INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS GERAIS - CAMPUS OURO PRETO
PÓS-GRADUAÇÃO EM INTELIGÊNCIA ARTIFICIAL

CHRISTHIAN DA SILVA GONÇALVES

FRAMEWORK DE VALIDAÇÃO SEMÂNTICA DE ORDENS DE MANUTENÇÃO COM LLM E RAG

Ouro Preto, 2025

CHRISTHIAN DA SILVA GONÇALVES

FRAMEWORK DE VALIDAÇÃO SEMÂNTICA DE ORDENS DE MANUTENÇÃO COM LLM E RAG

Trabalho de conclusão de curso apresentado ao Curso de Pós-Graduação em Inteligência Artificial do Instituto Federal de Minas Gerais - Campus Ouro Preto para obtenção do grau de Especialista em Inteligência Artificial.

Orientador: Prof. Rodrigo Cesar Pedrosa Silva, Ph.D.

Ouro Preto
2025

G635f Gonçalves, Christhian da Silva.
Framework de Validação Semântica de Ordens de Manutenção com LLM e RAG [manuscrito] / Christhian da Silva Gonçalves. – 2025.
52 f. : il.

Orientador: Rodrigo Cesar Pedrosa Silva.
Trabalho de Conclusão de Curso (especialização) – Instituto Federal de Minas Gerais. *Campus* Ouro Preto, 2025.

1. Inteligência artificial. 2. Semântica. 3. Manutenção predial. I. Silva, Rodrigo Cesar Pedrosa. II. Instituto Federal de Minas Gerais. *Campus* Ouro Preto. III. Título.

CDU: 004.8

Catalogação: Kelly Cristiane Santos Morais - CRB-6/3217

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Ouro Preto
Diretoria de Pesquisa, Inovação e Pós-Graduação
Coordenação do Curso de Pós-graduação em Inteligência Artificial
Rua Pandiá Calógeras, 898 - Bairro Bauxita - CEP 35400-000 - Ouro Preto - MG
- www.ifmg.edu.br

FOLHA DE APROVAÇÃO

CHRISTHIAN DA SILVA GONÇALVES

FRAMEWORK DE VALIDAÇÃO SEMÂNTICA DE ORDENS DE MANUTENÇÃO COM LLM E RAG

Trabalho de Conclusão de Curso apresentado ao curso de ESPECIALIZAÇÃO EM INTELIGÊNCIA ARTIFICIAL, ofertado pelo Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ouro Preto, como parte dos requisitos para a obtenção do título de ESPECIALISTA EM INTELIGÊNCIA ARTIFICIAL.

Aprovado em 13 de fevereiro de 2025, pela Banca Examinadora:

Prof. Dr. Rodrigo César Pedrosa Silva - UFOP - Orientador

Prof. Dr. Moisés Henrique Ramos Pereira - IFMG - Campus Ribeirão das Neves

Prof. Dr. Carlos Alberto Severiano Júnior - IFMG - Campus Sabará

Ouro Preto, 21 de julho de 2025.

Documento assinado eletronicamente por **Moises Henrique Ramos Pereira**, **Professor**, em 22/07/2025, às 13:55, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Rodrigo César Pedrosa Silva**, **Usuário Externo**, em 22/07/2025, às 15:02, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Carlos Alberto Severiano Junior**, **Professor**, em 24/07/2025, às 14:09, conforme Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade do documento pode ser conferida no site https://sei.ifmg.edu.br/consultadocs informando o código verificador **2392001** e o código CRC **D672C27C**.

“O progresso não é sempre o resultado de grandes saltos tecnológicos, sendo também produto de incontáveis pequenas modificações e melhorias." Documento brasileiro à Conferência das Nações Unidas sobre Ciências e Tecnologia para o desenvolvimento.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelo amparo e pela vida. À minha família, pelo carinho e base sólida, seus conselhos e valores foram o alicerce sobre o qual construí, não apenas minha formação acadêmica, mas também minha visão de mundo. Em especial, à minha esposa Amanda, companheira incansável desta caminhada, pelo apoio inabalável nos momentos mais desafiadores, por compreender as ausências e celebrar cada pequena conquista ao meu lado. Sua presença foi força, equilíbrio e motivação constante ao longo desta jornada. Aos professores, pela excelência acadêmica, pelo comprometimento com a formação crítica e pela constante busca pela inovação. Um agradecimento especial ao meu orientador, Rodrigo C. P. Silva, e à coordenadora do curso de especialização em IA, Sílvia G. M. Almeida, pelo apoio e paciência, suas contribuições foram fundamentais para a concretização deste trabalho. Agradeço ao IFMG Ouro Preto pela oportunidade dos meus estudos e pelo apoio através do programa de pós-graduação, estrutura e investimento na educação. Enfim, agradeço a todos que, de alguma forma, contribuíram para a realização deste trabalho.

RESUMO

A validação semântica de ordens e notas de manutenção constitui um elemento importante para assegurar a integridade das informações que sustentam as decisões técnicas e estratégicas no ambiente industrial da mineração. A ausência de precisão conceitual e semântica nesses registros pode comprometer não apenas a confiabilidade dos ativos, mas também os indicadores de desempenho, os planos de manutenção e os processos de melhoria contínua. Nesse cenário, o desenvolvimento de ferramentas automatizadas e inteligentes que permitam auditar, interpretar e justificar esses dados de forma autônoma torna-se uma resposta necessária e estratégica frente ao crescente volume, complexidade e criticidade das informações operacionais. Este trabalho propõe e desenvolve um *framework* para a validação semântica automatizada de ordens e notas de manutenção, utilizando os *Large Language Models*(LLMs) ou Grandes Modelos de Linguagem, executados localmente, apoiado por uma arquitetura de *Retrieval-Augmented Generation*(RAG) ou Recuperação e Geração Aumentada. O sistema, implementado em *Python*, emprega o modelo *Gemma-3:4B* via *Ollama*, garantindo a privacidade dos dados, e utiliza a biblioteca *LangChain* para orquestrar a interação. Bases de conhecimento customizadas, extraídas de dados de regras de negócio, são consultadas para fornecer contexto específico do domínio ao LLM. Foram utilizadas técnicas de engenharia de *prompt*, como *Chain of Thought* (CoT) que força o LLM a gerar raciocínio para promover explicabilidade, bem como uma gestão de memória conversacional para otimizar a eficiência e reduzir o processamento de *tokens*. Este estudo contribui com um protótipo funcional, uma metodologia sistemática para o desenvolvimento de agentes de IA especialistas em análise da conformidade semântica, visando melhorar a qualidade dos dados e apoiar decisões mais precisas na gestão da manutenção.

Palavras-chaves: Análise, *Chain of Thought*, confiabilidade, *framework*, Inteligência Artificial, *LangChain*, *Large Language Models*, manutenção industrial, *Prompt*, raciocínio, *Retrieval-Augmented Generation*, semântica, *tokens*.

ABSTRACT

The semantic validation of maintenance orders and notes is an important element to ensure the integrity of the information that supports technical and strategic decisions in the mining industry. The lack of conceptual and semantic accuracy in these records can compromise not only the reliability of assets, but also performance indicators, maintenance plans and continuous improvement processes. In this scenario, the development of automated and intelligent tools that allow auditing, interpreting and justifying this data autonomously becomes a necessary and strategic response to the growing volume, complexity and criticality of operational information. This work proposes and develops a framework for the automated semantic validation of maintenance orders and notes, using Large Language Models (LLMs) or Large Language Models, executed locally, supported by a Retrieval-Augmented Generation (RAG) architecture. The system, implemented in Python, uses the Gemma-3:4B model via Ollama, ensuring data privacy, and uses the LangChain library to orchestrate the interaction. Custom knowledge bases extracted from business rules data are queried to provide domain-specific context to the LLM. Prompt engineering techniques such as Chain of Thought (CoT) that force the LLM to generate reasoning to promote explainability, as well as conversational memory management to optimize efficiency and reduce token processing, were used. This study contributes with a working prototype, a systematic methodology for developing AI agents that are experts in semantic compliance analysis, aiming to improve data quality and support more accurate decisions in maintenance management.

Key-words: Analysis, Chain of Thought, reliability, framework, Artificial Intelligence, LangChain, Large Language Models, industrial maintenance, Prompt, reasoning, *Retrieval-Augmented Generation*, semantics, *tokens*.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface (Interface de Programação de Aplicações)
CoT	Chain of Thought (Cadeia de Pensamento)
CPU	Central Processing Unit (Unidade Central de Processamento)
CSV	Comma-Separated Values (Valores Separados por Vírgula)
ERP	Enterprise Resource Planning (Planejamento de Recursos Empresariais)
ETL	Extração, Transformação e Carga
FGV	Fundação Getúlio Vargas
FRACAS	Failure Reporting, Analysis, and Corrective Action System
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
IA	Inteligência Artificial
IoT	Internet das Coisas (Internet of Things)
KPI	Key Performance Indicator (Indicador Chave de Desempenho)
LLM	Large Language Model (Grande Modelo de Linguagem)
MAIA	Manutenção Assistida por Inteligência Artificial
MDPI	Multidisciplinary Digital Publishing Institute
MRP	Material Requirements Planning
MTBF	Mean Time Between Failures (Tempo Médio Entre Falhas)
MTTR	Mean Time To Repair (Tempo Médio Para Reparo)
NBR	Norma Brasileira
OM	Ordem de Manutenção
PCM	Planejamento e Controle da Manutenção
PLN	Processamento de Linguagem Natural

RAG	Retrieval-Augmented Generation (Geração Aumentada por Recuperação)
RCA	Root Cause Analysis (Análise de Causa Raiz)
RCM	Reliability Centered Maintenance (Manutenção Centrada em Confiabilidade)
RLHF	Reinforcement Learning from Human Feedback
SAP PM	SAP Plant Maintenance (Módulo de Manutenção de Planta do SAP)
SLA	Service Level Agreement (Acordo de Nível de Serviço)
SSMA	Segurança, Saúde e Meio Ambiente

SUMÁRIO

1 INTRODUÇÃO

1.1 Formulação do problema

Nos últimos anos, a manutenção industrial vem incorporando avanços de Inteligência Artificial (IA) e automação como parte de sua transformação digital. À medida que a Indústria 5.0 emerge como sucessora colaborativa da Indústria 4.0, o paradigma desloca-se do "tudo automatizar" para uma abordagem que posiciona o ser humano como elemento central, atuando em sinergia com sistemas inteligentes. O próprio termo "Indústria 5.0" é descrito pela Abraman (2023a) como a reconciliação entre pessoas e tecnologia, onde "uma não substitui a outra, mas se complementam" para a obtenção de resultados sustentáveis. No âmbito da manutenção, essa evolução se materializa em integrar tecnologias como IA, robótica, Internet das Coisas (IoT) e realidade aumentada para que sistemas inteligentes atuem como coadjuvantes técnicos, processando dados de múltiplas fontes – como monitoramento de condição, históricos de ordens de serviço e manuais técnicos – para fornecer *insights* e recomendações assertivas aos profissionais, mantendo o elemento humano como peça central no processo decisório. Nesse cenário, ganha destaque a ideia de Manutenção Assistida por IA, definida como uma nova categoria de manutenção que une as tecnologias da Indústria 4.0 de forma integrada para direcionar os mantenedores de modo mais assertivo na resolução de problemas (ABRAMAN, 2023b). Essa evolução conceitual acompanha a necessidade de aumentar a confiabilidade e eficiência operacional, ao mesmo tempo em que se preserva o papel crítico do expertise humano na gestão de ativos.

Paralelamente a esses avanços, persiste um desafio fundamental na engenharia de manutenção: a qualidade dos dados registrados nas ordens e notas de manutenção. Em sistemas corporativos como o SAP AG (2025) – amplamente utilizado para gestão de ordens de serviço – cada ordem de manutenção contém campos que devem ser preenchidos segundo boas práticas da engenharia de manutenção (descrição clara do problema, causa raiz, ações executadas, códigos de falha, prioridade, etc.). Entretanto, na prática, muitas ordens apresentam informações incompletas, inconsistentes ou erroneamente classificadas. Erros humanos comuns no preenchimento de campos textuais geram distorções significativas em indicadores de falha e confiabilidade, gerando esforço adicional dos analistas e engenheiros para interpretar resultados, reduzindo a confiabilidade das decisões tomadas com base nesses dados (CONTE et al., 2021). Além disso, é frequente que as notas de manutenção (registros textuais de falhas e intervenções) contenham jargões técnicos, abreviações e omissões de contexto, dificultando a compreensão posterior por outros profissionais (SEXTON et al., 2017). Esse problema de qualidade dos registros impacta diretamente no aprendizado organizacional, melhoria contínua e na capacidade de evitar recorrências de falhas em equipamentos. Normas de manutenção e gestão de ativos como a (STANDARDIZATION, 2016) que padroniza a coleta de dados de confiabilidade e

manutenção – ressaltam a importância de registros completos e consistentes. Contudo, garantir essa conformidade manualmente, em grande volume de ordens, é difícil e consome tempo, exigindo revisão humana minuciosa de cada registro.

Os *Large Language Models* (LLMs) - Grandes modelos de linguagem são modelos computacionais projetados para compreender e gerar textos. Estes modelos representam uma subárea do aprendizado de máquina, especificamente do Processamento de Linguagem Natural (PLN), e são caracterizados por sua escala massiva, tanto em termos do volume de dados com os quais são treinados, quanto no número de parâmetros que compõem sua arquitetura denominada como *Transformers* (LI et al., 2024). Os LLMs se destacam na identificação de padrões de linguagem complexos, permitindo um desempenho notável em uma variedade de tarefas de processamento de linguagem natural. Embora os LLMs representem um salto tecnológico, questões éticas, perpetuação de vieses através dos dados de treinamento e a intensiva necessidade de recursos computacionais persistem como áreas críticas, exigindo consideração cuidadosa para assegurar uma implantação imparcial e sustentável (RAZA M., 2025).

Embora os LLMs apresentem capacidade inerente de generalização, suas respostas podem carecer de exatidão factual ou refletir vieses quando o domínio exige conhecimento específico. A arquitetura *Retrieval-Augmented Generation* (RAG) permite que os LLMs, que já possuem um vasto conhecimento pré-treinado, acessem e utilizem informações específicas e atualizadas de uma base de conhecimento externa (como manuais técnicos, históricos de falhas, critérios e regras de negócio). Isso mitiga o risco de "alucinações" e garante que as respostas sejam fundamentadas em dados relevantes para um domínio específico (LEWIS et al., 2021).

Nesse contexto, o presente trabalho aborda o seguinte problema: como validar automaticamente, de forma sistemática e em larga escala, a qualidade e a conformidade das ordens de manutenção registradas em sistemas corporativos da indústria de mineração, utilizando modelos de linguagem de grande porte (LLMs) aliados à arquitetura de Recuperação e Geração Aumentada (RAG)? A dificuldade reside na necessidade de identificar inconsistências e na validação do conteúdo textual quanto à sua completude, coerência e conformidade com normas técnicas, procedimentos operacionais e boas práticas de engenharia de manutenção, sem depender exclusivamente de revisões humanas exaustivas e suscetíveis a erro.

Assim, o objetivo deste TCC é propor e desenvolver um *framework* que explore o potencial dos LLMs integrados com RAG para realizar a validação semântica automatizada de ordens de manutenção, garantindo que os registros estejam completos, coerentes e em conformidade com as normas de engenharia de manutenção e gestão de ativos. O *framework* visa reduzir o esforço manual, melhorar a confiabilidade dos dados e, consequentemente, apoiar decisões mais precisas e fundamentadas no contexto da manutenção assistida por inteligência artificial.

1.2 Justificativa do trabalho

A qualidade dos registros de manutenção é um fator crítico para a eficiência operacional, a segurança e a sustentabilidade dos processos industriais, especialmente em setores de alta complexidade e risco, como a indústria de mineração. Entretanto, a prática evidencia um problema recorrente e de grande impacto: o preenchimento inadequado das ordens de manutenção, com informações incompletas, inconsistentes ou imprecisas, comprometendo a confiabilidade das análises subsequentes e dificultando a tomada de decisão informada.

Tradicionalmente, a validação e a correção desses registros dependem de revisões manuais realizadas por especialistas, um processo moroso, suscetível a erros e economicamente inviável frente ao grande volume de ordens geradas diariamente nas operações industriais. Esse gargalo operacional evidencia a necessidade de soluções tecnológicas que automatizem e qualifiquem esse processo, reduzindo a sobrecarga dos profissionais e aumentando a confiabilidade dos dados.

Neste contexto, a aplicação de Modelos de Linguagem de Grande Porte (LLMs), aliados à arquitetura de Recuperação e Geração Aumentada (RAG), representa uma inovação disruptiva. A capacidade dos LLMs de interpretar linguagem natural com alta profundidade semântica, combinada com o acesso a bases de conhecimento específicas por meio do RAG, possibilita a validação automatizada das ordens de manutenção com contextualização técnica, mitigando o risco de interpretações incorretas ou de "alucinações" comuns aos modelos tradicionais.

Portanto, este trabalho justifica-se pela necessidade de desenvolver uma abordagem capaz de integrar essas ferramentas de Inteligência Artificial aos processos de gestão da manutenção. A proposta de um *framework* de validação semântica de ordens de manutenção com LLM e RAG busca não apenas aumentar a eficiência operacional e a confiabilidade dos registros, mas também contribuir para a transformação digital sustentável da indústria de mineração, alinhada aos princípios da Indústria 5.0, que valoriza a colaboração sinérgica entre humanos e sistemas inteligentes.

Além disso, o trabalho se justifica pela sua relevância acadêmica e prática, ao explorar e aplicar técnicas de ponta em Processamento de Linguagem Natural (PLN) no contexto específico da engenharia de manutenção, área ainda carente de soluções robustas baseadas em IA para o tratamento automatizado de dados textuais. A proposta contribui para a evolução das práticas de manutenção assistida por IA, ampliando as possibilidades de aplicação dos LLMs em domínios industriais críticos e, ao mesmo tempo, fornecendo um referencial metodológico para futuras pesquisas e desenvolvimentos na área.

1.3 Objetivos geral e específicos

Desenvolver um *framework* baseado em Modelos de Linguagem de Grande Porte (LLMs) integrados com a arquitetura de Recuperação e Geração Aumentada (RAG) para realizar a

validação semântica automatizada de ordens de manutenção registradas em sistemas corporativos de gestão de ativos industriais da mineração.

Sendo os objetivos específicos:

- Mapear e formalizar os requisitos de qualidade e conformidade.
- Projetar a arquitetura do *framework*, definindo as etapas de pré-processamento dos textos, integração com bases de conhecimento externas via RAG, mecanismos de inferência e validação semântica, além de critérios de avaliação dos resultados.
- Implementar um protótipo funcional do *framework*.
- Analisar os resultados obtidos

2 FUNDAMENTOS

O presente capítulo aborda a fundamentação teórica estudada, que guia o trabalho ao longo de seu desenvolvimento para que o devido trabalho se sustente teoricamente em relação ao tema abordado.

2.1 Manutenção industrial

A definição de manutenção, na literatura técnica, evoluiu de acordo com os desafios enfrentados pelas organizações ao longo do tempo. Segundo a NBR 5462 (ABNT, 1994), manutenção é a combinação de ações técnicas e administrativas destinadas a manter ou recolocar um item em um estado no qual possa desempenhar sua função requerida após sofrer uma falha ou defeito. Em outras palavras, envolve ações planejadas para garantir que máquinas, sistemas e componentes cumpram suas funções de forma confiável e segura. É entendido por função requerida o conjunto de condições de funcionamento para o qual o equipamento foi projetado, fabricado ou instalado. Falha é toda alteração física ou química no estado de funcionamento do equipamento que impede o desempenho de sua função requerida e o leva invariavelmente à indisponibilidade. O defeito é toda alteração física ou química no estado de funcionamento de um equipamento que não o impede de desempenhar sua função requerida, podendo ele operar com restrições.

(PINTO ALAN KARDEC; XAVIER, 2013) ampliam o entendimento da manutenção industrial ao contextualizá-la como uma função estratégica, destacando que ela deve garantir a disponibilidade da função dos equipamentos e instalações, com foco na confiabilidade, segurança, sustentabilidade ambiental e otimização dos custos operacionais ao longo do ciclo de vida do ativo. Eles argumentam que a manutenção deve ser planejada com base nos objetivos do negócio e integrada às demais áreas da empresa.

A manutenção moderna busca, de forma proativa, garantir que os ativos operem continuamente dentro dos padrões de desempenho esperados, visando otimizar o retorno sobre o investimento. Isso implica em tomar decisões baseadas em dados, considerando os riscos associados às falhas (Manutenção Baseada em Risco - MBR) e o impacto no negócio como um todo. A função manutenção evoluiu para um parceiro estratégico na busca por disponibilidade, confiabilidade, segurança e eficiência, contribuindo diretamente para a competitividade e sustentabilidade da organização (PINTO E XAVIER, 2018).

Nesse sentido, a manutenção deixa de ser apenas uma atividade técnica para se tornar um elemento essencial de gestão estratégica de ativos. Sua efetividade está diretamente relacionada ao grau de integração com os processos organizacionais, à qualidade dos dados analisados e à maturidade dos métodos empregados, sendo considerada um dos pilares da excelência

operacional na indústria contemporânea.

2.1.1 Evolução da Manutenção

A evolução da manutenção industrial, impulsionada pela Revolução Industrial e a subsequente necessidade de otimizar a capacidade produtiva, pode ser periodizada em cinco gerações distintas, conforme proposto por (PINTO ALAN KARDEC; XAVIER, 2013). Cada geração reflete avanços tecnológicos e novas concepções sobre a gestão de falhas e a performance dos ativos.

A Primeira Geração (até aproximadamente 1945), denominada "Mecanização", caracterizou-se por indústrias pouco mecanizadas com equipamentos simples e superdimensionados. A manutenção era predominantemente corretiva e não planejada, realizada apenas após a ocorrência de falhas, com o entendimento de que estas eram um resultado natural do desgaste.

A Segunda Geração (pós-Segunda Guerra Mundial, estendendo-se até a década de 1970), ou "Industrialização", foi marcada pela disseminação de linhas de produção contínuas e aumento da demanda. Isso exigiu maior disponibilidade dos equipamentos, levando ao desenvolvimento da manutenção preventiva, baseada no tempo. No entanto, os custos associados e a introdução de grandes computadores para planejamento e controle representaram desafios iniciais.

A Terceira Geração (a partir da década de 1970), referida como "Automatização", coincidiu com rápidas mudanças nos processos industriais, a adoção de tecnologias operando no limite da capacidade e a implementação de sistemas just-in-time. A criticidade das paradas de produção impulsionou a busca por maior confiabilidade e disponibilidade, fomentando a manutenção preditiva, o monitoramento da condição, a análise de risco e o uso de softwares especializados.

A Quarta Geração (aproximadamente a partir dos anos 2000) consolidou e expandiu os conceitos da anterior, com foco em maior confiabilidade e disponibilidade, preservação ambiental, segurança e o impacto da manutenção nos resultados do negócio. Houve uma intensificação do uso de técnicas de manutenção preditiva e monitoramento da condição para reduzir falhas prematuras e otimizar intervenções.

Finalmente, a Quinta Geração (a partir de 2010), centrada na "Gestão de Ativos", enfatiza os resultados empresariais e a competitividade. A manutenção objetiva a máxima capacidade produtiva sem falhas não previstas, por meio do monitoramento da condição on-line e off-line, um enfoque no ciclo de vida completo dos ativos, e a busca pela excelência em Engenharia de Manutenção, incluindo a consolidação da contratação por resultados. Esta evolução demonstra uma transição de uma abordagem reativa para estratégias proativas e integradas à gestão do negócio.

2.1.2 Tipos de manutenção

A manutenção industrial pode ser classificada de acordo com critérios técnicos e estratégicos, com base em diagnósticos operacionais, históricos de falha e dados em tempo real. Cada abordagem tem aplicação prática distinta e contribui, em maior ou menor grau, para a redução de paradas não programadas, otimização de recursos e aumento da vida útil dos ativos. .

2.1.2.1 Manutenção Corretiva

A manutenção corretiva é aquela realizada após a ocorrência de uma falha. Pode ser dividida conforme o nível de urgência e planejamento:

- Manutenção Corretiva Adiada: é à correção de falhas que não comprometem de forma imediata a segurança ou operação da planta. A intervenção é postergada e inserida em uma programação planejada, minimizando impacto na produção. É comum em ativos secundários e pouco críticos (PINTO ALAN KARDEC; XAVIER, 2013).
- Manutenção de Emergência: exige intervenção imediata, pois a falha é crítica e compromete a segurança, o meio ambiente ou a continuidade operacional. Geralmente, tem alto custo, maior tempo de inatividade e consumo excessivo de recursos.Essa forma de manutenção reflete deficiências em estratégias preventivas e exige análise para evitar recorrência.

2.1.2.2 Manutenção Preventiva

A manutenção preventiva é uma abordagem planejada que visa evitar falhas, preservar a confiabilidade dos ativos e reduzir o tempo de inatividade. De acordo com (PINTO ALAN KARDEC; XAVIER, 2013), envolve a realização sistemática de tarefas com base em critérios estabelecidos, como tempo de operação, número de ciclos ou medições de desgaste. Dentro da manutenção preventiva, há várias subcategorias:

- Manutenção Baseada em Tempo: intervenções agendadas com base em intervalos fixos de tempo ou uso (horas de funcionamento, quilômetros rodados, etc.)
- Manutenção de Encontro de Falha: utilizada para detectar falhas latentes, principalmente em sistemas de proteção ou dispositivos que permanecem inativos até que uma falha ocorra (ex.: alarmes, válvulas de segurança). Como esses sistemas não operam continuamente, suas falhas só são descobertas mediante testes específicos.
- Manutenção Baseada em Risco: essa abordagem prioriza a manutenção com base na criticidade do ativo e no risco associado à sua falha, considerando variáveis como impacto ambiental, segurança e custo. Segundo (MOBLEY, 2008), permite alocar recursos de

forma estratégica, concentrando esforços nos ativos que mais ameaçam a continuidade operacional e segurança.

- Manutenção Baseada em Condição: monitora parâmetros operacionais (vibração, temperatura, ruído, etc.) e determina o momento ideal para realizar uma intervenção.

2.1.2.3 Manutenção Preditiva

A manutenção preditiva é uma evolução da Manutenção Baseada em Condição e utiliza análise de dados históricos e atuais, combinada com tecnologias como termografia, ultrassom, análise de vibração e análise de óleo. O objetivo é prever quando uma falha ocorrerá e intervir antes disso.

2.1.3 Gestão da Manutenção

A gestão da manutenção é uma função estratégica fundamental para a sustentabilidade, produtividade e competitividade das organizações industriais, principalmente em ambientes com alta dependência de ativos físicos complexos. Seu objetivo principal é garantir a disponibilidade, confiabilidade e segurança operacional dos equipamentos, por meio da aplicação estruturada de políticas, métodos, tecnologias e práticas de engenharia. Ao otimizar o uso de recursos humanos, materiais e financeiros, a gestão da manutenção busca minimizar o impacto das falhas, aumentar a eficiência dos processos produtivos e preservar a integridade dos ativos ao longo de seu ciclo de vida.

Segundo (PINTO ALAN KARDEC; XAVIER, 2013), a manutenção moderna deve ser considerada como uma função gerencial e estratégica, que influencia diretamente nos resultados do negócio. Isso inclui não apenas o reparo de falhas, mas o planejamento proativo, a análise de riscos, o monitoramento por indicadores de qualidade e desempenho(KPIs), além da integração com sistemas corporativos como o ERP (*Enterprise Resource Planning*) - Planejamento dos Recursos Empresariais.

Além disso, autores como (MOBLEY, 2008) apontam que o uso de estratégias combinadas — preventiva, preditiva e corretiva — deve ser orientado por análises de custo-benefício, criticidade dos ativos e dados operacionais, promovendo uma gestão orientada à confiabilidade.

Na Figura 1, estão representadas as principais funções da gestão da manutenção e as suas relações.

2.1.3.1 Funções-Chave do Gerenciamento da Manutenção

O planejamento e programação da manutenção envolve definir o que precisa ser feito (escopo da ordem de serviço), como será feito (procedimentos), quem fará (recursos humanos), quais materiais e ferramentas são necessários, e quando será feito (programação). Para (PINTO ALAN KARDEC; XAVIER, 2013), o planejamento é o pilar central da confiabilidade, pois

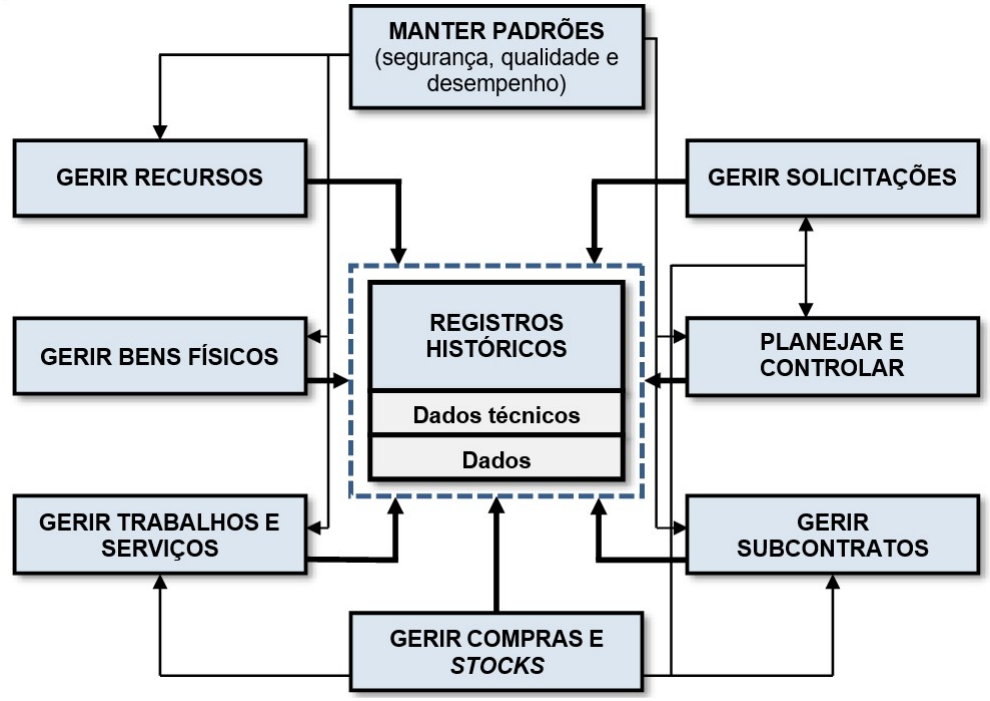

Figura 1 – Principais Funções da Gestão da Manutenção e suas Relações

Fonte: Adaptado de Pinto, 2013

antecipa falhas e organiza os meios para mitigá-las com eficiência. Principais responsabilidades: Criação e detalhamento de Ordens de Serviço (O.S.); Priorização de O.S. com base na criticidade e urgência; Estimativa de tempo e recursos; Alocação de mão de obra, materiais e ferramentas; programação de curto, médio e longo prazo; coordenação com a produção para janelas de manutenção; gerenciamento do *backlog* (serviços pendentes).

Na Figura 2, está representado o fluxograma macro dos processos dentro da célula PCM e as suas relações.

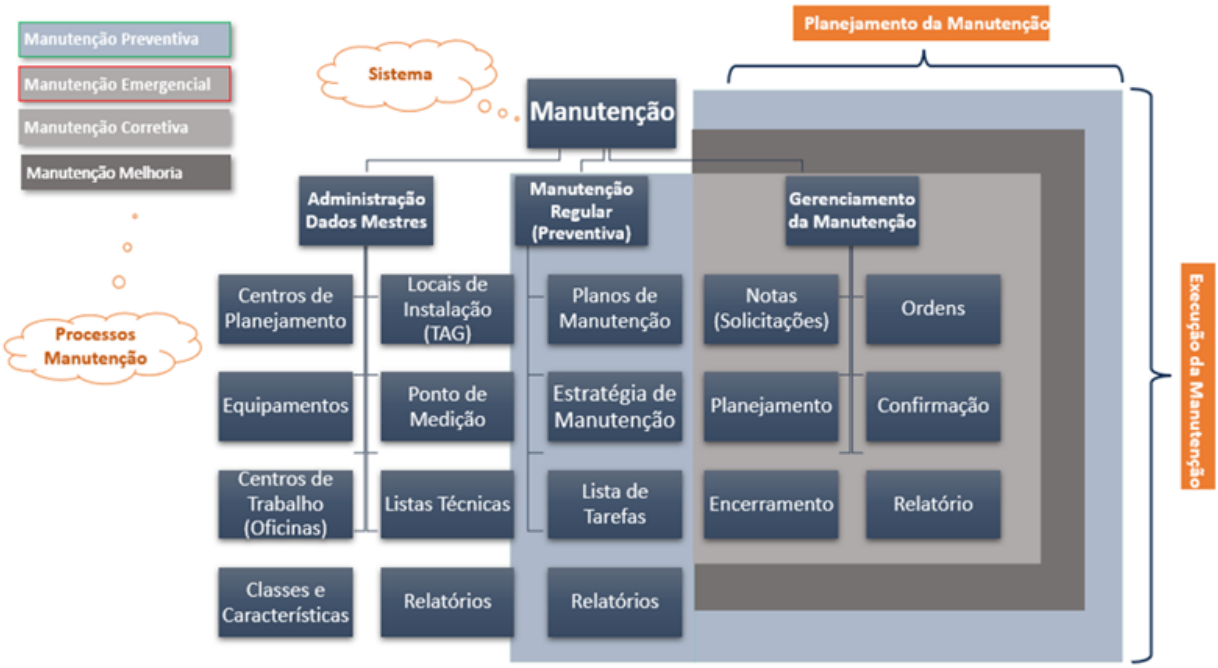

Figura 2 – Fluxograma macro dos processos dentro da célula PCM e as suas relações

Fonte: Adaptado de ABRAMAN, 2024

A gestão de ativos e estratégias de manutenção permeia o entendimento profundo do parque de ativos da empresa, sua criticidade, modos de falha e a definição das estratégias de manutenção mais adequadas para cada um (preventiva, preditiva, prescritiva, corretiva planejada, etc.). Assegura que os esforços de manutenção sejam direcionados para onde são mais necessários e eficazes, otimizando a confiabilidade e o custo total de propriedade. A gestão de ativos é definida pela norma (STANDARDIZATION, 2014) como a coordenação de atividades para obter valor real a partir dos ativos. Principais responsabilidades: cadastro técnico completo dos ativos; análise de criticidade dos ativos; definição e revisão de planos de manutenção; aplicação de metodologias para otimizar planos de manutenção; gestão do ciclo de vida dos ativos (desde a especificação até o descomissionamento).

A gestão de recursos realiza a administração eficiente dos insumos necessários para a execução da manutenção. Garante que a manutenção tenha os recursos necessários disponíveis quando preciso, ao menor custo possível, evitando atrasos e desperdícios. Principais responsabilidades:

- Materiais e Sobressalentes (Almoxarifado da Manutenção): Controle de estoque, definição de níveis mínimos e máximos, gestão de peças críticas, processos de compra e recebimento.
- Ferramentas e Equipamentos: Disponibilização, calibração, manutenção das próprias ferramentas.

- Orçamento e Custos: Elaboração e acompanhamento do orçamento de manutenção, controle de custos por ativo, por O.S. e análise de desvios.

A gestão da confiabilidade e melhoria contínua busca analisar o desempenho dos ativos e dos processos de manutenção para identificar oportunidades de melhoria, aumentar a confiabilidade e reduzir falhas recorrentes. Principais responsabilidades: coleta e análise de dados de falhas (tempo para falha, tempo para reparo); cálculo e monitoramento de Indicadores Chave de Desempenho (KPIs) como MTBF, MTTR, Disponibilidade, *Backlog*, Custo de Manutenção, Cumprimento da Programação; análise de Causa Raiz (RCA) para falhas crônicas ou significativas; implementação de tecnologias de Manutenção Preditiva e Baseada na Condição; proposição e implementação de projetos de melhoria.

A gestão de ordens de serviço (O.S.) fornece a rastreabilidade, o controle e os dados necessários para todas as outras funções de gerenciamento, especialmente para análise de custos e confiabilidade. Abrange desde a solicitação e criação da O.S., passando pela aprovação, planejamento, execução, até o seu encerramento e registro histórico. Abrangência: padronização do processo de O.S.; registro detalhado de todas as informações (problema, causa, solução, tempo, materiais, custos); acompanhamento do status de cada O.S.; geração de histórico de intervenções nos ativos.

A gestão da segurança, saúde e meio ambiente (SSMA) na Manutenção é responsável por assegurar que todas as atividades de manutenção sejam realizadas de forma segura, protegendo os colaboradores, o meio ambiente e cumprindo as legislações.

Já os sistemas de gerenciamento da manutenção, embora sejam uma ferramenta, a gestão do sistema é uma garantia de sua correta utilização, pois é a plataforma que suporta a maioria das outras funções, automatiza processos, centraliza informações, facilita a análise de dados e melhora a tomada de decisão em todas as esferas da manutenção. Possui a função de garantir a qualidade e integridade dos dados inseridos, configuração e customização do sistema, treinamento dos usuários, geração de relatórios e *dashboards* e a integração com outros sistemas.

A gestão de contratos de serviços é uma função estratégica que abrange desde a seleção criteriosa de fornecedores até o encerramento contratual, assegurando que os serviços prestados atendam aos padrões de qualidade, prazos e custos estabelecidos pela organização, gerenciando os riscos associados. Essa gestão envolve: definição do escopo dos serviços a serem terceirizados; seleção e qualificação de fornecedores; elaboração e gestão de contratos e Acordos de Nível de Serviço (SLAs); monitoramento do desempenho dos terceiros.

2.2 Sistema ERP como Plataforma para a Gestão da Manutenção: O Papel das Notas e Ordens de Manutenção

Historicamente, a gestão da manutenção em muitas organizações operava de forma relativamente isolada, com sistemas próprios ou processos manuais que dificultavam a comuni-

cação e a integração com outras áreas vitais da empresa. Originalmente concebidos a partir da evolução dos sistemas de Planejamento de Requisitos de Materiais (MRP) e Planejamento de Recursos de Manufatura (MRP II) nas décadas de 1970 e 1980, os ERPs transcenderam o foco inicial na manufatura para se tornarem plataformas integradas que visam gerenciar e coordenar a totalidade dos recursos, informações e funções de uma empresa (MONK; WAGNER, 2013). A incorporação da funcionalidade de manutenção dentro de um sistema ERP representa uma mudança de paradigma, posicionando a manutenção como uma parte intrínseca e interconectada do ecossistema empresarial.

Um sistema ERP é, em sua essência, um conjunto de softwares aplicativos modulares que operam sobre uma base de dados centralizada e compartilhada, permitindo que diferentes departamentos – como finanças, contabilidade, recursos humanos, vendas, compras, logística, produção e manutenção – acessem e processem informações de forma coesa e em tempo real. Promovem um fluxo de dados contínuo e consistente através da organização. Essa capacidade de integração não apenas otimiza processos individuais, mas também fornece uma visão holística do desempenho empresarial, fundamental para o planejamento estratégico e o controle gerencial.

2.2.1 Nota e Ordem de Manutenção

A maioria dos processos de manutenção são bem definidos pelos procedimentos internos e seguem um ciclo que se inicia com a identificação de uma necessidade. A Nota de Manutenção é a formalização dessa identificação, transformando uma observação informal em um dado estruturado que pode ser processado, analisado e rastreado, servindo como gatilho para o processo de planejamento e controle da manutenção. Ela é o precursor da Ordem de Manutenção, que autoriza a execução do trabalho e a alocação de recursos (SAP AG, 2025). Sem um registro inicial claro (a nota), o controle sobre o que precisa ser feito, por quê, onde e com qual urgência se torna caótico e ineficiente. O histórico de Notas de Manutenção, especialmente aquelas relacionadas a falhas (M2 - *Malfunction Report no SAP*), é uma fonte rica de dados para análises de confiabilidade. O estudo desses dados permite identificar os modos de falha (como os equipamentos estão falhando), taxas de falha (com que frequência falham, sendo base para cálculo de MTBF(*Mean Time Between Failures*) - tempo médio entre falhas e causas raíz (por que estão falhando).

Essas informações são cruciais para otimizar estratégias de manutenção preventiva, preditiva, ou para justificar modificações de projeto (Moubray, 1997 - RCM). A Nota de Manutenção é, portanto, um componente essencial de um sistema FRACAS (*Failure Reporting, Analysis, and Corrective Action System*) - Sistema de Relato, Análise e Ação Corretiva de Falhas. A qualidade da informação registrada na nota impacta diretamente a eficácia do planejamento, programação e execução das atividades de manutenção subsequentes.

No contexto da gestão de ativos industriais, a ordem de manutenção (OM) ou ordem de serviço (O.S.) representa um elemento central na estrutura de controle e execução das atividades

de manutenção(SAP AG, 2025). É um documento que formaliza a necessidade de um trabalho de manutenção em um equipamento ou planta e serve como um guia para os técnicos e outros envolvidos, detalhando o trabalho a ser feito, os recursos necessários e os prazos. No SAP, a ordem de manutenção é um elemento central do processo de planejamento, execução e controle das atividades de manutenção.

Essas ferramentas servem não apenas como um registro do trabalho a ser feito, mas como um mecanismo de controle e rastreabilidade das ações realizadas, dos custos incorridos e dos resultados obtidos. No SAP em seu módulo PM, ele é o ponto de convergência entre os módulos de manutenção, logística e finanças, garantindo um fluxo de dados coeso e em tempo real.

A Nota de Manutenção serve como o canal formal para a identificação e registro das necessidades de manutenção, enquanto a Ordem de Manutenção se configura como a espinha dorsal para o planejamento, execução, controle e análise das intervenções. A integração dessas funcionalidades com os demais processos empresariais dentro do ERP não apenas otimiza a função manutenção em si, mas também a coloca em um patamar estratégico, fornecendo informações vitais para a gestão de ativos e contribuindo diretamente para a eficiência e rentabilidade global da organização.

A eficácia de um sistema de gerenciamento de manutenção, como o SAP PM, reside não apenas em sua capacidade de processar transações, mas também em sua flexibilidade para categorizar e diferenciar os diversos tipos de demandas e trabalhos de manutenção. Essa categorização, realizada através de diferentes tipos de Notas e Ordens de Manutenção, permite um tratamento mais adequado a cada situação, alinhando a execução com as estratégias de manutenção da organização e facilitando análises gerenciais posteriores. PINTO Alan Kardec; XAVIER (2013) destacam que a tipificação dos serviços é condição para integrar a manutenção ao planejamento estratégico e à gestão de custos.

2.2.1.1 Tipos Conceituais de Notas de Manutenção (A Identificação da Necessidade)

A Nota de Manutenção é o registro inicial, a formalização de uma observação ou solicitação. Essas demandas podem ser categorizadas com base em sua necessidade ou urgência e sua tipologia deriva da natureza do evento. Abaixo, tipos mais utilizados de Notas de Manutenção.

- M1 - Solicitação de Atividade (*Activity Report / General Notification*): Utilizada para registrar solicitações de serviços de manutenção que não são necessariamente falhas emergenciais. Pode incluir pedidos de pequenas melhorias, modificações, instalações, ou serviços gerais de manutenção que podem ser planejados. Pode ser usada para registrar trabalho já concluído em ordens de planos criadas automaticamente pelo sistema.
- M2 - Relatório de Avaria/Falha (*Malfunction Report*): Destinada a registrar a ocorrência de uma falha, defeito, quebra ou qualquer mau funcionamento de um ativo (equipamento

ou local de instalação) que impacte ou possa impactar sua funcionalidade. É o principal gatilho para a manutenção corretiva.

- M3 - Nota de Manutenção de Emergência (*Emergency Maintenance Notification*): utilizada para identificar imediatamente demandas que requerem atenção e ação corretiva imediatas devido a riscos iminentes à segurança, ao meio ambiente, à produção crítica ou a danos patrimoniais significativos.

2.2.1.2 Tipos Conceituais de Ordens de Manutenção (O Planejamento e Execução do Trabalho)

Uma vez que a demanda é reconhecida e considerada válida, ela frequentemente se traduz em uma Ordem de Manutenção, que detalha o trabalho a ser realizado. Embora os códigos e descrições exatas possam variar entre sistemas e implementações customizadas, os conceitos subjacentes refletem as diversas naturezas e urgências das intervenções de manutenção. A tipologia das ordens reflete a natureza do trabalho planejado. Abaixo, alguns tipos mais utilizados:

- PM01 - Ordem de Manutenção Corretiva (Planejada/Programada): Este tipo de ordem é utilizado para planejar, executar e controlar trabalhos de reparo decorrentes de falhas ou mau funcionamento de ativos que não configuram uma emergência imediata. Embora a necessidade seja corretiva (ou seja, algo falhou), existe a possibilidade de realizar um planejamento mais estruturado das operações, alocação de materiais e mão de obra antes da execução. É tipicamente originada de uma Nota de Manutenção M2 (Relatório de Avaria).
- PM02 - Ordem de Manutenção Preventiva: Utilizada para executar tarefas de manutenção baseadas em calendário (tempo), contador de utilização do equipamento (desempenho) ou em eventos específicos definidos em um plano de manutenção preventiva ou preditiva. Essas ordens são frequentemente geradas automaticamente pelo sistema com base na programação estabelecida.
- PM03 - Ordem de Manutenção de Emergência: Este tipo de ordem é especificamente designado para falhas críticas e imprevistas que requerem ação corretiva imediata para evitar consequências graves, como riscos à segurança de pessoas, danos ambientais significativos, paradas de produção catastróficas ou perdas patrimoniais substanciais. A OM PM03 aciona um processo acelerado, muitas vezes com planejamento simplificado ou realizado “em tempo de execução”, e tem a mais alta prioridade.

2.2.1.3 Visão Geral do Ciclo de Vida Integrado da Gestão da Manutenção no Ambiente ERP SAP PM

Um fluxo maduro de manutenção parte do registro tempestivo de falhas ou solicitações, avança pela triagem técnica e priorização, passa pelo planejamento detalhado, segue para a

execução controlada, conclui-se pela liquidação de custos e se encerra com o *business completion*, quando *feedback* estruturado retroalimenta indicadores de confiabilidade e custo.

- Detecção e Registro – Criação da Notificação: O ciclo se inicia com a detecção de uma necessidade de manutenção. Isso pode ser uma falha em um equipamento, uma condição anormal observada durante uma inspeção, uma solicitação de melhoria por um operador, ou a necessidade de um serviço programado. Essa detecção é formalizada no SAP PM através da criação de uma Nota de Manutenção (também referida como "notificação").
- Triagem – Avaliação Técnica, Priorização e Decisão: Uma vez que a nota de manutenção é criada, ela passa por um processo de triagem pelo departamento de inspeção. Analisa-se a validade e a natureza do problema ou solicitação. Verifica-se se a informação está completa, se a urgência da nota com base em critérios como impacto na produção, segurança, custos, e criticidade do ativo. A prioridade ajudará a determinar a sequência de atendimento.
- Planejamento – Detalhamento de Operações, Materiais, Capacidades e Custos: etapa onde é criada a Ordem de Manutenção (OM), inicia-se a fase de planejamento detalhado. É nas operações da OM onde as tarefas específicas a serem realizadas, a sequência, a duração estimada e o centro de trabalho responsável são definidos. Dentro da OM também é realizada a reserva das peças de reposição e materiais de consumo necessários, com verificação de disponibilidade. Se não houver estoque, pode-se gerar requisições de compra.
- Execução – Liberação e Consumo de Peças: Após o planejamento e a devida aprovação, a OM é liberada, o que permite que os materiais sejam baixados do estoque, os serviços externos sejam requisitados, os técnicos realizem as tarefas conforme planejado.
- Encerramento – Apontamentos de Tempo, Conclusão Técnica e Comercial e *Feedback*: Após a execução completa do trabalho conforme planejado, é realizado, também, o registro dos relatos técnicos com as informações sobre o que foi feito, causas de falha encontradas, medições, etc., que podem ser atualizadas na OM ou na nota técnica. Após a realização de todos os apontamentos de horas trabalhadas e a liquidação dos custos, a OM passa pelo processo de encerramento. Após o encerramento técnico e comercial, a ordem é bloqueada para quaisquer outros lançamentos contábeis ou de materiais. Ela é considerada completamente encerrada e arquivada para fins históricos.

2.2.2 Qualidade dos dados em ordens de manutenção

A Nota e a Ordem de Manutenção constituem o principal repositório primário de informação, capturando a "realidade operacional" no momento anterior ou posterior à intervenção, porém a mera existência de dados não garante valor. A ausência de qualidade semântica transforma o que deveria ser um ativo informacional valioso em um labirinto de ambiguidades, inconsistências

e ruídos, dificultando a extração de *insights* acionáveis. Como salientado por Jardine, Lin e Banjevic (2006) sobre diagnóstico e prognóstico de máquinas, a disponibilidade de dados de alta qualidade é um pré-requisito indispensável para qualquer análise eficaz e decisão informada no âmbito da manutenção.

A qualidade semântica refere-se à capacidade dos dados de serem corretamente compreendidos e interpretados por seus usuários (humanos ou sistemas) no contexto pretendido, de forma clara, inequívoca e consistente com o significado que o produtor dos dados intencionou transmitir (BATINI C.; SCANNAPIECO, 2016). Não se trata apenas de os dados estarem corretos (acurácia) ou completos, mas de seu significado ser acessível e não ambíguo.

Portanto, enquanto todas as dimensões da qualidade de dados são importantes para a OM, a dimensão semântica é particularmente crítica para transformar os dados brutos de manutenção em inteligência acionável. Ela é a ponte entre o registro do evento e a sua correta compreensão na utilização para a melhoria contínua dos processos de manutenção e na confiabilidade dos ativos.

2.3 Manutenção Assistida por Inteligência Artificial(IA)

Nas últimas décadas, observou-se uma evolução gradual das abordagens de manutenção, que partiram da intervenção corretiva reativa, evoluíram para práticas preventivas baseadas em intervalos e avançaram para a manutenção preditiva baseada em condição (JARDINE; LIN; BANJEVIC, 2006). Contudo, os atuais ambientes de produção, cada vez mais digitalizados e orientados por dados, demandam soluções ainda mais proativas, adaptativas e integradas. Esse movimento culmina na chamada Manutenção Assistida por Inteligência Artificial (MAIA), ilustrado na figura 3.

Figura 3 – Integração de Técnicas Preditivas, Análise Prescritiva e Inteligência Artificial

Fonte: Adaptado de Abraman, 2023

No paradigma da Manutenção Assistida por Inteligência Artificial, algoritmos de IA extrapolam a função tradicional de detecção de falhas, atuando como um agente cognitivo que integra dados heterogêneos – históricos de manutenção, variáveis de processo, registros de sensores *IoT*, *check-lists* operacionais, relatórios de inspeção – e converte esse conhecimento

em *insights* prescritivos. Nessa categoria, a IA passa a ser um "braço direito" do técnico de manutenção, não apenas prevendo falhas, mas também sugerindo ações corretivas e assegurando a eficiência dos ativos, figura 4.

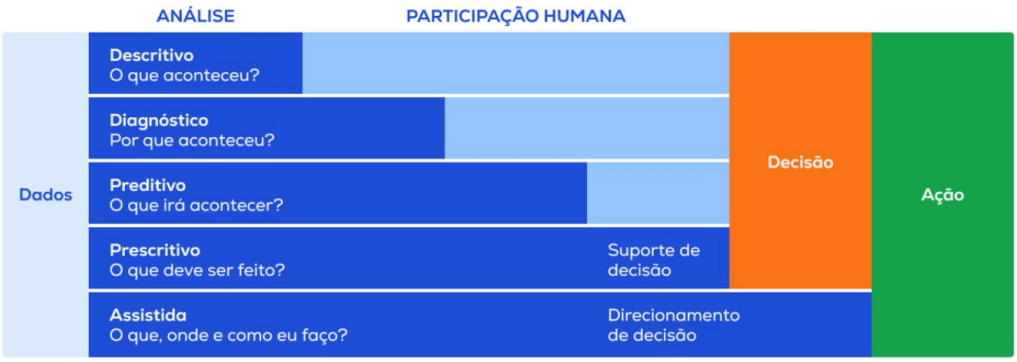

Figura 4 – Inteligencia Assistida por IA: Dados → Decisão → Ação

Fonte: Adaptado de Abraman, 2023

Em contraste com a manutenção preditiva clássica, onde a ênfase recai na estimativa do momento ideal para intervenção, a MAIA propõe sugestões contextualizadas de causa-raiz, priorização de recursos e sequenciamento de ordens de serviço, promovendo ganhos de produtividade, redução de custos e mitigação de riscos operacionais (ABRAMAN, 2023b). Figura 5.

Figura 5 – Workflow da Manutenção Assistido por IA

Fonte: Adaptado de Abraman, 2023

2.4 Inteligência artificial: *Large Language Model* (LLM) e Arquitetura *Transformer*

Os *Large Language Model* (LLM), ou Grandes Modelos de Linguagem, são um tipo de modelo de aprendizado de máquina, predominantemente baseado em redes neurais profundas,

que é projetado para tarefas de processamento de linguagem natural (PLN),capazes de gerar e compreender linguagem natural em diversos domínios. LLMs são caracterizados por possuírem um número massivo de parâmetros (frequentemente na casa dos bilhões ou trilhões) e são treinados em quantidades volumosas de dados textuais utilizando técnicas de aprendizado auto-supervisionado. Esses modelos tornaram-se viáveis e eficazes graças à introdução da arquitetura *Transformer* por Vaswani et al. (2017). No artigo "*Attention Is All You Need*", os autores propuseram o *Transformer* como uma nova arquitetura neural baseada unicamente em mecanismos de atenção, sem utilizar arquiteturas recorrentes (RNNs) ou convolucionais (CNNs), onde cada *token* deve ser processado um após o outro. Essa inovação permitiu maior paralelismo no processamento de sequências e capturou dependências de longo alcance de forma mais eficiente.

A arquitetura *Transformer* é estruturada em blocos de *Multi-Head Self-Attention* ou autoatenção multi-cabeças e camadas de Redes Neurais *feed-forward* posicionais. Cada palavra (ou *token*) de entrada é convertida em um vetor denso (*embedding*) e processada pelas camadas de atenção, onde o modelo aprende a "prestar atenção" aos elementos relevantes do contexto em cada posição Vaswani et al. (2017). O mecanismo de atenção calcula os pesos que indicam a importância de outras palavras do contexto para entender ou prever a palavra atual. Com múltiplas cabeças de atenção, o modelo consegue capturar simultaneamente diferentes tipos de relacionamentos linguísticos (por exemplo, relação sujeito-verbo, referência anáfora, dependências semânticas). Em seguida, redes *feed-forward* posicionais transformam os vetores atentivos e a arquitetura repete esse processo em várias camadas empilhadas (camadas *Transformer*). Adicionalmente, usa-se codificação de posição para indicar a ordem dos *tokens*, já que a atenção em si é independente de posição.

Os LLMs modernos geralmente seguem dois tipos de configurações arquiteturais baseadas em *Transformers*: modelos do tipo codificador-decodificador, arquitetura original proposta no artigo "*Attention Is All You Need*" (por exemplo, T5, BART), úteis para sequência para sequência (seq2seq) como tradução e resumo, e modelos somente decodificador (por exemplo, GPT-3, GPT-4) focados em geração autoregressiva de texto. Em ambos os casos, o treinamento prévio (pré-treinamento) é crucial: o modelo é exposto a um grande corpus e aprende a prever palavras mascaradas ou próximas (no caso de codificadores) ou a próxima palavra (no caso autoregressivo). Esse pré-treinamento constrói um conhecimento linguístico e factual vasto nos pesos do modelo. Posteriormente, é comum um ajuste fino (*fine-tuning*) supervisionado ou por reforço (como RLHF, *Reinforcement Learning from Human Feedback*) para especializar o LLM em tarefas específicas ou alinhá-lo a preferências (por exemplo, ser útil e seguir diretrizes em diálogos).

Graças à escala e à arquitetura, os LLMs atuais apresentam capacidades de compreensão robusta de contexto, geração de texto fluente e até resolução de problemas complexos via linguagem. Eles armazenam conhecimento factual absorvido do treinamento e podem usá-lo para

responder perguntas ou realizar inferências. No entanto, também enfrentam desafios: sua base de conhecimento é estática após o treinamento (dificuldade de atualização), podem "alucinar" fatos inexistentes e não fornecem garantias de veracidade ou justificativa de suas respostas (LEWIS et al., 2020). Ainda assim, a combinação de arquitetura *Transformer* e treinamento em larga escala consolidou os LLMs como o estado da arte em IA generativa, que revolucionou o processamento de linguagem natural. .

2.4.1 Avanços arquiteturais em Grandes Modelos de Linguagem (LLMs)

Desde a introdução do *Transformer* por Vaswani et al. (2017), os LLMs passaram por uma evolução arquitetural significativa. Além dos aprimoramentos incrementais em eficiência computacional, aumento de contexto e precisão, destaca-se a capacidade multimodal que, além de permitir a geração e compreensão de texto, proporciona também a compreensão e geração de áudio, imagens e vídeo. A evolução multimodal combina várias modalidades de entrada (texto, voz, imagem, vídeo) em um fluxo unificado de representação vetorial, habilitando a geração de saídas que também cruzam essas fronteiras. São capazes de integrar modalidades distintas em uma única arquitetura coesa, abrindo caminhos para aplicações em assistentes cognitivos, interfaces vocais, geração e análise audiovisual e etc. Além disso, os LLMs têm se consolidado como ferramentas robustas para a geração e revisão de código, beneficiando-se de corpus extensivos de repositórios de software e avanços em *embeddings* estruturados para linguagens de programação.

Este panorama de inovações arquiteturais demonstra como os LLMs evoluíram de estruturas densas generalistas para arquiteturas modulares, escaláveis e especializadas, estabelecendo uma nova geração de modelos capazes de operar com precisão e rapidez em contextos cada vez mais desafiadores.

2.5 Mecanismos de Tratamento da Semântica por LLMs: *Embeddings*, Atenção Contextual e Inferência Latente

Dada a importância da semântica na compreensão de linguagem, os LLMs incorporam diversos mecanismos para representar e manipular significados:

- *Embeddings* semânticos: Os LLMs usam representações vetoriais densas *Embeddings* para palavras, frases ou documentos, de modo que conceitos semanticamente relacionados fiquem próximos no espaço vetorial. Nos *Transformers*, as camadas internas refinam constantemente os *Embeddings* das palavras conforme o contexto em que aparecem – produzindo *Embeddings* contextuais. Assim, uma palavra polissêmica (como "banco") terá *Embeddings* diferentes dependendo do sentido na frase, pois a rede ajusta sua representação de acordo com as palavras vizinhas. Esses *Embeddings* servem como uma espécie de código semântico latente do conteúdo textual. Por exemplo, é possível pegar o *Embeddings*

final de uma sentença inteira e utilizá-lo para medir similaridade com outras sentenças (*semantic textual similarity*). Modelos baseados em *Transformers* ajustados, demonstram excelente desempenho em tarefas de busca semântica, recuperando textos similares em significado mesmo sem termos exatos em comum (NAQVI et al., 2022).

- Atenção contextual: O mecanismo de autoatenção dos *Transformers* é peça-chave para tratamento da semântica. Ele permite que o modelo relacione diferentes partes do texto, identificando quais palavras fornecem contexto semântico relevante umas para as outras. Por exemplo, na frase "O técnico consertou a máquina porque ela estava apresentando falha", a atenção adequada aprende que "ela" refere-se a "a máquina" e que "apresentando falha" é a causa do conserto – conectando semanticamente pronome, substantivo e verbo. As múltiplas cabeças de atenção frequentemente aprendem a focar em diferentes aspectos: algumas capturam relações sintáticas (p. ex., sujeito-verbo), outras capturam relações semânticas (p. ex., adjetivo-característica do substantivo) ou referências anafóricas. Essa atenção dinâmica e contextual faz com que o significado de cada palavra seja interpretado com base na sentença inteira, em vez de fixo isoladamente. Como resultado, o LLM garante consistência semântica na geração e dá ao modelo a capacidade de realizar inferências contextuais – ele pode inferir relações não explícitas no texto a partir da combinação de pistas distribuídas em diferentes partes do contexto.

- Memorização semântica e conhecimento latente: Durante o pré-treinamento, LLMs "leem" milhões de documentos, absorvendo inúmeras informações factuais e relações conceituais. Essas informações ficam armazenadas nos parâmetros da rede de forma distribuída. Assim, o LLM adquire uma vasta base de conhecimento implícita, que se assemelha a uma memória semântica de um agente cognitivo (LI; LI, 2024). Por exemplo, o modelo pode saber que "um polímero" é um tipo de "material", ou que "trocar óleo" é uma tarefa comum em manutenção de veículos. Essa memória latente permite ao modelo responder perguntas ou completar textos usando fatos aprendidos. No entanto, diferentemente de uma base de conhecimento estruturada, a recuperação dessa memória paramétrica não é diretamente controlável – o modelo recupera espontaneamente o que julga relevante via seus pesos e atenção, o que pode falhar às vezes (levando a erros factuais ou "alucinações"). Pesquisas recentes tentam complementar LLMs com componentes de memória externa ou mecanismos explícitos de recuperação para imitar melhor uma memória semântica organizada e persistente (LEWIS et al., 2020).

- Inferência latente e raciocínio: Mesmo sem algoritmos simbólicos explícitos, LLMs demonstram a capacidade de realizar inferências a partir do que foi aprendido. Ou seja, conseguem generalizar e deduzir novas informações combinando conhecimentos. Isso fica evidente quando um modelo responde corretamente a uma pergunta cuja resposta não estava literalmente em nenhum texto, mas pode ser inferida pela combinação de fatos (por exemplo, inferir consequências lógicas simples, resolver analogias ou cadeias

causais curtas). Essa inferência ocorre nos estados latentes da rede durante a geração da resposta. Em alguns casos, pesquisadores observaram que aumentar o tamanho do modelo gera habilidades emergentes de raciocínio não vistas em modelos menores (LI; LI, 2024). Entretanto, o raciocínio complexo (multi-etapas, lógico ou matemático) ainda é um ponto fraco – levando ao desenvolvimento de técnicas como encadear pensamentos (*chain-of-thought prompting*) ou o uso de módulos externos para cálculos. De todo modo, para muitos aspectos semânticos cotidianos – como entender relações de causa e efeito em uma descrição, ou inferir o sentimento de um texto – os LLMs já conseguem desempenhar inferência latente com sucesso, graças à combinação de uma representação semântica rica e padrões associativos aprendidos.

Em síntese, os LLMs tratam a semântica através de representações vetoriais profundas, atenção ao contexto e conhecimento distribuído nos pesos. Esses mecanismos permitem que compreendam e produzam texto de forma semanticamente sensata na maioria das vezes. Todavia, quando se requer precisão factual absoluta ou conhecimento muito específico fora da base treinada, podem ocorrer falhas – motivando extensões como a recuperação de conhecimento externo.

2.6 Engenharia de *Prompt* em grandes modelos de linguagem

A engenharia de *prompt* consolidou-se como um campo técnico essencial, cujo objetivo central é maximizar a performance dos LLMs mediante a formulação estratégica de instruções textuais — os *prompts*. Em termos técnicos, a engenharia de *prompt* consiste no projeto, seleção e refinamento sistemático das entradas fornecidas ao modelo, de modo a orientar sua resposta para os resultados desejados, sem a necessidade de ajustar os parâmetros internos do modelo, permitindo extrair capacidades latentes dos LLMs para uma vasta gama de tarefas.

A engenharia de *prompt* viabiliza o aproveitamento pleno dos conhecimentos e habilidades já incorporados nos LLMs, tornando possível adaptar modelos genéricos a domínios ou demandas específicas com baixo custo operacional e sem re-treinamento.Dentre seus objetivos estratégicos, destacam-se: maximizar o desempenho em tarefas específicas, aprimorando acurácia, relevância e precisão das respostas; direcionar o formato, o estilo e a coerência das saídas; mitigar vieses e inconsistências; reduzir ocorrências de respostas incorretas ou alucinações; evitar custos de re-treinamento e ajustes complexos, ampliando a aplicabilidade dos modelos pré-treinados(SAHOO et al., 2025).

2.6.1 *Chain-of-Thought Prompting*

A técnica *Chain-of-Thought Prompting* (CoT) ou Instrução de Cadeia de Pensamento representa um dos avanços mais significativos na engenharia de *prompt*, notadamente em tarefas que exigem raciocínio lógico, dedutivo ou multi-etapas. O conceito de *Chain-of-Thought*

Prompting foi sistematizado por (WEI et al., 2022), ao demonstrar que grandes modelos de linguagem podem apresentar melhorias substanciais em desempenho, quando induzidos a realizar explanações intermediárias de seu raciocínio, em vez de simplesmente fornecer respostas finais. Este paradigma explora a capacidade latente dos LLMs de decompor problemas complexos em etapas menores e sequenciais, aproximando-se do que se espera de um raciocínio humano estruturado.

O *Chain-of-Thought Prompting* consiste em instruir o modelo, via *prompt*, a explicitar o processo de pensamento subjacente à resolução de uma tarefa. O sucesso do *CoT Prompting* está fundamentado na exploração de conhecimento contextual profundo e inferência distribuída, permitindo ao modelo conectar diferentes fragmentos de conhecimento prévio para construir uma resposta coerente e fundamentada. Existem basicamente três abordagens principais: (i) *Few-shot CoT*, na qual exemplos completos com raciocínio passo a passo são fornecidos como demonstração no *prompt*; (ii) *Zero-shot CoT*, que utiliza instruções explícitas, como "vamos pensar passo a passo", induzindo o modelo a gerar autonomamente a cadeia de raciocínio; (iii) *Auto-CoT*, onde exemplos são automaticamente gerados pelo próprio modelo para compor o *prompt* (ZHANG et al., 2022).

Ao correlacionar o potencial do *Chain-of-Thought Prompting* ao contexto do trabalho aqui apresentado, observa-se que a capacidade dos LLMs de estruturar raciocínios em múltiplos passos se torna especialmente relevante para aplicações de validação semântica em sistemas complexos, como a gestão de ordens de manutenção. A utilização de CoT permite que o modelo não apenas avalie a conformidade sintática dos registros, mas também desenvolva justificativas detalhadas e contextualizadas para cada critério de validação, promovendo maior transparência e confiabilidade ao processo. Dessa forma, a incorporação do *Chain-of-Thought Prompting* amplia a robustez do *framework* de validação semântica, possibilitando auditorias mais precisas, explicações interpretáveis e decisões fundamentadas, alinhadas às melhores práticas de governança e qualidade de dados em manutenção industrial.

2.6.2 *Retrieval-Augmented Generation (RAG)*: Conceito, Importância para Dados Específicos e Complementariedade aos LLMs

Apesar do poder dos LLMs, sua dependência de conhecimento armazenado nos pesos apresenta limitações: eles não podem facilmente atualizar informações pós-treinamento, podem não conhecer dados específicos de um domínio (por exemplo, detalhes de procedimentos de uma empresa) e não fornecem referências para justificar suas respostas. Para contornar esses problemas, introduziu-se a técnica de Geração Aumentada por Recuperação (*Retrieval-Augmented Generation – RAG*). Em essência, um sistema RAG combina um modelo generativo paramétrico (o LLM) com uma fonte externa de conhecimento não-paramétrica (por exemplo, um banco de dados de documentos) (LEWIS et al., 2020). Durante a geração de uma resposta, o sistema realiza primeiro uma recuperação de informações relevantes à consulta do usuário e alimenta

esses trechos ao LLM, para que este elabore a saída baseando-se nessas evidências recuperadas.

Formalmente, Lewis et al. (2020) propuseram o RAG como um modelo híbrido no qual o LLM age como componente gerador, enquanto um módulo de busca neural consulta um índice de vetores. Ao combinar memória paramétrica com memória não-paramétrica, o RAG busca aproveitar o melhor dos dois mundos. As etapas simplificadas são: (1) dado um *input* (p. ex., uma pergunta em linguagem natural), extrair as *embeddings* da consulta; (2) comparar com um índice de *embeddings* de documentos ou textos específicos (usando métrica de similaridade) para recuperar os documentos mais relevantes; (3) concatenar ou fornecer esses documentos ao LLM; (4) o LLM gera a resposta usando tanto seu conhecimento interno quanto as evidências fornecidas. Esse arranjo provou-se efetivo para tarefas intensivas em conhecimento, superando modelos puramente paramétricos em vários cenários. Notavelmente, o RAG apresentou linguagem mais específica, diversificada e factualmente correta do que um modelo generativo isolado, nos experimentos originais (LEWIS et al., 2020).

Dessa forma, o modelo fundamenta sua saída em dados confiáveis e atualizados ao invés de depender apenas da memória estática. Isso também traz a vantagem de proveniência – sabe-se de onde veio a informação. O RAG atua, portanto, como complemento aos LLMs: ele supre a falta de memória de longo prazo atualizável e reduz alucinações, ao mesmo tempo em que permite que o LLM se concentre em interpretar e gerar linguagem de forma fluida e contextual.

Arquiteturas mais recentes estendem o paradigma RAG para cenários de raciocínio complexo multi-etapas. Por exemplo, o *framework CRP-RAG* (*Complex Reasoning and Planning* + RAG) introduzido por Xu et al. (2024) combina grafos de raciocínio com a geração, de forma a planejar cadeias lógicas e selecionar evidências iterativamente. O CRP-RAG demonstrou melhorias significativas em consultas complexas, obtendo respostas mais fiéis e consistentes, superando métodos RAG tradicionais em questões de múltiplos passos e verificação factual. Esse avanço ilustra que a integração de mecanismos de planejamento de conhecimento e verificação com *loop* de recuperação-geração pode tornar os sistemas ainda mais robustos.

3 DESENVOLVIMENTO

A elaboração e desenvolvimento deste trabalho se deu a partir da construção de um *framework* destinado à realizar o pré-processamento e carga dos dados de ordens e notas de manutenção, bem como a realização da validação semântica utilizando um LLM localmente e de técnicas de RAG e estratégias de Engenharia de *Prompt*. O *framework*, desenvolvido em Python, utiliza ferramentas específicas, incluindo o ambiente *Ollama* com um modelo considerado pequeno, o *gemma3* de quatro bilhões de parâmetros, que é possível executar em uma GPU comum, a biblioteca de abstração *LangChain* e bases de conhecimento organizadas, além de métodos sistemáticos para otimização e formulação de *prompts* com memória de contexto para guiar adequadamente a geração das respostas do modelo LLM. Para garantir que as validações sejam baseadas, não apenas no conhecimento geral do modelo, mas também em regras de negócio, foi disponibilizado para o modelo dados contextuais específicos.

3.1 Caracterização da área de estudo

O presente estudo pode ser caracterizado como uma pesquisa aplicada com desenvolvimento experimental, focada na criação e validação de um *framework* especialista para automação de processos de conformidade em ambientes industriais, que seja eficiente, rastreável, seguro e adaptável a regras de negócio customizadas. O desafio central enfrentado é a substituição do processo manual, tradicionalmente moroso e susceptível a falhas humanas, por um sistema automatizado que assegure alta eficiência, rastreabilidade robusta e total segurança na execução.

A pesquisa aplicada tem como objetivo gerar conhecimento com potencial de uso imediato, resolvendo problemas específicos com base em conhecimentos teóricos pré-existentes (LEME; OUTROS, 2018). Além disso, foi incorporado um componente experimental, na forma de implementação, testes e iterações com o *framework*, que se alinha às definições de pesquisa experimental, onde variáveis como *prompt* e parâmetros são manipuladas para validar suas influências no desempenho do sistema . De acordo com Gil (2007), a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto. Portanto, em termos metodológicos, classifica-se como uma pesquisa aplicada-experimental, de natureza interdisciplinar, mesclando atividades de desenvolvimento de software, testes de IA e avaliação de desempenho em cenários práticos.

3.2 Metodologia

Com a utilização de um estudo de caso, pretende-se reunir dados e informações relativas aos processos de gestão da manutenção industrial visando à implementação de uma ferramenta

de inteligência artificial que automatize a análise das ordens e notas registradas no sistema SAP PM. Ao articular a teoria e a prática, busca-se esclarecimentos e entendimento mais profundo e crítico sobre o tema e sua interação com a realidade em que se apresenta, dirimindo algumas dúvidas e contribuindo para o surgimento de outras indagações. A dimensão da pesquisa é composta por uma gama de processos de gestão da manutenção identificados no setor de PCM da mineradora estudada, que tem como referência o sistema ERP SAP de registro de atividades de manutenção.

A coleta inicial de dados foi realizada a partir de dados secundários relacionados aos processos de qualidade do setor de planejamento e controle da manutenção. Foram utilizadas normas, instruções normativas e procedimentos operacionais padrão. Além desses, usou-se também protocolos, bancos de dados, painéis e fluxogramas, bem como mapas, pareceres técnicos, relatórios de análises das ordens e notas de manutenção e demais documentos pertinentes aos processos. Foi realizado também o levantamento e análise de fontes científicas e técnicas com o propósito de fundamentar teoricamente e validar empiricamente a solução proposta.

Na pré-análise foi realizada a leitura de todo o material coletado. Posteriormente, a partir da exploração do material, foram identificados os prós e contras do sistema utilizado e as etapas utilizadas para alcançar as análises necessárias. No segundo passo, foram identificadas as oportunidades de melhoria com a utilização de ferramentas de IA. No terceiro passo, foi realizada a extração dos dados diretamente do Sistema de Gestão de Manutenção (SAP PM), selecionando registros provenientes das ordens e notas de manutenção. A amostra obtida foi composta por registros diversificados, abrangendo diferentes tipos de equipamentos e categorias de manutenção, oferecendo um panorama representativo do contexto operacional estudado. No quarto passo, foi realizada uma análise exploratória detalhada para identificar a estrutura dos dados sem alterar a integridade da base, realizando apenas a manipulação dos dados para se enquadrar às necessidades do *framework*. No quinto passo foi realizada a implementação e desenvolvimento de um *framework* para a realização da análise das Ordens e Notas de manutenção da rotina do setor de planejamento e controle da manutenção. No sexto passo, foi realizado o tratamento dos resultados obtidos e sua interpretação, bem como, foram estabelecidas as associações entre os resultados e o referencial teórico.

3.2.1 Critérios de avaliação dos campos das Ordens e Notas de manutenção

As ordens de manutenção apresentam desafios típicos de governança de dados industriais: terminologia variada, campos livres não padronizados, e ausência de justificativas formais. Devido aos fatos, foi necessária a elaboração e padronização de critérios para detectar campos omissos ou incoerentes, baseado-se em instruções normativas e procedimentos operacionais.

Inicialmente, foram levantados os principais campos das ordens e notas que precisam ser avaliados, e para cada campo, foram estabelecidos requisitos mínimos de preenchimento, coerência e relacionamento com demais atributos do registro, considerando aspectos como:

- Completude: presença obrigatória de informações essenciais.
- Consistência: alinhamento entre o valor do campo e as demais variáveis do registro.
- Aderência semântica: conformidade do conteúdo textual com a terminologia padronizada do domínio.
- Justificativa formal: quando requerido, presença de argumentação técnica que motive a decisão tomada.

Tais critérios foram estruturados em uma tabela, na qual cada coluna corresponde a uma classe de informação e cada linha contém o campo a ser avaliado e seus respectivos critérios de avaliação. Essa tabela foi salva em um arquivo no formato CSV, de modo a servir posteriormente como base estruturada para o *framework* de validação semântica.

Para que o *framework* de validação semântica pudesse operar com base nos critérios e regras de negócio específicos do ambiente SAP, foi fundamental a criação de mais duas bases de conhecimento auxiliar. Essas bases funcionam como uma "tabela verdade"que o *framework* recupera para validar os critérios. A aplicação foi projetada para consumir essas informações em um formato tabelar no formato de arquivos CSV, facilitando a extração, manutenção e atualização. A seguir, detalha-se a estrutura:

- Lista de Centros de Trabalho: No SAP AG (2025), existe uma relação estrita e padronizada entre o Centro de Trabalho Responsável e o Grupo de Planejamento. Um determinado Centro de Trabalho só pode ser associado a grupos de planejamento específicos.
- Dados de Campo de Origem: No SAP AG (2025), existe uma lógica de negócio que correlaciona o campo origem com o Tipo de Nota, o Código ABC (que indica a criticidade do equipamento) e a Prioridade da nota que estabelece uma combinação padronizada entre eles.

Estas tabelas formalizam essas relações para que o LLM possa validar se a combinação presente em uma ordem ou nota de manutenção é permitida.

3.2.2 Modulo de Extração, Transformação e Carga (ETL): Dados de notas e ordens

Dada a impossibilidade de obtenção de um único *dataset* unificado a partir do SAP, foi realizada a implantação de um módulo de carregamento de dados, decorrente das diferentes transações, para a consolidação dos dados. Os dados foram extraídos em lotes semanais, organizados em múltiplos arquivos do tipo `.xlsx`, que representam subconjuntos de ordens de manutenção, notas técnicas, códigos de falha, causas, ações corretivas e registros textuais. A estruturação do módulo visou tratar essas múltiplas fontes de forma padronizada e modular.

Para isso, foi implementada uma função genérica denominada `carregar_empilhar_arquivos`, responsável por iterar sobre uma lista de caminhos de arquivos e concatenar os dados em um único *dataframe* padronizado. Esse processo ocorre de forma independente para cada tipo de transação, gerando assim *dataframes* específicos como `df_IW49N` (ordens), `df_IW29` (notas), `df_IW69` (falhas), `df_IW65` (ações realizadas) e `df_TXT_NOTA` (textos descritivos das notas).

Adicionalmente, a função `check_and_fix_index` foi aplicada sistematicamente após cada carregamento, visando assegurar a unicidade dos índices e evitar colisões ou duplicações que poderiam comprometer os joins posteriores. Esse controle é essencial, pois os dados extraídos do SAP podem conter linhas duplicadas ou índices não únicos, especialmente em registros históricos acumulados.

Uma vez carregados e verificados, os *dataframes* de notas (`df_IW49N, df_IW29, df_IW69, df_IW65, df_TXT_NOTA`) foram unificados por meio de operações de mesclagem utilizando a chave comum “Nota”. Esse procedimento resultou no *dataframe* consolidado `df_Notas`, que reúne os principais atributos técnicos e textuais referentes às notificações de manutenção. Da mesma forma, o *dataframe* df_Ordens foi extraído diretamente do `df_IW49N`, contendo os cabeçalhos das ordens após a aplicação de filtros e seleção das colunas relevantes, além da criação de um campo adicional “TIPO DE MANUTENÇÃO” para posterior classificação e aplicação dos critérios de avaliação.

3.2.3 Arquitetura do *Framework*

3.2.3.1 Recuperação de Contexto (RAG) com Biblioteca Pandas

O LLM, por si só, não possui conhecimento sobre as regras de negócio específicas da empresa. A arquitetura RAG foi implementada para "ensinar"o modelo, fornecendo-lhe o contexto necessário para cada avaliação. Este componente é o núcleo da inteligência do *framework*, onde os dados de entrada e o contexto recuperado são orquestrados para formar uma instrução clara para o LLM.

Essa função é responsável por montar o contexto informacional para cada coluna que será avaliada. Executa-se uma busca exata em tabelas auxiliares, se o critério avaliado for o campo “CenTrab respon.” ou “Grp.plnj.PM”, da ordem ou da nota ele busca diretamente no *dataframe* `df_lista_ct_global` para obter informações do centro de trabalho e grupo de planejamento, caso a combinação entre eles não seja exata, não retorna valores e informa que não foi encontrado. Se avalia campos de nota como ‘’campo origem", ‘’tipo de nota", ‘’Prioridade"ou ‘’Código ABC", realiza uma busca se existe o padrão em `df_campo_origem_global` dos valores atuais da nota.

3.2.3.2 Engenharia de *Prompt* com of Thought (CoT)

A metodologia CoT instrui o LLM a "pensar em voz alta", gerando registros de auditoria do seu próprio raciocínio, sendo isso crucial para a explicabilidade e a confiança no sistema. No *framework* proposto, o CoT foi integrado ao pipeline de interação com o modelo com a abordagem *Zero-shot CoT*, que utiliza instruções explícitas, como “vamos pensar passo a passo”, induzindo o modelo a gerar autonomamente a cadeia de raciocínio no *prompt*. Cada *prompt* é estruturado para solicitar:

- PENSAMENTO: com o raciocínio intermediário com sequência lógica e justificativas intermediárias;
- AVALIAÇÃO FINAL: no formato padronizado, incluindo coluna avaliada, conformidade e justificativa concisa no formato: `COLUNA: <Nome> | CONFORMIDADE: [CONFORME / NÃO CONFORME] | JUSTIFICATIVA: <texto>.`

Esse formato estruturado orienta o modelo a justificar suas conclusões explicitamente, garantindo consistência na extração posterior dos resultados. Ou seja, o uso do CoT nos *prompt's* força o modelo a explanar seu “pensamento” e emitir a avaliação esperada em formato tabular, possibilitando a construção de funções de *parsing* (processamento de saídas do modelo), garantindo transparência e automação no pós-processamento.

3.2.3.3 Gestão de Memória Conversacional

A gestão de memória conversacional em LLMs refere-se ao conjunto de estratégias e estruturas utilizadas para manter, recuperar e atualizar informações contextuais ao longo de interações sucessivas com o modelo. Em aplicações que demandam múltiplas etapas de raciocínio, como a utilizada neste trabalho, que exigem validações cruzadas ou análise contextual, torna a capacidade de simular a continuidade conversacional um componente essencial para a coerência e efetividade da avaliação e resposta gerada.

O modelo utilizado, *gemma3:4B* via *Ollama* não possui uma memória nativa. Cada entrada *prompt* é processada de forma independente, e o estado interno não é retido entre chamadas, a menos que seja explicitamente reinjetado por meio do *prompt*. A interação com o LLM foi orquestrada pela biblioteca de abstração *LangChain* e foram empregadas estratégias de engenharia de *prompt* em especial a CoT, para instruir o modelo a detalhar seu raciocínio e usar uma memória simulada por meio da concatenação de históricos relevantes contextuais, reinserindo a entrada em um novo prompt, garantindo continuidade de raciocínio.

Este método reduz drasticamente a redundância de dados, otimiza o processamento e simula uma interação mais natural e inteligente com o modelo. Ao passar para uma nova Ordem ou uma Nota específica, uma sessão completamente nova é criada, garantindo o isolamento do contexto.

3.2.3.4 Interação com o LLM Local

Interface e Comunicação com o modelo é realizado por meio do *Client Ollama* que é uma ferramenta de código aberto que executa LLMs localmente. Sendo realizado via biblioteca *Langchain*, utilizando o `langchain.llms.Ollama` como interface com o servidor LLM local padrão `http://localhost:11434`. Essa escolha permite integração fluida com serviços locais, sem dependência de API's externas, garantindo confidencialidade dos dados.

A configuração de parâmetros é responsável por auxiliar na predição de *tokens* após a entrada ter sido processada por todos os blocos do *Transformer* do LLM, permitindo classificar esses *tokens* pela probabilidade de serem a próxima palavra. A combinação de *temperature*, *top-k* e *top-p* oferece uma geração robusta, consistente com os objetivos analíticos do *framework*.

- *Temperature*: Define a aleatoriedade da saída, onde valores menores geram respostas mais determinísticas e coerentes, enquanto valores maiores permitem criatividade. No contexto de avaliação estruturada, configurações baixas (0.7) reforçam precisão sem sacrificar fluidez.
- *Top-k*: Limita a seleção às k palavras mais prováveis, evitando *tokens* raros que possam introduzir ruído.
- *Top-p*: Inclui apenas *tokens* cuja probabilidade cumulativa atinja o limite p, promovendo equilíbrio entre diversidade e coerência.

Para o *framework* foram utilizados parâmetros recomendados pelo Ollama Inc. (2025). `Temperature = 1, top-k = 64, top-p = 0,95` bem como o `stop = [«end_of_turn>"]` utilizado para delimitação de término da resposta.

A resposta bruta do modelo é tratada por um mecanismo específico de *Parsing*, que localiza o trecho em formato padronizado contido na resposta. Essa técnica confere robustez, tornando o sistema tolerante a caracteres ou textos adicionais no *output* do modelo, evitando falhas no pós-processamento.

4 RESULTADOS

Conforme o apresentado na metodologia, o desenvolvimento do presente *framework* foi concebido como um processo iterativo e incremental. A jornada partiu da definição de um Problema de Negócio até a implementação de uma solução robusta e otimizada. A seguir, detalha-se o passo a passo da construção, desde a arquitetura inicial até os refinamentos finais.

4.1 Prova de conceito e validação da arquitetura base

O primeiro passo foi estabelecer a viabilidade da ideia central, verificando se um LLM local conseguiria, de fato, realizar avaliações semânticas complexas. A primeira atividade prática foi configurar o ambiente de execução local. A ferramenta *Ollama* foi escolhida por sua simplicidade e eficiência em servir modelos de linguagem. O modelo *Gemma-3:4B* foi selecionado como o LLM inicial devido ao seu excelente balanço entre performance e requisitos de hardware, tornando-o ideal para desenvolvimento em um hardware com GPU comum. Um teste inicial, com um *prompt* simples, foi executado no terminal com a chamada do modelo `ollama run gemma3:4b` para confirmar a comunicação entre o *Visual Studio Code* e o servidor *Ollama*.

A primeira versão do *prompt* foi construída de forma simples. Para uma única avaliação de coluna, todo o contexto era fornecido em uma única chamada, sendo os dados completos da ordem e os critérios de avaliação. Embora funcional para uma única validação, essa abordagem demonstrou limitações como a aleatoriedade das respostas e imprecisão na avaliação dos campos.

A confirmação da viabilidade computacional e a identificação dos limites do prompt simples, motivaram o desenvolvimento de uma solução mais robusta e adaptada que possibilite utilizar o LLM de forma personalizada.

4.2 Elaboração do modelo

4.2.1 Módulo auxiliar de extração, transformação e carga dos dados

Nesta etapa, os dados de notas e ordens foram obtidos a partir das transações *standard* de manutenção do sistema SAP (IW29, IW49N, IW65, IW69) e textos de notas associadas. Foi criado o arquivo `Notas_e_Ordens_Avaliar.py` que atua como um módulo dedicado a preparar e fornecer o *dataset* de entrada para o *framework*. Ele encapsula a lógica de definição dos dados que serão o objeto da análise. Utilizou-se o `pandas.read_csv` para ler arquivos exportados em formato .xlsx contendo esses registros. Em seguida, aplicou-se limpeza básica com a remoção de duplicatas com `drop_duplicates` do pandas e padronização de formatos. Essa fase garantiu que cada entrada de ordem de serviço e cada nota de manutenção estivessem

em um formato consistente para processamento subsequente.

Em seguida foi utilizado o `pd.concat` para empilhar em um único *dataframe*, múltiplos arquivos exportados da mesma transação, organizados por número da semana em que as atividades foram realizadas. Foi realizada também a seleção das colunas relevantes de cada *dataframe*. E então, foi replicada a mesma operação para todos os arquivos de cada transação separadamente.

A partir dos *dataframes* gerados pelos arquivos empilhados foi realizado o *merge* ou mesclagem dos *dataframes* `df_IW29, df_IW69, df_IW65 e df_TXT_NOTA` com base na coluna "Nota"dando origem ao `df_Notas`.

E finalmente foi criado o *dataframe* `df_Ordens` a partir do `df_IW49N`, sendo esse, adicionado uma coluna que classifica o tipo de manutenção de cada ordem. Passo importante para definir qual será o critério utilizado de avaliação para cada ordem e sua nota atrelada.

4.2.2 Módulo do *framework* de análise semântica de Notas e Ordens

Para o *framework*, foi desenvolvido o módulo principal denominado `Framework_Analise_Semantica.py`, contendo o código-fonte customizado, responsável por conduzir a avaliação semântica dos dados de ordens e notas de manutenção. Esse arquivo encapsula as seguintes funcionalidades:

- Inicialização do ambiente de execução: foi incluído a configuração do modelo de linguagem *Gemma3:4B* via *Ollama*, parâmetros de geração (*temperatura, top-k, top-p, stop sequences*), além de mecanismos de controle de taxa de requisições e limites de quantidade de ordens a serem avaliadas, ver Figura 6.

```
# --- Configurações de modelo e geração ---
MODEL_NAME = "gemma3:4b"
OLLAMA_HOST = 'http://localhost:11434'

# Parâmetros de geração do LLM
LLM_STOP_SEQUENCES = ["<end_of_turn>", "###"]
LLM_TEMPERATURE = 1
LLM_TOP_K = 64
LLM_TOP_P = 0.95

# --- Limites operacionais ---
REQUESTS_PER_MINUTE_LIMIT = 0       # Sem limitação de chamadas por minuto
MAX_ORDENS_AVALIAR = 2             # Limita avaliação às 2 primeiras ordens


llm = ChatOllama(
    model=MODEL_NAME,
    base_url=OLLAMA_HOST,
    callback_manager=CallbackManager([StreamingStdOutCallbackHandler()]),
    stop=LLM_STOP_SEQUENCES,
    temperature=LLM_TEMPERATURE,
    top_k=LLM_TOP_K,
    top_p=LLM_TOP_P
)
```

Figura 6 – Inicialização do Modelo LLM

Fonte: Elaborada pelo autor, 2025

- Carregamento e pré-processamento dos dados: é realizado por meio de uma função que faz a leitura das planilhas em formato CSV e definem os critérios técnicos de avaliação e tabelas auxiliares de referência (centros de trabalho, campos de origem), além das bases consolidadas de ordens e notas via `Notas_e_Ordens_Avaliar.py`, ver Figura 7.

```
def carregar_tabela_csv(caminho_arquivo):
    try:
        df = pd.read_csv(caminho_arquivo, sep=';', encoding='latin1')
        return df
    except Exception as e:
        print(f"Erro ao carregar {caminho_arquivo}: {e}")
        return None
```

Figura 7 – Função de carregamento das tabelas de critérios e auxiliares

Fonte: Elaborada pelo autor, 2025

- Carregamento dos dados de ordens e notas de manutenção: são recuperados em um *script* auxiliar nomeado `Notas_e_Ordens_Avaliar.py`, que possui os *dataframes* `df_Ordens` e `df_Notas`, ver Figura 8.

```
# Importação dinâmica do módulo auxiliar
file_path_noa = 'C:\\TCC_LLM\\Notas_e_Ordens_Avaliar.py'
spec_noa = importlib.util.spec_from_file_location("Notas_e_Ordens_Avaliar",
                                                  file_path_noa)
Notas_e_Ordens_Avaliar = importlib.util.module_from_spec(spec_noa)
spec_noa.loader.exec_module(Notas_e_Ordens_Avaliar)

# Atribuição dos DataFrames processados
df_Notas = Notas_e_Ordens_Avaliar.df_Notas
df_Ordens = Notas_e_Ordens_Avaliar.df_Ordens
```

Figura 8 – Importação e execução `Notas_e_Ordens_Avaliar.py` e atribuição dos dataframes de ordens e notas

Fonte: Elaborada pelo autor, 2025

- Gerenciamento de conversas estruturadas com CoT + RAG: adotando o padrão *Chain-of-Thought* que exige do LLM um raciocínio em etapas e uma saída formatada, para realizar avaliações semânticas de forma contextualizada, estruturada e auditável. Esse processo é viabilizado pela biblioteca *LangChain*, que permite manter um histórico de interações com o LLM baseado na arquitetura de mensagens sequenciais e padronizada, facilitando o parse automático e a rastreabilidade. As conversas são mantidas por meio de listas de mensagens (*SystemMessage, HumanMessage, AIMessage*), sendo que cada instância de ordem e nota recebe seu próprio "contexto de memória", identificado por uma chave, ver Figura 9. Cada nova conversa inicia com uma *SystemMessage* global, que define o papel e comportamento esperado do modelo é adicionada apenas uma vez por conversa (por chave de ordem ou nota), atuando como contexto estático, ver Figura 10.Após a mensagem de sistema, o *framework* realiza uma mensagem de *priming*, processo de preparar o LLM com o contexto necessário antes de fazer uma pergunta específica. Isso significa apresentar ao LLM todos os dados relevantes da ordem de manutenção ou da nota SAP em uma única mensagem inicial, para que ele inicialize essas informações antes de avaliar qualquer coluna. Ao iniciar a avaliação de uma ordem ou nota, o script verifica se aquela entidade já possui uma conversa iniciada. Se não houver, ele cria uma nova conversa e adiciona duas mensagens: *SystemMessage* – define o papel do modelo e *HumanMessage* – contém o *priming* com os dados completos da ordem ou nota, ver Figura 11.Para cada coluna a ser avaliada, o *framework* constrói um novo *prompt* que se baseia na memória existente (mensagens anteriores), inclui o critério técnico específico, acrescenta o contexto semântico recuperado via RAG e exige que o modelo utilize*Chain-of-Thought*, ver Figura 12. A chamada ao modelo é feita incluindo todo o histórico da conversa até o momento, mantendo coerência e continuidade, ver Figura 13.

```
# Dicionário que armazena o histórico de conversas
order_note_conversations = {}

# Chave: (ordem_id, nota_id), sendo nota_id = None para cabeçalho
conversation_key_order = (ordem_id, None)
conversation_key_note = (ordem_id, nota_id_atual)
```

Figura 9 – Dicionário que armazena o histórico de conversas e Chaves de conversação

Fonte: Elaborada pelo autor, 2025

```
system_message_global = SystemMessage(
    content=textwrap.dedent("""
    Você é um especialista em checklist de ordens e notas de manutenção.
    Sua tarefa é avaliar colunas específicas com base nos dados fornecidos e critérios.
    Forneça sempre a avaliação no formato:
    COLUNA: [Nome] | CONFORMIDADE: [CONFORME/NÃO CONFORME] | JUSTIFICATIVA: [Texto]
    ###""")
)
```

Figura 10 – Mensagem de Sistema *SystemMessage*

Fonte: Elaborada pelo autor, 2025

```
    initial_order_priming_content = textwrap.dedent(f"""
    Revise os DADOS COMPLETOS do CABEÇALHO da ORDEM de Manutenção (ID: {ordem_id},
    Tipo: {tipo_manutencao_ordem}) e as informações da primeira nota associada.

    DADOS DO CABEÇALHO DA ORDEM:
    {dados_cabecalho_para_priming_prompt}

    Estou pronto para avaliar as colunas individualmente.S
    """)

    # Adiciona mensagem de priming ao histórico de conversa
    order_note_conversations[conversation_key_order].append(
        HumanMessage(content=initial_order_priming_content)
    )
```

Figura 11 – Mensagem de *priming* com os dados da ordem ou da nota

Fonte: Elaborada pelo autor, 2025

```
prompt_cabecalho_cot_individual = textwrap.dedent(f"""
Avalie a COLUNA: '{coluna_sendo_avaliada}' da ORDEM ID {ordem_id}.
CRITÉRIO: {criterio_especifico_df['CRITÉRIO DA COLUNA AVALIADA'].iloc[0]}
CONTEXTO ADICIONAL:
{contexto_coluna_cabecalho}

INSTRUÇÕES:
1. PENSAMENTO: (explique passo a passo)
2. AVALIAÇÃO FINAL:
   COLUNA: {coluna_sendo_avaliada} | CONFORMIDADE: ... | JUSTIFICATIVA: ...
###
""")
```

Figura 12 – Geração do *prompt* de avaliação da coluna

Fonte: Elaborada pelo autor, 2025

```
current_messages_for_llm_call = list(order_note_conversations[conversation_key_order])
current_messages_for_llm_call.append(HumanMessage(content=prompt_cabecalho_cot_individual))

response_obj = llm.invoke(current_messages_for_llm_call)
resposta_bruta = response_obj.content.strip()

order_note_conversations[conversation_key_order].append(AIMessage(content=resposta_bruta))
```

Figura 13 – Execução com o Histórico de Conversa

Fonte: Elaborada pelo autor, 2025

- Registro dos resultados, com geração de dois arquivos de saída — um log detalhado em texto com histórico completo das interações, e uma tabela CSV estruturada contendo o numero da nota e ordem, conformidade, justificativas e *timestamp* — permitindo auditoria e posterior análise estatística.

Na Figura 14 temos a visualização da arquitetura do *framework* no formato de diagrama.

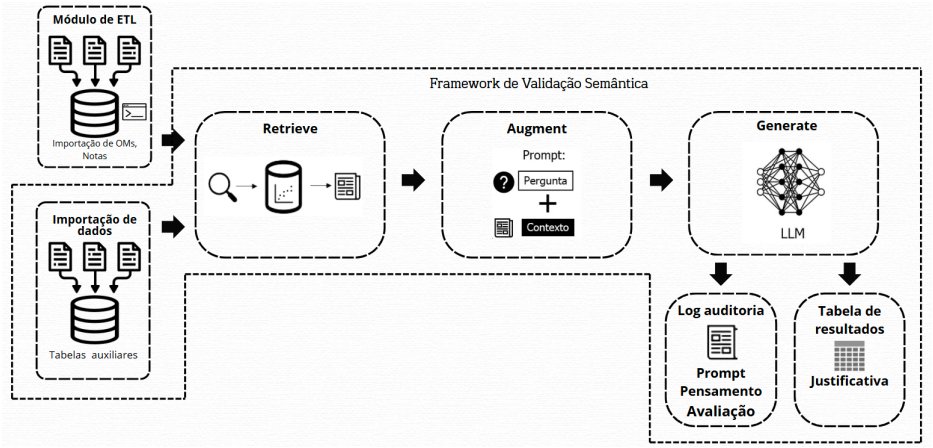

Figura 14 – Arquitetura do *framework*

Fonte: Elaborada pelo autor, 2025

4.3 Análise dos resultados

Foram avaliadas semanticamente uma amostra de 10 ordens e notas de serviço, cada uma com 9 e 13 campos avaliados respectivamente, perfazendo 220 avaliações individuais. Dos 220 casos avaliados, 29 apresentaram erro de avaliação semântica, resultando em uma taxa de acerto de aproximadamente 86,8%. Estes resultados quantitativos indicam que, em termos gerais, o sistema apresentou desempenho robusto na identificação de inconsistências semânticas nos campos avaliados. Os resultados qualitativos, baseados na análise das justificativas geradas pelo modelo, corroboraram esse bom desempenho.

O *framework* demonstrou capacidade satisfatória de identificar a maioria das não conformidades semânticas presentes nos documentos analisados. Em muitos casos, as justificativas fornecidas pelo modelo se mostraram úteis, claras e coerentes com os critérios técnicos estabelecidos para validação, reforçando a confiança nas decisões automatizadas. Adicionalmente, o tempo médio de processamento por avaliação foi de aproximadamente 12 segundos, valor considerado adequado para a operação prática, mesmo diante de limitações locais.

É importante salientar que os contextos com terminologia imprecisa ou informações subentendidas dificultaram a interpretação correta de alguns campos, elevando a taxa de erro semântico nesses casos. O desempenho do modelo mostrou-se fortemente dependente da qualidade e completude do contexto, o que aumentou a probabilidade de respostas incorretas.

5 CONCLUSÃO

A precisão conceitual e semântica no preenchimento dos campos de ordens e notas de manutenção não é um detalhe técnico, mas sim um pilar fundamental para a confiabilidade dos ativos, a integridade dos indicadores de desempenho e a eficácia dos planos de manutenção. Registros ambíguos ou incorretos ameaçam a tomada de decisão técnica e estratégica de forma profunda.Este trabalho contribui diretamente para a área de manutenção industrial ao propor um mecanismo inteligente de auditoria semântica, capaz de verificar conformidade textual com base em critérios técnicos e históricos recuperados dinamicamente. A geração de justificativas estruturadas (*Chain-of-Thought*), a integração com dados históricos e os registros em *log* auditável destacam-se como diferenciais do sistema.

A fase de desenvolvimento culminou em um *framework* funcional que demonstrou capacidade satisfatória na identificação de não conformidades semânticas. A arquitetura implementada, que prioriza a busca em bases de conhecimento tabulares, formalizando regras de negócio e do SAP antes de recorrer a mecanismos de inferência do LLM, provou ser uma abordagem equilibrada, frente ao resultado obtido de 86,8% da amostra analisada. Sendo este um indicador de robustez do sistema em identificar inconsistências.

Contudo, limitações foram observadas, especialmente em relação à ambiguidade dos contextos preenchidos, que comprometeu a precisão em determinados cenários. A qualidade da entrada (abreviações, descrições vagas, ausentes ou mal estruturadas) impactou diretamente a capacidade interpretativa do modelo, demonstrando que a confiabilidade das respostas está fortemente atrelada à integridade semântica dos dados, sendo esse um aspecto crítico, mas frequentemente negligenciado em ambientes industriais.

5.1 Sugestão para trabalhos futuro

O projeto estabelece uma base para pesquisas subsequentes e desenvolvimento contínuo. Considerar o *fine-tuning* do modelo *Gemma-3* (ou outro LLM local) com dados específicos de ordens e notas de manutenção já validadas por especialistas. Isso poderia aumentar a acurácia na interpretação da terminologia do domínio e na adesão aos formatos de resposta. A implementação de aprendizado ativo realizando o aprendizado por reforço com *feedback* humano para refinamento do modelo, também é uma alternativa para o aumento da acurácia do modelo.

Pesquisas futuras também podem explorar a utilização de conectores para integrar o *framework* diretamente aos sistemas de gestão de manutenção, permitindo que as validações ocorram em tempo real, e que os resultados sejam retroalimentados no sistema de origem. Além disso, é possível realizar a investigação da aplicação de diferentes tipos de LLMs, como os multimodais, para analisar documentos e fotos que possam estar anexados às ordens de

manutenção, bem como, realizar um estudo comparativo entre diferentes LLMs de código aberto executáveis localmente para avaliar qual oferece o melhor custo-benefício entre desempenho, acurácia e hardware para esta tarefa específica.

Em última análise, o projeto abre caminhos para a manutenção inteligente e serve como um catalisador para a inovação contínua, visando não apenas resolver desafios práticos imediatos, mas também gerar novas questões de pesquisa e desenvolvimentos tecnológicos subsequentes. Desta forma, a inovação contínua é estimulada pela própria natureza evolutiva da solução, que pode ser aprimorada, expandida e adaptada para enfrentar desafios cada vez mais complexos na busca por uma gestão de ativos verdadeiramente preditiva, prescritiva e autônoma.

REFERÊNCIAS

ABNT. Norma Brasileira, *NBR 5462: Confiabilidade e Mantenabilidade – Terminologia*. Rio de Janeiro: [s.n.], 1994. Citado na página 17.

ABRAMAN. *Indústria 5.0: o complemento da Indústria 4.0 e o foco na confiabilidade humana*. 2023. Disponível em: <https://abramanoficial.org.br/publicacoes/noticias/pindustria-50-o-complemento-da-industria-40-e-o-foco-na-confiabilidade-humanap>. Citado na página 13.

ABRAMAN. *Manutenção Assistida por IA: Como o uso de Inteligência Artificial redefine processos na Manutenção industrial*. 2023. Disponível em: <https://abramanoficial-.org.br/publicacoes/noticias/pmanutencao-assistida-por-ia-como-o-uso-de-inteligencia-artificial-redefine-processos-na-manutencao-industrialp>. Citado 2 vezes nas páginas 13 e 29.

BATINI C.; SCANNAPIECO, M. *Data and Information Quality: Dimensions, Principles and Techniques*. Cham: Springer, 2016. Citado na página 28.

CONTE, A. et al. The impact of data quality on maintenance work order analysis: A case study in hvac work durations. In: *6th European Conference of the Prognostics and Health Management Society*. [S.l.: s.n.], 2021. Citado na página 13.

GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. 4. ed. São Paulo, Brazil: Atlas, 2007. Define pesquisa experimental como seleção de variáveis, controle e observação sistemática dos efeitos. Citado na página 36.

JARDINE, A. K.; LIN, D.; BANJEVIC, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 2006. v. 20, n. 7, p. 1483–1510, 2006. ISSN 0888-3270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0888327005001512>. Citado na página 28.

LEME, M. T.; OUTROS. Pesquisa aplicada: conceitos e abordagens. *Periódicos FGV*, 2018. 2018. Publicado no repositório FGV, aborda definição e aplicação prática da pesquisa aplicada. Citado na página 36.

LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 2020. v. 33, p. 9459–9474, 2020. Citado 4 vezes nas páginas 31, 32, 34 e 35.

LEWIS, P. et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. Disponível em: <https://arxiv.org/abs/2005.11401>. Citado na página 14.

LI, J.; LI, J. Memory, consciousness and large language model. *arXiv preprint arXiv:2401.02509*, 2024. 2024. Citado 2 vezes nas páginas 32 e 33.

LI, Y. et al. *Large Language Models for Manufacturing*. 2024. Disponível em: <https://arxiv-.org/abs/2410.21418>. Citado na página 14.

MOBLEY, R. K. *Maintenance Engineering Handbook*. 7. ed. New York: McGraw-Hill, 2008. Citado 2 vezes nas páginas 19 e 20.

MONK, E.; WAGNER, B. *Concepts in Enterprise Resource Planning*. 4. ed. [S.l.]: Cengage Learning, 2013. Citado na página 24.

NAQVI, S. M. et al. Generating semantic matches between maintenance work orders for diagnostic decision support. *Annual Conference of the PHM Society*, 2022. v. 14, 10 2022. Citado na página 32.

Ollama Inc. *gemma3:4b/params*. 2025. https://ollama.com/library/gemma3:4b/blobs/3116c5225075. Acesso em: 02 maio 2025. Citado na página 41.

PINTO ALAN KARDEC; XAVIER, J. d. A. N. *Manutenção: Função Estratégica*. 4. ed. Rio de Janeiro: Qualitymark, 2013. ISBN 978-85-7303-323-6. Citado 5 vezes nas páginas 17, 18, 19, 20 e 25.

RAZA M., J. Z. R. M. e. a. *Industrial applications of large language models*. 2025. 13755 p. Published 21 April 2025. Open access. Citado na página 14.

SAHOO, P. et al. *A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications*. 2025. Disponível em: <https://arxiv.org/abs/2402.07927>. Citado na página 33.

SAP AG. *SAP PM (Plant Maintenance) Documentation*. 2025. Documentação disponível no SAP Help Portal, com atualização anual. O ano indicado refere-se ao ano da consulta, dada a natureza da publicação. Disponível em: <https://help.sap.com/docs/SAP_S4HANA_CLOUD-/2dfa044a255f49e89a3050daf3c61c11/bb1e318b3718425dbaa210bde2263228.html>. Citado 4 vezes nas páginas 13, 24, 25 e 38.

SEXTON, T. et al. Hybrid datafication of maintenance logs from ai-assisted human tags. In: *2017 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2017. p. 1769–1777. Citado na página 13.

STANDARDIZATION, I. O. F. *Asset Management — Overview, Principles and Terminology*. Genebra, Suíça, 2014. Disponível em: <https://www.iso.org/standard/55088.html>. Citado na página 22.

STANDARDIZATION, I. O. F. *Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment*. Genebra, Suíça, 2016. Citado na página 13.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, 2017. v. 30, 2017. Citado 2 vezes nas páginas 30 e 31.

WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022. v. 35, p. 24824–24837, 2022. Citado na página 34.

XU, K. et al. Crp-rag: A retrieval-augmented generation framework for supporting complex logical reasoning and knowledge planning. *Electronics*, 2024. v. 14, p. 47, 12 2024. Citado na página 35.

ZHANG, Z. et al. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022. 2022. Citado na página 34.