um *pipeline* de sincronização contínua capaz de manter a base vetorial atualizada em relação ao repositório oficial. O processo foi estruturado nas etapas Extracting, Transforming and Loading (ETL), ou Extração, Transformação e Carregamento.

4.1.1 Coleta Automatizada (Web Scraping)

A extração dos dados foi realizada diretamente no portal "Legislação Compilada"do MPES. Desenvolveu-se um *scraper*, ou coletor, em Python utilizando a biblioteca *Requests*[Reitz, 2023] para emular sessões de navegação.

Para contornar as limitações de interatividade do portal, o algoritmo foi projetado para manipular *tokens* de estado de visualização (`__VIEWSTATE` e `__EVENTVALIDATION`). Essa manipulação permitiu a paginação automatizada e a alteração programática da densidade de itens para 100 documentos por requisição, otimizando o tempo de coleta. A Figura 3 ilustra o processo.

A lógica de coleta implementou uma heurística de prioridade para a seleção do *link* do texto normativo:

1. **Prioridade 1:** Busca pelo *link* "Texto Compilado"(versão da lei com atualizações e revogações consolidadas).
2. **Prioridade 2:** Caso inexistente, busca pelo *link* "Texto Completo"(versão original da publicação).

4.1.2 Sincronização Incremental (Delta Load)

Para garantir a eficiência computacional e a consistência temporal da base, implementou-se uma lógica de carga delta. Antes do processamento textual, o sistema compara os identificadores (URLs) coletados no portal com os metadados já existentes no banco vetorial local:

- **Novos Documentos:** Apenas normas inéditas ($IDs_{portal} - IDs_{banco}$) são baixadas e enviadas para o fluxo de vetorização.
- **Documentos Obsoletos:** Normas que constavam no banco mas foram removidas do portal oficial ($IDs_{banco} - IDs_{portal}$) são detectadas e excluídas automaticamente da base vetorial, processo denominado *pruning*, mitigando o risco de recuperação de legislações revogadas.

4.1.3 Limpeza e Extração Textual

O conteúdo textual dos documentos foi extraído utilizando a biblioteca *BeautifulSoup*. O processo de limpeza consistiu na extração do texto visível com separadores de linha preservados, descartando-se *tags* HTML estruturais (`<div>`, `<table>`), *scripts* e folhas de estilo. O resultado é um *corpus* de texto plano associado ao seu título original e *link* de origem.

4.2 Estratégias de Segmentação e Vetorização

A preparação dos dados para o modelo RAG envolveu técnicas de *chunking* para adequar os textos às janelas de contexto dos Modelos de Linguagem e preservar a semântica jurídica.

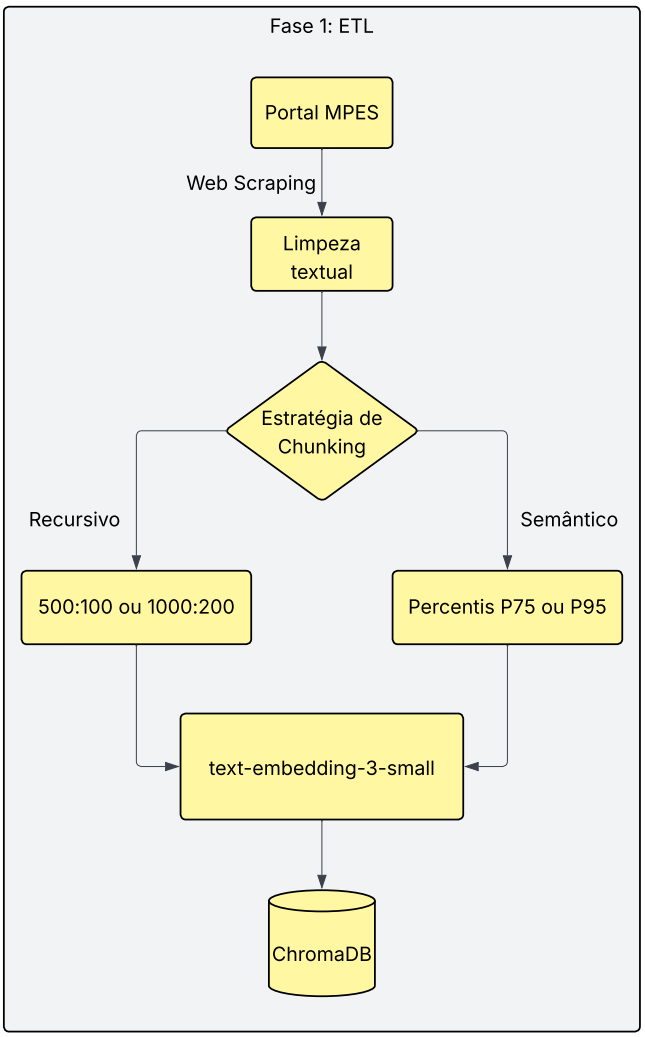

Figura 3. Fluxo ETL dos atos normativos do MPES

4.2.1 Segmentação Híbrida (Hybrid Chunking)

Cada documento processado foi submetido a múltiplas estratégias de segmentação simultâneas. O objetivo desta abordagem redundante é aumentar a probabilidade de recuperação, oferecendo janelas de contexto de tamanhos e naturezas variadas para o mesmo trecho de lei. As estratégias configuradas foram:

1. Chunking Recursivo: Utilizou-se a classe `RecursiveCharacterTextSplitter` do *framework LangChain*, configurada para respeitar delimitadores naturais (parágrafos e quebras de linha). Foram gerados dois índices com granularidades distintas:

- **Granularidade Fina:** Blocos de 500 caracteres com 100 caracteres de sobreposição.
- **Granularidade Média:** Blocos de 1000 caracteres com 200 caracteres de sobreposição.

2. Chunking Semântico: Implementou-se a classe `SemanticChunker`, que segmenta o texto baseando-se na similaridade de significado entre sentenças contíguas, e não apenas em caracteres fixos. O algoritmo calcula a distância de cosseno entre os *embeddings* das sentenças e define pontos de corte baseados em percentis da distribuição de distâncias do documento:

- **Nível 75 (P_{75}):** Gera segmentos menores, capturando

mudanças sutis de tópico.
- **Nível 95 (P_{95}):** Gera segmentos maiores, agrupando seções temáticas inteiras.

4.2.2 Verificação de Segurança de Tokens

Foi implementado um mecanismo de segurança utilizando a biblioteca `tiktoken` [OpenAI, 2023]. Caso qualquer segmento gerado (especialmente na estratégia semântica) exceda o limite de *tokens* suportado pelo modelo de *embedding* (8191 *tokens*), o sistema aplica recursivamente uma nova subdivisão forçada naquele bloco específico, prevenindo falhas de API durante a vetorização.

4.2.3 Vetorização e Armazenamento

Os segmentos resultantes foram convertidos em vetores densos (*embeddings*) utilizando o modelo `text-embedding-3-small` via *Azure OpenAI*. O armazenamento foi realizado no banco de dados vetorial *ChromaDB* em modo persistente.

Para garantir a rastreabilidade e a futura filtragem, cada vetor foi enriquecido com metadados estruturados contendo: o ID do documento de origem, o Título da Portaria, a Estratégia de *Chunking* utilizada e o identificador do Modelo de *Embedding*.

4.3 Arquitetura de Recuperação e Geração

A etapa de inferência foi operacionalizada através de uma API REST desenvolvida com o *framework FastAPI*. O sistema foi arquitetado para permitir a execução modular das etapas de recuperação da informação e geração de texto. A Figura 4 ilustra o pipeline de geração da resposta a partir da pergunta.

4.3.1 Estratégias de Recuperação da Informação

Para permitir otimizar o process, o motor de busca foi implementado de forma modular. Esta arquitetura possibilita a alternância entre três modalidades de recuperação via parâmetros da API, permitindo isolar o impacto de cada método nas métricas de qualidade [Montgomery, 2017]:

- **Recuperação Vetorial (*Semantic Search*):** Executa a busca por similaridade de cosseno no banco *ChromaDB* utilizando os *embeddings* densos gerados pelo modelo *text-embedding-3-small*[cite: 321, 362, 368].
- **Recuperação Lexical (*Keyword Search*):** Operacionalizada através do algoritmo probabilístico BM25. Para esta modalidade, os documentos passaram por um pré-processamento via biblioteca *NLTK*, incluindo tokenização, conversão para caixa baixa e remoção de *stopwords* em português para reduzir o ruído do índice invertido[cite: 370, 371, 374].
- **Recuperação Híbrida (*Hybrid Search*):** Integra os resultados das buscas vetorial e lexical em consultas paralelas, consolidando-os através do algoritmo *Reciprocal Rank Fusion* (RRF) [Cormack *et al.*, 2009][cite: 376, 377]. Conforme definido na fundamentação teórica, utilizou-se a constante de suavização $k = 60$ para priorizar documentos consensuais entre ambos os métodos de busca[cite: 379, 380, 381].

Esta estrutura modular foi desenhada para viabilizar a coleta isolada das métricas de recuperação (*Context Precision* e *Context Recall*) antes da etapa de geração, garantindo que a variabilidade introduzida pelos algoritmos de busca pudesse ser quantificada estatisticamente de forma independente da performance do LLM[cite: 389, 421].

4.3.2 Geração Aumentada e Engenharia de Prompt

O módulo de geração utiliza modelos da família GPT-4 (acessados via *Azure OpenAI*) para sintetizar a resposta final. A construção do *prompt* enviado ao modelo segue uma arquitetura de janelas deslizantes (*sliding window*) para gestão de memória conversacional, composta por quatro blocos sequenciais:

1. **Instrução do Sistema (*System Prompt*):** Define a persona do assistente como especialista em legislação do MPES e impõe restrições de alucinação (ex: "Se a informação não estiver no contexto, responda que não encontrou").
2. **Histórico de Conversação:** Injeta os últimos N turnos de interação (pergunta-resposta) para manter a coerência do diálogo.
3. **Contexto Recuperado (RAG):** Apresenta os trechos de legislação selecionados pela estratégia de recuperação ativa, formatados explicitamente com suas respectivas fontes.
4. **Pergunta do Usuário:** A consulta atual a ser respondida.

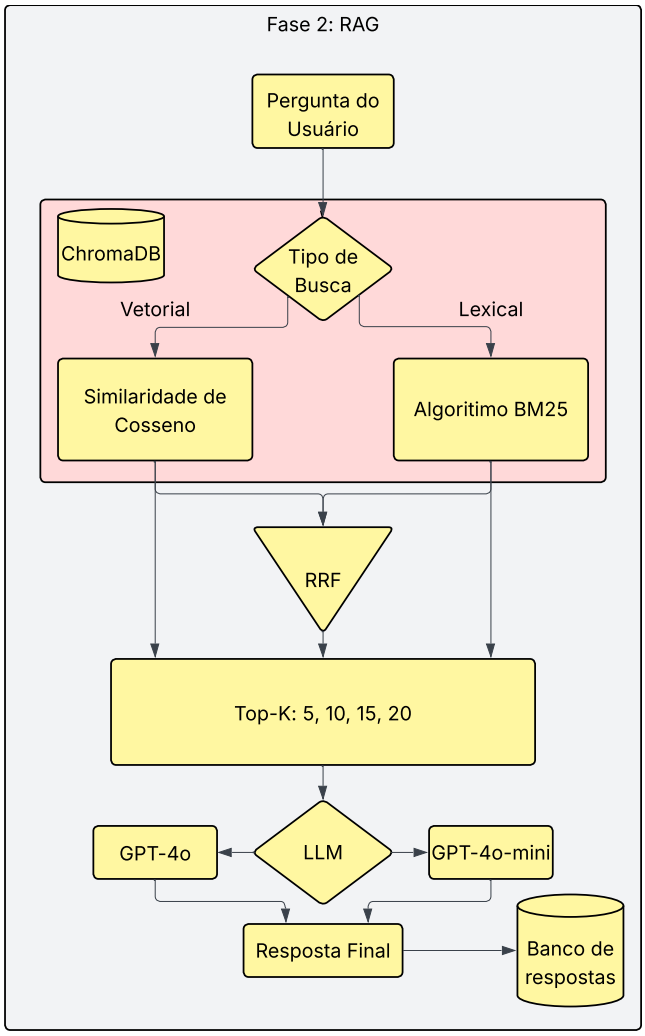

Figura 4. Arquitetura do pipeline RAG com motor de busca híbrida

4.3.3 Endpoints Experimentais

Para viabilizar o Delineamento de Experimentos (DoE), a API foi estruturada com *endpoints* dissociados que permitem mensurar o desempenho de cada componente isoladamente:

- `/recuperar_contexto`: Executa apenas a busca (Vetorial, Lexical ou Híbrida), permitindo o cálculo de métricas de recuperação (*Precision@K* e *Recall@K*) sem a latência da geração de texto.
- `/gerar_resposta`: Recebe um contexto fixo e uma pergunta, isolando a avaliação da capacidade de síntese e alucinação do LLM sob diferentes temperaturas e *prompts*.

4.4 Geração de Dados Sintéticos e

A avaliação objetiva de sistemas de recuperação exige um conjunto de referência, denominado *Golden Set* ou *Ground Truth*, composto por pares curados de pergunta e resposta ideal. Dada a inexistência de bases anotadas manualmente para o domínio específico do MPES, adotou-se uma estratégia de geração de dados sintéticos assistida por LLM (*LLM-as-a-judge/generator*).

O processo de construção do *Golden Set* foi sistematizado nas etapas de amostragem, engenharia de *prompt* e estruturação taxonômica, conforme descrito a seguir.

4.4.1 Amostragem e Preparação do Contexto

Para garantir a representatividade temática do acervo, realizou-se uma amostragem aleatória simples de $N = 100$ documentos a partir do *corpus* higienizado. O conteúdo textual destes documentos foi concatenado e formatado com marcadores explícitos de início e fim, preservando o metadado de fonte (URL) para fins de rastreabilidade.

4.4.2 Taxonomia de Perguntas (Complexidade Cognitiva)

Para evitar viés de simplicidade, onde o modelo gera apenas perguntas fáceis de recuperação direta, o algoritmo de geração foi instruído via *prompt* a distribuir o *dataset* equitativamente entre quatro categorias de complexidade cognitiva:

1. **Fatual (25%):** Perguntas objetivas que exigem a recuperação de um fato explícito (ex: "Qual o prazo para interposição de recursos?").
2. **Procedimental (25%):** Questões que demandam a descrição de um fluxo de ações ou requisitos legais (ex: "Descreva o rito para solicitação de teletrabalho").
3. **Síntese e Integração (25%):** Perguntas complexas que exigem que o modelo relacione informações de diferentes partes do texto ou compare normas (ex: "Quais as diferenças entre férias e licença-prêmio neste regulamento?").
4. **Agulha no Palheiro (*Needle-in-a-Haystack*) (25%):** Questões sobre detalhes numéricos ou específicos, geralmente obscuros, para testar a precisão fina da recuperação (ex: "Qual o valor exato do auxílio citado no artigo 3º?").

4.4.3 Engenharia de Prompt e Estrutura de Saída

Utilizou-se um Modelo de Linguagem de Larga Escala (LLM) com ampla janela de contexto para atuar como o gerador. O *system prompt* foi desenhado com uma persona de "Especialista Jurídico e Analista de Dados", incluindo restrições negativas estritas para impedir que a pergunta contivesse "dicas"artificiais, como o nome do arquivo ou a URL de origem. O prompt completo pode ser visualizado no Apêndice A.

O resultado final foi um conjunto de 40 triplas de avaliação estruturadas em formato JSON, contendo:

- ***Question*:** A pergunta do usuário simulado.
- ***Ground Truth Answer*:** A resposta ideal e completa.
- ***Evidence Quote*:** A citação exata (trecho do texto original) que fundamenta a resposta, permitindo a verificação de alucinações.
- ***Source*:** O *link* do documento original para validação de referência.

4.5 Procedimento Experimental e Controle de Variáveis

Para assegurar a validade interna e a reprodutibilidade do estudo, o experimento foi estruturado em um fluxo de execução desacoplado, permitindo o controle estrito sobre a estocasticidade e as variáveis de confusão.

4.5.1 Variáveis de Controle e Estabilidade

Para isolar o efeito dos fatores em estudo (hiperparâmetros do RAG), as seguintes variáveis foram mantidas fixas:

- **Determinismo Térmico:** A temperatura foi fixada em $T = 0,1$. Este valor, próximo ao limite inferior da API, minimiza a variabilidade criativa e favorece a consistência factual necessária ao domínio jurídico.
- **Padronização de Instrução:** Utilizou-se um único *System Prompt* (Apêndice A) simulando a persona de um servidor público, com restrições negativas estritas contra alucinações.
- **Slicing Determinístico de Contexto:** Para avaliar o impacto do tamanho da janela ($K \in \{5, 10, 15, 20\}$), o sistema recupera o máximo de documentos ($K_{max} = 20$) e recorta os subconjuntos menores em memória. Isso garante matematicamente que a informação contida em K_i seja um subconjunto estrito de K_{i+1}, permitindo a análise precisa do ganho marginal de informação.

4.5.2 Fluxo de Execução em Duas Fases

A coleta de dados foi automatizada em Python e dividida em duas etapas estanques para garantir que diferentes modelos fossem avaliados sob contextos idênticos:

Fase 1: Recuperação e Persistência (JSON) O algoritmo iterou sobre as combinações de parâmetros de recuperação (Estratégias de *Chunking* $\times$ Tipos de Busca). Para cada pergunta do *Golden Set*, os 20 documentos recuperados, incluindo metadados e *scores* de similaridade, foram serializados em disco. O uso de verificações de idempotência evitou reprocessamentos e garantiu a integridade do índice vetorial durante a execução dos testes.

Fase 2: Geração e Consolidação (CSV) O módulo gerador consumiu o contexto diretamente do arquivo estático gerado na Fase 1. O *script* formatou os subconjuntos de K via *slicing* e despachou as requisições para os modelos `gpt-4o` e `gpt-4o-mini`. Os resultados foram consolidados em um *dataset* final contendo a tripla essencial (Pergunta, Contexto,

Resposta) e as variáveis independentes do experimento, servindo de entrada direta para o cálculo de métricas no *framework Ragas* [Es *et al*., 2023; Gradients, 2024].

4.6 Protocolo de Avaliação Automatizada (*LLM-as-a-Judge*)

Dada a inviabilidade prática da avaliação humana manual para as 3.840 inferências geradas, adotou-se o *framework Ragas* (*Retrieval Augmented Generation Assessment*) para a mensuração objetiva de qualidade. Esta abordagem implementa o paradigma *LLM-as-a-Judge*, onde um modelo de linguagem de capacidade superior atua como juiz das respostas geradas pelo sistema.

4.6.1 Configuração do Juiz Sintético

Para garantir a confiabilidade das métricas, o avaliador foi instanciado utilizando o modelo **GPT-4o** (via *Azure OpenAI*). Diferente da fase de geração, o modelo-juiz foi configurado com temperatura nula ($T = 0$) e instruído a operar com determinismo estrito.

O *script* de avaliação implementou mecanismos de robustez, incluindo:

- **Retentativas Automáticas (*Retry Logic*):** Configurado para realizar até 3 tentativas em caso de falhas de comunicação com a API ou *timeouts*.
- **Sanitização de Dados:** Tratamento preventivo de valores nulos ou vazios (*NaN*) nas colunas de texto para evitar interrupções no *pipeline* de avaliação.

4.6.2 Alinhamento de Dados e Contexto

Um ponto crítico da metodologia foi o alinhamento correto das entradas para o avaliador. O *framework* recebeu a seguinte quádrupla para cada observação:

1. ***Question*:** A pergunta original do *Golden Set*.
2. ***Ground Truth*:** A resposta esperada ideal.
3. ***Answer*:** A resposta gerada pelo modelo em teste (GPT-4o ou GPT-4o-mini).
4. ***Contexts*:** O conteúdo exato da coluna `full_context_sent`.

É imperativo notar que a avaliação utilizou o contexto *efetivamente enviado* ao *prompt* de geração (após o *slicing* de K), e não a lista bruta de documentos recuperados. Isso assegura que as métricas de fidelidade avaliem se o modelo alucinou com base no que ele realmente "leu", e não no que estava disponível no banco de dados.

4.6.3 Métricas Computadas

O sistema calculou cinco métricas ortogonais para cobrir os diferentes aspectos da qualidade do RAG:

- ***Faithfulness* (Fidelidade):** Mede se a resposta gerada pode ser inferida a partir do contexto fornecido. O Juiz extrai afirmações da resposta e verifica se cada uma é suportada pelas evidências do contexto.
- ***Answer Relevancy* (Relevância):** Avalia se a resposta endereça a pergunta do usuário, penalizando respostas que, embora verdadeiras, são evasivas ou redundantes.
- ***Context Precision*:** Avalia a taxa de sinal/ruído nos documentos recuperados, verificando se os documentos relevantes estão bem posicionados no *ranking*.
- ***Context Recall*:** Mede se o contexto recuperado contém todas as informações necessárias para responder à pergunta (comparando com o *Ground Truth*).
- ***Answer Correctness*:** Avalia a precisão semântica e factual da resposta gerada em comparação direta com o gabarito (*Ground Truth*), combinando similaridade semântica (via *embeddings*) e verificação factual.

4.7 Análise Estatística e Tratamento dos Dados

O conjunto de dados final, contendo os *scores* das métricas calculadas pelo juiz sintético para cada uma das 3.840 observações, foi exportado para o ambiente estatístico R (versão 4.3). O roteiro de análise foi estruturado para validar as hipóteses do estudo e identificar quais fatores (Estratégia de *Chunking*, Tipo de Busca, Modelo LLM e *Top-K*) exercem influência significativa sobre a qualidade do sistema.

4.7.1 Análise Descritiva e Visualização

Inicialmente, realizou-se uma análise exploratória para caracterizar a distribuição dos *scores*. Foram calculadas as medidas de tendência central (média) e dispersão (desvio-padrão) para cada métrica estratificada pelos fatores experimentais.

Para a inspeção visual, geraram-se histogramas de frequência para verificar a normalidade dos dados e gráficos de interação (*interaction plots*) com barras de erro padrão (SE), permitindo a identificação preliminar de efeitos cruzados entre as estratégias de indexação e recuperação.

4.7.2 Modelagem Estatística (ART ANOVA)

Dada a natureza limitada dos *scores* de avaliação (intervalo $[0, 1]$) e a violação dos pressupostos de normalidade e homocedasticidade exigidos pela ANOVA paramétrica tradicional, optou-se pela aplicação do método ART.

Utilizando o pacote `ARTool`, ajustou-se um modelo fatorial completo não-paramétrico para cada uma das cinco métricas de resposta (Y). O modelo estatístico considerou os efeitos principais e as interações de até quarta ordem:

$$Y_{ijkl} \sim Chunking_i \times Busca_j \times Modelo_k \times TopK_l + \epsilon$$

4.7.3 Testes de Hipóteses e Comparações Múltiplas

A validação das hipóteses seguiu um fluxo hierárquico:

1. **Teste *Omnibus*:** Avaliou-se a tabela ANOVA resultante da transformação ART para identificar quais fatores ou interações apresentaram significância estatística global ($p < 0.05$).
2. **Contrastes *Post-Hoc*:** Para os fatores significativos, realizaram-se comparações par-a-par (*pairwise contrasts*) utilizando o método `art.con`.
3. **Correção de Bonferroni:** Para mitigar o risco de falsos positivos (Erro Tipo I) decorrente das múltiplas comparações simultâneas, aplicou-se a correção de Bonferroni ($p_{ajustado} = p \times m$) sobre os p-valores resultantes, garantindo um rigor conservador nas inferências finais.

5 Resultados e Discussão

Este capítulo apresenta a análise detalhada do desempenho do sistema de RAG, baseada em um esforço experimental de

$N = 3.840$ observações. A avaliação objetiva mensurar o impacto individual e combinado dos quatro hiperparâmetros controlados: Estratégia de Segmentação, Tipo de Busca, Janela de Contexto (k) e Modelo LLM, sobre a qualidade final das respostas jurídicas.

5.1 Análise Descritiva e Distribucional

A Figura 5 apresenta os histogramas de frequência para as cinco métricas de avaliação. A inspeção visual revela padrões distribucionais que justificam a adoção de métodos não-paramétricos robustos em detrimento da ANOVA tradicional.

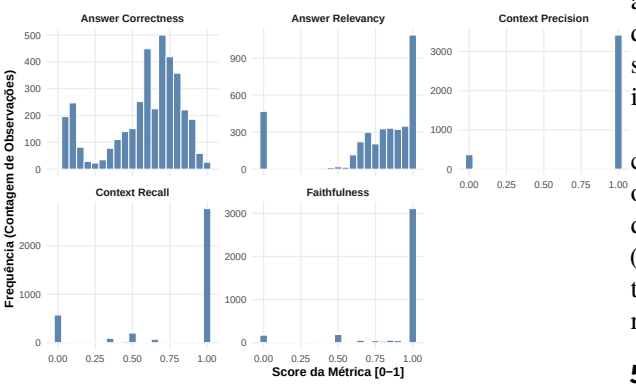

Figura 5. Distribuição de Frequência das Métricas de Avaliação.
Fonte: Elaborado pelo autor.

A análise dos histogramas permite três diagnósticos fundamentais sobre o comportamento do sistema:

1. **Efeito Teto e Polarização:** As métricas de *Context Recall* e *Context Precision* exibem um comportamento polarizado ("tudo ou nada"). A concentração massiva de valores em 1.0 (barras à direita) indica que, quando o sistema funciona, ele tende a ser perfeito. Contudo, a presença de valores em 0.0 (especialmente no *Recall*) denuncia falhas catastróficas de recuperação em *queries* difíceis.
2. **Alta Fidelidade do LLM:** O histograma de *Faithfulness* é fortemente assimétrico, com a maioria das observações concentradas em 1, demonstrando que os modelos testados raramente alucinam informações não contidas no contexto. A quase totalidade das respostas mantém-se fiel aos documentos recuperados.
3. **Multimodalidade na Corretude:** Diferente das demais, a métrica global *Answer Correctness* apresenta uma distribuição multimodal. Há um pico em valores baixos (erros factuais) e uma grande massa entre 0.60 e 0.90. Isso reflete a natureza híbrida da métrica, que penaliza severamente erros factuais mesmo em respostas semanticamente coerentes.

5.2 Análise dos Fatores Experimentais

Esta seção detalha o impacto isolado de cada hiperparâmetro sobre as métricas de avaliação. As tabelas a seguir apresentam a Média e o Desvio Padrão (entre parênteses) para cada configuração. O desvio padrão é um indicador de estabilidade: valores altos sugerem que o sistema oscila entre o sucesso total e a falha completa.

5.2.1 Estratégia de Segmentação (Chunking)

A Tabela 2 evidencia que a granularidade da segmentação é o fator mais crítico para o sucesso da recuperação.

Tabela 2. Desempenho por Estratégia de Chunking.

Estratégia	Recall	Precision	Faithfulness	Relevancy	Correctness
Recursive (1000)	0.81 (0.34)	**0.94 (0.23)**	**0.95 (0.18)**	**0.78** (0.27)	0.61 (0.22)
Recursive (500)	0.65 (0.42)	0.86 (0.35)	0.89 (0.27)	0.71 (0.35)	0.52 (0.25)
Semantic (P75)	0.81 (0.38)	0.87 (0.33)	0.88 (0.27)	0.74 (0.32)	0.59 (0.26)
Semantic (P95)	**0.89 (0.31)**	0.93 (0.25)	0.91 (0.23)	0.77 (0.28)	**0.62 (0.24)**

A estratégia ***Recursive 500*** (fragmentos pequenos) apresentou o pior desempenho global, com um *Context Recall* de apenas 0.65 e o maior desvio padrão (0.42). Isso indica que quebrar textos legais em blocos pequenos rompe a unidade semântica necessária para a recuperação, tornando o sistema instável.

Em contraste, a estratégia ***Semantic* (Percentil 95)**, que cria blocos baseados em grandes mudanças de tópico, atingiu o maior *Recall* (0.89) e Corretude (0.62). Observa-se também que a estratégia ***Recursive 1000*** obteve a maior *Precision* (0.94) e *Faithfulness* (0.95), sugerindo que janelas de contexto maiores fornecem informações mais completas ao LLM, reduzindo a chance de alucinações.

5.2.2 Estratégia de Busca

A Tabela 3 confirma a hipótese de que a combinação de métodos supera as abordagens isoladas.

Tabela 3. Desempenho por Tipo de Busca.

Busca	Recall	Precision	Faithfulness	Relevancy	Correctness
Híbrida	**0.83 (0.35)**	**0.93 (0.26)**	**0.93 (0.21)**	**0.78 (0.28)**	**0.62 (0.23)**
Textual	0.74 (0.41)	0.86 (0.35)	0.89 (0.27)	0.73 (0.34)	0.55 (0.26)
Vetorial	0.80 (0.36)	0.92 (0.27)	0.91 (0.24)	0.75 (0.30)	0.60 (0.24)

A **Busca Híbrida** obteve os melhores resultados em todas as cinco métricas avaliadas. A superioridade sobre a Busca Textual é expressiva (+0.09 em *Recall*), demonstrando que a correspondência exata de palavras-chave (BM25) é insuficiente para capturar a complexidade semântica das consultas jurídicas. A Busca Vetorial apresentou desempenho intermediário, mas a fusão (Híbrida) provou ser capaz de unir a precisão do léxico com a abrangência do vetor.

5.2.3 Janela de Contexto (Top-K)

A análise do volume de documentos recuperados (Tabela 4) revela um comportamento interessante na métrica de Precisão.

Tabela 4. Desempenho por Top-K.

Top-K	Recall	Precision	Faithfulness	Relevancy	Correctness
5	0.70 (0.42)	0.83 (0.37)	0.88 (0.28)	0.71 (0.36)	0.54 (0.27)
10	0.79 (0.37)	0.91 (0.29)	0.92 (0.23)	0.76 (0.30)	0.59 (0.25)
15	0.82 (0.35)	0.93 (0.25)	0.91 (0.24)	0.77 (0.29)	0.60 (0.24)
20	**0.85 (0.33)**	**0.94 (0.23)**	**0.92 (0.22)**	**0.78 (0.27)**	**0.62 (0.23)**

Geralmente, espera-se que aumentar o K reduza a precisão (devido à introdução de ruído). Contudo, observa-se aqui que a *Context Precision* aumentou de 0.83 ($K = 5$) para 0.94 ($K = 20$). Isso ocorre porque, com $K = 5$, o sistema frequentemente falha em recuperar o documento relevante (*Recall* baixo de 0.70), resultando em precisão zero. Ao expandir para $K = 20$, o documento correto é recuperado e, graças à eficácia dos algoritmos de *ranking* (especialmente na

busca híbrida), ele é posicionado no topo, elevando a métrica de precisão.

Nota-se um ganho marginal em *Correctness* ao passar de $K = 15$ (0.60) para $K = 20$ (0.62), sugerindo que janelas maiores são benéficas neste domínio.

5.2.4 Modelo LLM (Custo-Benefício)

A comparação entre os modelos geradores (Tabela 5) apresenta um resultado contra-intuitivo e de alto impacto para a viabilidade econômica do projeto.

Tabela 5. Desempenho por Modelo LLM.

Modelo	Recall	Precision	Faithfulness	Relevancy	Correctness
GPT-4o	0.79 (0.37)	0.90 (0.29)	0.90 (0.26)	0.73 (0.33)	0.58 (0.26)
GPT-4o-mini	0.79 (0.37)	0.90 (0.30)	**0.92 (0.22)**	**0.78 (0.28)**	**0.60 (0.24)**

Como esperado, as métricas de recuperação (*Recall* e *Precision*) são idênticas, pois independem do LLM gerador. Contudo, nas métricas de geração, o modelo **GPT-4o-mini** superou marginalmente o modelo **GPT-4o** em *Faithfulness* (0.92 vs 0.90) e *Answer Relevancy* (0.78 vs 0.73).

Uma análise qualitativa sugere que o modelo menor tende a ser mais conciso e direto, aderindo estritamente às instruções do *system prompt*, enquanto o modelo maior ocasionalmente produz respostas mais elaboradas que, embora corretas, podem divergir da objetividade exigida pela métrica de relevância. Este resultado valida o uso do modelo Mini como a escolha ideal de custo-benefício.

5.3 Análise Inferencial e Teste de Hipóteses

Para validar estatisticamente as diferenças observadas, aplicou-se o teste de Transformação de Postos Alinhados (*Aligned Rank Transform* - ART). Ajustaram-se modelos para mensurar o impacto dos efeitos principais sobre as métricas de qualidade.

A Tabela 6 apresenta o sumário dos resultados, exibindo a estatística F e a significância estatística.

Tabela 6. Sumário da ANOVA (ART): Efeitos Principais.

Métrica	Chunking	Busca	Top-K	Modelo
Recall	92.44***	20.88***	33.81***	0.19^{ns}
Precision	72.32***	61.48***	47.21***	38.44***
Faithfulness	9.22***	12.51***	7.75***	0.17^{ns}
Relevancy	2.23^{ns}	3.87*	3.68*	18.62***
Correctness	35.06***	18.97***	15.98***	5.45*
Legenda: Valores F. *** $p < .001$; * $p < .05$; ^{ns} não sig.				

A análise revela que os fatores são significativos para descrever as métricas obtidas, salvo algumas exceções. Indicando que o ajuste correto dos hiperparâmetros é fundamental para garantir a eficácia do *pipeline*.

5.3.1 Decomposição dos Efeitos (Post-hoc)

Para identificar as diferenças pontuais entre os níveis, realizaram-se contrastes par-a-par com correção de Bonferroni. As tabelas a seguir apresentam a razão t (*t-ratio*) para as comparações de interesse.

Efeito do Chunking A Tabela 7 utiliza a estratégia *Semantic P95* como referência. Valores negativos indicam desempenho inferior à referência.

Tabela 7. Contrastes de Chunking (Ref: Semantic P95).

Contraste	Recall	Precision	Faith	Corr
Recur. 500 - Sem. P95	-16.59***	-7.10***	-9.41***	-9.54***
Recur. 1000 - Sem. P95	-7.40***	+3.36**	$+0.02^{ns}$	-1.80^{ns}
Sem. P75 - Sem. P95	-8.63***	-9.41***	-4.41***	-2.79*
Legenda: Valores t. *** $p < .001$; ** $p < .01$; * $p < .05$; ^{ns} não sig.				

Confirma-se estatisticamente que fragmentar excessivamente o texto (*Recursive 500*) degrada todas as dimensões de qualidade. O *Recursive 1000* apresenta um *trade-off*: ganha em precisão ($t = 3.36$), mas perde significativamente em revocação ($t = -7.40$) comparado ao *Semantic P95*. Como a Corretude final não diferiu estatisticamente ($p > 0.05$), ambas as estratégias são viáveis, com preferência para a Semântica quando a exaustividade (*Recall*) for prioritária.

Efeito do Tipo de Busca A Tabela 8 detalha os ganhos da abordagem Híbrida.

Tabela 8. Contrastes de Busca (t-ratios).

Contraste	Rec	Prec	Faith	Corr
Híbrida - Textual	6.19***	10.08***	4.41***	6.05***
Híbrida - Vetorial	4.71***	1.05^{ns}	4.25***	2.06^{ns}
Vetorial - Textual	1.49^{ns}	9.05***	0.17^{ns}	4.01***
Legenda: *** $p < .001$; * $p < .05$; ^{ns} $p \geq 0.05$.				

A Busca Híbrida é estatisticamente superior à Vetorial nas métricas de suporte: Revocação ($t = 4.71$) e Fidelidade ($t = 4.25$). Embora essa vantagem não se traduza em um ganho estatisticamente significativo de Corretude imediata ($p = 0.12$), a maior robustez na recuperação de evidências justifica sua escolha arquitetural.

Efeito da Janela de Contexto (Top-K) A análise do tamanho da janela (Tabela 9) revelou que expandir o contexto beneficia a recuperação sem prejudicar a geração.

Tabela 9. Contrastes de Top-K (Ref: K=20).

Contraste	Recall	Precision	Correctness
K=5 - K=20	-9.81***	-11.80***	-6.56***
K=10 - K=20	-4.10***	-7.17***	-2.14^{ns}
K=15 - K=20	-3.01*	-6.91***	-1.50^{ns}
Nota: Valores t. *** $p < .001$; * $p < .05$; ^{ns} não sig.			

Observa-se que $K = 20$ é estatisticamente superior a $K = 15$ tanto em Revocação ($t = -3.01, p = 0.016$) quanto em Precisão ($t = -6.91, p < 0.001$). Esse resultado refuta a hipótese de saturação em $K = 15$ para a etapa de recuperação. Contudo, para a métrica final de Corretude, a diferença deixa de ser significativa ($p > 0.05$). Isso indica que, embora o sistema recupere melhores documentos com $K = 20$, o LLM atinge um platô de desempenho na geração da resposta. Recomenda-se, portanto, $K = 20$ como margem de segurança.

Efeito do Modelo LLM O contraste direto confirmou que o modelo *GPT-4o-mini* supera o *GPT-4o* na métrica de Relevância ($t = -4.31, p < 0.001$), indicando maior objetividade, sem perdas estatisticamente significativas nas demais dimensões de qualidade.

6 Conclusão

Este trabalho cumpriu seu objetivo geral ao desenvolver e validar uma metodologia experimental rigorosa para a otimização de sistemas RAG no domínio jurídico. Através de um experimento fatorial com $N = 3.840$ observações, superou-se a abordagem de tentativa e erro, substituindo-a por uma tomada de decisão baseada em evidências estatísticas. A aplicação da ART permitiu isolar os efeitos de cada hiperparâmetro, garantindo a validade das inferências mesmo diante da distribuição não-normal dos dados.

A seguir, detalha-se como cada objetivo específico foi alcançado à luz das novas evidências:

6.1 Cumprimento dos Objetivos Específicos

1. **Avaliar o impacto das estratégias de *Chunking*:** Demonstrou-se que a granularidade é o fator de maior impacto na arquitetura ($F = 35.06$ para Corretude). Os testes confirmaram a hipótese de que fragmentos pequenos (Recursiva 500) degradam severamente o desempenho ($p < 0.001$), rompendo a unidade semântica necessária para a recuperação legal. As estratégias **Semântica (Percentil 95)** e **Recursiva (1000)** mostraram-se estatisticamente equivalentes ($p > 0.05$), indicando que tanto a segmentação por tópicos quanto o uso de janelas amplas são soluções viáveis para mitigar a perda de contexto.

2. **Comparar a eficácia dos métodos de recuperação:** A análise consolidou a **Busca Híbrida (RRF)** como a abordagem mais segura para o domínio jurídico. Embora a métrica de Corretude final tenha apresentado empate técnico com a busca Vetorial ($p > 0.05$), a Híbrida mostrou-se estatisticamente superior na Revocação (*Recall*) e Fidelidade ($p < 0.001$). Conclui-se que o hibridismo atua como um mecanismo de "ancoragem", onde o componente léxico garante a recuperação de termos exatos da lei, prevenindo omissões críticas.

3. **Analisar o comportamento do sistema ao variar o *Top-K*:** Contrariando a expectativa inicial de que janelas muito grandes introduziriam ruído, os resultados mostraram que elevar K para 20 maximizou tanto a Revocação (0.85) quanto a Precisão (0.94). O aumento da precisão com janelas maiores sugere que, em janelas menores ($K = 5$ ou 10), o documento relevante sequer era recuperado; ao ampliar para $K = 20$, o documento é encontrado e corretamente posicionado no topo pelo *reranker* implícito da busca híbrida.

4. **Quantificar o ganho de desempenho entre modelos:** O experimento refutou a hipótese de que modelos massivos são indispensáveis para a tarefa. O modelo **GPT-4o-mini** apresentou desempenho estatisticamente superior ao modelo GPT-4o na métrica de Relevância ($t = 4.31, p < 0.001$) e marginalmente superior em Corretude. Observou-se que o modelo menor tende a ser mais conciso e aderente às instruções de formatação, enquanto o modelo maior ocasionalmente produz respostas prolixas, penalizando sua pontuação.

6.2 Recomendação Arquitetural para o MPES

Com base na síntese dos experimentos, recomenda-se a seguinte configuração de referência, que maximiza a qualidade técnica e a eficiência de custos para o assistente jurídico:

- **Ingestão:** Segmentação Semântica (Percentil 95) ou Recursiva (1000 *tokens*), priorizando blocos que preservem a integridade dos artigos de lei.
- **Busca:** Híbrida (Vetorial + BM25) com fusão por RRF, essencial para garantir alta revocação.
- **Contexto:** Janela de 20 documentos ($Top\text{-}K = 20$), pois não houve degradação por ruído e houve ganho de precisão.
- **Modelo:** GPT-4o-mini, dada sua superioridade em relevância e custo operacional significativamente menor.

6.3 Limitações e Trabalhos Futuros

Apesar do rigor metodológico, este estudo apresenta limitações inerentes ao seu desenho experimental. Uma restrição é a fixação de parâmetros de geração (como a temperatura em $T = 0.1$ e o *prompt* estático), uma decisão de escopo do trabalho, mas que impediu a análise de como a "criatividade"estocástica ou técnicas de engenharia de instrução poderiam refinar a resposta final. A avaliação dependente de um Juiz LLM, embora validada na literatura, também introduz potenciais vieses intrínsecos ao modelo avaliador.

Para superar essas barreiras e avançar no estado da arte, sugerem-se as seguintes linhas de investigação futura:

- **Expansão do Espaço de Hiperparâmetros:** Aplicar o DoE para investigar a influência da Temperatura e de variações na Engenharia de *Prompt* (ex: *Chain-of-Thought*), tratando-os como fatores experimentais e não apenas como constantes de controle.
- **Pré-processamento de Consultas:** Implementar e testar módulos de tratamento da pergunta do usuário antes da etapa de busca, utilizando técnicas de *Query Rewriting* ou expansão de consulta para mitigar o desalinhamento vocabular e melhorar a recuperação.
- **Comparativo com LLM Puro:** Realizar um estudo de base (*baseline*) que contraste o desempenho do sistema RAG otimizado contra o uso direto de modelos (GPT-4o) sem acesso a contexto externo. Isso permitirá quantificar o ganho real de especificidade jurídica e a redução percentual de alucinações que o RAG proporciona ao MPES.
- **Refinamento de Recuperação:** Explorar técnicas de *Re-ranking* neural com *Cross-Encoders* para reordenar a lista de documentos recuperados, buscando um compromisso entre a precisão adicional e a latência de inferência.
- **Validação Humana:** Conduzir testes qualitativos com promotores e servidores do MPES para aferir a utilidade percebida da ferramenta e a fluidez da interação em cenários reais de trabalho.

Declarações complementares

Contribuições dos autores

HCQ contribuiu com a concepção do estudo, desenvolvimento do

software e escrita. DRCD contribuiu com a orientação metodológica e revisão.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de artefatos de pequisa

Os dados e códigos criados neste estudo serão disponibilizados mediante solicitação.

Uso de Inteligência Artificial

Em conformidade com as diretrizes de integridade acadêmica, declara-se que ferramentas de IA Generativa (especificamente a família de modelos *GPT-4*) foram empregadas como componentes estruturais e de suporte a esta pesquisa em quatro frentes principais:

- **Geração de Dados Sintéticos:** Utilizou-se o modelo *GPT-4o* para a construção do *Golden Set*, gerando perguntas e respostas ideais a partir de documentos reais do MPES para garantir a diversidade do conjunto de teste.
- **Motor de Inferência RAG:** Os modelos atuaram como agentes generadores do pipeline RAG, sintetizando respostas jurídicas submetidas ao experimento fatorial.
- **Avaliação Automatizada (*LLM-as-a-Judge*):** O framework *Ragas* empregou o modelo *GPT-4o* como juiz para o cálculo das métricas de fidelidade, relevância e corretude [Es *et al*., 2023].
- **Assistência à Pesquisa e Redação:** Ferramentas de IA foram utilizadas no suporte à revisão bibliográfica, auxílio na estruturação de strings de busca para bases científicas, revisão gramatical e refinamento estilístico de trechos do texto final.

Ressalta-se que o uso de tais ferramentas serviu como suporte técnico e analítico, permanecendo a responsabilidade pela interpretação dos dados, validade das fontes citadas e redação intelectual final integralmente a cargo do autor.

Referências

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Chase, H. (2023). *LangChain: Building applications with LLMs through composability*. Acessado em: 23 jan. 2026.

Cormack, G. V., Clarke, C. L., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

de Aquino, I. V., dos Santos, M. M., Dorneles, C. F., and Carvalho, J. T. (2024). Extracting information from brazilian legal documents with retrieval augmented generation. In *Companion Proceedings of the 39th Brazilian Symposium on Data Bases*, Florianópolis, SC. Artigo aceito para publicação.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64. DOI: 10.1080/01621459.1961.10482090.

Elkin, L. A., Kay, M., Higgins, J. J., and Wobbrock, J. O. (2021). An aligned rank transform procedure for multifactor contrast tests. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 754–768, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3472749.3474784.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2023). Ragas: Automated evaluation of retrieval augmented generation.

Exploding Gradients (2024a). Answer correctness - ragas documentation. `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/answer_correctness/`. Acessado em: 23 jan. 2026.

Exploding Gradients (2024b). Context precision - ragas documentation. `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/context_precision/`. Acessado em: 23 jan. 2026.

Exploding Gradients (2024c). Context recall - ragas documentation. `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/context_recall/`. Acessado em: 23 jan. 2026.

Exploding Gradients (2024d). Faithfulness - ragas documentation. `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/`. Acessado em: 23 jan. 2026.

Exploding Gradients (2025). Answer relevancy - ragas documentation. `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/answer_relevance/`. Acessado em: 23 jan. 2026.

Gao, Y., Xiong, Y., Gao, X., Jia, K., *et al*. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Gradients, E. (2024). *Ragas: Evaluation Framework for your RAG Pipelines*. Exploding Gradients. Acessado em: 23 jan. 2026.

Gupta, S., Ranjan, R., and Singh, S. N. (2024). A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2404.10981*.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ji, Z., Lee, N., Frieske, R., Yu, T., *et al*. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Johnson, A., Smith, B., and Doe, J. (2023). Retrieval-augmented generation in healthcare: A scoping review. *Journal of Biomedical Informatics*, 145:104456.

Jurafsky, D. and Martin, J. H. (2024). *Speech and language processing*. Pearson. 3rd ed. draft.

Kay, M., Elkin, L., and Wobbrock, J. O. (2021). *ARTool: Aligned Rank Transform for Factorial Analysis*. R package version 0.11.1.

Lewis, P., Perez, E., Piktus, A., Petroni, F., *et al*. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Microsoft (2024). *Azure OpenAI Service Documentation*. Acessado em: 23 jan. 2026.

Ministério Público do Estado do Espírito Santo (2026). Consulta de legislação compilada. `https://mpes.legislacaocompilada.com.br/consulta-legislacao.aspx?situacao=1&interno=0`. Acessado em: 23 jan. 2026.

Montgomery, D. C. (2017). *Design and Analysis of Experiments*. John Wiley & Sons, 9th edition.

OpenAI (2023). *tiktoken: A fast BPE tokeniser for use with OpenAI's models*. Acessado em: 23 jan. 2026.

OpenAI (2024). New embedding models and api updates. `https://openai.com/index/new-embedding-models-and-api-updates/`. Acessado em:

23 jan. 2026.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. DOI: 10.3115/1073083.1073135.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramírez, S. (2018). *FastAPI framework, high performance, easy to learn, fast to code, ready for production*. Acessado em: 23 jan. 2026.

Reitz, K. (2023). *Requests: HTTP for Humans*. Acessado em: 23 jan. 2026.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Team, C. (2023). *Chroma: The AI-native open-source embedding database*. Acessado em: 23 jan. 2026.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., *et al*. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wang, X., Wang, Z., Gao, X., Zhang, F., *et al*. (2024). Searching for best practices in retrieval-augmented generation.

Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 143–146, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1978942.1978963.

Wu, S., Xiong, Y., Cui, Y., Wu, H., *et al*. (2025). Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2406.02126*.

Yu, Y., Zhuang, H., Zhang, J., Meng, Y., Ratner, A., Shang, J., Han, J., and He, C. (2024). Rankrag: Unifying context ranking with retrieval-augmented generation in llms.

Zhang, Y., Li, H., and Chen, X. (2024a). The power of chunking: How different strategies affect rag performance. *arXiv preprint arXiv:2409.14924*.

Zhang, Y., Li, H., and Chen, X. (2024b). The power of chunking: How different strategies affect rag performance. *arXiv preprint arXiv:2409.14924*.

Zhao, H., Zhao, C., Li, Y., Zhang, Z., and Liu, G. (2025). Thinking in a crowd: How auxiliary information shapes llm reasoning.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., *et al*. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena.

A Prompts Utilizados no Experimento

Para garantir a reprodutibilidade e transparência deste estudo, apresentamos a seguir a transcrição integral dos *prompts* utilizados.

A.1 Prompt para Geração de Dados Sintéticos (Golden Set)

Este prompt foi utilizado para instruir o modelo GPT-4o a atuar como um analista de dados e gerar as 100 perguntas de teste.

System Instruction: Gerador de Dados Sintéticos

 # CONTEXTO E PERSONA
Você é um analista de dados e especialista em legislação, encarregado de criar um rigoroso conjunto de avaliação para um sistema de chatbot de Inteligência Artificial (RAG). Sua tarefa é analisar os documentos legais do Ministério Público do Espírito Santo (MP-ES) fornecidos abaixo e gerar um conjunto diversificado e desafiador de pares de pergunta e resposta.

 # SIMULAÇÃO DE COMPORTAMENTO DO USUÁRIO
Formule as perguntas do ponto de vista de um usuário real que **NÃO SABE** em qual documento ou link a resposta se encontra.

- As perguntas devem ser naturais e gerais.
- **NUNCA** inclua o link da fonte ou qualquer nome de arquivo no texto da pergunta.
- A fonte (o link) deve ser usada APENAS para preencher o campo `"source"` no JSON de saída.

 # TAREFA PRINCIPAL
Com base EXCLUSIVAMENTE no conteúdo dos documentos fornecidos, gere um total de **40 perguntas**, divididas igualmente entre as 4 categorias a seguir (10 perguntas por categoria):

1. **FATUAL:** "Qual o prazo para a interposição de recursos?".
2. **SÍNTESE:** "Quais são as diferenças entre os procedimentos de férias e licença-prêmio?".
3. **PROCEDIMENTAL:** "Descreva o procedimento para solicitar o teletrabalho.".
4. **AGULHA NO PALHEIRO:** "Qual o valor exato do auxílio-alimentação citado?".

 # FORMATO DA SAÍDA
Sua resposta final deve ser APENAS um array JSON válido, sem introduções, seguindo o esquema abaixo.

Esquema JSON Esperado (Output)

```
[
  {
    "id": "ID_UNICO",
    "category": "NOME_DA_CATEGORIA",
    "question": "TEXTO DA PERGUNTA (GERAL)",
    "ground_truth_answer": "RESPOSTA COMPLETA E IDEAL",
    "evidence_quote": "CITACAO EXATA DO DOCUMENTO",
    "source": "http://link.da.fonte.gov.br/documento"
  }
]
```

Nota: O conteúdo dos documentos reais foi inserido dinamicamente no final do prompt durante a execução do script.

A.2 System Prompt do Assistente RAG

Este é o comando de instrução (*system role*) passado ao modelo durante a fase de inferência.

System Role: Assistente Jurídico MPES

"Você é um Assistente Jurídico Virtual do Ministério Público do Estado do Espírito Santo (MPES). Sua função é auxiliar servidores e cidadãos a entenderem os atos normativos da instituição.
Diretrizes de Resposta:

- Baseie sua resposta **EXCLUSIVAMENTE** no contexto fornecido (trechos recuperados). Não use seu conhecimento prévio para inventar leis.
- Sempre cite a fonte normativa (ex: 'Conforme o Art. 5º da Portaria Nº 123...').
- Se o contexto não contiver a informação necessária, diga: *'Desculpe, o contexto recuperado não contém informações suficientes sobre este tema.'* Não tente adivinhar.
- Mantenha um tom formal, objetivo e impessoal."