16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

UMA ANÁLISE QUALITATIVA DAS TÉCNICAS UTILIZADAS NO DESIGN DE UM PIPELINE DE RAG AVANÇADO PARA DOCUMENTOS ACADÊMICOS

ALICE DA SILVA MARINHO GOMES[1], PROFESSOR DR.NELSON NASCIMENTO JUNIOR[2]

[1] Graduando em Tecnologia de Análise e Desenvolvimento de Sistemas, Estudante PIVICT - Programa Institucional Voluntário de Iniciação Científica e/ou Tecnológica, IFSP, Campus Cubatão, alice.marinho@aluno.ifsp.edu.br
[2] Doutorado em Ciência da Computação pela Universidade Federal do ABC e Professor Titular do Instituto Federal de São Paulo, Campus Cubatão, nelsonjr@ifsp.edu.br
Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação

RESUMO: A aplicação de sistemas de Geração Aumentada por Recuperação (RAG) em domínios altamente factuais, como a análise de Projetos Pedagógicos de Curso (PPCs), demanda uma arquitetura que transcenda a recuperação e geração básicas. Este artigo apresenta um estudo qualitativo detalhando o design e a justificação de um *pipeline* de RAG avançado, focado na superação da ambiguidade em consultas de linguagem natural. Demonstramos como a seleção estratégica e o *design* de componentes, como o *self-querying retriever* para filtros específicos e a *sub-query decomposition* para perguntas multifacetadas, aliadas a estratégias de *prompting* como o *step-back*, visam mitigar falhas lógicas inerentes a sistemas mais simples. É importante notar que, embora o *design* seja robusto, a assertividade plena de certos módulos, como a classificação de rota e a efetividade do *step-back prompting*, são pontos de refinamento em andamento. Ao detalhar o processo decisório, este trabalho valida a engenharia precisa de componentes como fundamental para alcançar melhorias substanciais na precisão factual e na confiabilidade de sistemas RAG no contexto acadêmico.

PALAVRAS-CHAVE: Geração Aumentada por Recuperação; Pipeline RAG; Design de Componentes; *self-querying retriever; sub-query decomposition*; *step-back prompting.*

THE ARCHITECTURAL JUSTIFICATION: A QUALITATIVE ANALYSIS OF DESIGN DECISIONS IN AN ADVANCED RAG PIPELINE FOR ACADEMIC DOCUMENTS

ABSTRACT: The application of Retrieval-Augmented Generation (RAG) systems in highly factual domains, such as the analysis of Pedagogical Course Projects (PPPs), demands an architecture that transcends basic retrieval and generation. This paper presents a qualitative study detailing the design and justification of an advanced RAG pipeline focused on overcoming ambiguity in natural language queries. We demonstrate how the strategic selection and design of components, such as the self-querying retriever for specific filters and the sub-query decomposition for multifaceted questions, combined with prompting strategies such as step-back, aim to mitigate logical flaws inherent in simpler systems. It is important to note that, although the design is robust, the full assertiveness of certain modules, such as route classification and the effectiveness of step-back prompting, are points of ongoing refinement. By detailing the decision-making process, this work validates precise component engineering as fundamental to achieving substantial improvements in the factual accuracy and reliability of RAG systems in the academic context.

KEYWORDS: Retrieval-Augmented Generation; RAG Pipeline; Component Design; self-querying retriever; sub-query decomposition; step-back prompting

INTRODUÇÃO

A Geração Aumentada por Recuperação (RAG) tem se mostrado promissora para capacitar modelos de linguagem na geração de respostas precisas em domínios de alta fidelidade factual (Lewis et al., 2020). Contudo, em cenários complexos como a análise de Projetos Pedagógicos de Curso (PPCs), a ambiguidade das consultas em linguagem natural e a necessidade de precisão factual representam desafios significativos. É crucial que a intenção do usuário seja corretamente interpretada para garantir a confiabilidade desses sistemas, um recurso vital para docentes e gestores.

Nesse contexto, o *design* de sistemas RAG enfrenta um desafio central: como orquestrar as diversas técnicas disponíveis (ex: *self-querying*, *sub-query decomposition*) para superar as falhas lógicas de abordagens mais simples e garantir a precisão. Este artigo visa preencher essa lacuna, apresentando o processo de raciocínio e a justificativa para cada escolha de *design* de um pipeline RAG avançado (Izacard et al., 2023; Ram et al., 2023). Nossa hipótese é que a engenharia precisa e a justaposição estratégica de módulos, visando compreender e rotear a intenção da consulta, são fundamentais para alcançar melhorias na precisão e confiabilidade. Através de um estudo qualitativo, em andamento, demonstraremos o *design* e o refinamento de uma arquitetura adaptativa, validando um *framework* de raciocínio para construir sistemas RAG mais robustos no domínio acadêmico.

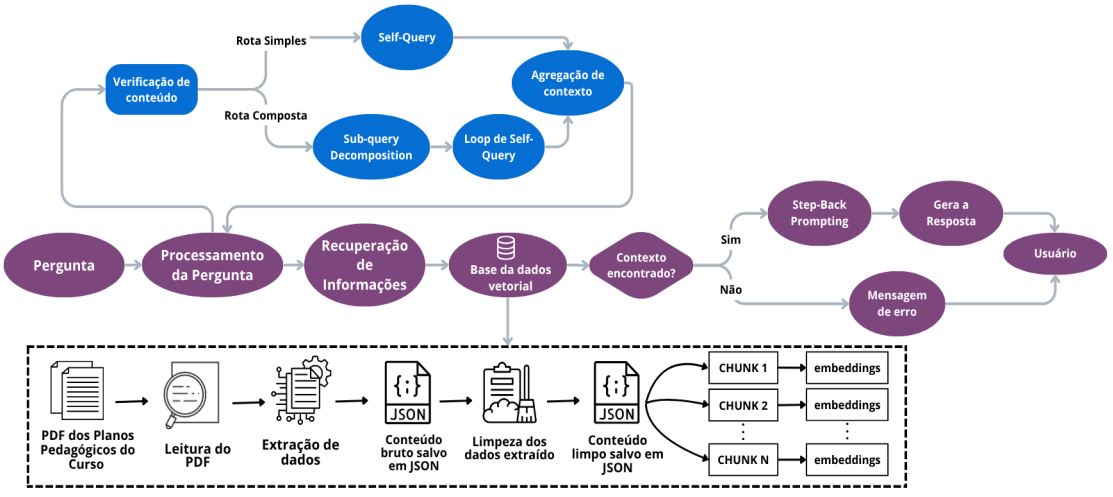
MATERIAL E MÉTODOS

Este estudo envolveu as seguintes etapas:

1. **Métodos de Processamento e Arquitetura do Sistema**

   O processo de construção do sistema RAG (Lewis et al., 2020), envolveu etapas de pré-processamento de documentos, indexação e o projeto de uma arquitetura adaptativa para orquestração de consultas, conforme ilustrado na Figura 1. O RAG não só expande o conhecimento do modelo, que antes se restringia ao treinamento, mas também mitiga alucinações, garantindo maior precisão e utilidade em domínios sensíveis, como o acadêmico (Sharma, 2025).

FIGURA 1. Fluxograma da Arquitetura do Sistema RAG



2. **Pré-processamento e Indexação de Documentos**
   Os PPCs em PDF foram submetidos a um processo de leitura unificada e seu conteúdo textual bruto foi extraído e armazenado em arquivos JSON. Uma etapa de limpeza foi aplicada para remover caracteres indesejados e padronizar o formato do conteúdo. Posteriormente, o texto limpo foi fragmentado em *chunks*, com cada matéria do curso constituindo um fragmento

distinto para preservar sua coerência semântica. Cada fragmento foi convertido em um *embedding* vetorial, e esses vetores foram armazenados no banco de dados vetorial *ChromaDB* (https://www.trychroma.com/), facilitando a recuperação de informações baseada em similaridade semântica.

3. **Arquitetura Adaptativa e Orquestração de Consultas**
   A arquitetura do sistema foi projetada para otimizar a extração de conteúdo relevante a partir das consultas dos usuários, baseando-se em um mecanismo de roteamento inteligente (conforme detalhado na Figura 1). Ao receber uma consulta em linguagem natural, o sistema propõe-se a executar uma verificação inicial da intenção da pergunta, visando direcioná-la para um dos dois caminhos de processamento.

4. **Rota Simples (self-querying retriever)**
   O *self-querying retriever*, uma funcionalidade do framework *LangChain* (LangChain Docs, 2025) , constitui um método que permite a conversão de consultas expressas em linguagem natural pelo usuário em consultas estruturadas. Em nossa pesquisa, este caminho foi planejado para consultas que buscam conteúdo único ou informações específicas de uma única matéria (ex: "carga horária de Metodologia Científica"). A proposta é que o *self-querying retriever* seja acionado para traduzir a consulta em uma estrutura que permita a busca direta por metadados associados aos *chunks* das matérias. O objetivo é que este método proporcione uma pesquisa mais assertiva nos fragmentos, estabelecendo a base para uma resposta precisa gerada pelo LLM.

5. **Rota Composta (sub-query decomposition)**
   A decomposição em *sub-queries* permite que perguntas complexas sejam fragmentadas em unidades menores e mais gerenciáveis (Six et al., 2025). Em nosso estudo podemos citar alguns exemplos de consultas complexas: "Quais as diferenças entre a grade curricular de Automação Industrial e Análise e Desenvolvimento de Sistemas, e os pré-requisitos das disciplinas básicas de ambos?". A abordagem prevê que o *sub-query decomposition* divida a consulta original em sub-perguntas menores e mais gerenciáveis. Cada sub-pergunta seria processada individualmente, incluindo a busca por metadados via *self-query*, para agregar o contexto necessário.

6. **Refinamento da Consulta (step-back prompting)**
   A arquitetura projetada prevê que, após a recuperação do contexto relevante, a técnica de *step-back prompting* seja aplicada para refinar a consulta e o contexto, fornecendo-os ao Llama 3 (https://www.llama.com/models/llama-3/) para a geração da resposta final. Caso nenhum contexto pertinente seja encontrado, o sistema será configurado para enviar uma mensagem de erro ao usuário. O *step-back prompting* é uma técnica avançada de *prompting* que instrui o LLM a abstrair ou gerar princípios gerais antes de solucionar uma requisição específica (Zheng et al., 2023).

O desenvolvimento do sistema envolveu múltiplas iterações de refinamento do design conceitual e da implementação inicial. Inicialmente, utilizamos uma abordagem baseada em manipulação de dados com Pandas. Posteriormente, optamos por integrar a solução, técnicas avançadas de RAG, como o *self-querying retriever.* Esta transição foi crucial para melhorar a busca das matérias, visto que o Pandas exige uma implementação de filtros mais explícita e verbosa, enquanto o *self-querying retriever* soluciona essa complexidade ao traduzir a linguagem natural do usuário diretamente em filtros de metadados, visando uma seleção de *chunks* mais assertiva. Essa otimização na construção da consulta (*query construction*) é fundamental para iniciar o processo de uma boa resposta da LLM.

Materiais Utilizados

O corpus textual utilizado consistiu em Projetos Pedagógicos de Curso (PPCs) de uma instituição de ensino, abrangendo os cursos de Análise e Desenvolvimento de Sistemas, Automação Industrial, Ensino Médio Integrado ao Técnico e Turismo. Todos os documentos estavam em formato PDF, com previsão de expansão do corpus para incluir todos os cursos da instituição em fases futuras do projeto. Para o processamento da linguagem natural e a geração de respostas, o Llama 3 foi selecionado como o Modelo de Linguagem de Grande Escala (LLM), após avaliações preliminares que indicaram seu custo-benefício favorável e além do desempenho robusto em tarefas de compreensão de linguagem acadêmica (Zheng et al., 2024). Para as etapas de pré-processamento, extração e fragmentação dos documentos, técnicas consagradas como *chunking* semântico e limpeza textual foram utilizadas. A criação dos *embeddings* foi realizada com modelo *Hugging Face* (https://huggingface.co/) para indexação vetorial eficiente em bancos como *ChromaDB*. Todo o projeto e os scripts foram implementados em *Python*.

RESULTADOS E DISCUSSÃO

Esta seção apresenta os achados qualitativos derivados do nosso estudo. Os resultados são confrontados com trabalhos relevantes na literatura sobre o tema.

1. Sucesso da Rota Simples e da Geração com Step-Back Prompting

A análise de consultas demonstrou alta assertividade na Rota Simples, traduzindo linguagem natural em buscas precisas para intenções e filtros claros (como ‘disciplina’ e ‘curso’). Como podemos observar, os casos C1 e C2 da Tabela 1, o sistema processa corretamente as perguntas sobre "Geografia 1" e "Física Teórica 1" validando a recuperação de contexto relevante via *Self-Querying Retriever*, crucial para a geração de respostas. A técnica de *Step-Back Prompting* estruturou as respostas, identificando princípios pedagógicos antes de detalhar, primeiro executando a "Etapa 1" para identificar o princípio pedagógico fundamental da disciplina, e depois utiliza essa ideia para construir a "Etapa 2", uma resposta detalhada e coerente. Validando o uso de *prompting* avançado para qualidade e organização das respostas, elevando a capacidade de raciocínio das LLMs.

TABELA 1. Análise de Casos de Sucesso da Rota Simples.

ID	Consulta do Usuário	Rota	Filtro Gerado	Resposta da IA	Qualidade da Resposta
C1	O que se espera que os alunos saibam ao final de 'Geografia 1'?	Busca Simples	componente_curricular = Geografia 1	Resposta completa e detalhada, com Etapa 1 e Etapa 2.	Sucesso completo. O Step-Back e o Self-Querying funcionaram em conjunto.
C2	Quais são os objetivos de aprendizagem de 'Física Teórica 1'?	Busca Simples	componente_curricular = Física Teórica 1	Resposta completa e detalhada, com Etapa 1 e Etapa 2.	Sucesso completo. O roteamento e a geração foram de alta qualidade.

2. Sucesso da Rota Composta e a Eficácia da Decomposição de Consultas

Em contraste com as consultas simples, a rota composta, baseada na Decomposição de Consultas, demonstrou resultados promissores ao lidar com a complexidade das perguntas que exigem a integração de conhecimentos de diferentes disciplinas. A estratégia de quebrar questões multifacetadas em sub-perguntas (mini-questões) e processá-las individualmente resultou na recuperação de um contexto significativamente mais abrangente e relevante. Esta abordagem permitiu ao sistema acessar e integrar informações de diferentes seções dos PPCs que, em abordagens mais simples ou sem decomposição, seriam negligenciadas.

A Tabela 2 apresenta exemplos que comprovam a eficácia dessa rota. Podemos notar na consulta como C3, o sistema não apenas identificou a complexidade da pergunta, mas também a dividiu em sub-perguntas lógicas. Cada sub-questão foi processada independentemente, e a agregação

dos contextos recuperados, seguida pela geração de resposta pelo Llama 3 (com *step-back prompting*), resultou em propostas de projeto e atividades notavelmente mais completas e relevantes.

TABELA 2. Análise de Casos de Sucesso da Rota Composta.

ID	Consulta do Usuário	Sub-Perguntas Geradas	Qualidade da Resposta
C3	Elabore um projeto que una 'Programação de Computadores' (Automação) e 'Desenvolvimento Web' (ADS).	1- “informações sobre 'Programação de Computadores'” 2- “informações sobre 'Desenvolvimento Web'”	Resposta abrangente

3. Limitações e Análise de Falhas do Sistema

A análise qualitativa revelou as principais limitações do sistema, categorizadas em duas falhas:

1. **Falha do Módulo de Recuperação (*Retriever*):** Em diversos casos, a falha não ocorreu na construção da consulta, mas na ausência de contexto relevante no corpus. O sistema, de forma robusta e transparente, evitou alucinações ao comunicar a falha com a mensagem "Desculpe, não encontrei informações suficientes...", validando a decisão de *design* de priorizar a confiabilidade, no entanto, ainda estamos investigando algumas limitações identificadas neste módulo.
2. **Falha de Classificação do Roteador:** O módulo de roteamento de consultas demonstrou inconsistências, classificando erroneamente algumas consultas complexas como Busca Simples. Por exemplo, a pergunta sobre um "projeto final para o curso de Turismo" foi roteada incorretamente. Em outros cenários de roteamento incorreto, o LLM e o *step-back prompting* conseguiram mitigar a falha e entregar uma resposta de alta qualidade, como no caso do projeto de "História e Arte", sugerindo que o poder do LLM pode, em alguns casos, compensar as falhas de orquestração.

Além das discussões supracitadas, nossa pesquisa validou a relevância do design de uma arquitetura RAG adaptativa que compreende a intenção da consulta antes da recuperação de contexto. A proposta de roteamento para o *self-querying retriever* ou para a *sub-query decomposition* representa um avanço em relação a abordagens mais genéricas (Six et al., 2025). Contudo, os desafios observados na assertividade da classificação da intenção e nas limitações do *retriever* indicam áreas críticas para pesquisa e desenvolvimento futuros.

CONCLUSÃO E OPORTUNIDADES DE PESQUISA FUTURA

Nosso estudo validou a relevância do *design* de uma arquitetura RAG adaptativa que compreende a intenção da consulta antes da recuperação de contexto. A proposta de roteamento para o *self-querying retriever* e para a *sub-query decomposition* representou um avanço significativo em relação a abordagens mais genéricas, demonstrando melhorias na precisão factual e na capacidade de lidar com consultas complexas no domínio de documentos acadêmicos.

Apesar do sucesso em grande parte dos cenários, a análise do nosso *framework* de raciocínio revelou desafios importantes. A classificação da intenção do usuário, em particular, demonstrou inconsistências, falhando em distinguir corretamente algumas consultas e, consequentemente, afetando a qualidade de certas respostas. Além disso, a capacidade de recuperação do módulo *retriever* mostrou-se limitada pela ausência de contexto no corpus para algumas perguntas específicas.

Estes achados apontam para as próximas etapas do nosso trabalho, que focarão no refinamento do módulo de recuperação para que ele trabalhe de forma mais sinérgica com a estrutura de roteamento estabelecida. Futuras pesquisas também buscarão refinar a lógica de classificação de consultas, visando aprimorar a assertividade do sistema e, assim, consolidar a aplicação do RAG como um recurso confiável e indispensável para a análise de PPCs. Além disso, pretendemos complementar nossas análises qualitativas com dados estatísticos sobre os resultados obtidos.

REFERÊNCIAS

IZACARD, Gautier et al. **Atlas: Few-shot Learning with Retrieval Augmented Language Models. Journal of Machine Learning Research**, v. 24, p. 1–43, 2023. Disponível em: https://jmlr.org/papers/v24/23-0037.html . Acesso em: 1 ago. 2025.

LANGCHAIN. **LangChain Docs.** [s.l.] Disponível em: https://python.langchain.com/api_reference/langchain/retrievers/langchain.retrievers.self_query.base.SelfQueryRetriever.html. Acesso em: 26 jul. 2025.

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; KÜTTLER, H.; LEWIS, M.; YIH, W.-T.; ROCKTÄSCHEL, T.; RIEDEL, S.; KIELA, D. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, vol. 33, p. 9459–9474, [s. l.], 2020.

RAM, Ori et al. **In-Context Retrieval-Augmented Language Models**. arXiv, [s.l.], Jan. 2023. Disponível em: https://arxiv.org/abs/2302.00083v3. Acesso em: 1 ago. 2025.

SHARMA, Chaitanya. **Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers**. arXiv, [s.l], 28 mai. 2025. Disponível em: https://arxiv.org/abs/2506.00054v1. Acesso em 31 jul. 2025.

SIX, Valentin; DUFRAISSE, Evan; DE CHALENDAR, Gaël. **Decompositional Reasoning for Graph Retrieval with Large Language Models**. arXiv preprint, 2506.13380v1, 16 Jun. 2025. Disponível em: https://arxiv.org/abs/2506.13380v1. Acesso em: 31 jul. 2025

ZHENG, Huaixiu Steven; MISHRA, Swaroop; CHEN, Xinyun; CHENG, Heng-Tze; CHI, Ed H.; LE, Quoc V.; ZHOU, Denny. **Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models.** arXiv, [s. l.], 2023. Disponível em: https://arxiv.org/abs/2208.03299v3. Acesso em 1 ago. 2025.