UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

IAGO TONELLO

Aplicação de Resposta a Perguntas baseada em LLMs e RAG para um Centro de Informação sobre Medicamentos

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof. Dra. Renata Galante
Co-orientador: Prof. Dra. Érika Fernandes Cota

Porto Alegre
2026

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Profª. Marcia Cristina Bernardes Barbosa
Vice-Reitor: Prof. Pedro de Almeida Costa
Pró-Reitora de Graduação: Profª. Nádya Pesce da Silveira
Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary
Coordenador do Curso de Engenharia de Computação: Prof. Paulo Francisco Butzen
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

AGRADECIMENTOS

Agradeço à minha família por todo o apoio ao longo desta jornada, especialmente desde a decisão de deixar uma pequena cidade e me mudar para a capital em busca dos meus sonhos. Em especial, à minha mãe, Rosemeri, e aos meus avós, Antônio e Iracema, por terem me criado com tanto amor, dedicação e valores. Obrigado por sempre acreditarem em mim, incentivarem meus estudos e me ensinarem que o esforço e a perseverança são o caminho para alcançar nossos objetivos. Esta conquista também é de vocês.

À minha namorada, Amanda, por caminhar ao meu lado em praticamente todos os dias da minha vida, desde o ensino fundamental até hoje. Nunca imaginei que teria o privilégio de compartilhar a vida com alguém tão especial. Vivemos inúmeros momentos, superamos desafios, comemoramos conquistas e construímos uma história que continua apenas começando. Obrigado por todo o carinho, apoio, paciência e incentivo durante essa caminhada. Agradeço também por me proporcionar a alegria de viver com a Elora, nossa cachorrinha, que trouxe ainda mais felicidade e leveza aos nossos dias. Eu não imaginava que fosse possível criar um vínculo tão forte com um animal.

Aos meus amigos de infância e aos amigos que a graduação me proporcionou, meu sincero agradecimento. Cada conversa, trabalho em grupo, momento de descontração e apoio nos períodos mais difíceis tornaram essa jornada muito mais leve e significativa. Levarei essas amizades para a vida.

Às professoras Renata Galante e Érika Cota, por me apresentarem um tema tão atual, relevante e desafiador, além de conduzirem a orientação deste trabalho com dedicação, paciência e incentivo. Agradeço pelos ensinamentos, pela confiança depositada em mim e por todas as contribuições que foram fundamentais para a realização deste trabalho.

Por fim, agradeço a todos que, de alguma forma, fizeram parte desta trajetória e contribuíram para que este sonho se tornasse realidade. Cada palavra de incentivo, cada gesto de apoio e cada aprendizado ao longo dessa caminhada tiveram um papel importante nesta conquista.

RESUMO

O Centro de Informação sobre Medicamentos (CIM) presta serviço de informação reativa a profissionais de saúde, demandando respostas precisas e rastreáveis sobre administração, preparo, diluição, estabilidade e compatibilidade de fármacos. Estudos anteriores demonstraram que Grandes Modelos de Linguagem (LLMs), isoladamente, são insuficientes para esse domínio clínico devido à estaticidade do conhecimento e ao risco de alucinação, e que implementações básicas de Geração Aumentada por Recuperação (RAG) ainda geram baixa similaridade semântica e elevada taxa de respostas evasivas. Dando continuidade a essa prova de conceito, este trabalho tem como entregável central um Produto Mínimo Viável (MVP): uma interface de uso, em modo pergunta–resposta única, que materializa uma configuração recomendada de *backend* RAG e estabelece o canal para a futura validação humana de utilidade e segurança. Para definir essa configuração com base em evidência, e não em ajuste arbitrário, o *framework* RAGAS é empregado como instrumento de investigação: uma série experimental, organizada em uma fase preliminar (Experimentos 1 e 2) e uma incremental cumulativa (Experimentos 3 a 8), na qual cada etapa altera um único componente do *pipeline* (ingestão, recuperação e geração), permite localizar em qual camada os ajustes trazem maior retorno e orientar as decisões de engenharia. A investigação indicou que os ajustes mais produtivos incidem sobre a fragmentação estrutural dos documentos e sobre o *grounding* leve no *prompt*, ao passo que o *Chain-of-Thought* degrada a ancoragem factual, um achado negativo que motivou sua exclusão da configuração final. Restrito ao modo pergunta–resposta única efetivamente coberto pela série experimental, o MVP é o artefato, pronto para implantação, por meio do qual a configuração recomendada alcançará os profissionais do CIM em trabalho futuro.

Palavras-chave: Processamento de Linguagem Natural. Modelos de Linguagem. RAG. Engenharia de Prompt. Aplicação de Resposta a Perguntas. Saúde.

ABSTRACT

Drug Information Centers (CIM) provide reactive information services to health professionals, requiring precise and traceable answers regarding the administration, preparation, dilution, stability, and compatibility of drugs. Previous work has shown that Large Language Models (LLMs), on their own, are insufficient for this clinical domain due to knowledge staleness and hallucination risk, and that basic Retrieval-Augmented Generation (RAG) implementations still yield low semantic similarity and a high rate of evasive answers. Building upon such a proof of concept, the central deliverable of this work is a Minimum Viable Product (MVP): a single-shot question-answering interface that exposes a recommended RAG *backend* configuration to the CIM and establishes the channel for the future human validation of utility and safety. To define this configuration on the basis of evidence rather than arbitrary tuning, the RAGAS framework is employed as an investigative instrument: an experimental series, organized into a preliminary phase (Experiments 1–2) and a cumulative incremental phase (Experiments 3–8), in which each step changes a single pipeline component (ingestion, retrieval, or generation), makes it possible to locate in which layer the adjustments yield the greatest return and to guide the engineering decisions. The investigation indicated that the most productive adjustments concern structural document chunking and light grounding in the prompt, whereas Chain-of-Thought reasoning degrades factual grounding, a negative finding that motivated its exclusion from the final configuration. Restricted to the single-shot mode actually covered by the experimental series, the MVP is the vehicle through which the recommended configuration reaches the CIM professionals.

Keywords: Natural Language Processing. Language Models. RAG. Prompt Engineering. Question Answering Application. Healthcare.

SUMÁRIO

LISTA DE FIGURAS **8**
LISTA DE TABELAS **9**
LISTA DE ABREVIATURAS E SIGLAS **10**
1 INTRODUÇÃO **11**
2 FUNDAMENTAÇÃO TEÓRICA **15**
2.1 Processamento de Linguagem Natural **15**
2.2 Grandes Modelos de Linguagem **16**
2.3 Geração Aumentada por Recuperação **17**
2.3.1 Arquitetura de Recuperação: Busca Híbrida 18
2.3.2 Fusão de Classificação Recíproca 19
2.4 Engenharia de Prompt **20**
2.5 Avaliação de Aplicações Baseadas em RAG **22**
2.6 Sistemas de Resposta a Perguntas **24**
3 TRABALHOS RELACIONADOS **26**
4 PROPOSTA **30**
4.1 Visão Geral **30**
4.2 Ingestão: Fragmentação Estrutural **31**
4.3 Recuperação: Busca Híbrida e Reranking **33**
4.4 Geração: Engenharia de Prompt **34**
5 EXPERIMENTOS **36**
5.1 Avaliação: Framework RAGAS **36**
5.2 Metodologia Experimental **37**
5.3 Configuração dos Experimentos **38**
5.4 Experimentos **40**
5.4.1 Experimento 1: Avaliação de Linha de Base 40
5.4.2 Experimento 2: Reestruturação da Ingestão e Ampliação de Contexto 42
5.4.3 Experimento 3: Baseline com Chunking Estrutural 43
5.4.4 Experimento 4: Busca Híbrida 45
5.4.5 Experimento 5: Reranking Semântico (Cross-Encoder) 46
5.4.6 Experimento 6: Grounding Leve no Prompt 47
5.4.7 Experimento 7: Few-Shot no Prompt de Geração 48
5.4.8 Experimento 8: Chain-of-Thought (CoT) no Prompt de Geração 50
5.5 Síntese da Evolução **51**
5.6 Análise das Abstenções ao Longo da Série Experimental **52**
5.7 Análise Geral dos Resultados **54**
5.7.1 Ameaças à Validade 55
6 DEMONSTRAÇÃO **56**
6.1 Produto Mínimo Viável para o CIM **56**
6.2 Cenários de Uso **57**
6.2.1 Cenário 1: Consulta Respondível 58
6.2.2 Cenário 2: Comportamento de Recusa 58
6.2.3 Estado Atual e Trabalhos Futuros 60
7 CONCLUSÃO **61**
DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA **64**

REFERÊNCIAS **65**
APÊNDICE A — PROMPT UTIIZADO NO SISTEMA RAG NO EXPERIMENTO 6 ..... **67**
APÊNDICE B — PROMPT UTILIZADO NO SISTEMA RAG NO EXPERIMENTO 7 ..... **68**
APÊNDICE C — PROMPT UTILIZADO NO SISTEMA RAG NO EXPERIMENTO 8 ..... **70**
ANEXO A — PROMPT ORIGINAL PROPOSTO POR Cota (2025) **72**

LISTA DE FIGURAS

Figura 2.1 Processos da arquitetura RAG 21
Figura 2.2 Exemplo de Chain-of-Thought prompting 23
Figura 2.3 Arquitetura geral de um sistema de QA 25

Figura 4.1 Arquitetura geral do pipeline RAG proposto 31
Figura 4.2 Comparação conceitual entre fragmentação por tamanho fixo e estrutural (premissa de projeto, corroborada empiricamente no Capítulo 5) .. 32
Figura 4.3 Fluxo detalhado do módulo de recuperação 34

Figura 5.1 Fase preliminar do delineamento experimental. O Experimento 2 modifica simultaneamente múltiplos componentes do pipeline, sendo utilizado apenas para orientar a definição da configuração base da fase incremental. 39
Figura 5.2 Delineamento experimental: fase incremental cumulativa (Experimentos 3 a 8), na qual cada configuração herda a do anterior e adiciona um único componente. 39
Figura 5.3 Evolução das métricas RAGAS nos Experimentos 1 a 8. A faixa cinza corresponde a fase preliminar (Exp. 1–2); a direita da linha pontilhada, a evolução incremental deste trabalho (Exp. 3–8). O salto entre os Exp. 2 e 3 evidencia o ganho do chunking estrutural; os Exp. 6 e 7 concentram os melhores resultados em Fidelidade e Correcao; o CoT (Exp. 8) reverte a ancoragem. 52

Figura 6.1 Tela inicial da interface (*Streamlit*): formulário de consulta em modo *single-shot* 57
Figura 6.2 Consulta respondível: resposta fundamentada no contexto recuperado, com citação do campo de origem 58
Figura 6.3 Comportamento de recusa diante de informação ausente: o sistema declara a indisponibilidade em vez de inferir a resposta 59

LISTA DE TABELAS

Tabela 3.1 Comparação das estratégias adotadas pelos trabalhos relacionados. 29

Tabela 5.1 Configuração fixa compartilhada pelos experimentos..... 39
Tabela 5.2 Modificação em cada experimento..... 40
Tabela 5.3 Resultados da avaliação RAGAS do pipeline RAG..... 41
Tabela 5.4 Resultados da avaliação RAGAS do pipeline RAG (Experimento 2) 43
Tabela 5.5 Resultados da avaliação RAGAS, Experimento 3..... 44
Tabela 5.6 Comparação entre Experimentos 2 e 3..... 44
Tabela 5.7 Resultados da avaliação RAGAS, Experimento 4..... 45
Tabela 5.8 Comparação entre Experimentos 3 e 4..... 45
Tabela 5.9 Resultados da avaliação RAGAS, Experimento 5..... 46
Tabela 5.10 Comparação entre Experimentos 4 e 5..... 46
Tabela 5.11 Resultados da avaliação RAGAS, Experimento 6..... 47
Tabela 5.12 Comparação entre os Experimentos 5 e 6..... 48
Tabela 5.13 Resultados da avaliação RAGAS, Experimento 7..... 49
Tabela 5.14 Comparação entre os Experimentos 6 e 7..... 49
Tabela 5.15 Resultados da avaliação RAGAS, Experimento 8..... 50
Tabela 5.16 Comparação entre os Experimentos 7 (Few-Shot) e 8 (Chain-of-Thought)..... 50
Tabela 5.17 Síntese da evolução das médias das métricas (Experimentos 1 a 8: fase preliminar 1–2 e incremental 3–8)..... 52
Tabela 5.18 Abstenções (“Não sei”) e cobertura por experimento, sobre as 70 perguntas de avaliação..... 53
Tabela 5.19 Ganho cumulativo das métricas até a melhor configuração (Experimento 7)..... 54

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CIM	*Centro de Informação sobre Medicamentos*
CoT	*Chain-of-Thought*
LLM	*Large Language Model*
MVP	*Minimum Viable Product*
NLP	*Natural Language Processing*
PoC	*Proof of Concept*
QA	*Question Answering*
QAS	*Question Answering System*
RAG	*Retrieval-Augmented Generation*
RAGAS	*Retrieval Augmented Generation Assessment*
REN	*Reconhecimento de Entidades Nomeadas*
RI	*Recuperação de Informação*
RNNLM	*Recurrent Neural Network Language Model*
RRF	*Reciprocal Rank Fusion*
TF-IDF	*Term Frequency–Inverse Document Frequency*

1 INTRODUÇÃO

A segurança do paciente em ambiente hospitalar depende, em larga medida, da qualidade da informação sobre medicamentos disponível aos profissionais de saúde. Decisões cotidianas envolvendo administração, preparo, diluição, estabilidade e compatibilidade de fármacos exigem respostas precisas e atualizadas, cujo erro pode comprometer diretamente o tratamento. Os Centros de Informação sobre Medicamentos (CIM) foram instituídos precisamente para suprir essa demanda: atuam como serviços de informação reativa que respondem, de forma fundamentada, às consultas de enfermeiros, técnicos de enfermagem, médicos e farmacêuticos, com base em fontes documentais confiáveis e rastreáveis (COTA, 2025).

Apesar de sua relevância, a operação desse serviço enfrenta desafios práticos. O volume e a complexidade das consultas, a necessidade de respostas rápidas e a constante atualização do conhecimento farmacêutico impõem uma carga significativa aos profissionais responsáveis. Essa conjunção de fatores motiva a busca por ferramentas computacionais capazes de apoiar a atividade do CIM, aumentando a velocidade e a consistência das respostas sem abrir mão do rigor e da rastreabilidade das fontes.

Os Grandes Modelos de Linguagem (*Large Language Models — LLMs*) alcançaram desempenho notável em tarefas de Processamento de Linguagem Natural (*Natural Language Processing — NLP*), incluindo Sistemas de Resposta a Perguntas (*Question Answering System – QAS*) (ZHAO et al., 2025). No entanto, quando aplicados isoladamente a domínios sensíveis, esses modelos apresentam duas limitações críticas: o conhecimento paramétrico estático, suscetível a desatualização, e o fenômeno da alucinação, pelo qual geram informações factualmente incorretas, ainda que fluentes e plausíveis (ZHANG et al., 2025). Em um contexto clínico, uma resposta incorreta pode ter consequências graves, o que torna inaceitável o uso direto de LLMs sem mecanismos que garantam a verificabilidade das informações.

A arquitetura de Geração Aumentada por Recuperação (*Retrieval-Augmented Generation — RAG*) surgiu como uma abordagem para mitigar essas limitações (LEWIS et al., 2021; GAO et al., 2024). Ao desacoplar o conhecimento do modelo e ancorar as respostas em trechos recuperados de uma base externa, o RAG aumenta a atualização e a rastreabilidade das informações. A eficácia dessa ancoragem, contudo, depende fortemente da qualidade da engenharia do *pipeline*, em especial, da forma como os documentos

são segmentados e indexados, da estratégia de recuperação empregada e da formulação do *prompt* de geração.

Nesse cenário, Cota (2025) desenvolveu uma Prova de Conceito (*Proof of Concept — PoC*) de um QAS baseado em uma implementação básica de RAG (*naive RAG*) para o CIM. O sistema indexava tabelas de medicamentos institucionais e concatenava os trechos mais relevantes ao *prompt* de diferentes LLMs, avaliando as respostas pela similaridade semântica em relação a um conjunto padrão-ouro derivado de consultas reais. Os resultados, todavia, evidenciaram limitações importantes: pontuações médias baixas de similaridade (entre 40,95 e 53,75), independentemente do modelo adotado, e uma taxa elevada de respostas evasivas do tipo "Não sei", indicando que o contexto recuperado era insuficiente para embasar respostas confiáveis. A autora concluiu que o sistema ainda não estava maduro o suficiente e apontou a necessidade de otimização da engenharia do RAG, envolvendo escolha de modelos de *embeddings*, dimensionamento dos fragmentos textuais, estratégias de *reranking* e expansão do contexto antes de se investir no custo e no esforço metodológico de uma validação humana pelos especialistas do CIM.

A lacuna identificada por Cota (2025) define, de forma natural, o ponto de partida do presente trabalho. Mais do que um ajuste pontual, o amadurecimento da engenharia do RAG é importante para que o sistema possa, no futuro, ser submetido a uma validação humana de utilidade e segurança com custo e esforço justificáveis.

Essa motivação é reforçada pela natureza do domínio. Em aplicações clínicas, o que é importante não é a fluência ou a relevância aparente da resposta, mas a sua ancoragem factual no contexto recuperado e a sua correção em relação à referência. Tais propriedades precisam ser examinadas de forma explícita e separada, não para maximizar escores, mas para orientar onde investir esforço de engenharia e caracterizar, com honestidade, o que o sistema é e o que ele ainda não é. Por essa razão, o presente trabalho tem como entregável central um Produto Mínimo Viável (*Minimum Viable Product — MVP*): uma interface de uso, em modo pergunta–resposta única, que materializa uma configuração recomendada de *backend* RAG e estabelece o canal para a futura validação humana de utilidade e segurança. Para definir essa configuração com base em evidência, e não em ajuste arbitrário, o *backend* é examinado por uma série experimental sistemática na qual o *framework* RAGAS (*Retrieval Augmented Generation Assessment*) (ES et al., 2025) atua como instrumento de investigação: suas métricas, interpretadas como sinais relativos, indicam em qual camada do *pipeline*(ingestão, recuperação ou geração) os ajustes trazem maior retorno. As métricas automáticas não substituem, portanto, a validação final, que

permanece reservada aos profissionais do CIM.

O objetivo principal deste trabalho é desenvolver de forma reproduzível um MVP destinado ao CIM, por meio de uma interface de uso em modo pergunta–resposta única, que materialize uma configuração recomendada de *backend* RAG como artefato para futura implantação e validação de utilidade e segurança pelos profissionais, dando continuidade à prova de conceito de Cota (2025). Para definir essa configuração com apoio em evidência, o *framework* RAGAS é empregado como instrumento de investigação, capaz de localizar, ao longo do *pipeline*, onde os ajustes se fazem necessários.

Os objetivos específicos são:

- Reestruturar a estratégia de ingestão documental, substituindo a fragmentação por tamanho fixo por uma fragmentação estrutural que preserve as relações semânticas entre os campos das tabelas farmacêuticas;
- Aprimorar a etapa de recuperação por meio de busca híbrida, com fusão de *rankings*, e de *reranking* semântico;
- Investigar estratégias de engenharia de *prompt* na etapa de geração, incluindo *grounding* leve, *few-shot prompting* e *Chain-of-Thought*;
- Empregar o *framework* RAGAS como instrumento de investigação, por meio de uma série experimental cumulativa que altera um componente do *pipeline* por vez, para localizar em qual camada (ingestão, recuperação ou geração) os ajustes trazem maior retorno;
- Conceber uma interface de uso, em modo pergunta–resposta única, que materialize a configuração recomendada como entregável central do MVP destinado ao CIM e constitua o canal para a futura validação humana de utilidade e segurança.

O restante deste documento está organizado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica que embasa o trabalho, abordando Processamento de Linguagem Natural, Grandes Modelos de Linguagem, a arquitetura de Geração Aumentada por Recuperação, engenharia de *prompt*, o *framework* de avaliação RAGAS e sistemas de resposta a perguntas. O Capítulo 3 discute trabalhos relacionados, com destaque para o estudo que originou este trabalho. O Capítulo 4 descreve a proposta, detalhando as três frentes de otimização e a concepção do MVP. O Capítulo 5 relata a série experimental cumulativa, apresenta a síntese dos resultados e discute as decisões adotadas na configuração final. O Capítulo 6 demonstra a interface do MVP em execução. Por fim, o Capítulo 7 sintetiza as contribuições, recapitula as limitações e indica caminhos para

trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conceitos teóricos fundamentais que embasam o desenvolvimento deste trabalho. A seguir, a seção está organizada de forma a detalhar os tópicos de Processamento de Linguagem Natural e Grandes Modelos de Linguagem, seguidos pela arquitetura de Geração Aumentada por Recuperação e Engenharia de Prompt. Por fim, são abordadas a Avaliação de Aplicações Baseadas em RAG e os Sistemas de Resposta a Perguntas.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural busca aprimorar a interação entre humanos e computadores, capacitando as máquinas a interpretar o significado da linguagem natural por meio da análise semântica e das relações de significado entre palavras, além de gerar respostas coerentes e contextualmente adequadas (KOUNTE et al., 2020).

Uma das primeiras aplicações de NLP foi o programa ELIZA (WEIZENBAUM, 1966), que simulava diálogos simples com regras pré-definidas. Com o avanço da computação, surgiram métodos estatísticos como Term Frequency–Inverse Document Frequency (TF-IDF), capazes de identificar termos relevantes em grandes volumes de texto. Nos anos 2000, o aprendizado de máquina tornou-se o estado da arte em NLP, com algoritmos como Naive Bayes e Decision Trees, melhorando tarefas de classificação de documentos, análise de sentimentos e recuperação de informação (JIANG et al., 2025).

A ascensão do aprendizado profundo, aliada à disponibilidade de grandes volumes de dados textuais, impulsionou o desenvolvimento dos *Recurrent Neural Network Language Models* (RNNLMs). Esses modelos foram projetados para capturar dependências sequenciais em dados textuais, demonstrando maior eficácia na modelagem de contexto em comparação com abordagens anteriores (RAIAAN et al., 2024).

No âmbito industrial e científico, o NLP apresenta usos práticos:

- **Ciência dos Materiais:** Ferramentas de NLP, como o ChemDataExtractor, varrem artigos científicos buscando e extraindo de forma automática informações complexas sobre rotas de síntese de materiais e suas propriedades. LLMs especializados na área, como o SCIBERT e o MatSciBERT, auxiliam pesquisadores até mesmo a

prever o comportamento de novos materiais e compostos através do cálculo de semelhanças. Além disso, agentes de Inteligência Artificial com NLP (como o Coscientist) conseguem ler documentações e executar experimentos de reações químicas no mundo físico de forma autônoma através de laboratórios robóticos (JIANG et al., 2025).

- **Saúde e Biomedicina:** Ferramentas de NLP são capazes de analisar volumes massivos de literatura médica e prontuários para auxiliar em diagnósticos, ou até mesmo atuar como recepcionistas virtuais, auxiliando na redução de custos e de contaminações nos hospitais (RAIAAN et al., 2024).

2.2 Grandes Modelos de Linguagem

Modelos de linguagem sequenciais possuíam uma limitação estrutural. Eles falhavam em capturar dependências de longo prazo. Essas dependências conectam palavras distantes e definem o significado da frase. A natureza sequencial também impedia a paralelização do treinamento.

A arquitetura *Transformer*, introduzida no trabalho *Attention Is All You Need*, representou um marco nesse contexto ao propor um mecanismo de autoatenção (*self-attention*) capaz de modelar de forma mais eficaz essas dependências, além de permitir treinamento altamente paralelizável, resultando em ganhos significativos de eficiência e escalabilidade (VASWANI et al., 2017).

Segundo Zhao et al. (2025), os LLMs são, em sua maioria, baseados na arquitetura *Transformer* e caracterizam-se pela utilização de centenas de bilhões, ou até mais, de parâmetros, treinados sobre grandes volumes de dados textuais. Nesse contexto, seu objetivo técnico fundamental consiste em modelar a distribuição de probabilidade de sequências de texto, permitindo prever o próximo *token* mais provável com base no contexto previamente fornecido.

Ao serem treinados em larga escala, com grandes volumes de dados e parâmetros, esses modelos passam a capturar padrões semânticos complexos, possibilitando sua aplicação em diferentes tarefas. Dessa forma, deixam de atuar apenas como geradores de texto e passam a desempenhar o papel de solucionadores de tarefas de propósito geral (ZHAO et al., 2025).

Os LLMs são utilizados em uma ampla variedade de cenários e domínios do mundo real, incluindo:

- **Tarefas Clássicas de NLP:** Aplicações como geração de texto, sistemas de perguntas e respostas, análise de sentimentos, reconhecimento de entidades nomeadas e extração de informações a partir de dados não estruturados (ZHAO et al., 2025).
- **Programação e Engenharia de Software:** Devido ao treinamento em grandes volumes de código, esses modelos são utilizados para síntese de programas, sugestão e completamento de código, sumarização e detecção ou correção de falhas (bugs) (ZHAO et al., 2025).
- **Aplicações em Domínios Específicos (Saúde):** Na área da saúde, os LLMs são aplicados na extração de informações clínicas, suporte à tomada de decisão médica, análise de saúde mental e simplificação de relatórios médicos, como laudos radiológicos (ZHAO et al., 2025).

Os LLMs demonstram desempenho avançado em uma ampla gama de aplicações e representam o estado da arte em NLP. Contudo, ainda apresentam desafios relevantes, em especial o fenômeno da alucinação. Esse termo descreve situações em que o modelo gera informações incorretas, imprecisas ou não fundamentadas nos dados de entrada (ZHANG et al., 2025). Esse comportamento pode comprometer a confiabilidade dos modelos, especialmente em contextos sensíveis, como na área da saúde, evidenciando a necessidade de técnicas que aumentem a precisão e a verificabilidade das respostas geradas.

2.3 Geração Aumentada por Recuperação

A principal motivação para o desenvolvimento da Geração Aumentada por Recuperação decorre das limitações intrínsecas dos modelos de linguagem em acessar, atualizar e manipular conhecimentos de forma precisa, especialmente em tarefas intensivas em conhecimento (LEWIS et al., 2021). Embora esses modelos armazenem uma quantidade significativa de informações factuais em seus parâmetros internos, eles apresentam dificuldades ao lidar com consultas que exigem conhecimento específico de domínio ou informações que não estão presentes, ou estão desatualizadas, em seus dados de treinamento (LEWIS et al., 2021; GAO et al., 2024).

Essa dependência exclusiva de uma memória paramétrica pode levar à geração de respostas factualmente incorretas, fenômeno conhecido como alucinação (GAO et al., 2024). O ajuste fino (*fine-tuning*), embora seja uma alternativa para incorporar novos padrões e especializar o modelo, demanda recursos computacionais significativos,

apresenta limitações na internalização de novos fatos e não elimina a estaticidade do conhecimento entre os ciclos de treinamento (GAO et al., 2024).

Em contraste, o RAG contorna essas limitações ao desacoplar o conhecimento do modelo, permitindo o acesso a informações atualizadas por meio de uma base de dados externa. Dessa forma, a atualização do sistema pode ser realizada de maneira incremental, sem a necessidade de retreinamento do modelo, tornando a abordagem mais adequada para ambientes dinâmicos e cenários que exigem informações recentes e verificáveis (LEWIS et al., 2021; GAO et al., 2024).

Na sua forma mais fundamental, conhecida como arquitetura "*Naive RAG*", o processo é iniciado por uma etapa de indexação, na qual os dados brutos são extraídos, segmentados em fragmentos textuais menores (*chunks*) e convertidos em representações vetoriais por meio de um modelo de codificação. Esses vetores são armazenados em um banco de dados especializado, estruturando o repositório externo para permitir buscas de similaridade semântica (GAO et al., 2024).

Após a organização do conhecimento externo, o sistema entra na fase de recuperação (*retriever*), iniciada a partir da consulta do usuário (GAO et al., 2024). A consulta é codificada no mesmo espaço vetorial utilizado na fase de indexação, por meio de *embeddings*, o que permite calcular a similaridade com as representações armazenadas. Com base nessas pontuações, o recuperador seleciona os *chunks* mais relevantes, que são então encaminhados ao modelo gerador (LEWIS et al., 2021; GAO et al., 2024).

Em arquiteturas mais avançadas, essa etapa pode incluir mecanismos adicionais antes da busca, como reescrita, expansão ou roteamento da consulta. Essas técnicas visam reduzir ambiguidades e melhorar a precisão da recuperação, especialmente em cenários com linguagem imprecisa ou incompleta (GAO et al., 2024).

2.3.1 Arquitetura de Recuperação: Busca Híbrida

A evolução das arquiteturas de recuperação de informação estabeleceu a busca híbrida como um padrão para sistemas que demandam alta precisão textual e contextual. Essa abordagem combina múltiplos métodos de recuperação, principalmente a busca lexical (esparsa) e a busca semântica ou vetorial (densa). Sob a perspectiva de arquitetura de software, a dependência de um único método de recuperação introduz um ponto de falha estrutural no sistema (SIVAKUMAR; SUGUMARAN; QIANG, 2026).

A busca lexical baseia-se na correspondência exata de termos e índices textuais,

apresentando alta eficácia para identificar terminologias rígidas, identificadores numéricos, chaves primárias e palavras-chave específicas. Entretanto, esse modelo é incapaz de capturar sinônimos, variações gramaticais ou a intenção conceitual por trás de uma consulta.

Por outro lado, a busca semântica mapeia consultas e documentos em um espaço vetorial contínuo por meio de codificadores de texto, permitindo a recuperação baseada na proximidade do significado latente. Embora trate ambiguidades com eficiência, esse método pode apresentar dificuldades ao lidar com jargões altamente especializados ou códigos exatos.

A combinação dessas abordagens produz uma infraestrutura de recuperação mais robusta, capaz de tolerar a variabilidade da linguagem natural sem comprometer a precisão terminológica necessária em domínios técnicos (SIVAKUMAR; SUGUMARAN; QIANG, 2026).

Como a recuperação híbrida produz múltiplas listas de resultados provenientes de mecanismos distintos, torna-se necessário empregar uma estratégia de agregação capaz de combinar essas classificações de forma consistente. Nesse contexto, técnicas de fusão de rankings são utilizadas para consolidar os resultados recuperados e produzir uma ordenação final mais eficaz (CORMACK; CLARKE; BUETTCHER, 2009).

2.3.2 Fusão de Classificação Recíproca

A Fusão de Classificação Recíproca (*Reciprocal Rank Fusion — RRF*) é uma técnica não supervisionada de agregação de rankings que combina listas ordenadas produzidas por diferentes sistemas de recuperação, considerando exclusivamente a posição ocupada por cada documento em cada ranking individual (CORMACK; CLARKE; BUETTCHER, 2009).

O escore atribuído a um documento é definido pela Equação 2.1.

$$RRF(d) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{2.1}$$

onde R representa o conjunto de rankings produzidos pelos sistemas de recuperação, r(d) corresponde à posição ocupada pelo documento d no ranking r, e k é uma constante utilizada para suavizar a influência das posições mais elevadas e reduzir a sensibilidade a variações individuais entre os rankings (CORMACK; CLARKE; BUETTCHER, 2009).

Uma das principais vantagens do RRF consiste em sua independência dos escores produzidos pelos mecanismos de busca subjacentes. Diferentemente de métodos baseados na combinação direta de pontuações, que frequentemente exigem processos adicionais de normalização, o RRF utiliza apenas a ordem relativa dos documentos, tornando a fusão mais simples e robusta diante de diferentes escalas de relevância (CORMACK; CLARKE; BUETTCHER, 2009).

Além disso, o método preserva evidências provenientes de múltiplos sistemas de recuperação. Documentos que aparecem em posições elevadas em diferentes rankings tendem a receber maiores pontuações agregadas, enquanto resultados altamente posicionados em apenas um mecanismo ainda mantêm capacidade de influenciar significativamente a classificação final (CORMACK; CLARKE; BUETTCHER, 2009).

Essa característica é particularmente relevante em arquiteturas de busca híbrida, nas quais diferentes estratégias de recuperação capturam aspectos complementares da relevância documental (SIVAKUMAR; SUGUMARAN; QIANG, 2026).

Dessa forma, o RRF apresenta baixo custo computacional, simplicidade de implementação e desempenho consistente na combinação de resultados heterogêneos (CORMACK; CLARKE; BUETTCHER, 2009), características que justificam sua ampla adoção em sistemas modernos de recuperação de informação e arquiteturas RAG (SIVAKUMAR; SUGUMARAN; QIANG, 2026).

Com a etapa de recuperação concluída, o pipeline avança para a fase de geração, na qual a consulta original e os documentos selecionados são combinados para compor um contexto expandido (GAO et al., 2024). Esse contexto é então fornecido ao componente gerador que condiciona suas predições às informações recuperadas para produzir a resposta final (LEWIS et al., 2021). A Figura 2.1 ilustra os processos da arquitetura RAG.

2.4 Engenharia de Prompt

A engenharia de prompt projeta e refina instruções para otimizar a interação com os modelos. O objetivo é guiar o comportamento do LLM. Isso estrutura as respostas e garante a execução de tarefas com precisão (AMATRIAIN, 2024).

Sob uma perspectiva arquitetural, a engenharia de prompt pode ser compreendida como a definição de uma interface textual que conecta os dados de entrada e os requisitos da aplicação à capacidade de processamento dos modelos de linguagem. Nesse contexto, o prompt atua de forma análoga a uma Interface de Programação de Aplicações (*Application*

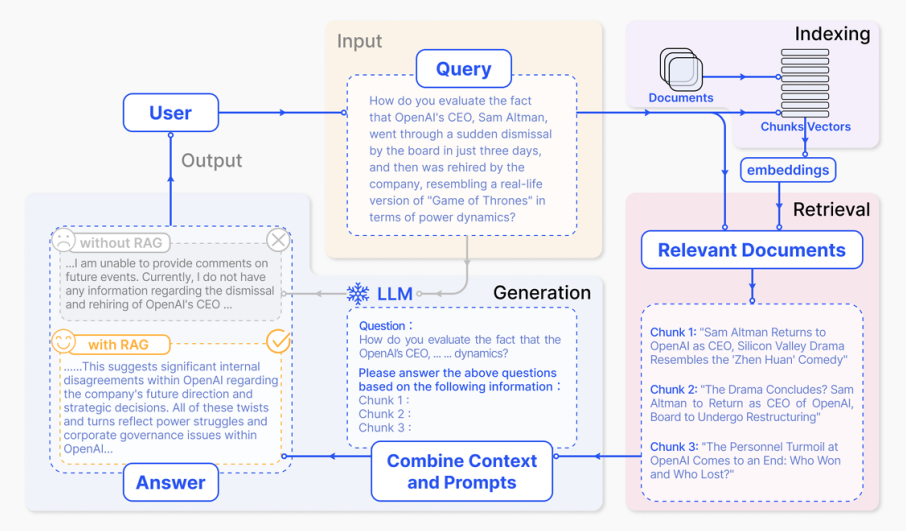


Figura 2.1 – Processos da arquitetura RAG

Fonte: (GAO et al., 2024)

Programming Interface — API), controlando tanto o comportamento quanto o formato das respostas geradas.

Sob uma perspectiva técnica, essa prática é motivada pela necessidade de mitigar limitações estruturais e operacionais inerentes aos LLMs (AMATRIAIN, 2024). Tais limitações exigem o uso de técnicas especializadas para viabilizar sua aplicação em cenários reais, destacando-se:

- **Alucinações:** Modelos de linguagem podem gerar informações semanticamente coerentes, porém factualmente incorretas. Para mitigar esse problema, utilizam-se técnicas de engenharia de prompt como a indução de raciocínio passo a passo (*Chain-of-Thought*) e instruções de ancoragem (*grounding*), que restringem o modelo a operar exclusivamente sobre o contexto fornecido, aumentando a confiabilidade das respostas (AMATRIAIN, 2024).
- **Alta Sensibilidade à Entrada:** LLMs apresentam alta sensibilidade a variações no prompt, incluindo mudanças sutis na estrutura textual, ordem dos exemplos ou escolha de palavras. Essas variações podem impactar significativamente o desempenho do modelo, exigindo processos iterativos de refinamento e validação dos prompts para garantir estabilidade e previsibilidade nas respostas (SCHULHOFF et al., 2025).

De forma análoga às práticas tradicionais da engenharia de software, a engenharia de prompt configura-se como um processo metodológico e iterativo, no qual a interface textual é continuamente modificada, testada e validada até que produza respostas consis-

tentes e alinhadas aos requisitos da aplicação (SCHULHOFF et al., 2025). Nesse contexto, diversas técnicas são empregadas para orientar e estabilizar o comportamento dos modelos de linguagem.

Dentre essas técnicas, destaca-se o *Few-Shot Prompting*, que consiste em fornecer ao modelo alguns exemplos da tarefa desejada diretamente no corpo do prompt. Esses exemplos funcionam como demonstrações do comportamento esperado, permitindo que o modelo aprenda padrões de execução e formatação sem a necessidade de re-treinamento ou atualização de seus parâmetros internos.

Um exemplo prático dessa técnica pode ser observado em prompts que apresentam preferências explícitas do usuário para induzir recomendações. Conforme ilustrado por (AMATRIAIN, 2024):

> "Here are some examples of TV shows I really like: Breaking Bad, Peaky Blinders, The Bear. I did not like Ted Lasso. What other shows do you think I might like?"

Nesse caso, o modelo utiliza os exemplos fornecidos como base para inferir padrões implícitos de preferência, sem necessidade de treinamento adicional.

Outra técnica relevante é o *Chain-of-Thought* (CoT), ou cadeia de pensamento, que consiste em uma sequência coerente de etapas intermediárias de raciocínio que conduzem à obtenção de uma resposta final. Essa abordagem se baseia na decomposição de problemas complexos em subetapas menores e mais simples, permitindo que cada parte seja resolvida de forma incremental antes da conclusão do resultado. Essa estratégia induz o modelo a tornar explícito seu processo de inferência, resultando em ganhos significativos de precisão em tarefas que envolvem múltiplas etapas de dedução, como problemas matemáticos e análise de regras complexas (WEI et al., 2023).

Um exemplo da técnica de *Chain-of-Thought* é apresentado na Figura 2.2, onde o modelo é induzido a explicitar seu raciocínio passo a passo antes de fornecer a resposta final.

2.5 Avaliação de Aplicações Baseadas em RAG

O avanço da arquitetura de RAG no campo do Processamento de Linguagem Natural impulsionou a necessidade de compreender e otimizar o desempenho desses sistemas em diferentes cenários de aplicação. Além disso, a garantia de confiabilidade e robustez tornou-se um aspecto central na avaliação de pipelines RAG. A recuperação de informações irrelevantes, ambíguas ou contraditórias pode comprometer significativamente a

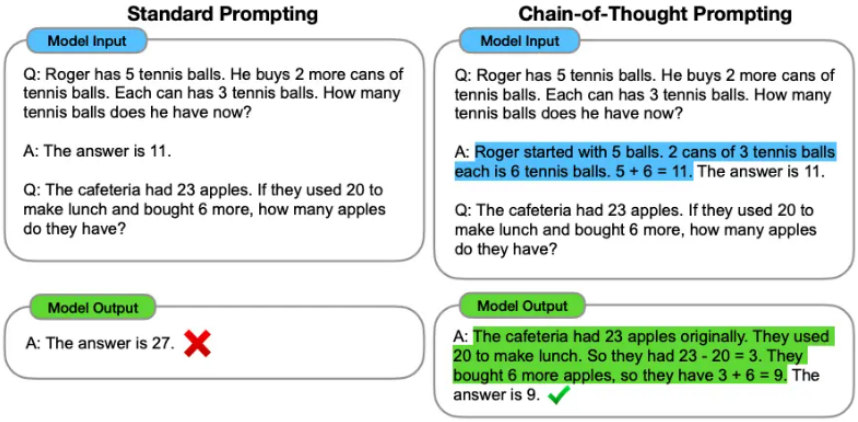


Figura 2.2 – Exemplo de Chain-of-Thought prompting

Fonte: (WEI et al., 2023)

qualidade da resposta gerada, resultando em saídas incorretas ou semanticamente inconsistentes. Em determinados contextos, especialmente em aplicações sensíveis, pode ser mais apropriado que o modelo se abstenha de responder do que produza uma resposta potencialmente equivocada (GAO et al., 2024).

Aplicações baseadas na arquitetura RAG exigem uma quantidade significativa de ajustes para operar de forma otimizada. O desempenho de um pipeline RAG não depende de um único fator isolado, sendo influenciado pelo modelo de recuperação utilizado, pela qualidade dos documentos presentes na base de conhecimento, pelo LLM empregado na geração e pela estratégia de formulação de prompts (ES et al., 2025).

Nesse contexto, Es et al. (2025) propuseram o RAGAS , um *framework* para a avaliação automatizada de pipelines RAG no qual um LLM atua como juiz para estimar diferentes dimensões de qualidade. Em sua concepção original, essas métricas são livres de referência, isto é, não demandam respostas-referência (ground truth) anotadas manualmente (ES et al., 2025).

A versão empregada neste trabalho (0.4.3) estende o projeto inicial com métricas baseadas em referência, que exploram uma resposta-referência quando disponível, operando sobre a pergunta, o contexto recuperado, a resposta gerada e, quando exigido, a resposta-referência.

A métrica de Fidelidade (*Faithfulness*) avalia se as afirmações da resposta podem ser inferidas do contexto recuperado: a resposta é decomposta em afirmações, cada uma verificada contra o contexto, e a pontuação é a razão $F = |V|/|S|$ entre afirmações suportadas e o total identificado. A Relevância da Resposta (*Answer Relevancy*) mede o alinhamento da resposta com a pergunta, penalizando respostas incompletas ou redundan-

tes, sendo calculada como a média da similaridade do cosseno entre a pergunta original e *n* perguntas geradas a partir da resposta.

A Precisão do Contexto (*Context Precision*) avalia se os fragmentos recuperados são relevantes para a consulta e se os relevantes ocupam as posições mais altas do ranking, por meio de uma precisão média ponderada pela posição. O Recall do Contexto (*Context Recall*) mede a cobertura da resposta-referência pelo contexto recuperado, como a proporção das afirmações da referência que encontram suporte no contexto. Por fim, a Correção da Resposta (*Answer Correctness*) compara a resposta gerada com a referência, combinando um escore factual do tipo F_1 (sobre acertos, omissões e acréscimos) com a similaridade semântica entre as duas respostas (Ragas Contributors, 2026).

Além de fornecer uma medida quantitativa da qualidade das respostas, o RAGAS auxilia na identificação da camada do pipeline RAG responsável por eventuais degradações de desempenho. Como as métricas avaliam aspectos distintos da recuperação e da geração, torna-se possível verificar se um problema decorre da recuperação de contextos irrelevantes ou incompletos, do uso inadequado do contexto pelo modelo de linguagem ou da baixa qualidade da resposta gerada. Essa separação facilita o processo de depuração e orienta a otimização dos diferentes componentes do sistema, como a estratégia de *chunking*, o mecanismo de recuperação, os modelos de *embedding*, o *retriever*, o *prompt* ou o próprio LLM empregado na geração (ES et al., 2025; Ragas Contributors, 2026).

2.6 Sistemas de Resposta a Perguntas

A utilização de sistemas de Resposta a Perguntas (Question Answering — QA) é motivada pela necessidade de fornecer respostas mais precisas e diretas aos usuários em linguagem natural. Diferentemente dos sistemas tradicionais de Recuperação de Informação (RI), que retornam listas de documentos relevantes para que o usuário encontre manualmente a informação desejada, os sistemas de QA buscam compreender a pergunta, recuperar informações relevantes e entregar apenas a resposta solicitada. Dessa forma, esses sistemas reduzem o esforço de busca e tornam a interação mais eficiente e útil para o usuário (LOPES; VIEIRA; OLIVEIRA, 2023).

Os sistemas de Resposta a Perguntas podem apresentar diferentes níveis de complexidade, pois envolvem diversas etapas de processamento responsáveis por compreender a pergunta e gerar uma resposta adequada. Entre essas etapas, destacam-se tarefas como classificação de texto, extração de tópicos e Reconhecimento de Entidades Nomeadas

(REN), cada uma possuindo desafios e problemas de pesquisa próprios. Além das abordagens tradicionais baseadas em múltiplas etapas explícitas, também existem modelos *end-to-end*, nos quais o sistema funciona como uma "caixa preta", recebendo uma pergunta como entrada e produzindo diretamente uma resposta. Esses modelos utilizam redes neurais profundas treinadas especificamente para a tarefa de QA, concentrando implicitamente diferentes processos de compreensão e geração de linguagem natural em suas camadas internas. Atualmente, esses sistemas têm alcançado quantidades extremamente elevadas de parâmetros neurais, variando de milhões até bilhões e trilhões, o que amplia sua capacidade de entendimento da linguagem natural, armazenamento de informação e geração de respostas (LOPES; VIEIRA; OLIVEIRA, 2023). A Figura 2.3 apresenta a arquitetura geral de um sistema de QA.

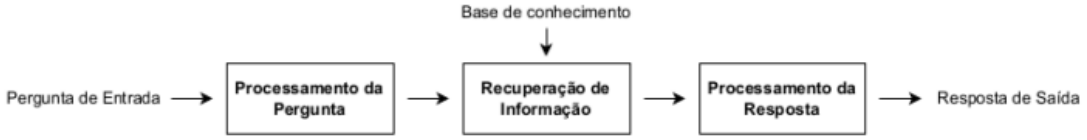


Figura 2.3 – Arquitetura geral de um sistema de QA

Fonte: (LOPES; VIEIRA; OLIVEIRA, 2023)

3 TRABALHOS RELACIONADOS

Em um estudo recente, (REIS; REIS; ROCHA, 2025) investigaram a utilização de LLMs combinados com a arquitetura RAG para aprimorar a geração de instruções em sistemas de prescrição médica eletrônica. O estudo destaca que, embora os LLMs possuam capacidade para produzir orientações personalizadas aos pacientes, esses modelos ainda apresentam limitações importantes em cenários clínicos mais complexos, podendo gerar respostas incompletas, inconsistentes ou até mesmo alucinações.

Para mitigar esses problemas, os autores propuseram uma abordagem denominada "RAG sob demanda". Diferentemente do modelo tradicional de RAG, que frequentemente recupera documentos extensos e nem sempre relevantes, a solução identifica o medicamento prescrito e extrai diretamente da base de bulas da Agência Nacional de Vigilância Sanitária (Anvisa) apenas a seção relacionada ao modo de administração do fármaco, como "Como devo usar este medicamento?". Em seguida, esse conteúdo é resumido e incorporado ao prompt de um modelo de código aberto, o Llama 3.1 de 8 bilhões de parâmetros, fornecendo um contexto mais específico e direcionado para a geração da resposta.

A abordagem foi avaliada por especialistas da área médica por meio de diversos casos clínicos ambulatoriais, comparando os resultados do modelo com RAG às versões que utilizavam exclusivamente engenharia de prompts. Os experimentos demonstraram que o uso do RAG melhorou significativamente a clareza e a adequação das prescrições geradas. Além disso, a utilização de informações extraídas diretamente das bulas reduziu de forma expressiva a ocorrência de erros clínicos críticos, como instruções vagas, contraditórias ou incorretas, além de limitar as alucinações do modelo.

Os autores concluem que o uso de LLMs ancorados em fontes documentais confiáveis representa um avanço importante para aumentar a segurança da aplicação de inteligência artificial na área da saúde. Ainda assim, o estudo ressalta que a supervisão e validação humana permanecem indispensáveis para a adoção prática dessas soluções em ambientes clínicos.

Hidayati e Atikah (2025) desenvolveram um chatbot médico baseado na arquitetura RAG para responder dúvidas relacionadas à adenomiose e à endometriose em idioma indonésio. O estudo destaca que, embora os LLMs apresentem potencial para auxiliar na área da saúde, esses sistemas ainda sofrem com problemas de alucinação, podendo gerar

informações incorretas ou sem embasamento confiável.

Para reduzir esse problema, os autores utilizaram uma arquitetura RAG. O sistema utilizava documentos médicos extraídos de artigos e arquivos PDF como fonte de conhecimento. Antes da busca pelos documentos, a pergunta do usuário era reformulada para melhorar a recuperação das informações mais relevantes. Os textos foram divididos em pequenos blocos e armazenados em um banco vetorial utilizando embeddings multilíngues.

A avaliação do sistema foi realizada com o framework RAGAS, utilizado para medir automaticamente a qualidade das respostas geradas pelo RAG. O framework analisava a relação entre a pergunta do usuário, os documentos recuperados e a resposta produzida pelo modelo. Entre as métricas avaliadas estavam a fidelidade da resposta aos documentos utilizados, a capacidade do sistema de recuperar informações relevantes e a precisão do contexto retornado.

Os resultados demonstraram bons níveis de fidelidade e recuperação de contexto, indicando que as respostas geradas permaneceram alinhadas às fontes médicas utilizadas e apresentaram baixos índices de alucinação. Entretanto, as métricas de relevância e precisão apresentaram desempenho inferior, pois o sistema frequentemente recuperava documentos genéricos sobre saúde feminina para perguntas específicas sobre endometriose e adenomiose.

Com base nesses resultados, os autores concluíram que a principal limitação do sistema estava na etapa de recuperação de documentos, e não na geração das respostas pelo LLM. Como trabalhos futuros, o estudo sugere aprimorar a recuperação das informações por meio da redução do número de documentos recuperados, melhorias na segmentação dos textos e utilização de embeddings especializados da área biomédica.

Em uma abordagem voltada à segurança medicamentosa, Bang et al. (2025) propuseram um sistema de perguntas e respostas baseado em RAG para identificar contraindicações farmacêuticas em populações sensíveis, como gestantes, crianças e pacientes polimedicados. Observando as limitações e a imprecisão de LLMs genéricos em tarefas clínicas críticas, os autores integraram o modelo GPT-4o-mini a uma base de dados regulatória de utilização de medicamentos. Para isso, a arquitetura adotou uma estratégia avançada de recuperação híbrida, combinando buscas semânticas e lexicais, complementada por uma etapa de re-ranqueamento. Os documentos foram segmentados em blocos semanticamente coesos, sem sobreposição, e indexados em um banco vetorial. A avaliação empírica, conduzida em um conjunto de 300 instâncias, demonstrou que o RAG

elevou a precisão do sistema substancialmente, saltando de patamares próximos a 50% nos modelos base para até 94% em casos relacionados à faixa etária. O estudo conclui que ancorar respostas em fontes médicas validadas mitiga as alucinações, embora destaque que a recuperação exata para interações medicamentosas complexas ainda represente um desafio técnico considerável.

Como fundamento principal para o desenvolvimento deste trabalho, destaca-se o estudo realizado por Cota (2025), que propôs e implementou a automação do serviço de informação reativa em um CIM. A autora desenvolveu uma PoC de um QAS estruturado sob uma implementação básica de RAG (naive RAG), que realizava a indexação vetorial de tabelas de medicamentos institucionais e concatenava os trechos mais relevantes diretamente ao prompt de seis modelos diferentes de LLM. A avaliação do sistema baseou-se na medição da similaridade semântica (avaliada de forma automatizada por outro LLM) entre as respostas geradas e um conjunto de dados padrão-ouro derivado de consultas históricas reais do CIM.

No entanto, o estudo de Cota (2025) demonstrou que o uso de uma estratégia básica de RAG foi insuficiente para atingir a precisão exigida pelo contexto clínico. Os resultados registraram pontuações médias baixas de similaridade (variando de 40,95 a 53,75) independentemente do LLM adotado, além de uma taxa significativamente alta de respostas evasivas do tipo "Não sei", indicando que o contexto recuperado e fornecido ao agente era insuficiente para embasar as respostas de forma confiável. A pesquisa concluiu que o sistema ainda não estava maduro o suficiente, apontando a necessidade premente de investigações focadas na otimização da engenharia do RAG, como a escolha do modelo de embeddings, dimensionamento dos recortes de texto (chunks), estratégias de reranking e aumento do contexto passado ao prompt, antes de se assumir os custos e o esforço metodológico de uma validação humana pelos especialistas do CIM.

De forma geral, os estudos analisados demonstram que a utilização de RAG contribui para reduzir alucinações e aumentar a confiabilidade de sistemas baseados em LLMs na área da saúde. Entretanto, os trabalhos também evidenciam que a qualidade das respostas depende fortemente da etapa de recuperação da informação. Enquanto Reis, Reis e Rocha (2025) e Bang et al. (2025) obtiveram melhorias significativas ao utilizar estratégias de recuperação direcionadas e bases documentais especializadas, Hidayati e Atikah (2025) identificaram limitações relacionadas à recuperação de documentos relevantes. Esses resultados sugerem que a otimização da arquitetura de recuperação constitui um fator crítico para o sucesso de aplicações RAG em contextos clínicos.

A Tabela 3.1 sintetiza as principais características técnicas dos trabalhos analisados. Observa-se que as pesquisas exploram diferentes estratégias para aprimorar a recuperação da informação, como busca híbrida, segmentação otimizada e reranqueamento. Nesse contexto, o presente trabalho reúne e adapta essas abordagens para o cenário de um CIM, propondo uma evolução da arquitetura apresentada por Cota (2025). O foco está no aprimoramento sistemático da etapa de recuperação da informação, buscando reduzir alucinações, aumentar a precisão das respostas e fornecer um contexto mais relevante ao modelo de linguagem.

Tabela 3.1 – Comparação das estratégias adotadas pelos trabalhos relacionados.

Trabalho	**Busca híbrida**	**Reranking**	**Chunking otimizado**	**Avaliação**
Reis et al. (2025)	Não	Não	Não	Não
Hidayati et al.	Não	Não	Parcial	RAGAS
Bang et al. (2025)	Sim	Sim	Sim	Sim
Cota (2025)	Não	Não	Não (Naive RAG)	Similaridade semântica
Presente trabalho	**Sim**	**Sim**	**Sim**	**RAGAS**

4 PROPOSTA

Este capítulo apresenta a proposta desenvolvida neste trabalho para o aprimoramento de um sistema de RAG aplicado ao domínio farmacêutico. A Seção 4.1 descreve a visão geral da solução, apresentando seus objetivos e a arquitetura geral do *pipeline* RAG, com destaque para as principais modificações em relação à prova de conceito anterior. A Seção 4.2 detalha a estratégia proposta para a etapa de ingestão, baseada na fragmentação estrutural dos documentos farmacêuticos. A Seção 4.3 aborda o aprimoramento da etapa de recuperação por meio de busca híbrida, fusão de *rankings* e *reranking* semântico. Por fim, a Seção 4.4 descreve as estratégias de engenharia de *prompt* investigadas na etapa de geração.

4.1 Visão Geral

O presente trabalho tem como objetivo otimizar o *backend* RAG e disponibilizá-lo como um MVP, por meio de uma interface, como artefato para futura implantação e validação pelos profissionais do CIM. Parte da arquitetura desenvolvida na prova de conceito inicial de Cota (2025), cujas limitações (baixa similaridade semântica, alta taxa de respostas evasivas e contexto recuperado insuficiente) motivaram uma rodada de otimização do *backend*. Os experimentos descritos neste trabalho (Capítulo 5) funcionam como um instrumento de avaliação comparativa que orienta a otimização do RAG e seleciona a configuração de melhor desempenho para a interface; a interface, por sua vez, é o veículo destinado a expor essa configuração otimizada aos profissionais, em condições reais de uso, e a permitir coletar a validação de utilidade e segurança que as métricas automáticas não cobrem.

A proposta concentra-se em três frentes:

- Ingestão: reestruturação da estratégia de segmentação documental;
- Recuperação: aprimoramento da recuperação por meio de busca híbrida e reranking;
- Geração: refinamento do processo de geração por meio de engenharia de prompt;

A Figura 4.1 apresenta uma visão geral da arquitetura proposta, destacando as principais etapas do pipeline RAG, desde a ingestão dos documentos até a geração e avaliação das respostas. As seções seguintes detalham cada um dos módulos que compõem essa

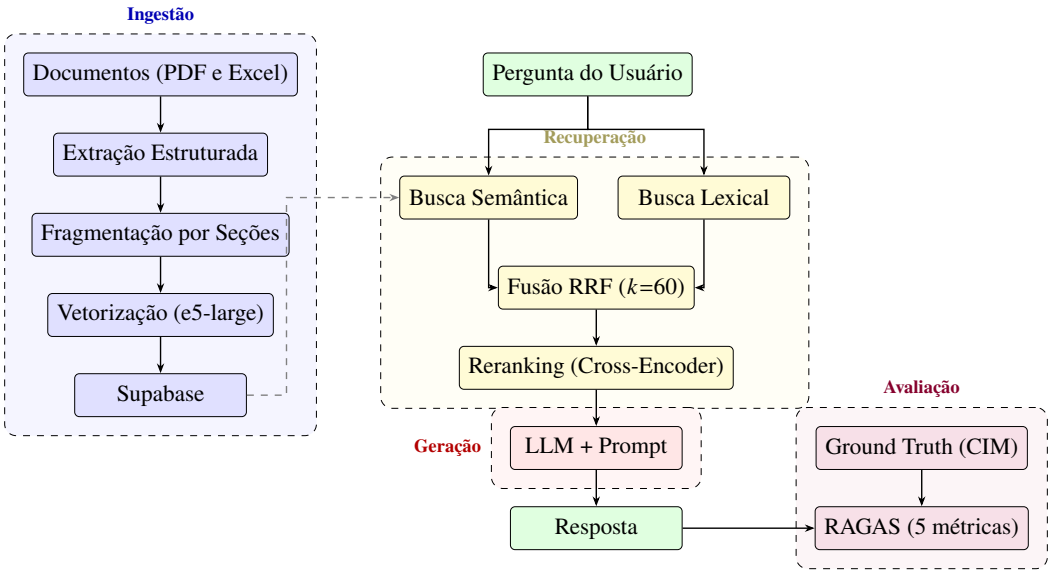


Figura 4.1 – Arquitetura geral do pipeline RAG proposto
Fonte: Elaborado pelo autor (2026).

arquitetura, descrevendo as estratégias adotadas para a ingestão, recuperação e geração. Por sua vez, o MVP desenvolvido a partir dessa arquitetura é apresentado no Capítulo 6, enquanto a avaliação experimental das diferentes configurações propostas é descrita no Capítulo 5.

4.2 Ingestão: Fragmentação Estrutural

No módulo de ingestão, o trabalho propõe substituir a fragmentação por tamanho fixo, utilizada no estudo anterior (COTA, 2025), por uma estratégia de Fragmentação Estrutural (*Document-Aware Chunking*) como demonstrada na Figura 4.2. Essa decisão é motivada pelas características dos documentos farmacêuticos utilizados como base de conhecimento, compostos majoritariamente por tabelas complexas em formato PDF com campos estruturados.

Em abordagens tradicionais de segmentação por tamanho fixo, informações semanticamente relacionadas podem ser separadas em diferentes *chunks*, resultando em perda de contexto e redução da qualidade da recuperação vetorial. No domínio farmacêutico, essa limitação torna-se crítica devido à forte dependência semântica entre atributos como medicamento, concentração, diluição, estabilidade, compatibilidade e vias de administração. Além disso, observações dos experimentos preliminares (Capítulo 5) sugerem que fragmentos extensos por tamanho fixo podem agregar múltiplos tópicos distintos em um único vetor, no qual informações potencialmente relevantes deixariam de atingir escores

suficientes na busca.

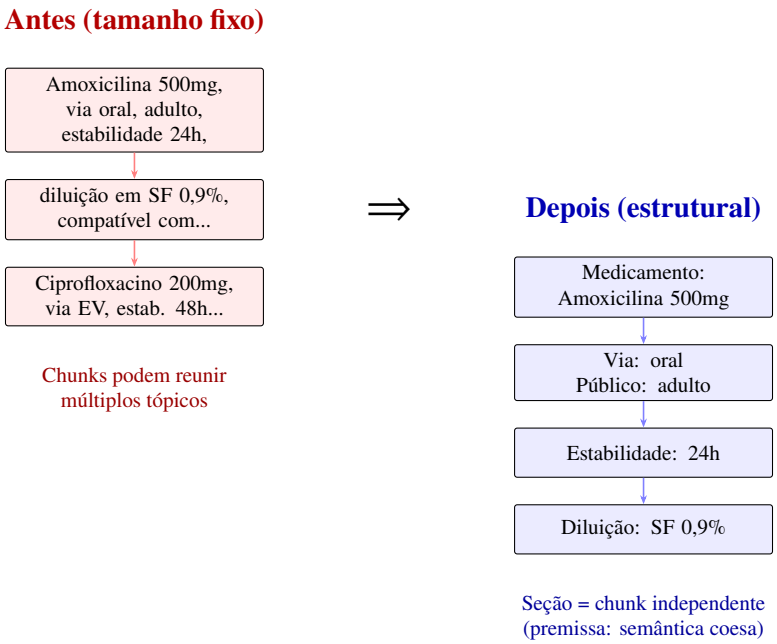


Figura 4.2 – Comparação conceitual entre fragmentação por tamanho fixo e estrutural (premissa de projeto, corroborada empiricamente no Capítulo 5)

Fonte: Elaborado pelo autor (2026).

A Figura 4.2 ilustra conceitualmente a diferença entre a estratégia de fragmentação utilizada na prova de conceito anterior e a estratégia proposta neste trabalho. À esquerda, a fragmentação por tamanho fixo gera *chunks* que podem reunir informações de diferentes tópicos ou dividir uma mesma informação entre múltiplos fragmentos, dificultando a recuperação de todo o contexto necessário para responder à consulta. À direita, a fragmentação estrutural organiza o conteúdo em unidades correspondentes às seções lógicas do documento, de forma que cada *chunk* represente um único atributo ou informação semanticamente coesa, como medicamento, via de administração, estabilidade ou diluição. Essa organização busca preservar as relações semânticas presentes no documento original e produzir representações vetoriais mais específicas, favorecendo a recuperação de informações relevantes durante a busca.

Dessa forma, a arquitetura proposta na Figura 4.1 utilizará uma estratégia estrutural em que cada seção lógica da tabela (apresentação, reconstituição, vias de administração, estabilidade, etc.) será tratada como um *chunk* independente. Os dados extraídos serão reorganizados em formato estruturado do tipo "Rótulo: valor", buscando preservar explicitamente as relações semânticas entre os campos do documento e favorecendo vetores mais coesos semanticamente.

O pipeline de processamento documental está dividido nas seguintes etapas:

1. extração estruturada dos dados dos documentos fonte;
2. fragmentação por seções lógicas, com campos rotulados por seção da tabela;
3. vetorização e indexação no banco de dados.

A estratégia de indexação utiliza o banco de dados Supabase (com a extensão `pgvector`) em conjunto com o modelo de *embeddings multilingual-e5-large* (1024 dimensões), permitindo representar semanticamente os documentos e consultas em um espaço vetorial compartilhado. A escolha justifica-se por duas razões: a variante *multilingual* cobre o português, idioma do corpus e das consultas, por ser treinada em cerca de cem línguas e é de pesos abertos, permitindo auto-hospedagem, sem custo por *token* e com os dados clínicos mantidos na própria infraestrutura; e a variante *large*, com 1024 dimensões, oferece representações mais expressivas que as variantes *base* e *small*, propriedade útil para distinguir atributos farmacêuticos próximos, como concentração, diluição e estabilidade.

4.3 Recuperação: Busca Híbrida e Reranking

Além da reestruturação da ingestão, a proposta inclui dois aprimoramentos na camada de recuperação como apresentado na Figura 4.3.

O primeiro é a busca híbrida, que combina busca semântica (vetorial) com busca lexical (*full-text search*). Os rankings produzidos pelos dois mecanismos serão consolidados por meio da técnica de fusão RRF, que utiliza exclusivamente a posição ocupada por cada documento em cada ranking individual para produzir uma ordenação final. O valor adotado para a constante de suavização k é 60, o mesmo proposto e validado no trabalho original do RRF (CORMACK; CLARKE; BUETTCHER, 2009) e adotado como padrão pelas principais implementações de busca híbrida, incluindo o Supabase utilizado neste trabalho. Com esse valor, as diferenças entre as primeiras posições de cada ranking individual são atenuadas, de modo que a ordenação final privilegia os documentos bem colocados em múltiplos rankings em detrimento daqueles altamente ranqueados em apenas um deles. A busca semântica captura relações de significado entre a consulta e os documentos, enquanto a busca lexical recupera termos técnicos exatos, como nomes de medicamentos e concentrações, que a busca vetorial pode negligenciar.

O segundo é o reranking semântico, feito com o modelo*cross-encoder/mmarco-mMiniLMv2-L12-H384-v1*, multilingual e otimizado para busca em português. Após a

busca híbrida retornar um conjunto de 10 candidatos, o *cross-encoder* pontuará cada par (pergunta, *chunk*) de forma conjunta, capturando interações semânticas mais ricas do que os bi-encoders utilizados na etapa de indexação. Os *chunks* serão reordenados por relevância e os 5 mais pertinentes serão selecionados para composição do contexto.

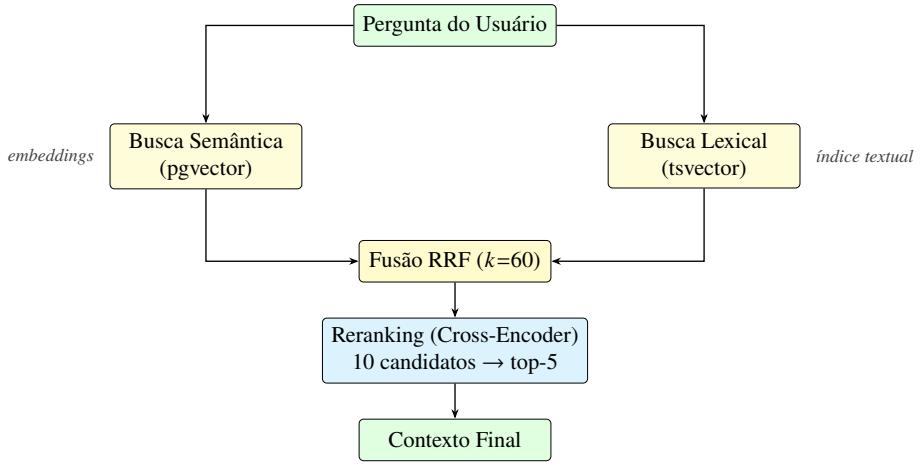


Figura 4.3 – Fluxo detalhado do módulo de recuperação
Fonte: Elaborado pelo autor (2026).

4.4 Geração: Engenharia de Prompt

No módulo de geração, o trabalho investigará estratégias de engenharia de prompt com o objetivo de reduzir alucinações e aumentar a fidelidade das respostas produzidas pelo modelo de linguagem *gpt-oss-120b*, acessado por meio da API do OpenRouter.

Entre as técnicas investigadas destacam-se:

- **Grounding leve:** instrução mínima de ancoragem que orienta o modelo a localizar as informações no contexto, formular a resposta usando apenas essas informações e citar a fonte ao final. Essa abordagem restringe o modelo a responder exclusivamente com base no contexto recuperado, sem adicionar mecanismos complexos.
- **Few-Shot Prompting:** fornecimento de exemplos representativos do domínio farmacêutico diretamente no corpo do prompt, cobrindo padrões distintos como resposta direta com dados de estabilidade, resposta com vias de administração e comportamento de recusa quando a informação não está disponível no contexto.
- **Chain-of-Thought (CoT):** substituição dos exemplos por um roteiro de raciocínio estruturado em quatro etapas internas, executadas antes da formulação da resposta: (i) classificação da pergunta como factual direta ou envolvendo cálculo; (ii) localização dos valores numéricos, concentrações, volumes, doses, vias e restrições

relevantes no contexto; (iii) verificação da suficiência do contexto, com interrupção e declaração de falta de dados quando insuficiente; e (iv) formulação da resposta conforme o tipo da pergunta.

Essas técnicas buscam aumentar a confiabilidade das respostas em um domínio crítico como o farmacêutico, garantindo que o modelo gere saídas fundamentadas no contexto recuperado e reduza a incidência de informações não verificáveis.

5 EXPERIMENTOS

Este capítulo apresenta a avaliação da proposta desenvolvida neste trabalho por meio de uma série de experimentos, com o objetivo de analisar o impacto das diferentes estratégias adotadas no pipeline RAG sobre a qualidade das respostas geradas e identificar oportunidades de melhoria.

Os experimentos foram organizados em duas fases. A fase preliminar (Experimentos 1 e 2) reproduziu, no Experimento 1, as configurações utilizadas por (COTA, 2025) para estabelecer uma linha de base e identificar limitações relacionadas às etapas de ingestão e recuperação da informação; em seguida, o Experimento 2 reestruturou simultaneamente a ingestão e a recuperação para servir de base à fase incremental. A partir dessa análise preliminar, foi definida a fase incremental cumulativa (Experimentos 3 a 8), na qual cada experimento introduz uma única modificação para avaliar a influência de diferentes estratégias sobre o desempenho do pipeline.

Para garantir uma comparação consistente entre os experimentos, todos foram avaliados utilizando o mesmo conjunto de métricas e o mesmo procedimento experimental, descritos na seção a seguir.

5.1 Avaliação: Framework RAGAS

A avaliação foi realizada utilizando o framework RAGAS, empregado neste trabalho como instrumento de mensuração dos componentes de recuperação e geração, e não como mecanismo de validação absoluta do pipeline. As métricas do RAGAS são calculadas por um modelo de linguagem atuando como juiz (*LLM-as-a-Judge*). Neste trabalho, foi utilizado o modelo DeepSeek V3 0324, acessado por meio da API do OpenRouter.

A utilização de um modelo pertencente a uma família distinta daquela empregada na geração das respostas (*gpt-oss-120b*) busca reduzir o viés decorrente da utilização da mesma arquitetura tanto para gerar quanto para avaliar as respostas. Além disso, o julgamento foi executado com temperatura zero, reduzindo a variabilidade das avaliações.

A escolha do DeepSeek deve-se à sua maior capacidade de raciocínio quando comparada ao modelo gerador, aumentando a confiabilidade da avaliação automática. Ainda assim, essa abordagem constitui uma limitação do estudo, uma vez que o avaliador também é um modelo de linguagem e não houve validação por especialistas humanos. Dessa

forma, os escores obtidos são interpretados como indicadores relativos de desempenho, e não como medidas absolutas de qualidade.

Diferentemente da abordagem adotada por (COTA, 2025), que utilizou a similaridade semântica entre a resposta gerada e uma resposta de referência (*ground truth*), este trabalho emprega métricas que avaliam separadamente os componentes de recuperação e geração do pipeline RAG. Essa estratégia permite identificar, com maior precisão, em qual etapa do sistema ocorrem perdas de desempenho, fornecendo evidências mais úteis para orientar sua otimização.

Entre as métricas avaliadas destacam-se:

- **Fidelidade (*Faithfulness*):** avalia se todas as afirmações na resposta podem ser inferidas a partir do contexto recuperado;
- **Precisão do Contexto (*Context Precision*):** mede se os *chunks* relevantes estão posicionados no topo do ranking recuperado;
- **Recall do Contexto (*Context Recall*):** mede se o contexto recuperado contém toda a informação necessária para responder à pergunta;
- **Correção da Resposta (*Answer Correctness*):** mede a similaridade semântica entre a resposta gerada e a resposta de referência do CIM;
- **Relevância da Resposta (*Answer Relevancy*):** mede se a resposta é tematicamente relevante em relação à pergunta.

Essa estratégia fornece indícios sobre se eventuais erros decorrem da recuperação inadequada de documentos, da segmentação textual ou do processo de geração do modelo de linguagem, orientando onde investir esforço de otimização.

5.2 Metodologia Experimental

A metodologia experimental adotará um delineamento cumulativo: cada experimento herda toda a configuração do anterior e altera um único componente do pipeline RAG. Dessa forma, é possível isolar o efeito de cada modificação sobre as métricas de qualidade. A série experimental incluirá um *baseline* com a fragmentação estrutural proposta e experimentos incrementais que adicionam, sucessivamente, busca híbrida com RRF, reranking semântico, grounding leve no prompt, *few-shot prompting* e *chain-of-thought*. A Figura 5.1 apresenta o delineamento experimental, com a fase preliminar (Experimentos 1 e 2) e a Figura 5.2 a fase incremental cumulativa (Experimentos 3 a 8).

O delineamento cumulativo cumpre, neste trabalho, um papel instrumental: ele é o instrumento de avaliação comparativa do *backend* que será disponibilizado como MVP pronto para implantação. Cada etapa da série acumula evidência sobre se o *pipeline*, ao receber um componente adicional, mantém ou melhora a ancoragem e a correção das respostas. A configuração que, ao final da série, apresentar as maiores Fidelidade e Correção é adotada como *backend* do MVP e destinada a ser exposta ao CIM por meio da interface descrita na Seção 6.1.

O conjunto de avaliação será composto por 70 perguntas elaboradas por profissionais do CIM, cada uma com uma resposta de referência (*ground truth*) produzida pelo próprio CIM, utilizada como base para o cálculo das métricas.

A execução dos experimentos foi automatizada por meio de um script desenvolvido em Python. Inicialmente, o script lê um arquivo CSV contendo as 70 perguntas do conjunto de avaliação. Em seguida, cada pergunta é submetida ao pipeline RAG configurado para o experimento em execução, obtendo a resposta gerada pelo modelo. Ao término da execução, é produzido um arquivo CSV contendo, para cada pergunta, a questão original, a resposta gerada pelo pipeline e a resposta de referência (*ground truth*) elaborada pelo CIM. Esse arquivo é utilizado como entrada para um segundo script, também implementado em Python, responsável por executar a avaliação com o framework RAGAS e calcular as métricas apresentadas neste capítulo.

Em razão das características operacionais dos modelos de linguagem envolvidos, cada experimento é executado em configuração única sobre o conjunto de avaliação, com geração e juiz operando com temperatura zero. As etapas de recuperação (vetorização, RRF e *cross-encoder*) são determinísticas; a variabilidade observada decorre da natureza não determinística dos modelos de linguagem envolvidos, inclusive do juiz do RAGAS. Para cada métrica e cada experimento, computam-se a média e o desvio-padrão (acompanhados da mediana e dos quartis) sobre o conjunto de perguntas. Cabe ressaltar que o desvio-padrão reportado expressa a dispersão *entre perguntas*, e não a incerteza experimental agregada. Em consequência da configuração única, não foi aplicado teste estatístico formal de significância.

5.3 Configuração dos Experimentos

Os experimentos foram organizados em uma fase preliminar (Experimentos 1 e 2) e uma fase incremental cumulativa (Experimentos 3 a 8). O Experimento 1 reprodu-

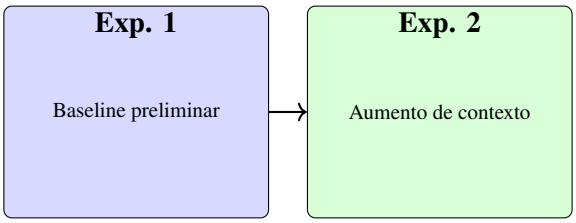


Figura 5.1 – Fase preliminar do delineamento experimental. O Experimento 2 modifica simultaneamente múltiplos componentes do pipeline, sendo utilizado apenas para orientar a definição da configuração base da fase incremental.

Fonte: Elaborado pelo autor (2026).

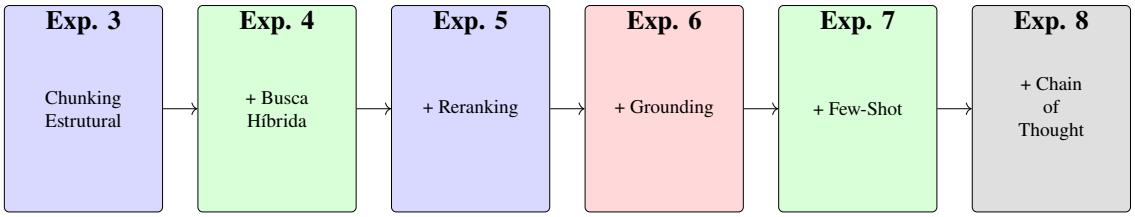


Figura 5.2 – Delineamento experimental: fase incremental cumulativa (Experimentos 3 a 8), na qual cada configuração herda a do anterior e adiciona um único componente.

Fonte: Elaborado pelo autor (2026).

ziu a configuração proposta por (COTA, 2025), estabelecendo uma linha de base para comparação; o Experimento 2 reestrutura simultaneamente a ingestão e a recuperação, servindo de transição para a fase incremental. Nos Experimentos 3 a 8, cada configuração introduz uma única modificação no pipeline RAG. Em todos os experimentos, foi adotada a configuração base apresentada na Tabela 5.1.

Tabela 5.1 – Configuração fixa compartilhada pelos experimentos

Componente	**Configuração**
Conjunto de avaliação (TONELLO, 2026)	70 perguntas e respostas-referência elaboradas pelo CIM
Corpus de medicamentos	*PDF de medicamentos injetáveis com 97 páginas e Excel com 7 planilhas com dados de estabilidade de produtos.*
LLM de geração	*Modelo gpt-oss-120b*, temperatura = *0.0*
LLM-juiz (RAGAS)	*Modelo DeepSeek V3 0324*, temperatura = 0.0
Versão do RAGAS	*0.4.3*
Modelo de *embeddings* (Exp. 2–8)	`intfloat/multilingual-e5-large` (1024 dimensões)
Métricas	Fidelidade, Precisão e Recall do Contexto, Correção e Relevância da Resposta

Fonte: Elaborado pelo autor (2026).

As modificações incrementais descritas na Tabela 5.2 foram planejadas para avaliar, de forma isolada, a contribuição de cada técnica incorporada ao pipeline RAG.

O componente de geração do *pipeline* foi implementado com o *gpt-oss-120b*, acessado por meio da API do OpenRouter. A escolha justifica-se, principalmente, por se

tratar de um modelo de pesos abertos, distribuído sob licença Apache 2.0: essa condição torna o modelo auditável e viabiliza a futura auto-hospedagem do MVP em infraestrutura controlada pelo CIM, evitando a dependência de uma API proprietária de caixa-preta. Vale destacar, ainda, que a abertura dos pesos não compromete o desempenho: em *benchmarks* de raciocínio, o *gpt-oss-120b* apresenta resultados comparáveis aos de modelos proprietários de referência da mesma faixa, como o *o4-mini*. Observa-se que, nesta etapa da pesquisa, o modelo foi consumido por API, de modo que a auto-hospedagem permanece como potencialidade para a implantação futura, aspecto retomado na discussão das limitações.

Tabela 5.2 – Modificação em cada experimento

Exp.	**Componente Alterado**	**Modificação Introduzida**
1	Baseline preliminar	Chunking fixo (500 caracteres), top-k = 3
2	Ingestão e embeddings	Chunking de 1000 caracteres, E5-Large, top-k = 5
3	Baseline deste trabalho	Chunking estrutural
4	Recuperação	Inclusão de busca híbrida (RRF, $k = 60$)
5	Recuperação	Inclusão de reranking com Cross-Encoder
6	Geração	Grounding leve incorporado ao prompt
7	Geração	Few-shot prompting com três exemplos
8	Geração	Chain-of-Thought prompting

Fonte: Elaborado pelo autor (2026).

5.4 Experimentos

Esta seção apresenta os resultados dos experimentos realizados. Para cada experimento, são descritos o objetivo e a configuração adotada, seguidos pelos resultados obtidos e pela respectiva análise, evidenciando o impacto das modificações propostas no desempenho do pipeline RAG.

5.4.1 Experimento 1: Avaliação de Linha de Base

O objetivo deste experimento é estabelecer uma linha de base para o desempenho da arquitetura RAG com a base de conhecimento de medicamentos. Para definir o ponto de partida, a arquitetura foi configurada com as mesmas especificações utilizadas por Cota (2025):

- **Fragmentação (*Chunking*):** blocos de tamanho fixo de 500 caracteres, com sobre-

posição (*overlap*) de 50 caracteres entre blocos adjacentes.

- **Modelo de *Embeddings*:** *paraphrase-multilingual-mpnet-base-v2*, com suporte a processamento de linguagem natural em português.
- **Recuperação (*Retrieval*):** busca vetorial que retorna os 3 fragmentos mais relevantes (*top-k = 3*) baseados na similaridade semântica com a pergunta.
- **Prompt:** instrução simples de resposta direta baseada apenas no contexto recuperado, sem técnicas avançadas de ancoragem. A versão integral do prompt encontra-se no Anexo A.

Durante a execução, o componente de geração retornou "Não sei" para 37 das 70 perguntas. O sistema aplicou as métricas do RAGAS nas 33 respostas geradas com sucesso. A Tabela 5.3 apresenta os resultados.

Tabela 5.3 – Resultados da avaliação RAGAS do pipeline RAG

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	43.10	0.00	50.00	66.67	0.00	100.00	39.99
Precisão do Contexto	60.10	0.00	83.33	100.00	0.00	100.00	43.84
Recall do Contexto	53.03	0.00	100.00	100.00	0.00	100.00	49.91
Correção da Resposta	42.34	22.52	23.01	59.97	21.93	98.12	25.00
Relevância da Resposta	89.86	90.95	92.34	94.45	0.00	99.84	16.40

Fonte: Elaborado pelo autor (2026).

A análise dos resultados evidencia gargalos nas camadas de recuperação e geração do pipeline.

A **Fidelidade** apresentou média de 43,10 e desvio padrão elevado (39,99), indicando comportamento inconsistente do modelo. Em diversos casos, o LLM gerou informações que não constavam nos documentos recuperados, configurando alucinação. Isso sugere limitações tanto na quantidade de contexto recuperado quanto na ausência de mecanismos mais rígidos de *grounding* no prompt.

Na camada de recuperação, a combinação de fragmentos pequenos (500 caracteres) e recuperação limitada a apenas três documentos (*top-k = 3*) restringiu o volume de contexto disponível para o modelo. Como consequência, o **Recall do Contexto** atingiu apenas 53,03, indicando que o sistema frequentemente não consegue recuperar todos os trechos necessários para responder às consultas.

A **Precisão do Contexto** (60,10) complementa esse diagnóstico ao mostrar que parte dos fragmentos recuperados era pouco relevante para a consulta, evidenciando limitações na capacidade discriminativa da busca vetorial.

Como efeito dessas restrições, a **Correção da Resposta** apresentou média de

42,34. O modelo recebe contexto parcial e tenta compensar inferindo informações que não estão na base documental, o que reduz a exatidão das respostas.

A única métrica com desempenho elevado foi a **Relevância da Resposta** (89,86). Esse resultado mostra que o modelo compreende bem a intenção das consultas e produz respostas alinhadas ao domínio farmacêutico. No entanto, relevância textual não significa precisão factual.

5.4.2 Experimento 2: Reestruturação da Ingestão e Ampliação de Contexto

Com base nos problemas identificados na linha de base, o segundo experimento reestrutura a estratégia de *chunking* e substitui o modelo de *embeddings*. O objetivo é fornecer ao modelo de geração um contexto mais coerente e completo, reduzindo a taxa de abstenção ("Não sei") e melhorando o Recall do Contexto.

As seguintes alterações foram implementadas em relação ao Experimento 1:

- **Fragmentação (*Chunking*):** blocos expandidos de 500 para 1000 caracteres, com sobreposição ajustada para 150 caracteres.
- **Modelo de *Embeddings*:** substituído pelo *multilingual-e5-large*.
- **Recuperação (*Retrieval*):** limite de documentos recuperados ampliado de 3 para 5 fragmentos (*top-k* = *5*).
- **Modelo de Linguagem e Dataset:** mantidos idênticos ao Experimento 1 para garantir o isolamento de variáveis.

A mudança do tamanho do *chunk* para 1000 caracteres busca preservar a integridade semântica dos dados do domínio de medicamentos. Essa expansão tornou necessária a substituição do modelo de *embeddings*: o modelo anterior (*paraphrase-multilingual-mpnet-base-v2*) tinha um limite de 128 *tokens* (cerca de 500 caracteres). Manter esse modelo com blocos de 1000 caracteres causaria truncamento silencioso durante a vetorização, perdendo informação de forma imperceptível. O *multilingual-e5-large* resolve esse problema com janela de contexto nativa de até 512 *tokens*.

A ampliação do *top-k* para 5 trabalha em conjunto com os blocos maiores. O pipeline passa a injetar cerca de 5000 caracteres de contexto no prompt do LLM, contra 1500 no experimento anterior. A expectativa é que esse volume maior de informação reduza as respostas nulas e melhore a correção.

Na avaliação com 70 perguntas, o componente de geração produziu respostas para

Tabela 5.4 – Resultados da avaliação RAGAS do pipeline RAG (Experimento 2)

Métrica	Média	Q25%	Q50%	Q75%	Mín	Máx	Std
Fidelidade	56.15	41.25	55.56	75.00	0.00	100.00	29.13
Precisão do Contexto	75.26	58.61	83.33	100.00	0.00	100.00	27.99
Recall do Contexto	50.00	0.00	50.00	100.00	0.00	100.00	50.00
Correção da Resposta	42.79	30.38	39.69	53.77	21.48	93.25	17.16
Relevância da Resposta	87.46	90.60	93.29	95.39	0.00	99.92	23.98

Fonte: Elaborado pelo autor (2026).

59 consultas. A taxa de abstenção caiu de aproximadamente 52% no Experimento 1 para 15% nesta iteração, validando a decisão de expandir o contexto. O LLM passou a receber informação suficiente para gerar respostas com mais frequência.

A Tabela 5.4 apresenta as métricas obtidas pelo RAGAS sobre as 59 respostas.

A **Precisão do Contexto** evoluiu de 60,10 para 75,26, com redução no desvio padrão. Isso indica que o modelo *multilingual-e5-large*, combinado com fragmentos maiores, produziu representações vetoriais mais estáveis, reduzindo ruído na recuperação.

A **Fidelidade** também cresceu de forma relevante, passando de 43,10 para 56,15. O modelo passou a gerar menos informações não fundamentadas nos documentos.

Por outro lado, o **Recall do Contexto** permaneceu praticamente estagnado, variando de 53,03 para 50,00. Isso evidencia uma limitação da fragmentação por tamanho fixo: fragmentos extensos passaram a agregar múltiplos tópicos distintos em um único vetor, reduzindo a especificidade semântica dos *embeddings*.

A estagnação do Recall impactou a **Correção da Resposta**, que permaneceu praticamente inalterada (42,79). Embora o sistema tenha reduzido alucinações, a ausência recorrente de informações críticas no contexto continuou limitando a exatidão.

5.4.3 Experimento 3: Baseline com Chunking Estrutural

O Experimento 3 estabelece o *baseline* do pipeline RAG proposto neste trabalho. Em relação ao Experimento 2, a única alteração é a estratégia de fragmentação na camada de ingestão. Todos os demais componentes foram mantidos idênticos para isolar o efeito dessa variável.

Em vez de fragmentação por tamanho fixo, este experimento usa fragmentação estrutural (*Document-Aware Chunking*). Cada registro de medicamento é convertido em blocos textuais onde os campos originais da bula (“Apresentação”, “Reconstituição”, “Via de administração”, “Estabilidade”, etc.) são preservados com rótulos semânticos. Cada

chunk contém um campo rotulado, produzindo vetores semanticamente mais puros e mitigando a diluição vetorial observada no Experimento 2.

A busca é puramente vetorial: a pergunta do usuário é convertida em *embedding* e os 5 *chunks* mais similares são retornados via busca por vizinhança.

- **Fragmentação:** substituição da segmentação por tamanho fixo (1000 caracteres) por fragmentação estrutural por seções lógicas da bula.
- **Modelo de *Embeddings*:** *multilingual-e5-large* (1024 dimensões), mantido idêntico ao Experimento 2.
- **Recuperação:** busca puramente vetorial por similaridade de cosseno, com *top-k* = *5*.
- **Geração:** configurações idênticas ao Experimento 2.

Tabela 5.5 – Resultados da avaliação RAGAS, Experimento 3

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	75.58	60.65	75.00	92.56	25.00	100.00	19.09
Precisão do Contexto	76.03	64.73	83.63	100.00	0.00	100.00	27.95
Recall do Contexto	74.75	45.83	100.00	100.00	0.00	100.00	42.95
Correção da Resposta	54.03	38.46	52.34	67.83	22.14	98.88	20.04
Relevância da Resposta	89.66	89.47	90.83	92.61	0.00	96.27	11.32

Fonte: Elaborado pelo autor (2026).

A Tabela 5.6 compara os Experimentos 2 e 3.

Tabela 5.6 – Comparação entre Experimentos 2 e 3

Métrica	**Exp 2**	**Exp 3**	Δ
Fidelidade	56.15	75.58	+19.43
Precisão do Contexto	75.26	76.03	+0.77
Recall do Contexto	50.00	74.75	+24.75
Correção da Resposta	42.79	54.03	+11.24
Relevância da Resposta	87.46	89.66	+2.20

Fonte: Elaborado pelo autor (2026).

A fragmentação estrutural produziu o maior salto isolado da série experimental. O **Recall do Contexto** subiu de 50,00 para 74,75 (+24,75), revertendo a estagnação observada no Experimento 2. Esse resultado confirma a premissa de projeto: ao preservar a relação semântica entre os campos da bula em *chunks* coesos, a recuperação passa a alcançar trechos que a fragmentação por tamanho fixo diluía em um único vetor.

A **Fidelidade** acompanhou o ganho de contexto (+19,43), pois um contexto mais completo reduz a necessidade de o modelo inferir informações ausentes, e a **Correção da Resposta** também evoluiu (+11,24). A **Precisão do Contexto** permaneceu praticamente

estável (+0,77), indicando que o avanço não veio de posicionar melhor os *chunks* relevantes no topo do ranking, mas de recuperar trechos que antes não eram alcançados. A **Relevância** variou marginalmente (+2,20).

Apesar do salto, o ponto de referência ainda apresenta limitações. O Recall do Contexto, com desvio padrão de 42,95 e percentil 25 de 45,83, indica que aproximadamente um quarto das perguntas ainda recebe recuperação insuficiente. A Correção da Resposta permanece relativamente baixa, mostrando que mesmo com contexto adequado o modelo nem sempre gera respostas alinhadas à referência do CIM, e o valor mínimo de 0,00 na Relevância sinaliza a presença de *outliers*, ou seja, respostas completamente fora do escopo.

Esses resultados justificam adotar o Experimento 3 como *baseline* deste trabalho e motivam os aprimoramentos subsequentes de recuperação (Experimentos 4 e 5).

5.4.4 Experimento 4: Busca Híbrida

O Experimento 4 substitui a busca puramente vetorial por uma busca híbrida que combina busca semântica (*embeddings*) com busca lexical (*full-text search*). Os rankings são combinados via RRF com $k = 60$.

Tabela 5.7 – Resultados da avaliação RAGAS, Experimento 4

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	80.21	67.74	83.33	100.00	33.33	100.00	19.66
Precisão do Contexto	78.69	64.88	92.67	100.00	0.00	100.00	27.96
Recall do Contexto	72.95	0.00	100.00	100.00	0.00	100.00	44.38
Correção da Resposta	53.27	37.55	50.46	65.36	21.77	98.48	21.63
Relevância da Resposta	89.83	89.37	91.38	92.55	0.00	97.76	11.25

Fonte: Elaborado pelo autor (2026).

A Tabela 5.8 compara os Experimentos 3 e 4.

Tabela 5.8 – Comparação entre Experimentos 3 e 4

Métrica	**Exp 3**	**Exp 4**	Δ
Fidelidade	75.58	80.21	+4.63
Precisão do Contexto	76.03	78.69	+2.66
Recall do Contexto	74.75	72.95	−1.80
Correção da Resposta	54.03	53.27	−0.76
Relevância da Resposta	89.66	89.83	+0.17

Fonte: Elaborado pelo autor (2026).

A busca híbrida produziu o maior ganho isolado de Fidelidade entre as intervenções de recuperação (+4,63), posteriormente superado, na fase de geração, pelo *grounding* leve

no Experimento 6 (+7,72). O Recall do Contexto apresentou variação marginal (−1,80), enquanto a Precisão do Contexto melhorou (+2,66). Correção e Relevância permaneceram estáveis.

O resultado confirma que a combinação de busca semântica e lexical captura informações que cada abordagem isoladamente perde. Termos técnicos farmacêuticos (nomes de medicamentos, concentrações exatas) são melhor recuperados pelo componente lexical, enquanto relações semânticas são capturadas pelo componente vetorial.

5.4.5 Experimento 5: Reranking Semântico (Cross-Encoder)

O Experimento 5 adiciona uma camada de *reranking* ao pipeline. Após a busca híbrida retornar 10 chunks, o modelo *cross-encoder/mmarco-mMiniLMv2-L12-H384-v1* pontua cada par (pergunta, chunk) e reordena os resultados por relevância, selecionando os cinco chunks mais relevantes.

O *cross-encoder* é um modelo de classificação de pares que, diferentemente dos bi-encoders usados para gerar *embeddings*, processa a pergunta e o chunk de forma conjunta, capturando interações semânticas mais ricas. O modelo selecionado é multilingual e otimizado para busca em português.

Tabela 5.9 – Resultados da avaliação RAGAS, Experimento 5

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	74.41	60.00	78.24	100.00	20.00	100.00	23.19
Precisão do Contexto	79.72	62.50	100.00	100.00	0.00	100.00	31.10
Recall do Contexto	77.21	100.00	100.00	100.00	0.00	100.00	41.82
Correção da Resposta	54.84	36.51	56.42	72.46	21.67	98.56	21.18
Relevância da Resposta	90.82	89.35	91.08	92.33	83.83	96.73	2.56

Fonte: Elaborado pelo autor (2026).

A Tabela 5.10 compara os Experimentos 4 e 5.

Tabela 5.10 – Comparação entre Experimentos 4 e 5

Métrica	**Exp 4**	**Exp 5**	Δ
Fidelidade	80.21	74.41	−5.80
Precisão do Contexto	78.69	79.72	+1.03
Recall do Contexto	72.95	77.21	+4.26
Correção da Resposta	53.27	54.84	+1.57
Relevância da Resposta	89.83	90.82	+0.99

Fonte: Elaborado pelo autor (2026).

O *reranking* melhorou o Recall do Contexto (+4,26), enquanto a Precisão apresentou apenas uma variação discreta (+1,03), mantendo-se essencialmente estável. Esses

resultados confirmam que a reordenação por relevância é capaz de posicionar *chunks* mais informativos nas primeiras posições do ranking.

A Fidelidade, no entanto, apresentou queda (−5,80). Uma hipótese é que o *reranker* promove chunks com alta relevância lexical que podem conter informações periféricas, levando o modelo a incluir detalhes nem sempre ancorados no contexto. Outra possibilidade é a variabilidade natural da API de geração.

A Correção da Resposta apresentou uma melhora discreta (+1,57) e, considerando a variabilidade inerente ao processo de geração do modelo, esse resultado pode ser interpretado como estabilidade da métrica.

5.4.6 Experimento 6: Grounding Leve no Prompt

O Experimento 6 substitui o prompt de geração utilizado desde o Experimento 1 por uma versão com instrução mínima de ancoragem. Todos os demais componentes são mantidos idênticos ao Experimento 5. A mudança isola o efeito da instrução ao modelo na qualidade das respostas.

O prompt foi simplificado para três passos diretos: (1) localize as informações no contexto, (2) formule a resposta usando apenas essas informações, (3) cite a fonte ao final. A ideia é que uma instrução enxuta seja suficiente para ancorar o modelo no contexto sem adicionar complexidade desnecessária. O texto completo do prompt é apresentado no Apêndice A.

Tabela 5.11 – Resultados da avaliação RAGAS, Experimento 6

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	82.13	72.92	100.00	100.00	0.00	100.00	29.86
Precisão do Contexto	79.58	70.00	100.00	100.00	0.00	100.00	30.03
Recall do Contexto	72.92	0.00	100.00	100.00	0.00	100.00	44.39
Correção da Resposta	59.52	35.40	60.88	73.64	19.53	99.25	25.72
Relevância da Resposta	87.65	89.65	91.51	93.89	0.00	98.75	19.84

Fonte: Elaborado pelo autor (2026).

A Tabela 5.12 compara os Experimentos 5 e 6.

A **Fidelidade** apresentou o maior ganho isolado da fase de geração (+7,72), atingindo 82,13. A mediana atingiu 100,00 e o percentil 75 também igual a 100,00, indicando que a maioria das respostas é completamente fundamentada no contexto. A **Correção da Resposta** atingiu 59,52 (+4,68), o melhor valor até este ponto, com a mediana subindo de 56,42 para 60,88.

Tabela 5.12 – Comparação entre os Experimentos 5 e 6

Métrica	**Exp 5**	**Exp 6**	Δ
Fidelidade	74.41	82.13	+7.72
Precisão do Contexto	79.72	79.58	−0.14
Recall do Contexto	77.21	72.92	−4.29
Correção da Resposta	54.84	59.52	+4.68
Relevância da Resposta	90.82	87.65	−3.17

Fonte: Elaborado pelo autor (2026).

Esses resultados mostram que uma instrução simples de ancoragem no prompt é suficiente para melhorar de forma consistente a fidelidade das respostas. A simplicidade do prompt se mostrou vantajosa: sem mecanismos complexos, o modelo consegue gerar respostas mais fundamentadas no contexto.

As métricas de recuperação apresentaram queda (−4,29 Recall, −0,14 Precisão), o que mostra que a variabilidade é atribuível à natureza não determinística da API e não ao prompt.

5.4.7 Experimento 7: Few-Shot no Prompt de Geração

O Experimento 7 adiciona três exemplos *few-shot* ao prompt de geração. Os exemplos cobrem padrões distintos do domínio farmacêutico:

1. **Estabilidade:** resposta direta com dado temporal e condições de armazenamento (ciprofloxacino, pronto uso, descartar).
2. **Via de administração:** resposta com vias permitidas, velocidade de infusão e observações relevantes (glucagon, EV direto 3 a 5 min, via opcional).
3. **Informação ausente:** demonstra o comportamento correto quando o contexto não contém a resposta (paclitaxel, "não sei").

Todos os demais componentes são idênticos ao Experimento 6. A variável isolada é o efeito da técnica de *few-shot prompting* sobre a qualidade da geração. O texto completo do prompt é apresentado no Apêndice B.

A Tabela 5.13 apresenta os resultados do Experimento 7.

A Tabela 5.14 compara os Experimentos 6 e 7.

Os incrementos de Fidelidade (+1,14) e Correção (+1,43) foram pequenos e, considerando a variabilidade do processo de geração e avaliação, indicam estabilidade dessas métricas. Assim, o principal benefício do *few-shot* não está no aumento das médias, mas

Tabela 5.13 – Resultados da avaliação RAGAS, Experimento 7

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	83.27	66.67	100.00	100.00	20.00	100.00	21.16
Precisão do Contexto	84.04	75.28	100.00	100.00	0.00	100.00	28.09
Recall do Contexto	78.84	100.00	100.00	100.00	0.00	100.00	40.74
Correção da Resposta	60.95	46.11	60.61	79.44	21.99	98.82	23.69
Relevância da Resposta	86.85	89.17	90.67	93.15	0.00	96.76	19.76

Fonte: Elaborado pelo autor (2026).

Tabela 5.14 – Comparação entre os Experimentos 6 e 7

Métrica	**Exp 6**	**Exp 7**	Δ
Fidelidade	82.13	83.27	+1.14
Precisão do Contexto	79.58	84.04	+4.46
Recall do Contexto	72.92	78.84	+5.92
Correção da Resposta	59.52	60.95	+1.43
Relevância da Resposta	87.65	86.85	−0.80

Fonte: Elaborado pelo autor (2026).

na maior consistência das respostas.

Na Fidelidade, o desvio-padrão reduziu de 29,86 para 21,16, indicando respostas mais homogêneas e com menos casos extremos de baixa aderência ao contexto. Na Correção, o percentil 25 aumentou de 35,40 para 46,11 (+10,71), mostrando que o *few-shot* melhora principalmente as respostas de menor qualidade. Esse "efeito de piso" é mais relevante do que os ganhos observados na média.

Em conjunto, a redução da dispersão na Fidelidade e a elevação do piso da Correção indicam que o Experimento 7 produz respostas mais consistentes e mais seguras nos casos desfavoráveis. Por esse motivo, essa configuração foi adotada como *backend* do MVP destinado ao CIM.

As métricas de recuperação (Precisão +4,46 e Recall +5,92) não são atribuídas ao *few-shot*, uma vez que a etapa de recuperação é independente do prompt. Essas diferenças refletem a variabilidade natural do LLM.

Por fim, a Relevância apresentou variação marginal ($-0,80$), permanecendo estável.

Em síntese, o *few-shot* agrega valor ao *grounding* leve ao tornar as respostas mais consistentes e elevar a qualidade dos casos mais críticos, justificando sua adoção na configuração final do sistema.

5.4.8 Experimento 8: Chain-of-Thought (CoT) no Prompt de Geração

O Experimento 8 adiciona a técnica de Chain-of-Thought (CoT) ao prompt de geração. O prompt completo é apresentado no Apêndice C.

Enquanto o Experimento 7 orienta o modelo por meio de exemplos pergunta–resposta, o novo prompt substitui os exemplos por um roteiro de raciocínio estruturado em quatro etapas internas, executadas antes da formulação da resposta:

1. **Classificação:** a pergunta é rotulada como TIPO A (factual direta: estabilidade, via de administração, compatibilidade) ou TIPO B (envolve cálculo: dosagem por peso, volume de diluição, concentração final, taxa de infusão).
2. **Localização:** extração de todos os valores numéricos, concentrações, volumes, doses, vias e restrições relevantes presentes no contexto.
3. **Verificação:** conferência da suficiência do contexto; caso seja insuficiente, o modelo interrompe e declara falta de dados.
4. **Formulação:** a resposta segue o *template* correspondente ao tipo da pergunta.

Todos os demais componentes são idênticos ao Experimento 7.

A Tabela 5.15 apresenta os resultados do Experimento 8.

Tabela 5.15 – Resultados da avaliação RAGAS, Experimento 8

Métrica	**Média**	**Q25%**	**Q50%**	**Q75%**	**Mín**	**Máx**	**Std**
Fidelidade	62.67	50.00	66.67	75.00	0.00	100.00	28.09
Precisão do Contexto	91.42	91.67	100.00	100.00	25.00	100.00	18.49
Recall do Contexto	74.56	50.00	100.00	100.00	0.00	100.00	42.39
Correção da Resposta	54.33	44.00	53.58	60.69	21.64	88.57	17.25
Relevância da Resposta	93.36	89.66	94.73	97.36	83.19	99.95	4.60

Fonte: Elaborado pelo autor (2026).

Tabela 5.16 – Comparação entre os Experimentos 7 (Few-Shot) e 8 (Chain-of-Thought)

Métrica	**Exp 7**	**Exp 8**	Δ
Fidelidade	83.27	62.67	−20.60
Precisão do Contexto	84.04	91.42	+7.38
Recall do Contexto	78.84	74.56	−4.28
Correção da Resposta	60.95	54.33	−6.62
Relevância da Resposta	86.85	93.36	+6.51

Fonte: Elaborado pelo autor (2026).

Ao contrário do esperado, a introdução do *Chain-of-Thought* estruturado reduziu drasticamente a Fidelidade (−20,60), ao mesmo tempo em que elevou a Precisão do Contexto (+7,38) e a Relevância da Resposta (+6,51). Esse padrão é compatível com a

hipótese de que o raciocínio passo a passo induz o modelo a inferir informações ausentes do contexto, sobretudo nas perguntas do TIPO B (envolvendo cálculo). As respostas resultantes são fluentes e topicalmente relevantes, porém desancoradas, configurando um caso de alucinação induzida por raciocínio.

A **Correção da Resposta** (−6,62) confirma o efeito negativo sobre a exatidão, ao passo que o aumento da Relevância indica que as respostas permanecem bem formuladas do ponto de vista linguístico. A literatura reporta que técnicas de CoT tendem a aumentar a confiança aparente da resposta sem, necessariamente, garantir ancoragem factual, o que é particularmente crítico em domínio clínico, no qual uma inferência incorreta pode ter consequências graves.

Desse modo, a melhor configuração da série experimental em termos de Fidelidade e Correção permanece sendo a do Experimento 7 (*few-shot* sobre *grounding* leve).

5.5 Síntese da Evolução

A Tabela 5.17 mostra uma evolução gradual do desempenho do pipeline ao longo dos experimentos. O Experimento 1 apresentou os menores valores de Fidelidade (43,10) e Correção (42,34), evidenciando limitações iniciais na recuperação e na geração das respostas. Com a ingestão baseada em embeddings (Experimento 2), a Fidelidade aumentou para 56,15 e a Precisão de Contexto para 75,26, enquanto o Recall permaneceu praticamente estagnado.

O *chunking* estrutural (Experimento 3) promoveu o maior ganho inicial, elevando a Fidelidade (75,58), o Recall (74,75) e a Correção (54,03). A busca híbrida com RRF (Experimento 4) aumentou a Fidelidade para 80,21, sem ganhos relevantes em Correção. O reranqueamento com *Cross-Encoder* (Experimento 5) alcançou o maior Recall até então (77,21) e um aumento discreto na Correção (54,84).

As maiores melhorias em Correção ocorreram com mudanças no prompt. O *grounding* leve (Experimento 6) elevou a Correção para 59,52 e a Fidelidade para 82,13. O *few-shot prompting* (Experimento 7) apresentou o melhor desempenho geral, alcançando os maiores valores de Fidelidade (83,27), Recall (78,84) e Correção (60,95) e elevada Precisão (84,04).

Por fim, o *Chain-of-Thought* (Experimento 8) obteve a maior Precisão de Contexto (91,42) e Relevância de Resposta (93,36), mas reduziu a Fidelidade (62,67) e a Correção (54,33), indicando que o raciocínio explícito não trouxe benefícios para este cenário de

perguntas factuais baseadas em conhecimento recuperado. A Figura 5.3 apresenta a evolução gráfica dessas métricas ao longo dos experimentos.

Tabela 5.17 – Síntese da evolução das médias das métricas (Experimentos 1 a 8: fase preliminar 1–2 e incremental 3–8)

Exp	Técnica incorporada	Fidel.	Precis.	Recall	Correção	Relev.
1	Baseline preliminar	43.10	60.10	53.03	42.34	89.86
2	Ingestão e embeddings	56.15	75.26	50.00	42.79	87.46
3	Baseline (chunking estrutural)	75.58	76.03	74.75	54.03	89.66
4	Busca híbrida (RRF)	80.21	78.69	72.95	53.27	89.83
5	Reranking (Cross-Encoder)	74.41	79.72	77.21	54.84	90.82
6	Grounding leve no prompt	82.13	79.58	72.92	59.52	87.65
7	Few-shot prompting	83.27	84.04	78.84	60.95	86.85
8	Chain-of-Thought	62.67	91.42	74.56	54.33	93.36

Fonte: Elaborado pelo autor (2026).

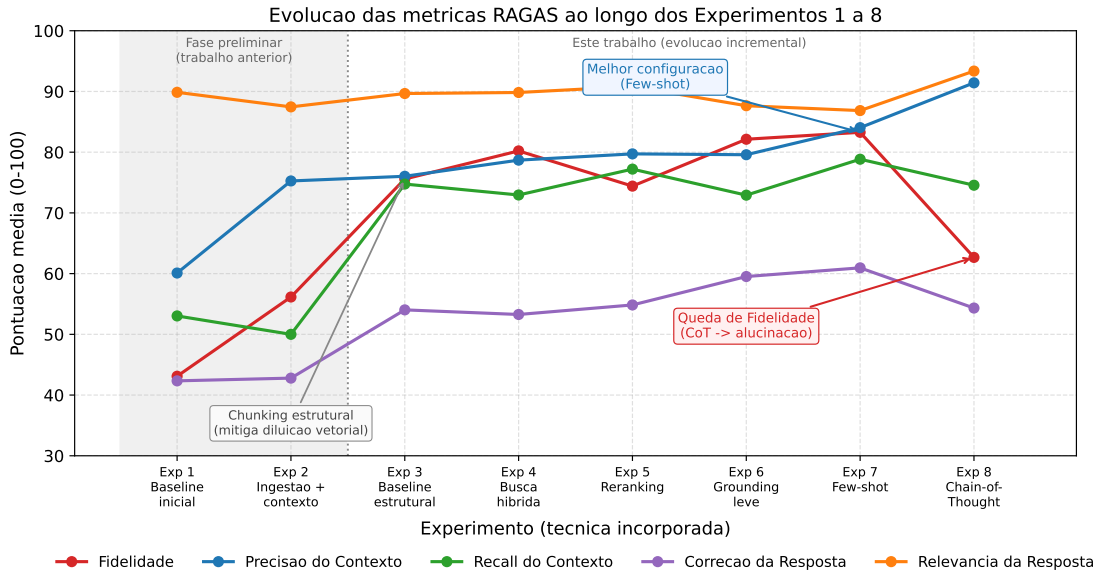


Figura 5.3 – Evolução das métricas RAGAS nos Experimentos 1 a 8. A faixa cinza corresponde a fase preliminar (Exp. 1–2); a direita da linha pontilhada, a evolução incremental deste trabalho (Exp. 3–8). O salto entre os Exp. 2 e 3 evidencia o ganho do chunking estrutural; os Exp. 6 e 7 concentram os melhores resultados em Fidelidade e Correcao; o CoT (Exp. 8) reverte a ancoragem.

5.6 Análise das Abstenções ao Longo da Série Experimental

Ao longo da série experimental, o componente de geração retornou respostas abstentivas (“Não sei”) em proporções distintas. Como todas as perguntas do conjunto de avaliação possuem uma resposta de referência fornecida pelo CIM, uma abstenção representa sempre uma falha em entregar a resposta esperada, e não um caso sem gabarito.

A Tabela 5.18 consolida a quantidade de abstenções e a cobertura (proporção de perguntas efetivamente respondidas) em cada experimento.

Tabela 5.18 – Abstenções (“Não sei”) e cobertura por experimento, sobre as 70 perguntas de avaliação

Exp	**“Não sei”**	**Respondidas**	**Cobertura (%)**
1	37	33	47.1
2	11	59	84.3
3	1	69	98.6
4	0	70	100.0
5	1	69	98.6
6	7	63	90.0
7	5	65	92.9
8	8	62	88.6

Fonte: Elaborado pelo autor (2026).

As métricas RAGAS foram computadas exclusivamente sobre as respostas efetivamente geradas, de modo que o tamanho do conjunto avaliado varia entre 33 (Experimento 1) e 70 (Experimento 4). Como o modelo tende a se abster preferencialmente das perguntas cujo contexto recuperado é insuficiente, a exclusão das abstenções não opera de forma aleatória: ela remove sistematicamente os casos mais desfavoráveis. Em consequência, variações de métrica entre experimentos com coberturas muito distintas devem ser interpretadas com cautela, pois parte do delta pode refletir uma mudança no subconjunto avaliado, e não apenas na qualidade do pipeline.

Ainda assim, a evolução da cobertura é coerente com as intervenções realizadas e oferece evidência complementar às métricas. As melhorias na ingestão e na recuperação (Experimentos 2 a 5) reduziram as abstenções de 11 para próximo de zero, sendo o *chunking* estrutural (Experimento 3) responsável pela maior queda isolada (de 15,7% para 1,4% de abstenção). Isso indica que a reestruturação do contexto não apenas elevou as métricas, mas aumentou a própria capacidade do modelo de produzir respostas. Já os Experimentos 6 a 8 apresentaram um aumento controlado das abstenções (5 a 8 casos), coincidente com a introdução das instruções de *grounding*, que orientam o modelo a declinar da resposta quando o contexto recuperado não sustenta uma afirmação. Em um domínio clínico, essa abstenção induzida é desejável: prefere-se uma recusa explícita a uma resposta desancorada.

Por fim, os ganhos mais expressivos (*chunking* estrutural, *grounding* leve e *few-shot prompting*, bem como a degradação observada com *Chain-of-Thought*) ocorrem justamente nos pontos em que as variações de cobertura são menores.

5.7 Análise Geral dos Resultados

A Tabela 5.19 resume os ganhos obtidos até a melhor configuração, o Experimento 7. Em relação ao Experimento 1, os maiores avanços ocorreram em Fidelidade (+40,17), Recall do Contexto (+25,81) e Precisão do Contexto (+23,94), indicando que as mudanças na ingestão e recuperação melhoraram de forma consistente a qualidade das evidências recuperadas.

Tabela 5.19 – Ganho cumulativo das métricas até a melhor configuração (Experimento 7)

Métrica	**Δ Exp. 1 $\rightarrow$ 7**	**Δ Exp. 3 $\rightarrow$ 7**
Fidelidade	+40.17	+7.69
Precisão do Contexto	+23.94	+8.01
Recall do Contexto	+25.81	+4.09
Correção da Resposta	+18.61	+6.92
Relevância da Resposta	-3.01	-2.81

Fonte: Elaborado pelo autor (2026).

O *chunking* estrutural (Experimento 3) foi a intervenção com maior impacto inicial, elevando substancialmente o Recall e a Fidelidade. As estratégias seguintes de recuperação, busca híbrida e *reranking*, produziram ganhos menores, sugerindo que o principal gargalo passou a estar na geração das respostas.

As maiores melhorias em Fidelidade e Correção vieram da engenharia de *prompt*. O *grounding* leve (Experimento 6) proporcionou o maior ganho isolado em Fidelidade, enquanto o *few-shot prompting* (Experimento 7) apresentou o melhor desempenho geral, alcançando os maiores valores de Fidelidade (83,27) e Correção (60,95), além de respostas mais consistentes.

A Relevância da Resposta permaneceu elevada em todos os experimentos, inclusive em cenários com menor Fidelidade. Esse resultado mostra que respostas aparentemente relevantes nem sempre estão adequadamente ancoradas no contexto recuperado. Por isso, a avaliação conjunta das métricas é essencial, especialmente em aplicações clínicas.

Por fim, na ausência de teste estatístico formal, evita-se atribuir significância a diferenças pequenas entre experimentos, interpretadas como estabilidade da métrica; os ganhos observados com o *chunking* estrutural, o *grounding* leve e o *few-shot prompting* são amplos em magnitude e consistentes em direção, sendo tratados como efeitos robustos.

Dessa forma, a configuração recomendada corresponde ao Experimento 7, composta por *chunking* estrutural, busca híbrida com *reranking* por *cross-encoder*, *grounding* leve e *few-shot prompting*. O *Chain-of-Thought* (Experimento 8) não apresentou benefícios para este cenário de perguntas factuais ancoradas em base de conhecimento.

A série experimental também funcionou como um instrumento de avaliação comparativa para o *backend* RAG do MVP destinado ao CIM. Assim, o MVP adota a configuração do Experimento 7 e restringe sua utilização ao modo de pergunta e resposta única (*single-shot*), correspondente ao cenário efetivamente coberto pelos experimentos.

5.7.1 Ameaças à Validade

A interpretação dos resultados experimentais está sujeita a ameaças à validade que delimitam a confiança nos escores obtidos:

- **Juiz LLM não validado contra humano:** as métricas do RAGAS são computadas pelo DeepSeek como juiz, sem contraprova contra julgamento humano do CIM. Ainda que o juiz seja de capacidade superior à do gerador (gpt-oss-120b), o uso de um LLM como juiz permanece suscetível a vieses inerentes, de modo que os escores devem ser interpretados como sinais relativos de otimização entre os experimentos da série cumulativa, e não como medidas absolutas de qualidade.
- **Configuração única e ausência de teste estatístico formal:** os experimentos foram conduzidos em configuração única, sem repetições que permitissem estimar a incerteza agregada das métricas ou aplicar testes de significância. A variabilidade residual do juiz e o roteamento interno da API introduzem ruído, de modo que os resultados são interpretados pela direção e pela magnitude das variações, e não por significância estatística; apenas as variações de maior magnitude (notadamente as atribuíveis à fragmentação estrutural e ao *grounding* leve) são tratadas como efeitos robustos, ao passo que diferenças pequenas são consideradas estabilidade da métrica.
- **Reuso do conjunto de avaliação:** as mesmas 70 perguntas foram empregadas tanto para selecionar a melhor configuração quanto para reportar o desempenho, de modo que os valores apresentados constituem limites superiores sobre o conjunto usado nas decisões e podem superestimar o desempenho em perguntas inéditas.

Tais ameaças não invalidam a série experimental como instrumento de avaliação comparativa do *backend*, mas circunscrevem a interpretação dos escores a sinais relativos de otimização entre as configurações; a validação de utilidade e segurança em condições reais de uso permanece reservada ao CIM, por meio da interface do MVP.

6 DEMONSTRAÇÃO

Este capítulo demonstra a interface em execução, em ambiente de desenvolvimento, para evidenciar comportamentos que as métricas automáticas não capturam de forma isolada: a ancoragem da resposta ao contexto recuperado (com citação da fonte) e o comportamento de recusa diante de informações indisponíveis.

6.1 Produto Mínimo Viável para o CIM

A série experimental descrita neste trabalho visa otimizar o *backend* RAG. Para que essa otimização se traduza em valor concreto para o CIM, o *backend* é disponibilizado como um MVP, por meio de uma interface de uso destinada a expor a configuração recomendada (Experimento 7) aos profissionais, em condições reais de consulta. A interface é, portanto, o entregável central deste trabalho: o veículo por meio do qual a otimização medida no Capítulo 5 alcançará os profissionais em etapa futura de implantação. É também por meio dela que será coletada a validação de utilidade e segurança que as métricas automáticas não conseguem capturar, em particular o comportamento de recusa diante de informações indisponíveis.

A interface adota um modelo de interação de pergunta–resposta única: o farmacêutico formula uma consulta, o sistema retorna uma resposta fundamentada no contexto recuperado, com a citação da fonte, e a interação encerra-se. Não há memória de conversa nem diálogo multi-turno. Essa escolha é uma decisão deliberada de escopo, fundamentada no que a série experimental efetivamente avaliou: todos os experimentos avaliam perguntas isoladas, sem histórico, de modo que a série cobre exatamente o modo de operação *single-shot*. Introduzir um componente de memória de conversa, que nunca foi testado, estenderia a interface para além do escopo avaliado e reabriria, sem controle, o modo de falha mais sensível em domínio clínico. O modelo *single-shot* mantém a interface inteiramente dentro do que foi mensurado.

A implementação organiza-se nas três camadas projetadas, cada uma realizada por um componente distinto da pilha tecnológica:

- **Camada de uso (interface):** aplicação web em *Streamlit*, que oferece o formulário de consulta e a área de resposta com a citação da fonte.

- **Camada de serviço (API):** serviço em *FastAPI* que recebe a pergunta, orquestra a recuperação híbrida com *reranking* e a geração com *grounding* leve e *few-shot*, e devolve a resposta e a fonte.
- **Camada de conhecimento (*backend* RAG):** o pipeline do Experimento 7 (fragmentação estrutural, *embeddings multilingual-e5-large*, busca híbrida com RRF, *reranking* por *cross-encoder*).

6.2 Cenários de Uso

Esta seção apresenta cenários representativos de utilização do MVP, com o objetivo de demonstrar o comportamento do sistema em situações típicas de consulta ao CIM. Os exemplos ilustram tanto a geração de respostas fundamentadas no contexto recuperado quanto o comportamento de recusa diante de informações não disponíveis na base de conhecimento, complementando a avaliação quantitativa apresentada no Capítulo 5. A Figura 6.1 apresenta a tela inicial.

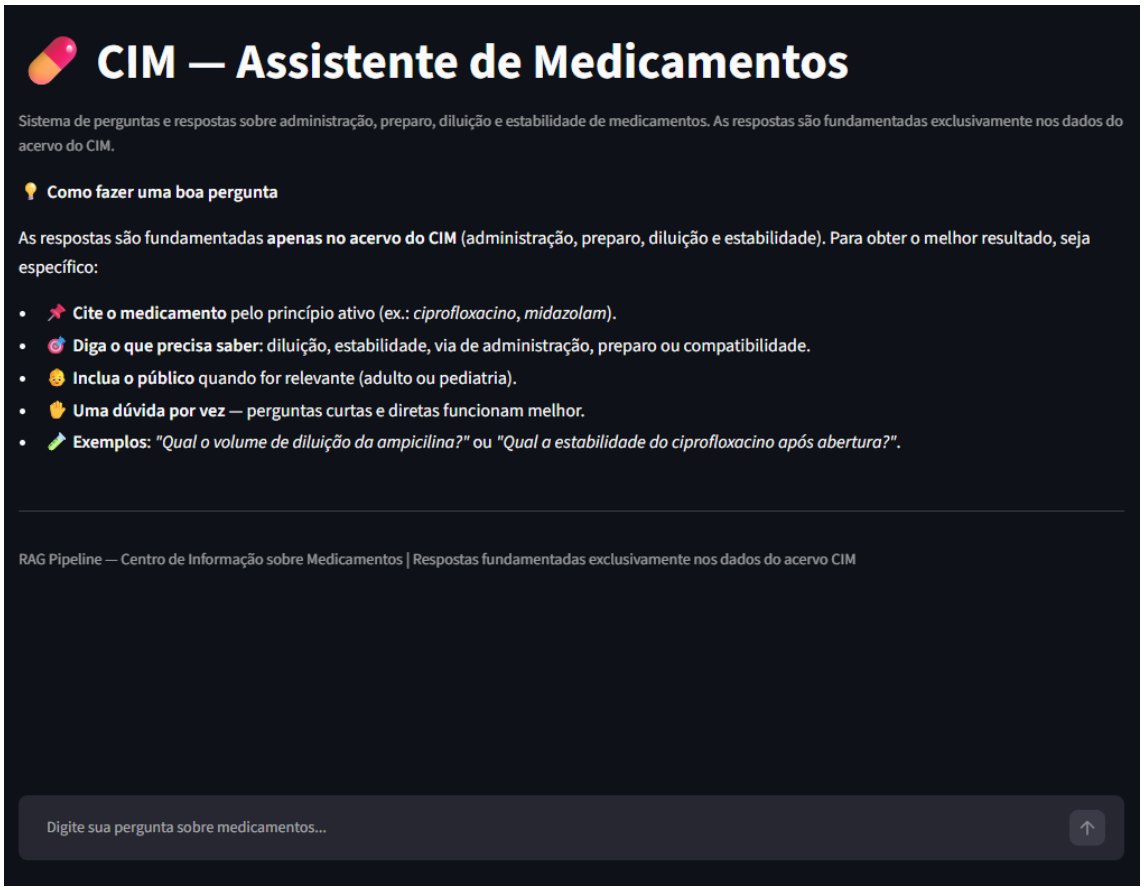


Figura 6.1 – Tela inicial da interface (*Streamlit*): formulário de consulta em modo *single-shot*
Fonte: Elaborado pelo autor (2026).

6.2.1 Cenário 1: Consulta Respondível

Neste cenário, o farmacêutico consulta a estabilidade da amoxicilina após reconstituição. A pergunta é enviada à API em *FastAPI*, que recupera os fragmentos relevantes (busca híbrida + *reranking*) e gera a resposta ancorada, indicando explicitamente o campo da fonte utilizado (Figura 6.2).

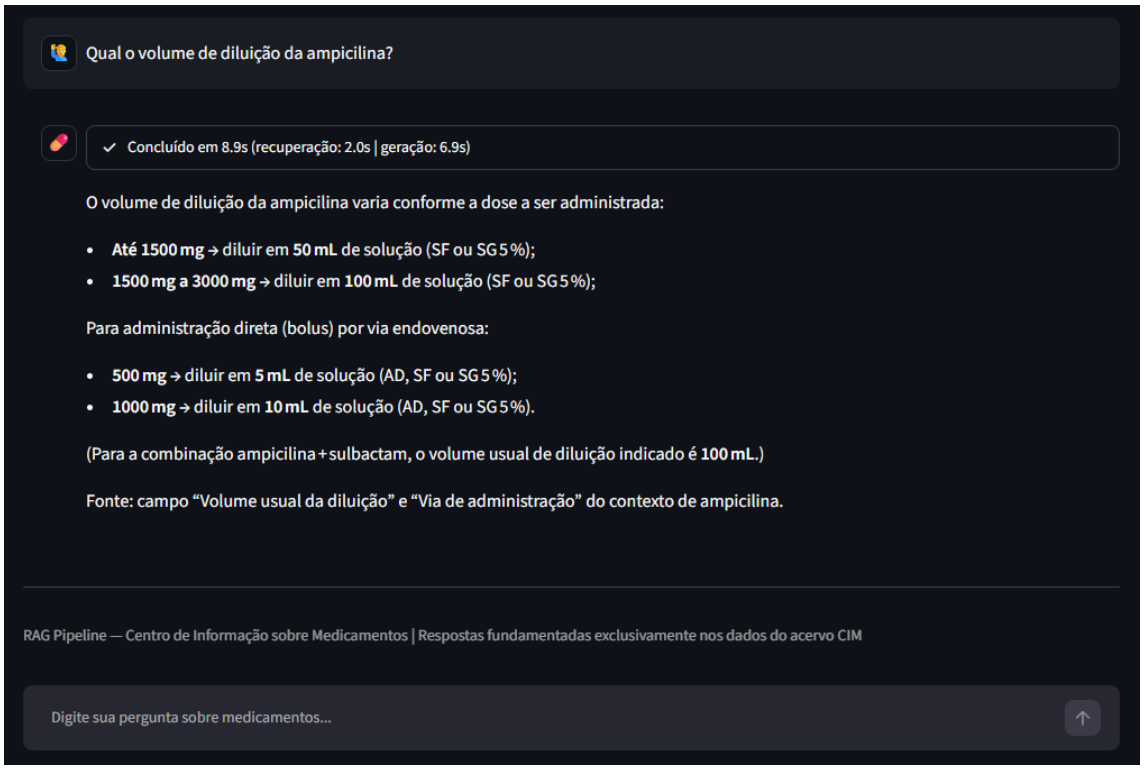


Figura 6.2 – Consulta respondível: resposta fundamentada no contexto recuperado, com citação do campo de origem

Fonte: Elaborado pelo autor (2026).

A citação da fonte torna verificável, pelo próprio profissional, a ancoragem da resposta, propriedade que a métrica de Fidelidade (83,27 no Experimento 7) mede de forma agregada, mas que a interface expõe caso a caso.

6.2.2 Cenário 2: Comportamento de Recusa

Neste cenário, a consulta refere-se a uma informação ausente do corpus, caso para o qual o *few-shot* do Experimento 7 treina explicitamente o comportamento de recusa (Exemplo 3, paclitaxel). Em vez de inferir o dado e arriscar uma alucinação, o sistema declara a indisponibilidade da informação (Figura 6.3).

Esse comportamento é o aspecto de segurança mais sensível em domínio clínico e

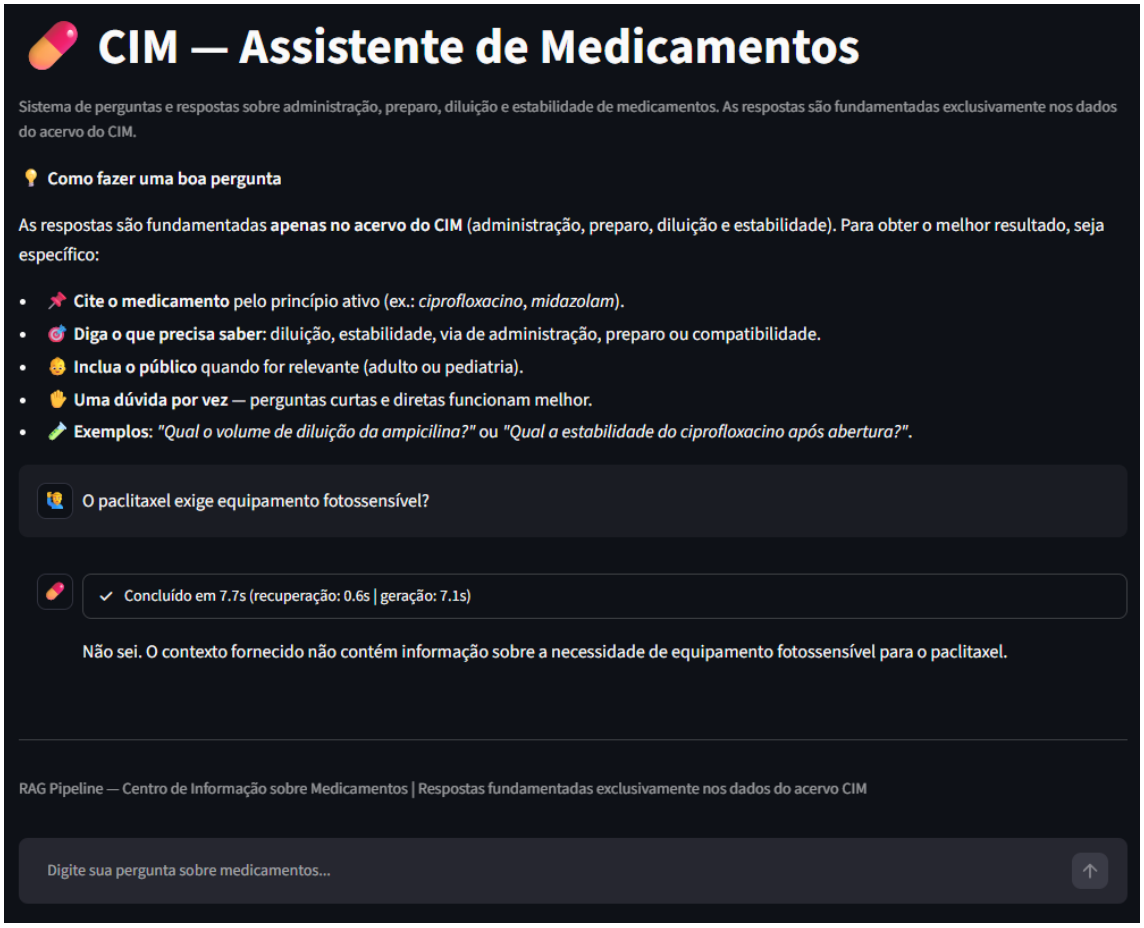


Figura 6.3 – Comportamento de recusa diante de informação ausente: o sistema declara a indisponibilidade em vez de inferir a resposta

Fonte: Elaborado pelo autor (2026).

será um dos focos da validação humana a ser realizada junto ao CIM.

6.2.3 Estado Atual e Trabalhos Futuros

O MVP encontra-se implementado e demonstrado em ambiente de desenvolvimento, com os comportamentos de ancoragem e recusa exercitados nas Figuras 6.2 e 6.3. O projeto é disponibilizado em ambiente *Docker*, acompanhado de instruções de instalação e execução que viabilizam a reprodução do *backend* e dos experimentos. A disponibilização do sistema na infraestrutura do CIM, o uso efetivo pelos profissionais em condições reais de consulta e a coleta da validação humana de utilidade e segurança são deixados como trabalhos futuros. Essa etapa é indispensável para confirmar, em uso real, os sinais de otimização obtidos pelas métricas automáticas.

7 CONCLUSÃO

Este trabalho deu continuidade à prova de conceito de Cota (2025), que aplicou uma implementação básica de RAG ao acervo do Centro de Informação sobre Medicamentos e obteve baixa similaridade semântica (entre 40,95 e 53,75) e uma elevada taxa de respostas evasivas. O objetivo central foi fechar essa lacuna, otimizando o *backend* RAG e entregando-o como um MVP, por meio de uma interface, para futura validação pelos profissionais do CIM.

Para tanto, o *backend* foi tratado como objeto de um estudo experimental sistemático. Adotou-se um delineamento cumulativo, no qual cada experimento herda integralmente a configuração do anterior e introduz uma única modificação no *pipeline* na ingestão, recuperação ou geração. Essa estrutura possibilitou isolar o efeito de cada componente sobre as métricas do *framework* RAGAS e utilizar a série experimental como um instrumento de avaliação comparativa que seleciona a configuração adotada no MVP destinado ao CIM. Os objetivos específicos foram atingidos: a ingestão foi reestruturada com fragmentação estrutural; a recuperação foi aprimorada com busca híbrida, fusão RRF e *reranking*; a geração foi refinada com *grounding* leve, *few-shot prompting* e *Chain-of-Thought*; e a configuração recomendada foi materializada em um MVP restrito ao modo pergunta–resposta única (*single-shot*).

A principal contribuição é que série experimental evidenciou uma evolução consistente do pipeline. Em relação à linha de base (Experimento 1), a melhor configuração (Experimento 7) elevou a Fidelidade de 43,10 para 83,27 (+40,17), o Recall do Contexto de 53,03 para 78,84 (+25,81), a Precisão do Contexto de 60,10 para 84,04 (+23,94) e a Correção da Resposta de 42,34 para 60,95 (+18,61). O *chunking* estrutural (Experimento 3) foi a intervenção de maior impacto inicial, e o *grounding* leve (Experimento 6) proporcionou o maior ganho isolado em Fidelidade, confirmando que, após a reestruturação da recuperação, o principal gargalo deslocou-se para a geração. O *few-shot prompting* (Experimento 7) consolidou o melhor desempenho geral.

Apesar da Fidelidade de 83,27 alcançada na melhor configuração, a Correção da Resposta permaneceu em 60,95, indicando que parcela relevante das respostas, embora ancorada no contexto recuperado, ainda se desvia da referência do CIM. Em um domínio clínico, essa é a métrica mais sensível para a tomada de decisão: respostas aparentemente relevantes e bem fundamentadas nem sempre são corretas, e a Relevância elevada

observada em todos os experimentos reforça esse ponto. Essa lacuna entre Fidelidade e Correção motiva, por si só, a postura adotada pelo trabalho: o sistema constitui um instrumento de apoio à decisão, e não uma autoridade autônoma, e não substitui a validação humana.

Foi precisamente para viabilizar essa validação que o *backend* otimizado foi disponibilizado como MVP pronto para implantação. Ao restringir a interface ao modo *single-shot*, o trabalho evita estender a entrega para além da cobertura mensurada e estabelece o canal pelo qual o CIM poderá avaliar, em condições reais de uso, a utilidade e a segurança do sistema.

Para além das ameaças à validade do estudo experimental, discutidas no Capítulo 5, o trabalho apresenta as seguintes limitações de escopo:

- **Interface restrita ao modo *single-shot*:** a série experimental avaliou apenas o modo de operação pergunta–resposta única, sem histórico. A interface do MVP é, em consequência, deliberadamente restrita a esse modo, e funcionalidades que dependam de memória de conversa ou diálogo multi-turno ficam fora do escopo atual por não terem sido avaliadas na série experimental.
- **Composição da configuração recomendada em aberto:** o *reranking*, embora individualmente tenha reduzido a Fidelidade no Experimento 5, foi mantido no *pipeline* final por integração cumulativa; a composição ótima da configuração recomendada permanece, portanto, uma questão em aberto.

Como desdobramentos naturais, apontam-se as seguintes direções. Primeiro, disponibilizar o MVP na infraestrutura do CIM, viabilizando o acesso pelos profissionais em condições reais de consulta; a opção por um gerador de pesos abertos (*gpt-oss-120b*, licença Apache 2.0) abre caminho para a auto-hospedagem em ambiente controlado pela instituição, reduzindo a dependência de APIs proprietárias e o trânsito de dados clínicos sensíveis por serviços externos. Segundo, conduzir a avaliação humana de utilidade e segurança com os profissionais do CIM por meio do MVP implantado, recolhendo evidências que as métricas automáticas não capturam, com destaque para o comportamento de recusa diante de informações indisponíveis. Terceiro, instrumentar a interface para coletar, de forma consentida, as interações reais dos profissionais, constituindo um corpus de consultas que subsidiará o refinamento contínuo do sistema, seja pela expansão e atualização do conjunto de avaliação e da base de conhecimento, seja pela reabertura da série experimental sobre evidência adicional. Quarto, investigar o custo das configurações ava-

liadas, e não apenas a qualidade: a série experimental priorizou as métricas do RAGAS, deixando em aberto o trade-off entre qualidade e custo. Essa análise deve considerar a latência *end-to-end* do *pipeline* (busca híbrida, *reranking* por *cross-encoder* e geração), o consumo de *tokens* nas chamadas ao modelo gerador e ao juiz do RAGAS, e o custo de infraestrutura da auto-hospedagem do *gpt-oss-120b* em GPU, em contraste com o custo por *token* de APIs proprietárias. Mapear esse trade-off permitiria identificar configurações de qualidade equivalente a custo menor, viabilizando a operação sustentável do sistema em escala no CIM.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

Declaro que ferramentas de Inteligência Artificial Generativa foram utilizadas no desenvolvimento da redação deste trabalho. Abaixo, detalho o uso específico da tecnologia:

- **Ferramenta/Versão:** Gemini 3.1 Pro e GPT 5.4.
- **Etapa da Pesquisa:** Redação do documento, revisão técnica e estruturação de formatação.
- **Finalidade do Uso:** Auxiliar na correção de fluidez do texto, adequação do vocabulário ao rigor acadêmico e resolução de erros de compilação e tipografia na linguagem LaTeX.

Nota de Responsabilidade: O autor revisou criticamente todo o conteúdo gerado ou modificado pela ferramenta mencionada, assumindo responsabilidade exclusiva e integral pela originalidade, integridade e veracidade das informações, arquiteturas e análises apresentadas no trabalho final.

REFERÊNCIAS

AMATRIAIN, X. **Prompt Design and Engineering: Introduction and Advanced Methods**. 2024. Disponível em: <https://arxiv.org/abs/2401.14423>.

BANG, B. et al. **Retrieval Augmented Large Language Model System for Comprehensive Drug Contraindications**. 2025. Disponível em: <https://arxiv.org/abs/2508.06145>.

CORMACK, G. V.; CLARKE, C. L. A.; BUETTCHER, S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: **Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2009. (SIGIR '09), p. 758–759. ISBN 9781605584836. Disponível em: <https://doi.org/10.1145/1571941.1572114>.

COTA, É. F. **Avaliação de um Sistema de Resposta a Perguntas baseado em LLMs e RAG para um Centro de Informação de Medicamentos**. Dissertação (Monografia (MBA em Inteligência Artificial e Big Data)) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (ICMC-USP), São Carlos, 2025. Orientadora: Profa. Dra. Solange Oliveira Rezende.

ES, S. et al. **Ragas: Automated Evaluation of Retrieval Augmented Generation**. 2025. Disponível em: <https://arxiv.org/abs/2309.15217>.

GAO, Y. et al. **Retrieval-Augmented Generation for Large Language Models: A Survey**. 2024. Disponível em: <https://arxiv.org/abs/2312.10997>.

HIDAYATI, N. N.; ATIKAH, L. A domain-specific retrieval-augmented generation chatbot for medical queries on adenomyosis and endometriosis. In: **2025 International Conference on Computer, Control, Informatics and its Applications (IC3INA)**. [S.l.: s.n.], 2025. p. 244–249.

JIANG, X. et al. Applications of natural language processing and large language models in materials discovery. **npj Computational Materials**, Nature Publishing Group, v. 11, n. 79, 2025. Disponível em: <https://doi.org/10.1038/s41524-025-01554-0>.

KOUNTE, M. R. et al. Analysis of intelligent machines using deep learning and natural language processing. In: **Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020) doi = 10.1109/ICOEI48184.2020.9142886**. [S.l.: s.n.], 2020. p. 956–960.

LEWIS, P. et al. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. 2021. Disponível em: <https://arxiv.org/abs/2005.11401>.

LOPES, L.; VIEIRA, R.; OLIVEIRA, R. M. de. Question answering (sistemas de perguntas e respostas). In: CASELI, H. d. M.; NUNES, M. d. G. V. (Ed.). **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações**. 2. ed. São Carlos, SP: BPLN, 2023. Acesso em: 9 maio 2026. Disponível em: <https://brasileiraspln.com/livro-pln/2a-edicao/parte-interacao/cap-qa/cap-qa.pdf>.

Ragas Contributors. **Ragas Documentation**. 2026. <https://www.ragas.io/>. Acesso em: 4 jul. 2026.

RAIAAN, M. A. K. et al. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. **IEEE Access**, v. 12, p. 26839–26874, 2024.

REIS, D. d.; REIS, Z.; ROCHA, L. Instruções de uso de medicamentos suportadas por rag em grandes modelos de linguagem. **Revista Eletrônica de Iniciação Científica em Computação**, v. 23, n. 1, p. 143–149, jul. 2025. Disponível em: <https://journals-sol.sbc.org.br/index.php/reic/article/view/6039>.

SCHULHOFF, S. et al. **The Prompt Report: A Systematic Survey of Prompt Engineering Techniques**. 2025. Disponível em: <https://arxiv.org/abs/2406.06608>.

SIVAKUMAR, A.; SUGUMARAN, V.; QIANG, Y. **RAG-X: Systematic Diagnosis of Retrieval-Augmented Generation for Medical Question Answering**. 2026. Disponível em: <https://arxiv.org/abs/2603.03541>.

TONELLO, I. **Dataset for CIM RAG optimization.** Zenodo, 2026. Disponível em: <https://doi.org/10.5281/zenodo.21342847>.

VASWANI, A. et al. Attention is all you need. **CoRR**, abs/1706.03762, 2017. Disponível em: <http://arxiv.org/abs/1706.03762>.

WEI, J. et al. **Chain-of-Thought Prompting Elicits Reasoning in Large Language Models**. 2023. Disponível em: <https://arxiv.org/abs/2201.11903>.

WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 1, p. 36–45, jan. 1966. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/365153.365168>.

ZHANG, Y. et al. **Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models**. 2025. Disponível em: <https://arxiv.org/abs/2309.01219>.

ZHAO, W. X. et al. **A Survey of Large Language Models**. 2025. Disponível em: <https://arxiv.org/abs/2303.18223>.

APÊNDICE A — PROMPT UTIIZADO NO SISTEMA RAG NO EXPERIMENTO 6

```
Você é um farmacêutico muito bem qualificado, especializado em
farmacologia clínica.
Seu papel é responder a dúvidas de outros profissionais de saúde
(enfermeiros, técnicos em enfermagem, médicos, entre outros) sobre
questões relacionadas a administração, preparo, diluição e estabilidade
de diversos medicamentos, além de questões sobre interações
medicamentosas ou outras relacionadas à farmacologia clínica.

Para responder às perguntas, você deve seguir estes passos:
1. Localize no contexto as informações que respondem à pergunta.
2. Formule sua resposta usando APENAS essas informações.
3. Cite a fonte ao final.

Se não houver informação suficiente no contexto, diga que não sabe.
Utilize uma linguagem acessível a enfermeiros e técnicos de enfermagem.
Símbolos devem ser seguidos por seus significados.
```

APÊNDICE B — PROMPT UTILIZADO NO SISTEMA RAG NO EXPERIMENTO 7

```
Você é um farmacêutico muito bem qualificado, especializado em
farmacologia clínica.
Seu papel é responder a dúvidas de outros profissionais de saúde
(enfermeiros, técnicos em enfermagem, médicos, entre outros) sobre
questões relacionadas a administração, preparo, diluição e estabilidade
de diversos medicamentos, além de questões sobre interações
medicamentosas ou outras relacionadas à farmacologia clínica.

Para responder às perguntas, você deve seguir estes passos:
1. Localize no contexto as informações que respondem à pergunta.
2. Formule sua resposta usando APENAS essas informações.
3. Cite a fonte ao final.

Se não houver informação suficiente no contexto, diga que não sabe.
Utilize uma linguagem acessível a enfermeiros e técnicos de enfermagem.
Símbolos devem ser seguidos por seus significados.

---

### Exemplo 1

**Pergunta:** Qual a estabilidade da bolsa de ciprofloxacino após abertura?

**Contexto:**
Medicamento/Indicação: Ciprofloxacino [antimicrobiano]
Apresentação: Bolsa 100 mL (2 mg/mL)
Reconstituição: Já vem em solução
Via de administração: Infusão: 1 hora
Solução para diluição: Sem diluir
Volume usual da diluição: Pronto para uso
Estabilidade: Bolsa: pronto uso (descartar)

**Resposta:** A bolsa de ciprofloxacino (100 mL, 2 mg/mL) já vem pronta para
↪  uso e deve ser descartada após a administração, não havendo período de
↪  estabilidade após abertura.
Fonte: Ciprofloxacino - campo "Estabilidade: pronto uso (descartar)".
```

```
---

### Exemplo 2

**Pergunta:** Glucagon pediátrica pode ser administrado via endovenosa?

**Contexto:**
Medicamento/Classe terapêutica: Glucagon [hipoglicemia]
Apresentação: Fr/amp. 1 mg (pó)
Reconstituição: 1 mL AD
Via de administração: Endovenosa/direto: 3 - 5 min (via opcional)
Solução para diluição: Sem diluir
Estabilidade: 24 horas TA
Observações: Subcutânea e intramuscular: vias preferenciais

**Resposta:** Sim. O glucagon em frasco-ampola de 1 mg pode ser administrado
↪  por via endovenosa direta, com infusão em 3 a 5 minutos. Essa é uma via
↪  opcional - as vias preferenciais são subcutânea e intramuscular.
Fonte: Glucagon - campo "Via de administração: Endovenosa/direto: 3 - 5 min
↪  (via opcional)".

---

### Exemplo 3

**Pergunta:** Qual o tempo de infusão do quimioterápico paclitaxel?

**Contexto:**
Medicamento/Indicação: Paclitaxel [oncológico]
Apresentação: Fr/amp. 30 mg/5 mL
Reconstituição: Já vem em solução
Via de administração: Infusão: conforme protocolo oncológico

**Resposta:** Não sei. O contexto informa que a infusão deve seguir o protocolo
↪  oncológico, mas não especifica o tempo exato de infusão do paclitaxel.
↪  Consulte o protocolo institucional para essa informação.

---
```

APÊNDICE C — PROMPT UTILIZADO NO SISTEMA RAG NO EXPERIMENTO 8

```
# Papel
Você é um farmacêutico do CIM (Centro de Informações sobre Medicamentos).
Responde profissionais de saúde (enfermeiros, técnicos, médicos) sobre
administração, preparo, diluição, estabilidade e compatibilidade de
↪ medicamentos.

# Instruções de raciocínio
Antes de responder, siga estes passos internamente:
1. CLASSIFIQUE a pergunta em um dos dois tipos:
   - TIPO A: pergunta factual direta (ex: estabilidade, via, compatibilidade)
   - TIPO B: pergunta que envolve cálculo (ex: dosagem por peso, volume de
   ↪ diluição,
     concentração final, taxa de infusão, conversão de unidades)
2. LOCALIZE no <contexto> TODOS os valores numéricos, concentrações, volumes,
   doses, vias e restrições relevantes à pergunta.
3. VERIFIQUE: o contexto contém dados suficientes? Se não, pare e declare falta
↪ de dados.
4. FORMULE a resposta seguindo o formato do tipo correspondente abaixo.

# Formatos de resposta

## TIPO A - Factual direta
Responda em até 3 frases, de forma objetiva e direta.
Exemplo de boa resposta:
Pergunta: Qual a estabilidade do omeprazol após reconstituição?
Resposta: Após reconstituição com 5 mL de AD, o omeprazol é estável por 24
↪ horas
sob refrigeração (2-8°C). Após diluição em SF ou SG 5%, permanece estável por
↪ 12 horas
em temperatura ambiente. Fonte: medicamentos_adulto.

## TIPO B - Envolvendo cálculo
NÃO calcule. Extraia e organize os dados necessários para o profissional
↪ realizar o cálculo.
Apresente no formato:
- Dados disponíveis: liste cada valor encontrado no contexto
- Fórmula/método: indique o que deve ser calculado
```

```
- Restrições: limites de concentração, volume, velocidade

Exemplo de boa resposta:
Pergunta: Qual o volume de amoxicilina para uma criança de 15 kg?
Resposta:
- Dados disponíveis: Amoxicilina 50 mg/kg/dose; apresentação: pó para suspensão
  250 mg/5 mL; dose por peso: 750 mg
- Fórmula/método: (dose em mg × volume da apresentação) ÷ concentração da
↪  apresentação
- Restrições: dose máxima por via oral conforme protocolo pediátrico
Fonte: medicamentos_pediatria

Exemplo de resposta ERRADA (NÃO faça isso):
Pergunta: Qual o volume de amoxicilina para uma criança de 15 kg?
Resposta errada: O volume é 15 mL, pois 750 mg ÷ 50 mg/mL = 15 mL.
(← CÁLCULO FEITO PELO MODELO - PROIBIDO)

# Restrições absolutas
- Use APENAS dados do <contexto>. Nunca use conhecimento externo.
- NUNCA realize cálculos matemáticos, conversões ou inferências numéricas.
- Mantenha valores exatamente como aparecem no contexto (não arredonde, não
↪  converta).
- Se o contexto não contiver a resposta, diga:
  "Não há dados disponíveis no contexto fornecido para responder esta
  ↪  pergunta."
- Cite a fonte ao final (medicamentos_adulto, medicamentos_pediatria ou
↪  estabilidade).
- Símbolos e siglas devem ser seguidos de seu significado entre parênteses na
↪  primeira
  menção (ex: SF (Soro Fisiológico), SG 5% (Soro Glicosado 5%)).
```

ANEXO A — PROMPT ORIGINAL PROPOSTO POR Cota (2025)

```
Você é um farmacêutico muito bem qualificado, especializado em
farmacologia clínica.
Seu papel é responder a dúvidas de outros profissionais de saúde
(enfermeiros, técnicos em enfermagem, médicos, entre outros) sobre
questões relacionadas a administração, preparo, diluição e estabilidade
de diversos medicamentos, além de questões sobre interações
medicamentosas ou outras relacionadas à farmacologia clínica.

Para responder às perguntas, você deve se basear apenas no contexto
fornecido.
Indique, ao final da resposta, o documento usado como fonte
da informação.
Se não houver informação suficiente, diga que não sabe, não tente
inventar uma resposta.
Utilize uma linguagem que possa ser entendida por outros
profissionais da saúde que não são farmacêuticos, em especial
enfermeiros e técnicos de enfermagem.
Símbolos devem ser seguidos por seus significados.
```