UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Uma Metodologia Baseada em Agentes com Avaliação Autônoma para Extração de Conhecimento em Textos Técnicos

Matheus Gomes Diniz Andrade

Orientador: Prof. Dr. Ivanovitch Medeiros Dantas da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PPgEEC: M702
Natal, RN, dezembro de 2025

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Andrade, Matheus Gomes Diniz.
Uma metodologia baseada em agentes com avaliação autônoma para extração de conhecimento em textos técnicos / Matheus Gomes Diniz Andrade. - 2025.
83 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, RN, 2025.
Orientação: Prof. Dr. Ivanovitch Medeiros Dantas da Silva.

1. Geração Aumentada por Recuperação (RAG) - Dissertação. 2. Modelos de linguagem de grande escala - Dissertação. 3. Sistemas multiagente - Dissertação. 4. Redes industriais - Dissertação. 5. Recuperação de conhecimento - Dissertação. 6. Documentação técnica - Dissertação. I. Silva, Ivanovitch Medeiros Dantas da. II. Título.

RN/UF/BCZM CDU 004.7

Elaborado por Jackeline dos Santos Pinheiro da Silva Maia Cavalcanti - CRB-15/317

Aos meus pais, que são meu porto seguro e minha maior inspiração.

Agradecimentos

Primeiramente, agradeço a Deus pelo dom da vida e por ter me dado a oportunidade de chegar onde jamais imaginei, e a Nossa Senhora por me acompanhar em toda minha trajetória.

Aos meus pais, Maria Rozelia Gomes Andrade e Paulo Salviano Diniz Andrade, pelo exemplo de integridade, pelos valores transmitidos e pelo suporte incondicional à minha trajetória acadêmica, pilares essenciais para esta conquista.

À minha namorada, Sabrinne Sampaio, que com paciência e carinho soube me encorajar nos momentos mais difíceis. Sua presença ao meu lado tornou esta jornada muito mais significativa.

Ao meu orientador, Ivanovitch Medeiros Dantas da Silva, pelas valiosas oportunidades concedidas. Expresso minha profunda gratidão por sua generosidade em sempre me receber em sua sala e pela paciência durante meu aprendizado. Seu incentivo constante foi essencial para concretizar meus objetivos.

À Marianne Batista Diniz da Silva, cuja paciência e orientação foram fundamentais para o desenvolvimento deste trabalho. Minha profunda gratidão pelos ensinamentos que enriqueceram significativamente minha trajetória acadêmica. Ao seu esposo, Breno Santana Santos, pelos conselhos compartilhados e pelos momentos de descontração que trouxeram leveza aos meus dias durante esta jornada.

Agradeço aos professores Luiz Affonso e Carlos Viegas por todos os ensinamentos compartilhados.

Agradeço aos professores Dennis Brandão e Paolo Ferrari, pelas contribuições que tornaram este trabalho possível.

Aos meus amigos Miguel Eurípedes e Ana Luisa, por serem pilares de força e amizade. Nossas conversas ajudaram a melhorar minha visão de mundo e me mostraram como qualquer jornada se torna mais leve quando compartilhada com pessoas especiais como vocês.

Aos amigos e colegas do Conect2ai, pela colaboração e companheirismo. Em especial, ao Fellipe Milomem, pelas conversas matinais no laboratório que traziam tranquilidade ao início do dia; à Thaís Medeiros, pelas dicas valiosas, momentos de descontração e pelo bom humor que tornou a rotina muito mais leve; ao Morsinaldo Medeiros, por cada café e

troca de ideias que me ajudaram a manter a motivação nesta jornada; ao Hilton Thallyson, pelas risadas e reflexões compartilhadas; e à Gisliany Oliveira, pelos momentos de leveza e conversas que tanto contribuíram para o meu amadurecimento pessoal.

Expresso também minha gratidão a todos que, de forma direta ou indireta, colaboraram para o meu amadurecimento e estiveram presentes, de algum modo, na realização desta importante conquista.

Resumo

No cenário atual de convergência *Operational Technology* (OT), *Information Technology* (IT), e Indústria 5.0, a manutenção e resiliência de sistemas de OT dependem da continuidade do conhecimento. A dificuldade e ineficiência em acessar o conhecimento técnico disperso em documentação não estruturada de sistemas legados, como o protocolo PROFIBUS (*Process Field Bus*), impactam criticamente a manutenção e a tomada de decisão. Diante disso, este trabalho investiga o uso de arquiteturas de Inteligência Artificial Generativa para recuperar e sintetizar esse conhecimento industrial legada. O estudo compara o desempenho de três configurações distintas: (i) um LLM autônomo (*Baseline*), (ii) um modelo RAG de agente único, e (iii) uma arquitetura RAG Multiagente. Por meio de um experimento baseado em textos técnicos do PROFIBUS, o desempenho foi avaliado usando métricas quantitativas (ROUGE, BERTScore) e avaliação qualitativa (*LLM-as-a-Judge*). Os resultados demonstram que a orquestração *Multi-Agent* estabelece uma hierarquia de desempenho superior. A colaboração e especialização entre agentes aumentam a precisão contextual e a acurácia factual das respostas, ao mesmo tempo em que reduzem a alucinação em 80% de fidelidade ao contexto. O estudo valida a RAG *Multi-Agent* como um padrão arquitetônico para garantir a confiabilidade na transferência de conhecimento de engenharia, fornecendo um caminho para a automação centrada no ser humano na Indústria 5.0.

Palavras-chave: Geração Aumentada por Recuperação (RAG), Modelos de Linguagem de Grande Escala, Sistemas Multiagente, Redes Industriais, Recuperação de Conhecimento, Documentação Técnica.

Abstract

In the current scenario of Operational Technology (OT) and Information Technology (IT) convergence and Industry 5.0, the maintenance and resilience of OT systems depend on knowledge continuity. The difficulty and inefficiency in accessing the dispersed technical knowledge within unstructured documentation of legacy systems, such as the PROFIBUS (Process Field Bus) protocol, critically impact maintenance and decision-making. Therefore, this work investigates the use of Generative Artificial Intelligence architectures to retrieve and synthesize this legacy industrial knowledge. The study compares the performance of three distinct configurations: (i) a standalone LLM (Baseline), (ii) a Single-Agent RAG (Retrieval-Augmented Generation) model, and (iii) a Multi-Agent RAG architecture. Through an experiment based on PROFIBUS technical texts, performance was evaluated using quantitative metrics (ROUGE, BERTScore) and qualitative assessment (LLM-as-a-Judge). The results demonstrate that Multi-Agent orchestration establishes a superior performance hierarchy. The collaboration and specialization among agents significantly increase the contextual precision and factual accuracy of the responses, while simultaneously reducing hallucination to 80% faithfulness to the context. The study validates the Multi-Agent RAG as a necessary architectural pattern to guarantee reliability in the transfer of engineering knowledge, providing a pathway for human-centric automation in Industry 5.0.

Keywords: Retrieval-Augmented Generation (RAG), Large Language Models, Multi-Agent Systems, Industrial Networks, Knowledge Retrieval, Technical Documentation.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

API	Interface de Programação de Aplicações
DRL	*Deep Reinforcement Learning*
GDMs	*Generative Diffusion Models*
IA	Inteligência Artificial
IEC	*International Electrotechnical Commission*
IIoT	Internet Industrial das Coisas
IT	Tecnologia da Informação
LLM	Modelo de Linguagem de Grande Escala
MAS	Sistemas Multi-Agentes
OT	Tecnologia Operacional
PLN	Processamento de Linguagem Natural
PROFIBUS	*Process Field Bus*
Q&A	Perguntas e Respostas
QnA	Perguntas e Respostas
RAG	Geração Aumentada por Recuperação
ReAct	*Reasoning and Acting*

Seção 1

Introdução

Esta seção introduz a contextualização e a problemática deste trabalho, apresentando a motivação da pesquisa proposta. Discute o principal problema abordado, apresenta os métodos desenvolvidos para enfrentá-lo e destaca as contribuições realizadas para o avanço do estado da arte. Por fim, apresenta-se a estrutura da dissertação, oferecendo um roteiro para os capítulos subsequentes.

1.1 Motivação

A transformação digital na indústria está remodelando a relação entre a Tecnologia Operacional (OT) e a Tecnologia da Informação (IT), convergindo para o surgimento de sistemas de produção mais inteligentes e interconectados (Hollerer et al. 2024). Sabe-se que a Indústria 4.0 estabeleceu as bases para a automação, fundamentada em sistemas ciber-físicos e na Internet Industrial das Coisas (IIoT) (Vitturi et al. 2019). Em convergência, a Indústria 5.0 expande essa perspectiva, incorporando a centralidade humana, a resiliência e a sustentabilidade como princípios norteadores para a inovação no setor industrial (Buri & T. Kiss 2025).

Apesar desses avanços, a efetiva concretização da transformação digital ainda está condicionada ao uso de infraestruturas de comunicação legadas (Yanytska 2025). Tais sistemas, como o protocolo PROFIBUS (*Process Field Bus*), destacam-se como tecnologias

de automação amplamente difundidas e confiáveis, essenciais para a continuidade operacional devido ao seu determinismo e tolerância a falhas (Wollschlaeger et al. 2017, Sisinni et al. 2018). No entanto, essa infraestrutura legada é desafiadora de integrar com ambientes modernos de análise baseados em Ethernet e IIoT, resultando em subsistemas frequentemente isolados dentro de empresas orientadas a dados (Paiva et al. 2025, Sisinni et al. 2018, Yanytska 2025).

Deste modo, a convergência entre OT e IT também eleva a complexidade da gestão de segurança, segurança operacional e confiança digital, agora reconhecidas como pilares da transformação industrial (Bhole et al. 2025). No contexto da Indústria X, essa convergência exige a avaliação contínua de riscos ciber-físicos e de dependências de conformidade em sistemas distribuídos (Ahmad et al. 2024). A demanda por verificação dinâmica e raciocínio de conformidade automatizado é impulsionada pela escassez de expertise de domínio e pela limitação de escalabilidade dos processos de validação manual (AlMadhoun 2025).

Essa escassez é intensificada pelo desafio da continuidade do conhecimento. Sistemas legados, como o PROFIBUS, acumulam décadas de conhecimento de engenharia (heurísticas de configuração, diagnóstico de falhas) que permanecem dispersos em documentos não estruturados e formatos proprietários (Korodi et al. 2025). Essa dispersão dificulta a recuperação e a interpretação do conhecimento técnico relevante durante tarefas críticas de manutenção e tomada de decisão (Cavalcanti Hernandes et al. 2025). Neste contexto, a capacidade de capturar, sintetizar e disponibilizar este conhecimento de forma contextual torna-se um viabilizador importante para a inteligência industrial, por exemplo, ao permitir a geração de recomendações detalhadas de diagnóstico de falhas (de Moura et al. 2024, Adewale et al. 2025).

Modelos de Linguagem de Grande Escala (em inglês, *Large Language Models*, LLMs) têm demonstrado recentemente capacidades em compreensão de linguagem natural, raciocínio contextual e síntese de texto específica de domínio (Zhao, Fan, Li, Liu, Mei,

Wang, Wen, Wang, Zhao, Tang & Li 2024, Pan, Luo, Wang, Chen, Wang & Wu 2024). Quando combinados com a Geração Aumentada por Recuperação (em inglês, *Retrieval-Augmented Generation*, RAG), esses modelos podem fundamentar o raciocínio generativo em documentação técnica verificada, melhorando a precisão e reduzindo o risco de alucinação (Medeiros et al. 2023, Omrani et al. 2024).

No entanto, a avaliação de arquiteturas generativas em contextos industriais permanece um desafio. Métricas linguísticas tradicionais (como BLEU ou ROUGE) medem apenas a similaridade lexical, fornecendo detalhes limitados sobre a corretude e a relevância no domínio técnico (Chatoui & Ata 2021). Para superar essa limitação, avanços recentes em metodologias de LLMs, como o $LLM-as-a-Judge$, oferecem uma perspectiva complementar. Nesta abordagem, um LLM atua como avaliador, produzindo julgamentos estruturados sobre factualidade, coerência e completude (Crupi et al. 2025). Este paradigma possibilita uma avaliação escalável e contextualizada em domínios especializados, onde a anotação por especialistas é onerosa ou inviável (Sollenberger et al. 2024). É neste ponto, na validação de modelos generativos para a recuperação de conhecimento em sistemas legados, a principal contribuição deste trabalho.

1.2 Problemática e Hipóteses

A consulta a documentação técnica é uma prática para a operação, manutenção e diagnóstico de sistemas industriais complexos (Vasaniya et al. 2025). No entanto, apesar da crescente digitalização, documentos como manuais de protocolos, guias de configuração e esquemas elétricos, costumam ser extensos e organizados de modo que dificulta a localização rápida e precisa de informações específicas (Medeiros et al. 2025). Especialmente em ambientes de automação industrial, onde a agilidade é importante, a dificuldade em encontrar dados pontuais pode levar à tomada de decisões inadequadas ou ao adiamento de procedimentos (Vasaniya et al. 2025). Como consequência, aumentam-se os riscos de

falhas operacionais, interrupção da produção e comprometimento da segurança e eficiência dos sistemas industriais.

Diante desse cenário, foram elaboradas as seguintes questões de pesquisa:

QP 1 - Qual o impacto da arquitetura de orquestração (*Multi-Agent*) na precisão lexical, coerência semântica e acurácia factual das respostas em domínios técnicos, comparada às arquiteturas Single-Agent e LLM Baseline?

QP 2 - De que forma a complexidade da arquitetura RAG (comparando Single-Agent e Multi-Agent) influencia a confiabilidade do conteúdo gerado, especificamente na capacidade de manter a *faithfulness* (fundamentação no documento-fonte) e evitar alucinações?

QP 3 - Como a introdução do RAG (Single-Agent) transforma a tarefa do LLM de um raciocínio de *recall* para uma síntese contextual, e qual é o impacto líquido dessa transformação na eficiência computacional (latência), comparada ao LLM Baseline?

QP 4 - Qual é o *trade-off* operacional entre a alta qualidade (precisão e acurácia) da arquitetura *Multi-Agent* e sua eficiência computacional (tempo de resposta e emissões de carbono), e como esses custos se comparam aos demais cenários?

QP 5 - De que forma o contexto explícito nos cenários RAG (*Single* e *Multi-Agent*) influencia a convergência entre a avaliação sintética (*LLM-as-a-judge*) e o julgamento especializado humano, e qual a implicação dessa divergência para a confiabilidade na validação da acurácia do LLM *Baseline*?

Logo, a hipótese que se pretende refutar é:

Hipótese nula $\mathbf{H_0}$: A utilização de arquiteturas de orquestração *Multi-Agent* não resulta em melhorias na precisão factual, coerência semântica ou na mitigação de alucinações em comparação com abordagens menos complexas (*Single-Agent* ou *LLM Baseline*).

Hipótese alternativa $\mathbf{H_1}$: A utilização de arquiteturas de orquestração *Multi-Agent* resulta em melhorias na precisão factual, coerência semântica ou na mitigação de alucina-

ções em comparação com abordagens menos complexas (*Single-Agent* ou *LLM Baseline*).

1.3 Objetivos da Pesquisa

Para a realização desta pesquisa, tem-se os seguintes objetivos geral e específicos.

1.3.1 Objetivo Geral

O objetivo geral deste estudo é investigar o uso de arquiteturas de IA generativa para recuperar e sintetizar conhecimento a partir de documentação industrial legada, utilizando o PROFIBUS como caso representativo. São analisadas três configurações: (i) um LLM autônomo (*standalone*), (ii) um modelo RAG de agente único e (iii) uma configuração RAG multiagente. O desempenho destas configurações é avaliado por meio de métricas linguísticas quantitativas (ROUGE, METEOR, MATTR, BERTScore) e avaliações qualitativas baseadas na metodologia *LLM-as-a-Judge*, permitindo uma comparação abrangente da consistência factual e da coerência semântica entre as abordagens generativas. Os resultados indicam que a orquestração multiagente aprimora a precisão contextual e a estruturação do conteúdo no processamento de textos industriais complexos.

1.3.2 Objetivos Específicos

Para o alcance do objetivo geral, foram definidos os seguintes objetivos específicos:

(a) Estruturar e processar um *corpus* documental técnico do protocolo PROFIBUS, convertendo documentos não estruturados em bases de conhecimento vetoriais otimizadas para recuperação semântica;

(b) Implementar e orquestrar três cenários distintos (Baseline, *Single-Agent*, *Multi-Agent*) de arquitetura para acesso ao conhecimento;

(c) Avaliar a qualidade das respostas geradas sob as perspectivas léxica e semântica, utilizando métricas quantitativas (ROUGE, METEOR, BERTScore) e a metodologia qualitativa *LLM-as-a-Judge* (fidelidade e correção);

(d) Analisar o *trade-off* operacional entre as arquiteturas propostas, comparando o ganho de acurácia da abordagem multiagente em relação ao custo computacional (latência e emissões de carbono);

(e) Validar a hipótese de que a especialização e colaboração entre agentes reduzem a alucinação e aumentam a confiabilidade da informação em domínios de engenharia complexos.

1.4 Estrutura da Dissertação

Este documento está organizado da seguinte maneira. Esta Seção discutiu a motivação principal para o trabalho proposto. Desse modo, foram apresentadas a contextualização da pesquisa, problemática e hipóteses. A Seção 2 explana os conceitos e termos necessários para o entendimento do trabalho. Na Seção 3, discute-se as pesquisas correlatas, ressaltando suas diferenças com relação ao estudo proposto. Em seguida, a Seção 4 detalha a abordagem empregada, descrevendo o *pipeline* de processamento de dados, a geração de dados e a implementação dos cenários de agentes. Na Seção 5, o estudo em questão é classificado com relação aos aspectos metodológicos (natureza, abordagem dos dados, objetivos e procedimentos técnicos), assim como são detalhadas a instrumentação a ser utilizada e a abordagem proposta para extração de conhecimento em produções científicas. A Seção 6 apresenta os resultados da aplicação da metodologia proposta em três textos técnicos. Por fim, na Seção 7, são discutidas as conclusões, bem como os artigos publicados.

Seção 2

Fundamentação Teórica

Nesta seção, apresentam-se os fundamentos teóricos essenciais para a compreensão deste trabalho. Inicialmente, a Seção 2.1 introduz os conceitos de Processamento de Linguagem Natural. Em seguida, na Seção 2.2, discute-se a evolução e as características dos Sistemas de Perguntas e Respostas. A Seção 2.3 explana o funcionamento e a importância dos Bancos de Dados Vetoriais. Posteriormente, a Seção 2.4 descreve os mecanismos da técnica *Retrieval-Augmented Generation*. Avançando para paradigmas mais complexos, a Seção 2.5 aborda o conceito de agentes baseados em LLM. Por fim, a Seção 2.6 expande a discussão para arquiteturas de Multiagentes LLM.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) constitui um campo da Inteligência Artificial (IA) que visa desenvolver metodologias para capacitar máquinas a compreenderem e produzirem a linguagem humana (Cai et al. 2025). Esse campo tem experimentado um crescimento substancial, o que se reflete em uma ampla gama de produtos agora disponíveis no mercado. Tais avanços recentes melhoraram notavelmente a capacidade do PLN de alcançar uma compreensão de linguagem próxima à humana (Gardazi et al. 2025).

Historicamente, diversas abordagens foram concebidas para abordar os desafios do PLN. Estas incluem os métodos baseados em regras e as técnicas de aprendizado de má-

quina (em inglês, *machine learning*), que evoluíram para modelos híbridos e culminaram no aprendizado profundo (em inglês, *deep learning*). A trajetória dessas metodologias tem sido fundamental para impulsionar a área, permitindo que os sistemas de IA interajam de maneira cada vez mais natural com os humanos (Gardazi et al. 2025).

No contexto das aplicações recentes de PLN, uma das evoluções mais notáveis é o surgimento dos Modelos de Linguagem Grande Escala (do inglês, *Large Language Models*). Esses modelos representam o estado da arte do aprendizado profundo e têm sido cada vez mais reconhecidos como ferramentas promissoras para automatizar e otimizar a extração e análise de textos complexos (Dagli et al. 2024).

2.1.1 Modelos de Linguagem de Grande Escala

Os Modelos de Linguagem de Grande Escala são modelos de IA generativa que, em sua formulação, são predominantemente autorregressivos, sendo treinados para predizer o próximo termo em uma sequência. Esta característica lhes confere a capacidade de realizar uma ampla variedade de tarefas de PLN a partir de uma única arquitetura (Gallifant et al. 2025).

Para que os LLMs possam ser treinados em vastos *datasets* e gerenciar as longas dependências contextuais inerentes à linguagem natural de forma eficiente, é essencial uma arquitetura de rede neural que suporte o processamento paralelo e o cálculo ponderado de relações entre todos os *tokens* da sequência. Por essa razão, a base de todos os LLMs contemporâneos reside na arquitetura *Transformer* (Wu et al. 2024).

2.1.2 Arquitetura *Transformer*

A Arquitetura *Transformer* estabeleceu-se como um método para modelagem de sequências, representando a fundação dos LLMs modernos. Ao contrário das arquiteturas recorrentes anteriores, o *Transformer* introduziu um mecanismo de autoatenção (*self-*

attention). Este mecanismo permite que o modelo pondere a importância das diferentes partes da sequência de entrada ao processar cada parte da saída, modelando as dependências de forma paralela e eficaz (Wu et al. 2024).

No entanto, a arquitetura *Transformer* possui uma limitação inerente relacionada à capacidade de entrada de sequência (janela de contexto). Assim, sequências de texto extremamente longas, como documentos complexos ou grandes volumes de dados textuais, podem exceder a capacidade máxima de *tokens* que um modelo baseado em *Transformer* pode processar de forma eficiente (Roy, Morrell, Zhao & Homayouni 2024).

2.1.3 *Chunking*

Para mitigar a restrição de comprimento da sequência de entrada imposta pela complexidade do *Transformer* e habilitar os LLMs a processar e compreender documentos extensos, a técnica de *Chunking* é largamente empregada. Essa estratégia consiste em segmentar o contexto longo em segmentos menores (Tao et al. 2025).

2.2 Sistemas de Perguntas e Respostas

Sistemas de perguntas e respostas são reconhecidos como meios populares e frequentemente eficazes de busca por informação (Biancofiore et al. 2024). Uma ampla variedade de tarefas pode ser realizada usando Perguntas e Respostas, como recuperação de informação. Embora existam muitos outros métodos para responder a uma pergunta, o tipo mais comum envolve selecionar um segmento de texto de um contexto específico que responda adequadamente à questão formulada (Nassiri & Akhloufi 2023).

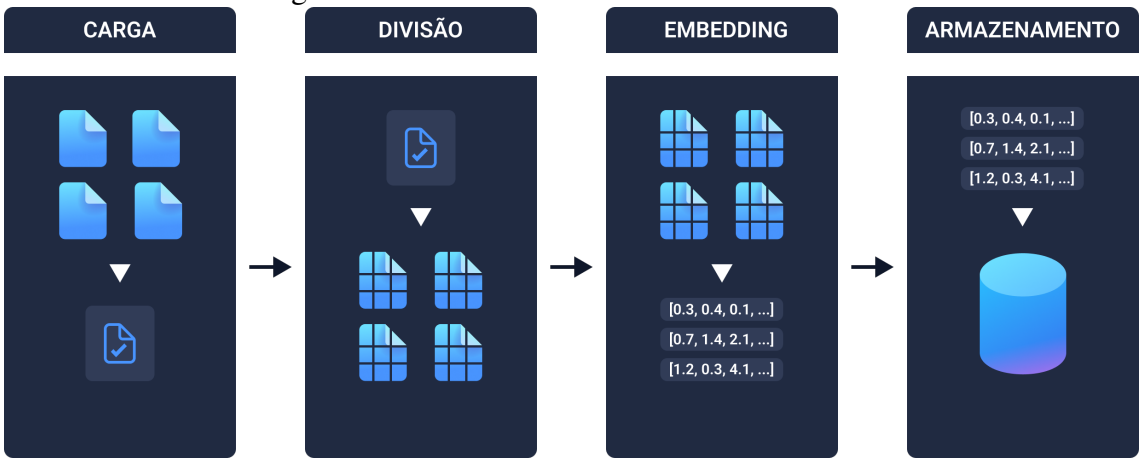
Os primeiros sistemas de Perguntas e Respostas, como o BASEBALL em 1961, operavam com modelos de linguagem baseados em regras e acesso a bancos de dados para fornecer respostas sobre tópicos específicos. Ao longo das décadas, houve avanços com melhores analisadores sintáticos e semânticos e a capacidade de converter linguagem na-

tural em consultas lógicas para bancos de dados (Nassiri & Akhloufi 2023).

2.3 Banco de Dados Vetoriais

Um banco de dados vetorial é um sistema projetado especificamente para armazenar, gerenciar e pesquisar dados na forma de vetores de alta dimensionalidade (Taipalus 2024). Esse tipo de banco consiste em um processador de consultas, responsável por interpretar e executar buscas de similaridade, e um gerenciador de armazenamento, que lida com a organização física e a indexação eficiente desses vetores (Pan, Wang & Li 2024).

Figura 2.1: Armazenamento em Banco Vetorial.



Fonte: Adaptado de LangChain (2025).

O fluxo de processamento e indexação de dados em um banco vetorial, conforme apresentado na Figura 2.1, compreende quatro etapas fundamentais:

1. **Carga**: os documentos ou fontes de dados brutos são carregados para o sistema;
2. **Divisão**: a informação é fragmentada em unidades menores (*chunks*), facilitando o processamento semântico;
3. ***Embedding***: cada fragmento de texto é convertido em uma representação numérica vetorial através de modelos de linguagem;

4. **Armazenamento**: os vetores resultantes são organizados e salvos no banco de dados vetorial, permitindo buscas de similaridade eficientes.

Aplicações como as baseadas em LLMs são, de fato, frequentemente intensivas em leitura, demandando alta taxa de transferência de consultas e baixa latência para recuperar rapidamente os vetores mais relevantes. A pontuação de similaridade, que classifica os vetores que o usuário pretende recuperar, é a principal característica das consultas de busca vetorial (Pan, Wang & Li 2024). Uma pontuação de similaridade mapeia dois vetores, **a** e **b**, de dimensão D em um escalar $f(\mathbf{a},\mathbf{b})$.

$$f : \mathbb{R}^{D} \times \mathbb{R}^{D} \rightarrow \mathbb{R} \tag{2.1}$$

Uma das possíveis maneiras de armazenar a informação em um banco vetorial é no formato de *embedding*. Esse formato é uma maneira compacta e densa de representar a informaçao em um espaço dimensional (Medeiros et al. 2023).

Eles são construídos por algoritmos não supervisionados que analisam grandes volumes de texto para capturar características essenciais das palavras, permitindo que palavras sejam mapeadas para vetores numéricos, refletindo relações semânticas e sintáticas no espaço vetorial (Medeiros et al. 2023).

2.3.1 Similaridade Cosseno

Para diferentes métricas de similaridade, o valor da pontuação final pode ter diferentes significados. No caso da similaridade cosseno, o valor final representa o ângulo entre **a** e **b**, como pode ser observado na Equação 2.2.

$$f(\mathbf{a},\mathbf{b}) = \frac{\langle \mathbf{a},\mathbf{b} \rangle}{\|\mathbf{a}\| \, \|\mathbf{b}\|} \tag{2.2}$$

No cálculo da similaridade de cosseno, uma pontuação de 1 indica que os vetores possuem a mesma direção e sentido, representando máxima similaridade. Uma pontuação

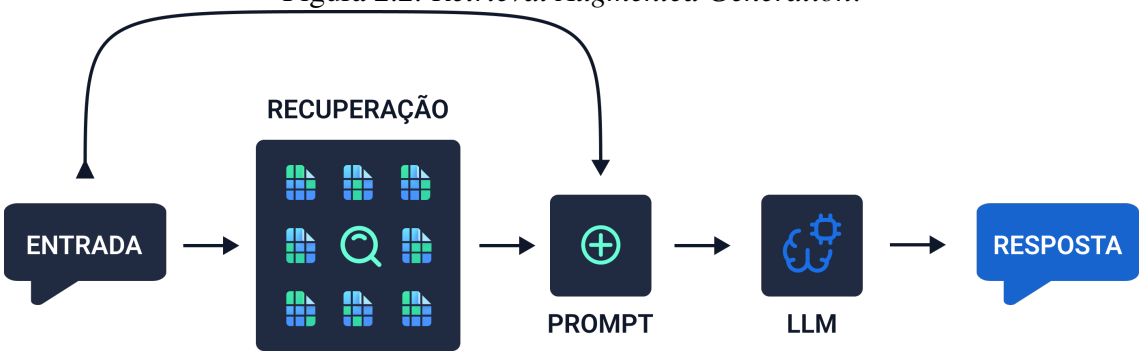
de 0 significa que os vetores são ortogonais. Por fim, um valor de -1 aponta que os vetores têm direções opostas.

2.4 *Retrieval-Augmented Generation*

A técnica de *Retrieval-Augmented Generation* (RAG) é utilizada para aprimorar o desempenho de LLMs, buscando mitigar problemas como a geração de informações incorretas ou "alucinações". Embora as abordagens envolvendo LLMs possuam capacidades avançadas, frequentemente enfrentam desafios como a produção de dados factualmente errados, conhecimento desatualizado e falta de especialização em domínios específicos (Chen et al. 2024).

Para contornar tais limitações, o RAG complementa o processo de geração dos LLMs ao introduzir uma etapa de recuperação de informações relevantes a partir de bases de conhecimento externas e dinâmicas antes que a resposta seja formulada. Essa abordagem combina o conhecimento parametrizado dos LLMs (armazenado em seus pesos durante o treinamento) com bases de conhecimento não parametrizadas, que podem ser consultadas em tempo real. Essa junção demonstrou aprimorar significativamente a precisão das respostas e reduzir a alucinação do modelo, especialmente em tarefas que exigem conhecimento intensivo e factual (Salemi & Zamani 2024).

Figura 2.2: *Retrieval Augmented Generation*.



Fonte: Adaptado de LangChain (2025).

A Figura 2.2 ilustra o fluxo fundamental da técnica de RAG, detalhando visualmente as etapas sequenciais que permitem aos LLMs acessar e utilizar conhecimento externo para aprimorar suas respostas. Como sugere o nome da técnica, esse processo é composto essencialmente pelos estágios principais:

1. Recuperação: a etapa de Recuperação é o ponto de partida do fluxo RAG, sendo acionada pela entrada de uma Busca do usuário. Conforme visualizado na Figura 2.2, o sistema utiliza essa pergunta para consultar uma base de conhecimento externa. O objetivo principal desta fase é identificar e extrair os trechos de informação — tecnicamente chamados de *chunks* — mais relevantes que possam conter a resposta ou o contexto necessário para responder à pergunta original. Técnicas de busca semântica ou por similaridade são frequentemente empregadas aqui, como exemplo a apresentada em 2.3.1, para garantir que os documentos mais pertinentes sejam selecionados.

2. Geração: com o *prompt* contendo a busca original e o contexto recuperado, ele é enviado ao LLM, conforme indicado na Figura 2.2, para produzir uma resposta. O LLM, então, utiliza suas capacidades de processamento de linguagem natural para analisar a busca do usuário levando em consideração o contexto fornecido e gerar uma resposta final. Espera-se que essa resposta seja mais precisa, factual, relevante e menos suscetível a erros ou "alucinações", pois está ancorada nas informações específicas recuperadas, em vez de depender exclusivamente do conhecimento internalizado pelo modelo durante seu treinamento.

2.5 Agentes LLM

Enquanto a técnica RAG oferece um mecanismo para ancorar as respostas de um LLM em conhecimento específico, os agentes LLM representam um avanço significativo

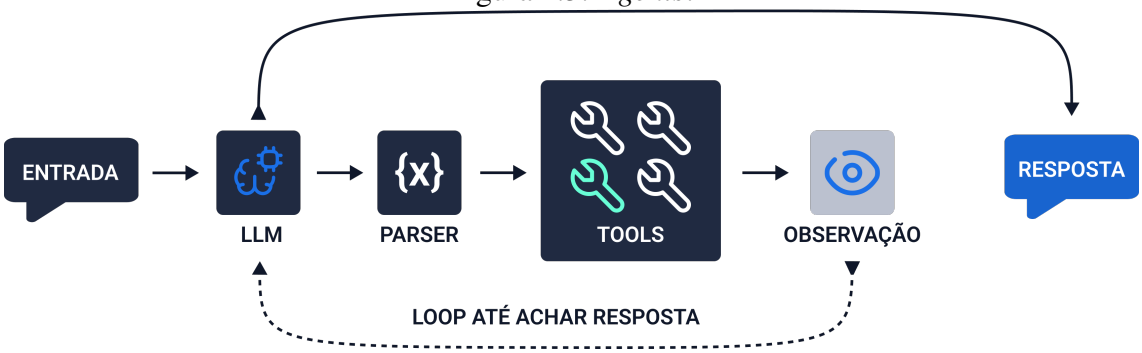
em direção à autonomia dos modelos de linguagem. Um agente LLM é definido como uma entidade que interage com um ambiente: ele observa o estado atual, executa uma ação com base nessa observação, e essa ação leva o ambiente a um novo estado (Zhao, Wang, Xiang, Zhang, Guo, Turkay, Zhang & Chen 2024).

Do ponto de vista das características individuais, um único agente é dotado de um conjunto de atributos e capacidades únicas que definem seus padrões de comportamento e seu papel no ambiente. Esses atributos podem incluir os objetivos, o conhecimento, as habilidades e os modos de interação do agente com outros agentes (Li et al. 2024).

2.5.1 Ferramentas

Uma Ferramenta (do inglês, *tool*), para um agente LLM, refere-se a um recurso ou capacidade externa que o LLM pode utilizar para aprimorar suas respostas e tomadas de decisões. A Ferramenta pode, por exemplo, permitir que os LLMs recuperem informações atuais através de buscas na web ou apliquem conhecimento especializado de fontes externas (Zhang et al. 2024).

Figura 2.3: *Agents*.



Fonte: Adaptado de LangChain (2025).

Como pode ser observado na Figura 2.3, o fluxo de um agente LLM inicia-se com uma entrada do usuário. Essa entrada é processada pelo LLM, que, utilizando suas capacidades de raciocínio, pode determinar a necessidade de informações adicionais e, consequentemente, executar chamadas a *tools* para superar suas limitações inerentes, como desatua-

lização de conhecimento (*knowledge cutoffs*) ou habilidades aritméticas deficientes. De fato, essa evolução permitiu que os LLMs selecionem e coordenem múltiplas funções com base no contexto para lidar com problemas mais complexos (Kim et al. 2024).

A decisão do LLM sobre qual ferramenta invocar e com quais argumentos é então processada por um *parser*. Este *parser* formata a requisição e extrai os parâmetros necessários para a ferramenta selecionada. Essas Ferramentas podem variar desde funções simples como uma calculadora, chamadas de API ou até mesmo ser outros agentes LLM especializados para uma tarefa específica.

Assim como demonstrado na Figura 2.3, após o uso da ferramenta, existe a observação (o resultado da execução da ferramenta). Com base nessa observação, o LLM analisa os resultados e raciocina sobre a próxima ação, decidindo se a informação retornada é relevante e suficiente para a entrada do usuário, ou se será necessária a utilização de outras ferramentas em um processo iterativo. Este ciclo de pensamento e ação continua até que uma resposta final seja alcançada.

2.5.2 *Agentic* RAG

O paradigma da *Agentic AI* descreve sistemas autônomos projetados para alcançar objetivos complexos, destacando-se pela adaptabilidade e por capacidades avançadas de tomada de decisão. Diferentemente da IA tradicional, que opera com base em instruções estruturadas, esses sistemas atuam de maneira dinâmica em ambientes em constante evolução (Acharya et al. 2025).

Uma aplicação direta desse paradigma é o conceito de *Agentic RAG*, que se refere à integração das capacidades de agentes de LLM às estruturas tradicionais de RAG. O objetivo dessa junção é justamente aprimorar a adaptabilidade e o desempenho dos sistemas de recuperação de informações (Singh 2024).

2.5.3 *ReAct Agents*

A arquitetura ReAct (do inglês, *Reasoning and Acting*), para agentes LLMs, representa um paradigma onde o agente não apenas age, mas também deve planejar suas ações. Em sua trajetória, o agente gera um "pensamento", que consiste em um passo de raciocínio interno, que informa a subsequente "ação" que o agente decide tomar. O conjunto de ações possíveis é definido por um conjunto fixo de *tools* disponíveis (Roy, Zhang, Bhave, Bansal, Las-Casas, Fonseca & Rajmohan 2024).

Figura 2.4: *ReAct Agent*.

Fonte: Autoria Própria (2025).

Este ciclo iterativo de pensamento-ação-observação, como pode ser observado na Figura 2.4, permite ao agente explorar diferentes caminhos para atingir um objetivo, aprender com os resultados de suas ações e, potencialmente, corrigir erros ou refinar sua estratégia ao longo do processo. A repetição desses passos continua até que a tarefa seja concluída.

2.6 Multiagentes LLM

Embora os sistemas de agente único se destaquem em tarefas específicas, eles podem encontrar limitações ao lidar com problemas complexos. Um Sistemas Multiagentes LLM é um sistema complexo composto por múltiplos agentes inteligentes que interagem entre si. Esse sistema é capaz de simular interações sociais e trabalho em equipe do mundo real, aprimorando a adaptabilidade e a eficiência geral por meio de processos de tomada de decisão descentralizados e compartilhamento de informações (Li et al. 2024).

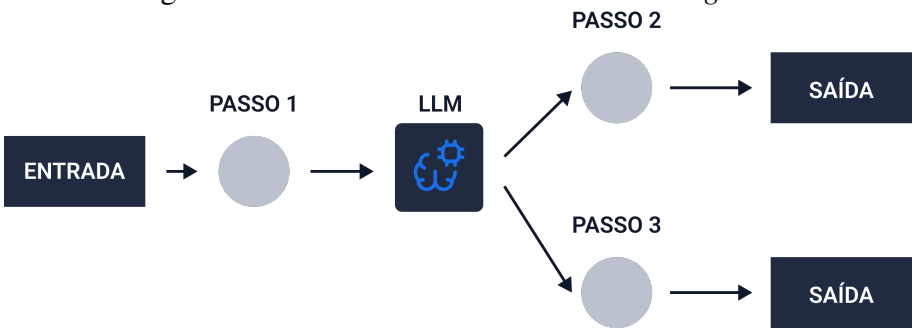
Nesse tipo de sistema, cada agente possui um grau de autonomia, sendo capaz de perceber o ambiente e tomar decisões de forma independente. Eles também podem interagir com outros agentes simulando padrões de colaboração do mundo real, como cooperação e organização hierárquica, aprimorando assim a eficiência colaborativa geral (Li et al. 2024).

2.6.1 Roteamento

Considerando os elevados custos operacionais e a latência inerente ao uso de múltiplos LLMs distribuídos, faz-se necessário que sistemas de consulta multi-modelo direcionem as requisições de forma eficiente. Uma estratégia viável para mitigar esses desafios, garantindo precisão e eficiência econômica, é a adoção do roteamento inteligente. Esta técnica consiste em selecionar, a partir de um conjunto heterogêneo de modelos especialistas, aquele mais apto para processar uma consulta específica (Stripelis et al. 2024).

A Figura 2.5 esquematiza a atuação de um LLM como um agente roteador. Nessa configuração, o modelo analisa a intenção da solicitação de entrada (Passo 1) e determina o fluxo de execução ou o especialista mais adequado para sua resolução (Passo 2 ou Passo 3).

Figura 2.5: *Mecanismo de Roteamento Inteligente*.



Fonte: Adaptado de LangChain (2025).

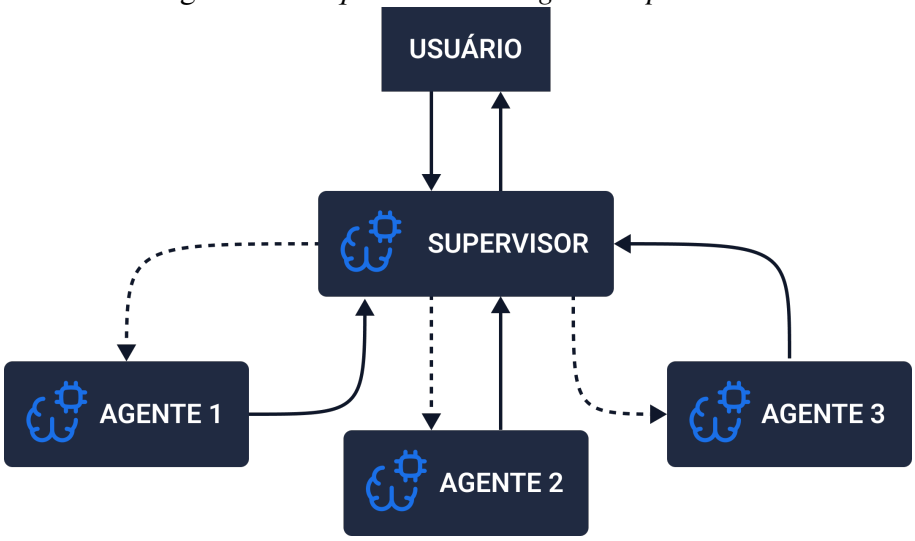
2.6.2 Supervisor

O modelo de Supervisor representa uma evolução arquitetural que incorpora e expande a lógica de Roteamento (2.6.1). Enquanto o roteamento simples opera geralmente como um mecanismo de despacho único (*one-shot*), o Supervisor utiliza essa capacidade de decisão de forma iterativa e autônoma. Nessa configuração, o LLM central atua como um orquestrador que analisa a demanda e decide dinamicamente quais e quantos especialistas consultar, possibilitando a colaboração entre múltiplos agentes para resolver problemas complexos que um único especialista não conseguiria isoladamente (Cai et al. 2024).

A função do Supervisor assemelha-se à gestão em uma estrutura social: ele avalia o estado atual da resolução e determina se o conteúdo gerado está em conformidade com as normas ou se necessita de complementação. Se a resposta for insuficiente, o Supervisor roteia a tarefa para o próximo especialista mais adequado, mantendo o controle sobre o fluxo até que a solução final seja satisfatória (Cai et al. 2024).

A Figura 2.6 ilustra essa topologia centralizada. Diferentemente do roteamento simples, onde a responsabilidade é transferida definitivamente, o Supervisor mantém a gestão do ciclo de vida da interação. Ele pode, por exemplo, acionar o Agente 1 para uma tarefa inicial e, com base no resultado, decidir rotear autonomamente a próxima etapa para o Agente 2 ou 3, coordenando assim a expertise de múltiplos domínios em um único fluxo

Figura 2.6: *Arquitetura com Agente Supervisor*.



Fonte: Adaptado de LangChain (2025).

de trabalho.

Seção 3

Trabalhos Relacionados

A aplicação de LLMs em domínios de conhecimento especializados tornou-se uma área para diversas linhas de pesquisa. Embora os LLMs de propósito geral demonstrem capacidades em raciocínio e síntese, sua eficácia em campos de alto risco e tecnicamente densos é limitada por conhecimento desatualizado e uma tendência a gerar informações factualmente incorretas, conhecidas como alucinações (Liu et al. 2025, Wahidur et al. 2025).

Em resposta a esses desafios, as abordagens de adaptação de LLMs, como o RAG, estão sendo aplicadas como solução em diversos setores, incluindo saúde e biomedicina (Harrer 2023), educação (Ye 2025), e o domínio jurídico (Wahidur et al. 2025). No entanto, a transposição bem-sucedida dessa arquitetura para o domínio da OT, especialmente para resgatar e interpretar conhecimento disperso em sistemas legados como o PROFIBUS, ainda é um campo em desenvolvimento. Além disso, a avaliação da performance do RAG nestes contextos especializados exige mais do que métricas linguísticas tradicionais, impulsionando a necessidade de abordagens de avaliação contextualizadas e escaláveis, como a metodologia *LLM-as-a-Judge*.

Deste modo, sabe-se que antes da ascensão do RAG, a adaptação de modelos era explorada via fine-tuning. A vantagem desta técnica foi demonstrada por (Zhang et al. 2024a), que aplicaram o fine-tuning em tarefas de mineração de texto na química, alcançando melhoria na acurácia e superando abordagens baseadas apenas em prompt en-

gineering. Contudo, o fine-tuning impõe desafios, ele é descrito como computacionalmente exigente, intensivo em recursos e dependente de hardware (Liu et al. 2025, Zhang et al. 2024a).

Em contraste, um estudo comparativo direto realizado por (Budakoglu & Emekci 2025) destacou que, embora o fine-tuning possa alcançar alta precisão semântica, o RAG é mais eficiente em termos de recursos, mitigando os custos de dados rotulados e treinamento associados ao fine-tuning. Essa eficiência faz do RAG uma solução preferencial para grandes repositórios de documentação legada. A eficácia do RAG, no entanto, depende da qualidade de seu componente de recuperação, que enfrenta o desafio do "*term mismatch*", onde a consulta do usuário não corresponde lexicalmente aos documentos relevantes. Abordagens de aprimoramento, como a Expansão de Consulta (Esposito et al. 2020) e mecanismos avançados como o QuIM-RAG (Saha et al. 2024), que utiliza a geração de perguntas a partir dos *chunks* para refinar a busca, têm sido propostas para mitigar a information dilution e melhorar a precisão da recuperação em domínios fechados.

Assim, para superar os desafios persistentes do RAG ingênuo e lidar com a complexidade crescente das consultas, a pesquisa tem avançado em direção a arquiteturas mais sofisticadas. Abordagens tradicionais de RAG são unidirecionais e limitadas na gestão de consultas complexas ou formatos de dados diversos (Singh et al. 2025). Em resposta, a técnica de Agentic RAG emergiu, incorporando agentes autônomos com goal reasoning e permitindo um processamento iterativo e adaptativo da informação (James et al. 2025).

O "Agentic RAG" é definido por (James et al. 2025) como uma arquitetura onde agentes LLM autônomos colaboram usando planejamento, ferramentas e decomposição de tarefas para melhorar significativamente a consistência factual e a relevância em datasets heterogêneos. A aplicação de Sistemas Multi-Agentes (MAS) em conjunto com o RAG demonstrou sucesso em áreas especializadas como o domínio jurídico (LQ-RAG, que usa múltiplos agentes e *feedback* recursivo para refinar respostas (Wahidur et al. 2025)) e a

operabilidade de geodatabases (onde um MAS decompõe consultas em linguagem natural em subtarefas precisas (Peng et al. 2025)). De modo similar, o framework MechAgents (Ni & Buehler 2024) utiliza a colaboração multiagente com divisão de trabalho (planejamento, codificação, crítica) para resolver problemas complexos de engenharia mecânica, validando o potencial da orquestração de agentes na automação de tarefas de engenharia e síntese de conhecimento.

No domínio específico de redes industriais, foco deste trabalho, a aplicação de IA também está em ascensão. Contudo, as pesquisas tem se concentrado na análise de dados operacionais (OT) e na detecção de anomalias, em vez de na extração de conhecimento documental. Exemplos incluem o uso de Machine Learning não supervisionado para detecção de anomalias em tráfego PROFINET e Ethernet/IP (Sestito et al. 2022), ou a exploração de IA Generativa (GenAI), como *Generative Diffusion Models* (GDMs) e *Deep Reinforcement Learning* (DRL), para tarefas de otimização de redes veiculares (Du et al. 2024, Zhang et al. 2024b).

Esses estudos validam a aplicação de GenAI em redes, mas não endereçam o desafio da extração de conhecimento a partir da vasta e dispersa documentação técnica de sistemas legados. O presente estudo, dá continuidade e expande a linha de pesquisa iniciada por (Medeiros et al. 2025), que estabeleceu a viabilidade da aplicação de RAG sobre documentos do protocolo PROFIBUS, focando em estratégias de recuperação e métricas de qualidade textual. Portanto, a contribuição avança essa pesquisa ao introduzir e validar a arquitetura de RAG de Multi-Agente no contexto do PROFIBUS, um passo importante para superar as limitações do RAG ingênuo na interpretação de documentos heterogêneos e no raciocínio complexo. Além disso, incorporamos o método *LLM-as-a-Judge* para oferecer uma avaliação de factualidade e coerência mais robusta e contextualizada do que as métricas linguísticas tradicionais empregadas em trabalhos anteriores.

Por fim, com o intuito de correlacionar a lacuna e a contribuição deste estudo, a Tabela 3.1 demonstra um comparativo dos trabalhos relacionados mais pertinentes. A compara-

Tabela 3.1: Resumo comparativo dos trabalhos relacionados

Características / Trabalho	Redes Industriais	RAG	Agentes	Multi-Agentes	Emissões de CO_2
(Esposito et al. 2020)	✖	✖	✖	✖	✖
(Sestito et al. 2022)	✓	✖	✖	✖	✖
(Zhang et al. 2024a)	✖	✖	✖	✖	✖
(Saha et al. 2024)	✖	✓	✖	✖	✖
(Ni & Buehler 2024)	✖	✖	✓	✓	✖
(Zhang et al. 2024b)	✓	✖	✖	✖	✖
(Budakoglu & Emekci 2025)	✖	✓	✖	✖	✖
(Ye 2025)	✖	✓	✓	✖	✖
(James et al. 2025)	✖	✓	✓	✓	✖
(Wahidur et al. 2025)	✖	✓	✓	✖	✖
(Peng et al. 2025)	✖	✓	✓	✓	✖
(Medeiros et al. 2025)	✓	✓	✖	✖	✖
Este Trabalho (2025)	✓	✓	✓	✓	✓

ção destaca que, embora o RAG e os sistemas multi-agentes sejam explorados em diversas áreas (Jurídica, Geodatabases), este trabalho, com base na pesquisa realizada até o momento, é o primeiro a investigar a orquestração de multi-agentes no RAG para a síntese de conhecimento a partir de documentação de protocolos industriais legados (PROFIBUS). A análise foca nas dimensões de domínio de aplicação, uso do RAG e de arquiteturas multi-agentes, e no impacto na qualidade e custo computacional, diferenciando-se dos estudos que se concentram apenas na análise de tráfego de rede.

Conforme demonstrado na Tabela 3.1, nenhum trabalho correlato combina simultaneamente o domínio de Redes Industriais com arquiteturas RAG e Multiagente. Assim, a principal contribuição deste estudo é a validação dessa interseção tecnológica (RAG e Multiagentes) voltada especificamente para a síntese de conhecimento em protocolos industriais. Além disso, a inclusão da análise de emissão de CO_2 (Eficiência Computacional) distingue este trabalho, oferecendo uma nova dimensão de avaliação referente à sustentabilidade e ao custo computacional.

Seção 4

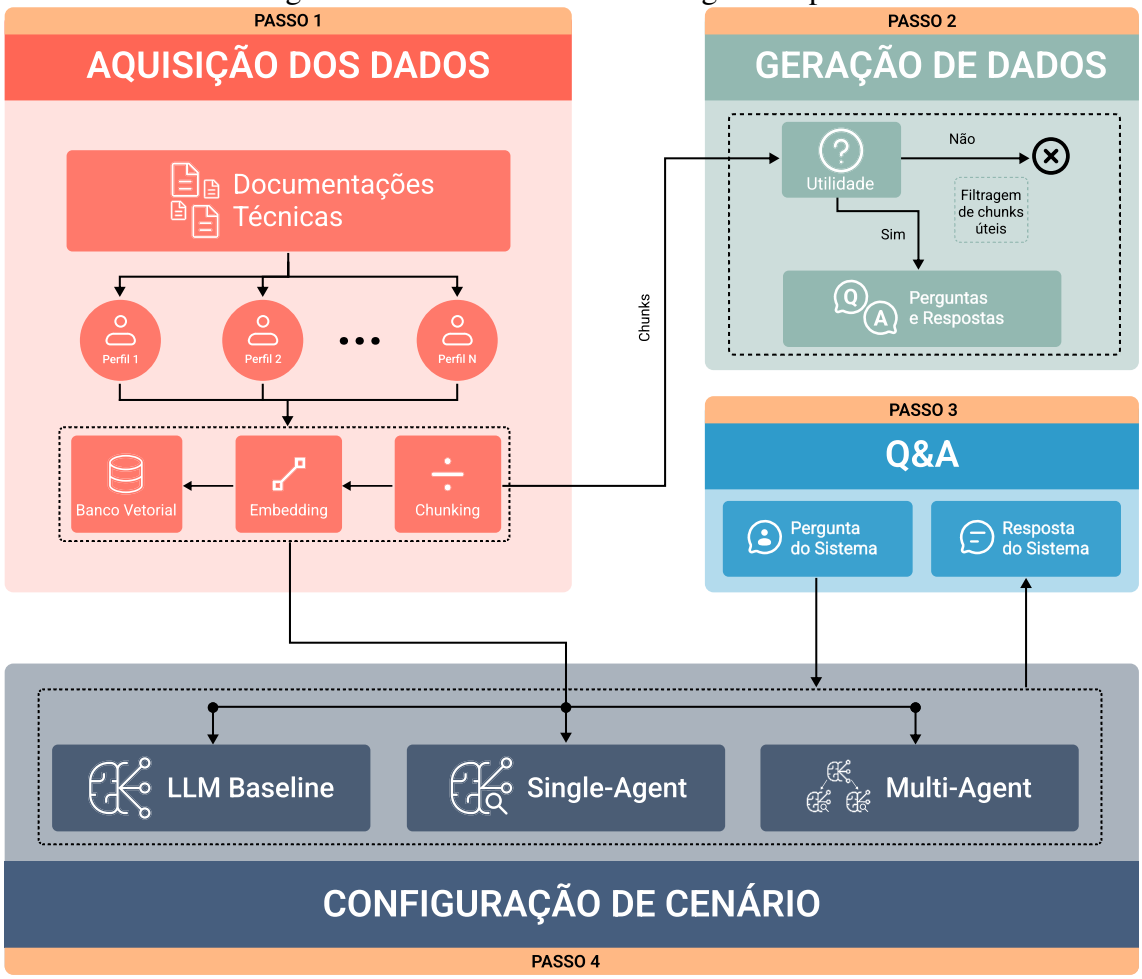
Abordagem Proposta

Nesta seção, apresenta-se a abordagem proposta que tem como objetivo melhorar a precisão, relevância e adaptabilidade no acesso ao conhecimento técnico do protocolo PROFIBUS para diagnóstico e manutenção em automação industrial. Deste modo, a abordagem contempla três fases, conforme observado na Figura 4.1. A fase (1) Aquisição de Dados, que engloba o processamento de um *corpus* documental em bases de conhecimento e a geração de questões de avaliação. Já na fase (2) Geração de Dados, ocorre a filtragem e geração dos dados utilizados de maneira automatizada. Sequencialmente, a fase (3) Q&A é responsável pela seleção das perguntas e aquisição das respostas. Por fim, a fase (4) Configuração de Cenário define três arquiteturas baseadas em LLMs (*baseline*, *Single-Agent* e *Multi-Agent*) para acesso ao conhecimento técnico do protocolo PROFIBUS.

4.1 Aquisição de Dados

Este estudo irá utilizar um corpus (documentos técnicos especializados) de 20 normas técnicas relacionadas ao protocolo PROFIBUS, fornecidas e curadas por especialistas do domínio. Para simular diferentes necessidades de usuários, o corpus foi particionado em dois perfis, o “Desenvolvedor” (com 13 documentos focados em especificações de protocolo) e o “Engenharia” (com 7 documentos centrados em instalação e manutenção de

Figura 4.1: Visão Geral da Abordagem Proposta.



Fonte: Elaborada pelo autor (2025).

rede). Deste modo, o processamento do conhecimento seguiu um *pipeline* para preparar os documentos para uso nas arquiteturas de LLMs.

Parsing e *Chunking* – o processo começa com o *Parsing*, que converte os documentos originais para um formato estruturado (como *Markdown*), facilitando a manipulação. Em seguida, ocorre o *Chunking*, na qual divide documentos longos em blocos de texto menores (*chunks*) que cabem na janela de contexto de um LLM. Neste estudo, o *chunking* foi realizado em duas etapas, segmentação baseada em frases (sintática) e um refinamento para respeitar a estrutura hierárquica (títulos e subtítulos) do documento. Isso garante que cada *chunk* não apenas contenha um texto coerente, mas também mantenha o contexto

original de documentação técnica.

Embedding – transforma cada *chunk* de texto em um vetor (uma lista de números) que captura seu significado semântico. Deste modo, uma estratégia de modelo duplo foi utilizada para maximizar a precisão. Assim, um modelo foi usado para a segmentação estrutural e um modelo de *embedding* foi empregado para gerar as representações vetoriais finais. Essa separação assegura que cada tarefa (estruturação e recuperação semântica) seja realizada pelo modelo mais adequado.

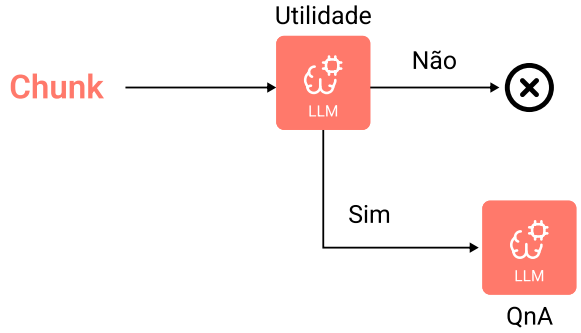
Armazenamento (Banco vetorial) – os vetores resultantes são indexados em Bancos Vetoriais, que são bancos de dados otimizados para a busca por similaridade. Esta configuração permite que a consulta do usuário seja comparada de forma rápida e eficiente aos vetores dos documentos. O armazenamento em três Bancos Vetoriais distintos (um para cada perfil e um que é a junção dos perfis) permite flexibilidade na busca e simulação de cenários específicos de usuário.

4.1.1 Geração de Dados

A criação do conjunto de dados de avaliação foi realizada por um *workflow* de geração sintética de questões baseada em agentes, buscando assegurar que as perguntas fossem significativas e representassem desafios reais do domínio.

O *workflow* de geração foi implementado utilizando a biblioteca LangGraph e empregou o modelo Gemma 3 com 27 bilhões de parâmetros com a possibilidade de processamento com raciocínio. Conforme ilustrado na Figura 4.2, o processo foi estruturado em duas etapas, com o intuito de aumentar a qualidade e relevância das questões finais. Deste modo, primeiro o LLM avalia a utilidade do bloco de documento para gerar uma boa questão (Utilidade); em seguida, gera o par de pergunta e resposta (QnA). Ambas as etapas utilizam modelo de classes com Pydantic para garantir uma saída estruturada e consistente.

Figura 4.2: Fluxo de geração sintética de dados.



Fonte: Elaborada pelo autor (2025).

4.1.2 Perguntas e Respostas (Q&A)

Para garantir uma cobertura equilibrada e representativa dos domínios técnicos do PROFIBUS, foi selecionada uma amostra estratificada do pool resultante, compondo o conjunto de avaliação final com 300 questões (150 amostradas aleatoriamente de cada perfil).

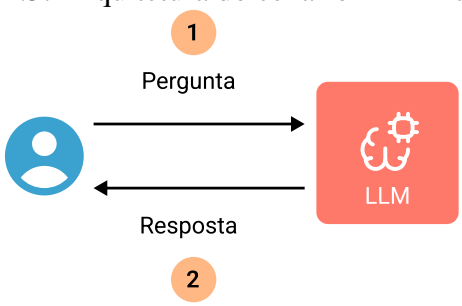
O grupo inicial de questões gerado sinteticamente no Passo 2 serviu como base para a construção do *dataset* de avaliação final. Para garantir uma cobertura equilibrada e representativa dos domínios técnicos do PROFIBUS associados aos perfis “Desenvolvedor” e “Engenharia”, realizou-se uma amostragem aleatória estratificada. Foram selecionadas 150 questões de cada perfil, compondo um conjunto de 300 questões únicas para avaliação (entrada do Passo 3 e Passo 4).

Assim, cada uma destas 300 questões foi submetida às três arquiteturas distintas definidas nos cenários de teste (Passo 4), LLM Baseline, *Single-Agent* e *Multi-Agent*. As respostas geradas por cada arquitetura para cada questão foram sistematicamente coletadas e armazenadas. Este processo resultou na criação do *dataset* final de avaliação, contendo os pares de pergunta-resposta para cada um dos três cenários, pronto para as análises quantitativas e qualitativas subsequentes.

4.2 Configuração de Cenário

Para isolar o impacto das escolhas de arquiteturas no desempenho generativo, três cenários distintos são implementados. É importante destacar que o mesmo LLM é empregado em todas as configurações, garantindo que as diferenças de desempenho observadas sejam atribuídas apenas à arquitetura de processamento de conhecimento

Figura 4.3: Arquitetura do cenário LLM Baseline.

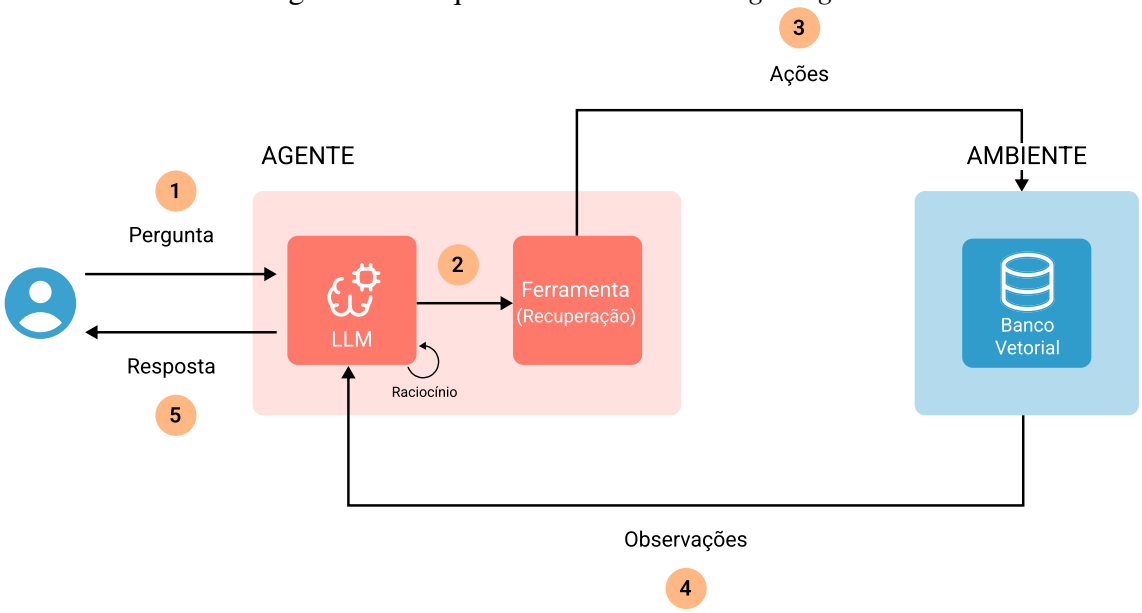


Fonte: Elaborada pelo autor (2025).

O primeiro cenário, denominado LLM Baseline, atuou como grupo de controle, representando uma configuração *zero-shot*. Nesta abordagem, ilustrada na Figura 4.3, o LLM foi acionado diretamente com as questões, operando sem acesso às bases de conhecimento vetoriais curadas. O objetivo principal foi avaliar o desempenho do modelo com base exclusivamente no conhecimento encapsulado em seus parâmetros pré-treinados, servindo como referência para as abordagens subsequentes.

O segundo cenário, *Single-Agent*, foi implementado como um agente ReAct (do inglês, *Reasoning and Acting*), cuja arquitetura é apresentada na Figura 4.4. Este agente foi configurado com uma única ferramenta de recuperação, capaz de buscar informações em uma base de conhecimento combinada (agregada a partir de todos os perfis documentais). Esta configuração foi marcada pela sua autonomia, permitindo que o agente decidisse heuristicamente se utilizaria a ferramenta de recuperação para contexto adicional ou se responderia diretamente com base no seu conhecimento interno, adaptando-se à natureza da consulta. O seu comportamento foi monitorado por um único registrador, que gravou

Figura 4.4: Arquitetura do cenário *Single-Agent*.



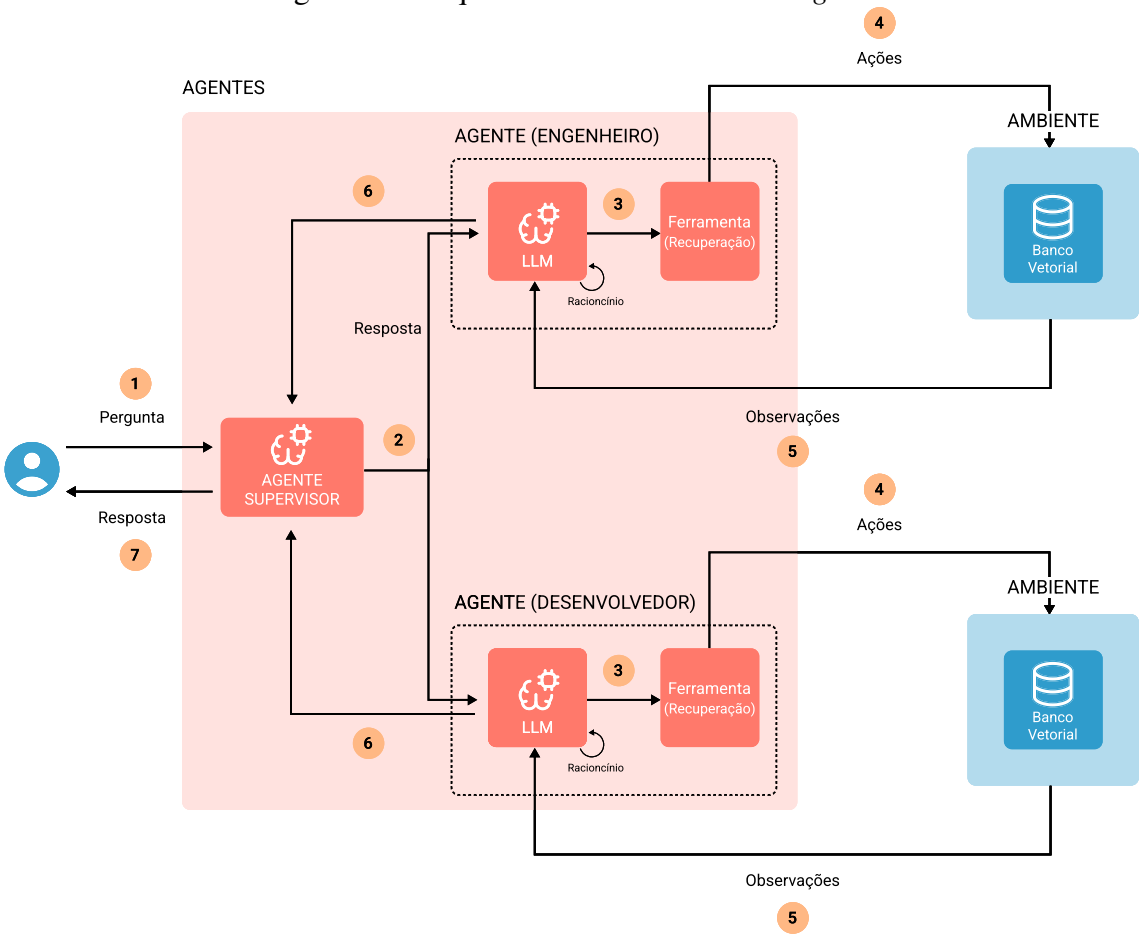
Fonte: Elaborada pelo autor (2025).

todos os blocos (*chunks*) recuperados ao longo do processo.

A terceira configuração, *Multi-Agent*, representa a arquitetura mais complexa, caracterizada pela orquestração de múltiplos agentes especializados, como detalhado na Figura 4.5. Utilizando um padrão Supervisor-Trabalhador, um agente Supervisor central analisava cada questão recebida e era responsável por roteá-la para um de dois trabalhadores especialistas, o agente Desenvolvedor ou o agente Engenheiro. Cada trabalhador foi implementado como um agente ReAct individual, atribuído de um *system prompt* específico do seu domínio e uma ferramenta de recuperação dedicada que pesquisava exclusivamente em sua base de conhecimento correspondente. Para analisar o comportamento de recuperação e a eficácia do roteamento, foram estabelecidos dois monitores independentes, um para cada agente especialista. Esta abordagem permitiu não apenas a avaliação da qualidade da resposta, mas também a medição da precisão do roteamento, verificando se o supervisor encaminhou corretamente as consultas ao agente apropriado.

Por fim, cada uma das 300 questões do conjunto de avaliação foi processada sequencialmente por todos os três cenários. Todas as respostas geradas, bem como os documentos

Figura 4.5: Arquitetura do cenário *Multi-Agent*.



Fonte: Elaborada pelo autor (2025).

fonte recuperados nos cenários RAG (*Single-Agent* e *Multi-Agent*), foram salvos em formatos estruturados para permitir a análise subsequente detalhada nas próximas seções.

Seção 5

Método

Esta seção detalha a metodologia de pesquisa empregada neste trabalho e a sua classificação segundo critérios técnicos e de abordagem. Em seguida, é apresentada a implementação específica da metodologia em um experimento focado no domínio das normas técnicas do PROFIBUS. Este domínio foi selecionado devido à alta densidade de informações complexas e especializadas, tornando-o um caso de teste para avaliar a capacidade das arquiteturas de IA Generativa de lidar com consultas técnicas que envolvem nuances.

5.1 Classificação da Pesquisa

O delineamento metodológico desta pesquisa, quanto à finalidade, classifica-se como Pesquisa Aplicada. Esta classificação se justifica por transcender o âmbito teórico e buscar mitigar gargalos operacionais reais, especificamente a dificuldade de acesso à expertise técnica do protocolo PROFIBUS.

Em relação aos objetivos, possui caráter exploratório, justificado pela investigação inédita da sinergia entre a orquestração de múltiplos agentes de IA e a Geração Aumentada por Recuperação, aplicada à gestão de conhecimento em protocolos industriais legados.

Com relação à abordagem dos dados, adota-se uma natureza Híbrida (Quali-Quantitativa) a dimensão quantitativa é atendida por métricas de similaridade lexical e eficiência com-

putacional. Enquanto a vertente qualitativa é operacionalizada pela metodologia *LLM-as-a-Judge*, que converte critérios subjetivos de fidelidade e corretude semântica em escalas mensuráveis, permitindo uma análise estatística da qualidade.

Por fim, quanto aos procedimentos técnicos, o método adotado é o experimental, visto que a validação das hipóteses ocorre por meio de experimentos controlados em ambiente laboratorial, confrontando o desempenho de arquiteturas distintas (LLM *Baseline*, RAG *Single-Agent* e RAG *Multi-Agent*) sob as mesmas condições de teste.

5.2 Instrumentação

Para a aplicação da metodologia proposta, o experimento foi desenvolvido com base no protocolo PROFIBUS (do inglês, *Process Field Bus*). O PROFIBUS, consolidado como um dos *fieldbus* mais difundidos em redes industriais, é projetado para conectar sensores, atuadores e controladores. Essa arquitetura garante uma comunicação rápida e determinística, sendo essencial para setores críticos como manufatura, energia e processos contínuos (Medeiros et al. 2025).

A escolha deste protocolo justifica-se pela sua complexidade técnica e pela sua função crítica em ambientes desafiadores. O sistema destaca-se pela capacidade de operar em áreas classificadas como risco, atendendo a requisitos de segurança intrínseca para prevenir a ignição em atmosferas explosivas. Projetado para substituir a transmissão analógica tradicional, o protocolo permite arquiteturas digitais que integram dispositivos de campo a sistemas de controle e redes de nível superior, como o Profinet (De P. Paiva et al. 2025).

Complementar, o corpus documental que serve de base para este estudo foi estruturado e dividido em dois perfis, como mencionado anteriormente, Desenvolvedor e Engenheiro. Esta segmentação visa representar as diferentes demandas de conhecimento do ambiente industrial e permitir uma avaliação comparativa do desempenho das arquiteturas de IA em subdomínios técnicos específicos.

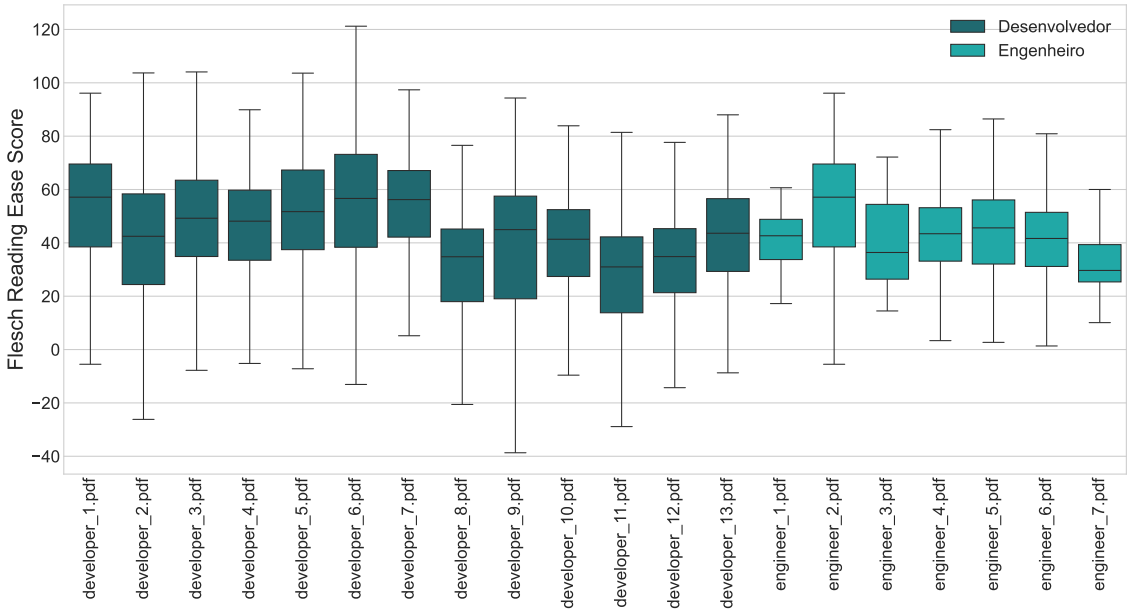
Assim, o perfil Desenvolvedor é o mais extenso e normativo, totalizando 5.182 páginas em 13 documentos, dos quais 10 são normas IEC. Seu escopo temático principal abrange especificações técnicas e perfis de protocolo para redes, detalhando a arquitetura de camadas (física, enlace de dados, aplicação), perfis de segurança (PROFIsafe) e perfis de dispositivos (PROFIdrive, GSD).

Em contraste, o perfil Engenheiro é mais conciso, com um volume de 739 páginas distribuídas em 7 documentos, e é caracterizado por não conter normas IEC. O seu escopo foca em guias e diretrizes voltados para a engenharia de implementação e manutenção de redes. O conteúdo aborda aspectos práticos como projeto, comissionamento, montagem, tecnologia de interconexão, integração de dispositivos (GSD) e procedimentos de aterramento e blindagem. Esta distinção na composição documental é importante para testar a adaptabilidade das arquiteturas de IA generativa em distintos contextos de informação industrial.

A dificuldade na interpretação desses distintos perfis documentais é confirmada pela distribuição dos *scores* de Flesch, ilustrada na Figura 5.1, fornece evidência quantitativa da dificuldade do corpus. Essa métrica avalia a complexidade textual a partir do comprimento médio de palavras e sentenças. A premissa desta métrica é que quanto mais curtas forem as palavras e as frases, maior será a facilidade de leitura do documento. Dessa forma, um valor mais elevado no índice indica uma menor complexidade e maior facilidade de compreensão (Oliveira et al. 2025).

A análise presente na Figura 5.1 revela que a maioria dos documentos (9/13 no perfil Desenvolvedor e 6/7 no perfil Engenheiro) apresentou uma mediana de escore abaixo de 50. Assim, de acordo com as interpretações estabelecidas, esses valores indicam textos complexos e de difícil leitura, exigindo um nível educacional universitário. Este resultado confirma a necessidade de ferramentas de inteligência artificial para auxiliar na interpretação e síntese deste conhecimento especializado.

Figura 5.1: Distribuição do *Score* de Flesch *Reading Ease* para cada documento.



5.3 Preparação

O *pipeline* de processamento do conhecimento foi parametrizado para estruturar o corpus de documentação do protocolo PROFIBUS. Inicialmente, o *Parsing* foi conduzido utilizando o DoclingLoader para converter os documentos técnicos originais ao formato Markdown, garantindo a uniformidade. Em seguida, *Chunking* seguiu um processo em duas fases para preservar o contexto e a hierarquia: (1) o HybridChunker, utilizando o *tokenizer* sentence-transformers/all-miniLM-l6-v2, foi empregado para uma segmentação sintática inicial, resultando em 13.725 *chunks* para o perfil Desenvolvedor e 1.007 *chunks* para o perfil Engenheiro.

Em sequência, o (2) MarkdownHeaderTextSplitter foi configurado para dividir o texto com base em cinco níveis de cabeçalho, assegurando que a estrutura hierárquica do documento fosse mantida nos *chunks* finais. Para a Vetorização (*Embedding*), adotou-se o modelo `mxbai-embed-large` (335 milhões de parâmetros) para gerar as representações vetoriais finais de 512 dimensões. Por fim, no Armazenamento, os *embeddings* resultantes

foram indexados em um total de três Bancos Vetoriais FAISS separados e pesquisáveis. Um foi dedicado ao perfil Desenvolvedor, um ao Engenheiro, e um terceiro, um Banco Vetorial Combinado, usado para cenários de busca que exigem o cruzamento do conhecimento entre os dois perfis.

5.4 *Design* Experimental

O design experimental foi concebido para avaliar o desempenho das três cenários (LLM *Baseline*, *Single-Agent* e *Multi-Agent*) sob as mesmas condições de teste. Inicialmente, o conjunto final de 300 pares de perguntas e respostas (Q&A), gerado na fase de metodologia, foi estabelecido como a verdade fundamental para a avaliação.

O protocolo de avaliação seguiu um teste cruzado. Cada uma das 300 perguntas do conjunto de dados foi inserida sequencialmente e processada por cada uma das três arquiteturas de cenário de forma independente, garantindo que cada modelo fosse exposto à mesma informação de teste. Deste modo, cada arquitetura gerou uma resposta correspondente a cada pergunta. Nos cenários RAG (*Single-Agent* e *Multi-Agent*), a geração da resposta foi precedida pela recuperação de contexto relevante dos Bancos Vetoriais.

Após a geração, as respostas de cada um dos três cenários, juntamente com os blocos de contexto recuperados nos casos RAG, foram coletadas e salvas em formatos estruturados para análise subsequente. A etapa final consistiu na avaliação, em que as respostas geradas foram comparadas com a *ground-truth* utilizando um conjunto abrangente de métricas. Este desenho experimental de teste cruzado permitiu isolar o impacto da arquitetura de processamento de conhecimento, atribuindo as diferenças de desempenho observadas à metodologia de recuperação e orquestração de agentes.

5.5 Métricas de Avaliação

A qualidade quantitativa das respostas geradas foi avaliada utilizando um conjunto de métricas automatizadas. Antes da avaliação, todos os textos (tanto de referência quanto gerados) passaram por um processo de normalização, que incluiu a conversão para minúsculas, a remoção de pontuação, preservando letras e números, e o uso de um *stemmer* (algoritmo de redução morfológica) para reduzir as palavras à sua forma raiz, conforme implementado no código de avaliação. A sobreposição lexical foi medida utilizando a métrica ROUGE (do inglês, *Recall-Oriented Understudy for Gisting Evaluation*), com seus cálculos centrais para Precisão, Revocação e F-score definidos abaixo.

$$P = \frac{\text{matching n-grams}}{\text{n-grams in generated}} \quad , \quad P_{LCS} = \frac{\text{LCS}(\text{reference}, \text{generated})}{\text{length of generated}} \tag{5.1}$$

A Equação (5.1) define a Precisão para ROUGE, tanto baseada em n-grama (P) quanto em LCS (P_{LCS}), que se refere à Maior Subsequência Comum (*Longest Common Subsequence*). Essa métrica avalia a concisão e a relevância da resposta gerada, medindo a proporção de seu conteúdo que também está presente no texto de referência.

$$R = \frac{\text{matching n-grams}}{\text{n-grams in reference}} \quad , \quad R_{LCS} = \frac{\text{LCS}(\text{reference}, \text{generated})}{\text{length of reference}} \tag{5.2}$$

A Equação (5.2) define a Revocação (*Recall*) para ROUGE, tanto baseada em n-grama (R) quanto em LCS (R_{LCS}). Essa métrica avalia a completude da resposta gerada, medindo a proporção do conteúdo do texto de referência que foi capturado com sucesso.

$$F_\beta = (1+\beta^2)\frac{P \cdot R}{(\beta^2 P) + R} \quad , \quad F_{\beta LCS} = \frac{(1+\beta^2)P_{LCS} \cdot R_{LCS}}{(\beta^2 P_{LCS}) + R_{LCS}} \tag{5.3}$$

A Equação (5.3) fornece a Pontuação-F (F-score) generalizada (F_β), que é a média harmônica ponderada da Precisão e da Revocação. O parâmetro β permite ajustar a importância da revocação sobre a precisão. Neste trabalho, foi utilizada a Pontuação-F1

($F1$-score), que corresponde a $\beta = 1$.

Nestas fórmulas, o termo LCS (referência, geração) é o comprimento da Subsequência Comum Mais Longa (do inglês, *Longest Common Subsequence*), que encontra a sequência mais longa de palavras que aparecem na mesma ordem em ambos os textos, sem a necessidade de serem contíguas. Para complementar as pontuações ROUGE, a pontuação METEOR (do inglês, *Metric for Evaluation of Translation with Explicit ORdering*) também foi calculada. O METEOR fornece uma análise lexical mais sofisticada, alinhando unigramas nos textos gerado e de referência, considerando correspondências exatas, raízes de palavras e sinônimos. A pontuação é calculada usando as seguintes fórmulas.

$$P = \frac{m}{w_t} \quad , \quad R = \frac{m}{w_r} \tag{5.4}$$

A Equação (5.4) define a Precisão (P) e a Revocação (R) de unigramas. Nestas fórmulas, m é o número de unigramas na resposta gerada que também são encontrados na referência, w_t é o número total de unigramas na resposta gerada e w_r é o número total de unigramas na referência.

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P} \tag{5.5}$$

A Equação (5.5) calcula a média harmônica ponderada (F_{mean}) da Precisão e da Revocação. Essa fórmula é projetada para atribuir mais peso à Revocação, priorizando a captura de conteúdo do texto de referência.

$$P_m = 0.5 \left(\frac{c}{u_m} \right)^3 \tag{5.6}$$

A Equação (5.6) calcula a penalidade de fragmentação (P_m), que é projetada para recompensar a fluência, penalizando respostas onde os unigramas correspondentes não estão dispostos em blocos contíguos (chunks). A variável c representa o número de sequências contíguas de unigramas idênticos, e u_m é o número total de unigramas mapeados.

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - P_m) \tag{5.7}$$

A Equação (5.7) calcula a pontuação METEOR final. Ela aplica a penalidade de fragmentação à média harmônica, resultando em uma única pontuação que equilibra a similaridade lexical com a fluência da resposta gerada.

A diversidade lexical das respostas foi medida usando o MATTR (do inglês, *Moving-Average Type-Token Ratio*). Essa métrica calcula a média das TTRs (do inglês, *Type-Token Ratios*) em todas as possíveis janelas deslizantes de tamanho fixo em todo o texto. A fórmula é definida como:

$$\text{MATTR} = \frac{1}{N-n+1} \sum_{i=1}^{N-n+1} \frac{V_i}{n} \tag{5.8}$$

Onde N é o número total de tokens no texto, n é o tamanho da janela e V_i é o número de tokens únicos na i-ésima janela. Para este estudo, foi utilizado um tamanho de janela de $n = 50$ tokens.

Passando da similaridade lexical para a semântica, o BERTScore foi empregado. Ao contrário das métricas anteriores que dependem da sobreposição de palavras, o BERTScore avalia o alinhamento no significado, calculando a similaridade de cosseno entre os *embeddings* contextuais dos *tokens* dos textos gerado e de referência. O cálculo envolve a paridade de cada *token* de um texto com o *token* mais semelhante no outro.

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^T \hat{\mathbf{x}}_j \tag{5.9}$$

A Equação (5.9) define a Revocação (*Recall*) do BERTScore (R_{BERT}). Ela é calculada pela média da pontuação máxima de similaridade de cosseno para cada token no texto de referência (x) com os tokens no texto gerado ($\hat{x}$). Isso mede quão bem o conteúdo semântico do texto de referência é capturado pela resposta gerada. Os termos $\mathbf{x}_i$ e $\hat{\mathbf{x}}_j$ representam os embeddings vetoriais contextuais para cada token.

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^T \hat{\mathbf{x}}_j \tag{5.10}$$

A Equação (5.10) define a Precisão do BERTScore (P_{BERT}). Simetricamente à revocação, ela é calculada pela média da pontuação máxima de similaridade de cosseno para cada token no texto gerado ($\hat{x}$) com os tokens no texto de referência (x). Isso mede o quanto do conteúdo semântico da resposta gerada é relevante para a referência.

$$F_{\beta BERT} = (1+\beta^2)\frac{P_{BERT} \cdot R_{BERT}}{(\beta^2 P_{BERT}) + R_{BERT}} \tag{5.11}$$

A Equação (5.11) fornece F-score generalizada ($F_{\beta BERT}$), que é a média harmônica ponderada da Precisão e da Revocação. O parâmetro β permite ajustar a importância da revocação sobre a precisão. Neste trabalho, foi utilizada a Pontuação-F1 ($F1$-score), que corresponde a $\beta = 1$.

Devido ao comprimento dos textos avaliados exceder os limites de entrada dos modelos transformer padrão, foi implementada uma abordagem personalizada por blocos. Primeiro, os textos de referência e gerados foram segmentados em blocos de, no máximo, 1500 palavras. O BERTScore foi então calculado para cada par correspondente de blocos usando o modelo `allenai/longformer-base-4096`. A pontuação final para o documento inteiro é a média ponderada pelo comprimento das pontuações de cada bloco, com os pesos sendo proporcionais ao número de palavras em cada bloco do texto gerado.

O desempenho e a eficiência de cada arquitetura foram analisados para quantificar seus custos operacionais. Duas métricas-chave foram registradas para cada resposta gerada. A primeira foi o tempo de resposta, medido em segundos desde o momento em que uma consulta foi submetida até que a resposta final completa fosse recebida, servindo como um indicador direto da latência de cada cenário. A segunda métrica foi a emissão de carbono estimada, utilizada como um indicador do custo computacional e impacto ambiental. Para esta medição, foi utilizada a biblioteca `codecarbon`. Essa ferramenta rastreia

o consumo de energia dos componentes de *hardware* (CPU e GPU) durante a execução do código e, em seguida, converte esse uso em um equivalente estimado de CO_2 em gramas, usando dados sobre a intensidade de carbono da rede elétrica local onde os experimentos foram executados.

Finalmente, para avaliar atributos de nível superior além do escopo das métricas lexicais automatizadas, foi realizada uma avaliação qualitativa usando uma abordagem de *LLM-as-a-judge*. Essa metodologia foi escolhida porque métricas tradicionais como ROUGE muitas vezes falham em capturar nuances mais profundas em tarefas generativas complexas, enquanto a avaliação humana especializada é difícil de escalar. O paradigma *LLM-as-a-judge* mescla a escalabilidade dos métodos automáticos com o raciocínio detalhado e sensível ao contexto encontrado em julgamentos especializados.

Para este estudo, o modelo Gemma 3 com 27 bilhões de parâmetros foi usado como juiz. O formato de avaliação foi concebido como uma escolha binária, que é descrita na literatura como um método simples e direto para avaliar o alinhamento factual e a precisão do conhecimento. O juiz foi solicitado a avaliar cada saída em duas dimensões específicas:

- Fidelidade - avalia se a resposta gerada é totalmente suportada pelo contexto recuperado. Uma resposta fiel não introduz informações ou faz afirmações que não estão presentes nos documentos de origem fornecidos. O cenário do LLM de Linha de Base (*Baseline* LLM) foi excluído desta avaliação, pois opera sem um mecanismo de recuperação e, portanto, não recebe nenhum contexto ao qual ser fiel.

- Correção - avalia a precisão factual da resposta, independentemente do contexto fornecido. Uma resposta correta é factualmente verdadeira de acordo com o conhecimento estabelecido do domínio.

5.5.1 Protocolo de Validação da Acurácia por Especialistas

A avaliação foi complementada por um protocolo de validação humana, para aferir a utilidade prática e a acurácia factual do conteúdo gerado nos domínios técnicos especializados.

O processo iniciou-se com a revisão, de forma independente, do conjunto de 150 questões do perfil Engenheiro para identificar aquelas mais representativas de demandas reais de usuários. A seleção de questões consideradas mais relevantes (utilidade prática) por dois especialistas de domínio (o primeiro especialista selecionou 74 questões, enquanto o segundo selecionou 92 questões). A interseção dessas duas seleções resultou em um subconjunto final de 53 perguntas que avançaram para a avaliação detalhada.

É importante destacar que a avaliação foi realizada em um regime cego (*blind*), onde os especialistas não tinham conhecimento prévio sobre qual arquitetura (*Baseline*, *Single-Agent* ou *Multi-Agent*) havia gerado cada resposta, garantindo a imparcialidade do julgamento. Para este subconjunto de respostas, foram conduzidas duas fases de avaliação:

1. Análise Binária: classificação das respostas como Corretas ou Erradas para cada cenário, estabelecendo uma linha de base de acerto factual.

2. Análise Gradual (Escala Likert): implementação de uma escala Likert de cinco pontos para mensurar o grau de acurácia (de "Totalmente Incorreta"a "Totalmente Correta"), fornecendo uma visão refinada da qualidade do conteúdo para comparação com as métricas sintéticas (*LLM-as-a-judge*).

Seção 6

Resultados

Esta seção detalha os resultados do experimento, com as constatações organizadas em três análises distintas para fornecer uma avaliação abrangente. A avaliação inicia-se com uma Análise Quantitativa (6.1), a qual afere a qualidade textual e semântica das respostas geradas utilizando métricas automatizadas. Segue-se uma Análise de Desempenho e Eficiência (6.2), mensurando custos computacionais como latência e emissões de carbono. A seção conclui com uma Análise Qualitativa (6.3), que examina a acurácia factual e a fidelidade das respostas. No âmbito de cada uma dessas análises, os resultados são apresentados por perfil documental (Desenvolvedor e Engenheiro).

Na etapa inicial de avaliação, investigou-se quantitativamente o desempenho de cada cenário utilizando a família de métricas ROUGE. Especificamente, empregaram-se as métricas ROUGE-1, ROUGE-2 e ROUGE-L para mensurar a similaridade textual entre as respostas geradas e as respostas de referência (*ground-truth*). Para cada uma dessas métricas, são reportados a Precisão (P), a Revocação (R) e o F1-Score (F1), visando fornecer um perfil de desempenho abrangente.

6.1 Análise Quantitativa

A análise das pontuações ROUGE para o perfil Desenvolvedor, apresentada na Tabela 6.1, destaca uma hierarquia de desempenho clara e consistente entre os três cenários

Tabela 6.1: Rouge - Desenvolvedor

Tipo	**Métricas**	**LLM**	***Single-Agent***	***Multi-Agent***
Rouge-1	Precisão	0,1198	0,1434	0,2290
	Revocação	0,6781	0,6616	0,6342
	F1	0,1996	0,2217	0,3129
Rouge-2	Precisão	0,0327	0,0386	0,0674
	Revocação	0,1883	0,1856	0,1909
	F1	0,0547	0,0609	0,0927
Rouge-L	Precisão	0,0657	0,0789	0,1239
	Revocação	0,3787	0,3657	0,3407
	F1	0,1097	0,1214	0,1667

avaliados. A arquitetura *Multi-Agent* demonstra uma capacidade superior de gerar texto que se alinha às respostas de referência (*ground-truth*), em comparação tanto ao *Single-Agent* quanto ao LLM.

O indicador mais expressivo desse desempenho é o F1-score, que fornece uma medida equilibrada entre precisão e revocação. O cenário *Multi-Agent* obteve os maiores F1-scores em todas as métricas (0,3129 para ROUGE-1, 0,0927 para ROUGE-2 e 0,1667 para ROUGE-L). Isso representa uma melhoria e sugere que o sistema *Multi-Agent* foi mais eficaz na captura dos termos-chave e conceitos das respostas de referência.

Uma análise mais aprofundada dos componentes do F1-score explica tal tendência, visto que o ganho de desempenho é impulsionado principalmente por um aumento na precisão. A precisão do ROUGE-1 para o cenário *Multi-Agent* (0,2290) é quase o dobro daquela observada no LLM (0,1198). Essa melhoria é consistente também nas demais métricas, em que a precisão do ROUGE-2 mais que dobrou (de 0,0327 para 0,0674), e a precisão do ROUGE-L apresentou um aumento similar de quase duas vezes (de 0,0657 para 0,1239).

As pontuações de Revocação apresentam uma relação inversa. O LLM baseline exibe a maior revocação, sugerindo que suas respostas são frequentemente prolixas e capturam uma ampla gama de palavras do texto de referência, embora de forma imprecisa. A revo-

cação ligeiramente inferior do cenário *Multi-Agent* para ROUGE-1 e ROUGE-L, quando analisada em conjunto com sua precisão significativamente superior, demonstra um clássico *trade-off* entre precisão e revocação. O F1-score superior do cenário *Multi-Agent* confirma que os ganhos substanciais em precisão compensam amplamente a diminuição marginal na revocação, resultando em uma saída mais equilibrada e quantitativamente eficaz.

Com o intuito de verificar se essas tendências de desempenho se sustentam em diferentes domínios de conhecimento, aplicou-se o mesmo protocolo de avaliação ao perfil de Engenheiro. Os resultados, sumarizados na Tabela 6.2, possibilitam avaliar a robustez de cada cenário no processamento de um conjunto distinto de documentos técnicos.

Tabela 6.2: Rouge - Engenheiro

Tipo	**Métricas**	**LLM**	***Single-Agent***	***Multi-Agent***
Rouge-1	Precisão	0,1046	0,1208	0,1970
	Revocação	0,6763	0,6778	0,5979
	F1	0,1770	0,2021	0,2764
Rouge-2	Precisão	0,0263	0,0314	0,0498
	Revocação	0,1727	0,1764	0,1538
	F1	0,0446	0,0525	0,0709
Rouge-L	Precisão	0,0561	0,0634	0,1030
	Revocação	0,3695	0,3629	0,3100
	F1	0,0950	0,1064	0,1420

As pontuações ROUGE para o perfil Engenheiro, apresentadas na Tabela 6.2, reproduzem as tendências de desempenho observadas no conjunto de dados de questões do perfil Desenvolvedor. O cenário *Multi-Agent* mais uma vez supera tanto as configurações *Single-Agent* quanto o LLM isolado em todas as principais métricas.

Essa superioridade é demonstrada de forma mais clara pelos F1-scores. O sistema *Multi-Agent* obteve as pontuações mais altas para ROUGE-1 (0,2764), ROUGE-2 (0,0709) e ROUGE-L (0,1420). Mais uma vez, esse desempenho aprimorado atribui-se a um aumento substancial na precisão. A precisão do ROUGE-1 para o modelo *Multi-Agent*

(0,1970) é quase 90% superior à do LLM (0,1046). Este resultado sugere fortemente que a arquitetura *Multi-Agent* é mais eficaz na geração de respostas relevantes e concisas.

O *trade-off* entre precisão e revocação também se faz evidente nesta análise. Os modelos LLM isolado e *Single-Agent* apresentam alta revocação, porém isso ocorre em detrimento de uma precisão muito baixa. A pontuação de revocação inferior do modelo *Multi-Agent* é compensada por seus ganhos em precisão. Isso confirma que o modelo não está meramente omitindo informações, mas sim focando no conteúdo mais relevante para construir uma resposta mais acurada e eficaz.

Assim, os F1-scores superiores do cenário *Multi-Agent* foram observados tanto no perfil Desenvolvedor quanto no Engenheiro, sugerindo uma vantagem de desempenho consistente em relação aos demais cenários deste estudo. Embora as pontuações ROUGE ofereçam *insights* sobre a sobreposição lexical, elas podem não capturar totalmente o significado semântico do texto gerado.

Tabela 6.3: MATTR e METEOR por Perfil

Cenário	**METEOR**		**MATTR**	
	Desenvolvedor	**Engenheiro**	**Desenvolvedor**	**Engenheiro**
LLM	0,2567	0,2431	0,7865	0,8155
Single-Agent	0,2667	0,2638	0,7897	0,8201
Multi-Agent	0,3018	0,2785	0,7886	0,8230

A análise foi ampliada para incluir as pontuações METEOR e MATTR, que fornecem *insights* sobre a qualidade linguística e a diversidade lexical, respectivamente, conforme apresentado na Tabela 6.3. As pontuações METEOR mostram-se consistentes com as métricas anteriores. Para o perfil Desenvolvedor, o cenário *Multi-Agent* obteve uma pontuação de 0,3018, superior à do *Single-Agent* (0,2667) e do LLM baseline (0,2567). Essa tendência também foi confirmada no perfil de Engenheiro.

O cenário *Multi-Agent* obteve novamente a maior pontuação (0,2785), seguido pelo *Single-Agent* (0,2638) e pelo LLM (0,2431). A consistência dessa vantagem de desem-

penho em ambos os conjuntos de dados reforça as constatações das avaliações ROUGE. Essa pontuação METEOR superior indica um melhor alinhamento global entre os textos gerados e os de referência em termos de escolha de palavras (incluindo sinônimos e radicais) e estrutura frasal.

A pontuação superior do modelo *Multi-Agent* sugere que suas respostas são mais acuradas em conteúdo do que os demais cenários. Além disso, essas respostas estão dispostas em uma ordem mais fluida e lógica que melhor corresponde à resposta de referência, indicando uma qualidade linguística global mais elevada. Por fim, calculou-se o MATTR para avaliar a diversidade lexical das respostas geradas.

Em contraste com as demais métricas, as pontuações MATTR na Tabela 6.3 revelam que as diferenças na diversidade lexical são mínimas. Para o perfil Desenvolvedor, todos os cenários registraram pontuações muito similares (em torno de 0,78-0,79). Embora os resultados para o perfil Engenheiro demonstrem uma leve tendência de alta, com o modelo *Multi-Agent* alcançando a maior pontuação (0,8230), as margens entre os modelos permanecem muito estreitas.

Esta constatação em ambos os perfis sugere que a vantagem consistente de desempenho do cenário *Multi-Agent* não decorre de um vocabulário mais diversificado. Em vez disso, sua superioridade reside na seleção e na disposição das palavras para alcançar maior precisão lexical e fluência estrutural, conforme demonstrado pelas demais métricas. Para fornecer uma avaliação mais abrangente e refinada, a análise foi estendida para incluir o BERTScore.

Tabela 6.4: BERTScore por Perfil

Cenário	**Desenvolvedor**			**Engenheiro**		
	Precisão	**Revocação**	**F1**	**Precisão**	**Revocação**	**F1**
LLM	0,8175	0,8798	0,8474	0,8133	0,8813	0,8459
Single-Agent	0,8207	0,8772	0,8479	0,8164	0,8823	0,8480
Multi-Agent	0,8289	0,8769	0,8520	0,8252	0,8756	0,8495

A análise dos resultados do BERTScore para o perfil Desenvolvedor, apresentada na Tabela 6.4, fornece uma visão em nível semântico do desempenho dos cenários e reforça as constatações da análise lexical. De modo geral, todos os cenários alcançaram F1-scores elevados, com o cenário *Multi-Agent* obtendo a maior pontuação tanto no perfil Desenvolvedor (0,8520) quanto no Engenheiro (0,8495). Isso indica que todos os modelos foram eficazes na geração de respostas semanticamente similares às respostas de referência. Contudo, o sistema *Multi-Agent* ainda detém uma leve vantagem na qualidade semântica global.

Os resultados revelam também o mesmo *trade-off* entre precisão e revocação observado na análise ROUGE. O cenário *Multi-Agent* obteve a maior precisão em ambos os perfis Desenvolvedor (0,8289) e Engenheiro (0,8252), sugerindo que suas respostas são as mais semanticamente precisas e contêm a menor quantidade de informações semanticamente irrelevantes. Por outro lado, a maior revocação foi obtida pelo LLM baseline no perfil Desenvolvedor (0,8798) e pelo *Single-Agent* no perfil Engenheiro (0,8823), indicando que suas saídas são as mais semanticamente abrangentes, embora menos precisas.

A avaliação do BERTScore corrobora a hierarquia de desempenho identificada pelas métricas lexicais. Conforme demonstrado na Tabela 6.4, todos os cenários alcançaram F1-scores elevados, indicando que todos os modelos geraram respostas semanticamente adequadas. No entanto, o cenário *Multi-Agent* obteve o maior F1-score em ambos os perfis Desenvolvedor (0,8520) e Engenheiro (0,8495), confirmando sua leve vantagem na qualidade semântica global. Embora a diferença de desempenho entre os cenários seja menor nesta análise semântica, o sistema *Multi-Agent* demonstra um melhor equilíbrio entre precisão e revocação.

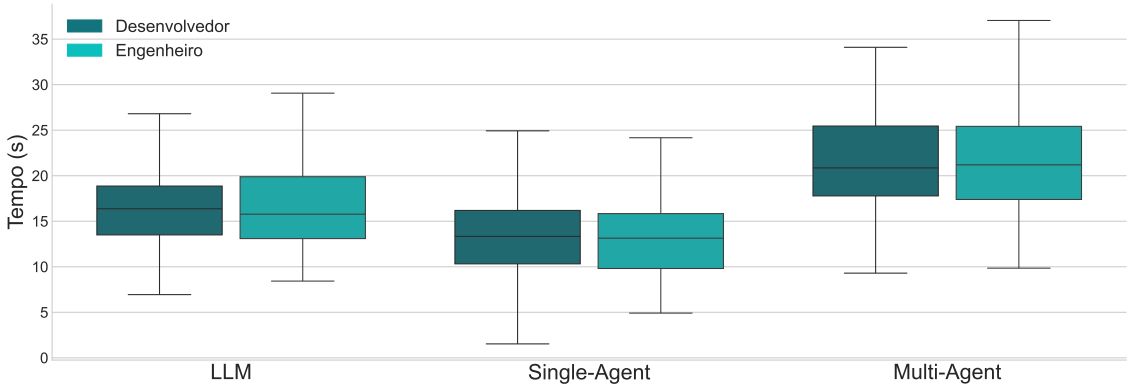
Essa vantagem decorre de um claro *trade-off* entre precisão e revocação. O cenário *Multi-Agent* alcançou a maior precisão em ambos os perfis Desenvolvedor (0,8289) e Engenheiro (0,8252), sugerindo que suas respostas são as mais semanticamente focadas. Por outro lado, a maior revocação foi obtida pelo LLM baseline no perfil Desenvolvedor

(0,8798) e pelo *Single-Agent* no perfil Engenheiro (0,8823). Esse padrão confirma que a superioridade da arquitetura *Multi-Agent* advém da geração de conteúdo mais preciso e relevante, conduzindo ao maior F1-score em ambos os domínios de conhecimento.

6.2 Análise de Eficiência

Para avaliar a eficiência computacional e o impacto ambiental de cada cenário, foram registrados tanto o tempo de geração da resposta quanto as emissões estimadas de carbono.

Figura 6.1: *Resultados de Tempo*

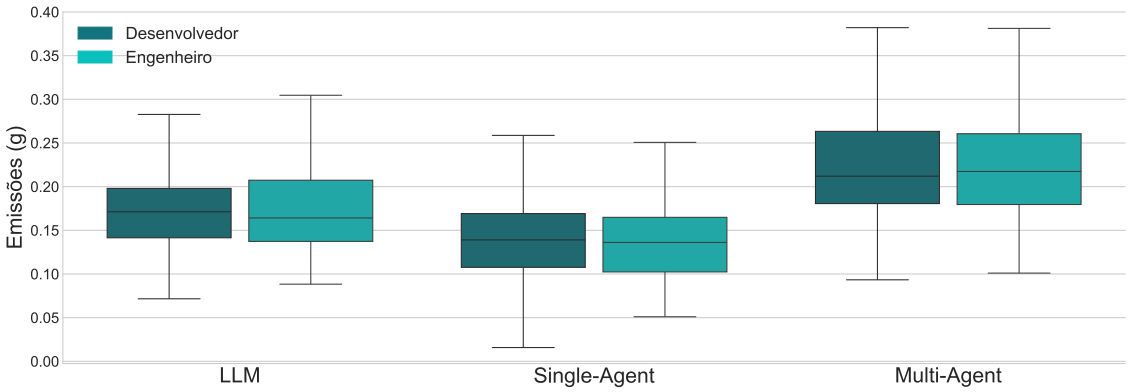


A análise do tempo de resposta, ilustrada na Figura 6.1, revela diferenças entre os cenários. O cenário *Single-Agent* foi, em média, a configuração mais rápida, com uma mediana de tempo de resposta de aproximadamente 13-14 segundos. Em contrapartida, o cenário *Multi-Agent* foi o mais lento, demandando uma mediana de tempo de aproximadamente 20-21 segundos. Além disso, a arquitetura *Multi-Agent* não apenas demandou mais tempo, mas também exibiu a maior variabilidade de desempenho, conforme evidenciado pelo seu intervalo interquartil mais amplo, sugerindo que sua maior complexidade conduz a tempos de processamento menos previsíveis.

Uma explicação plausível para o desempenho mais rápido do cenário *Single-Agent* em comparação ao LLM baseline, a despeito da etapa adicional de recuperação, pode

residir em uma mudança fundamental na tarefa executada pelo modelo de linguagem. Levanta-se a hipótese de que, no cenário baseline, o modelo se engaja em uma tarefa de raciocínio aberto e revocação computacionalmente mais intensiva. Tal processo seria provavelmente mais complexo, o que é consistente com a prolixidade observada nas saídas e com o tempo de geração mais longo. Inversamente, a tarefa no cenário *Single-Agent* parece ser transformada em uma de síntese contextual. O fornecimento de informações relevantes diretamente no *prompt* provavelmente simplifica o raciocínio necessário, reduzindo potencialmente a carga computacional e viabilizando a geração mais concisa e veloz observada nos resultados. Isso sugere que a economia de tempo durante uma fase de geração mais focada pode ser suficiente para compensar o *overhead* da etapa de recuperação de documentos.

Figura 6.2: *Resultados de Emissões*



As emissões estimadas de carbono, apresentadas na Figura 6.2, demonstram uma correlação forte e direta com os resultados de tempo de resposta. O cenário *Single-Agent* mostrou-se o mais eficiente, registrando a menor mediana de emissões (aproximadamente 0,14 g). O cenário *Multi-Agent*, devido à sua maior demanda computacional, apresentou o maior impacto ambiental, com emissões medianas em torno de 0,22 g.

6.3 Análise Qualitativa

Embora as métricas quantitativas mensurem o desempenho, elas não capturam certos aspectos qualitativos das respostas geradas. Para endereçar essa questão, conduziu-se uma avaliação qualitativa utilizando a metodologia *LLM-as-a-judge*. Nesta abordagem, um LLM independente avalia a saída de cada cenário com base em critérios específicos. Os dois critérios avaliados foram fidelidade e correção.

Tabela 6.5: Análise de Fidelidade por Perfil

Perfil	**Cenário**	**Fiel**	**Infiel**
Desenvolvedor	*Single-Agent*	71	79
	Multi-Agent	120	30
Engenheiro	*Single-Agent*	97	53
	Multi-Agent	121	29

Os resultados da avaliação de fidelidade para o perfil Desenvolvedor, apresentados na Tabela 6.5, demonstram uma diferença de desempenho entre os dois cenários RAG. O cenário *Multi-Agent* demonstra um alto grau de fidelidade, com 120 de 150 (80%) respostas julgadas como corretamente fundamentadas nos documentos-fonte fornecidos.

Em contrapartida, o cenário *Single-Agent* foi avaliado como não fiel na maioria dos casos, com apenas 71 de 150 respostas (47,3%) consideradas fiéis ao contexto recuperado. Estes dados indicam que a arquitetura *Multi-Agent* foi mais confiável na adesão às informações-fonte e na prevenção de alucinação de conteúdo para este perfil.

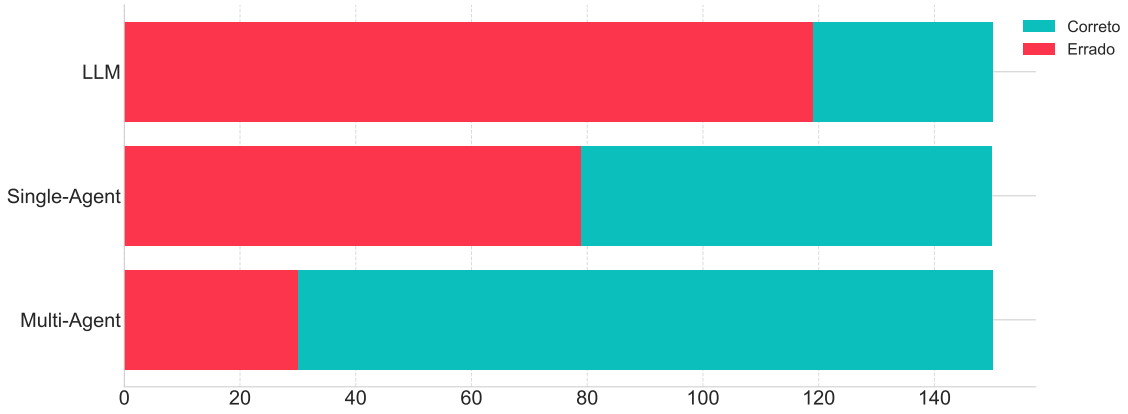
Uma tendência similar foi observada para o perfil Engenheiro, conforme demonstrado na Figura 6.5. O cenário *Multi-Agent* demonstrou novamente um nível elevado e consistente de fidelidade, com 121 de 150 respostas (80,7%) fundamentadas no contexto fornecido. O desempenho do *Single-Agent* apresentou melhora para este perfil, mas permaneceu inferior ao do *Multi-Agent*, alcançando uma taxa de fidelidade de 64,7% (97 de 150 respostas).

Considerada em conjunto, a avaliação de fidelidade indica que a arquitetura *Multi-*

Agent é consistentemente confiável na adesão às informações-fonte, mantendo uma taxa de fidelidade de aproximadamente 80% em ambos os domínios especializados.

Enquanto a fidelidade avalia se uma resposta está fundamentada no material-fonte fornecido, conduziu-se uma avaliação separada para determinar a correção factual das respostas, independentemente de sua origem.

Figura 6.3: *Correção - Desenvolvedor*

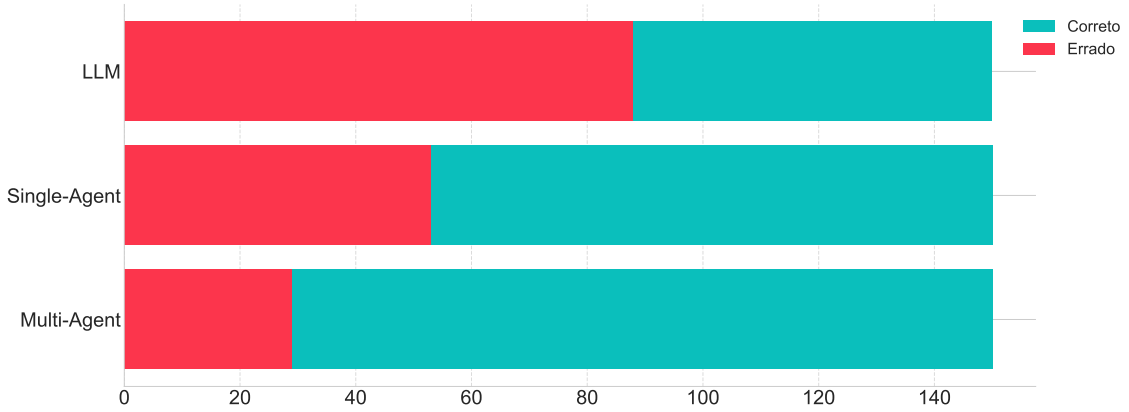


Os resultados da avaliação de correção para o perfil Desenvolvedor, apresentados na Figura 6.3, demonstram uma clara hierarquia de desempenho. O cenário *Multi-Agent* produziu as respostas mais acuradas, alcançando uma taxa de correção de 80% (120 de 150 respostas).

O *Single-Agent* apresentou desempenho intermediário, com pouco menos da metade de suas respostas julgadas como corretas, atingindo 47,3% (71 de 150). Demonstrando o menor desempenho, o LLM baseline forneceu respostas factualmente corretas em apenas 20,7% dos casos (31 de 150). Isso indica que a configuração arquitetural do sistema *Multi-Agent* está associada a uma taxa maior de saídas acuradas para o perfil Desenvolvedor.

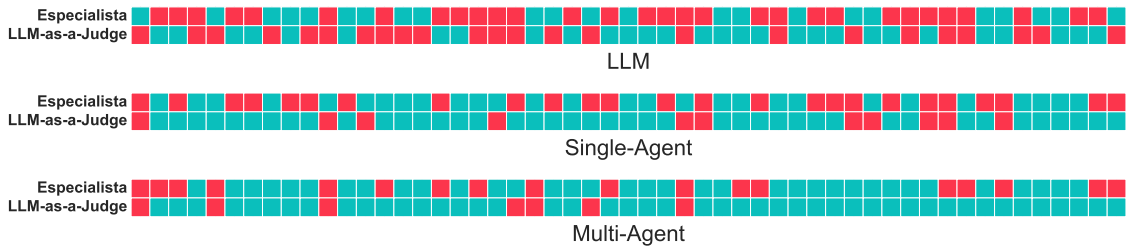
Essa hierarquia de desempenho manteve-se consistente no perfil Engenheiro, conforme demonstrado na Figura 6.4. O cenário *Multi-Agent* alcançou novamente a maior taxa de correção, com 87,3% (131 de 150 respostas). O desempenho tanto do *Single-Agent* quanto do LLM baseline foi superior neste conjunto de dados, alcançando taxas de correção de 64,7% (97 de 150) e 41,3% (62 de 150), respectivamente.

Figura 6.4: *Correção - Engenheiro*



Complementar às métricas automatizadas, conforme detalhado no experimento, conduziu-se uma avaliação humana envolvendo especialistas do domínio para aferir a utilidade prática dos modelos. Assim, os resultados da análise dos especialistas, ilustrados na Figura 6.5, corroboram as tendências observadas na avaliação automatizada de correção. O cenário *Multi-Agent* alcançou o desempenho mais elevado, com uma taxa de correção de 66,04%. O cenário *Single-Agent* seguiu com 52,83%, enquanto o LLM Baseline obteve a menor pontuação, atingindo 43,40%.

Figura 6.5: Comparação da Avaliação Binária entre Especialistas e *LLM-as-a-judge*

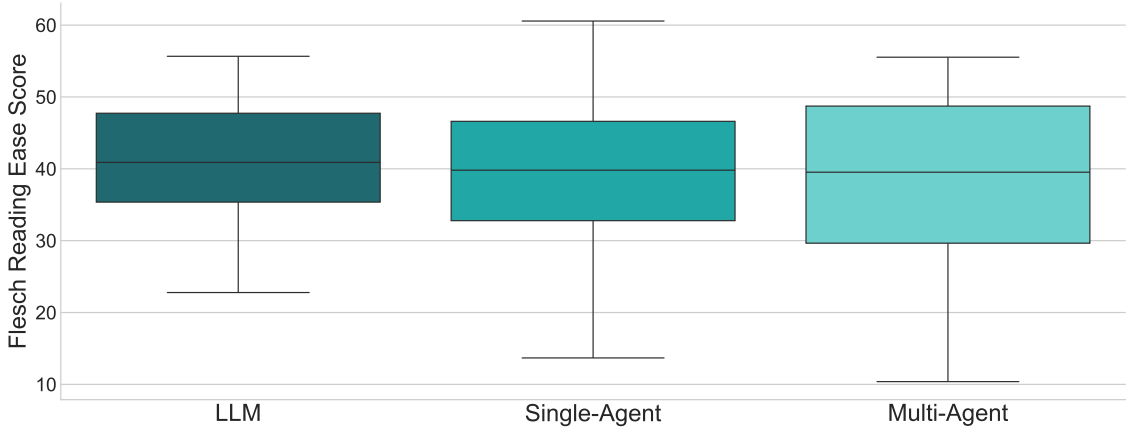


No entanto, ao confrontar as métricas, observa-se uma discordância mais acentuada entre os especialistas e o LLM-as-a-judge especificamente no cenário LLM Baseline. Nos cenários assistidos por RAG, observou-se uma maior convergência entre as avaliações humana e sintética (*LLM-as-a-judge*). Em contraste, a avaliação do modelo isolado (LLM Baseline) apresentou maior divergência, sugerindo que a detecção de inconsistências em respostas geradas por conhecimento paramétrico apresenta maior complexidade para a

avaliação do *LLM-as-a-judge*. Consequentemente, esta constatação implica que a presença de contexto explícito em cenários com recuperação de informação atua como um mecanismo de fundamentação não apenas para a geração, mas também para a validação. Deste modo, possibilita reduzir a ambiguidade que pode conduzir ao desalinhamento entre as métricas automatizadas e o julgamento humano.

Ademais, uma análise comparativa da distribuição de acertos entre os cenários revela padrões distintos. Em 18,87% dos casos, todos os três cenários forneceram a resposta correta, ao passo que 11,32% das questões não foram respondidas corretamente por nenhum modelo. Assim, percebe-se que 32,08% das questões foram respondidas corretamente por apenas um cenário específico, e 35,85% foram resolvidas corretamente por dois cenários simultaneamente. Essa distribuição sugere que, embora exista uma sobreposição de capacidade para consultas mais simples, a arquitetura *Multi-Agent* demonstra robustez superior no processamento do subconjunto específico de questões consideradas as mais relevantes pelos especialistas do domínio.

Figura 6.6: *Score* de Flesch *Reading* para as Respostas dos Cenários.

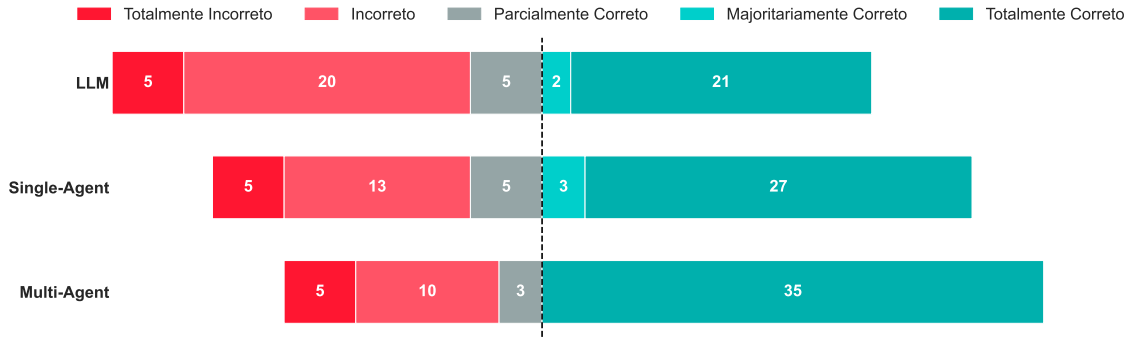


A Figura 6.6 apresenta a distribuição das pontuações de legibilidade nos três cenários. Uma observação central é a estabilidade da pontuação mediana, que permanece consistente nas arquiteturas LLM Baseline, *Single-Agent* e *Multi-Agent*, situando-se aproximadamente em torno de 40.

No entanto, a dispersão dos dados revela diferenças estruturais na geração. Enquanto o LLM exibe uma distribuição mais compacta, os cenários baseados em agentes apresentam intervalos interquartis mais largos e uma extensão das hastes maior. O cenário *Multi-Agent*, em particular, atinge as pontuações mínimas mais baixas do experimento (próximas a 10), o que se aproxima melhor da informação obtida na Figura 5.1. Essa maior variância implica que a arquitetura *Multi-Agent* possui maior flexibilidade, embora mantenha uma legibilidade média similar à do baseline. Deste modo, ela demonstra capacidade de produzir explicações tecnicamente mais densas e complexas (pontuações inferiores) quando a profundidade da consulta assim o exige.

Para fornecer uma avaliação mais granular da qualidade das respostas, para além da métrica binária de correção, conduziu-se uma avaliação suplementar utilizando uma escala Likert de 5 pontos. Este método permitiu a mensuração do grau de acurácia, classificando as respostas nas seguintes categorias: (-2) Totalmente Incorreta, (-1) Incorreta, (0) Parcialmente Correta, (1) Majoritariamente Correta e (2) Totalmente Correta. Essa avaliação foi aplicada ao mesmo subconjunto de 53 questões selecionadas pelos especialistas.

Figura 6.7: Resultado da Avaliação da Escala de Likert.



Para quantificar o desempenho global de cada cenário, calculou-se uma pontuação (S) utilizando a soma ponderada das contagens de respostas para cada categoria. A pontuação é definida pela Equação 6.1.

$$S = \sum_{i=1}^{5} w_i \cdot n_i \tag{6.1}$$

Onde w_i representa o peso atribuído à i-ésima categoria. Desse modo, tem-se que $w \in \{-2,-1,0,1,2\}$ constitui o conjunto de valores ordinais, e n_i denota a quantidade de respostas associadas a essa classificação específica. Este mecanismo de pontuação penaliza informações incorretas ao passo que recompensa respostas de alta qualidade, com respostas parcialmente corretas atuando como uma linha de base neutra.

A aplicação desta métrica ao subconjunto de 53 questões selecionadas por especialistas resultou em perfis de desempenho distintos para cada arquitetura. O cenário *Multi-Agent* alcançou a maior pontuação, totalizando 50. O cenário *Single-Agent* seguiu com uma pontuação de 34, enquanto o LLM baseline registrou o menor desempenho, com uma pontuação de 14.

Estes resultados evidenciam uma disparidade qualitativa significativa. A diferença substancial entre a pontuação do *Multi-Agent* (50) e a do LLM Baseline (14) sugere que a arquitetura *Multi-Agent* minimiza efetivamente a geração de conteúdo alucinado ou enganoso (pesos negativos), ao mesmo tempo em que maximiza a densidade de informações acuradas.

6.4 Discussão

A análise e interpretação dos resultados são apresentadas nesta seção, onde dados quantitativos e qualitativos são integrados para fornecer respostas às cinco Questões de Pesquisa.

Deste modo, com relação à análise da qualidade do conteúdo, estabelece a arquitetura *Multi-Agent* como superior, respondendo a QP 1 sobre o impacto da orquestração na precisão lexical, coerência semântica e acurácia factual. O desempenho aprimorado do *Multi-Agent* não é apenas marginal, mas qualitativo, sendo impulsionado por um ganho

na precisão.

Este ganho é evidenciado pelas métricas ROUGE, nas quais o *Multi-Agent* alcançou os maiores F1-scores, predominantemente devido ao aumento da precisão. A precisão ROUGE-1 do *Multi-Agent* (0,2290 para Desenvolvedor) foi quase o dobro da observada no LLM Baseline (0,1198). Essa acurácia lexical se estende ao domínio semântico, com as maiores pontuações BERTScore F1 e METEOR, confirmando que a orquestração aprimora a seleção e a disposição das informações de maneira mais fluida e lógica.

Em termos de acurácia factual, a superioridade se torna crítica. O *Multi-Agent* atingiu as maiores taxas de correção (*LLM-as-a-judge*), variando de 80% a 87,3%, em forte contraste com o LLM Baseline (20,7% a 41,3%). A avaliação em escala Likert (*Multi-Agent* 50; LLM Baseline 14) reitera que o modelo minimiza a geração de conteúdo enganoso ou incorreto.

Com relação a QP 2 que aborda a complexidade do RAG e sua influência na confiabilidade, especificamente na capacidade de manter a *faithfulness* e evitar alucinações. Os resultados demonstram que a complexidade da arquitetura *Multi-Agent* é fundamental para garantir a adesão ao documento-fonte.

O *Multi-Agent* alcançou aproximadamente 80% de fidelidade ao contexto recuperado. Em contrapartida, o *Single-Agent*, que utiliza uma forma mais simples de injeção de contexto, falhou em manter a *faithfulness* em quase metade dos casos para o perfil Desenvolvedor (47,3% de fidelidade). Essa discrepância prova que a simples recuperação de dados não é suficiente para impedir que o LLM recorra ao seu conhecimento paramétrico. A orquestração *Multi-Agent*, por meio de etapas de verificação e planejamento, atua como um mecanismo de fundamentação robusto, garantindo que a alta acurácia factual seja inerente ao contexto e não uma alucinação.

Em sequência, antes de analisar o *trade-off* de custo total, é importante entender a eficiência base do RAG, conforme a QP 3. Os resultados demonstram que a introdução do RAG (*Single-Agent*) transforma a tarefa do LLM e impacta positivamente a latência.

O *Single-Agent* foi o cenário mais rápido ($\approx 13 - 14$s), superando o LLM Baseline. Esta anomalia valida a hipótese de que o RAG transforma a tarefa do LLM de um raciocínio de *recall* aberto e computacionalmente intensivo em uma síntese contextual focada. O fornecimento de informações relevantes no *prompt* simplifica o raciocínio, e a economia de tempo na fase de geração é suficiente para compensar o *overhead* da etapa de recuperação de documentos. Este resultado confirma que a síntese contextual é mais eficiente em termos de latência do que o raciocínio de *recall* puro.

Complementar, apesar da alta eficiência do *Single-Agent* demonstrada na QP 3, a QP 4 exige uma avaliação do *trade-off* operacional completo do *Multi-Agent* (qualidade vs. eficiência). Os resultados confirmam que a superioridade qualitativa do *Multi-Agent* exige um custo computacional maior. O *Multi-Agent* foi o cenário mais lento (mediana de 20-21 segundos) e apresentou o maior impacto ambiental (emissões medianas $\approx 0,22$ g). O custo não é apenas mais elevado, mas também mais variável, evidenciado pelo maior intervalo interquartil no tempo de resposta, o que reduz a previsibilidade. Percebe-se que o *Multi-Agent* oferece a melhor qualidade pelo maior custo e menor previsibilidade. O *Single-Agent* representa o ponto de maior eficiência (menor custo e maior velocidade) dentro dos cenários RAG, provendo uma qualidade intermediária que já é superior ao LLM Baseline.

Por fim, o estudo conclui examinando a confiabilidade das métricas por meio da comparação entre a avaliação sintética (*LLM-as-a-judge*) e o julgamento especializado humano (QP 5).

Os resultados demonstram que a presença de contexto explícito nos cenários RAG (*Single-Agent* e *Multi-Agent*) influencia positivamente a convergência entre as avaliações. O LLM Baseline apresentou a maior divergência, sugerindo que o *LLM-as-a-judge* tem dificuldade em validar respostas geradas puramente por conhecimento paramétrico.

A implicação dessa divergência é que o RAG não apenas melhora a resposta, mas também facilita a validação da sua acurácia. O contexto de recuperação serve como um me-

canismo de fundamentação para o próprio avaliador sintético, reduzindo a ambiguidade na detecção de inconsistências. Portanto, a confiabilidade da validação automatizada é maior em sistemas RAG onde o contexto de verificação é explicitamente fornecido.

Seção 7

Conclusão

Esta seção apresenta as conclusões do estudo realizado nesta dissertação, partindo da visão dos objetivos e da metodologia proposta, e abordando a abrangência, as limitações dos resultados e a projeção para trabalhos futuros.

O presente estudo teve como objetivo geral investigar o uso e o impacto da complexidade arquitetural em sistemas de Inteligência Artificial generativa, especificamente na recuperação e síntese de conhecimento a partir de documentação industrial legada. A análise focou na comparação entre o LLM *Baseline*, o *Single-Agent* RAG e o *Multi-Agent* RAG, com o intuito de estabelecer a relação entre a orquestração de agentes e a qualidade da informação gerada.

Para o cumprimento deste objetivo, foi estabelecida uma abordagem que envolveu a estruturação de um corpus documental técnico, a implementação dos três cenários arquiteturais e uma avaliação multifacetada, utilizando métricas quantitativas (ROUGE, BERTScore) e avaliação qualitativa com a metodologia *LLM-as-a-Judge* e validação por especialistas. Os resultados dessas atividades permitiram alcançar os objetivos específicos propostos e fornecer um conjunto de conclusões, respondendo integralmente às questões de pesquisa definidas.

Os resultados do estudo fornecem evidências para todas as questões de pesquisa levantadas, as quais estão detalhadas na Seção 1. A análise comparativa estabelece a superioridade da arquitetura de orquestração *Multi-Agent*, impactando positivamente todos

os vetores de qualidade de saída (Resposta à QP 1). O *Multi-Agent* superou os demais cenários em precisão lexical, coerência semântica e, em acurácia factual, impulsionado por um ganho substancial na precisão. Essa constatação estabelece uma nova hierarquia de qualidade para sistemas RAG.

A complexidade da arquitetura RAG demonstrou um impacto direto na confiabilidade (Resposta à QP 2). O *Multi-Agent* alcançou $\approx 80\%$ de *faithfulness* (fidelidade), o que comprova que a orquestração age como um mecanismo robusto de fundamentação, ideal para mitigar a alucinação de conteúdo que aflige o *Single-Agent*. A introdução do RAG gerou um salto na acurácia factual, e o fato de o *Single-Agent* ser mais rápido que o LLM *Baseline* valida a hipótese de que o RAG transforma a tarefa do LLM de um raciocínio de recall computacionalmente intensivo em uma síntese contextual mais eficiente (Resposta à QP 3).

Complementar, o *trade-off* operacional foi estabelecido (Resposta à QP 3). Embora o *Multi-Agent* tenha fornecido a mais alta qualidade, foi o cenário mais lento e com maior custo computacional. Em contraste, o *Single-Agent* emergiu como a solução de maior eficiência. Por fim, a confiabilidade das métricas de avaliação foi abordada (Resposta à QP 5). O contexto explícito nos cenários RAG demonstrou influenciar positivamente a convergência entre a avaliação sintética (*LLM-as-a-judge*) e o julgamento especializado humano. A maior divergência no LLM *Baseline* implica que a presença de contexto facilita a validação da acurácia e aumenta a confiabilidade das métricas automatizadas.

Ademais, os resultados encontrados também fornecem evidências para a análise das hipóteses propostas na Seção 1. Com base na performance consistente e superior do *Multi-Agent* em acurácia e fidelidade, a hipótese nula (H_0) é refutada em favor da hipótese alternativa (H_1), a qual determina que a utilização de arquiteturas de orquestração *Multi-Agent* resulta em melhorias na precisão factual, coerência semântica e na mitigação de alucinações.

Assim, a confirmação da H_1 implica que, em domínios técnicos com baixa tolerância

a erros, a complexidade arquitetural se justifica e é obrigatória pelo ganho em segurança e acurácia. A especialização e a colaboração entre agentes garantem que o conteúdo gerado não só seja relevante, mas que a sua origem e fundamentação (*faithfulness*) sejam verificadas durante o processo de geração. Assim, a metodologia proposta e validada neste trabalho contribui, fornecendo um *benchmark* sobre os *trade-offs* de custo e qualidade em sistemas Agentic-RAG.

7.1 Trabalhos Futuros

Nesta seção, são apresentadas algumas sugestões de trabalhos futuros que complementam a pesquisa realizada nesta dissertação, mas não estão limitados a:

- Generalização das Descobertas: testar a robustez dos resultados, replicando o experimento com diferentes LLMs fundamentais (*foundation models*) e aplicando a metodologia a outros domínios técnicos especializados, a fim de validar a generalização dos *trade-offs* observados.

- Otimização da Eficiência Arquitetural: explorar melhorias no sistema *Multi-Agent* para reduzir a latência e a variabilidade, incluindo a aplicação de Small Language Models (SLMs) para as tarefas de orquestração.

- Avanço nas Técnicas RAG: incorporar e avaliar o impacto de técnicas avançadas de RAG no processo, visando aprimorar a busca e a síntese de respostas nos cenários *Single-Agent* e *Multi-Agent*.

- Expansão do *Framework Multi-Agent*: estender a arquitetura através da inclusão de novos agentes especialistas dedicados, um agente especializado na decodificação e análise de pacotes de rede, para lidar com tarefas de engenharia ainda mais específicas.

Referências Bibliográficas

Acharya, Deepak Bhaskar, Karthigeyan Kuppan & B. Divya (2025), 'Agentic AI: Autonomous intelligence for complex goals—a comprehensive survey', *IEEE Access* **13**, 18912–18936.

Adewale, David, Iliyasu Kayode Okediran, Musibaudeen Olatunde Idris, Abideen Temitayo Oyewo & Abdulhafiz Ademola Adefajo (2025), 'Application in product labeling: Benefits, challenges, key components and technologies: A review', *Borobudur Engineering Review* **5**(01), 01–33.

Ahmad, Ijaz, Felipe Rodriguez, Tanesh Kumar, Jani Suomalainen, Senthil Kumar Jagatheesaperumal, Stefan Walter, Muhammad Zeeshan Asghar, Gaolei Li, Nikolaos Papakonstantinou, Mika Ylianttila, Jyrki Huusko, Thilo Sauter & Erkki Harjula (2024), 'Communications security in Industry X: A survey', *IEEE Open Journal of the Communications Society* **5**, 982–1025.

AlMadhoun, Ashraf Said (2025), Communication protocols, *em* 'PLC SCADA for Beginners: Understanding and Implementing Industrial Automation Systems', Springer, pp. 149–205.

Bhole, Mukund, Thilo Sauter, Sabrina Semper & Wolfgang Kastner (2025), 'Why to fail fast and often: A strategy for ot safety and security evaluation', *IEEE Access* **13**, 51793–51812.

Biancofiore, Giovanni Maria, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio & Fedelucio Narducci (2024), 'Interactive question answering systems: Literature review', *ACM Comput. Surv.* **56**(9).

Budakoglu, Gülsüm & Hakan Emekci (2025), 'Unveiling the power of large language models: A comparative study of retrieval-augmented generation, fine-tuning, and their synergistic fusion for enhanced performance', *IEEE Access* **13**, 30936–30951.

Buri, Zsolt & Judit T. Kiss (2025), 'Digitalisation in the context of industry 4.0 and industry 5.0: A bibliometric literature review and visualisation', *Applied System Innovation* **8**(5), 137.

Cai, Jing, Alex E Hadjinicolaou, Angelique C Paulk, Daniel J Soper, Tian Xia, Alexander F Wang, John D Rolston, R Mark Richardson, Ziv M Williams & Sydney S Cash (2025), 'Natural language processing models reveal neural dynamics of human conversation', *Nature Communications* **16**(1), 3376.

Cai, Jinyu, Jialong Li, Mingyue Zhang, Munan Li, Chen-Shu Wang & Kenji Tei (2024), Language Evolution for Evading Social Media Regulation via LLM-Based Multi-Agent Simulation, *em* '2024 IEEE Congress on Evolutionary Computation (CEC)', pp. 1–10.

Cavalcanti Hernandes, Leonardo, Anderson Luis Szejka & Fernando Mas (2025), 'Intelligent product manufacturing cost estimation framework driven by semantic technologies and knowledge-based systems', *International Journal of Computer Integrated Manufacturing* pp. 1–22.

Chatoui, Hajar & Oğuz Ata (2021), Automated evaluation of the virtual assistant in bleu and rouge scores, *em* '2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)', pp. 1–6.

Chen, Jiawei, Hongyu Lin, Xianpei Han & Le Sun (2024), 'Benchmarking large language models in retrieval-augmented generation', *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(16), 17754–17762.

Crupi, Giuseppe, Rosalia Tufano, Alejandro Velasco, Antonio Mastropaolo, Denys Poshyvanyk & Gabriele Bavota (2025), 'On the Effectiveness of LLM-as-a-Judge for Code Generation and Summarization', *IEEE Transactions on Software Engineering* **51**(8), 2329–2345.

Dagli, Mert Marcel, Yohannes Ghenbot, Hasan S Ahmad, Daksh Chauhan, Ryan Turlip, Patrick Wang, William C Welch, Ali K Ozturk & Jang W Yoon (2024), 'Development and validation of a novel AI framework using NLP with LLM integration for relevant clinical data extraction through automated chart review', *Scientific reports* **14**(1), 26783.

de Moura, Ralf Luis, Tiago Tadeu Wirtti, Filipe Andersonn Teixeira da Silveira, Rodrigo Rosetti Binda & Brenda Aurora Pires Moura (2024), Strategies for protecting serial (non-IP) industrial networks in cyber-physical systems 2.0, *em* 'Cyber Physical System 2.0', CRC Press, pp. 253–281.

De P. Paiva, João Pedro Magalhães, Lucas Andrade Marques, Lucas Faria Teixeira, Jônatas Alves Sena, Alexandre Baratella Lugli, Egidio Raimundo Neto, João Paulo Carvalho Henriques & Tales Cleber Pimenta (2025), Use case comparison between profibus pa and profinet pa networks, *em* '2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)', Vol. 1, pp. 1–5.

Du, Hongyang, Ruichen Zhang, Yinqiu Liu, Jiacheng Wang, Yijing Lin, Zonghang Li, Dusit Niyato, Jiawen Kang, Zehui Xiong, Shuguang Cui, Bo Ai, Haibo Zhou & Dong In Kim (2024), 'Enhancing deep reinforcement learning: A tutorial on ge-

nerative diffusion models in network optimization', *IEEE Communications Surveys Tutorials* **26**(4), 2611–2646.

Esposito, Massimo, Emanuele Damiano, Aniello Minutolo, Giuseppe De Pietro & Hamido Fujita (2020), 'Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering', *Information Sciences* **514**, 88–105.

Gallifant, Jack, Majid Afshar, Saleem Ameen, Yindalon Aphinyanaphongs, Shan Chen, Giovanni Cacciamani, Dina Demner-Fushman, Dmitriy Dligach, Roxana Daneshjou, Chrystinne Fernandes et al. (2025), 'The TRIPOD-LLM reporting guideline for studies using large language models', *Nature medicine* **31**(1), 60–69.

Gardazi, Nadia Mushtaq, Ali Daud, Muhammad Kamran Malik, Amal Bukhari, Tariq Alsahfi & Bader Alshemaimri (2025), 'BERT applications in natural language processing: a review', *Artificial Intelligence Review* **58**(6), 1–49.

Harrer, Stefan (2023), 'Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine', *eBioMedicine* **90**.

Hollerer, Siegfried, Thilo Sauter & Wolfgang Kastner (2024), 'A survey of ontologies considering general safety, security, and operation aspects in ot', *IEEE Open Journal of the Industrial Electronics Society* **5**, 861–885.

James, Antony, Marcello Trovati & Simon Bolton (2025), 'Retrieval-augmented generation to generate knowledge assets and creation of action drivers', *Applied Sciences* **15**(11).

Kim, Sehoon, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer & Amir Gholami (2024), An LLM compiler for parallel function calling, *em* 'Forty-first International Conference on Machine Learning'.

Korodi, Adrian, Vlad-Cristian Vesa & Raul-Andrei Dontu (2025), Efficient and human centered industry 5.0 data propagation on the operational technology level. case study with opc ua interfacing, node-red and ignition, *em* '2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)', IEEE, pp. 292–297.

LangChain (2025), 'Langchain documentation'.
URL: *https://python.langchain.com/*

Li, Xinyi, Sai Wang, Siqi Zeng, Yu Wu & Yi Yang (2024), 'A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges', *Vicinagearth* **1**(1), 9.

Liu, Siru, Allison B McCoy & Adam Wright (2025), 'Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines', *Journal of the American Medical Informatics Association* **32**(4), 605–615.

Medeiros, Thaís, Morsinaldo Medeiros, Mariana Azevedo, Marianne Silva, Ivanovitch Silva & Daniel G Costa (2023), 'Analysis of language-model-powered Chatbots for query resolution in PDF-based automotive manuals', *Vehicles* **5**(4), 1384–1399.

Medeiros, Thaís, Morsinaldo Medeiros, Matheus Andrade, Marianne Silva, Ivanovitch Silva, Massimiliano Gaffurini, Dennis Brandão & Paolo Ferrari (2025), Tailoring rag strategies for industrial protocols: A comparative study on profibus document retrieval using gemma and gpt models, *em* '2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)', pp. 1–8.

Nassiri, Khalid & Moulay Akhloufi (2023), 'Transformer models used for text-based question answering systems', *Applied Intelligence* **53**(9), 10602–10635.

Ni, Bo & Markus J. Buehler (2024), 'Mechagents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge', *Extreme Mechanics Letters* **67**, 102131.

Oliveira, Gisliany Lillian Alves de, Breno Santana Santos, Marianne Silva & Ivanovitch Silva (2025), 'Exploring legislative textual data in brazilian portuguese: Readability analysis and knowledge graph generation', *Data* **10**(7).

Omrani, Pouria, Alireza Hosseini, Kiana Hooshanfar, Zahra Ebrahimian, Ramin Toosi & Mohammad Ali Akhaee (2024), Hybrid retrieval-augmented generation approach for llms query response enhancement, *em* '2024 10th International Conference on Web Research (ICWR)', pp. 22–26.

Paiva, João Pedro Magalhães De P, Lucas Andrade Marques, Lucas Faria Teixeira, Jônatas Alves Sena, Alexandre Baratella Lugli, Egidio Raimundo Neto, João Paulo Carvalho Henriques & Tales Cleber Pimenta (2025), Use Case Comparison Between Profibus PA and Profinet PA Networks, *em* '2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)', Vol. 1, IEEE, pp. 1–5.

Pan, James Jie, Jianguo Wang & Guoliang Li (2024), 'Survey of vector database management systems', *The VLDB Journal* **33**(5), 1591–1615.

Pan, Shirui, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang & Xindong Wu (2024), 'Unifying large language models and knowledge graphs: A roadmap', *IEEE Transactions on Knowledge and Data Engineering* **36**(7), 3580–3599.

Peng, Ziming, Xi Kuai, Shuisong Ke, Xuehui Dong & Renzhong Guo (2025), 'Enhancing geodatabases operability: advanced human-computer interaction through rag and multi-agent systems', *Big Earth Data* **9**(2), 217–242.

Roy, Devjeet, Xuchao Zhang, Rashi Bhave, Chetan Bansal, Pedro Las-Casas, Rodrigo Fonseca & Saravan Rajmohan (2024), Exploring LLM-Based Agents for Root

Cause Analysis, *em* 'Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering', FSE 2024, Association for Computing Machinery, New York, NY, USA, p. 208–219.

Roy, Sujoy, Shane Morrell, Lili Zhao & Ramin Homayouni (2024), 'Large-scale identification of social and behavioral determinants of health from clinical notes: comparison of latent semantic indexing and generative pretrained transformer (gpt) models', *BMC Medical Informatics and Decision Making* **24**(1), 296.

Saha, Binita, Utsha Saha & Muhammad Zubair Malik (2024), 'QuIM-RAG: Advancing Retrieval-Augmented Generation With Inverted Question Matching for Enhanced QA Performance', *IEEE Access* **12**, 185401–185410.

Salemi, Alireza & Hamed Zamani (2024), Evaluating retrieval quality in retrieval-augmented generation, *em* 'Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '24, Association for Computing Machinery, New York, NY, USA, p. 2395–2400.

Sestito, Guilherme Serpa, Afonso Celso Turcato, Andre Luis Dias, Paolo Ferrari & Maíra Martins da Silva (2022), 'Versatile unsupervised anomaly detection method for rte-based networks', *Expert Systems with Applications* **206**, 117751.

Singh, Abhai Pratap, Adit Jamdar & Prerna Kaul (2025), 'Agentic Retrieval-Augmented Generation: Advancing AI-Driven Information Retrieval and Processing', *International Journal of Computer Trends and Technology (IJCTT)* **73**(1), 91–97.

Singh, Ajit (2024), 'Agentic RAG Systems for Improving Adaptability and Performance in AI-Driven Information Retrieval', *SSRN Electronic Journal* . Available at SSRN: https://ssrn.com/abstract=5188363.

Sisinni, Emiliano, Abusayeed Saifullah, Song Han, Ulf Jennehag & Mikael Gidlund (2018), 'Industrial internet of things: Challenges, opportunities, and directions', *IEEE Transactions on Industrial Informatics* **14**(11), 4724–4734.

Sollenberger, Zachariah, Jay Patel, Christian Munley, Aaron Jarmusch & Sunita Chandrasekaran (2024), LLM4VV: Exploring LLM-as-a-Judge for Validation and Verification Testsuites, *em* 'SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis', pp. 1885–1893.

Stripelis, Dimitris, Zhaozhuo Xu, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Jipeng Zhang, Tong Zhang, Salman Avestimehr & Chaoyang He (2024), TensorOpera router: A multi-model router for efficient LLM inference, *em* F.Dernoncourt, D.Preoţiuc-Pietro & A.Shimorina, eds., 'Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track', Association for Computational Linguistics, Miami, Florida, US, pp. 452–462.

Taipalus, Toni (2024), 'Vector database management systems: Fundamental concepts, use-cases, and current challenges', *Cognitive Systems Research* **85**, 101216.

Tao, Wei, Bin Zhang, Xiaoyang Qu, Jiguang Wan & Jianzong Wang (2025), Cocktail: Chunk-Adaptive Mixed-Precision Quantization for Long-Context LLM Inference, *em* '2025 Design, Automation & Test in Europe Conference (DATE)', IEEE, pp. 1–7.

Vasaniya, Raj, Meet Visodiya & Anand K. Patel (2025), TechAssist: A RAG-Based Chatbot for Accessing Technical Information from StackOverflow, *em* '2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)', pp. 1–6.

Vitturi, Stefano, Claudio Zunino & Thilo Sauter (2019), 'Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G', *Proceedings of the IEEE* **107**(6), 944–961.

Wahidur, Rahman S. M., Sumin Kim, Haeung Choi, David S. Bhatti & Heung-No Lee (2025), 'Legal Query RAG', *IEEE Access* **13**, 36978–36994.

Wollschlaeger, Martin, Thilo Sauter & Juergen Jasperneite (2017), 'The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0', *IEEE Industrial Electronics Magazine* **11**(1), 17–27.

Wu, Dongyuan, Liming Nie, Rao Asad Mumtaz & Kadambri Agarwal (2024), 'A LLM-based hybrid-transformer diagnosis system in healthcare', *IEEE Journal of Biomedical and Health Informatics* .

Yanytska, Lesia (2025), 'The rise of human-centric manufacturing in the industry 5.0 era', *The International Journal of Advanced Manufacturing Technology* pp. 1–11.

Ye, Zhenhong (2025), Intelligent Tutoring Agent with Retrieval-Augmented Generation: A Case Study of Quality Management System Course, *em* '2025 5th International Conference on Consumer Electronics and Computer Engineering (ICCECE)', pp. 189–194.

Zhang, Ruichen, Ke Xiong, Hongyang Du, Dusit Niyato, Jiawen Kang, Xuemin Shen & H. Vincent Poor (2024b), 'Generative AI-Enabled Vehicular Networks: Fundamentals, Framework, and Case Study', *IEEE Network* **38**(4), 259–267.

Zhang, Wei, Qinggong Wang, Xiangtai Kong, Jiacheng Xiong, Shengkun Ni, Duanhua Cao, Buying Niu, Mingan Chen, Yameng Li, Runze Zhang, Yitian Wang, Lehan Zhang, Xutong Li, Zhaoping Xiong, Qian Shi, Ziming Huang, Zunyun Fu & Mingyue Zheng (2024a), 'Fine-tuning large language models for chemical text mining', *Chem. Sci.* **15**, 10600–10611.

Zhang, Wentao, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang & Bo An (2024), A Multimodal Foundation Agent for Financial Trading: Tool-Augmented, Diversified, and Generalist, *em* 'Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining', KDD '24, Association for Computing Machinery, New York, NY, USA, p. 4314–4325.

Zhao, Yuheng, Junjie Wang, Linbing Xiang, Xiaowen Zhang, Zifei Guo, Cagatay Turkay, Yu Zhang & Siming Chen (2024), 'Lightva: Lightweight visual analytics with LLM Agent-based task planning and execution', *IEEE Transactions on Visualization and Computer Graphics* **31**(9), 6162–6177.

Zhao, Zihuai, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang & Qing Li (2024), 'Recommender Systems in the Era of Large Language Models (LLMs)', *IEEE Transactions on Knowledge and Data Engineering* **36**(11), 6889–6907.