FarmacIA: A RAG-Based Pharmaceutical Assistant for Brazilian Portuguese Medicine Leaflets

Sílvio R. Fernandes[1], Pablo Ernesto[1] Joás G. Barros de Sousa[1], Sérgio R. F. Araújo[2],

[1]Departamento de Computação - Universidade Federal Rural do Semi Árido (UFERSA)
Mossoró – RN – Brasil

[2]Departamento de Farmácia - Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

silvio@ufersa.edu.br, {pablo.ernesto, joas.sousa}@alunos.ufersa.edu.br,

sergio.araujo@ufrn.br

Abstract. *This paper presents a retrieval-augmented pharmaceutical assistant designed to answer questions about Brazilian Portuguese medicine leaflets. Seven RAG pipelines were evaluated, combining different strategies for routing, retrieval, evidence fusion, reranking, and safety-oriented validation, using both general-purpose and healthcare-oriented small language models. We built a benchmark with questions covering three medicines, stratified by difficulty and annotated with expected sections and supporting evidence. The evaluation combined retrieval metrics, textual similarity measures, and LLM-as-a-judge assessment. Results show that more controlled pipelines, especially Guarded RAG, improve evidence retrieval and response grounding, although at a higher computational cost. The findings also indicate that the generator model remains crucial for complex questions.*

Resumo. *Este trabalho apresenta um assistente farmacêutico baseado em Retrieval-Augmented Generation para responder perguntas sobre bulas de medicamentos em português do Brasil. Foram avaliados sete pipelines RAG, com diferentes estratégias de roteamento, recuperação, fusão de evidências, reranking e mecanismos de segurança, utilizando modelos de linguagem de pequeno porte gerais e especializados em saúde. O estudo construiu um benchmark com perguntas sobre três medicamentos, estratificadas por dificuldade e anotadas com seções e evidências esperadas. A avaliação combinou métricas de recuperação, similaridade textual e julgamento automatizado por LLM. Os resultados indicam que pipelines mais controlados, especialmente o Guarded RAG, melhoram a recuperação de evidências e a fundamentação das respostas, embora com maior custo computacional. Também se observou que a qualidade do modelo gerador permanece decisiva em perguntas complexas.*

1. Introdução

Nos últimos anos, os Modelos de Linguagem de Grande Escala (*Large Language Models* — LLMs) tornaram-se uma das tecnologias mais relevantes em Processamento de Linguagem Natural, com aplicações crescentes na área da saúde. Nesse contexto, a técnica de *Retrieval-Augmented Generation* (RAG) tem se destacado por combinar geração textual com recuperação de conhecimento externo, permitindo que modelos respondam com

base em fontes documentais específicas e potencialmente mais atualizadas. Em domínios sensíveis, como saúde e farmácia, essa abordagem é particularmente relevante por favorecer maior rastreabilidade das respostas, redução de alucinações e uso de bases confiáveis [Thirunavukarasu et al. 2023]. Além disso, a complexidade linguística das bulas brasileiras pode dificultar o acesso dos pacientes às informações de segurança e uso dos medicamentos [Scalercio et al. 2026].

Nesse domínio, o uso de RAG apresenta desafios de recuperação e geração. A qualidade das respostas depende não apenas do modelo gerador, mas também da estratégia de recuperação, da granularidade dos documentos, da filtragem de evidências e da capacidade de lidar com perguntas que exigem integração de múltiplas informações. Mesmo quando há recuperação externa, sistemas RAG podem produzir respostas incompletas ou incorretas quando recuperam trechos irrelevantes, omitem evidências importantes ou extrapolam o conteúdo das fontes. Esses riscos são especialmente críticos em perguntas sobre medicamentos, nas quais contraindicações, interações, posologia e alertas de segurança precisam ser preservados de forma fiel.

Além disso, grande parte da literatura sobre RAG em saúde concentra-se em *benchmarks* médicos amplos, documentos em inglês ou modelos de maior porte, enquanto cenários em português do Brasil e com restrições computacionais ainda são pouco explorados. No contexto específico de bulas brasileiras, trabalhos recentes já investigaram o uso de RAG para perguntas sobre medicamentos, mas ainda há espaço para avaliações mais detalhadas que comparem diferentes arquiteturas de recuperação, considerem níveis de dificuldade das perguntas e analisem o comportamento de modelos pequenos, tanto gerais quanto especializados em saúde [Navarro et al. 2025]. Essa lacuna é relevante porque aplicações locais, privadas e de menor custo podem ser mais viáveis em ambientes regulados ou com infraestrutura limitada.

Diante desse cenário, este trabalho propõe e avalia um assistente farmacêutico baseado em RAG para responder perguntas sobre bulas de medicamentos em português do Brasil, utilizando modelos de linguagem de pequeno porte (*Small Language Models* — SLMs). As principais contribuições são: (i) a construção de um *benchmark* com perguntas e respostas sobre bulas, estratificado por nível de dificuldade; (ii) a comparação sistemática de sete pipelines RAG, incluindo abordagens ingênuas, agenticas, híbridas, baseadas em fusão, grafo e mecanismos de segurança; e (iii) a avaliação conjunta da recuperação de evidências e da qualidade das respostas geradas por modelos gerais e especializados em saúde.

2. Trabalhos Relacionados

SLMs são atraentes em saúde porque operam melhor em ambientes com restrição de custo, energia e infraestrutura, sem exigir o mesmo nível de computação de LLMs maiores [Garg et al. 2026] . Em tarefas clínicas, métodos eficientes reforçam a viabilidade de soluções menores para uso especializado [Taylor et al. 2024].

Nos estudos comparativos, RAG frequentemente supera *fine-tuning* (FT) isolado em QA médica, mesmo quando os modelos de base não são os maiores. Em [Pingua et al. 2025], RAG e FT+RAG superaram FT sozinho na maior parte das métricas. O mesmo trabalho também mostra uma nuance importante: o híbrido FT+RAG nem sempre supera RAG puro, sugerindo *trade-offs* reais na integração entre adaptação paramétrica e

recuperação externa.

Benchmarks mais amplos sustentam essa relevância. O *benchmark* MIRAGE e o *toolkit* MEDRAG avaliou 41 combinações de corpora, *retrievers* e LLMs em 7.663 questões médicas, mostrando ganhos de até 18% de acurácia sobre c*hain-of-thought*, o que indica que o ganho vem da arquitetura RAG em si, não apenas do tamanho do modelo [Xiong et al. 2024] .

Considerando a metodologia, parte da literatura concentra-se na comparação sistemática de configurações de RAG. Alguns trabalhos propõem refinamentos arquiteturais, como recuperação iterativa com perguntas de seguimento, filtragem guiada por justificativa, estratégias *map-reduce* para contexto longo e avaliação diagnóstica separando erros de recuperação e geração, o que reforça que o desempenho de RAG depende fortemente do desenho do *pipeline* e não apenas do modelo gerador [Xiong et al. 2024] , [Sohn et al. 2025], [Zhang et al. 2025].

O QA-RAG reforça a relevância de modelos menores ou especializados quando a tarefa depende de documentos regulatórios complexos [Kim et al. 2025]. Nesse trabalho, o RAG convencional foi considerado insuficiente para compliance farmacêutico, e a variante QA-RAG introduziu uma recuperação em dois trilhos usando tanto a consulta original quanto respostas geradas por um LLM ajustado, melhorando precisão, recall e F1.

No contexto de sistemas baseados em RAG aplicados a bulas de medicamentos em português do Brasil, [Navarro et al. 2025] propuseram o RagPharma, um chatbot focado na mitigação do viés de sobreposição de dados por meio de um *framework* de avaliação *dual-dataset*. Os autores relatam que, com essa metodologia, o sistema atingiu 81% de acurácia com Mistral 7B obteve uma melhora de 60% em relação a LLMs sem RAG. Embora o estudo aborde o mesmo domínio de aplicação, ele apresenta limitações metodológicas e arquiteturais que são endereçadas em nosso trabalho. Primeiramente, os autores utilizam LLMs na faixa de 7 a 8 bilhões de parâmetros, restritos às suas versões de instrução gerais (*instruct*). Além disso, o trabalho concentra-se primariamente em atestar o viés da base de avaliação. Por fim, o banco de perguntas construído por Navarro et al. não possui categorização de complexidade e sua avaliação foi baseada em perguntas e respostas de múltipla escolha.

Mais recentemente, Garcia et al.[2026] avaliaram diferentes estratégias de RAG sobre um corpus com mais de 7.000 bulas, observando ganhos em perguntas diretamente fundamentadas nos documentos, mas também limitações do RAG ingênuo em questões de maior abstração clínica. Diferentemente desses trabalhos, avaliamos sistematicamente sete arquiteturas RAG com SLMs gerais e especializados, utilizando perguntas estratificadas por dificuldade e *ground truth* de evidências em nível de *chunk*.

3. Metodologia

A seguir, detalhamos as etapas envolvidas, incluindo a criação do *dataset* de perguntas e respostas, a categorização das perguntas por nível de dificuldade, o desenvolvimento de 7 diferentes pipelines baseados em RAG para recuperação e geração das respostas e o processo de avaliação de cada um. Os materiais complementares necessários para a re-

produtibilidade dos experimentos estão disponíveis no repositório[1], incluindo os *prompts* utilizados para geração do *benchmark*, os *prompts* empregados na construção do *ground truth* para avaliação da recuperação em nível de *chunk* e os detalhes operacionais dos pipelines RAG implementados.

3.1. Modelos e Frameworks

Os experimentos foram executados em uma estação com Intel Core Ultra 9, 32 GB de RAM e NVIDIA RTX 4070 8 GB. Os pipelines utilizaram all-MiniLM-L6-v2, ChromaDB e LangChain, com modelos locais servidos via Ollama. O Fusion RAG também empregou BM25, Reciprocal Rank Fusion e BAAI/bge-reranker-base. O GPT-5.5 foi usado na geração do *benchmark* e na avaliação automatizada.

Foram utilizados quatro SLMs, todos com aproximadamente 4 bilhões de parâmetros. Dois de propósito geral (Gemma 4 e Phi-4-mini) e duas variantes refinadas para o domínio da saúde (MedGemma e MediPhi), derivadas das respectivas versões gerais de terceira geração. A escolha por modelos de menor porte foi motivada pela possibilidade de execução em ambientes com restrição computacional, favorecendo cenários nos quais privacidade, custo e implantação local, sem abrir mão de capacidades de compreensão e raciocínio necessárias para tarefas de resposta a perguntas sobre bulas de medicamentos. O modelo MediPhi foi utilizado a partir da versão disponibilizada no HuggingFace[2], e integrado ao fluxo experimental por meio do Ollama. Para manter compatibilidade com os demais modelos, seus pesos foram convertidos para o formato GGUF, quantizados em Q4_K_M e registrados localmente via *Modelfile* customizado. Assim, o MediPhi pôde ser acessado pelo LangChain pela mesma interface dos demais modelos, permitindo sua comparação direta nas diferentes pipelines RAG sem alterações no código experimental.

3.2. Fonte de Informação

A fonte de informação consiste em bulas de medicamentos obtidas por meio da API pública da ANVISA. Foram utilizadas bulas de 3 medicamentos popularmente conhecidos em suas categorias: Tylenol® (analgésico), Amoxil® (antibiótico) e Rivotril® (ansiolítico). Optou-se por utilizar marcas registradas dos medicamentos, pois a busca pelo princípio ativo do medicamento pela API pública da Anvisa resulta em bulas de diversos fabricantes diferentes, reduzindo a reprodutibilidade destes experimentos. Também vale salientar que foram utilizadas as bulas dos pacientes e não as dos profissionais.

Para cada medicamento foram criadas 15 perguntas/respostas, sendo 5 fáceis, 5 médias e 5 difíceis, formando o *benchmark* de avaliação, com apoio do GPT 5.5. Tanto os *prompts* para geração das perguntas/respostas quanto a revisão e ajustes da versão final do *benchmark* foram propostos/revisados em colaboração com o profissional farmacêutico. De forma resumida, os níveis das perguntas são as seguintes:

- *Fácil*: aquelas cuja resposta pode ser obtida de forma simples e direta a partir de trechos explícitos em uma única seção da bula
- *Médio*: exigem inferência simples, isto é, a combinação ou interpretação de informações presentes na bula, como categorização por idade ou peso, correlação

[1] https://github.com/GESyCA/FarmacIA

[2] https://huggingface.co/microsoft/MediPhi-Instruct

entre dose e perfil do paciente, comparação entre apresentações ou síntese de relações de causa e efeito descritas no documento

- *Difícil*: envolvem cenários clínicos mais complexos, nos quais a resposta requer a integração de múltiplas informações da bula, como interações medicamentosas, comorbidades, contraindicações, ajustes de uso ou identificação de eventos adversos a partir de sintomas descritos, sem recorrer a conhecimento externo ao conteúdo das bulas

Além das perguntas e respostas esperadas, também foram incluídas no *benchmark* as seções da bula onde estas informações podem ser encontradas (seções esperadas). Todas estas informações foram usadas como *ground truth* nas avaliações das fases de recuperação e geração do RAG.

Adicionalmente, para avaliar a etapa de recuperação dos pipelines RAG, foi construído um *ground truth* em nível de *chunks* a partir do *benchmark dataset*, com apoio do GPT 5.5. Para cada pergunta, foram mapeados os identificadores dos *chunks* (*chunk IDs*) da bula que continham as evidências necessárias para a resposta, considerando a mesma granularidade operacional utilizada no ChromaDB. Esses fragmentos foram classificados por grau de relevância, distinguindo *chunks* essenciais, que continham a informação indispensável para responder à pergunta, e *chunks* de suporte, que forneciam contexto auxiliar. Com base nesse mapeamento, a recuperação foi avaliada por métricas como Hit/Recall, Evidence Set Recall e nDCG, permitindo verificar não apenas se a seção correta da bula foi recuperada, mas se os trechos específicos que fundamentavam a resposta apareceram entre os resultados retornados.

3.3. Desenho do Experimento

O fluxo de dados do experimento inicia-se com o pré-processamento das bulas, seguido por indexação, execução dos pipelines de RAG (Seção 3.4) e avaliação de cada pipeline, como ilustrado na Figura 1.

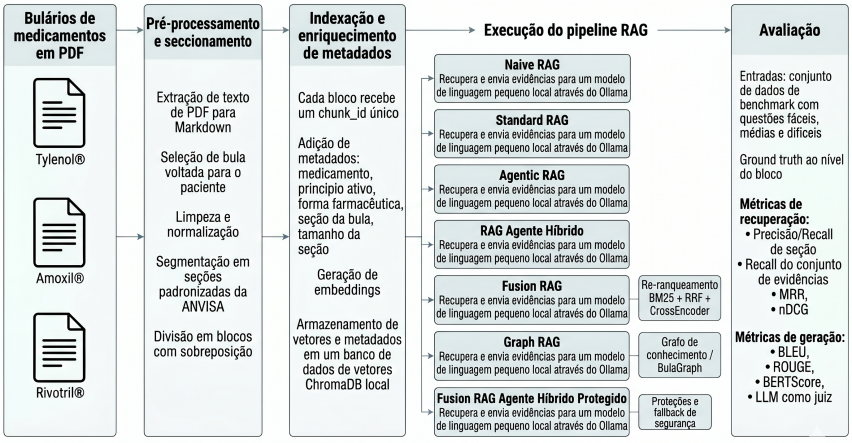


Figura 1. Fluxo Experimental Completo do Assistente farmacêutico baseado em RAG para bulas de medicamentos brasileiros

Inicialmente, o texto dos documentos é extraído e convertido para uma representação em *Markdown*, utilizando a biblioteca *pymupdf4llm*. Em seguida, quando o arquivo contém múltiplas versões de bula, como a bula do paciente e a bula do profissional de saúde, são identificados marcadores textuais para isolar apenas o conteúdo voltado

ao paciente. O texto resultante é limpo para remoção de elementos não informativos, como histórico de alterações e referências residuais, e segmentado em seções padronizadas da bula por meio de expressões regulares, considerando títulos como indicação, contraindicações, modo de uso, reações adversas e armazenamento.

Após a delimitação das seções, cada trecho textual é dividido em *chunks* utilizando um *MarkdownTextSplitter*, com tamanho de 1600 caracteres e sobreposição de 300 caracteres. Essa estratégia busca preservar a granularidade da informação sem misturar conteúdos de seções distintas. Cada *chunk* recebe um identificador único baseado em *hash MD5* e é enriquecido com metadados, incluindo nome do medicamento, princípio ativo, forma farmacêutica, seção de origem e contagem de caracteres da seção. Esses metadados são utilizados posteriormente para filtragem, roteamento e avaliação da recuperação.

Na etapa de indexação, os *chunks* são convertidos em representações vetoriais por meio do modelo de embeddings configurado, utilizando a integração *HuggingFaceEmbeddings*. Os textos, vetores, identificadores e metadados são então armazenados de forma persistente em uma coleção local do ChromaDB. Essa estrutura permite a execução de buscas por similaridade semântica combinadas com filtros de metadados, como restringir a recuperação a um medicamento específico ou a determinadas seções da bula.

A avaliação dos pipelines é conduzida a partir do *benchmark dataset* construído para o estudo, composto por perguntas e respostas organizadas por medicamento e nível de dificuldade. O motor de experimentação carrega, por meio de arquivos YAML, as configurações de cada rodada, incluindo os pipelines a serem executados, os modelos de linguagem servidos localmente via Ollama, os conjuntos de perguntas e as métricas habilitadas. Para cada pergunta, o pipeline executa seu fluxo completo de classificação, recuperação e geração, registrando informações como tempo de recuperação, latência de inferência, seções acessadas, *chunk IDs* recuperados e resposta produzida pelo modelo. Os resultados de cada execução são consolidados em arquivos CSV para análise posterior.

A avaliação utiliza o *benchmark* descrito anteriormente e configurações experimentais definidas em arquivos YAML. Para cada pergunta, são registrados os *chunks* recuperados, seções acessadas, tempos de recuperação e inferência e a resposta gerada. As métricas de recuperação e geração são detalhadas na Seção 3.5.

3.4. Arquiteturas RAG avaliadas

Foram implementados sete pipelines RAG, organizados em ordem crescente de complexidade quanto às estratégias de classificação da pergunta, recuperação de evidências e geração da resposta. O **Naive RAG** adotou uma recuperação vetorial global sobre os *chunks* da bula, filtrando apenas pelo medicamento. O **Standard RAG** adicionou uma etapa de roteamento estático, na qual um LLM classificava a pergunta em seções predefinidas da bula, restringindo a busca vetorial aos metadados correspondentes. O **Agentic RAG** substituiu esse roteamento fixo por um planejador baseado em LLM, capaz de selecionar dinamicamente múltiplas seções relevantes. O **Hybrid Agentic RAG** manteve o planejamento agêntico, mas incorporou uma estratégia sensível ao tamanho do contexto, recuperando seções completas quando compatíveis com a janela do modelo e recorrendo a *chunks* semânticos quando necessário. O **Fusion RAG** combinou recuperação densa e lexical, integrando resultados por *Reciprocal Rank Fusion* e reordenando as evidências com um *CrossEncoder*. O **Graph RAG** utilizou uma representação estruturada das bulas em

grafo de conhecimento, recuperando evidências por meio de entidades e relações, em vez de similaridade vetorial. Por fim, o **Guarded Hybrid Agentic Fusion RAG** (*Guarded RAG*, doravante) integrou planejamento hierárquico, expansão determinística de seções, busca híbrida ponderada, reranking e mecanismos de validação de confiança, acionando recuperação de fallback e alertas de segurança em perguntas de risco clínico ou baixa confiança nas evidências.

3.5. Metodologia de Avaliação

A avaliação do sistema RAG se deu em 2 aspectos: (1) recuperação das informações e (2) a geração das respostas, com métricas específicas para cada aspecto.

A avaliação da etapa de recuperação foi realizada em três dimensões complementares: roteamento por seções, recuperação em nível de *chunks* e cobertura do banco vetorial. Na avaliação por seções, as seções acessadas por cada pipeline foram comparadas às seções esperadas no *benchmark dataset*, utilizando métricas como Section Precision, Section Recall, F1-Score, Section Hit, Exact Match e All Required. Essa análise permitiu medir se o pipeline direcionou a busca para as regiões corretas da bula.

Em seguida, a avaliação operacional em nível de *chunks* comparou os *chunk IDs* recuperados com o *ground truth* definido para cada pergunta. Foram consideradas métricas em diferentes janelas Top-K, incluindo Hit@K, Evidence Set Recall@K, MRR@K e nDCG@K, a fim de verificar se os fragmentos essenciais foram recuperados e posicionados adequadamente no ranking. Por fim, a métrica Corpus Coverage foi utilizada para mensurar a proporção de *chunks* únicos recuperados em relação ao total indexado no ChromaDB, indicando o grau de exploração do corpus por cada pipeline.

A avaliação da geração combinou métricas textuais tradicionais e uma camada complementar baseada em LLM-as-a-Judge, implementada com DeepEval quando habilitada na configuração experimental. As métricas BLEU, ROUGE e BERTScore foram utilizadas para comparar a resposta gerada com a resposta de referência do *benchmark*. Como essas métricas capturam sobretudo similaridade lexical ou semântica, um LLM juiz externo avaliou também a tríade composta pela pergunta, pelo contexto recuperado da bula e pela resposta produzida pelo pipeline RAG.

A avaliação por LLM-as-a-Judge foi organizada em três camadas com métricas customizadas para o domínio farmacêutico. A camada de contexto avaliou a qualidade das evidências recuperadas por meio de *Contextual Recall, Contextual Precision e Evidence Sufficiency*. A camada de resposta avaliou fidelidade ao contexto, relevância, completude e presença de afirmações não sustentadas. A camada clínica incluiu a segurança clínica, preservação de alertas, clareza para pacientes e controle de inferências médicas indevidas.

Cada métrica julgada pelo LLM recebeu uma nota discreta de 1 a 5, posteriormente normalizada para o intervalo de 0,0 a 1,0. Notas 5 e 4 indicam desempenho excelente ou adequado, nota 3 indica resposta parcialmente adequada, e notas 2 ou 1 caracterizam falhas críticas, como distorções, omissões relevantes, alucinações ou risco clínico. O *score* global foi calculado por média ponderada, com maior peso para métricas associadas à fidelidade e segurança: camada de resposta, peso 50; camada de contexto, peso 40; e camada clínica, peso 35. Caso alguma métrica não fosse computada, os pesos eram renormalizados entre as métricas disponíveis. Além disso, foi aplicado um mecanismo de veto: respostas com nota igual ou inferior a 2 em *Faithfulness, Unsupported Claims,*

Tabela 1. Desempenho de recuperação agregado por pipeline, com média calculada entre os modelos e níveis de dificuldade das perguntas.

Pipeline	Section F1	Sec. Recall	All Required	Hit@10	MRR@10	nDCG@10
Naive RAG	N/A	N/A	0.000	0.157	0.103	0.108
Standard RAG	**0.571**	**0.723**	0.572	0.149	0.138	0.123
Agentic RAG	0.509	0.586	0.417	0.129	0.120	0.106
Hybrid Agentic RAG	0.490	0.547	0.394	0.122	0.105	0.094
Fusion RAG	0.395	0.463	0.328	0.101	0.132	0.099
Graph RAG	0.378	0.457	0.317	0.096	0.086	0.072
Guarded RAG	0.475	0.858	**0.744**	**0.201**	**0.212**	**0.178**

Clinical Safety, Warning Preservation ou *Inference Control* foram marcadas com falha crítica global, independentemente da média final.

4. Resultados e Discussões

A Tabela 1 apresenta o desempenho médio de recuperação dos sete pipelines avaliados, agregado sobre modelos e níveis de dificuldade. O *Standard RAG* obteve o maior *Section F1* médio, sugerindo melhor equilíbrio entre precisão e revocação no roteamento por seções. Por outro lado, o *Guarded RAG* apresentou o maior *Section Recall*, *All Required*, *Hit@10*, *MRR@10* e *nDCG@10*, indicando maior capacidade de recuperar evidências relevantes e posicioná-las adequadamente entre os dez primeiros resultados. Esse ganho, entretanto, veio acompanhado de maior custo computacional, com tempo médio de recuperação substancialmente superior ao dos demais pipelines. O Naive RAG apresentou desempenho competitivo em Hit@10 e nDCG@10, apesar de não realizar roteamento explícito por seções. Por esse motivo, as métricas Section F1 e Section Recall foram tratadas como não aplicáveis para esse pipeline. Essa distinção evita interpretar a ausência de roteamento por seções como uma falha direta do método. O Graph RAG, por sua vez, apresentou o menor tempo médio de recuperação, mas com desempenho inferior nas métricas ranqueadas, especialmente nDCG@10 e MRR@10.

A Tabela 2 mostra que o *Guarded RAG* foi o melhor pipeline em qualidade textual e uso do contexto. Ele teve os maiores valores de BLEU, ROUGE-L e BERTScore, sugerindo maior proximidade lexical e semântica com as respostas de referência. Mais importante, também obteve os melhores resultados em *Contextual Recall*, *Contextual Precision* e *Evidence Sufficiency*, indicando respostas mais bem fundamentadas nas evidências recuperadas. O *Naive RAG* também teve desempenho forte nas métricas de contexto, provavelmente por fornecer contexto mais amplo ao gerador, mas isso deve ser discutido junto ao seu custo de inferência muito mais alto. Já *Standard RAG* e *Agentic RAG* apresentaram bons valores de BERTScore, mas menor precisão contextual e suficiência de evidência, sugerindo que respostas semanticamente próximas nem sempre foram igualmente bem ancoradas no contexto recuperado.

Considerando os resultados da avaliação do LLM como juíz, a Figura 2 mostra que o *Final Score* é influenciado tanto pelo modelo de linguagem quanto pelo *pipeline* de RAG, com redução geral do desempenho à medida que a dificuldade das perguntas aumenta. O *Gemma 4:4b* apresenta os maiores *scores* na maioria dos *pipelines* e níveis de dificuldade, sugerindo maior robustez na etapa de geração. O *Guarded RAG* aparece

Tabela 2. Desempenho médio por pipeline em métricas lexicais e contextuais

Pipeline	BLEU	ROUGE-L	BERTScore	C. Recall	C. Precision	Evid. Suff.
Naive RAG	0.107	0.294	0.685	0.931	0.731	0.906
Standard RAG	0.099	0.300	0.732	0.731	0.526	0.650
Agentic RAG	0.089	0.283	0.726	0.736	0.537	0.657
Hybrid Agentic RAG	0.084	0.266	0.719	0.611	0.422	0.542
Fusion RAG	0.090	0.280	0.722	0.590	0.483	0.526
Graph RAG	0.045	0.168	0.608	0.719	0.618	0.656
Guarded RAG	**0.125**	**0.342**	**0.749**	**0.951**	**0.812**	**0.938**

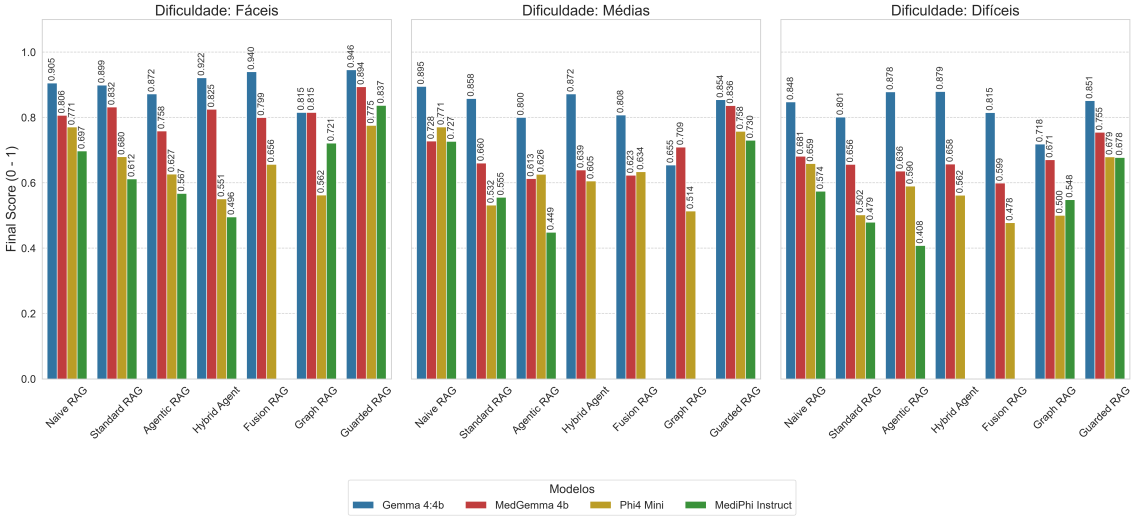


Figura 2. *Score* final de cada modelo por pipeline e dificuldade

como o *pipeline* mais consistentemente competitivo, especialmente nas perguntas fáceis e médias, nas quais mantém *scores* elevados para todos os modelos. Nas perguntas difíceis, o *Guarded RAG* continua entre as melhores configurações, mas a diferença entre modelos se torna mais evidente, indicando que, em perguntas mais complexas, a capacidade do gerador passa a ter papel ainda mais importante.

Além disso, alguns pipelines parecem ser mais sensíveis ao modelo utilizado. O *Hybrid Agent* e o *Fusion RAG*, por exemplo, apresentam *scores* elevados com *Gemma 4:4b*, mas desempenho bastante inferior ou valores nulos/ausentes para alguns modelos menores, especialmente em perguntas médias e difíceis. O *Graph RAG* apresenta comportamento mais irregular, com bons resultados em algumas combinações de modelo e dificuldade, mas desempenho inferior em outras.

A Figura 3 oferece uma visão multidimensional dos melhores pipelines para cada modelo em cada nível de dificuldade. Nas perguntas fáceis, as configurações apresentam perfis relativamente equilibrados, com altos valores de *recall contextual*, fidelidade, relevância da resposta e controle de inferência. À medida que a dificuldade aumenta, os perfis se tornam menos uniformes, mostrando que alto suporte contextual nem sempre se traduz em alta completude, segurança clínica, preservação de alertas ou controle de inferência. Esse efeito é mais evidente nas perguntas médias e difíceis, nas quais a distância

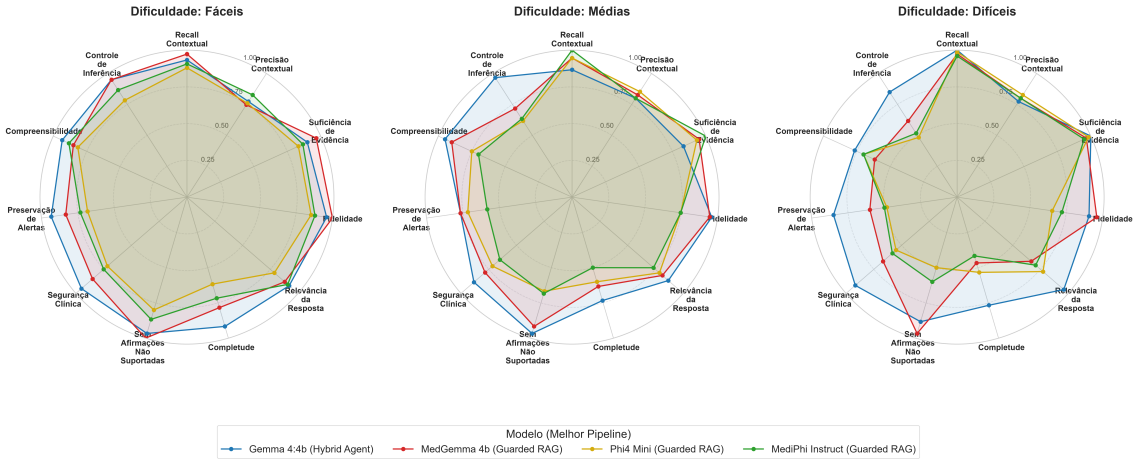


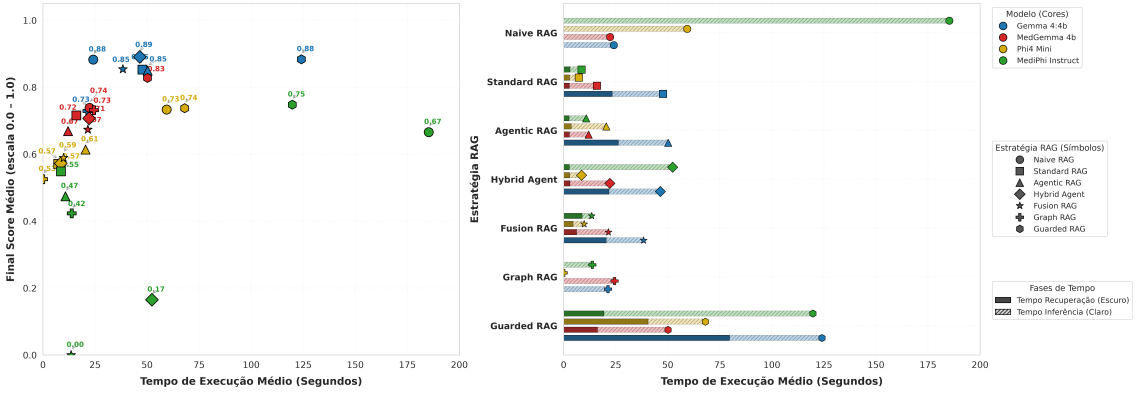
Figura 3. Avaliações do melhor pipeline de cada modelo por dificuldade

entre os modelos aumenta em várias dimensões avaliadas pelo LLM juíz.

Os gráficos da Figura 3 também mostram que o *Guarded RAG* é o melhor *pipeline* para a maioria dos modelos, nos diferentes níveis de dificuldade. Isso reforça a interpretação de que o *Guarded RAG* melhora o *grounding* das respostas ao favorecer maior *recall contextual* e suficiência de evidência. No entanto, o *Gemma 4:4b* atinge seu melhor desempenho com o *Hybrid Agent*, sugerindo que modelos geradores mais fortes podem se beneficiar de estratégias de recuperação e orquestração mais flexíveis. Nas perguntas difíceis, o *Gemma 4:4b* apresenta o perfil mais equilibrado, especialmente em relevância da resposta, completude, segurança clínica, preservação de alertas e controle de inferência. Isso sugere que a qualidade da recuperação é necessária, mas não suficiente: a qualidade clínica final também depende da capacidade do modelo de usar a evidência recuperada de forma segura e completa.

A Figura 4(a) apresenta a relação entre tempo total de execução e *Final Score*, evidenciando os diferentes compromissos entre custo e qualidade. O *Gemma 4:4b* concentra os maiores *Final Scores*, geralmente entre 0.85 e 0.89, mas com tempos intermediários ou altos dependendo do pipeline. O *Guarded RAG* aparece como uma configuração de alta qualidade, especialmente para *Gemma 4:4b*, *MedGemma 4b* e *MediPhi Instruct*, embora frequentemente associado a maior tempo total. Por outro lado, *pipelines* como *Graph RAG* e algumas configurações com *Phi4 Mini* apresentam menor tempo, mas também menor desempenho médio. O *Naive RAG* com *MediPhi Instruct* aparece como um caso de alto custo sem ganho proporcional de qualidade, reforçando que fornecer mais contexto ao gerador nem sempre resulta em melhor desempenho.

Já a Figura 4(b) detalha os tempos de recuperação e inferência por modelo e pipeline, permitindo identificar quais configurações concentram maior custo computacional. Esta figura mostra que a inferência representa a maior parcela do tempo de execução na maioria das configurações, indicando forte influência do modelo gerador sobre o custo total. A principal exceção ocorre no Guarded RAG, cujas etapas adicionais de recuperação, reranking e validação aumentam significativamente o custo de retrieval em alguns modelos. Também se destaca o elevado tempo do Naive RAG com MediPhi Instruct, predominantemente associado à inferência.



(A) Relação entre Final Score e Tempo Total de Execução

(B) Tempos de Recuperação e Inferência por Estratégia RAG

Figura 4. Relação entre custo computacional e qualidade das respostas

5. Conclusão

Este trabalho apresentou e avaliou um assistente farmacêutico baseado em RAG para responder perguntas sobre bulas de medicamentos em português do Brasil, utilizando modelos de linguagem de pequeno porte. A principal contribuição foi a comparação sistemática de sete pipelines RAG, combinada à construção de um *benchmark* estratificado por dificuldade e anotado com seções e evidências esperadas das bulas.

Os resultados mostram que a escolha da arquitetura RAG influencia diretamente a qualidade da recuperação e da geração. O Standard RAG apresentou bom equilíbrio no roteamento por seções, enquanto o Guarded RAG obteve os melhores resultados agregados em recuperação de evidências, métricas textuais e avaliação baseada em LLM, indicando respostas mais bem fundamentadas e clinicamente mais seguras. Entretanto, esse ganho esteve associado a maior custo computacional, evidenciando um compromisso entre qualidade, latência e viabilidade de implantação local. Também foi observado que, em perguntas difíceis, a qualidade do modelo gerador torna-se especialmente relevante, sugerindo que boa recuperação é necessária, mas não suficiente para garantir respostas completas e seguras.

Como limitações, o estudo considerou apenas três medicamentos, um conjunto reduzido de perguntas e respostas restritas a uma única bula por pergunta, sem análise de interações entre diferentes medicamentos. O principal limitante para o tamanho do *benchmark* são a validação humana e o tempo necessário para executar todos os experimentos com os recursos computacionais disponíveis.

Trabalhos futuros devem ampliar o *benchmark*, incluir mais classes terapêuticas, bulas para profissionais, avaliar cenários multi-medicamento (interação medicamentosa), ampliar validação humana com profissionais de saúde, avaliar o impacto de outros *embeddings* e investigar estratégias de redução de custo para pipelines RAG mais robustos, combinando diferentes SLMs. Ainda assim, os resultados indicam que RAG com SLM é uma alternativa promissora para aplicações farmacêuticas locais, desde que acompanhado de mecanismos explícitos de recuperação, verificação de evidências e controle de segurança.

6. Agradecimentos

Aos colaboradores, ao CNPq e à UFERSA pelo fomento das bolsas PIBIC e PICI via edital PROPPG/UFERSA N°21/2024, e ao Laboratório GESyCA da UFERSA pelo apoio.

Referências

Garcia, G. L., Paiola, P. H., Correia, J. V. M., Rodrigues, D., and Papa, J. P. (2026). Retrieval-augmented generation for clinical question answering in Portuguese drug leaflets: Benefits and limitations. In Souza, M., de Dios-Flores, I., Santos, D., Freitas, L., Souza, J. W. d. C., and Ribeiro, E., editors, *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026) - Vol. 2*, pages 112–120, Salvador, Brazil. Association for Computational Linguistics.

Garg, M., Raza, S., Rayana, S., Liu, X., and Sohn, S. (2026). The rise of small language models in healthcare: A comprehensive survey. *Computer Science Review*, 62:100999.

Kim, J., Hur, M., and Min, M. (2025). From rag to qa-rag: Integrating generative ai for pharmaceutical regulatory compliance process. *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*.

Navarro, L. C., Mutz, F., Paixão, T. M., Zanetti, G. G., Badue, C., Souza, A. F. D., and Oliveira-Santos, T. (2025). Ragpharma: A rag-based chatbot for medicine leaflets with a dual-dataset evaluation framework. *J. Braz. Comput. Soc.*, 31:1137–1149.

Pingua, B., Sahoo, A., Kandpal, M., Murmu, D., Rautaray, J., Barik, R., and Saikia, M. (2025). Medical llms: Fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12.

Scalercio, A., Bertotto, E., Jesus, S., Finatto, M. J., and Paes, A. (2026). Annotation guidelines and challenges for automatic simplification of Portuguese drug leaflets. In Souza, M., de Dios-Flores, I., Santos, D., Freitas, L., Souza, J. W. d. C., and Ribeiro, E., editors, *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026) - Vol. 2*, pages 121–127, Salvador, Brazil. Association for Computational Linguistics.

Sohn, J., Park, Y., Yoon, C., Park, S., Hwang, H., Sung, M., Kim, H., and Kang, J. (2025). Rationale-guided retrieval augmented generation for medical question answering. In Chiruzzo, L., Ritter, A., and Wang, L., editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12739–12753, Albuquerque, New Mexico. Association for Computational Linguistics.

Taylor, N., Ghose, U., Rohanian, O., Nouriborji, M., Kormilitzin, A., Clifton, D. A., and Nevado-Holgado, A. (2024). Efficiency at scale: Investigating the performance of diminutive language models in clinical tasks. *Artificial intelligence in medicine*, 157:103002.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, 29(8).

Xiong, G., Jin, Q., Lu, Z., and Zhang, A. (2024). Benchmarking retrieval-augmented generation for medicine. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, Bangkok, Thailand. Association for Computational Linguistics.

Zhang, G., Xu, Z., Jin, Q., Chen, F., Fang, Y., Liu, Y., Rousseau, J. F., Xu, Z., Lu, Z., Weng, C., and Peng, Y. (2025). Leveraging long context in retrieval augmented language models for medical question answering. *NPJ Digital Medicine*, 8.