Universidade Federal da Paraíba
Centro de Energias Alternativas e Renováveis
Graduação em Engenharia Elétrica

# Integração de agentes de IA em sistemas supervisórios no SCADA-LTS

Mateus Navarro Ribeiro de Lima

João Pessoa - PB
2026

Mateus Navarro Ribeiro de Lima

# Integração de agentes de IA em sistemas supervisórios no SCADA-LTS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Energias Alternativas e Renováveis da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Juan Moises Mauricio Villanueva

João Pessoa - PB
2026

Mateus Navarro Ribeiro de Lima

# Integração de agentes de IA em sistemas supervisórios no SCADA-LTS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Energias Alternativas e Renováveis da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Engenharia Elétrica.

gov.br
Documento assinado digitalmente
JUAN MOISES MAURICIO VILLANUEVA
Data: 04/08/2026 11:45:47-0300
Verifique em https://validar.iti.gov.br

**Prof. Dr. Juan Moises Mauricio Villanueva**
Orientador

Ademar Virgolino da S. Netto

**Prof. Dr. Ademar Virgolino da Silva Netto**

gov.br
Documento assinado digitalmente
JOSE MAURICIO RAMOS DE SOUZA NETO
Data: 07/08/2026 10:23:43-0300
Verifique em https://validar.iti.gov.br

**Prof. Dr. José Mauricio Ramos de Souza Neto**
Examinador

Fabio P. Durão

**Me. Fabio Prudente Durão**
Examinador

João Pessoa - PB
2026

**Catalogação na publicação**
**Seção de Catalogação e Classificação**

L732i Lima, Mateus Navarro Ribeiro de.
Integração de agentes de IA em sistemas supervisórios no SCADA-LTS / Mateus Navarro Ribeiro de Lima. - João Pessoa, 2026.
70 f. : il.

Orientação: Juan Moises Mauricio Villanueva.
TCC (Graduação) - UFPB/CEAR.

1. Sistemas SCADA. 2. Modelos de linguagem de grande porte. 3. Geração Aumentada por Recuperação. 4. Human-in-the-Loop. 5. Automação industrial. I. Villanueva, Juan Moises Mauricio. II. Título.

UFPB/BSCT CDU 621.3(043.2)

Elaborado por ROSANGELA GONCALVES PALMEIRA - CRB-216

*Dedico este trabalho a Germana Navarro Ribeiro de Lima e Cícero José Cavalcanti de Lima, que desde pequeno me ensinaram o valor da educação.*

# Agradecimentos

Gostaria de agradecer a Deus, que sempre intercedeu por mim durante minha vida e à minha família: Germana Navarro Ribeiro de Lima, Cícero José Cavalcanti de Lima, Thayana Maria Navarro Ribeiro de Lima Martins e Tiago Navarro Ribeiro de Lima. Obrigado por sempre me apoiarem e estarem comigo durante todo o tempo, eu não seria nada sem vocês.

Ademais, gostaria de agradecer aos meus amigos que estudaram comigo desde o ensino fundamental até o ensino médio e com quem mantenho uma amizade há mais de uma década. Sou grato a todos e espero sempre poder contar com vocês e desejo compartilhar mais momentos com cada um.

À irmã que conheci na universidade, Maria Thais, que sempre me apoiou e esteve presente durante a maior parte da minha graduação, incentivando-me a sempre melhorar tanto como pessoa quanto como universitário e também me ajudando em momentos difíceis da Universidade e da minha vida pessoal, muito obrigado.

A um irmão que conheci na universidade, Carlos Henrique, muito obrigado. Sua disciplina e seu foco incentivaram-me a sempre dar o meu máximo nas disciplinas, algo que só consegui dividindo esse peso com você, seja em provas, trabalhos ou atividades. Obrigado também por sempre estar disponível pra me escutar nos momentos difíceis.

Também agradeço a João Carlos, Gustavo Adelino, Pedro Marcelo, Patricio Valentin e Gabriel Oliveira, que me ajudaram em diferentes momentos da minha graduação. Valorizo muito a amizade que tenho com cada um de vocês.

Gostaria de agradecer também ao professor Juan Maurício pela paciência e dedicação desde meu período no PIBIT até o presente, bem como pelas oportunidades disponibilizadas e por todo o auxílio prestado durante meus últimos períodos na universidade. Além disso, agradeço aos professores Ademar Virgolino e José Maurício, que fazem parte da banca avaliadora deste trabalho e tiveram participação na minha graduação em projetos como o PET Elétrica e o PROBEM. Sempre presentes, eles me auxiliaram com orientações e conselhos que me ajudaram a chegar até aqui.

Por fim, a todos os amigos que conheci durante a minha graduação e que fizeram parte da minha vida, obrigado.

# Resumo

Os sistemas SCADA permitem que operadores acompanhem variáveis, alarmes e históricos relacionados à operação de uma planta. Entretanto, a interpretação dessas informações pode exigir a consulta a diferentes manuais, registros e documentos técnicos. Nesse contexto, este trabalho teve como objetivo desenvolver e avaliar uma aplicação de suporte à decisão que integra o SCADA-LTS de uma bancada hidráulica do LENHS-UFPB a um agente baseado em modelo de linguagem. Para isso, foi desenvolvida uma arquitetura composta por uma interface operacional, um *middleware* em FastAPI, ferramentas para consulta do estado atual e do histórico da planta, um mecanismo de Geração Aumentada por Recuperação (RAG) e uma etapa de aprovação humana para ações de escrita. A avaliação foi realizada por meio de cinco Cenários Operacionais de Teste, comparando respostas geradas sem e com o uso do RAG, mantendo o mesmo modelo, as mesmas instruções e as demais ferramentas disponíveis ao agente. Os resultados mostraram que o RAG não produziu uma melhoria uniforme em todos os cenários. Em perguntas gerais, o contexto permanente da planta foi suficiente para gerar respostas adequadas. Entretanto, quando a pergunta dependia de uma informação específica da documentação, o RAG apresentou um ganho factual e acrescentou proveniência à resposta. Nas execuções realizadas, a latência média observada foi 26,5% maior com RAG, que também não eliminou extrapolações sem fundamentação. O mecanismo *Human-in-the-Loop* permitiu submeter propostas de escrita à decisão do operador, registrando fluxos de aprovação e recusa. Dessa forma, a arquitetura mostrou-se adequada como ferramenta de apoio ao operador, desde que as respostas possam ser verificadas e que as ações sobre a planta permaneçam condicionadas à supervisão humana.

**Palavras-chave**: Sistemas SCADA. Modelos de linguagem de grande porte. Geração Aumentada por Recuperação. *Human-in-the-Loop*. Automação industrial.

# Abstract

SCADA systems allow operators to monitor variables, alarms, and historical data related to plant operation. However, interpreting this information may require consulting different manuals, records, and technical documents. In this context, this study aimed to develop and evaluate a decision-support application that integrates the SCADA-LTS system of a hydraulic test bench at LENHS-UFPB with a language model-based agent. To this end, an architecture was developed comprising an operator interface, a FastAPI middleware, tools for querying the current state and historical data of the plant, a retrieval-augmented generation (RAG) mechanism, and a human approval stage for write actions. The evaluation was conducted through five Operational Test Scenarios by comparing responses generated without and with RAG while maintaining the same model, instructions, and other tools available to the agent. The results showed that RAG did not provide uniform improvement across all scenarios. For general questions, the permanent plant context was sufficient to generate adequate responses. However, when a question depended on specific information from the documentation, RAG provided a factual gain and added provenance to the response. In the executions performed, the observed average latency was 26.5% higher with RAG, which also did not eliminate unsupported extrapolations. The Human-in-the-Loop mechanism allowed write proposals to be submitted for the operator's decision, recording both approval and rejection workflows. Thus, the architecture proved suitable as an operator-support tool, provided that the responses remain verifiable and actions affecting the plant remain subject to human supervision.

**Keywords**: SCADA systems. Large language models. Retrieval-augmented generation. Human-in-the-Loop. Industrial automation.

# Lista de tabelas

Tabela 1 – Parâmetros considerados para os componentes físicos da bancada. . . . 33
Tabela 2 – Meios de comunicação entre as áreas da interface e o *backend*. . . . . 38
Tabela 3 – Áreas e finalidades da interface da aplicação. . . . . 40
Tabela 4 – Documentos que compõem o corpus utilizado na avaliação do RAG. . . 46
Tabela 5 – Cenários de teste para avaliação do RAG local. . . . . 49
Tabela 6 – Escalas utilizadas na avaliação humana das respostas e dos documentos recuperados. . . . . 50
Tabela 7 – Síntese da consulta ao histórico operacional do PT2. . . . . 57
Tabela 8 – Quantidade de páginas dos documentos antes e depois da curadoria. . . 59
Tabela 9 – Síntese qualitativa da bateria final de Cenários Operacionais de Teste. 61
Tabela 10 – Tempo de execução do agente e consumo de *tokens* na bateria final sem e com RAG. . . . . 63
Tabela 11 – Comparação da qualidade das respostas sem e com RAG. . . . . 63
Tabela 12 – Avaliação da qualidade dos documentos recuperados na condição com RAG. . . . . 63

# Lista de ilustrações


Figura 1 – Arquitetura básica de um sistema SCADA. . . . . 20
Figura 2 – Arquitetura funcional conceitual do SCADA-LTS. . . . . 22
Figura 3 – Integração conceitual de uma aplicação externa com o SCADA-LTS por meio de uma API Web. . . . . 23
Figura 4 – Associações de duas bombas idênticas em série e em paralelo e respectivas curvas equivalentes. . . . . 25
Figura 5 – Representação simplificada da geração autorregressiva do próximo *token* a partir do contexto de entrada. . . . . 26
Figura 6 – Exemplo de agentes baseados na mesma LLM e configurados com papéis especializados. . . . . 28
Figura 7 – Fluxo simplificado de preparação, recuperação e geração de uma resposta em um sistema RAG. . . . . 30
Figura 8 – Vistas anotadas da bancada experimental Série-Paralelo do LENHS-UFPB. . . . . 33
Figura 9 – Topologia de instrumentação, controle e supervisão da configuração atual da bancada Série-Paralelo. . . . . 34
Figura 10 – Tela principal de controle e monitoramento do supervisório Série-Paralelo no SCADA-LTS. . . . . 35
Figura 11 – Arquitetura de implantação da aplicação de integração SCADA–Agente. 37
Figura 12 – Comunicação entre os componentes da aplicação. . . . . 42
Figura 13 – Organização funcional do agente e dos seus mecanismos de consulta e atuação supervisionada. . . . . 43
Figura 14 – Composição da entrada enviada ao modelo de linguagem pelo agente. . 44
Figura 15 – Fluxo do protocolo de avaliação das respostas com e sem recuperação documental. . . . . 51
Figura 16 – Correspondência entre as camadas da arquitetura e as evidências da aplicação implementada. . . . . 52
Figura 17 – Dashboard da aplicação SCADA *Agent* com indicadores de conexão e cartões de monitoramento em tempo real. . . . . 53
Figura 18 – Interface conversacional da aplicação SCADA Agent em interações sem e com recuperação documental. . . . . 54
Figura 19 – Tela de Configurações com o catálogo do corpus autorizado e a descrição da topologia operacional. . . . . 55
Figura 20 – Consulta ao estado atual da bancada com identificação da configuração em paralelo a partir da telemetria e da topologia operacional. . . . . 56

Figura 21 – Consulta ao histórico do PT2 com período efetivamente coberto, número de amostras e estatísticas agregadas. . . . . 57
Figura 22 – Proposta de acionamento de M1 apresentada pelo agente e condicionada à aprovação humana explícita. . . . . 58

# Sumário


**1 INTRODUÇÃO . . . . . 13**
**1.1 Objetivo Geral . . . . . 15**
1.1.1 Objetivos Específicos . . . . . 15
**1.2 Organização do Trabalho . . . . . 16**

**2 REVISÃO BIBLIOGRÁFICA . . . . . 17**

**3 REFERENCIAL TEÓRICO . . . . . 20**
**3.1 Sistemas SCADA e SCADA-LTS . . . . . 20**
3.1.1 Arquitetura e funções dos sistemas SCADA . . . . . 20
3.1.2 SCADA-LTS em ambientes industriais e acadêmicos . . . . . 21
3.1.3 Integração entre aplicações por APIs Web . . . . . 22
**3.2 Sistemas Hidráulicos de Bombeamento . . . . . 24**
3.2.1 Grandezas hidráulicas fundamentais . . . . . 24
3.2.2 Associações de bombas em série e em paralelo . . . . . 25
**3.3 Modelos de Linguagem e Agentes Inteligentes . . . . . 26**
3.3.1 Agentes baseados em modelos de linguagem e uso de ferramentas . . 27
**3.4 Geração Aumentada por Recuperação e Recuperação Vetorial . . . . 29**

**4 METODOLOGIA E DESENVOLVIMENTO . . . . . 32**
**4.1 Caracterização da Bancada Física do LENHS-UFPB . . . . . 32**
**4.2 Arquitetura Geral da Aplicação . . . . . 36**
4.2.1 Camada 1 — Integração supervisória . . . . . 37
4.2.2 Camada 2 — Orquestração da aplicação . . . . . 38
4.2.3 Camada 3 — Serviços especializados . . . . . 39
4.2.4 Camada 4 — Acesso e supervisão humana . . . . . 41
4.2.5 Implantação e comunicação entre as camadas . . . . . 41
**4.3 O Agente Inteligente Baseado em Agno . . . . . 42**
4.3.1 Estrutura e configuração do agente . . . . . 43
4.3.2 Fontes de contexto e ferramentas operacionais . . . . . 44
4.3.3 Recuperação documental agentiva . . . . . 45
**4.4 Metodologia de Ensaio e Validação do Sistema . . . . . 45**
4.4.1 Formação e curadoria do corpus documental . . . . . 46
4.4.2 Avaliação do RAG Local por Cenários . . . . . 48

**5 RESULTADOS E DISCUSSÕES . . . . . 52**

**5.1 Apresentação da Aplicação Desenvolvida . . . 52**
5.1.1 Dashboard de monitoramento . . . 52
5.1.2 Interface conversacional do agente . . . 53
5.1.3 Visualização e controle da base documental . . . 54
**5.2 Resultados da Integração Operacional e Uso de Ferramentas . . . 55**
5.2.1 Consulta do estado atual da bancada . . . 56
5.2.2 Consulta ao histórico operacional . . . 57
5.2.3 Proposta de escrita supervisionada . . . 58
**5.3 Formação e Recuperação do Corpus Técnico . . . 59**
**5.4 Avaliação do RAG por Cenários . . . 60**
5.4.1 Resultados dos Cenários Operacionais de Teste . . . 60

**6 CONCLUSÕES E TRABALHOS FUTUROS . . . 65**
**6.1 Síntese das Atividades Desenvolvidas . . . 65**
**6.2 Principais Contribuições do Trabalho . . . 65**
**6.3 Limitações do Estudo . . . 66**
**6.4 Trabalhos Futuros . . . 66**
6.4.1 Predição de Variáveis Futuras via LLM vs. Redes Neurais . . . 66
6.4.2 Implementação e Validação do Controle PI Físico . . . 67

**REFERÊNCIAS . . . 68**

# 1 Introdução

Com o avanço da Indústria 4.0, os sistemas de controle supervisório e aquisição de dados (SCADA, do inglês *Supervisory Control and Data Acquisition*) expandiram significativamente seu potencial, impulsionados pelo aumento exponencial de dados provenientes do chão de fábrica, integração de conexões sem fio, maior capacidade de armazenamento, sistemas de cibersegurança, Internet das Coisas (IoT, *Internet of Things*) e computação em nuvem (SVERKO; GRBAC; MIKUC, 2022). Consequentemente, surge um desafio substancial para o operador, que precisa monitorar e controlar variáveis em ecossistemas cada vez mais complexos e multivariáveis. Os sistemas SCADA têm um papel fundamental no ambiente industrial contemporâneo, fornecendo informações críticas para que os operadores da sala de controle tomem decisões oportunas, garantindo a operação segura e eficaz da planta (WANG et al., 2016).

Nesse contexto, esses supervisórios têm sido aplicados de forma relevante em sistemas de saneamento e distribuição de água, devido à facilidade de controle e leitura de sensores de pressão e vazão, bem como no acionamento de válvulas eletromagnéticas e conjuntos motobomba (GÖRKEM; KARAMAN; HAYBER, 2024). Especificamente na otimização energética e hidráulica, a compreensão e a simulação de arranjos de motobombas são essenciais: a associação em série é fundamental para vencer grandes desníveis topográficos através do ganho de pressão (altura manométrica), enquanto a associação em paralelo atende a demandas variáveis de consumo por meio do incremento de vazão (FLORES, 2018). Contudo, para que a operação desses arranjos seja eficiente e a resposta a anomalias seja rápida, o desenvolvimento de interfaces supervisórias claras e representativas do sistema físico real torna-se indispensável, pois a organização das informações apresentadas ao operador influencia diretamente a tomada de decisão em ambientes industriais.

Atualmente, essa problemática é abordada principalmente por meio dos recursos usuais dos sistemas SCADA, como telas sinóticas, alarmes e históricos das variáveis do processo, que centralizam as informações necessárias à análise do operador (WANG et al., 2016). Em paralelo, pesquisas recentes têm incorporado modelos de linguagem e arquiteturas de agentes para converter solicitações feitas por operadores em linguagem natural no intuito de fazer consultas, recuperar informações técnicas, detectar anomalias e apoiar a tomada de decisão (DALZELL et al., 2024; RUSSELL-GILBERT et al., 2025; KOZIOLEK et al., 2024; ARAZZI et al., 2025). Quando essas abordagens se aproximam da atuação sobre o sistema físico, mecanismos de correção, validação e supervisão humana são utilizados para reduzir o risco de comandos inadequados (XIA et al., 2025; FAN et al., 2025; HELLERT; MONTENEGRO; SULC, 2026). Porém, os estudos analisados tratam essas capacidades principalmente de forma isolada e as avaliam com dados históricos ou

simulações, permanecendo o desafio de integrá-las em uma mesma solução vinculada ao sistema SCADA, com supervisão humana.

O presente trabalho tem como principal alvo de estudo o Laboratório de Eficiência Energética e Hidráulica em Saneamento (LENHS) presente na Universidade Federal da Paraíba. O laboratório possui diversas atribuições e atua nos eixos de ensino, pesquisa e extensão, com enfoque principal no uso eficiente de energia e água no saneamento. Além disso, o laboratório busca ser uma referência principal para as companhias de saneamento do Brasil e possui uma parceria direta com a Companhia de Água e Esgotos da Paraíba (CAGEPA).

No contexto do LENHS, a plataforma de código aberto SCADA-LTS é utilizada como ferramenta de supervisão para representar variáveis, comandos e estados operacionais de sistemas experimentais. A plataforma permite configurar fontes de dados, integrar variáveis das plantas, gerenciar telas gráficas, alarmes e visualizar históricos de valores/operações. Por ser um *software* gratuito e otimizado para diferentes sistemas operacionais, a ferramenta se torna adequada para aplicações acadêmicas e industriais, principalmente para soluções que envolvem o desenvolvimento de sistemas supervisórios sem depender de licenças (UDDIN; BAIG; IQBAL, 2022). Nessa linha, o laboratório possui dois supervisórios principais, denominados Aquário e Série-Paralelo, sendo este último associado à bancada estudada nesse trabalho.

Apesar da versatilidade dos sistemas SCADA no monitoramento e controle de plantas industriais, sua utilização ainda exige capacidade técnica para interpretar sinais, alarmes e variáveis correlacionadas, identificando as informações relevantes para a condição atual da planta (WANG et al., 2016). Essa dificuldade também afeta a capacitação de novos operadores, que precisam consultar manuais, procedimentos e descrições da bancada para compreender seu contexto, suas rotinas e seus aspectos operacionais.

Desse modo, o auxílio de um modelo de linguagem de grande porte (LLM, do inglês *Large Language Model*), apoiado por uma ferramenta de Geração Aumentada por Recuperação (RAG, do inglês *Retrieval-Augmented Generation*) pode facilitar o acesso à documentação técnica. Nessa abordagem, trechos pertinentes de uma base documental são recuperados a partir da pergunta do usuário e fornecidos ao modelo como contexto para a geração da resposta. Desse modo, a ferramenta pode facilitar a consulta à documentação técnica por operadores, tanto durante a operação da planta quanto em sua capacitação, ao apresentar essas informações em linguagem natural (DALZELL et al., 2024; FUKUI et al., 2025).

Contudo, a inserção de um agente baseado em modelo de linguagem possui ressalvas, já que os modelos podem gerar respostas aparentemente plausíveis, mas não sustentadas pelo contexto ou pelos fatos verificáveis, fenômeno conhecido como alucinação (HUANG et al., 2025). Como forma de mitigar esse problema, o RAG auxilia na redução de falhas

associadas à ausência de conhecimento específico durante a geração da resposta pelo agente. Entretanto, a confiabilidade da resposta também depende da qualidade e da pertinência dos documentos recuperados. Por isso, o uso do RAG não elimina o risco de respostas inadequadas em uma planta física (FAN et al., 2024).

Por conta disso, apesar dos avanços recentes, as LLMs devem atuar, no contexto industrial, como apoio à decisão do operador, e não como controladores autônomos. Nessa linha, o agente pode realizar leituras, consultar o histórico de variáveis e recuperar contexto técnico por meio do RAG para fornecer suporte informacional. Entretanto, toda solicitação de escrita no supervisório deve ser apresentada explicitamente ao operador, que poderá aprová-la ou recusá-la antes da execução do comando. Esse mecanismo caracteriza o *Human-in-the-Loop* e preserva a responsabilidade humana sobre as ações que podem afetar a planta (BHATTACHARYA et al., 2023).

Diante desse cenário, este trabalho busca responder à seguinte questão de pesquisa: como integrar um agente baseado em modelo de linguagem, com recuperação de contexto técnico por RAG, ao supervisório da bancada hidráulica experimental Série-Paralelo, implementado em SCADA-LTS no LENHS, para oferecer suporte operacional e auxiliar na tomada de decisão, preservando a supervisão humana nas ações de escrita e os limites de segurança da planta?

## 1.1 Objetivo Geral

Desenvolver e avaliar uma aplicação de suporte supervisionado à decisão operacional que integre o SCADA-LTS da bancada hidráulica do LENHS a uma interface com agente baseado em modelo de linguagem, com recurso de RAG local e histórico de dados, além de ferramentas de leitura e escrita para variáveis do sistema, com aprovação humana para ações de escrita.

### 1.1.1 Objetivos Específicos

- Desenvolvimento de uma aplicação integrada ao SCADA-LTS da bancada hidráulica do LENHS, para reunir os dados e os recursos necessários à supervisão do sistema em uma única interface.
- Implementação de um agente baseado em modelo de linguagem, com acesso a ferramentas operacionais e à recuperação de documentação técnica por RAG, para auxiliar o operador na interpretação das informações e na tomada de decisão.
- Avaliação da solução proposta por meio de cinco casos de teste relacionados à operação da bancada, para analisar a contribuição do RAG, o desempenho da aplicação e as limitações observadas.

## 1.2 Organização do Trabalho

Esse trabalho é estruturado com o Capítulo 2 sendo uma revisão bibliográfica dos trabalhos recentes na área de estudo. Em sequência, o Capítulo 3 apresenta o referencial teórico sobre sistemas SCADA e de bombeamento, modelos de linguagem, agentes e RAG. O Capítulo 4 descreve a metodologia e o desenvolvimento da solução, incluindo a arquitetura de integração e os cenários de validação. O Capítulo 5 apresenta os resultados e as discussões dos ensaios realizados. Por fim, o Capítulo 6 reúne as conclusões do trabalho, suas limitações e propostas para trabalhos futuros.

# 2 Revisão Bibliográfica

Este capítulo cita trabalhos recentes no contexto do uso de agentes no meio industrial, abordado na introdução, com a revisão organizada nos eixos de modelos de linguagem em sistemas SCADA, recuperação documental, agentes industriais e segurança operacional.

Dalzell et al. (2024) propuseram uma abordagem baseada em modelo de linguagem para apoiar operadores no gerenciamento de alarmes de redes elétricas supervisionadas por sistemas SCADA. A solução utilizou o modelo Llama 3 70B para extrair entidades e relações de registros de alarmes, estruturar essas informações em um grafo de conhecimento no Neo4j e converter perguntas escritas em linguagem natural para consultas no grafo de conhecimento. Para isso, foi desenvolvida uma interface conversacional com o assistente para utilização de um operador. Em sequência, nos estudos de caso, o sistema respondeu corretamente a questões sobre os dispositivos e suas conexões representadas na base de conhecimento. Embora os resultados demonstrem o potencial das LLMs para facilitar o acesso a grandes volumes de dados de alarmes, a avaliação foi em sua maioria qualitativa e realizada com um conjunto de dados, sem integração operacional com uma planta.

Xia et al. (2025) desenvolveram uma arquitetura multiagente baseada em modelos de linguagem para o planejamento e o controle de um sistema físico de automação industrial. A solução integrou os equipamentos por meio de gêmeos digitais e interfaces, empregando um agente gerenciador para decompor requisições dos usuários feitas em linguagem natural, agentes operadores para executar chamadas de funções nos módulos de automação e um agente de sumarização para apresentar os eventos do processo ao usuário. Na avaliação com 100 casos de teste, o GPT-4o sem ajuste apresentou 81% de acerto na geração dos comandos, enquanto a segunda versão submetida a ajuste supervisionado atingiu 100% no conjunto avaliado. Apesar desse trabalho demonstrar a execução de operações físicas a partir de comandos em linguagem natural, os próprios autores concluem que taxas inferiores a 100% não são confiáveis para um controle direto por agentes. Além disso, a solução não utilizou um mecanismo de recuperação documental.

Hellert, Montenegro e Sulc (2026) desenvolveram o Osprey, uma estrutura para utilização de agentes baseados em modelos de linguagem em sistemas de controle críticos. A arquitetura adota um orquestrador que transforma solicitações em linguagem natural em planos de execução explícitos, um mecanismo de seleção das ferramentas necessárias, conectores independentes do protocolo de comunicação e ambientes isolados para execução de código. Antes de qualquer escrita no sistema físico, o plano e o código gerados são submetidos à detecção de operações de controle, à verificação dos limites das variáveis e à aprovação do operador. O Osprey foi demonstrado em um ambiente simulado e implantado

no Advanced Light Source, onde interagiu com centenas de milhares de variáveis do sistema EPICS, recuperou dados históricos e executou procedimentos após supervisão humana. Contudo, apesar de demonstrar uma implantação operacional com mecanismos de segurança e aprovação humana, o trabalho foi baseado principalmente em estudos de caso, sem métricas quantitativas sobre a qualidade das respostas ou comparação entre configurações com e sem recuperação documental.

Russell-Gilbert et al. (2025) desenvolveram o RAAD-LLM, uma arquitetura de detecção adaptativa de anomalias voltada à manutenção preditiva em ambientes industriais. A solução utilizou o modelo Llama 3.1 8B, executado localmente por meio do Ollama e sem ajuste fino, integrado a um mecanismo de RAG que recupera comparações estatísticas para auxiliar na classificação de janelas de séries temporais. O método foi avaliado com dados de falhas em uma linha de extrusão de plásticos e com o conjunto público SKAB, alcançando 88,6% de acurácia no caso industrial, em comparação aos 70,7% obtidos pela arquitetura anterior sem RAG. Contudo, os experimentos utilizaram apenas conjuntos de dados estáticos, analisados após a ocorrência das falhas. Dessa forma, a capacidade da abordagem de realizar detecção on-line em fluxos de dados industriais não foi avaliada.

Fan et al. (2025) desenvolveram uma estrutura hierárquica baseada em agentes LLM para interpretar instruções em linguagem natural, associar tarefas a parâmetros de processo, projetar trajetórias e executar operações de robótica industrial. A abordagem foi avaliada em simulações no PyBullet com diferentes modelos, entre os quais o GPT-4 obteve a maior taxa média de conclusão, alcançando 81,88% nas dez repetições das 16 tarefas selecionadas. Contudo, os testes permaneceram restritos à simulação, que simplificou operações como fresamento e impressão 3D, enquanto o robô físico ainda se encontrava em fase de depuração. Além disso, o GPT-4 alcançou apenas 45% de sucesso na geração de trajetórias tridimensionais sem o módulo de correção, evidenciando limitações na compreensão de relações espaciais.

Fukui et al. (2025) desenvolveram um sistema RAG utilizando 40 manuais internos de medicina nuclear de um hospital japonês. Para isso, o artigo apresenta uma arquitetura que combinou busca vetorial com o modelo `text-embedding-3-small` e busca por palavras-chave com BM25, enquanto GPT-3.5 e GPT-4o foram utilizados para gerar as respostas. A avaliação utilizou 100 pares de perguntas e respostas, analisados por três especialistas em uma escala de quatro níveis: resposta ideal, adequada porém redundante, parcialmente incompleta ou inadequada. Essa avaliação foi complementada por métricas automatizadas do RAGAS, como correção factual e cobertura do contexto recuperado. Com base nisso, a configuração formada por recuperação híbrida e GPT-4o obteve o melhor resultado, atingindo 84 pontos em uma escala de 0 a 100. Contudo, o estudo foi realizado com documentos e questões provenientes de uma única instituição, sem validação externa, o que restringe a generalização dos resultados para outros cenários. Apesar do contexto do estudo ser voltado à medicina nuclear, o método avaliativo aplicado serviu de referência

para o protocolo de avaliação descrito posteriormente na Metodologia.

Koziolek et al. (2024) propuseram um método baseado em LLM e RAG para gerar código de controle industrial em *Structured Text*. A abordagem organizou as especificações dos blocos de função da biblioteca OSCAT em uma base vetorial. A partir da descrição de uma tarefa de controle, o sistema recuperava os blocos mais relacionados e acrescentava suas informações ao contexto fornecido ao GPT-4. Com esse contexto, o modelo gerava o código em *Structured Text*, posteriormente compilado e simulado no OpenPLC. A partir disso, o protótipo foi avaliado em três testes envolvendo amostragem de sinais, geração de ondas e controle PID. Embora o modelo tenha selecionado os blocos de função adequados e produzido códigos posteriormente compilados e avaliados, a maioria dos testes exigiu correções manuais em variáveis ou chamadas de funções. Além disso, a avaliação foi restrita a três amostras pontuais, não permitindo determinar a robustez da abordagem em aplicações industriais mais complexas.

Arazzi et al. (2025) desenvolveram o RAG-IoE, uma arquitetura que utiliza um grafo de conhecimento para filtrar sensores e atuadores conforme a atividade e as permissões do operador. Em seguida, uma busca vetorial recupera as descrições técnicas mais relacionadas à tarefa, fornecendo-as a um modelo de linguagem local para selecionar os dispositivos e gerar orientações operacionais. Para a avaliação, os autores reuniram aproximadamente 50 tipos de dispositivos, com descrições obtidas de documentos técnicos e complementadas pelo Llama 2 7B, e elaboraram 16 atividades de diferentes níveis de complexidade. Os cenários combinaram dispositivos relevantes e aleatórios para testar a capacidade de seleção de cinco modelos. As respostas foram avaliadas por dois anotadores e pela biblioteca RAGAS. Como resultado, o modelo de 8B apresentou um melhor desempenho nas atividades fáceis e médias, enquanto o modelo de 14B se destacou nas atividades difíceis.

Diante dos trabalhos abordados, é evidente o uso de LLMs na recuperação de informações técnicas, na detecção de anomalias e no apoio ao operador (DALZELL et al., 2024; RUSSELL-GILBERT et al., 2025; KOZIOLEK et al., 2024; ARAZZI et al., 2025). Porém, dos materiais revisados, essas capacidades foram avaliadas em sua maioria com dados históricos e simulações, enquanto as soluções próximas ao controle físico necessitaram de mecanismos adicionais de correção, validação ou de supervisão humana (XIA et al., 2025; FAN et al., 2025; HELLERT; MONTENEGRO; SULC, 2026). Nesse contexto, este trabalho integra a um sistema SCADA a consulta de dados atuais e históricos, a recuperação de informações de uma base documental e a proposição de ações submetidas à aprovação do usuário. Assim, o agente baseado em LLM tem como principal função atuar como ferramenta de apoio operacional, mantendo a decisão humana sobre as ações capazes de modificar o processo.

# 3 Referencial Teórico

Neste capítulo são apresentados os principais conceitos utilizados para o desenvolvimento deste trabalho. Inicialmente, são abordados os sistemas SCADA, a plataforma SCADA-LTS e a integração entre aplicações por meio de APIs *Web*, seguidos pelos fundamentos dos sistemas hidráulicos de bombeamento. Em seguida, são apresentados os modelos de linguagem, os agentes inteligentes e o uso de ferramentas externas. Por fim, são discutidos a geração aumentada por recuperação e a recuperação vetorial.

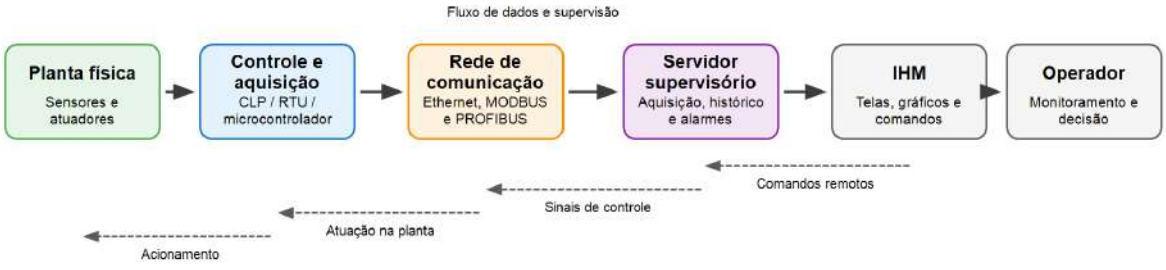
## 3.1 Sistemas SCADA e SCADA-LTS

### 3.1.1 Arquitetura e funções dos sistemas SCADA

Os sistemas SCADA (*Supervisory Control and Data Acquisition*) são utilizados para monitorar, controlar e registrar informações de processos físicos ou industriais, recebendo dados de sensores, enviando comandos para atuadores e apresentando informações relevantes do sistema em telas supervisórias. Por isso, seu principal objetivo é auxiliar os operadores no acompanhamento da planta em tempo real, permitindo uma visão mais clara do estado operacional do sistema.

Segundo Sverko, Grbac e Mikuc (2022), a arquitetura de um sistema SCADA normalmente envolve sensores e atuadores, unidades de controle ou aquisição de dados, redes de comunicação, servidores de supervisão e interfaces homem-máquina. A organização desses elementos e os fluxos de dados e comandos podem ser vistos na Figura 1.

Figura 1 – Arquitetura básica de um sistema SCADA.




Fonte: Elaboração própria com base em Sverko, Grbac e Mikuc (2022).

Conforme ilustrado na Figura 1, o fluxo de dados se inicia na planta, onde os

sensores convertem as grandezas do processo em sinais que são enviados às unidades de controle. O CLP recebe e organiza essas informações e pode encaminhar comandos aos atuadores e aos inversores responsáveis pelo acionamento dos motores. Na sequência, por meio de protocolos de comunicação industriais como *Ethernet*, Modbus ou *PROFIBUS*, as variáveis são enviadas para o sistema supervisório. Por fim, a interface homem-máquina (IHM) é responsável por apresentar ao operador os valores, estados e históricos da planta. No sentido inverso, os comandos remotos enviados pela IHM percorrem as mesmas camadas até os dispositivos da planta (SVERKO; GRBAC; MIKUC, 2022; WANG et al., 2016). Desse modo, essas funções são fundamentais para ambientes industriais e de saneamento, pois ajudam a identificar falhas, acompanhar tendências operacionais e melhorar a resposta dos operadores diante de alterações no processo (WANG et al., 2016).

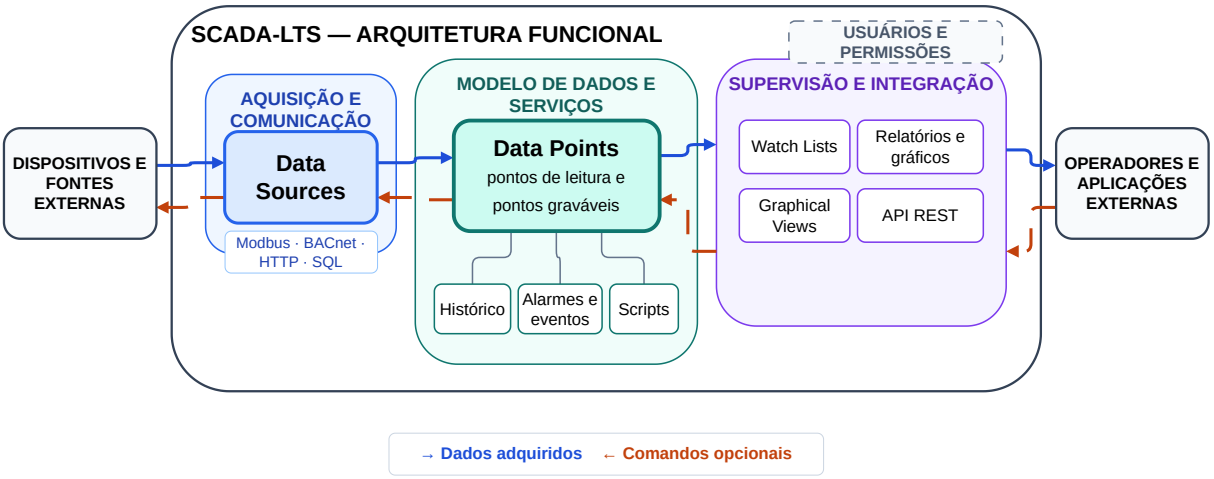
Nessa linha, existem diferentes plataformas que podem ser utilizadas para implementar sistemas supervisórios, variando entre soluções proprietárias e alternativas de código aberto. Entre essas alternativas, surge o SCADA-LTS, utilizado no LENHS-UFPB em atividades de supervisão, ensino, pesquisa e desenvolvimento de sistemas aplicados ao saneamento.

### 3.1.2 SCADA-LTS em ambientes industriais e acadêmicos

O SCADA-LTS é uma plataforma *open-source*, gratuita, voltada para a criação de sistemas SCADA. Semelhante a qualquer aplicação SCADA, permite configurar fontes de dados, integrar variáveis do sistema, gerenciar telas gráficas, alarmes e visualizar históricos de valores/operações. Por ser uma solução gratuita e otimizada para diferentes sistemas operacionais, a ferramenta se torna adequada para aplicações acadêmicas e industriais, principalmente para soluções que envolvem o desenvolvimento de sistemas supervisórios sem depender de licenças (UDDIN; BAIG; IQBAL, 2022).

A organização funcional desses recursos e seus fluxos de aquisição, supervisão e comando é sintetizada na Figura 2.

Figura 2 – Arquitetura funcional conceitual do SCADA-LTS.




Fonte: Elaboração própria com base na documentação oficial do SCADA-LTS.

Conforme ilustrado, os dispositivos e as fontes externas se comunicam com o SCADA-LTS por meio de *Data Sources*, que definem a origem dos dados e os meios utilizados para a sua aquisição. Os valores adquiridos são organizados em *Data Points*, que representam pontos de leitura ou pontos configurados como graváveis e podem ser associados a históricos, alarmes, eventos e *scripts*. Na sequência, esses dados são disponibilizados aos recursos de supervisão e integração. As *Watch Lists* reúnem variáveis selecionadas para monitoramento, enquanto os relatórios e gráficos permitem acompanhar os valores registrados. As *Graphical Views* permitem ao operador elaborar painéis gráficos, e uma interface de programação de aplicações (API, do inglês *Application Programming Interface*) baseada no estilo REST possibilita a integração com aplicações externas. Quando um *Data Point* permite escrita, os comandos podem percorrer o fluxo inverso até o dispositivo de origem.

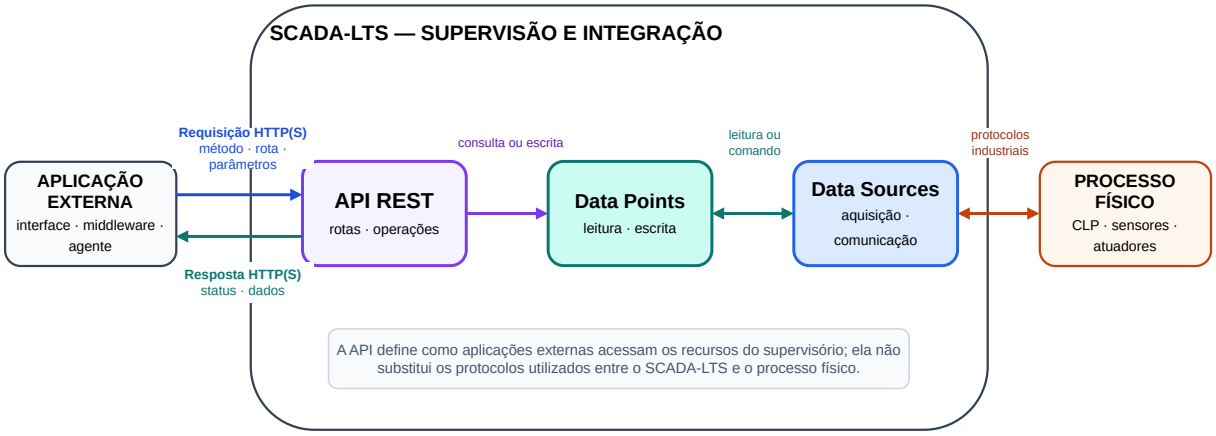
### 3.1.3 Integração entre aplicações por APIs Web

A integração de um sistema supervisório com aplicações externas depende de interfaces capazes de disponibilizar dados e operações de forma definida. Uma API estabelece um ponto de acesso com sintaxe e funcionalidades conhecidas, permitindo que uma aplicação utilize recursos oferecidos por outra sem depender dos seus detalhes internos de implementação (National Institute of Standards and Technology, 2026). Em aplicações Web, essas interfaces podem expor rotas para consultar informações, enviar dados ou solicitar operações.

O Protocolo de Transferência de Hipertexto (HTTP, do inglês *Hypertext Transfer Protocol*) é um protocolo de aplicação sem estado utilizado na comunicação entre clientes e servidores. Isso significa que cada requisição é tratada de forma independente, sem depender de informações mantidas pelo servidor sobre solicitações anteriores. Por meio dele,

um cliente envia uma requisição relacionada a determinado recurso e o servidor devolve uma resposta com o resultado da operação (FIELDING; NOTTINGHAM; RESCHKE, 2022). As APIs Web podem organizar essas interações segundo o estilo arquitetural Transferência de Estado Representacional (REST, do inglês *Representational State Transfer*), que estabelece restrições como a separação entre cliente e servidor, a ausência de estado da sessão no servidor entre requisições e uma interface uniforme para os recursos (FIELDING, 2000). Desse modo, REST não constitui um protocolo de comunicação, mas um estilo arquitetural frequentemente aplicado a APIs que utilizam HTTP.

Figura 3 – Integração conceitual de uma aplicação externa com o SCADA-LTS por meio de uma API Web.




Fonte: Elaboração própria com base em Fielding, Nottingham e Reschke (2022) e SCADA-LTS (2026).

Conforme apresentado na Figura 3, a API REST atua como uma interface entre as aplicações externas e os recursos disponibilizados pelo SCADA-LTS. A aplicação envia uma requisição HTTP ou HTTPS para uma rota definida, indicando a operação e os parâmetros necessários, e recebe como resposta o estado da operação e os dados solicitados. No interior do supervisório, a API acessa recursos associados aos *Data Points*, que representam as variáveis de leitura ou os pontos configurados como graváveis. Esses pontos permanecem vinculados aos *Data Sources*, responsáveis pela comunicação com o CLP e os dispositivos da planta por meio dos protocolos industriais configurados. Desse modo, a API não substitui a camada de aquisição de dados do sistema SCADA; ela fornece uma interface padronizada para que outras aplicações consultem ou solicitem operações sobre os recursos disponibilizados pelo SCADA-LTS (SCADA-LTS, 2026).

Para trocas contínuas de dados, o protocolo *WebSocket* permite manter um canal bidirecional após o estabelecimento da conexão, de modo que cliente e servidor possam enviar mensagens sem iniciar uma nova requisição HTTP a cada atualização (FETTE; MELNIKOV, 2011). Em sistemas supervisórios, essas formas de comunicação possuem funções complementares: HTTP ou HTTPS podem atender a consultas e operações

pontuais, enquanto WebSocket ou WSS podem ser utilizados para atualizar continuamente as variáveis apresentadas ao operador.

## 3.2 Sistemas Hidráulicos de Bombeamento

### 3.2.1 Grandezas hidráulicas fundamentais

Os sistemas de bombeamento são utilizados para transportar água entre diferentes pontos de um percurso hidráulico, alcançando reservatórios, redes ou pontos de consumo com pressão e vazão adequadas. Em sistemas de abastecimento e distribuição de água, esses conjuntos permitem vencer desníveis geométricos, perdas de carga nas tubulações e as variações de demanda ao longo da operação.

Para a análise desses sistemas, é necessário compreender as grandezas hidráulicas fundamentais. A vazão representa o volume de água transportado por unidade de tempo, enquanto a pressão está relacionada à energia disponível no fluido em determinado ponto da tubulação. Por fim, a altura manométrica representa a energia por unidade de peso que a bomba deve fornecer ao escoamento, considerando a diferença de nível entre os pontos do sistema, as perdas de carga e as condições de sucção e recalque (FLORES, 2018).

É importante citar as perdas de carga do sistema, que correspondem às perdas de energia que ocorrem durante o escoamento da água, principalmente devido ao atrito com as paredes da tubulação e à presença de conexões, curvas, válvulas e registros. Dessa forma, quanto maior a resistência hidráulica do sistema, maior tende a ser a energia necessária para manter o escoamento desejado.

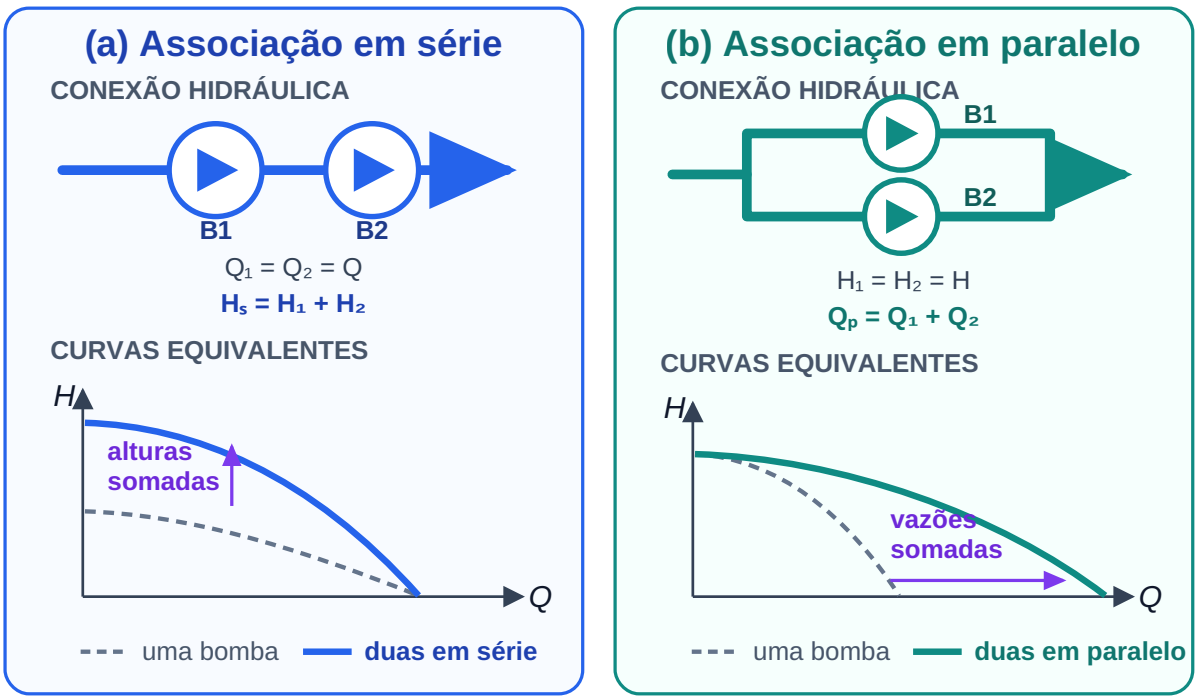
Nos sistemas de bombeamento, os conjuntos motobomba são responsáveis por fornecer energia ao fluido. O motor converte energia elétrica em energia mecânica, enquanto a bomba transfere essa energia para a água, permitindo o aumento da pressão e o deslocamento do fluido ao longo da tubulação. Em aplicações com inversores de frequência, a rotação do motor pode ser alterada, modificando o comportamento hidráulico do sistema e permitindo diferentes condições de operação.

Além das variáveis hidráulicas, a operação de sistemas de bombeamento também envolve grandezas elétricas associadas ao acionamento dos motores. Variáveis como frequência, corrente, tensão e potência permitem acompanhar o funcionamento dos conjuntos motobomba e avaliar a condição operacional do sistema. Por isso, em uma bancada experimental, o monitoramento dessas informações é importante para relacionar o comportamento hidráulico com o consumo energético e apoiar atividades de supervisão, controle e análise de desempenho.

### 3.2.2 Associações de bombas em série e em paralelo

Em sistemas hidráulicos que utilizam mais de um conjunto motobomba, a forma de associação influencia a altura manométrica e a vazão fornecidas. Segundo Flores (2018), duas configurações comuns são as associações em série e em paralelo, empregadas de acordo com a necessidade hidráulica do sistema. Na Figura 4 são sintetizadas as conexões entre duas bombas idênticas e a construção qualitativa de suas curvas equivalentes. Na figura, $Q$ representa a vazão e $H$, a altura manométrica.

Figura 4 – Associações de duas bombas idênticas em série e em paralelo e respectivas curvas equivalentes.




Fonte: Elaboração própria com base em Flores (2018).

Conforme o painel (a) da Figura 4, na associação em série a descarga da primeira bomba é conectada à sucção da segunda, de modo que a mesma vazão atravessa os dois conjuntos, ou seja, $Q_1 = Q_2 = Q$. As alturas manométricas fornecidas por cada bomba são somadas, resultando em $H_s = H_1 + H_2$. Por isso, a curva equivalente da associação é obtida pela soma vertical das alturas manométricas correspondentes a uma mesma vazão. Essa configuração é empregada quando o sistema requer maior altura manométrica, como no atendimento a maiores desníveis ou perdas de carga.

No painel (b), as bombas operam em ramais paralelos ligados a linhas comuns de sucção e recalque. Nessa configuração, ambas estão submetidas à mesma altura manométrica, $H_1 = H_2 = H$, enquanto a vazão total corresponde à soma das contribuições

dos dois ramais, $Q_p = Q_1 + Q_2$. Assim, a curva equivalente é construída pela soma horizontal das vazões correspondentes a uma mesma altura manométrica. A associação em paralelo é empregada quando se deseja ampliar a vazão disponível para atender a maiores demandas de escoamento.

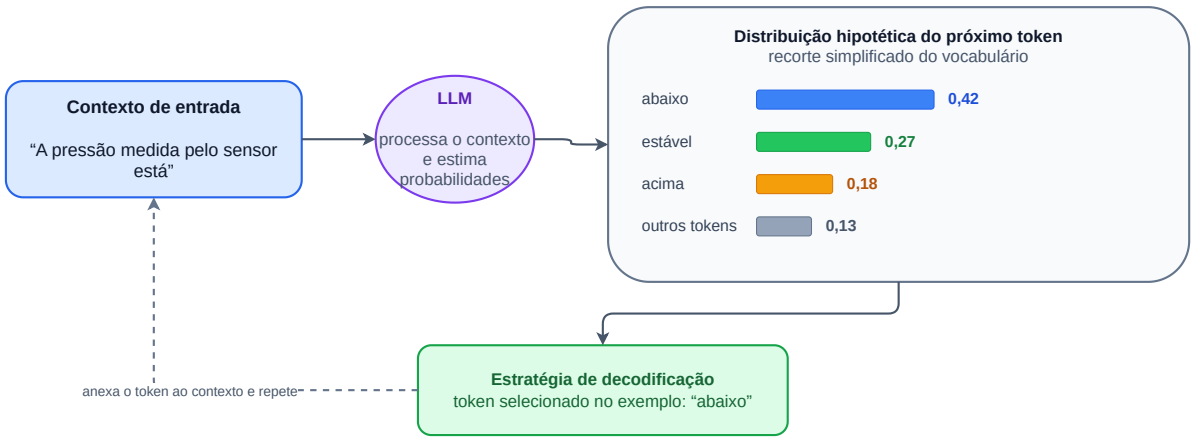
## 3.3 Modelos de Linguagem e Agentes Inteligentes

Os modelos de linguagem de grande porte, conhecidos como LLMs (*Large Language Models*), são modelos computacionais treinados sobre grandes coleções de texto para aprender regularidades linguísticas e padrões presentes em contextos diversos. Durante o pré-treinamento do modelo, ele é treinado para estimar a probabilidade dos próximos elementos de uma sequência textual.

Esses modelos podem ser utilizados em tarefas como resposta a perguntas, resumo de textos e extração de informações, permitindo a interação com sistemas por meio da linguagem natural (ZHAO et al., 2023).

Desse modo, a geração de uma resposta ocorre de forma progressiva e probabilística: cada elemento textual (chamado de *token*) é escolhido de acordo com o texto produzido anteriormente e com as informações fornecidas na entrada do modelo (ZHAO et al., 2023). Essa distribuição de probabilidades é discutida por Wang, Zhu e Shi (2025) e representada de forma simplificada na Figura 5.

Figura 5 – Representação simplificada da geração autorregressiva do próximo *token* a partir do contexto de entrada.




Fonte: Elaboração própria com base em Wang, Zhu e Shi (2025).

Conforme ilustrado na Figura 5, o modelo recebe o contexto de entrada e estima uma distribuição de probabilidades para os possíveis próximos *tokens*. No exemplo apresentado, a estratégia de decodificação seleciona o *token* ”abaixo”, que possui a maior probabilidade hipotética. Por fim, o elemento selecionado é acrescentado ao contexto, formando uma nova

entrada para a próxima etapa. Esse processo se repete progressivamente até a formação da resposta completa (WANG; ZHU; SHI, 2025).

No momento da utilização, a entrada do modelo pode conter instruções, uma pergunta e informações contextuais relacionadas à tarefa, o que auxilia temporariamente na geração de uma resposta, mas não modifica os parâmetros aprendidos. Por isso, é importante distinguir o conhecimento paramétrico que é incorporado ao modelo no treinamento das informações fornecidas no contexto de uma consulta por um usuário (ZHAO et al., 2023). Essa distinção é particularmente importante em aplicações que dependem de informações técnicas ou operacionais específicas.

Contudo, embora os textos produzidos pelos modelos sejam coerentes e plausíveis, a geração é orientada pela probabilidade do próximo *token*, e não por um mecanismo próprio de verificação factual. Assim, uma resposta fluente não necessariamente corresponde a uma resposta correta nem sustentada por uma fonte. Segundo Huang et al. (2025), esse fenômeno, conhecido como alucinação, corresponde à geração de conteúdos incorretos, não verificáveis ou inconsistentes com as informações disponíveis. Em aplicações industriais, essas limitações indicam que as respostas geradas não devem ser interpretadas diretamente como comandos confiáveis, sendo necessário recorrer a fontes externas, mecanismos de validação e supervisão humana.

### 3.3.1 Agentes baseados em modelos de linguagem e uso de ferramentas

Os agentes baseados em modelos de linguagem utilizam uma LLM como componente de interpretação e decisão dentro de um sistema capaz de receber informações e realizar ações. Segundo Xi et al. (2025), a estrutura desses agentes pode ser compreendida a partir dos componentes de percepção, decisão e ação. Desse modo, os agentes podem ser configurados para contextos específicos e auxiliar na tomada de decisões e na geração de respostas. Vale destacar que a resposta textual gerada por um agente mantém a natureza probabilística da LLM, mas o intuito do agente é atribuir um propósito às entradas e saídas associadas a ele, de acordo com as instruções inseridas em seu contexto, de forma semelhante à definição de uma identidade. Como exemplo, é possível criar múltiplos agentes com papéis especializados, conforme pode ser visto na Figura 6.

Figura 6 – Exemplo de agentes baseados na mesma LLM e configurados com papéis especializados.

Fonte: Elaboração própria com base em Xi et al. (2025).

Como ilustrado no exemplo, é feita uma mesma solicitação para dois agentes com instruções diferentes. O agente de diagnóstico hidráulico direciona sua resposta para a análise das variáveis e da configuração do sistema, enquanto o agente de segurança operacional prioriza a verificação de riscos, intertravamentos e limites operacionais. Dessa forma, mesmo utilizando o mesmo modelo e recebendo a mesma solicitação, cada agente produz uma resposta orientada pelo contexto e pelas instruções associadas ao seu papel (XI et al., 2025).

Ademais, os agentes baseados em modelos de linguagem não precisam estar limitados à geração textual. Eles podem utilizar ferramentas externas para ampliar suas capacidades, incluindo funções computacionais, mecanismos de busca, bancos de dados e APIs. Segundo Qu et al. (2025), a integração com ferramentas permite que os modelos interajam dinamicamente com recursos externos, ampliando sua capacidade de solucionar tarefas que exigem informações ou operações não disponíveis apenas durante a geração textual.

Para que o modelo de linguagem possa decidir quando e como utilizar cada ferramenta, são fornecidas, durante a execução do agente, descrições da finalidade de cada recurso e dos parâmetros exigidos para sua chamada. Com base nessas informações, o modelo pode identificar a necessidade de uma ferramenta, selecionar o recurso adequado, fornecer os parâmetros de entrada e utilizar o resultado retornado na elaboração da

resposta (QU et al., 2025).

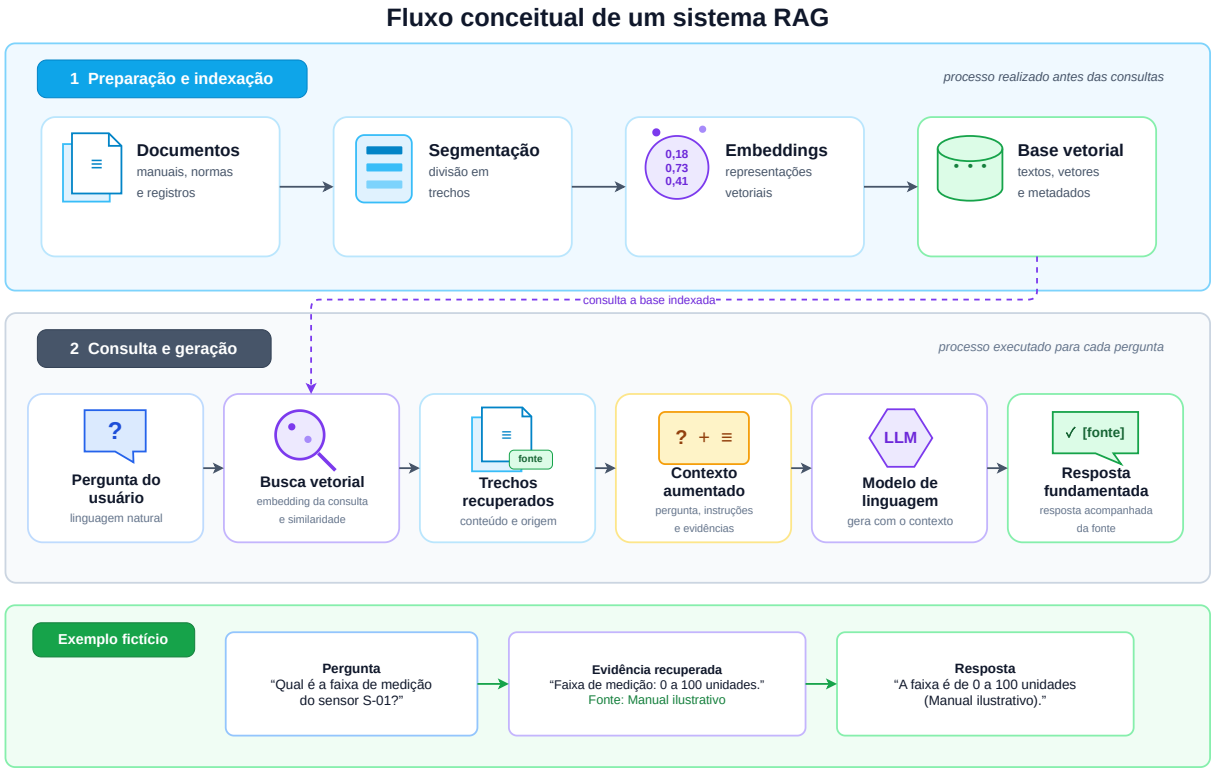
Diante disso, em sistemas supervisórios, as ferramentas podem estabelecer uma interface entre o agente e as informações disponíveis no processo industrial, como a consulta de variáveis operacionais, a recuperação de documentos técnicos, a realização de cálculos estatísticos e a proposição de alterações nas variáveis do sistema. Esse último ponto, entretanto, não representa, por si só, a execução da ação no processo. A operação deve ser realizada por um componente externo e pode depender de verificações, permissões e aprovação humana.

## 3.4 Geração Aumentada por Recuperação e Recuperação Vetorial

A Geração Aumentada por Recuperação, conhecida pela sigla RAG, do inglês *Retrieval-Augmented Generation*, é uma estratégia utilizada para fornecer informações externas ao modelo de linguagem no momento de uma consulta. Em vez de depender apenas do conhecimento paramétrico adquirido durante o treinamento, o sistema pode recuperar informações de uma base documental e acrescentá-las ao contexto utilizado na geração da resposta. Segundo Fan et al. (2024), esse processo envolve os componentes de recuperação, geração e aumento de contexto, nos quais os documentos relacionados à consulta são recuperados e combinados com a entrada enviada ao modelo.

A organização dessas etapas é sintetizada na Figura 7.

Figura 7 – Fluxo simplificado de preparação, recuperação e geração de uma resposta em um sistema RAG.




Fonte: Elaboração própria com base em Fan et al. (2024).

Como mostrado na Figura 7, é importante destacar que a recuperação e a geração representam etapas distintas. O mecanismo de recuperação é responsável por localizar os conteúdos considerados relacionados à consulta, enquanto o modelo de linguagem interpreta esses conteúdos e elabora a resposta. Desse modo, os documentos recuperados fornecem um contexto adicional sem modificar os parâmetros do modelo. Essa característica permite utilizar informações específicas ou atualizadas durante a geração, mas não elimina a natureza probabilística da resposta nem garante que o conteúdo recuperado será utilizado corretamente (FAN et al., 2024).

A faixa superior da Figura 7 representa a etapa de preparação e indexação. Antes que os documentos possam ser recuperados, eles precisam passar por uma etapa de preparação. Esse processo pode envolver a extração do conteúdo textual, a divisão do documento em trechos e a associação de metadados relacionados à origem de cada conteúdo. A segmentação permite que partes menores de um documento sejam recuperadas sem que todo o arquivo seja inserido no contexto do modelo. Conforme discutido por Fan et al. (2024), a granularidade utilizada na recuperação, como documento, passagem ou trecho, influencia a eficiência da busca e a quantidade de informações relevantes ou desnecessárias fornecidas ao modelo.

Após a segmentação, os trechos podem ser transformados em representações nu-

méricas denominadas *embeddings*. Na recuperação densa, tanto os documentos quanto a consulta são representados em um espaço vetorial, no qual a proximidade entre os vetores indica uma relação semântica entre os conteúdos (FAN et al., 2024). Esses vetores são armazenados e indexados em um banco vetorial, juntamente com metadados que permitem relacionar cada trecho ao seu documento de origem. Dessa forma, além de auxiliar na busca, a preservação dos metadados permite identificar a proveniência das informações recuperadas.

A faixa inferior da Figura 7 representa a etapa de consulta e geração. No momento da consulta, a entrada do usuário também é representada vetorialmente e comparada com os conteúdos indexados. Os trechos considerados mais próximos são retornados pelo mecanismo de busca e acrescentados ao contexto fornecido ao modelo de linguagem. Assim, o banco vetorial não é responsável por gerar a resposta, mas por armazenar, indexar e recuperar os conteúdos que podem auxiliar sua elaboração. A resposta final continua sendo produzida pelo modelo a partir da consulta, das instruções e dos trechos recuperados.

A qualidade de um sistema RAG pode ser analisada a partir da recuperação documental e da resposta gerada, pois a localização de um conteúdo relevante não garante, por si só, que o modelo o utilizará corretamente. Segundo Salemi e Zamani (2024), avaliações baseadas apenas na relação entre a consulta e o documento recuperado apresentam correlação limitada com o desempenho final do sistema. Por isso, na recuperação, devem ser observadas a pertinência, a utilidade e a origem dos trechos; na resposta, devem ser consideradas a correção, a utilidade e a fidelidade às informações disponibilizadas. Essa separação permite identificar se uma falha ocorreu durante a busca documental ou durante a interpretação e a geração realizadas pelo modelo.

# 4 Metodologia e Desenvolvimento

Neste capítulo são descritos os procedimentos metodológicos e as etapas de desenvolvimento adotadas na realização deste projeto. Nele, apresenta-se a caracterização física da bancada de bombeamento do LENHS, a arquitetura geral do *middleware* assíncrono em *FastAPI*, a dinâmica de funcionamento do RAG local e as estratégias estruturadas de cibersegurança operacional para conter alucinações do modelo de linguagem.

## 4.1 Caracterização da Bancada Física do LENHS-UFPB

A validação empírica do sistema de integração foi conduzida na bancada hidráulica de distribuição de água de eficiência energética e hidráulica em saneamento do LENHS. Esta infraestrutura é composta por:

- Dois conjuntos motobomba centrífugos idênticos;
- Cinco transmissores de pressão DRUCK PTX 7217;
- Dois medidores eletromagnéticos de vazão VMS Pro;
- Dois inversores de frequência Siemens SINAMICS V20, associados individualmente aos conjuntos motobomba;
- Uma válvula de controle proporcional BELIMO ARB24-SR, empregada para alterar a perda de carga e representar variações de demanda hidráulica;
- Um Controlador Lógico Programável Siemens S7-1200, responsável pela aquisição dos sinais de campo e pela comunicação com os dispositivos de acionamento e supervisão.

Durante a etapa de levantamento da planta, foi analisado de que forma os componentes são conectados à sala de controle. A partir disso, cada componente recebeu um nome de variável e foi mapeado no diagrama de comunicação e instrumentação do sistema. Também foram observados os pontos da planta que deveriam ser representados no supervisório, como os estados de acionamento dos motores e as medições de pressão e vazão.

A Figura 8 apresenta duas vistas anotadas da instalação física. As identificações relacionam a nomenclatura adotada neste trabalho à posição dos conjuntos motobomba, dos instrumentos, da válvula de controle e dos registros empregados na definição das associações em série e em paralelo.

Figura 8 – Vistas anotadas da bancada experimental Série-Paralelo do LENHS-UFPB.

**(a)** Vista oblíqua. **(b)** Vista frontal.

Fonte: Autoria própria.

Os parâmetros considerados para os componentes físicos da bancada foram organizados conforme a Tabela 1. Essa organização relaciona os equipamentos identificados nas fotografias às grandezas disponibilizadas pelo sistema de supervisão.

Tabela 1 – Parâmetros considerados para os componentes físicos da bancada.

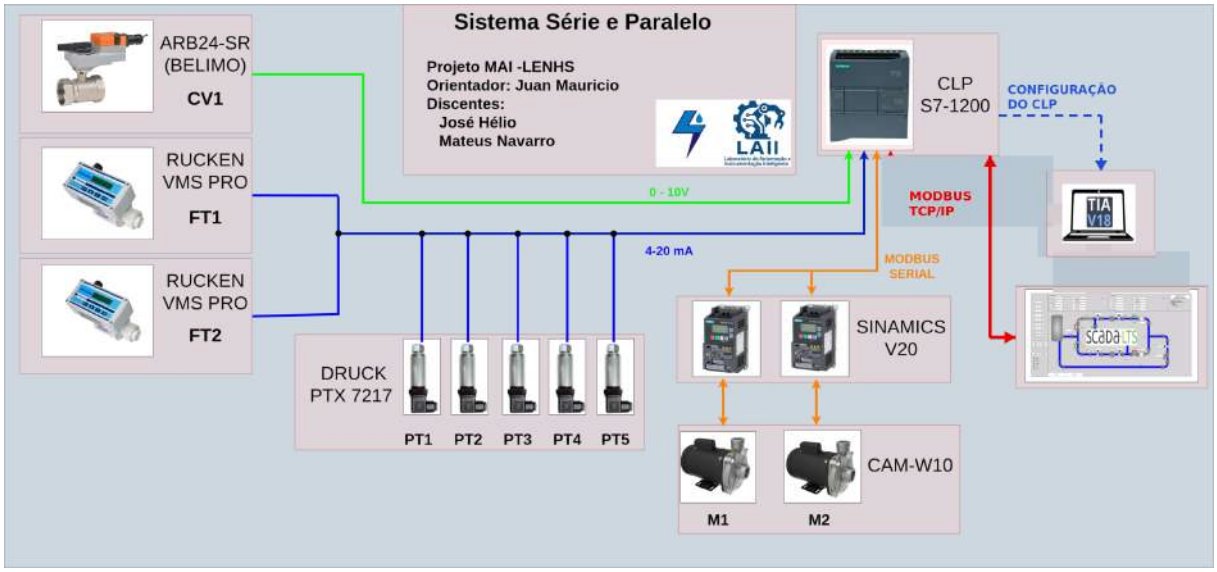
| **Componente físico** | **Parâmetros considerados** |
|---|---|
| Conjuntos motobomba M1 e M2 | Acionamento, frequência, velocidade, corrente, torque, potência real, energia, tensão do barramento CC, referência de frequência, potência nominal e tensão de saída. |
| Transdutores de pressão | Pressões medidas nos pontos da bancada. |
| Transdutores de vazão | Vazões medidas nos trechos da bancada. |
| Válvula de controle proporcional | Posição ou abertura associada à variação da condição hidráulica do sistema. |
| Registros manuais e chaves de configuração | Estados utilizados para indicar a condição operacional e a configuração da bancada. |

Fonte: Autoria própria.

Na arquitetura atual, os transmissores de pressão e vazão enviam sinais de 4–20 mA ao CLP, enquanto a válvula proporcional recebe um sinal de comando de 0–10 V. Os dois inversores SINAMICS V20 comunicam-se com o S7-1200 por Modbus Serial. O supervisório SCADA-LTS, instalado em um computador local, comunica-se diretamente com o CLP por Modbus TCP/IP para ler as variáveis disponibilizadas e encaminhar comandos de operação. O TIA Portal é utilizado apenas para configurar e programar o CLP, não participando da comunicação entre o SCADA-LTS e o S7-1200 durante a execução da aplicação. A seleção hidráulica entre as associações em série e em paralelo permanece

dependente da configuração dos registros manuais da bancada; a válvula proporcional não realiza essa comutação. A Figura 9 apresenta a organização geral da instrumentação, dos acionamentos e da comunicação da bancada, além do recurso utilizado para a configuração do CLP.

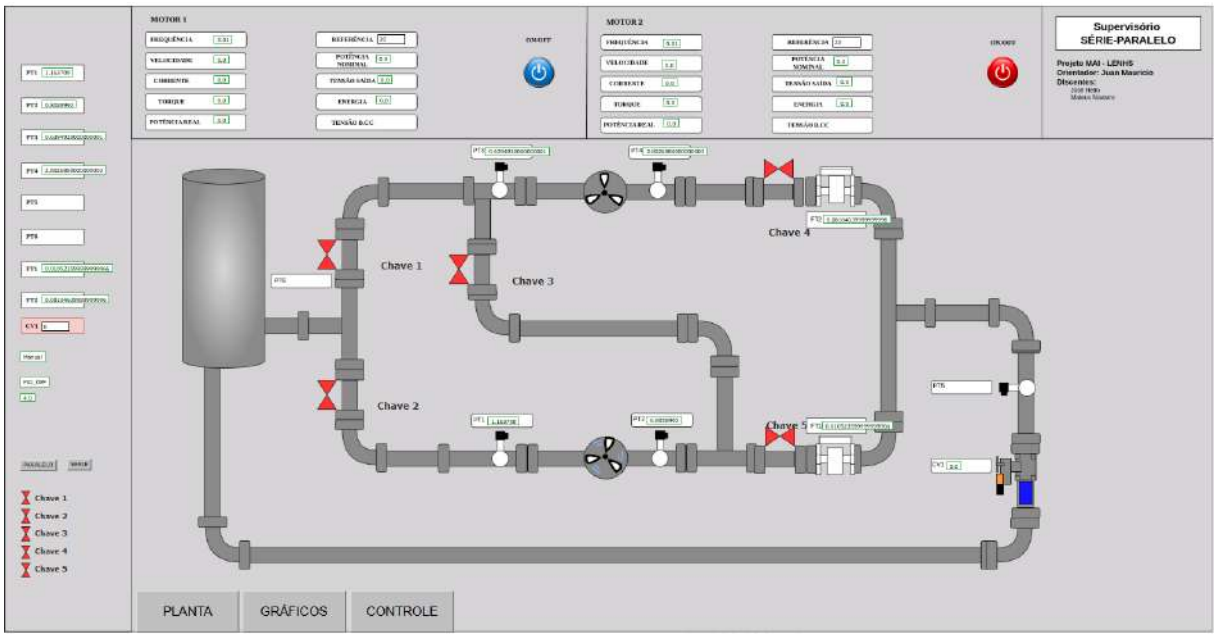
Figura 9 – Topologia de instrumentação, controle e supervisão da configuração atual da bancada Série-Paralelo.




Fonte: Autoria própria.

A correspondência entre os componentes físicos e sua representação supervisória pode ser observada na tela principal do SCADA-LTS, apresentada na Figura 10. A interface reproduz o percurso hidráulico e reúne os conjuntos M1 e M2, os pontos de pressão e vazão, a válvula CV1 e as cinco chaves de configuração. Essa representação serve como referência para relacionar os estados e valores exibidos pelo supervisório aos componentes identificados nas fotografias da Figura 8.

Figura 10 – Tela principal de controle e monitoramento do supervisório Série-Paralelo no SCADA-LTS.




Fonte: Autoria própria.

O trabalho de Flores (2018) constitui a principal referência para a estrutura hidráulica, a disposição dos instrumentos e o comportamento das associações de motobombas. Entretanto, a configuração de aquisição e supervisão empregada naquele experimento era distinta: utilizava a interface NI USB-6229, o ambiente LabVIEW e os inversores CFW-11 e Altivar 31. Na configuração atual, esses elementos foram substituídos pela arquitetura baseada no CLP S7-1200, em dois inversores SINAMICS V20 e no SCADA-LTS. Essa distinção permite aproveitar os fundamentos e resultados hidráulicos apresentados por Flores sem atribuir à planta atual uma camada de automação que já não corresponde à instalação existente.

A configuração descrita constitui o ambiente no qual a aplicação proposta foi integrada. Por meio do SCADA-LTS, o agente pode consultar as variáveis disponibilizadas pelo CLP e formular recomendações ou propostas de escrita relacionadas ao processo.

Diante desse cenário, a utilização de agentes baseados em modelos de linguagem em sistemas ciberfísicos exige cuidados adicionais, pois as respostas geradas podem ser utilizadas para recomendar ou propor ações que afetam componentes físicos. Diferentemente de uma resposta apenas textual, uma operação de escrita pode alterar *setpoints*, estados de atuadores ou outras variáveis do processo. Por isso, uma resposta gerada pelo modelo não deve ser transformada diretamente em um comando de controle. Segundo Hellert, Montenegro e Sulc (2026), sistemas baseados em agentes destinados a ambientes críticos devem tornar as ações planejadas visíveis, verificar os limites operacionais e submeter as operações sobre o *hardware* à revisão humana antes de sua execução.

Nesse contexto, o *Human-in-the-Loop* (HITL) representa a inclusão de uma pessoa na operação ou no monitoramento de um sistema, permitindo que o conhecimento humano participe das decisões realizadas em conjunto com os recursos computacionais. Essa participação pode ocorrer em diferentes níveis, desde o acompanhamento das informações até a avaliação e intervenção em uma ação proposta. De acordo com Bhattacharya et al. (2023), os operadores continuam exercendo um papel importante em sistemas de manufatura inteligentes, nos quais a automação deve ampliar as capacidades humanas e favorecer a colaboração entre pessoas e máquinas.

Desse modo, é importante distinguir as operações de leitura, a geração de recomendações, a solicitação de escrita e a execução de uma ação. A leitura e a recomendação fornecem informações ao operador sem alterar o estado da planta. Por outro lado, quando o agente propõe uma escrita, a ação deve permanecer pendente até que o operador avalie e decida por sua aprovação ou recusa. Assim, o agente atua como um recurso de apoio na camada supervisória, enquanto as malhas de controle em tempo real permanecem responsáveis pelo controle direto do processo e as proteções tradicionais continuam impondo os limites operacionais da planta. Essa separação preserva a autoridade do operador e mantém a supervisão inteligente como uma camada complementar, e não como substituta dos mecanismos de controle e segurança existentes (HELLERT; MONTENEGRO; SULC, 2026).

## 4.2 Arquitetura Geral da Aplicação

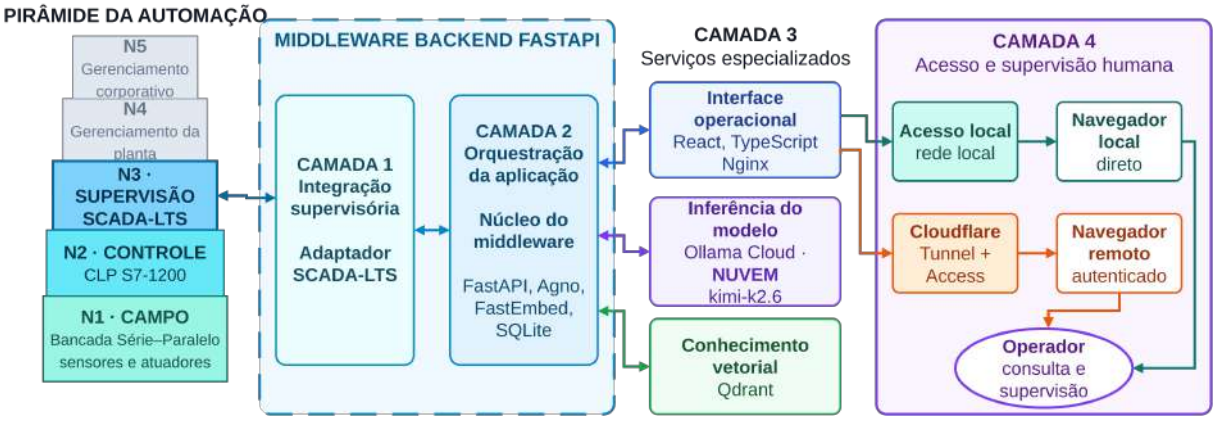
Antes da definição dos componentes, foram estabelecidos requisitos gerais para orientar a arquitetura da aplicação e sua integração com a bancada. Esses requisitos foram definidos considerando as características do sistema supervisório existente, os recursos computacionais disponíveis e a necessidade de manter o operador responsável pelas ações sobre o processo. Desse modo, a aplicação deveria atender aos seguintes requisitos:

- **Integração:** comunicar-se com o SCADA-LTS existente para consultar os valores atuais e históricos das variáveis e encaminhar operações de escrita, sem substituir as camadas de controle e supervisão já instaladas;
- **Desempenho:** realizar a coleta e a atualização periódica dos dados em uma frequência compatível com o acompanhamento supervisório, mantendo o processamento do agente separado das malhas de controle em tempo real;
- **Execução local e em nuvem:** manter localmente a coleta, a persistência dos dados e a recuperação documental, permitindo que a inferência do modelo de linguagem seja realizada localmente ou por um serviço externo, de acordo com os recursos computacionais disponíveis;

- **Rastreabilidade:** registrar os dados operacionais, as sessões do agente, as fontes documentais utilizadas e as decisões de aprovação ou recusa tomadas pelo operador;
- **Segurança operacional:** impedir que o modelo de linguagem execute diretamente ações sobre a bancada, condicionando as propostas de escrita à aprovação humana e à validação realizada pelo *backend*;
- **Viabilidade e custo:** utilizar tecnologias compatíveis com o computador disponível na bancada e priorizar componentes que pudessem ser executados localmente, reduzindo a necessidade de infraestrutura computacional adicional.

A aplicação foi organizada com o objetivo de separar a interface utilizada pelo operador, as regras e integrações do sistema e o armazenamento vetorial da base documental. A Figura 11 apresenta essa organização e o fluxo de comunicação entre o operador, a aplicação, o SCADA-LTS e o serviço de inferência do modelo de linguagem.

Figura 11 – Arquitetura de implantação da aplicação de integração SCADA–Agente.




Fonte: Autoria própria.

### 4.2.1 Camada 1 — Integração supervisória

A primeira camada constitui a fronteira de integração entre a aplicação e o SCADA-LTS. Essa comunicação é realizada por meio de um adaptador responsável por manter a sessão com o supervisório e encapsular as particularidades de sua API REST. Além disso, possui funções como identificar os pontos pelos XIDs, converter os tipos de dados e executar as operações de leitura e escrita.

A rotina de coleta utiliza esse adaptador para consultar periodicamente os pontos configurados e disponibilizar as medições ao núcleo da aplicação. Como também, em ações de escrita validadas, o adaptador converte o valor para o formato esperado e encaminha para o SCADA-LTS.

### 4.2.2 Camada 2 — Orquestração da aplicação

A segunda camada corresponde ao núcleo da orquestração da aplicação. Sua função é coordenar os fluxos do sistema, manter o estado da aplicação e intermediar a comunicação entre a interface, o agente, a camada de integração e os serviços de armazenamento.

O núcleo da aplicação foi implementado com FastAPI, um *framework* Python utilizado para disponibilizar rotas da API e coordenar a comunicação entre os componentes. Essa tecnologia foi escolhida por permitir reunir, no mesmo *backend* em Python, as rotas HTTP e o canal WebSocket utilizados pela interface. Em paralelo, o agente desenvolvido com Agno é executado nessa camada da aplicação, que acessa as medições, o histórico, os registros e a base de conhecimento por meio de ferramentas gerenciadas pelo *software*. A escolha do Agno permitiu organizar em uma mesma estrutura o modelo de linguagem, as instruções, as ferramentas, o histórico das sessões e a base de conhecimento.

Desse modo, a comunicação entre a interface e o *backend* é realizada por meio da API HTTP, permitindo que a interface envie solicitações pontuais, como mensagens ao agente, consultas ao histórico, acesso aos registros e decisões de aprovação ou rejeição.

Para a atualização em tempo real dos valores da planta, foi utilizado o protocolo de comunicação *WebSocket*, já que, diferentemente de requisições HTTP, a conexão continua aberta durante o uso da aplicação e garante uma comunicação bidirecional. Assim, permite que o *backend* envie os dados operacionais mais recentes sem que o navegador solicite uma atualização.

As áreas apresentadas na Tabela 3 utilizam diferentes meios de comunicação com o *backend*, de acordo com o tipo de informação trocada. Essa relação é apresentada na Tabela 2.

Tabela 2 – Meios de comunicação entre as áreas da interface e o *backend*.

| **Área da interface** | **Meio de comunicação principal** |
|---|---|
| Acompanhamento operacional | WebSocket para o recebimento das medições e requisições HTTP para o estado dos serviços. |
| Interface conversacional | Requisições HTTP para o envio das mensagens e o recebimento das respostas do agente. |
| Análise histórica | Requisições HTTP para consultar os dados de uma variável no intervalo selecionado. |
| Auditoria | Requisições HTTP para consultar os eventos registrados pela aplicação. |
| Acesso ao SCADA-LTS | Comunicação HTTP intermediada pelo Nginx e pelo *backend*. |
| Configuração | Requisições HTTP para consultar ou alterar as informações utilizadas pela aplicação. |

Fonte: Autoria própria.

A aplicação precisa manter diferentes tipos de informações durante o seu funciona-

mento, como o histórico das variáveis coletadas do SCADA-LTS, as sessões de conversa do agente e os documentos cadastrados na base de conhecimento. Desse modo, foram utilizados mecanismos de persistência diferentes de acordo com a finalidade de cada informação.

Para os dados estruturados gerados pela aplicação, foi utilizado o SQLite, um banco de dados local que não necessita da execução de um serviço separado. Essa escolha foi realizada por permitir uma persistência local compatível com o computador da bancada e com os dados estruturados produzidos pela aplicação. Esse armazenamento é independente do banco de dados interno do SCADA-LTS. Um dos bancos SQLite é utilizado para armazenar as amostras históricas das variáveis, associando o horário, o ponto e o valor coletado. Além disso, bancos SQLite separados são utilizados para preservar as sessões e o histórico das conversas do agente e para manter o catálogo dos conteúdos inseridos na base de conhecimento.

Já para a recuperação documental, o FastEmbed é utilizado no *backend* para transformar os trechos dos documentos e as consultas em representações numéricas denominadas *embeddings*. Ele foi escolhido por permitir a geração local dessas representações no mesmo ambiente Python da aplicação, sem depender de uma API externa para essa etapa. Essas representações e seus metadados são armazenados no Qdrant, apresentado na Camada 3 como um serviço especializado.

### 4.2.3 Camada 3 — Serviços especializados

A terceira camada reúne os serviços especializados que se comunicam com o núcleo da aplicação. Ela é composta pela interface operacional, pelo serviço de inferência do modelo de linguagem e pelo serviço de conhecimento vetorial. Vale citar que esses componentes têm funções específicas e são executados de forma separada.

A interface operacional corresponde ao *frontend* da aplicação, responsável por disponibilizar ao operador o acompanhamento da bancada e a interface conversacional. Por meio dela, são enviadas as solicitações ao agente e apresentadas as respostas, os dados operacionais, as fontes documentais e as propostas de ação que dependem de aprovação.

A interface foi desenvolvida com React, uma biblioteca JavaScript utilizada para construir interfaces por meio de componentes. Com isso, foi possível separar as áreas de acordo com suas finalidades. Além disso, os componentes foram escritos com TypeScript, uma linguagem baseada em JavaScript que adiciona tipagem ao desenvolvimento e permite estruturar os dados recebidos do *backend*. As duas tecnologias foram escolhidas devido à integração entre elas e à possibilidade de organizar a aplicação web em componentes. A localização das tecnologias utilizadas nas camadas da aplicação é apresentada na Figura 11. Diante disso, cada área da interface foi definida com uma finalidade, conforme apresentado na Tabela 3.

Tabela 3 – Áreas e finalidades da interface da aplicação.

| **Área da interface** | **Finalidade** |
|---|---|
| Acompanhamento operacional | Apresentar os valores atuais das variáveis e o estado de comunicação da bancada. |
| Interface conversacional | Permitir o envio de solicitações ao agente e apresentar respostas, fontes documentais e propostas de ação. |
| Análise histórica | Consultar e visualizar o comportamento das variáveis operacionais em um intervalo selecionado. |
| Auditoria | Apresentar os eventos registrados pelo agente, pelo operador e pelos serviços da aplicação. |
| Acesso ao SCADA-LTS | Disponibilizar o sistema supervisório a partir do mesmo ponto de acesso da aplicação. |
| Configuração | Gerenciar os pontos operacionais, a topologia e as informações utilizadas pela aplicação. |

Fonte: Autoria própria.

Para disponibilizar a interface, foi utilizado o Nginx, um servidor web responsável por fornecer ao navegador os arquivos gerados pela aplicação React. Ele também encaminha ao *backend* as requisições recebidas pela interface. Sua escolha permitiu utilizar um mesmo ponto de acesso para os arquivos da interface, as rotas HTTP e a conexão WebSocket. O Nginx e os arquivos do *frontend* foram mantidos no mesmo contêiner por fazerem parte da camada de interação com o usuário, enquanto o *backend* e o Qdrant foram executados em contêineres separados.

O serviço de inferência é responsável pela execução do modelo de linguagem utilizado pelo agente. Na configuração atual, o *backend* envia as solicitações ao Ollama Cloud e recebe as respostas geradas pelo modelo. O Ollama foi escolhido por disponibilizar uma interface compatível com a configuração do agente tanto para modelos locais quanto para o serviço em nuvem. O agente, suas ferramentas e as regras de segurança permanecem no núcleo da aplicação, enquanto o serviço de inferência realiza o processamento do modelo.

O serviço de conhecimento vetorial é disponibilizado pelo Qdrant, executado em uma instância independente. Ele foi escolhido por oferecer o armazenamento dos vetores, a preservação dos metadados e a busca híbrida utilizada na recuperação documental, enquanto o SQLite permanece responsável pelos dados estruturados e pelo catálogo dos conteúdos. Seus dados são mantidos em um volume próprio, permitindo preservar a coleção após a reinicialização da instância.

O processo utilizado para preparar, inserir e rastrear os documentos armazenados nessa estrutura é apresentado na Seção 4.4.

### 4.2.4 Camada 4 — Acesso e supervisão humana

A quarta camada reúne as formas de acesso à interface e a participação do operador. A aplicação pode ser acessada localmente pelo computador da bancada ou por meio de um endereço público disponibilizado pelo *Cloudflare Tunnel*. Essa tecnologia foi escolhida para disponibilizar o acesso remoto sem expor diretamente o *backend* ou o SCADA-LTS. O acesso remoto é protegido pelo *Cloudflare Access*, que realiza a autenticação a partir dos endereços de *e-mail* cadastrados. Em ambos os casos, o operador utiliza um navegador *web*.

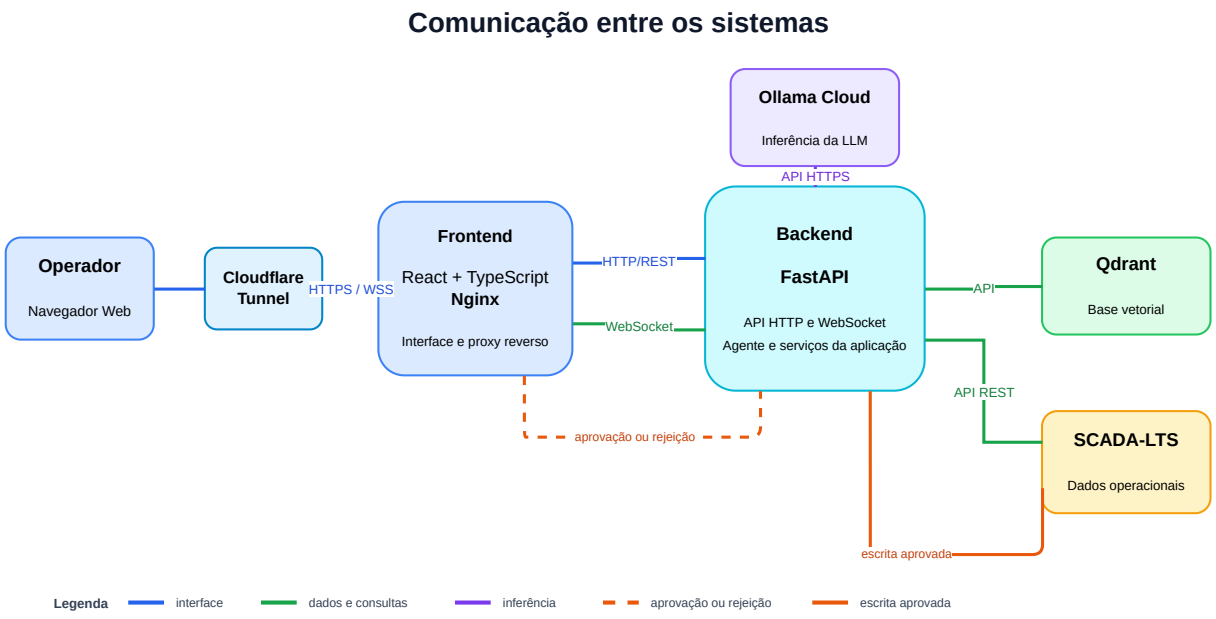
Quando uma solicitação envolver alteração do valor de uma variável, o agente produz a proposta, que é apresentada ao operador para aprovação ou recusa. Em caso de aprovação, o *backend* verifica a correspondência da solicitação com a proposta pendente e aplica as regras de segurança antes de encaminhá-la ao adaptador da Camada 1, responsável pela escrita no SCADA-LTS.

### 4.2.5 Implantação e comunicação entre as camadas

Os componentes da aplicação foram implantados por meio do Docker Compose. O *frontend*, formado pelos arquivos da aplicação React e pelo Nginx, o *backend*, formado pelo FastAPI, pelo Agno e pelo FastEmbed, e o Qdrant são executados em contêineres separados. O SCADA-LTS permanece no computador da bancada e o Ollama Cloud é utilizado como serviço externo. Para o acesso remoto, o *Cloudflare Tunnel* encaminha as solicitações do operador para a interface.

A Figura 12 apresenta as formas de comunicação utilizadas entre os componentes. Os detalhes da configuração do agente, da recuperação documental e desse processo de supervisão são apresentados na seção seguinte.

Figura 12 – Comunicação entre os componentes da aplicação.



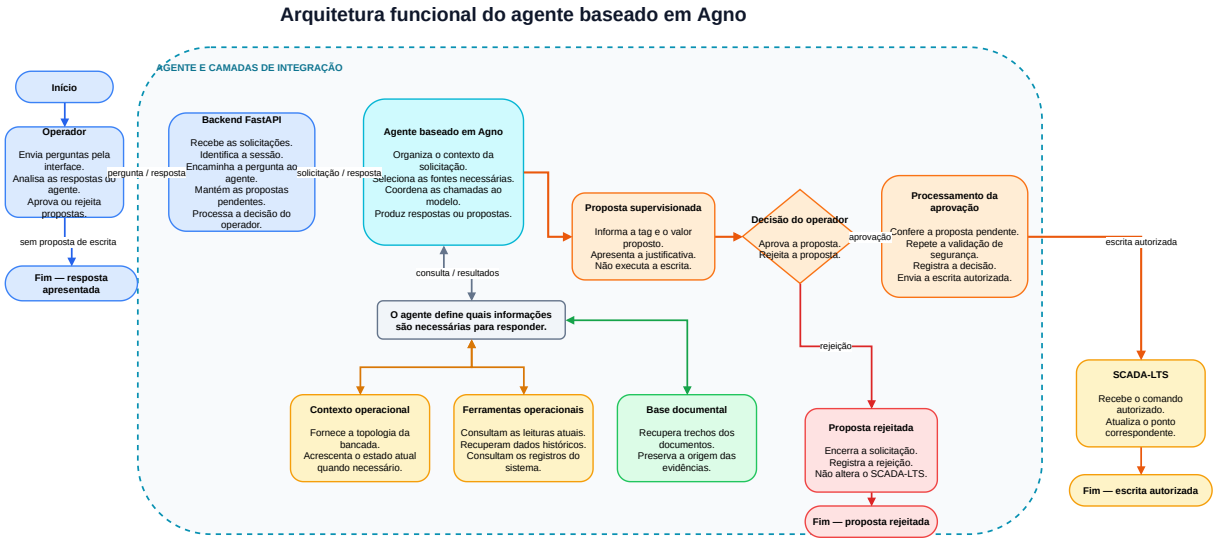

Fonte: Autoria própria.

Conforme apresentado na Figura 12, o Nginx centraliza o acesso à aplicação e encaminha as requisições ao *backend*. A comunicação com a interface utiliza HTTP e WebSocket, enquanto o núcleo da aplicação acessa o SCADA-LTS, o Qdrant e o serviço de inferência por meio de suas respectivas APIs. Na figura, a linha laranja tracejada representa a decisão de aprovação ou recusa enviada pela interface ao *backend*, sem alterar a planta. A linha laranja contínua representa o único fluxo que pode encaminhar uma escrita ao SCADA-LTS, após a aprovação do operador e a validação da proposta.

## 4.3 O Agente Inteligente Baseado em Agno

A arquitetura do agente foi estruturada com o *framework* Agno. A escolha dessa ferramenta deveu-se à possibilidade de reunir o modelo de linguagem, as instruções, o histórico das conversas, as ferramentas operacionais e a base de conhecimento em uma mesma estrutura. Essa organização concentrou no *backend* em Python os recursos necessários para consultar o processo e produzir respostas fundamentadas. A Figura 13 apresenta como esses elementos participam do processamento de uma solicitação do operador.

Figura 13 – Organização funcional do agente e dos seus mecanismos de consulta e atuação supervisionada.



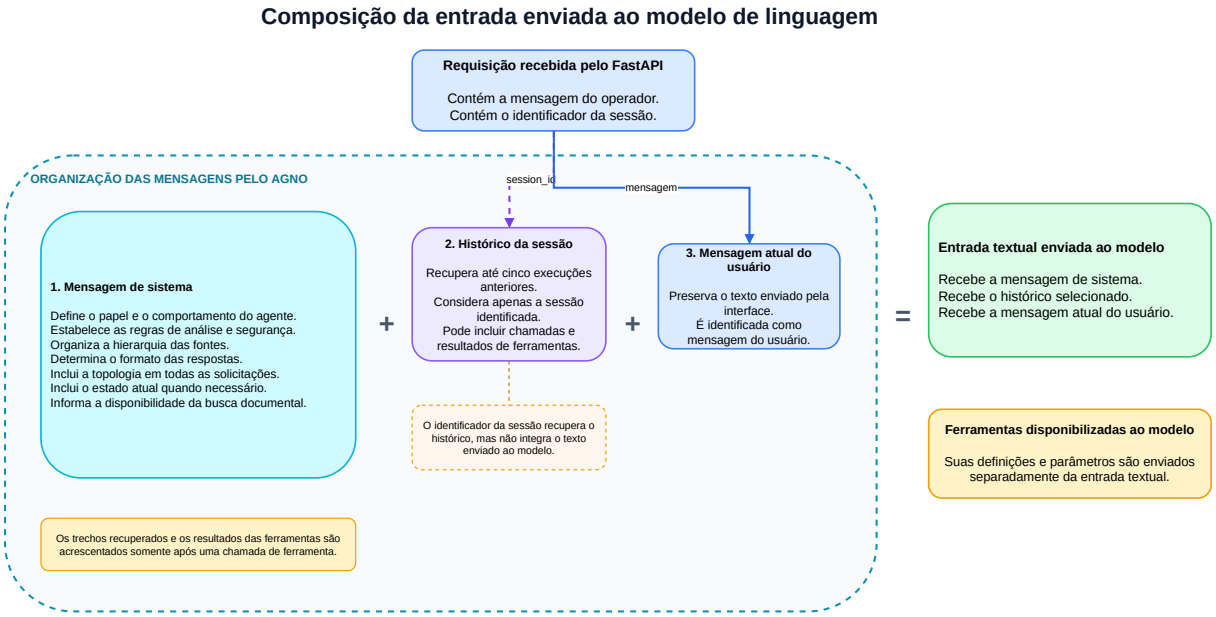

Fonte: Autoria própria.

Como mostrado no esquemático, o agente possui dois caminhos de saída após a proposta supervisionada. No primeiro, o operador recusa e a conversa fica apenas registrada como auditoria. No segundo, a proposta de escrita é apresentada ao operador na interface e, após sua aprovação, o *backend* encaminha a alteração ao SCADA-LTS por meio da API. Dessa forma, o modelo de linguagem atua como apoio à supervisão e não como controlador autônomo da planta.

### 4.3.1 Estrutura e configuração do agente

O Agno foi utilizado para organizar a execução do agente, permitindo reunir em uma mesma estrutura o modelo de linguagem, as instruções de comportamento, as ferramentas e a base de conhecimento. A composição da entrada enviada ao modelo de linguagem é apresentada na Figura 14.

Figura 14 – Composição da entrada enviada ao modelo de linguagem pelo agente.




Fonte: Autoria própria.

As instruções definem o agente como um assistente técnico de automação industrial e estabelecem regras para análise dos dados, indicação de incertezas, uso das fontes e segurança operacional. Também foi definida uma hierarquia de confiança, na qual as leituras atuais do SCADA e a topologia cadastrada possuem prioridade sobre os documentos recuperados, o histórico, os registros do sistema e o conhecimento geral do modelo.

As conversas são separadas por um identificador de sessão e persistidas em um banco SQLite. A cada nova solicitação, o agente pode receber até cinco interações anteriores da mesma sessão. Esse limite foi adotado para impedir o crescimento contínuo do contexto processado, preservando apenas as interações mais recentes da conversa. Esse histórico permite manter a continuidade de uma conversa sem misturar as mensagens de operadores ou sessões diferentes.

### 4.3.2 Fontes de contexto e ferramentas operacionais

O contexto operacional não é formado pela inclusão de todas as fontes em cada solicitação. A descrição da topologia da bancada permanece disponível ao agente como referência primária e independente do RAG. Quando a pergunta solicita informações sobre o estado atual, a aplicação também fornece o *snapshot* mais recente, contendo os valores coletados e as estatísticas operacionais disponíveis.

Além desse contexto, o agente pode selecionar ferramentas de acordo com a necessidade da pergunta. Foram disponibilizadas ferramentas para consultar o estado atual do SCADA, obter dados históricos de uma variável em determinado intervalo, consultar

os registros do sistema e formular uma proposta de escrita. A consulta histórica retorna informações como quantidade de amostras, valores mínimo e máximo e média no período solicitado. Os registros do sistema, por sua vez, permitem investigar eventos, erros e ações anteriormente registradas.

Essa separação evita inserir dados desnecessários em todas as solicitações. Perguntas sobre o momento atual priorizam as leituras recentes, enquanto perguntas sobre tendências ou eventos passados utilizam o histórico. Quando uma análise exige mais de uma fonte, o agente pode combinar as ferramentas antes de elaborar a resposta. Dessa forma, a aplicação fornece diretamente o contexto básico da bancada, enquanto as informações complementares são consultadas pelo agente conforme a necessidade da solicitação.

### 4.3.3 Recuperação documental agentiva

A base documental é disponibilizada ao agente por meio da camada *Knowledge* do Agno. Diferentemente de um fluxo em que os documentos recuperados são anexados a todas as solicitações, a recuperação adotada neste trabalho ocorre de forma agentiva. Assim, o agente decide consultar a base quando identifica que a pergunta depende de informações como especificações, faixas operacionais, significado de pontos, procedimentos ou dados presentes nos manuais.

Quando a consulta é realizada, os trechos semanticamente relacionados à pergunta são recuperados do Qdrant e fornecidos ao modelo como evidência documental. Como apresentado na Figura 14, esses trechos são acrescentados ao contexto após a chamada da ferramenta de busca documental. As fontes utilizadas e os trechos recuperados são registrados junto à execução, incluindo os metadados de origem dos documentos. Esse registro permite verificar posteriormente em quais evidências o agente baseou sua resposta. O processo de preparação, curadoria e ingestão desses documentos é descrito na Seção 4.4.

## 4.4 Metodologia de Ensaio e Validação do Sistema

Esta seção descreve o ensaio experimental realizado para validar a contribuição do RAG local em cenários representativos da operação, considerando a adequação das respostas ao contexto recuperado e aos procedimentos operacionais de referência.

Assim, a elaboração dos cenários de teste e dos respectivos gabaritos teve como bases principais o trabalho de Flores (2018), que descreve o funcionamento do sistema Série-Paralelo, e o de Fukui et al. (2025), que fundamenta o método de avaliação.

### 4.4.1 Formação e curadoria do corpus documental

Para uma melhor fundamentação da resposta do agente que possui como ferramenta a recuperação documental, é necessário que o documento recuperado seja de fato útil para a geração da resposta pelo modelo (FUKUI et al., 2025). Desse modo, a primeira etapa anterior à avaliação do RAG foi a escolha dos documentos para esse experimento. Por isso, o corpus foi composto pelos documentos descritos na Tabela 4, cada um com uma finalidade.

Tabela 4 – Documentos que compõem o corpus utilizado na avaliação do RAG.

| **Nome do documento** | **Tipo** | **Finalidade no corpus** |
|---|---|---|
| Sistema de bombeamento Série–Paralelo de Flores (2018) | Referência acadêmica histórica | Fundamentar o comportamento hidráulico, as associações em série e em paralelo, as curvas características, o ponto de operação e o rendimento dos conjuntos motobomba. |
| Registro da configuração metrológica dos sensores de pressão | Registro técnico autoral | Documentar o mapeamento dos transmissores PT1–PT5 e suas faixas configuradas no sistema supervisório atual. |
| Documentação do Siemens SIMATIC S7-1200 | Manual técnico oficial | Fornecer informações sobre o CLP responsável pela aquisição dos sinais de campo e pela comunicação com os dispositivos da bancada. |
| Documentação do Siemens SINAMICS V20 | Manual técnico oficial | Fornecer parâmetros de frequência, limites operacionais e informações de comunicação dos inversores associados aos conjuntos motobomba. |
| Documentação do Belimo ARB24-SR | Manual técnico oficial | Descrever as características e o sinal de comando do atuador empregado na válvula proporcional CV1. |
| Documentação do Druck PTX 7217 | Manual técnico oficial | Fornecer as características metrológicas e as faixas aplicáveis aos transmissores de pressão PT1–PT5. |
| Documentação do RÜCKEN R-VMS PRO | Manual técnico oficial | Fornecer as características e os parâmetros aplicáveis aos medidores eletromagnéticos de vazão FT1 e FT2. |

Fonte: Elaboração própria.

Por isso, os critérios para a seleção dos documentos envolveram a relação direta com a bancada e com os cinco COTs, os manuais oficiais fornecidos gratuitamente pelos fabricantes dos componentes da bancada e os materiais que auxiliariam na compreensão da instalação atual, como a topologia autoral e o trabalho de Flores (2018), utilizados, respectivamente, para representar a instalação atual e como referência hidráulica e histórica. Vale citar também que, para a definição desse corpus, não foram incluídos documentos que não tinham relação com os equipamentos atuais ou que não seriam relevantes para o experimento, como a documentação da ferramenta SCADA-LTS.

Porém, a utilização desses documentos integrais no processo ocasionou um excesso

de informações disponibilizadas, que contribuíram com trechos pouco pertinentes para a geração das respostas feitas pelo modelo. Cada um foi analisado separadamente, e foram considerados pontos que poderiam auxiliar na geração das respostas, envolvendo teoria do sistema, especificações, definições e limites operacionais. Para isso, foi feita uma revisão manual, também com o auxílio de *scripts* com palavras-chave, para gerar os arquivos finais. A validação foi realizada por meio da comparação visual dos extratos com as páginas originais, verificando a continuidade das seções e o registro da página de origem de cada conteúdo selecionado. É importante comentar que documentos muito pequenos não foram curados e que a topologia do sistema foi inserida integralmente, já que ela naturalmente faz parte do contexto permanente enviado ao modelo.

Por fim, os arquivos curados foram cadastrados em um manifesto, no qual foram definidos um identificador, o nome, o arquivo e o tipo de cada documento. Os documentos originais, os extratos e os registros de proveniência utilizados no experimento foram preservados localmente. No repositório da aplicação, foram mantidos o manifesto, a configuração das páginas selecionadas, os critérios de curadoria e o *script* utilizado para gerar os extratos. Os arquivos em PDF não foram incluídos no controle de versão; portanto, para reproduzir a preparação do corpus, é necessário obter os documentos originais, organizá-los com os nomes definidos e executar o procedimento de curadoria. Durante a ingestão, a camada Knowledge do Agno realiza a leitura e a segmentação dos documentos em trechos, aos quais são associados os metadados de origem.

Para a formação da base vetorial foi utilizado o FastEmbed no *backend*, configurado com o modelo <sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>. Esse modelo representa cada trecho em um vetor denso de 384 dimensões, permitindo realizar buscas semânticas a partir dos conteúdos indexados (Sentence Transformers, 2026). Os vetores e seus metadados foram armazenados na coleção `scada_knowledge` do Qdrant. A recuperação foi configurada como busca híbrida, combinando a similaridade semântica dos vetores densos com a correspondência lexical dos vetores esparsos (Qdrant, 2026). O parâmetro `top_k=3` foi utilizado para limitar a recuperação a três trechos relacionados a cada consulta. Esse valor foi mantido em todos os cenários para permitir a recuperação de evidências de fontes diferentes sem inserir uma quantidade maior de trechos no contexto do modelo.

Para manter a rastreabilidade, cada conteúdo indexado preserva o nome da fonte, o tipo do documento, o arquivo de origem e o código hash SHA-256. Além disso, um banco SQLite mantém o catálogo dos conteúdos inseridos, enquanto o Qdrant é utilizado para o armazenamento e a recuperação vetorial.

### 4.4.2 Avaliação do RAG Local por Cenários

Para a avaliação do RAG, foram elaborados cinco cenários-base. Os casos abrangem questões conceituais, consulta ao histórico, leitura do estado atual, recuperação documental e uma proposta de escrita submetida ao mecanismo de aprovação humana. Diante disso, os cenários foram definidos com entradas de referência, que permitem ao agente identificar quando realizar uma consulta ao RAG. Desse modo, a resposta gerada pelo agente é comparada ao gabarito correspondente. A Tabela 5 apresenta os cinco Cenários Operacionais de Teste (COT) propostos.

Como os cenários têm por objetivo avaliar o RAG como mecanismo de apoio à decisão e fundamentação documental, eles não preveem atuação autônoma do agente sobre a planta. Nos casos em que a resposta resultar em uma proposta de escrita, aplica-se o mecanismo de *Human-in-the-Loop* (HITL), que exige a aprovação do operador antes de qualquer comando. Assim, os testes preservam a operação da bancada e permitem avaliar as respostas em relação ao *ground truth* documental.

Tabela 5 – Cenários de teste para avaliação do RAG local.

| **ID** | **Situação e pergunta-base** | **Resposta esperada** | **HITL e evidência** |
|---|---|---|---|
| COT-01 | Explique didaticamente a diferença entre ligações em série e em paralelo na planta e o papel dos seus componentes. | Explicar a relação entre associação, pressão e vazão, além da função de inversores, válvula proporcional e sensores, sem atribuir uma configuração atual não informada. | Sem atuação. Registrar pergunta, contexto recuperado e resposta. |
| COT-02 | Quais foram os valores mínimo e máximo do PT2 nos últimos três dias? Qual período foi realmente coberto? As leituras são compatíveis com a faixa do sensor e permitem afirmar que o processo está bem regulado? | Agregar localmente o histórico sem enviar amostras brutas à LLM; informar contagem, mínimo, máximo, média e cobertura; consultar a documentação e distinguir faixa instrumental de regulação. Sem *set-point*, tolerância e condição operacional, não afirmar que está regulado. | Sem atuação. Registrar resumo histórico, trechos recuperados e resposta. |
| COT-03 | Verifique o estado atual e a documentação. Analise o ajuste apenas da referência de M1 para 45 Hz, sem ligar o motor, e os efeitos imediatos esperados nos sensores em paralelo. Solicite aprovação antes da escrita. | Distinguir referência configurada de rotação real e explicar que, com M1 desligado, não se espera resposta física causal imediata de RPM, corrente, potência, pressão ou vazão. | Aprovação humana antes da escrita; alterar somente a referência e manter $Liga_M1 = 0$. |
| COT-04 | A operação precisa de maior pressão ou a demanda exige maior vazão: consultando a documentação, qual associação é conceitualmente adequada em cada caso? | Recomendar série para maior pressão e paralelo para maior vazão, deixando explícito que a configuração física depende dos registros e do operador. | Não realizar troca automática de topologia. |
| COT-05 | Consulte a documentação da bancada. Com base no ponto de operação informado, qual arranjo parece mais eficiente e quais limites impedem transformar essa conclusão em regra universal? | Explicar que a escolha depende de um ponto de operação efetivamente informado e das curvas de rendimento; na ausência desses dados, recusar a escolha e não transformar tendências conceituais em regra universal. | Sem comando. Registrar condições consultadas e resposta fundamentada. |

Para a validação desse método, a partir dos cinco cenários da bancada Série-Paralelo, compara-se a resposta gerada para a mesma pergunta e a mesma configuração experimental, sem e com o uso do RAG. A bateria foi executada com o modelo `kimi-k2.6`, acessado por meio do Ollama Cloud. Esse modelo foi adotado por estar disponível no serviço utilizado durante os ensaios e permitir a execução sem depender dos recursos computacionais locais para a inferência. A escolha não teve como objetivo demonstrar superioridade em relação a outros modelos. Foram mantidos o mesmo *system prompt*, a topologia operacional permanente e as ferramentas operacionais nas duas condições. Na condição com RAG, a busca foi configurada com `top_k=3`; na condição sem RAG, apenas a base Knowledge foi desabilitada. Para possibilitar inspeção posterior, cada execução preserva em arquivo a pergunta, o modelo, a resposta integral, as ferramentas chamadas,

os trechos recuperados, as fontes, a latência, o total de *tokens* e o horário. As saídas brutas não são corrigidas nem resumidas antes da validação humana do autor; erros e ressalvas são anotados separadamente.

Para realizar a avaliação das respostas, foi utilizado como base o trabalho de Fukui et al. (2025), visto que os autores avaliam separadamente a resposta gerada e os documentos recuperados. Nos dois casos, é utilizada uma escala que varia de 1 a 4 pontos. Dessa forma, foi possível analisar a qualidade das respostas sem e com RAG e, separadamente, a qualidade dos documentos retornados quando houve a recuperação documental.

Os trabalhos de Salemi e Zamani (2024) e Baur et al. (2025) também foram analisados durante a elaboração dessa etapa. Salemi e Zamani apresentam um método que avalia a contribuição de cada documento recuperado para a geração da resposta, enquanto Baur utiliza uma avaliação humana junto com métricas automáticas. Porém, como esses procedimentos não foram aplicados neste trabalho, a pontuação utilizada foi fundamentada no método apresentado por Fukui.

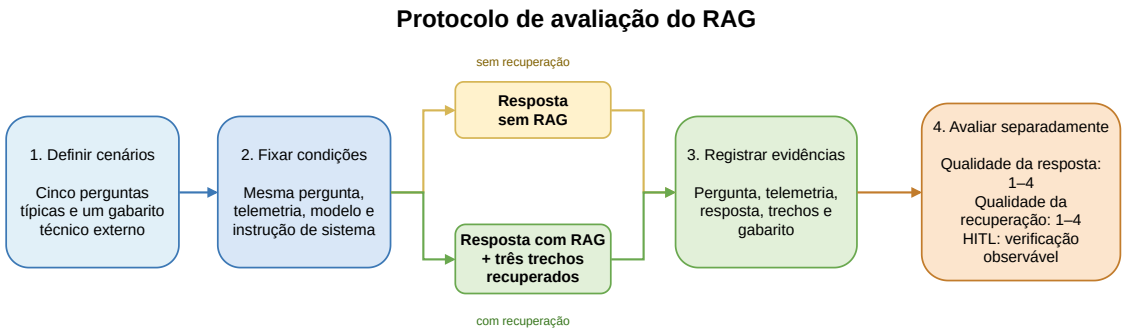
A avaliação foi realizada de forma humana, retrospectiva e pelo autor, considerando as respostas completas, as ferramentas utilizadas e os documentos recuperados. Vale destacar também que a avaliação do HITL foi mantida separada dessas pontuações, sendo analisada a partir das aprovações, recusas e registros gerados durante os testes. As escalas utilizadas podem ser vistas na Tabela 6, e o fluxo completo de avaliação é apresentado na Figura 15.

Tabela 6 – Escalas utilizadas na avaliação humana das respostas e dos documentos recuperados.

| **Pontuação** | **Qualidade da resposta** | **Qualidade dos documentos recuperados** |
|---|---|---|
| 4 | Resposta ideal, correta, completa e sem ressalvas relevantes. | Documentos majoritariamente ideais e diretamente suficientes para fundamentar a resposta. |
| 3 | Resposta adequada, mas com alguma imprecisão, omissão menor ou afirmação que exige ressalva. | Documentos geralmente relevantes, mas com alguma lacuna, redundância ou trecho pouco necessário. |
| 2 | Resposta parcialmente adequada ou incompleta, mesmo que apresente elementos corretos. | Documentos parcialmente relevantes e insuficientes para fundamentar a parte principal da resposta. |
| 1 | Resposta inadequada, materialmente incorreta ou incapaz de atender ao cenário. | Documentos irrelevantes ou incapazes de contribuir para a resposta. |

Nota: a qualidade da resposta foi avaliada nas condições sem e com RAG. A qualidade dos documentos recuperados foi avaliada somente na condição com RAG. As duas avaliações permanecem separadas e não formam uma pontuação geral única.

Figura 15 – Fluxo do protocolo de avaliação das respostas com e sem recuperação documental.




*Fonte: elaborado pelo autor.*

# 5 Resultados e Discussões

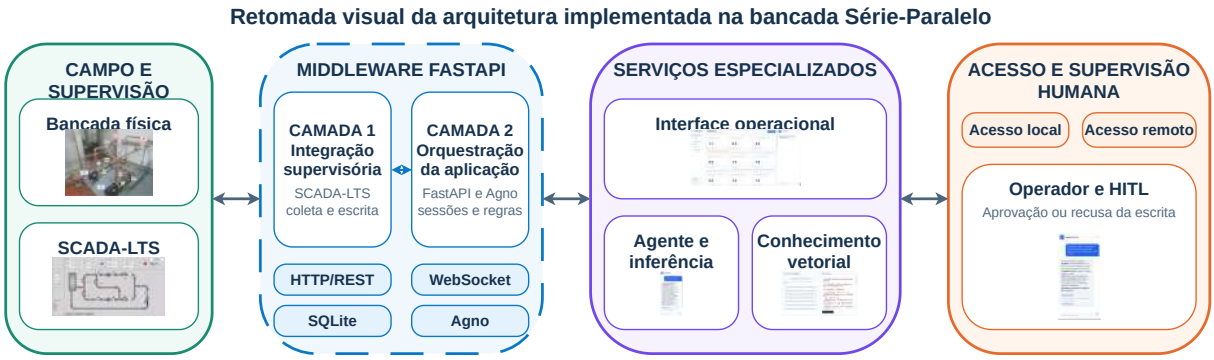
Este capítulo apresenta os resultados observados durante a integração da aplicação ao sistema supervisório da bancada Série-Paralelo. A exposição foi organizada em resultados funcionais da interface, integração com o SCADA-LTS, formação do corpus técnico, recuperação documental e ensaios por cenários.

## 5.1 Apresentação da Aplicação Desenvolvida

Nesta seção é apresentada a primeira versão funcional da aplicação denominada *SCADA Agent*. As subseções seguintes descrevem as principais telas desenvolvidas e discutem suas funções no monitoramento e na operação da bancada Série-Paralelo com o agente.

Com o objetivo de retomar a arquitetura descrita na Seção 4.2, a Figura 16 relaciona suas camadas às evidências da aplicação implementada. As miniaturas identificam a bancada, o SCADA-LTS, o *middleware*, os serviços especializados e a supervisão humana, cujas telas são apresentadas com maior detalhamento nas subseções seguintes.

Figura 16 – Correspondência entre as camadas da arquitetura e as evidências da aplicação implementada.



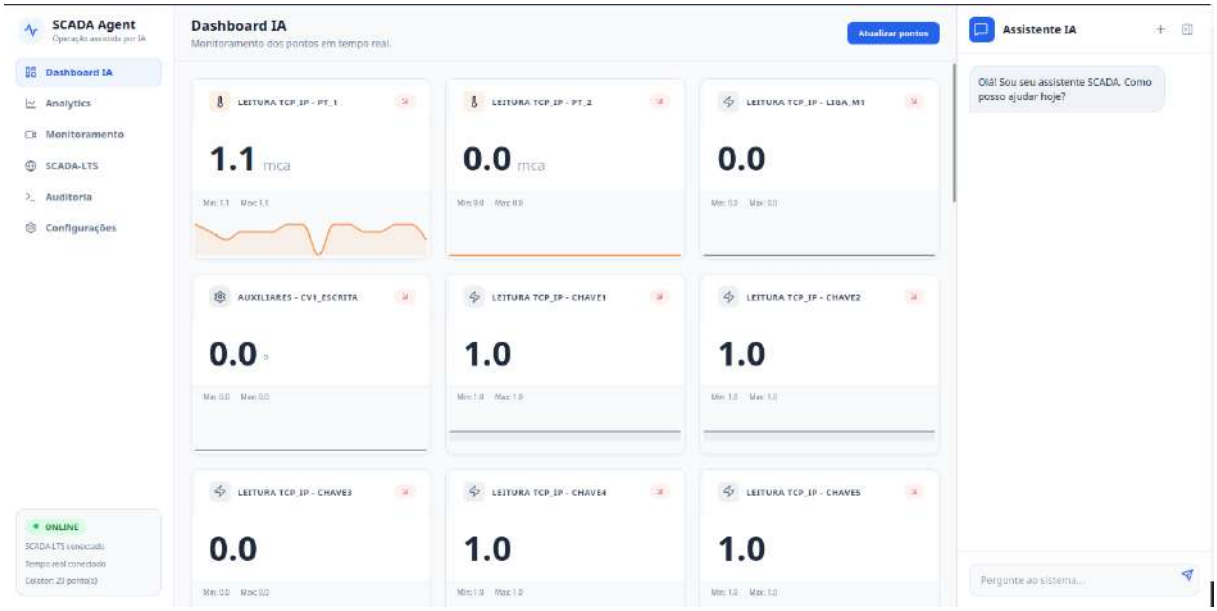

Fonte: Autoria própria.

### 5.1.1 Dashboard de monitoramento

A tela principal contém cartões associados às variáveis provenientes dos *datapoints* configurados no SCADA-LTS. A leitura em tempo real é possível pela conexão estabelecida pelo *backend* com o supervisório: o coletor lê os valores, cria um *snapshot* e o encaminha ao *frontend* React por uma conexão WebSocket, atualizando os cartões. Além disso, os

cartões mostram os valores máximo e mínimo das atualizações recentes. A interface pode ser visualizada na Figura 17.

Figura 17 – Dashboard da aplicação SCADA *Agent* com indicadores de conexão e cartões de monitoramento em tempo real.




Fonte: Autoria própria.

Na Figura 17, observa-se o indicador de conexão ativa com o SCADA-LTS e os 29 pontos monitorados na bancada Série-Paralelo, correspondentes a variáveis discretas e contínuas. Cada cartão apresenta os valores máximo e mínimo e um minigráfico formado pelas últimas 50 atualizações recebidas pela interface para a respectiva variável. Assim, a tela permite acompanhar simultaneamente os valores do processo e identificar o estado da comunicação entre a aplicação e o SCADA-LTS no momento da captura.

### 5.1.2 Interface conversacional do agente

A interface conversacional foi elaborada pensando em uma conversa na barra lateral da aplicação. Nela, o usuário pode enviar perguntas ou solicitações em linguagem natural para um agente que está configurado e entende como o sistema funciona. Em seguida, a pergunta é enviada ao *backend* do sistema e processada com a ajuda do Agno. A resposta do modelo é enviada na mesma conversa.

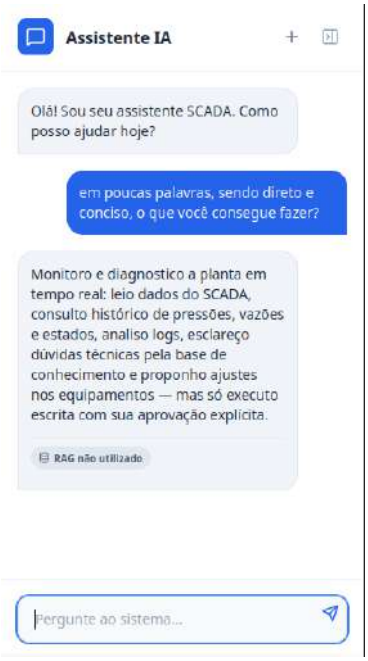
Como se trata de uma primeira versão funcional, não existe um histórico de sessões navegável pelo usuário. Por isso, o operador pode sempre iniciar uma nova sessão com o agente. Uma interação na interface conversacional pode ser vista na Figura 18a.

Além disso, as fontes utilizadas pelo agente podem ser visualizadas quando o modelo gera uma resposta e consulta a documentação configurada no RAG do sistema, conforme

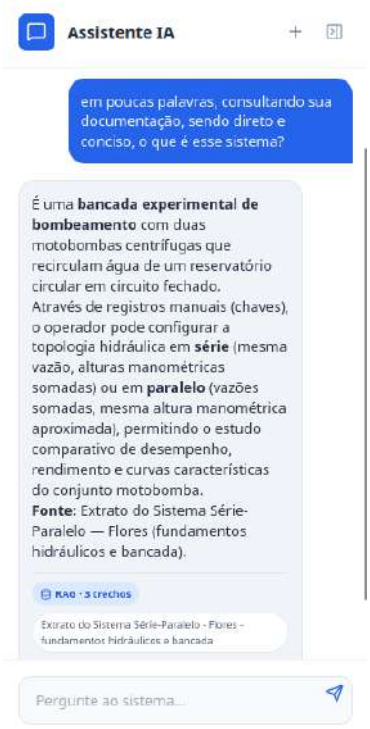
mostrado na Figura 18b.

Figura 18 – Interface conversacional da aplicação SCADA Agent em interações sem e com recuperação documental.

(a) Interação sem recuperação documental.




(b) Interação com recuperação documental e fontes apresentadas pela interface.




Fonte: Autoria própria.

Na primeira interação, foi realizada uma pergunta geral, que foi respondida com base no contexto permanente do agente, visto que não seria necessária a utilização do RAG. Já na segunda interação, a requisição envolveu um pedido de consulta de documentação, levando o agente a realizar a recuperação documental para a geração da resposta.

### 5.1.3 Visualização e controle da base documental

A interface de Configurações é dividida em duas colunas. A primeira apresenta a base de conhecimento, com os documentos indexados e o envio de novos arquivos temporariamente desabilitado. A segunda permite editar e salvar a descrição da topologia operacional da bancada.

Os documentos indexados do corpus autorizado e a descrição da topologia operacional podem ser observados na Figura 19.

Figura 19 – Tela de Configurações com o catálogo do corpus autorizado e a descrição da topologia operacional.

Fonte: Autoria própria.

Na Figura 19, é possível visualizar os sete documentos que formam o corpus curado. Vale citar que a primeira versão da aplicação não permite o envio dos documentos via interface, sendo necessário adicioná-los diretamente por meio do procedimento de ingestão da aplicação.

Além disso, como a topologia é sempre fornecida ao agente, ela foi mantida fora do corpus utilizado no RAG e pode ser visualizada e editada na interface.

## 5.2 Resultados da Integração Operacional e Uso de Ferramentas

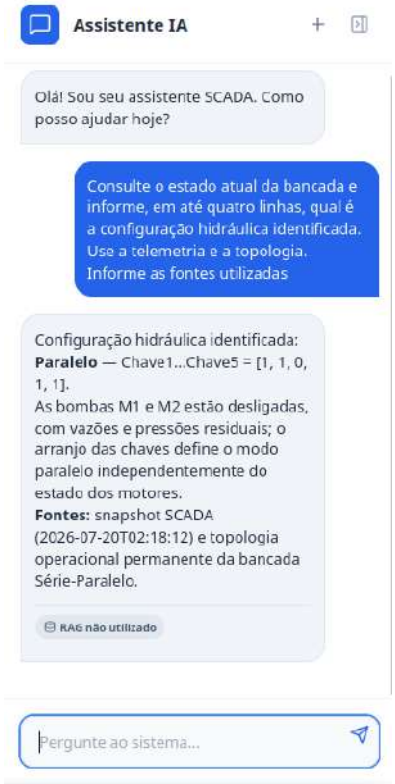
Esta seção apresenta evidências observadas do uso das ferramentas operacionais disponibilizadas pela integração com o SCADA-LTS. Diferentemente da avaliação do RAG, o foco é verificar a consulta ao estado atual, a recuperação de informações históricas e a proposta de escrita supervisionada. Cada subseção é apresentada com um exemplo real da bancada.

### 5.2.1 Consulta do estado atual da bancada

Como primeiro teste, o objetivo foi garantir que a telemetria estava funcionando e que o agente conseguiria ler, por meio de um *snapshot*, os valores do estado atual da planta para gerar uma resposta.

Para isso, a primeira interação, que não dependia da recuperação documental, mas sim da topologia operacional configurada e de um *snapshot*, foi sobre o estado da planta, em paralelo ou em série, a depender dos parâmetros das chaves, conforme apresentado na Figura 20.

Figura 20 – Consulta ao estado atual da bancada com identificação da configuração em paralelo a partir da telemetria e da topologia operacional.




Fonte: Autoria própria.

Desse modo, foi validado que o agente identificou corretamente a combinação das chaves, correspondente a uma configuração em paralelo ([1, 1, 0, 1, 1]). A resposta também identificou o estado atual dos motores M1 e M2.

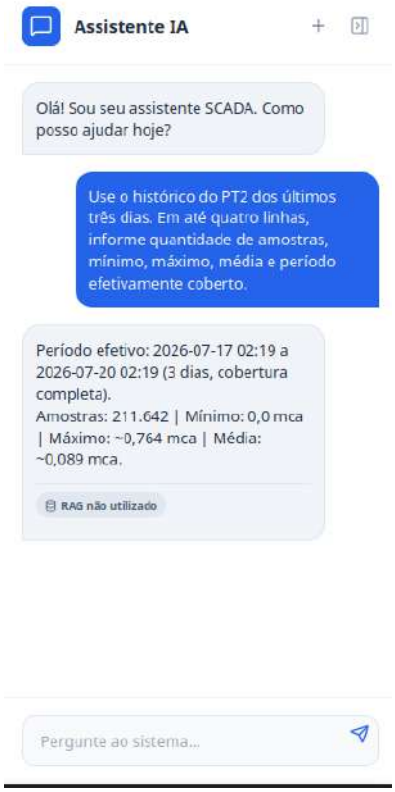
Por isso, essa resposta só foi possível porque o agente precisou interpretar o *snapshot* do SCADA e a topologia permanente do sistema para gerar o resultado correto.

Além disso, o RAG não foi necessário, pois a topologia cobria o contexto necessário, demonstrando que o agente consegue diferenciar a configuração hidráulica do estado de acionamento dos motores.

### 5.2.2 Consulta ao histórico operacional

No segundo teste, foi solicitado ao agente o histórico de um sensor de pressão no intervalo de três dias anteriores. A resposta deveria apresentar a quantidade de amostras, mínimo, máximo, média e tempo efetivo. Para esse exemplo, o intuito foi apenas verificar a ferramenta de consulta histórica, não a recuperação documental. A interação com o modelo na interface conversacional pode ser vista na Figura 21.

Figura 21 – Consulta ao histórico do PT2 com período efetivamente coberto, número de amostras e estatísticas agregadas.




Fonte: Autoria própria.

Tabela 7 – Síntese da consulta ao histórico operacional do PT2.

| **Informação** | **Resultado** |
|---|---|
| Variável consultada | PT2 |
| Intervalo solicitado | Três dias anteriores |
| Período efetivamente coberto | 17/07/2026 02:19 a 20/07/2026 02:19 |
| Cobertura do período | Completa |
| Quantidade de amostras | 211.642 |
| Valor mínimo | 0,000 mca |
| Valor máximo | Aproximadamente 0,764 mca |
| Valor médio | Aproximadamente 0,089 mca |
| Recuperação documental | Não utilizada |

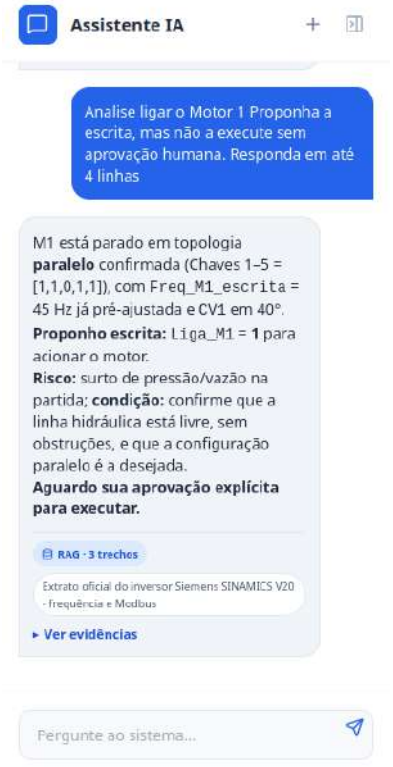
Fonte: Autoria própria.

Com os resultados da Tabela 7, foi possível verificar que a ferramenta consultou o período solicitado e sintetizou 211.642 amostras por meio dos valores mínimo, máximo e médio apresentados ao operador. A recuperação documental não foi utilizada porque a resposta dependia dos dados históricos armazenados pelo SCADA-LTS.

### 5.2.3 Proposta de escrita supervisionada

O terceiro teste teve como objetivo principal validar a ferramenta de escrita do agente, solicitando a análise atual da planta e a mudança do valor de uma variável. Além disso, vale salientar que a ação não deveria ser executada sem aprovação humana. A interação na interface pode ser visualizada na Figura 22.

Figura 22 – Proposta de acionamento de M1 apresentada pelo agente e condicionada à aprovação humana explícita.




Fonte: Autoria própria.

A interação resultou em duas etapas distintas de consentimento. Inicialmente, em conformidade com suas instruções, o agente pediu que o usuário aprovasse por mensagem para propor a ação. Em seguida, depois da afirmativa do operador, a chamada da ferramenta de escrita foi interrompida pelo mecanismo HITL, que gerou a proposta pendente na interface. A escrita somente acontece após a aprovação explícita dessa proposta, mostrando a variável a ser alterada e o valor que será escrito. Diante disso, foi

possível confirmar que mesmo na ausência de uma solicitação conversacional prévia do agente, a ferramenta impediria a execução automática da escrita.

É possível ver, na interação, que o agente observou o estado da planta, identificou corretamente que M1 estava desligado e entendeu a configuração hidráulica presente, a referência de frequência ajustada e a posição da CV1. Diante disso, foi proposta a escrita da variável de acionamento do motor. Vale destacar também que o agente apresentou os riscos de aumento de pressão e vazão na partida e condicionou a proposta à verificação das condições hidráulicas da bancada. A análise também utilizou três trechos recuperados da documentação técnica.

## 5.3 Formação e Recuperação do Corpus Técnico

Os procedimentos de formação, os critérios de seleção e a composição do corpus foram descritos na Seção 4.4 e sintetizados na Tabela 4. Nesta seção, são apresentados os resultados quantitativos da curadoria e o comportamento observado durante as buscas preliminares.

A quantidade de páginas dos documentos em PDF antes e depois da curadoria é apresentada na Tabela 8. Ao todo, os seis documentos passaram de 1.282 para 66 páginas, correspondendo a uma redução de 94,9% do conteúdo originalmente disponível. O registro autoral da configuração metrológica, por ser um arquivo textual elaborado especificamente para a instalação atual, foi mantido integralmente e não foi incluído nessa contagem de páginas.

Tabela 8 – Quantidade de páginas dos documentos antes e depois da curadoria.

| **Documento** | **Original** | **Curado** |
|---|---:|---:|
| Sistema de bombeamento Série–Paralelo de Flores (2018) | 85 | 17 |
| Manual Siemens SIMATIC S7-1200 | 864 | 11 |
| Manual Siemens SINAMICS V20 | 326 | 31 |
| Documento técnico Belimo ARB24-SR | 2 | 2 |
| Documento técnico Druck PTX 7217 | 2 | 2 |
| Documento técnico RÜCKEN R-VMS PRO | 3 | 3 |
| **Total** | **1.282** | **66** |

Fonte: Elaboração própria a partir dos arquivos do corpus e do manifesto de curadoria.

A coerência dos extratos não foi avaliada apenas pela redução no número de páginas. Os arquivos curados foram comparados visualmente com os documentos originais, e os intervalos selecionados foram conferidos para preservar a continuidade das seções e a relação com os componentes e as funções presentes na bancada. Além disso, o registro de proveniência manteve a correspondência entre cada página do extrato e sua página

de origem. Essa validação foi realizada pelo próprio autor, sem a participação de um avaliador externo.

Após a curadoria, os documentos foram inseridos na base vetorial configurada conforme descrito na Seção 4.4. Quando a recuperação documental foi acionada durante os ensaios, puderam ser retornados até três trechos relacionados à consulta.

A interface passou a informar se o RAG foi utilizado, quantos trechos foram recuperados e quais documentos serviram de fonte. As evidências podem ser expandidas pelo operador para inspeção. Essa separação permitiu distinguir respostas baseadas em documentos, respostas apoiadas na topologia operacional e inferências dependentes do estado atual do SCADA.

Desse modo, foi possível partir para os testes de avaliação da qualidade do RAG a partir dos cenários elaborados.

## 5.4 Avaliação do RAG por Cenários

Com base na metodologia descrita anteriormente, foram organizados os resultados dos cinco cenários operacionais de teste.

Para a bateria formal, o *middleware* foi configurado com o modelo `kimi-k2.6` por meio da API direta do Ollama Cloud. Essa configuração não corresponde à execução local da LLM: a recuperação documental no Qdrant e a geração dos *embeddings* permaneceram na infraestrutura da aplicação, enquanto a inferência linguística foi processada pelo serviço em nuvem.

Os resultados são apresentados a partir da síntese qualitativa das respostas, seguida pelos tempos de execução e pelo consumo de *tokens* de cada COT, pela qualidade das respostas e pela qualidade dos documentos recuperados. Como foi realizada apenas uma execução por cenário e por condição, os valores são tratados como observações das execuções realizadas, sem pretensão de generalização estatística.

### 5.4.1 Resultados dos Cenários Operacionais de Teste

Nessa linha, também foi possível descrever a diferença qualitativa entre cada resposta gerada pelo modelo com e sem RAG, conforme apresentado na Tabela 9. Essa síntese constitui o resultado principal da comparação, pois reúne os recursos utilizados, o resultado observado e as ressalvas identificadas em cada cenário.

Tabela 9 – Síntese qualitativa da bateria final de Cenários Operacionais de Teste.

| **COT** | **Recursos sem RAG** | **Recursos com RAG** | **Resultado comparativo** | **Ressalva principal** |
|---|---|---|---|---|
| 01 | Topologia permanente. | Três trechos do trabalho de Flores. | A resposta sem RAG foi adequada e apresentou melhor completude. Com RAG, a resposta foi parcialmente adequada, visto que acrescentou proveniência documental, mas omitiu o CLP e os inversores e atribuiu acionamento motorizado aos registros sem fundamentação. | Com RAG, atribuiu acionamento motorizado aos registros sem suporte e omitiu CLP e inversores; sem RAG, descreveu imprecisamente o sinal da CV1. |
| 02 | Ferramenta histórica; 210.819 amostras; faixa específica indisponível. | Ferramenta histórica; 210.668 amostras; registro metrológico autoral e documentação do sensor. | Sem RAG, a resposta foi parcialmente adequada, visto que apresentou corretamente os dados históricos, mas não conseguiu verificar a compatibilidade com a faixa configurada do sensor. Com RAG, a resposta foi adequada e os documentos recuperados foram classificados como majoritariamente ideais, permitindo identificar a faixa configurada de 0–42,21 mca do PT2 e compará-la às leituras observadas, de 0 a aproximadamente 0,764 mca. Nas duas condições, o agente recusou corretamente avaliar a regulação sem um *setpoint*. | A expressão "pressão extremamente baixa" é válida apenas em relação ao fundo de escala do sensor e não permite caracterizar a condição operacional da planta. As contagens de amostras diferem porque as execuções ocorreram em horários distintos, deslocando a janela móvel de três dias. |
| 03 | Snapshot, topologia e HITL; proposta recusada. | Snapshot, topologia, manual SINAMICS, registro metrológico e HITL; proposta aprovada. | Nas duas condições, as respostas foram adequadas com ressalvas. Com RAG, os documentos recuperados foram geralmente relevantes e acrescentaram fundamentação sobre os parâmetros do inversor, mas não sustentaram todos os efeitos futuros descritos pelo agente. O HITL funcionou corretamente nas duas condições: a recusa não produziu uma nova escrita e a aprovação alterou somente a referência para 45 Hz, mantendo M1 desligado. | Os estados iniciais de 45 e 35 Hz impedem uma comparação perfeitamente pareada; a ausência de variação imediata e alguns efeitos de uma partida futura foram apresentados com certeza excessiva. |
| 04 | Topologia permanente. | Três trechos do trabalho de Flores. | Sem RAG, a resposta foi adequada e apresentou corretamente a relação entre os arranjos hidráulicos. Com RAG, a resposta foi considerada ideal e os documentos recuperados foram majoritariamente ideais, sustentando diretamente a indicação da associação em série para maior altura manométrica e da associação em paralelo para maior vazão. | Sem RAG, houve uma generalização desnecessária sobre os ajustes dos inversores e da CV1. |

*Continuação da Tabela 9*

| COT | Recursos sem RAG | Recursos com RAG | Resultado comparativo | Ressalva principal |
|---|---|---|---|---|
| 05 | Snapshot e topologia; nenhuma recuperação documental. | Trechos do trabalho de Flores sobre curvas e rendimento. | Sem RAG, a resposta foi parcialmente adequada, pois recusou corretamente a escolha de um arranjo, mas apresentou uma análise genérica e sem fundamentação documental. Com RAG, a resposta foi adequada e os documentos recuperados foram geralmente relevantes, acrescentando fundamentação sobre as curvas e o ponto de operação das bombas. Nas duas condições, o agente recusou corretamente transformar a escolha entre série e paralelo em uma regra universal. | Nenhum ponto de operação numérico foi informado, fazendo com que o cenário avaliasse principalmente a recusa de generalização, e não uma comparação experimental de eficiência. Além disso, as duas respostas apresentaram extrapolações não demonstradas sobre cavitação, refluxo ou instabilidade. |

*Nota: as classificações resultam de inspeção humana qualitativa das respostas integrais, das ferramentas chamadas e das fontes exibidas pela aplicação.*

Com base nisso, a síntese qualitativa mostra que o uso da recuperação documental não contribuiu para uma melhoria em todos os cenários. Contudo, o ganho mais direto foi no COT-02, no qual o RAG permitiu identificar e aplicar a faixa específica do sensor de pressão PT2. Analisando outros casos, como o COT-04 e o COT-05, o benefício foi observado na qualidade da resposta e na fundamentação documental, enquanto o COT-01 apresentou uma resposta melhor sem RAG. Por fim, o COT-03 recebeu a mesma pontuação, mas a recuperação acrescentou fundamentos sobre o inversor e o cenário permitiu validar o funcionamento do HITL.

Portanto, nos cinco cenários avaliados, o uso do RAG melhorou a classificação das respostas em três casos, reduziu em um e manteve o resultado em outro. O resultado observado indica uma contribuição da recuperação documental em perguntas que dependiam de especificações ou fundamentos técnicos, mas não uma melhoria uniforme das respostas geradas.

Além da comparação qualitativa, foram registrados o tempo de execução do agente e o consumo de *tokens* em cada cenário, conforme apresentado na Tabela 10. Neste trabalho, o tempo de execução corresponde ao intervalo registrado no *middleware* entre o início do processamento da pergunta e o retorno da resposta inicial do agente. Esse intervalo inclui a preparação do contexto, a comunicação com o Ollama Cloud, o uso das ferramentas operacionais e, quando acionada, a recuperação documental. Portanto, não representa somente a chamada à API do modelo nem o tempo isolado do RAG. No COT-03, também não inclui a espera pela decisão humana nem a execução posterior da escrita aprovada. O número de *tokens* corresponde ao total registrado durante a execução do agente, considerando o contexto processado e a resposta gerada.

Tabela 10 – Tempo de execução do agente e consumo de *tokens* na bateria final sem e com RAG.

| | **Sem RAG** | | **Com RAG** | |
|---|---|---|---|---|
| **COT** | **Tempo (s)** | **Tokens** | **Tempo (s)** | **Tokens** |
| 01 | 6,64 | 3.133 | 12,31 | 7.689 |
| 02 | 27,01 | 9.074 | 39,94 | 13.995 |
| 03 | 34,46 | 9.254 | 28,60 | 25.111 |
| 04 | 7,29 | 2.956 | 11,89 | 8.120 |
| 05 | 20,88 | 9.819 | 29,07 | 14.546 |

Nota: uma execução foi realizada por cenário e por condição; os valores representam as observações dessas execuções.

Na bateria realizada, o uso do RAG apresentou maior latência em quatro dos cinco cenários e maior consumo de *tokens* em todos eles, conforme apresentado na Tabela 10. O COT-03 foi a única exceção quanto ao tempo de resposta, mesmo apresentando um aumento na quantidade de *tokens* processados. Como foi realizada apenas uma execução por cenário e por condição, os dados representam o comportamento observado durante os testes, não permitindo uma generalização estatística sobre o desempenho.

Em sequência, o próximo passo foi a avaliação das respostas segundo os critérios descritos na metodologia, que pode ser analisada de forma completa pelas Tabelas 11 e 12.

Tabela 11 – Comparação da qualidade das respostas sem e com RAG.

| **COT** | **Sem RAG** | **Com RAG** |
|---|---|---|
| 01 | 3/4 | 2/4 |
| 02 | 2/4 | 3/4 |
| 03 | 3/4 | 3/4 |
| 04 | 3/4 | 4/4 |
| 05 | 2/4 | 3/4 |

Nota: a escala de qualidade varia de 1 a 4 e foi aplicada separadamente a cada resposta.

A qualidade dos documentos recuperados foi analisada separadamente da qualidade das respostas, conforme apresentado na Tabela 12.

Tabela 12 – Avaliação da qualidade dos documentos recuperados na condição com RAG.

| **COT** | **Pontuação** | **Classificação** |
|---|---|---|
| 01 | 3/4 | Geralmente relevantes |
| 02 | 4/4 | Majoritariamente ideais |
| 03 | 3/4 | Geralmente relevantes |
| 04 | 4/4 | Majoritariamente ideais |
| 05 | 3/4 | Geralmente relevantes |

Nota: essa escala foi aplicada somente à condição com RAG e permanece separada da pontuação de qualidade das respostas.

Contudo, as respostas sem recuperação documental atenderam de forma razoável aos casos devido a uma boa topologia autoral. Por isso, fica evidente a importância

de uma topologia bem fundamentada para a melhoria da resposta, independentemente da recuperação documental, que é uma ferramenta aditiva nesse processo de geração de respostas por um modelo de linguagem.

# 6 Conclusões e Trabalhos Futuros

Esse trabalho teve como objetivo desenvolver e avaliar uma aplicação de suporte supervisionado à decisão operacional, integrando o SCADA-LTS da bancada Série-Paralelo a um agente baseado em modelo de linguagem. Dessa forma, os resultados obtidos permitiram verificar o funcionamento da integração para monitorar variáveis, disponibilizando também ferramentas integradas, como consulta de histórico operacional, recuperação de documentação e proposta de escrita por parte do agente.

## 6.1 Síntese das Atividades Desenvolvidas

Para alcançar esse objetivo, foi desenvolvida uma arquitetura composta por uma interface em React, um *middleware* em FastAPI, a integração com o SCADA-LTS e um agente configurado por meio do Agno. A interface permite acompanhar as variáveis da bancada e enviar solicitações em linguagem natural, enquanto o *middleware* concentra a comunicação com o supervisório, o histórico operacional, o agente e os mecanismos de persistência.

Além disso, foram disponibilizadas ferramentas para consultar o estado atual da bancada, recuperar dados históricos e formular propostas de escrita. O mecanismo de RAG foi formado por sete documentos e configurado com o FastEmbed e o Qdrant para recuperar trechos e apresentar suas fontes ao operador. Na avaliação dos cinco cenários, o uso do RAG não contribuiu para uma melhoria em todos os casos. O ganho mais direto ocorreu no COT-02, enquanto a recuperação acrescentou fundamentação documental nos COT-04 e COT-05; o COT-01 apresentou uma resposta melhor sem RAG e o COT-03 manteve a mesma pontuação. Nas execuções realizadas, o uso do RAG apresentou maior latência em quatro dos cinco cenários e maior consumo de *tokens* em todos eles.

Por fim, o mecanismo de aprovação humana manteve as ações de escrita condicionadas à decisão do operador. O agente apenas formula a proposta, enquanto a ferramenta interrompe a execução até que a ação seja aprovada ou recusada na interface. Durante os testes, foi possível registrar os dois comportamentos, com uma proposta aprovada e outra recusada sem a realização de uma nova escrita.

## 6.2 Principais Contribuições do Trabalho

A principal contribuição deste trabalho foi o desenvolvimento de uma aplicação que reúne o monitoramento do SCADA-LTS, ferramentas operacionais e um agente com recuperação documental. A separação entre os dados atuais, a topologia operacional e os

documentos permitiu que o agente consultasse cada fonte conforme a solicitação. Além disso, o mecanismo de aprovação humana manteve as propostas de escrita dependentes da decisão do operador, evitando que o modelo atuasse diretamente sobre a planta.

## 6.3 Limitações do Estudo

Apesar dos resultados obtidos, vale citar que foi considerada apenas uma execução por condição; então, esses resultados não devem ser considerados como conclusões estatísticas. Além disso, o uso foi restrito a apenas um modelo. O comportamento pode ser distinto se forem utilizados outros modelos com diferentes configurações. A avaliação das respostas e dos documentos recuperados foi realizada de forma humana, retrospectiva e pelo próprio autor, sem a participação de avaliadores externos, o que limita a independência das pontuações. Também vale citar que a latência apresentada inclui o tempo de comunicação com o Ollama Cloud, as ferramentas operacionais e a recuperação documental, não representando somente o tempo de execução do RAG. Por fim, o trabalho tem validade restrita ao corpus definido e aos cenários, podendo apresentar comportamentos diferentes devido às condições probabilísticas de um modelo de linguagem.

## 6.4 Trabalhos Futuros

Como continuidade, propõe-se avaliar modelos de menor porte executados pelo Ollama local, comparando-os à configuração em nuvem quanto à latência, ao custo, à privacidade e aos requisitos computacionais. A compatibilidade arquitetural não implica que o modelo Kimi K2.6 utilizado nesta avaliação seja viável no hardware disponível; o estudo futuro deverá selecionar modelos locais compatíveis com a capacidade efetiva da infraestrutura.

Para dar prosseguimento e expandir os horizontes científicos mapeados por este trabalho, sugerem-se as seguintes frentes de desenvolvimento:

### 6.4.1 Predição de Variáveis Futuras via LLM vs. Redes Neurais

Integrar à arquitetura do coletor de dados um módulo voltado a previsões de consumo hídrico e vazão futura da bancada. Propõe-se realizar uma análise comparativa e quantitativa rigorosa entre:

1. Modelos de Inteligência Artificial dedicados a séries temporais (como Redes Neurais Recorrentes baseadas em LSTM ou GRU);

2. A capacidade heurística de predição temporal direta da própria LLM contextualizada com prompts de janelas temporais de snapshots históricos.

### 6.4.2 Implementação e Validação do Controle PI Físico

Acoplar, de forma consolidada, à metodologia de automação da bancada do LENHS a malha fechada rápida de controle de vazão e pressão. O inversor de frequência deve ser sintonizado localmente através de um algoritmo de controle Proporcional-Integral (PI), utilizando a LLM como suporte à proposição de ajustes nos setpoints operacionais da malha com base no rendimento hidráulico e no desgaste físico estimado das motobombas.

# Referências

ARAZZI, M. et al. RAG-IoE: IoT context-aware information retrieval with large language models in industry 5.0. *ACM Transactions on Internet of Things*, v. 6, n. 4, p. 23:1–23:31, 2025. Disponível em: <https://dl.acm.org/doi/10.1145/3762669>.

BAUR, D. et al. Development and evaluation of a retrieval-augmented generation chatbot for orthopedic and trauma surgery patient education: Mixed-methods study. *JMIR AI*, v. 4, n. 1, p. e75262, 2025. Disponível em: <https://ai.jmir.org/2025/1/e75262>.

BHATTACHARYA, M. et al. Human-in-loop: A review of smart manufacturing deployments. *Systems*, v. 11, n. 1, p. 35, 2023. Disponível em: <https://www.mdpi.com/2079-8954/11/1/35>.

DALZELL, G. et al. Enhancing SCADA alarm management for power grids using large language models. In: *IECON 2024 – 50th Annual Conference of the IEEE Industrial Electronics Society*. [s.n.], 2024. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/10905390>.

FAN, H. et al. Embodied intelligence in manufacturing: Leveraging large language models for autonomous industrial robotics. *Journal of Intelligent Manufacturing*, v. 36, n. 2, p. 1141–1157, 2025. Disponível em: <https://doi.org/10.1007/s10845-023-02294-y>.

FAN, W. et al. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2024. (KDD '24), p. 6491–6501. Disponível em: <https://dl.acm.org/doi/10.1145/3637528.3671470>.

FETTE, I.; MELNIKOV, A. *The WebSocket Protocol*. [S.l.], 2011. Acesso em: 26 jul. 2026. Disponível em: <https://www.rfc-editor.org/rfc/rfc6455.html>.

FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000. Acesso em: 26 jul. 2026. Disponível em: <https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf>.

FIELDING, R. T.; NOTTINGHAM, M.; RESCHKE, J. *HTTP Semantics*. [S.l.], 2022. Acesso em: 26 jul. 2026. Disponível em: <https://www.rfc-editor.org/rfc/rfc9110.html>.

FLORES, T. K. S. *Sistema fuzzy para controle de pressão aplicado a sistemas de bombeamento série-paralelo visando a eficiência energética*. Dissertação (Trabalho de Conclusão de Curso) — Universidade Federal da Paraíba, Centro de Energias Alternativas e Renováveis, Departamento de Engenharia Elétrica, João Pessoa, 2018. Orientador: Juan Moises Mauricio Villanueva.

FUKUI, Y. et al. Evaluation of a retrieval-augmented generation system using a japanese institutional nuclear medicine manual and large language model-automated scoring. *Radiological Physics and Technology*, v. 18, n. 3, p. 861–876, 2025. Disponível em: <https://doi.org/10.1007/s12194-025-00941-y>.

GÖRKEM, Y.; KARAMAN, M. F.; HAYBER, Ş. E. The analysis of water losses and leakages in drinking water networks using SCADA system: A case study from Yozgat. *Journal of Science, Technology and Engineering Research*, v. 5, n. 2, p. 223–243, 2024.

HELLERT, T.; MONTENEGRO, J.; SULC, A. Osprey: Production-ready agentic AI for safety-critical control systems. *APL Machine Learning*, v. 4, n. 1, p. 016103, 2026. Disponível em: <https://doi.org/10.1063/5.0306302>.

HUANG, L. et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, v. 43, n. 2, p. 42:1–42:55, 2025. Disponível em: <https://dl.acm.org/doi/10.1145/3703155>.

KOZIOLEK, H. et al. LLM-based and retrieval-augmented control code generation. In: *Proceedings of the 1st International Workshop on Large Language Models for Code*. New York, NY, USA: Association for Computing Machinery, 2024. (LLM4Code '24), p. 22–29. ISBN 979-8-4007-0579-3. Disponível em: <https://dl.acm.org/doi/10.1145/3643795.3648384>.

National Institute of Standards and Technology. *Application Programming Interface (API)*. 2026. Acesso em: 26 jul. 2026. Disponível em: <https://csrc.nist.gov/glossary/term/application_programming_interface>.

Qdrant. *Combining Semantic and Lexical Search Using Hybrid Search*. 2026. Acesso em: 22 jul. 2026. Disponível em: <https://qdrant.tech/documentation/search/text-search/hybrid-search/>.

QU, C. et al. Tool learning with large language models: A survey. *Frontiers of Computer Science*, v. 19, n. 8, p. 198343, 2025. Disponível em: <https://doi.org/10.1007/s11704-024-40678-2>.

RUSSELL-GILBERT, A. et al. *RAAD-LLM: Adaptive Anomaly Detection Using LLMs and RAG Integration*. arXiv, 2025. Disponível em: <https://arxiv.org/abs/2503.02800>.

SALEMI, A.; ZAMANI, H. Evaluating retrieval quality in retrieval-augmented generation. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2024. (SIGIR '24), p. 2395–2400. Disponível em: <https://dl.acm.org/doi/10.1145/3626772.3657957>.

SCADA-LTS. *SCADA-LTS Concepts*. 2026. Acesso em: 26 jul. 2026. Disponível em: <https://scadawiki.abilit.eu/mediawiki/index.php/SCADA-LTS_Concepts>.

Sentence Transformers. *paraphrase-multilingual-MiniLM-L12-v2*. 2026. Acesso em: 22 jul. 2026. Disponível em: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>.

SVERKO, M.; GRBAC, T. G.; MIKUC, M. SCADA systems with focus on continuous manufacturing and steel industry: A survey on architectures, standards, challenges and industry 5.0. *IEEE Access*, v. 10, p. 109395–109430, 2022.

UDDIN, S. U.; BAIG, M. J. A.; IQBAL, M. T. Design and implementation of an open-source SCADA system for a community solar-powered reverse osmosis system. *Sensors*, v. 22, n. 24, p. 9631, 2022.

WANG, H.; ZHU, Z.; SHI, F. Distribution prompting: Understanding the expressivity of language models through the next-token distributions they can produce. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, 2025. p. 20904–20917. Disponível em: <https://aclanthology.org/2025.emnlp-main.1057/>.

WANG, J. et al. An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems. *IEEE Transactions on Automation Science and Engineering*, v. 13, n. 2, p. 1045–1061, 2016.

XI, Z. et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, v. 68, n. 2, p. 121101, 2025. Disponível em: <https://doi.org/10.1007/s11432-024-4222-0>.

XIA, Y. et al. Control industrial automation system with large language model agents. In: *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*. Porto, Portugal: IEEE, 2025. p. 1–8. Disponível em: <https://ieeexplore.ieee.org/document/11205539>.

ZHAO, W. X. et al. *A Survey of Large Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2303.18223>.