

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Desenvolvimento de Benchmark e Análise Experimental de RAG na Recuperação de Informações do Diário Oficial

Gustavo Almeida Valentim

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia da Computação

Orientador
Prof. Dr. Thiago de Paulo Faleiros

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Desenvolvimento de Benchmark e Análise Experimental de RAG na Recuperação de Informações do Diário Oficial

Gustavo Almeida Valentim

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia da Computação

Prof. Dr. Thiago de Paulo Faleiros (Orientador)
CIC/UnB

Prof. Dr.a Roberta Barbosa Oliveira
CIC/UnB

Prof. Me. Raul Carvalho de Souza
CIC/UnB

Prof. Dr. Guilherme Novaes Ramos
Coordenador do Bacharelado em Engenharia da Computação

Brasília, 10 de Dezembro de 2025

Dedicatória

Dedico a Deus, que me deu força, e à minha família, especialmente aos meus pais e irmãos, que me deram apoio. Com vocês ao meu lado, tudo se torna possível.

Agradecimentos

Agradeço primeiramente a Deus, pois nada sou sem Ele.

Aos meus pais, Givanildo e Soraya, e aos meus irmãos, Felipe e Guilherme, pela base sólida, pelo sacrifício e pelo incentivo incondicional. Agradeço a cada um de vocês por darem sentido à minha vida. Obrigado por tudo, especialmente aos meus pais, por todo o trabalho e esforço que permitiram que eu chegasse até aqui. Estendo também minha gratidão a toda a minha família.

À Maria Eduarda, pelo apoio emocional constante, pela paciência durante minhas longas horas de estudo e por tornar essa caminhada mais leve. Sua presença e seu carinho foram essenciais para a conclusão desta etapa; você foi fundamental para que ela fosse completa.

Ao meu orientador, Prof. Dr. Thiago de Paulo Faleiros, pela oportunidade e pela confiança. Agradeço pelas direções precisas, pela paciência com minhas dúvidas e por ter me guiado com rigor científico na construção deste trabalho.

À Universidade de Brasília (UnB) e ao corpo docente do curso de Engenharia da Computação, por me proporcionarem o ambiente e o conhecimento necessários para minha formação profissional.

Por fim, agradeço aos amigos e colegas que, de alguma forma, contribuíram para a realização deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A recuperação de informações em documentos públicos enfrenta a escassez de *benchmarks* consolidados para a avaliação de tecnologias. O objetivo central deste trabalho é mitigar essa lacuna através do desenvolvimento e disponibilização do *benchmark* `DiarioOficial-Contratos-BR-GT`, um conjunto validado contendo 3.324 pares de perguntas e respostas sobre contratos públicos. Utilizando este material como base de testes, foi conduzida uma análise experimental aplicando técnicas de Recuperação Aumentada por Geração (RAG) com variadas estratégias de segmentação e busca. Os resultados dos experimentos demonstram o desempenho do sistema frente aos dados coletados. O trabalho contribui, assim, com a entrega deste *benchmark*, disponibilizando um recurso estruturado para testes de recuperação de informação no Diário Oficial.

Palavras-chave: RAG, Processamento de Linguagem Natural, Diário Oficial, Busca Semântica, Recuperação de Informação

Abstract

Information retrieval from public documents faces a scarcity of consolidated benchmarks for technology evaluation. The central objective of this work is to mitigate this gap through the development and release of the `DiarioOficial-Contratos-BR-GT` benchmark, a validated set containing 3,324 question-answer pairs regarding public contracts. Using this material as a testing ground, an experimental analysis was conducted applying Retrieval-Augmented Generation (RAG) techniques with various segmentation and retrieval strategies. The experimental results demonstrate the system's performance against the collected data. This work thus contributes by delivering this benchmark, providing a structured resource for information retrieval tests in the Official Gazette.

Keywords: RAG, Natural Language Processing, Official Gazette, Semantic Search, Information Retrieval

Sumário

1 Introdução **1**
1.1 Contextualização 1
1.2 Objetivo 2
1.2.1 Estrutura do Documento 2

2 Fundamentação Teórica **4**
2.1 Recuperação Aumentada e Processamento Vetorial 4
2.1.1 Indexação e Busca Vetorial 5
2.1.2 Estratégias de Segmentação 5
2.1.3 Vizinhança vs. Sobreposição 5
2.2 Dados do Diário Oficial do Distrito Federal 6
2.2.1 DODFMiner e Dados Estruturados 6

3 Metodologia **10**
3.1 Ambiente de Execução e Ferramentas 10
3.2 Construção e Validação do Benchmark 11
3.2.1 Processo de Validação 12
3.2.2 Perguntas e Respostas 13
3.3 Arquitetura Experimental do Sistema RAG 18
3.3.1 Criação da Base de Conhecimento Vetorial 18
3.3.2 Fluxo de Recuperação e Geração de Respostas 19
3.4 Processo de Avaliação 20

4 Resultados **23**
4.1 Resultados do Corpus de Ajuste 23
4.2 Análise Agregada por Hiperparâmetro 26
4.3 Definição das Configurações Finais 27
4.3.1 Configuração Final para Base A (Consultas Amplas) 27
4.3.2 Configuração Final para Base B (Consultas Específicas) 28
4.4 Resultados do Corpus de Avaliação 28

4.5 Análise dos Resultados 29

5 Conclusões e Trabalhos Futuros **31**

5.1 Síntese das Contribuições e Validação dos Objetivos 31

5.2 Limitações do Estudo 32

5.3 Trabalhos Futuros 32

Referências **33**

Lista de Figuras

2.1 Exemplo real de um registro estruturado da base de dados (DODFMiner). Nota-se a estruturação das entidades, mas a ausência de campo indicando o arquivo PDF de origem. 7
2.2 Recorte do Diário Oficial (DODF 195 11-10-2019 INTEGRA.PDF) original correspondente ao extrato da Figura 2.1. 8
2.3 Diagrama do Processo de União e Validação entre a Base de Extratos Estruturada e o Corpus de Documentos (PDFs). 9

3.1 Fluxo de validação e construção do *Ground Truth*, detalhando as etapas de processamento e os critérios de filtro. 13
3.2 Prompts exatos (sistema, Base A e Base B) utilizados para a geração sintética de perguntas. 15
3.3 Exemplo real de um par de perguntas (Base A e Base B) geradas para o mesmo extrato do *benchmark*, ilustrando a diferença de especificidade. . . . 17
3.4 Diagrama da Arquitetura RAG experimental: Criação da Base Vetorial (esquerda) e Fluxo de Recuperação (direita). 18
3.5 Prompt do sistema usado para instruir o LLM. 20
3.6 Fluxo de registro dos resultados da avaliação e visualização da estrutura de dados (`.jsonl`) gerada. 21

Lista de Tabelas

3.1 Principais Bibliotecas e Ferramentas de Software Utilizados 11
3.2 Especificações Técnicas dos Modelos Utilizados (via Ollama) 11
3.3 Distribuição de extratos e perguntas por documento PDF no *ground truth* 16
3.4 Detalhamento da geração de perguntas (3 variações por base) 18
3.5 Caracterização dos Corpus de Documentos para Ajuste e Avaliação 19
3.6 Hiperparâmetros testados no Corpus de Ajuste 21

4.1 Resultados completos dos 64 testes no Corpus de Ajuste. (Acurácia arredondada para duas casas decimais) 23
4.2 Resumo Agregado (Médias) - **Base A** (Ampla). (Acurácia arredondada para duas casas decimais) 26
4.3 Resumo Agregado (Médias) - **Base B** (Específica). (Acurácia arredondada para duas casas decimais) 26
4.4 Resultados Finais no Corpus de Avaliação Completo (Comparativo Cruzado). Acurácia arredondada para duas casas decimais. 29
4.5 Análise de Cobertura Documental (Capacidade do sistema de acertar ao menos uma pergunta por extrato). 29

Lista de Abreviaturas e Siglas

FAISS Facebook AI Similarity Search.

LLM Large Language Model (Modelo de Linguagem de Grande Porte).

PDF Portable Document Format (Formato de Documento Portátil).

RAG Retrieval-Augmented Generation (Geração Aumentada por Recuperação).

Capítulo 1

Introdução

1.1 Contextualização

O avanço dos Large Language Model (Modelo de Linguagem de Grande Porte) (LLM)s e das técnicas de Processamento de Linguagem Natural redefiniu a interação com documentos, permitindo que a consulta a grandes *corpus* textuais se aproxime da naturalidade da linguagem humana. Esse progresso é evidenciado pelo desenvolvimento de modelos, como a família GPT (OpenAI) [1], Llama (Meta) [2] e arquiteturas baseadas em Transformers [3], que demonstraram capacidade sem precedentes de processar, resumir e raciocinar sobre grandes volumes de informação não estruturada. No entanto, esse avanço tecnológico contrasta com a realidade da administração pública brasileira, onde a publicação de atos oficiais, como no Diário Oficial, ainda depende de documentos não estruturados, técnicos e de difícil processamento, muitas vezes restritos ao formato Portable Document Format (Formato de Documento Portátil) (PDF) e à complexa linguagem jurídico-administrativa.

Este cenário é agravado pela escassez de *benchmarks* e recursos de Processamento de Linguagem Natural voltados especificamente para o domínio jurídico brasileiro. Embora trabalhos recentes como o JurisTCU [4] ofereçam contribuições relevantes, o volume de conjuntos de dados criados e validados no país ainda é pouco. Sem essa referência consolidada, torna-se desafiador distinguir entre a recuperação precisa de informações e a geração de respostas semanticamente coerentes, porém factualmente incorretas (alucinações).

Nesse contexto, a técnica de Retrieval-Augmented Generation (Geração Aumentada por Recuperação) (RAG) [5] é utilizada neste trabalho como uma ferramenta de análise experimental aplicado a este *benchmark*. A implementação proposta adota padrões *open-source* acessíveis para investigar, na prática, o desempenho de modelos em um domínio onde a linguagem do Diário Oficial diverge do português padrão, caracterizando-se por alta tecnicidade e uso frequente de jargões.

A avaliação de arquiteturas RAG impõe desafios metodológicos, pois métricas tradicionais de texto não capturam a ligação entre recuperação e geração [6]. É complexo diferenciar erros de busca (quando a fonte relevante é ignorada) de alucinações na geração. Portanto, são indispensáveis *benchmarks* rotulados para validar se a informação correta foi recuperada, garantindo precisão factual.

Este trabalho propõe-se, portanto, a abordar o problema da falta de métricas. O foco central reside na estruturação e disponibilização de um método de avaliação confiável para este domínio. Ao restringir o escopo ao segmento de contratos públicos e submeter um sistema RAG a este novo *benchmark*.

1.2 Objetivo

O objetivo geral deste trabalho é:

Desenvolver um *benchmark* (conjunto de dados de avaliação) específico para contratos do Diário Oficial do Distrito Federal e utilizá-lo para realizar uma análise experimental de um sistema de Recuperação Aumentada por Geração (RAG), visando mensurar o desempenho real e identificar os gargalos da recuperação de informação neste domínio.

Os objetivos específicos deste trabalho são:

- Criar e curar um *benchmark* personalizado, composto por duas bases de perguntas (uma centrada em objetos de contratos e outra baseada no extrato completo), estabelecendo um padrão de referência (*ground truth*);
- Definir uma metodologia de avaliação baseada no critério de correspondência exata (*exact match*), priorizando a precisão na recuperação de dados sensíveis como valores monetários;
- Implementar um *pipeline* funcional de RAG para processamento de documentos PDF e recuperação de informações contratuais, servindo como ambiente de teste para o *benchmark* criado;
- Executar experimentos controlados variando parâmetros de busca (como *chunking*, *Top-K* e estratégias de vizinhança) para avaliar a robustez do *benchmark* e como diferentes configurações impactam a recuperação.

1.2.1 Estrutura do Documento

A monografia está organizada conforme os seguintes capítulos:

- **Capítulo 2: Fundamentação Teórica** Apresenta os conceitos essenciais necessários para o entendimento do experimento, abordando a arquitetura RAG, o processamento de dados do Diário Oficial (incluindo ferramentas como o DODFMiner).
- **Capítulo 3: Metodologia** Descreve o processo de construção e validação do *benchmark*, a implementação da arquitetura experimental de RAG e a configuração dos testes realizados.
- **Capítulo 4: Resultados e Discussão** Apresenta os dados obtidos nos experimentos, analisa o comportamento das diferentes configurações de busca e discute os motivos das falhas e acertos do sistema frente ao *benchmark*.
- **Capítulo 5: Conclusão e Perspectivas Futuras** Sintetiza as contribuições, com ênfase na entrega do *benchmark*, aponta as limitações metodológicas e sugere caminhos para a evolução de tecnologias nacionais na área.

Capítulo 2

Fundamentação Teórica

Este capítulo estabelece a base teórica necessária para a compreensão dos experimentos e da engenharia de dados desenvolvida neste trabalho. A estrutura organiza-se em duas frentes principais: os conceitos de arquitetura de sistemas inteligentes e os fundamentos metodológicos para a construção do *benchmark*.

A **Seção 2.1** aborda a Recuperação Aumentada por Geração (RAG), detalhando o funcionamento dos LLMs, a representação vetorial (*embeddings*) e as estratégias de indexação e segmentação. Já a **Seção 2.2** fundamenta os pré-requisitos para a construção de *benchmarks*, contextualizando o papel do projeto DODFMiner, cuja base estruturada serviu de suporte e guia para o desenvolvimento deste trabalho.

2.1 Recuperação Aumentada e Processamento Vetorial

A técnica de Recuperação Aumentada por Geração (RAG)[5], representa uma evolução nas arquiteturas de Processamento de Linguagem Natural. Ela resolve uma limitação crítica dos LLMs tradicionais: a dependência exclusiva do conhecimento adquirido durante o treinamento. No RAG, o modelo generativo é acoplado a um componente de recuperação de informação (*Retriever*), permitindo que o sistema consulte uma base de conhecimento externa e atualizada antes de gerar uma resposta.

No contexto deste trabalho, isso se traduz na capacidade de acoplar as informações do Diário Oficial como fonte externa. Assim, mesmo que o LLM não tenha sido treinado especificamente com os dados recentes ou específicos do DODF, o RAG permite que ele processe essas informações, desde que o mecanismo de busca forneça o contexto correto.

Para viabilizar essa busca semântica, o texto passa primeiramente por um processo de tokenização. Um *token* é a unidade fundamental de processamento para o modelo,

podendo representar uma palavra inteira, parte de uma palavra ou até um caractere, dependendo do vocabulário do modelo.

Após a tokenização, esses fragmentos são convertidos em vetores numéricos de alta dimensão, conhecidos como ***embeddings***. Modelos modernos treinados com grandes volumes de dados e técnicas de aprendizado profundo (*deep learning*) aprendem a posicionar frases (ou sequências de tokens, chamadas de *chunks*) com significados similares em regiões próximas dentro de um espaço vetorial multidimensional. Embora tanto o LLM quanto o modelo de *embedding* operem sobre esses princípios vetoriais, o RAG explora justamente essa propriedade geométrica: a capacidade de calcular a distância matemática entre a pergunta do usuário e os documentos disponíveis para recuperar os trechos mais relevantes semanticamente.

2.1.1 Indexação e Busca Vetorial

A eficiência da recuperação em grandes volumes de dados depende de estruturas de indexação otimizadas. O **Facebook AI Similarity Search (FAISS)** (Facebook AI Similarity Search) [7] é a biblioteca para esta tarefa, permitindo a busca por similaridade (utilizando métricas como a distância euclidiana ou produto escalar) em escalas de bilhões de vetores. Diferente de bancos de dados tradicionais que buscam palavras-chave exatas, o *Vector Store* busca a proximidade semântica, permitindo encontrar trechos relevantes mesmo que não compartilhem o mesmo vocabulário da pergunta.

2.1.2 Estratégias de Segmentação

Um componente crítico nesta arquitetura é a Segmentação ou *Chunking*. Como os modelos de *embedding* e os LLMs possuem limites rígidos de entrada (janela de contexto), documentos longos precisam ser divididos em fragmentos menores para processamento.

A estratégia de segmentação afeta diretamente a performance do sistema. Fragmentos muito pequenos podem perder o contexto necessário para responder à pergunta, enquanto fragmentos muito grandes introduzem ruído excessivo. A abordagem mais comum, conhecida como *Recursive Character Text Splitter* [8], tenta dividir o texto preservando a estrutura semântica, priorizando quebras em parágrafos e frases antes de recorrer à quebra de palavras.

2.1.3 Vizinhança vs. Sobreposição

Para mitigar o risco de cortar uma informação vital no meio (ex: separar o objeto do valor de um contrato), a literatura tradicional sugere o uso de **sobreposição** (*overlap*), onde o final de um bloco repete o início do seguinte.

No entanto, em documentos altamente densos e repetitivos, o *overlap* pode introduzir redundância indesejada no contexto do LLM. Uma alternativa superior é a estratégia de Recuperação de Vizinhança (*Neighborhood Retrieval*). Nesta abordagem, o sistema recupera o fragmento que melhor atende à busca vetorial, mas entrega ao LLM uma janela expandida que inclui os fragmentos vizinhos (imediatamente anteriores e posteriores). Isso garante que o contexto completo seja preservado sem a necessidade de duplicar dados no índice vetorial, oferecendo um equilíbrio ideal entre precisão de busca e completude da informação.

2.2 Dados do Diário Oficial do Distrito Federal

O Diário Oficial do Distrito Federal (DODF) é um documento extenso que contém diversas seções, atos administrativos e muitas outras informações. O presente trabalho limita o escopo para focar na avaliação da extração de informação dos extratos de contrato.

Embora o corpus de documentos utilizado nos experimentos contenha os PDFs com a informação completa, o *benchmark* foi construído para testar unicamente o desempenho do sistema RAG sobre esses extratos. O critério de sucesso foi específico: a capacidade do sistema de recuperar o valor monetário exato de um contrato.

Esta escolha visa testar a precisão do RAG em localizar um dado crítico dentro de um contexto textual denso. Dados subjetivos são mais difíceis de avaliar, portanto o escopo limita-se a responder somente o valor exato. Essa é uma maneira eficaz de medir se a busca obtém uma resposta precisa, já que é improvável o sistema acertar o valor ao acaso, facilitando uma avaliação objetiva de acurácia.

O ponto de partida consistiu em duas fontes de dados independentes: um corpus de documentos em formato PDF do Diário Oficial e uma base de dados estruturada contendo informações específicas dos extratos de contrato, como valor, objeto, CNPJ e contratada.

2.2.1 DODFMiner e Dados Estruturados

Uma referência central na extração de dados para este domínio é o **DODFMiner**, ferramenta proposta por Guimarães et al. [9]. Este trabalho desenvolveu uma abordagem automatizada para o processamento de atos oficiais, operando em três etapas distintas: pré-processamento e segmentação dos documentos, classificação dos atos textuais via expressões regulares e Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - NER) utilizando algoritmos de aprendizado de máquina (*Conditional Random Fields* - CRF). O resultado desse processo é a disponibilização de dados em formatos estruturados. A Figura 2.1 apresenta um registro real extraído da base utilizada. É possível observar a

riqueza dos campos identificados, como número do contrato e valor, mas também as limitações, como a presença de valores nulos (`NaN`) e a ausência de metadados que indiquem a posição exata do texto no PDF original.


```
{
  "texto": "EXTRATO DO CONTRATO DE PRESTACAO DE SERVICOS No 03/2019\n
   NOS TERMOS DO PADRAO CEB\nPROCESSO: 00145-00000647/2019-23; DAS PARTES:
   ADMINISTRACAO REGIONAL DO RECANTO DAS EMAS - RA-XV x CEB DISTRIBUICAO S.A...
   DO OBJETO: Prestacao e utilizacao de servico publico de energia...
   DO VALOR: O valor anual estimado do contrato e de R$ 133.242,81...",
  "numero_contrato": "03/2019",
  "processo_gdf": "00145-00000647/2019-23",
  "orgao_contratante": "ADMINISTRACAO REGIONAL DO RECANTO DAS EMAS - RA-XV",
  "entidade_contratada": "CEB DISTRIBUICAO S.A",
  "objeto_contrato": "Prestacao e utilizacao de servico publico de energia
   eletrica entre a distribuidora e o consumidor...",
  "valor_contrato": "133.242,81",
  "unidade_orcamentaria": "09117",
  "programa_trabalho": "04.128.6001.8517.0055",
  "natureza_despesa": "3.3.90.39",
  "fonte_recurso": "120",
  "nota_empenho": "2019NE00204",
  "data_assinatura_contrato": "08 de outubro de 2019",
  "vigencia_contrato": "O Contrato tera vigencia de 12 (doze) meses...",
  "cnpj_entidade_contratada": NaN,
  "codigo_siggo": NaN,
  "cnpj_orgao_contratante": NaN,
  "nome_responsavel": NaN
}
```


Figura 2.1: Exemplo real de um registro estruturado da base de dados (DODFMiner). Nota-se a estruturação das entidades, mas a ausência de campo indicando o arquivo PDF de origem.

Para ilustrar o desafio de alinhamento, a Figura 2.2 exibe o recorte do documento original correspondente ao registro acima. Nota-se a linguagem técnica do texto e a ausência de uma maneira direta de ligar o PDF ao extrato do DODFMiner, visto que campos como CNPJ, objeto e data podem se repetir em diversos contratos, dificultando a localização direta.

SECRETARIA EXECUTIVA DAS CIDADES
ADMINISTRAÇÃO REGIONAL DO RECANTO DAS EMAS

EXTRATO DO CONTRATO DE PRESTAÇÃO DE SERVIÇOS Nº 03/2019
NOS TERMOS DO PADRÃO CEB

PROCESSO: 00145-00000647/2019-23; DAS PARTES: ADMINISTRAÇÃO REGIONAL DO RECANTO DAS EMAS - RA-XV x CEB DISTRIBUIÇÃO S.A - CEB; DO FUNDAMENTO LEGAL: Inexigibilidade de Licitação com base no caput do artigo 25 da Lei 8.666/1993; DO OBJETO: Prestação e utilização de serviço público de energia elétrica entre a distribuidora e o consumidor, de acordo com o Projeto Básico que integra o presente termo. DO VALOR: O valor anual estimado do contrato é de R$ 133.242,81 (cento e trinta e três mil, duzentos e quarenta e dois reais e oitenta e um centavos); DA DOTAÇÃO ORÇAMENTÁRIA: Unidade Orçamentária: 09117; Programa de Trabalho: 04.128.6001.8517.0055; Natureza da Despesa: 3.3.90.39; Fonte de Recursos: 100 e 120; Nota de Empenho nº 2019NE00204, no valor inicial de R$ 9.993,20 (nove mil, novecentos e noventa e três reais e vinte centavos); DA VIGÊNCIA: O Contrato terá vigência de 12 (doze) meses a contar da data da assinatura, permitida a prorrogação na forma da Lei vigente. DATA DE ASSINATURA: 08 de outubro de 2019. SIGNATÁRIOS: Pela Administração Regional do Recanto das Emas - RA-XV, CARLOS DALVAN SOARES DE OLIVEIRA, na qualidade de Administrador Regional, e pela contratada, SELMA BATISTA DO RÊGO LEAL, na qualidade de Gerente de Grandes Clientes.

Figura 2.2: Recorte do Diário Oficial (DODF 195 11-10-2019 INTEGRA.PDF) original correspondente ao extrato da Figura 2.1.

Esses dados estruturados serviram como base fundamental para a construção do *benchmark*. No entanto, identificou-se rapidamente um desafio que tornava a avaliação impraticável, visto que havia a ausência de uma ponte que conectasse um extrato da base estruturada à sua exata localização no PDF correspondente. Dispunha-se de uma base rica, mas sem uma chave direta para vincular o Extrato X ao PDF Y. Sem essa ligação, qualquer teste era inconclusivo. Ao receber uma resposta errada, tornava-se difícil diagnosticar se a falha era do sistema RAG em encontrar a informação ou se o extrato de fato não estava presente no corpus de PDFs. Essa incerteza e a desconfiança sobre a integridade do vínculo entre os dados motivaram a etapa metodológica, que consistiu na construção de um fluxo de validação para criar um *benchmark* confiável, restabelecendo a ponte entre o extrato de contrato e o PDF.

A Figura 2.3 esquematiza a composição do *benchmark*. O diagrama evidencia as duas fontes primárias, constituídas pelos dados estruturados e pelos não estruturados (PDF bruto), utilizadas para a consolidação da base final.

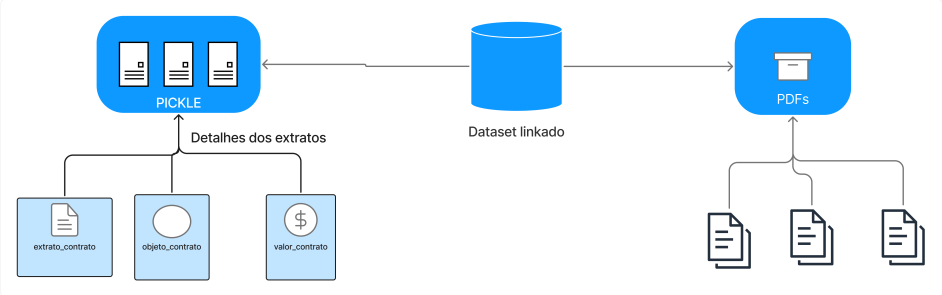


Figura 2.3: Diagrama do Processo de União e Validação entre a Base de Extratos Estruturada e o Corpus de Documentos (PDFs).

Capítulo 3

Metodologia

Este capítulo detalha a metodologia empregada para a construção e avaliação do sistema RAG. A estrutura foi organizada de forma sequencial, partindo dos recursos técnicos utilizados, passando pela preparação dos dados de teste, até a implementação da arquitetura e o protocolo de avaliação.

A **Seção 3.1** estabelece a base técnica do experimento, descrevendo o ambiente computacional e as ferramentas de *software*. A **Seção 3.2** detalha a organização dos conjuntos de dados e a geração sintética das perguntas que compõem o *benchmark* (*ground truth*).

Na sequência, a **Seção 3.3** descreve a implementação da arquitetura RAG experimental, especificando os modelos e as estratégias de busca adotadas. Por fim, a **Seção 3.4** explica o processo automatizado de testes e as métricas utilizadas para mensurar os resultados.

3.1 Ambiente de Execução e Ferramentas

Para a realização dos experimentos, foi utilizada uma máquina específica. Os experimentos principais foram conduzidos em um ambiente Linux (Ubuntu 22.04 LTS) equipado com processador Intel Core i5-9400F, 32GB de RAM e uma GPU NVIDIA GeForce RTX 3060 (12GB VRAM).

O *pipeline* foi desenvolvido utilizando o `Ollama` [10] para a execução local dos modelos. A Tabela 3.1 resume as bibliotecas essenciais que compõem a arquitetura.

Tabela 3.1: Principais Bibliotecas e Ferramentas de Software Utilizados

Ferramenta	**Propósito**
`Ollama` [10]	Servidor local de inferência para LLMs.
`LangChain` [11]	Orquestração do fluxo RAG.
`FAISS` [7]	Indexação e busca vetorial de alta performance.
`pypdf` [12]	Extração de texto dos documentos PDF.
`thefuzz` [13]	*Fuzzy matching* para validação de dados.

A seleção dos modelos foi determinada pela capacidade de execução no ambiente disponível. Durante a fase de implementação, observou-se que arquiteturas maiores excediam o limite de memória ou apresentavam tempos de resposta inviáveis para o projeto. Portanto, optou-se pelos modelos listados na Tabela 3.2, pois apresentaram estabilidade operacional aliada a um desempenho satisfatório nos testes práticos realizados.

Tabela 3.2: Especificações Técnicas dos Modelos Utilizados (via Ollama)

Especificação	**LLM (Llama 3.1)**	**Embedding (BGE-M3)**
Modelo (Ollama ID)	`llama3.1:8b-32k`	`bge-m3:latest`
Parâmetros	8.0 Bilhões (8B)	566.7 Milhões
Quantização	**Q4_K_M (4-bit)**	**F16 (16-bit)**
Dimensões	4096	1024
Contexto	10.000 Tokens	8192 Tokens

Vale destacar o uso da quantização `Q4_K_M` para o LLM. Essa técnica de compressão reduziu o tamanho do modelo para 4.9 GB, viabilizando sua execução na GPU de 12 GB disponível, um compromisso necessário para a execução local.

3.2 Construção e Validação do Benchmark

A construção do *benchmark* final exigiu um processo estruturado de verificação e organização dos dados. Esta seção detalha o fluxo de trabalho adotado, iniciando pelo *pipeline* de validação, desenvolvido para assegurar o vínculo entre os extratos estruturados e os documentos originais (PDFs), e avançando para a definição das dimensões de teste e geração das perguntas que compõem o *corpus* de avaliação.

3.2.1 Processo de Validação

Para resolver o problema, apoiando-se na base estruturada do projeto DODFMiner (Seção 2.2), e criar o *benchmark*, foi desenvolvido um *pipeline* de validação. O objetivo era encontrar na base de extratos estruturada quais deles tinham uma correspondência exata dentro do corpus de PDFs, referenciando inclusive o arquivo para auxiliar na implementação futura do RAG e nos testes. O processo não foi uma busca simples, mas sim um procedimento refinado em múltiplas etapas. Vários *scripts* foram necessários para lidar com a complexidade de unir dados estruturados com texto bruto e ruidoso (os PDFs).

A abordagem implementada utilizou a biblioteca `pdfplumber` [14] para extrair o texto dos PDFs, seguida da normalização via `unicodedata` [15]. Para o cruzamento, aplicou-se *fuzzy matching* (`thefuzz` [13]) baseado na distância de Levenshtein. Em termos simples, essa métrica quantifica a diferença entre dois textos contando o número mínimo de edições (inserções, deleções ou substituições de caracteres) necessárias para transformar um texto no outro.

Contudo, testes preliminares indicaram limitações na abordagem de comparar o texto completo. Ao tentar unir todo o conteúdo do extrato, observou-se que a repetição de termos comuns no vocabulário jurídico (ex: "Brasília", "Lei nº 8.666", "firmado entre") tendia a gerar *scores* de similaridade altos com documentos não relacionados. Esse comportamento evidenciou o risco de falsos positivos baseados apenas em palavras genéricas, motivando o refinamento da estratégia para focar no conteúdo específico do contrato.

Para eliminar esses falsos positivos, o *script* de validação final foi mais rigoroso. A lógica do *pipeline* foi refinada para exigir um critério de múltiplos campos (âncoras). Um extrato só era considerado Válido se o texto do PDF contivesse:

- Obrigatoriamente o valor exato do Valor contrato;
- E, pelo menos DOIS campos da seguinte lista: [Processo, Nota de Empenho, Contratada].

Esta foi uma escolha metodológica. A adição desta regra de campos tornou o filtro mais rigoroso. Isso aumentou os *falsos negativos* (descartando extratos que poderiam ser válidos), mas, em contrapartida, reduziu drasticamente os *falsos positivos*, garantindo a integridade do par pergunta-resposta. aqui cite o gt.png que tem um diagrama explicando como foi realizado esse funil

Além do *pipeline* automatizado, foi realizado um esforço crítico de curadoria manual. O *corpus* inicial de PDFs, embora extenso, não cobria a totalidade da base estruturada. Nos casos em que o algoritmo retornava scores baixos, assumia-se a possibilidade de o documento original estar ausente. Para esses casos, utilizou-se os dados do extrato (como

CNPJ ou trechos únicos do objeto) para realizar buscas manuais diretamente na ferramenta de pesquisa do portal do DODF [16]. Quando a correspondência era confirmada visualmente, o PDF era baixado e integrado ao *corpus* do projeto, permitindo reincorporar extratos que, de outra forma, seriam perdidos.

A prioridade final foi garantir um *dataset* de alta confiança. Foi considerado metodologicamente injusto para a avaliação do RAG incluir no *benchmark* extratos-fonte duvidosos. No final, após os ajustes finos no *pipeline* e a adição manual de PDFs, o objetivo foi alcançado: um *dataset* confiável, onde um resultado ruim indicaria uma falha de implementação (do RAG), e não um dado ausente ou incorreto na base.

Esse fluxo de validação atuou como um funil de qualidade. Dos 1.592 extratos iniciais, o processo resultou em:

- **554 extratos** validados e unidos com sucesso.
- **1.038 extratos** descartados, por não serem localizados no corpus ou por não apresentarem os critérios mínimos para a união.

Este conjunto de 554 extratos válidos tornou-se o *ground truth* sobre o qual toda a avaliação subsequente foi construída. A Figura 3.1 sintetiza visualmente as etapas desse processo de validação e filtragem.

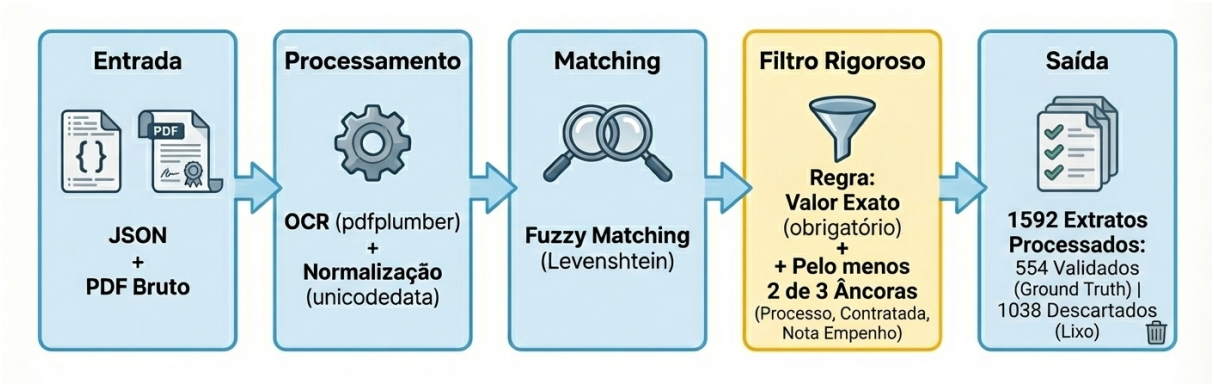


Figura 3.1: Fluxo de validação e construção do *Ground Truth*, detalhando as etapas de processamento e os critérios de filtro.

3.2.2 Perguntas e Respostas

Primeiro, estabeleceu-se uma dimensão de **Tamanho**, dividindo os dados em dois grupos:

- **Conjunto Pequeno (Ajuste):** Um subconjunto de **87 extratos** (oriundos de 36 PDFs). O propósito deste conjunto foi permitir o ajuste ágil de hiperparâmetros (*hyperparameter tuning*).

- **Conjunto Grande (Avaliação):** O conjunto completo de **554 extratos** (oriundos de 58 PDFs), que inclui os dados do conjunto de ajuste. O propósito foi a avaliação final e definitiva do sistema.

A incorporação das amostras de ajuste ao Conjunto de Avaliação teve como objetivo maximizar o volume de dados da etapa final. Essa estratégia assegura que a análise contemple a totalidade do *benchmark* construído (554 extratos), evitando o desperdício de dados válidos e garantindo a avaliação sobre todos os documentos.

Segundo, definiu-se a dimensão de **Tipo de Consulta**, criando dois padrões de perguntas para *ambos* os conjuntos:

- **Base A (Consulta Ampla):** Perguntas geradas focando *apenas* no campo **objeto** (ex: Qual o valor do contrato para prestação de serviços...?). O objetivo metodológico é testar a capacidade do sistema de lidar com ambiguidade em buscas menos específicas.
- **Base B (Consulta Específica):** Perguntas geradas utilizando o extrato inteiro como contexto, fornecendo detalhes como número do processo e nome da empresa. O objetivo é testar a precisão do sistema quando composto de múltiplos pontos diferentes.

Para a geração das perguntas, utilizou-se o modelo **Llama 3.1** [17]. Foram criadas três variações de pergunta para a Base A e três para a Base B para cada um dos extratos processados. Os *prompts* utilizados para instruir o modelo são detalhados na Figura 3.2, ao passo que a Tabela 3.3 apresenta a distribuição final de extratos por documento (PDF).

```
SYSTEM_PROMPT_GERACAO = """
Você é um assistente de IA especializado em criar dados de treinamento para um sistema de RAG
(Retrieval-Augmented Generation).
Sua tarefa é gerar perguntas com base no texto fornecido, seguindo estritamente as instruções do
usuário.
Não faça comentários. Gere apenas o output solicitado.
"""
}

PROMPT_DA_BASE_A = """
Dado o objeto de contrato abaixo, gere exatamente 3 perguntas diferentes e específicas sobre o
valor do contrato.
IMPORTANTE:
- NÃO faça perguntas genéricas, não use expressões como "conforme contrato" ou "segundo extrato".
- Cada pergunta deve ser detalhada o bastante para que, ao pesquisar em um sistema de buscas,
seja possível encontrar apenas este contrato.
- O output deve ser apenas a lista de perguntas (**não escreva explicação ou comentários**).

Exemplo correto:
1. Qual é o valor total da locação do imóvel destinado ao Núcleo de Atendimento à Família e aos
Autores de Violência Doméstica?
2. Qual é o valor total do contrato para a aquisição das tendas?
3. Qual é o valor total da aquisição de gêneros alimentícios não perecíveis, especificamente
Feijão Carioca in natura, para o Programa de Alimentação Escolar do Distrito Federal?

Agora gere perguntas para o extrato a seguir:

Extrato:
{text}

Perguntas:
"""

PROMPT_DA_BASE_B = """
Dado o extrato de contrato abaixo, gere exatamente 3 perguntas diferentes e específicas sobre o
valor do contrato.
IMPORTANTE:
- Cada pergunta GERADA deve obrigatoriamente citar UM ou MAIS dos seguintes: número do contrato,
número do processo, nome da parte contratante ou contratada, objeto da contratação OU data de
assinatura.
- NÃO faça perguntas genéricas, não use expressões como "conforme contrato" ou "segundo extrato".
- Cada pergunta deve ser detalhada o bastante para que, ao pesquisar em um sistema de buscas,
seja possível encontrar apenas este contrato.
- O output deve ser apenas a lista de perguntas (**não escreva explicação ou comentários**).

Exemplo correto:
1. Qual é o valor total do contrato nº 01/2016 estabelecido entre a Administração Regional do
Park Way e a FUNAP para a contratação de 12 sentenciados?
2. No processo 305.000.016/2016 envolvendo a Administração Regional do Park Way, qual o valor
destinado à contratação dos serviços prestados pela FUNAP?
3. A assinatura do contrato realizada em 06 de Outubro de 2019 entre Park Way e FUNAP prevê qual
valor total?

Agora gere perguntas para o extrato a seguir:

Extrato:
{text}

Perguntas:
"""
```

Figura 3.2: Prompts exatos (sistema, Base A e Base B) utilizados para a geração sintética de perguntas.

Tabela 3.3: Distribuição de extratos e perguntas por documento PDF no *ground truth*

Nome do PDF	**Nº de Extratos Válidos**	**Nº de Perguntas**
DODF 215 18-11-2021 INTEGRA.pdf	61	183
DODF 044 07-03-2022 INTEGRA.pdf	52	156
DODF 195 11-10-2019 INTEGRA.pdf	27	81
DODF 244 24-12-2019 INTEGRA.pdf	26	78
DODF 011 16-01-2019 INTEGRA.pdf	20	60
DODF 094 21-05-2019 INTEGRA.pdf	19	57
DODF 226 28-11-2019 INTEGRA.pdf	19	57
DODF 236 20-12-2021 INTEGRA.pdf	19	57
DODF 232 14-12-2021 INTEGRA.pdf	18	54
DODF 224 02-12-2021 INTEGRA.pdf	17	51
DODF 129 10-07-2018 INTEGRA.pdf	14	42
DODF 227 29-11-2019 INTEGRA.pdf	14	42
DODF 181 24-09-2021 INTEGRA.pdf	13	39
DODF 233 15-12-2021 INTEGRA.pdf	11	33
DODF 102 31-05-2019 INTEGRA.pdf	10	30
DODF 191 08-10-2021 INTEGRA.pdf	9	27
DODF 238 16-12-2019 INTEGRA.pdf	9	27
DODF 109 11-06-2019 INTEGRA.pdf	9	27
DODF 126 07-07-2021 INTEGRA.pdf	9	27
DODF 123 02-07-2021 INTEGRA.pdf	8	24
DODF 226 06-12-2021 INTEGRA.pdf	8	24
DODF 242 20-12-2019 INTEGRA.pdf	7	21
DODF 125 06-07-2021 INTEGRA.pdf	7	21
DODF 011 17-01-2022 INTEGRA.pdf	7	21
DODF 236 12-12-2019 INTEGRA.pdf	7	21
DODF 224 26-11-2019 INTEGRA.pdf	6	18
DODF 113 18-06-2021 INTEGRA.pdf	6	18
DODF 173 14-09-2021 INTEGRA.pdf	6	18
DODF 132 16-07-2019 INTEGRA.pdf	6	18
DODF 055 22-03-2022 INTEGRA.pdf	6	18
DODF 245 27-12-2018 INTEGRA.pdf	6	18
DODF 053 18-03-2022 INTEGRA.pdf	5	15
DODF 152 13-08-2019 INTEGRA.pdf	5	15
DODF 145 02-08-2019 INTEGRA.pdf	5	15
DODF 060 29-03-2019 INTEGRA.pdf	5	15
DODF 019 27-01-2022 INTEGRA.pdf	5	15
DODF 222 22-11-2018 INTEGRA.pdf	5	15
DODF 184 29-09-2021 INTEGRA.pdf	5	15
DODF 138 24-07-2019 INTEGRA.pdf	4	12
DODF 039 25-02-2019 INTEGRA.pdf	4	12
DODF 115 19-06-2019 INTEGRA.pdf	4	12
DODF 110 15-06-2021 INTEGRA.pdf	4	12
DODF 235 17-12-2021 INTEGRA.pdf	4	12
DODF 207 30-10-2019 INTEGRA.pdf	4	12
DODF 071 15-04-2019 INTEGRA.pdf	4	12
DODF 143 30-07-2021 INTEGRA.pdf	4	12
DODF 172 10-09-2018 INTEGRA.pdf	4	12
DODF 108 08-06-2018 INTEGRA.pdf	3	9
DODF 029 09-02-2018 INTEGRA.pdf	3	9
DODF 117 24-06-2021 INTEGRA.pdf	3	9
DODF 124 04-07-2019 INTEGRA.pdf	3	9

Continua na próxima página...

Tabela 3.3 – Continuação

Nome do PDF	Nº de Extratos	Nº de Perguntas
DODF 130 12-07-2019 INTEGRA.pdf	3	9
DODF 107 10-06-2021 INTEGRA.pdf	3	9
DODF 193 09-10-2019 INTEGRA.pdf	2	6
DODF 020 29-01-2019 INTEGRA.pdf	2	6
DODF 102 01-06-2021 INTEGRA.pdf	2	6
DODF 121 30-06-2021 INTEGRA.pdf	2	6
DODF 231 06-12-2018 INTEGRA.pdf	1	3
TOTAL	**554**	**1.662**

Para mostrar a diferença entre as bases, a Figura 3.3 apresenta os dados reais de um extrato do *benchmark* e as respectivas perguntas geradas. Note como a pergunta da Base A foca no objeto, enquanto a Base B utiliza identificadores específicos (número do contrato, partes envolvidas) para reduzir a ambiguidade.

Texto Original (Extrato):

"EXTRATO DO CONTRATO DE PRESTAÇÃO DE SERVIÇOS Nº 03/2019
NOS TERMOS DO PADRÃO CEB
PROCESSO: 00145-00000647/2019-23 DAS PARTES: Administração Regional do Recanto das Emas - RA-XV x CEB Distribuição S.A - CEB DO FUNDAMENTO LEGAL: Inexigibilidade de Licitação com base no caput do artigo 25 da Lei nº 8.666/1993 DO OBJETO: Prestação e utilização de serviço público de energia elétrica entre a distribuidora e o consumidor, de acordo com o Projeto Básico que integra o presente termo. DO VALOR: O valor anual estimado do contrato é de RS 133.242,81 (cento e trinta e três mil, duzentos e quarenta e dois reais e oitenta e um centavos). DA DOTACAO ORÇAMENTÁRIA: Unidade Orçamentária: 09117; Programa de Trabalho: 04.128.6001.8517.0055; Natureza da Despesa: 3.3.90.39; Fonte de Recursos: 100 e 120; Nota de Empenho nº 2019NE00204, no valor inicial de RS 9.993,20 (nove mil, novecentos e noventa e três reais e vinte centavos). DA VIGÊNCIA: O Contrato terá vigência de 12 (doze) meses a contar da data da assinatura, permitida a prorrogação na forma da Lei vigente. DATA DE ASSINATURA: 08 de outubro de 2019. SIGNATÁRIOS: Pela Administração Regional do Recanto das Emas - RA-XV, Carlos Dalvan Soares de Oliveira, na qualidade de Administrador Regional; e pela contratada, Selma Batista do Rego Leal, na qualidade de Gerente de Grandes Clientes"

Pergunta Gerada - Base A (Ampla):

"Quanto é o valor da prestação de serviços públicos de energia elétrica para o consumo residencial entre a distribuidora e o consumidor?"

Pergunta Gerada - Base B (Específica):

"Qual é o valor anual estimado do contrato No 03/2019 entre a Administração Regional do Recanto das Emas e a CEB Distribuição S.A?"

Resposta Esperada (Ground Truth):

"133.242,81"

Figura 3.3: Exemplo real de um par de perguntas (Base A e Base B) geradas para o mesmo extrato do *benchmark*, ilustrando a diferença de especificidade.

A estratégia de gerar múltiplas variações de perguntas visou garantir a robustez do *benchmark*. A Tabela 3.4 detalha a distribuição final das perguntas, evidenciando o volume utilizado nas fases de ajuste e avaliação.

Tabela 3.4: Detalhamento da geração de perguntas (3 variações por base)

Tipo de Pergunta	**Conjunto Ajuste**	**Conjunto Avaliação**
Perguntas Base A (Ampla)	261	1.662
Perguntas Base B (Específica)	261	1.662
Total de Perguntas	**522**	**3.324**

3.3 Arquitetura Experimental do Sistema RAG

Com o *benchmark* construído, esta seção detalha a implementação da arquitetura RAG testada. A arquitetura divide-se em duas etapas principais: a criação da base de conhecimento vetorial e o fluxo de recuperação e geração.

A Figura 3.4 detalha o fluxo operacional da arquitetura. A seção à esquerda esquematiza a etapa de ingestão, na qual os documentos são convertidos em vetores e armazenados, enquanto o fluxo à direita representa o processo de recuperação, onde as consultas do usuário interagem com a base vetorial para gerar a resposta final.

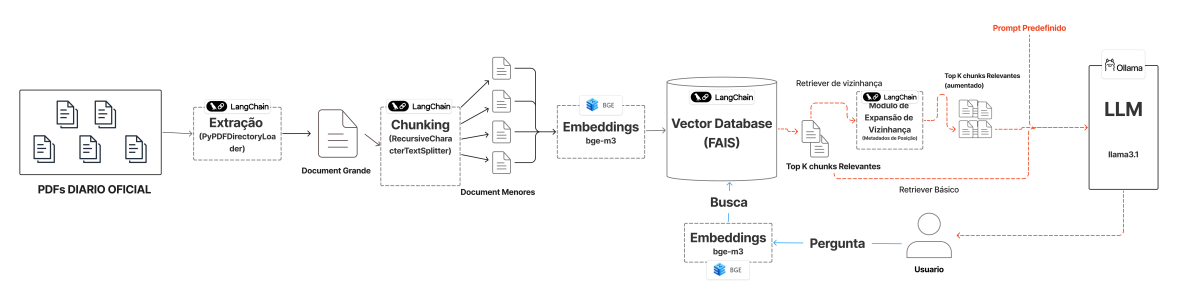


Figura 3.4: Diagrama da Arquitetura RAG experimental: Criação da Base Vetorial (esquerda) e Fluxo de Recuperação (direita).

3.3.1 Criação da Base de Conhecimento Vetorial

A criação da base consistiu no processamento dos *corpus* de documentos (PDFs) definidos na Seção 3.2. A Tabela 3.5 caracteriza o volume de dados processado.

Tabela 3.5: Caracterização dos Corpus de Documentos para Ajuste e Avaliação

Métrica do Corpus	**Ajuste**	**Avaliação**
Documentos PDF (DODF)	36	58
Tamanho em Disco (PDFs)	322 MB	444 MB
Tamanho do Índice FAISS (ex: chunk=600)	225 MB	362 MB
Total de Páginas	2.756	4.213
Total de Palavras (Aprox.)	~4.1 M	~6.6 M
Total de Tokens (cl100k, Aprox.)	~9.7 M	~15.7 M

O processo de ingestão utilizou a biblioteca `LangChain` [11]. A extração de texto foi realizada via `PyPDFDirectoryLoader` [18], seguida pela etapa crítica de segmentação.

Para a segmentação, adotou-se o `RecursiveCharacterTextSplitter` [8], configurado para priorizar a coesão semântica (divisão por parágrafos e quebras de linha). O **tamanho do chunk** foi tratado como um hiperparâmetro variável nos testes.

Diferente das abordagens tradicionais que utilizam sobreposição (*overlap*), este trabalho optou pela estratégia de recuperação de vizinhança. Para habilitá-la, o índice posicional de cada *chunk* foi armazenado como metadado durante a indexação.

A vetorização foi realizada pelo modelo `bge-m3:latest` [19], escolhido por sua capacidade multilíngue e desempenho. Os vetores resultantes foram indexados utilizando a biblioteca FAISS [7], otimizada para busca de similaridade em alta performance.

3.3.2 Fluxo de Recuperação e Geração de Respostas

O fluxo de consulta inicia com a vetorização da pergunta do usuário pelo modelo `bge-m3`. Em seguida, o sistema aplica uma das duas estratégias de recuperação implementadas para comparação:

- **Estratégia 1: Retriever Básico.** Realiza uma busca de similaridade padrão no FAISS, retornando os k *chunks* mais relevantes semanticamente.
- **Estratégia 2: Retriever de Vizinhança.** Realiza a busca dos k *chunks* mais relevantes e, utilizando os metadados posicionais, anexa os *chunks* adjacentes (anterior e posterior) a cada resultado. Nos experimentos, o parâmetro de vizinhança ('n') foi fixado em 1, expandindo o contexto sem introduzir a repetição típica do *overlap*.

O LLM responsável pela geração da resposta foi o `llama3.1:8b-32k` [17], configurado com temperatura 0. O modelo recebe o contexto recuperado e a pergunta original,

sendo instruído por um *prompt* de sistema rigoroso (Figura 3.5) a extrair apenas o valor monetário exato, minimizando alucinações.

```
PROMPT_TEMPLATE_LEGAL = """
Você é um assistente especializado em encontrar valores monetários em contratos públicos do
Diário Oficial brasileiro.

INSTRUÇÕES:
1. Analise cuidadosamente TODOS os contextos fornecidos.
2. Sua tarefa é identificar o valor monetário EXATO que responde à pergunta abaixo.
3. Considere: contrato, processo, partes, objeto ou data mencionadas na pergunta.
4. Só responda se encontrar o valor EXATO em UM dos contextos fornecidos.
5. Não liste múltiplos valores. Traga APENAS a resposta correta, exatamente como aparece no
texto.
6. Se não encontrar o valor em nenhum contexto, responda: "A informação solicitada não foi
encontrada no contexto fornecido."
7. NUNCA invente, deduza ou estime valores.

FORMATO OBRIGATÓRIO DA RESPOSTA:

Se encontrou o valor:
  "Valor encontrado: R$ [valor exato]
   Trecho encontrado: \"[trecho original do documento]\""

Se NÃO encontrou:
  "A informação solicitada não foi encontrada no contexto fornecido."

--------

CONTEXTOS FORNECIDOS:
{context}

PERGUNTA:
{question}
"""
```

Figura 3.5: Prompt do sistema usado para instruir o LLM.

A orquestração final utiliza a cadeia `RetrievalQA` [11] da LangChain com o método `chain_type="stuff"`, que insere todo o contexto recuperado diretamente no *prompt* de entrada do modelo.

3.4 Processo de Avaliação

Para avaliar o desempenho das diferentes configurações da arquitetura RAG (Seção 3.3) contra o *benchmark* (Seção 3.2), foi desenvolvido um **processo de testes automatizado**. Esta **rotina de automação** simula o fluxo de um usuário (ilustrado na Figura 3.4). O processo consiste em iterar sobre cada pergunta do *benchmark* e enviá-la ao sistema. A

resposta gerada pelo LLM é capturada e comparada com a resposta esperada (o *ground truth*).

Todos os resultados dos experimentos foram registrados sistematicamente em arquivos `.jsonl` (JSON Lines), assegurando a rastreabilidade completa de cada iteração de teste. A Figura 3.6 ilustra de forma integrada esse processo, apresentando o fluxo de dados durante a avaliação e destacando a estrutura exata dos campos-chave armazenados em cada registro JSON final.

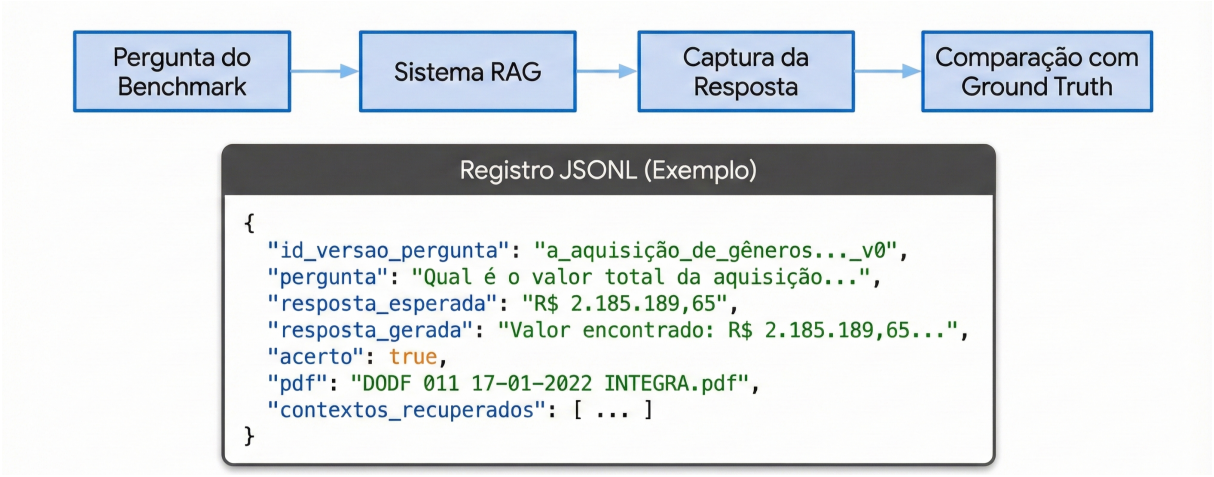


Figura 3.6: Fluxo de registro dos resultados da avaliação e visualização da estrutura de dados (`.jsonl`) gerada.

Este processo viabilizou a execução de uma bateria de **64 testes automatizados** sobre o Corpus de Ajuste, combinando os hiperparâmetros detalhados na Tabela 3.6.

Tabela 3.6: Hiperparâmetros testados no Corpus de Ajuste

Hiperparâmetro	**Valores Testados**
`Tamanho do Chunk`	400, 600, 800, 1000
`Retriever (Vizinhos 'n')`	0 (Básico), 1 (Com Vizinhança)
`Top-K ('k')`	1, 3, 5, 7
`Base de Perguntas`	Base A (Ampla), Base B (Específica)

A métrica de avaliação adotada foi a **Correspondência Exata** (*Exact Match*). Em um domínio de contratos públicos, a precisão absoluta é obrigatória; uma resposta aproximada é considerada incorreta. Portanto, uma resposta só foi validada como acerto se a *string* do valor extraído (após normalização básica) fosse idêntica ao valor do *benchmark*.

Com base nos resultados da fase de ajuste de hiperparâmetros, foram definidas as **configurações finais** para o teste em larga escala. Estas foram executadas contra o **Corpus de Avaliação** completo, gerando os resultados definitivos que são apresentados e discutidos detalhadamente no Capítulo 4.

A implementação completa do código-fonte, incluindo a rotina de avaliação e a arquitetura RAG, está publicamente disponível no GitHub [20]. Recomenda-se a consulta ao repositório para análise detalhada da implementação, visto que esta monografia foca na metodologia e nos resultados.

Capítulo 4

Resultados

Este capítulo apresenta os resultados experimentais e o desempenho do sistema RAG. A avaliação segue a metodologia definida no Capítulo 3, com o objetivo de examinar os dados dos testes de hiperparâmetros, identificar as configurações adequadas e validar o desempenho final do sistema.

O capítulo está estruturado da seguinte forma: a **Seção 4.1** apresenta os resultados brutos dos 64 testes executados sobre o Corpus de Ajuste. A **Seção 4.2** demonstra as tendências gerais, analisando o impacto médio de cada hiperparâmetro. A **Seção 4.3** constitui o núcleo da análise, identificando as duas configurações finais (para a Base A e Base B). A **Seção 4.4** apresenta os resultados dessas configurações selecionadas contra o Corpus de Avaliação. Por fim, a **Seção 4.5** discute as implicações dos resultados.

4.1 Resultados do Corpus de Ajuste

A primeira etapa da experimentação, conforme o Processo de Avaliação (Seção 3.4), consistiu na execução de 64 testes automatizados sobre o Corpus de Ajuste (composto por 261 perguntas). A Tabela 4.1 detalha o desempenho obtido. Nesta análise, a métrica de Acurácia (*Exact Match*) define o sucesso da extração: considerou-se acerto quando o valor gerado pelo modelo, após a normalização, coincidiu exatamente com o dado de referência.

Tabela 4.1: Resultados completos dos 64 testes no Corpus de Ajuste. (Acurácia arredondada para duas casas decimais)

Configuração	**Chunk**	**Vizinho (n)**	**k**	**Base**	**Acurácia**
400_0_1_a_llama3	400	0	1	a	44.06%
400_0_1_b_llama3	400	0	1	b	26.44%
400_0_3_a_llama3	400	0	3	a	41.76%

Tabela 4.1 – Continuação

Configuração	**Chunk**	**Vizinho (n)**	k	**Base**	**Acurácia**
400_0_3_b_llama3	400	0	3	b	19.92%
400_0_5_a_llama3	400	0	5	a	43.68%
400_0_5_b_llama3	400	0	5	b	27.59%
400_0_7_a_llama3	400	0	7	a	46.74%
400_0_7_b_llama3	400	0	7	b	27.20%
400_1_1_a_llama3	400	1	1	a	63.22%
400_1_1_b_llama3	400	1	1	b	52.11%
400_1_3_a_llama3	400	1	3	a	68.58%
400_1_3_b_llama3	400	1	3	b	62.45%
400_1_5_a_llama3	400	1	5	a	68.20%
400_1_5_b_llama3	400	1	5	b	65.13%
400_1_7_a_llama3	**400**	**1**	**7**	**a**	**69.73%**
400_1_7_b_llama3	400	1	7	b	67.82%
600_0_1_a_llama3	600	0	1	a	44.83%
600_0_1_b_llama3	600	0	1	b	26.44%
600_0_3_a_llama3	600	0	3	a	42.15%
600_0_3_b_llama3	600	0	3	b	19.92%
600_0_5_a_llama3	600	0	5	a	44.44%
600_0_5_b_llama3	600	0	5	b	26.05%
600_0_7_a_llama3	600	0	7	a	46.36%
600_0_7_b_llama3	600	0	7	b	28.35%
600_1_1_a_llama3	600	1	1	a	52.49%
600_1_1_b_llama3	600	1	1	b	57.47%
600_1_3_a_llama3	600	1	3	a	56.32%
600_1_3_b_llama3	600	1	3	b	70.88%
600_1_5_a_llama3	600	1	5	a	69.35%
600_1_5_b_llama3	**600**	**1**	**5**	**b**	**82.76%**
600_1_7_a_llama3	600	1	7	a	65.13%
600_1_7_b_llama3	600	1	7	b	80.84%
800_0_1_a_llama3	800	0	1	a	41.38%
800_0_1_b_llama3	800	0	1	b	37.16%
800_0_3_a_llama3	800	0	3	a	52.49%
800_0_3_b_llama3	800	0	3	b	39.46%
800_0_5_a_llama3	800	0	5	a	51.34%
800_0_5_b_llama3	800	0	5	b	38.70%
800_0_7_a_llama3	800	0	7	a	52.87%

Tabela 4.1 – Continuação

Configuração	**Chunk**	**Vizinho (n)**	k	**Base**	**Acurácia**
800_0_7_b_llama3	800	0	7	b	43.68%
800_1_1_a_llama3	800	1	1	a	52.49%
800_1_1_b_llama3	800	1	1	b	63.98%
800_1_3_a_llama3	800	1	3	a	62.45%
800_1_3_b_llama3	800	1	3	b	73.18%
800_1_5_a_llama3	800	1	5	a	65.13%
800_1_5_b_llama3	800	1	5	b	77.01%
800_1_7_a_llama3	800	1	7	a	65.13%
800_1_7_b_llama3	800	1	7	b	77.78%
1000_0_1_a_llama3	1000	0	1	a	32.57%
1000_0_1_b_llama3	1000	0	1	b	35.25%
1000_0_3_a_llama3	1000	0	3	a	39.85%
1000_0_3_b_llama3	1000	0	3	b	39.46%
1000_0_5_a_llama3	1000	0	5	a	37.93%
1000_0_5_b_llama3	1000	0	5	b	44.83%
1000_0_7_a_llama3	1000	0	7	a	38.31%
1000_0_7_b_llama3	1000	0	7	b	43.68%
1000_1_1_a_llama3	1000	1	1	a	38.70%
1000_1_1_b_llama3	1000	1	1	b	54.79%
1000_1_3_a_llama3	1000	1	3	a	45.21%
1000_1_3_b_llama3	1000	1	3	b	68.20%
1000_1_5_a_llama3	1000	1	5	a	44.83%
1000_1_5_b_llama3	1000	1	5	b	72.80%
1000_1_7_a_llama3	1000	1	7	a	47.13%
1000_1_7_b_llama3	1000	1	7	b	73.95%

A Tabela 4.1 revela os dados brutos que servem de base para a análise a seguir. Vale ressaltar que o arquivo individual de cada teste pode ser analisado detalhadamente na pasta `resultados` do repositório do projeto [20].

O pico de performance para a Base A foi de **69.73%** (na configuração `400_1_7_a`), enquanto o pico absoluto para a Base B foi de **82.76%** (na `600_1_5_b`). Esta diferença de comportamento observada na fase de ajuste sugere, inicialmente, que as bases podem demandar configurações distintas, uma hipótese que será verificada na avaliação final do sistema.

4.2 Análise Agregada por Hiperparâmetro

Para identificar tendências gerais, os resultados dos 64 testes foram agregados para analisar o impacto médio de cada hiperparâmetro individualmente. A Tabela 4.2 resume as médias para a Base A, e a Tabela 4.3 para a Base B.

Tabela 4.2: Resumo Agregado (Médias) - **Base A** (Ampla). (Acurácia arredondada para duas casas decimais)

Parâmetro	**Valor**	**Acurácia Média**
`Chunk Size`	400	55.75%
	600	52.63%
	800	55.41%
	1000	40.57%
`Top-K (k)`	1	46.22%
	3	51.10%
	5	53.11%
	7	53.93%
`Retriever`	0 (Básico)	43.80%
	1 (Vizinho)	**58.38%**

Tabela 4.3: Resumo Agregado (Médias) - **Base B** (Específica). (Acurácia arredondada para duas casas decimais)

Parâmetro	**Valor**	**Acurácia Média**
`Chunk Size`	400	43.58%
	600	49.09%
	800	56.37%
	1000	54.12%
`Top-K (k)`	1	44.20%
	3	49.19%
	5	54.36%
	7	55.41%
`Retriever`	0 (Básico)	32.76%
	1 (Vizinho)	**68.82%**

Destas tabelas de médias, podemos extrair as seguintes tendências:

- **Retriever de Vizinhança:** É o fator de maior impacto. Aumentou a acurácia média em mais de 14 pontos percentuais para a Base A e em mais de 36 pontos para a Base B. Isso corrobora a hipótese da Seção 3.3.1 de que a fragmentação de contexto era um problema crítico, confirmando que a estratégia de vizinhança é uma solução eficaz.
- **Top-K (k):** Para ambas as bases, a acurácia média cresce com `k`, atingindo o máximo em `k=7`. Isso sugere que fornecer mais contexto ao LLM é, em geral, benéfico, aumentando a chance de o *chunk* correto ser recuperado.
- **Chunk Size:** Aqui, as médias divergem. A Base A (Ampla) prefere *chunks* menores, com o melhor resultado médio em `400`. A Base B (Específica), por outro lado, prefere *chunks* maiores, com seu pico médio em `800`.

Contudo, como será discutido na próxima seção, essas médias podem ser enganosas. Elas revelam tendências gerais, mas mascaram as interações cruciais entre os hiperparâmetros, que serão analisadas a seguir.

4.3 Definição das Configurações Finais

A análise dos resultados brutos (Tabela 4.1) revela que as médias agregadas mascaram interações cruciais entre os hiperparâmetros. Portanto, a definição das configurações finais priorizou os picos de performance locais em detrimento das médias globais.

Essa seleção também considerou a realidade do **Corpus de Avaliação**: um volume de dados maior exige maior robustez na recuperação. Observou-se que não há uma configuração universal imediata; consultas Amplas (A) e Específicas (B) demonstraram, nesta fase, comportamentos distintos que justificam configurações especializadas.

4.3.1 Configuração Final para Base A (Consultas Amplas)

Geradas apenas com o campo `objeto`, as perguntas da Base A são inerentemente ambíguas e exigem *chunks* precisos para evitar ruído.

- **Análise:** Houve convergência entre as métricas. O *chunk* de 400 liderou tanto na média (55.75%) quanto no pico absoluto de performance da base (**`400_1_7_a`** com **69.73%**).
- **Escolha:** A configuração foi selecionada com base na validação dupla (média e pico), priorizando a granularidade fina.

Configuração Final (Base A):

- `Tamanho do Chunk: 400`
- `Retriever: Vizinhança (n=1)`
- `Top-K (k): 7`

4.3.2 Configuração Final para Base B (Consultas Específicas)

Ao utilizar o extrato completo, a Base B reduz a ambiguidade, mas exige que o sistema conecte evidências dispersas (processo, empresa, data), demandando maior contexto.

- **Análise:** Houve divergência entre a média (que favoreceu *chunk* 800) e o pico de performance. O melhor resultado absoluto do experimento (**82.76%**) foi atingido com *chunk* 600 (`600_1_5_b`), superando significativamente o melhor cenário de 800 (77.78%).
- **Escolha:** Optou-se pelo *chunk* 600 baseando-se no pico de performance. Em relação ao `k`, selecionou-se 7 (em vez do pico local de 5) visando garantir maior robustez de recuperação para o *corpus* de avaliação expandido.

Configuração Final (Base B):

- `Tamanho do Chunk: 600`
- `Retriever: Vizinhança (n=1)`
- `Top-K (k): 7`

4.4 Resultados do Corpus de Avaliação

A etapa final consistiu na validação das duas **Configurações Finais** (definidas na Seção 4.3) contra o **Corpus de Avaliação** completo (554 extratos, 1.662 perguntas por base).

Vale ressaltar a limitação prática desta etapa: devido ao alto custo computacional de re-indexação e inferência no *corpus* expandido, a execução dos 64 testes originais tornou-se inviável. Portanto, a validação restringiu-se às configurações candidatas mais promissoras, assumindo-se o compromisso de buscar a melhor performance provável dentro dos recursos disponíveis.

Os resultados de acurácia (*Exact Match*), incluindo o cruzamento das configurações contra ambas as bases, são apresentados na Tabela 4.4.

Tabela 4.4: Resultados Finais no Corpus de Avaliação Completo (Comparativo Cruzado). Acurácia arredondada para duas casas decimais.

Configuração	**Base Testada**	**Nº Perguntas**	**Acurácia Final**
Final A (400_1_7)	Base A (Ampla)	1.662	44.40%
Final A (400_1_7)	Base B (Específica)	1.662	49.40%
Final B (600_1_7)	Base A (Ampla)	1.662	**48.50%**
Final B (600_1_7)	Base B (Específica)	1.662	**65.40%**

A comparação com a fase de ajuste revela uma redução geral na acurácia, indicando o impacto do aumento do volume de dados (ruído) na precisão do sistema. A Configuração Final B (`600_1_7`) apresentou o melhor desempenho em ambos os cenários, atingindo **65.40%** na Base B e **48.50%** na Base A.

Os dados do *corpus* completo não sustentaram a tendência observada na fase anterior, onde *chunks* menores (400) haviam performado melhor para perguntas amplas. Neste cenário expandido, a configuração com *chunks* de 600 mostrou-se mais consistente, superando a configuração de 400 inclusive na base para a qual esta havia sido estudada.

Complementarmente, a análise de **cobertura documental** (Tabela 4.5) verificou se o sistema era capaz de recuperar o valor correto em *pelo menos uma* das variações de pergunta para cada extrato.

Tabela 4.5: Análise de Cobertura Documental (Capacidade do sistema de acertar ao menos uma pergunta por extrato).

Métrica	**Quantidade**	**Percentual**
Extratos Recuperados ($\geq$ 1 acerto)	**463**	**83,6%**
Extratos Não Recuperados (0 acertos)	91	16,4%

O índice de 83,6% de recuperação indica que a informação alvo está acessível ao modelo na maioria dos documentos. A discrepância entre a acurácia média (65,4%) e a cobertura total (83,6%) sugere uma sensibilidade do sistema à formulação da pergunta no lado do usuário.

4.5 Análise dos Resultados

Esta seção interpreta os dados da Seção 4.4, discutindo as implicações no contexto do Diário Oficial. Os resultados reforçam que atuar neste domínio é uma tarefa complexa: a

padronização textual de múltiplos documentos gera ambiguidade, elevando a dificuldade de busca para o RAG.

Devido ao escopo do projeto, a experimentação restringiu-se ao uso de modelos fundacionais generalistas, especificamente o `bge-m3` [19], não contemplando a comparação com *embeddings* treinados exclusivamente para o domínio jurídico. Os resultados indicam que, embora o `bge-m3` tenha demonstrado estabilidade operacional, sua acurácia sugere um limiar técnico: modelos generalistas, sem ajuste fino (*fine-tuning*) ou técnicas complementares, processam o contexto, mas apresentam limitações de precisão ao interpretar as complexidades legais do *benchmark* proposto.

A comparação entre configurações revelou comportamentos decisivos:

- **Robustez do Contexto:** Contrariando a hipótese inicial, *chunks* médios (600) com vizinhança superaram os menores. O contexto adicional auxiliou na diferenciação de trechos similares.
- **Sensibilidade ao Ruído:** O desempenho inferior do *chunk* de 400 no *corpus* completo sugere que, em ambientes ruidosos, a fragmentação excessiva prejudica a recuperação ao perder conexões semânticas e diluir o contexto.

Apesar do `bge-m3` [19] ter superado outros modelos, a análise indica que a complexidade inerente ao *corpus* é a barreira determinante. A repetitividade do Diário Oficial, onde múltiplos trechos compartilham textos quase idênticos, desafia a diferenciação por nuances (como CNPJ) em modelos não-especializados. Os resultados confirmam que este domínio exige estratégias que priorizem a robustez do contexto (*chunks* maiores e vizinhança).

Vale destacar, contudo, que a cobertura documental foi alta, mesmo com um RAG não especializado. Isso valida a qualidade do *benchmark*: ele cumpriu seu papel de expor o cenário real, demonstrando que a informação é recuperável, mas que a navegação precisa neste domínio exige ferramentas mais refinadas do que as soluções generalistas atuais.

Capítulo 5

Conclusões e Trabalhos Futuros

O objetivo geral deste trabalho foi desenvolver um *benchmark* específico para contratos do Diário Oficial do Distrito Federal e utilizá-lo para realizar uma análise experimental de um sistema de Recuperação Aumentada por Geração (RAG). Conclui-se que este objetivo foi plenamente atingido. A pesquisa entregou o conjunto de dados proposto e validou, através da experimentação, a viabilidade e os desafios de aplicar modelos de linguagem de grande escala e estratégias de recuperação semântica neste domínio complexo.

A seguir, detalha-se o cumprimento de cada objetivo específico estabelecido na introdução.

5.1 Síntese das Contribuições e Validação dos Objetivos

A principal contribuição material deste trabalho é a entrega do *benchmark* [21], preenchendo a lacuna de dados padronizados para o domínio público brasileiro.

Em relação aos objetivos específicos, a validação ocorreu da seguinte forma:

- **Criação do Benchmark:** O objetivo foi cumprido com a construção e curadoria de um conjunto de dados contendo 3.324 perguntas, divididas em bases focadas em objetos e em extratos completos, estabelecendo o *ground truth* necessário para a área.
- **Metodologia de Avaliação:** A definição da métrica foi validada pela aplicação do critério de correspondência exata (*exact match*). Esta abordagem garantiu o rigor necessário para a mensuração de dados sensíveis (como valores monetários), permitindo uma comparação objetiva entre os testes.

- **Implementação do Pipeline RAG:** O fluxo completo foi implementado com sucesso, processando documentos PDF brutos até a geração de respostas. O sistema funcionou como o ambiente de teste estável para todas as validações do *benchmark*.
- **Execução de Experimentos:** A realização de 64 testes controlados, variando *chunking*, *Top-K* e vizinhança, permitiu mapear o comportamento do sistema.

5.2 Limitações do Estudo

Apesar do cumprimento dos objetivos, o estudo observou limitações técnicas e metodológicas importantes. No aspecto de infraestrutura, a escassez de recursos impediu o *fine-tuning* de *embeddings*, mantendo a dependência de modelos generalistas. Somado a isso, o uso de uma janela de contexto restrita a 10.000 tokens no LLM limitou a capacidade do sistema de processar grandes volumes de informação recuperada simultaneamente.

Metodologicamente, destaca-se o viés inerente à construção do *benchmark* via *model-in-the-loop*, aceito dada a inviabilidade da curadoria manual em larga escala. O escopo da avaliação também foi restringido por custos computacionais, focando nas configurações mais promissoras em detrimento de uma busca exaustiva no *corpus* completo. Além disso, o domínio foi limitado a **Extratos de Contrato**, deixando em aberto desafios de outros atos administrativos. Por fim, a métrica *Exact Match*, embora precisa para valores, não avalia a qualidade semântica ou a integridade contextual da resposta textual gerada.

5.3 Trabalhos Futuros

Para a continuidade desta pesquisa, sugere-se priorizar a evolução do *benchmark*. É fundamental expandir o *ground truth* para cobrir todo o espectro do Diário Oficial, incluindo outros atos administrativos além de contratos. Recomenda-se também a criação de perguntas de maior complexidade semântica e a adoção de *frameworks* de avaliação baseados em modelos, como o RAGAS [22], para mensurar a qualidade da resposta além da correspondência exata.

No que tange ao sistema de recuperação, melhorias pontuais podem incluir o treinamento de *embeddings* específicos para o domínio e a adoção de abordagens híbridas de busca.

Em suma, este trabalho reforça a necessidade crítica de se criarem mais *benchmarks* como este para diferentes domínios públicos. O avanço das tecnologias listadas é fundamental para o desenvolvimento de ferramentas que auxiliem efetivamente a população brasileira. Com dados de avaliação sólidos e modelos nacionais robustos será possível entregar soluções que resolvam problemas reais do nosso país.

Referências

[1] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal *et al.*: *Language models are few-shot learners*. Advances in Neural Information Processing Systems, 2020. 1

[2] Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie Anne Lachaux, Timothée Lacroix *et al.*: *Llama: Open and efficient foundation language models*. arXiv preprint arXiv:2302.13971, 2023. 1

[3] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser e Illia Polosukhin: *Attention is all you need*, 2023. `https://arxiv.org/abs/1706.03762`. 1

[4] Caricio, L. *et al.*: *Juristcu: A brazilian portuguese information retrieval dataset with query relevance judgments*. arXiv preprint arXiv:2503.08379, 2025. Disponível em: https://arxiv.org/abs/2503.08379. 1

[5] Karpukhin, Vladimir, Barlas Oguz, Sewon Chen, Patrick Pinter, Patrick Lewis, Ledell Goswami, Sebastian Kementchedjhiev, Wenzhe Chang, Umar Khattak, Rodrigo Nogueira, Yashar Mallya, Aleksander Zayats, Jeon Lee, Tom Kwiatkowski, Wen tau Yih, Nathan Liskov e Mike Lewis: *Retrieval-augmented generation for knowledge-intensive nlp tasks*. arXiv:2005.11401, 2020. 1, 4

[6] Brehme, Lorenz, Thomas Huetter, Patrick Mennig e Andreas K. H. Weiss: *Can llms be trusted for evaluating rag systems? a survey of methods and datasets*, 2025. `https://arxiv.org/abs/2504.20119`. 2

[7] Johnson, Jeff, Matthijs Douze e Hervé Jégou: *Faiss: A library for efficient similarity search and clustering of dense vectors*, 2025. `https://github.com/facebookresearch/faiss`. 5, 11, 19

[8] LangChain: *Langchain text splitters*, 2025. `https://docs.langchain.com/oss/python/integrations/splitters/index#text-splitters`. 5, 19

[9] Guimarães, Gabriel M. C., Felipe X. B. da Silva, Andrei L. Queiroz, Ricardo M. Marcacini, Thiago P. Faleiros, Vinicius R. P. Borges e Luís P. F. Garcia: *Dodfminer: An automated tool for named entity recognition from official gazettes*, 2024. `https://doi.org/10.1016/j.neucom.2023.127064`. 6

[10] Ollama: *Ollama: Run llms locally*, 2025. `https://docs.ollama.com/`. 10, 11

[11] LangChain: *Langchain documentation*, 2025. https://docs.langchain.com/. 11, 19, 20

[12] Wijn, Martin: *pypdf: A pure-python library built as a pdf toolkit*, 2025. https://github.com/py-pdf/pypdf. 11

[13] Cohen, Adam: *thefuzz: Fuzzy string matching in python*, 2025. https://pypi.org/project/thefuzz/. 11, 12

[14] Singer-Vine, Jeremy: *pdfplumber: Plumb a pdf for detailed information*, 2025. https://pypi.org/project/pdfplumber/. 12

[15] Python Software Foundation: *unicodedata — work with unicode databases*, 2025. https://docs.python.org/3/library/unicodedata.html. 12

[16] Distrito Federal: *Diário oficial do distrito federal*, 2025. https://dodf.df.gov.br/dodf. 13

[17] Meta AI: *Introducing meta llama 3.1: Our most capable, efficient and open models yet*, 2024. https://ai.meta.com/blog/meta-llama-3-1/. 14, 19

[18] LangChain: *Langchain document loaders*, 2025. https://docs.langchain.com/oss/python/integrations/document_loaders/index. 19

[19] Shitao, Xiao, Liu Zheng, Zhang Peitian e etc.: *Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity*, 2024. https://arxiv.org/abs/2402.03216. 19, 30

[20] Valentim, Gustavo: *rag-diario-oficial: Implementação rag para o diário oficial*, 2025. https://github.com/valxntim/rag-diario-oficial. 22, 25

[21] Valentim, Gustavo: *Diariooficial-contratos-br-gt: Ground truth para extração de contratos do diário oficial*, 2025. https://huggingface.co/datasets/valxntim/DiarioOficial-Contratos-BR-GT. 31

[22] Es, Shahul, Jithin James *et al.*: *Ragas: Automated evaluation of retrieval augmented generation*, 2024. https://docs.ragas.io/, Accessed: 2025-11-24. 32