UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E TRANSPORTES

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

DESENVOLVIMENTO E VALIDAÇÃO DE UM AGENTE INTELIGENTE PARA AUTOMATIZAR O SUPORTE TÉCNICO DIAGNÓSTICO EM SISTEMAS AVANÇADOS DE PLANEJAMENTO E SEQUENCIAMENTO (APS)

LETÍCIA GONÇALVES ALMEIDA LACERDA

Orientador: Michel Anzanello

PORTO ALEGRE
JUNHO/2026

SUMÁRIO

RESUMO 4

ABSTRACT 5

GLOSSÁRIO 6

1. INTRODUÇÃO 8

2. REFERENCIAL TEÓRICO 10

2.1. DESAFIOS NA PRESTAÇÃO DE SUPORTE EM SISTEMAS APS 10

2.2. MAPEAMENTO E AUTOMAÇÃO DE PROCESSOS DE SUPORTE TÉCNICO 12

2.3. DIAGNÓSTICO AUTOMATIZADO POR MEIO DE AGENTES CONVERSACIONAIS 13

3. PROCEDIMENTOS METODOLÓGICOS 15

3.1. CLASSIFICAÇÃO DA PESQUISA 15

3.2. ETAPAS DO TRABALHO 15

3.2.1. REVISÃO DA LITERATURA E ANÁLISE DOCUMENTAL 16

3.2.2. MAPEAMENTO E MODELAGEM DO PROCESSO DE DIAGNÓSTICO 16

3.2.3. DEFINIÇÃO DA ARQUITETURA E DESENVOLVIMENTO DO FLUXO DE AUTOMAÇÃO 17

3.2.4. IMPLEMENTAÇÃO DO AGENTE ESPECIALISTA 18

3.2.5. VALIDAÇÃO FUNCIONAL DO PROTÓTIPO 18

3.2.6. COMPARAÇÃO DE PERFOMANCE E EFICIÊNCIA 19

4. RESULTADOS E DISCUSSÃO 19

4.1. REVISÃO DA LITERATURA E ANÁLISE DOCUMENTAL 20

4.2. MAPEAMENTO E MODELAGEM DO PROCESSO DE DIAGNÓSTICO . 23

4.3. DEFINIÇÃO DA ARQUITETURA E DESENVOLVIMENTO DO FLUXO DE AUTOMAÇÃO 24

4.4. IMPLEMENTAÇÃO DO AGENTE ESPECIALISTA 29

4.5. VALIDAÇÃO FUNCIONAL DO PROTÓTIPO 30

4.6. COMPARAÇÃO DE PERFORMANCE E EFICIÊNCIA 32

5. CONCLUSÃO 36

6. REFERÊNCIAS 38

7. APÊNDICES 40

RESUMO

Sistemas Avançados de Planejamento e Sequenciamento (APS) caracterizam-se pelo elevado grau de customização exigido por cada cliente, o que torna o suporte técnico uma atividade morosa e dependente de conhecimento especializado. Esse cenário sobrecarrega equipes de *Customer Success* com chamados diagnósticos repetitivos e reduz o tempo disponível para atividades de evolução do sistema. Este trabalho teve como objetivo desenvolver e validar um agente inteligente para automatizar a etapa inicial de diagnóstico de chamados de suporte no sistema Opcenter APS. A pesquisa, de natureza aplicada e abordagem mista, foi conduzida em uma empresa de consultoria especializada e estruturou-se em seis etapas: revisão da literatura, análise documental de 2.856 chamados históricos, mapeamento do processo de diagnóstico via notação BPMN, construção de fluxo de automação na plataforma *low-code* N8N integrado a um agente de Inteligência Artificial baseado em GPT-5 com três bases de conhecimento, validação funcional e comparação de desempenho com o atendimento humano. A validação foi realizada com 14 chamados, sendo sete reais e sete simulados, avaliando-se acurácia, eficiência e qualidade das respostas geradas. O protótipo alcançou taxa de acerto de 71% tanto no diagnóstico quanto no escalonamento, com redução de 99,77% no tempo médio de primeira resposta em relação ao atendimento humano. A análise qualitativa evidenciou que, mesmo nos casos de diagnóstico parcial, as conferências sugeridas mantiveram-se aderentes ao escopo do problema, e que as falhas se concentram em situações cuja causa-raiz depende de contexto operacional não documentado. Conclui-se que a integração entre automação de processos e agentes especialistas é viável e efetiva como ferramenta de apoio ao consultor, contribuindo para a otimização do atendimento em ambientes industriais altamente customizados.

Palavras-chave: Sistemas APS; automação de processos; agentes especialistas; suporte técnico; Customer Success.

ABSTRACT

Advanced Planning and Scheduling (APS) systems are characterized by the high degree of customization required by each client, which makes technical support a time-consuming activity that depends on specialized knowledge. This scenario overloads Customer Success teams with repetitive diagnostic tickets and reduces the time available for activities related to system evolution. This study aimed to develop and validate an intelligent agent to automate the initial diagnostic stage of support tickets in the Opcenter APS system. The applied, mixed-methods research was conducted in a specialized consulting company and was structured in six stages: literature review, document analysis of 2,856 historical tickets, mapping of the diagnostic process through BPMN notation, construction of an automation flow on the low-code N8N platform integrated with a GPT-5-based Artificial Intelligence agent supported by three knowledge bases, functional validation, and performance comparison with human service. Validation was carried out with 14 tickets, seven real and seven simulated, assessing accuracy, efficiency, and quality of the generated responses. The prototype achieved a 71% success rate for both diagnosis and escalation, with a 99.77% reduction in mean time to first response compared to human service. The qualitative analysis showed that, even in cases of partial diagnosis, the suggested checks remained aligned with the scope of the problem, and that failures concentrate on situations in which the root cause depends on operational context that is not documented. It is concluded that the integration of process automation and specialist agents is feasible and effective as a support tool for consultants, contributing to the optimization of service delivery in highly customized industrial environments.

Keywords: APS systems; process automation; expert agents; technical support; Customer Success.

GLOSSÁRIO

Advanced Planning and Scheduling (APS) – Sistema de Planejamento e Sequenciamento Avançado da Produção.

API (Application Programming Interface) – Interface de Programação de Aplicações utilizada para comunicação entre sistemas.

AWS S3 (Amazon Web Services Simple Storage Service) – Serviço de armazenamento em nuvem utilizado para armazenar arquivos.

BPMN (Business Process Modeling Notation) – Notação gráfica utilizada para modelagem de processos de negócio.

Chatbot – Programa capaz de simular conversas humanas de forma automatizada.

Common Table Expression (CTE) – Estrutura SQL utilizada para criar consultas temporárias reutilizáveis, frequentemente em consultas recursivas.

Customer Success (CS) – Área responsável por garantir que o cliente obtenha sucesso com o uso do produto.

Dashboard – Painel visual utilizado para monitoramento de indicadores e métricas.

Database – Banco de dados.

Endpoint – Ponto de acesso de uma API ou serviço web.

Escalonamento (Scaling) – Encaminhamento de um chamado para um especialista humano.

Goal-oriented – Orientado a objetivos.

GPT-5 – Modelo de linguagem da OpenAI utilizado para processamento de linguagem natural.

Helpdesk – Plataforma de central de ajuda e gerenciamento de chamados técnicos.

HTML (HyperText Markup Language) – Linguagem de marcação utilizada para estruturar páginas web.

HTTP (HyperText Transfer Protocol) – Protocolo utilizado para comunicação entre sistemas web.

HTTP Request – Requisição realizada via protocolo HTTP.

Interface – Meio de interação entre usuário e sistema.

JavaScript – Linguagem de programação utilizada para desenvolvimento web e automações.

JSON (JavaScript Object Notation) – Formato textual utilizado para troca estruturada de dados entre sistemas.

LangChain – Framework utilizado para construção de aplicações com modelos de linguagem.

Low-code – Abordagem de desenvolvimento baseada em pouca programação manual.

Markdown – Linguagem de marcação leve utilizada para formatação textual.

Mean Time for First Response (MTFR) – Tempo médio para primeira resposta ao chamado.

Mean Time for Resolution (MTR) – Tempo médio total para resolução do chamado.

N8N – Plataforma de automação de fluxos low-code.

OpenAI GPT-5 – Modelo de Inteligência Artificial utilizado no agente conversacional.

Payload – Conjunto de dados enviados em uma requisição.

POST Method – Método HTTP utilizado para envio de dados.

PostgreSQL – Sistema gerenciador de banco de dados relacional.

Prompt – Conjunto de instruções fornecidas ao modelo de IA.

Prompt Engineering – Processo de construção e refinamento de prompts.

SaaS (Software as a Service) – Modelo de software disponibilizado como serviço pela internet.

Setup – Configuração inicial ou preparação de máquina/processo.

Slack – Plataforma corporativa de comunicação.

Trigger – Evento que inicia automaticamente um fluxo ou processo.

View – Tabela de visualização de dados.

Webhook – Mecanismo de comunicação automática entre sistemas baseado em eventos.

1. INTRODUÇÃO

Atualmente, o ambiente da indústria de manufatura caracteriza-se por um alto nível de competitividade (Simchi-Levi et al., 2000). Para enfrentar os desafios decorrentes dessa competição, um número crescente de empresas manufatureiras tem adotado métodos da Indústria 4.0, como digitalização e Sistemas Ciberfísicos (CPS). Esses métodos descentralizam a tomada de decisão dos seres humanos para análises baseadas em sistemas computacionais, possibilitando elevados níveis de orientação a serviços (Hermann et al., 2015).

No centro dessa transformação industrial, os sistemas de *Advanced Planning and Scheduling* (APS), ou Sequenciamento e Planejamento Avançados de Produção, desempenham um papel relevante. Esses sistemas auxiliam a indústria de manufatura por meio do uso de algoritmos matemáticos para otimizar e planejar operações com base em restrições do mundo real. O sucesso na adoção de sistemas APS depende de uma modelagem consistente e atualizada do planejamento de produção no sistema (Mauergauz, 2016). No entanto, a manutenção desses sistemas apresenta desafios significativos, uma vez que sua relevância e utilidade dependem de processos contínuos de melhoria (Bennett & Rajlich, 2000). Esse cenário está diretamente relacionado à necessidade de integração e sincronização com outros sistemas de manufatura, os quais também evoluem constantemente para atender às demandas dos clientes.

Este estudo concentra-se em otimizar o atendimento de suporte de um time de *Customer Success*. O *Customer Success* é a unidade organizacional responsável por fornecer ao cliente a melhor versão possível do produto (Michael Porter & Heppelmann, 2015). As principais funções desse setor são o suporte técnico e a melhoria contínua. Considerando que os sistemas APS são altamente customizáveis, a demanda por chamados de suporte é constante, e sua análise tende a ser morosa, uma vez que cada modelagem é única e o diagnóstico completo envolve múltiplas etapas. De acordo com Bennett & Rajlich (2000), um software ou sistema bem-sucedido estimula solicitações de mudança e melhoria por parte dos usuários. Para viabilizar esse processo de melhoria contínua, é necessário minimizar o tempo dedicado a chamados de suporte técnico, para que o tempo dos consultores seja voltado à melhoria contínua do sistema.

Nesse contexto, a automação de solicitações simples por meio de um *chatbot*, um modelo digital de conversação, pode otimizar o tempo de atendimento ao cliente, reduzindo a necessidade de conhecimento especializado de um consultor para responder

a questões básicas e simplificando o trabalho de verificação quando necessário. A principal justificativa para a utilização de *chatbots* é o aumento da produtividade, uma vez que, geralmente, tais recursos fornecem respostas imediatas (Brandtzaeg & Følstad, 2017). Dessa forma, os *chatbots* podem auxiliar consultores de suporte de sistemas APS a obter respostas mais rapidamente, o que desonera sua carga, permitindo dedicar mais tempo a intervenções mais especializadas.

Sob uma perspectiva prática, o principal problema deste estudo reside no fato de que consultores de APS despendem uma quantidade significativa de tempo realizando tarefas repetitivas de diagnóstico. Essas atividades, embora estruturadas em etapas previamente definidas, exigem conhecimento específico sobre a modelagem de cada cliente, o que torna sua execução morosa. Um exemplo disso é a análise dos critérios que levaram à alocação de uma ordem de produção a um determinado centro de trabalho, processo que envolve múltiplas verificações sequenciais.

A Inteligência Artificial (IA) pode ser utilizada para automatizar passos e simplificar o serviço de atendimento, ao seguir etapas que simulam a cognição humana (Engel et al., 2022). Isso conduz à seguinte questão de pesquisa: como a integração entre um chatbot e uma plataforma de automação pode contribuir para a otimização do suporte técnico no sistema Opcenter APS, por meio da replicação de análises e deduções lógicas baseadas em regras de negócio?

Diante disso, o objetivo deste trabalho é propor e desenvolver um protótipo de *chatbot* para automação da lógica inicial no atendimento de chamados diagnósticos no sistema Opcenter APS. Para atingir esse objetivo, foram definidos os seguintes objetivos específicos: (i) mapear o processo padrão de atendimento de chamados diagnósticos; (ii) desenvolver um fluxo automatizado de diagnóstico utilizando a plataforma de automação N8N; (iii) integrar o *chatbot* à plataforma de atendimento ao cliente já existente; e (iv) validar a efetividade do protótipo na resolução de solicitações comuns de suporte técnico.

Este projeto é relevante para a empresa, uma vez que otimiza a atuação dos consultores, considerando que sistemas híbridos tendem a apresentar maior eficiência de serviço (Messinger et al., 2009). Para os clientes da empresa, o projeto pode aumentar a produtividade e a disponibilidade de consultores para o desenvolvimento de melhorias em seus sistemas. Do ponto de vista acadêmico, este trabalho contribui para a literatura ao apresentar uma implementação prática de automação cognitiva em um nicho industrial especializado, integrando conceitos de engenharia de produção, tecnologia da informação e inteligência artificial.

Além dessa Introdução, o estudo estrutura-se em quatro seções: a Seção 2 apresenta o referencial teórico, abordando conceitos de sistemas APS, padrões e desafios do suporte técnico e o uso de *chatbots* na automação de sistemas. A Seção 3 descreve os métodos do projeto. A Seção 4 apresenta os resultados do desenvolvimento do protótipo. Por fim, a Seção 5 discute as implicações do estudo e apresenta as considerações finais, além de apontar direções para melhorias futuras na área de pesquisa.

2. REFERENCIAL TEÓRICO

Esta seção apresenta os fundamentos teóricos que sustentam o trabalho. Inicialmente, discute-se o papel dos sistemas APS no contexto da Indústria 4.0, bem como os principais desafios relacionados à sua implementação e suporte. Em seguida, aborda-se o uso de plataformas *low-code* no mapeamento e na automação de processos de serviço. Por fim, são apresentados os conceitos relacionados a agentes inteligentes, com foco no papel de agentes conversacionais e sistemas especialistas de IA na automação de serviços de consultoria.

2.1. DESAFIOS NA PRESTAÇÃO DE SUPORTE EM SISTEMAS APS

Como referido anteriormente, o ambiente contemporâneo da manufatura é caracterizado por elevados níveis de competitividade, exigindo das empresas decisões cada vez mais eficientes e economicamente inteligentes, para se manterem relevantes em um cenário de produção ágil para demandas altamente personalizadas (Simchi-Levi et al., 2000). Nesse contexto, Kagermann et al. (2016) destacam que sistemas orientados a dados são essenciais para a otimização da produção e para a geração de valor ao produto. Assim, os sistemas APS configuram-se como ferramentas relevantes para a análise de diferentes espaços produtivos, permitindo ao usuário apoiar a tomada de decisão no processo de sequenciamento da produção (Ivert & Jonsson, 2010).

Tais sistemas modelam plantas produtivas, considerando capacidade finita, previsões de demanda e disponibilidade de recursos. Com base nessas informações e restrições, estruturam problemas de otimização matemática voltados à redução de *setups*, preparações de máquina, e ao atendimento dos prazos de demanda (Mauergauz, 2016). Devido a sua flexibilidade, os sistemas APS são fundamentais para a tradução de objetivos estratégicos em planos de produção otimizados e executáveis. No entanto, a implementação e a manutenção desses sistemas não são processos triviais, uma vez que

a modelagem de ambientes produtivos e o processamento de seus dados constituem atividades complexas e contínuas. Além disso, as indústrias passam por mudanças frequentes para acompanhar avanços tecnológicos e variações na demanda, o que reforça a necessidade de constante adaptação desses sistemas (Mauergauz, 2016).

Bennett & Rajlich (2000) ressaltam que as atividades de manutenção e evolução de *software* são custosas e demandam tempo, embora sejam inevitáveis, já que mudanças externas, frequentemente, exigem ajustes para que o sistema permaneça relevante. Nesse sentido, equipes de suporte técnico precisam otimizar a alocação de seu tempo, sendo capazes de diagnosticar rapidamente problemas mais simples, para terem maior disponibilidade para tais evoluções. Essa priorização é especialmente relevante em empresas que operam sob o modelo de *software* como um serviço, ou *Software as a Service* (SaaS), conforme descrito por Camargo (2019). Segundo os autores, a área de sucesso do cliente, ou *Customer Success* (CS), das empresas desempenha um papel fundamental na manutenção e evolução do sistema, uma vez que as interações entre cliente e consultor geram resultados valiosos, tanto para a melhoria de funcionalidades existentes quanto para o desenvolvimento de novas soluções.

A complexidade do *Customer Success* em sistemas APS é ampliada por sua natureza altamente customizável. O valor desses sistemas reside na sua capacidade de modelar com precisão a realidade de diferentes plantas produtivas, incorporando suas regras e lógicas de produção (Mauergauz, 2016). Embora a customização seja essencial para o sucesso da implementação, ela torna as etapas de manutenção e evolução mais desafiadoras para as equipes de suporte (Bennett & Rajlich, 2000). Cada modelo representa uma configuração única do sistema, adaptada às necessidades específicas do cliente, o que exige conhecimento especializado por parte de consultores para a resolução de chamados.

A necessidade de conhecimento prévio e especializado configura-se como um gargalo para as equipes de *Customer Success*. Embora seu objetivo seja garantir que o cliente extraia o máximo valor do sistema (Camargo, 2019), a rotina dessas equipes é frequentemente marcada por um alto volume de chamados diagnósticos repetitivos. Esse cenário gera um conflito de prioridades, uma vez que todos os chamados devem ser atendidos em tempo hábil, ao mesmo tempo em que é necessário dedicar esforços à identificação e implementação de melhorias e novas funcionalidades que agreguem valor ao sistema e aos serviços associados (Bennett & Rajlich, 2000).

2.2. MAPEAMENTO E AUTOMAÇÃO DE PROCESSOS DE SUPORTE TÉCNICO

O mapeamento de serviços constitui uma etapa fundamental para o desenvolvimento de agentes automatizados, pois estabelece a base lógica para os fluxos de automação (Deufemia, 2007). Nesse contexto, a *Business Process Modeling Notation* (BPMN), ou Notação de Modelagem de Processos de Negócio, apresenta-se como uma ferramenta adequada para a modelagem de processos orientados a serviços, ao permitir a representação clara de pontos de decisão e dos diferentes caminhos possíveis, com base nas interações com o sistema. Deufemia (2007) também diferencia abordagens orientadas a objetivos (*goal-oriented*) e orientadas a serviços (*service-oriented*), destacando que esta última é mais adequada em situações nas quais o resultado desejado não está completamente definido. Esse entendimento indica que as atividades desempenhadas por consultores de APS podem ser melhor representadas por agentes orientados a serviços, uma vez que envolvem procedimentos estruturados, baseados na verificação sequencial de elementos específicos do sistema e na formulação de conclusões a partir dessas análises.

Uma vez mapeado, o processo pode ser automatizado. Nesse contexto, Hirzel (2023) destaca que plataformas *low-code*, ou seja, plataformas em que sistemas podem ser programados com pouco ou nenhum código, permitem que profissionais especialistas em suas áreas automatizem suas próprias atividades sem a necessidade de conhecimentos avançados em programação ou do suporte direto de desenvolvedores. Essa abordagem reduz a dependência de soluções complexas e de alto custo, ao mesmo tempo em que viabiliza a automação de tarefas repetitivas baseadas em regras.

Plataformas de automação *low-code*, como o N8N, utilizam interfaces gráficas que permitem ao usuário estruturar fluxos de automação de forma visual. Nesse ambiente, o profissional pode organizar as etapas do processo por meio de nós interconectados, cada um representando uma ação específica. Essa lógica de construção se assemelha à de diagramas BPMN, uma vez que possibilita a representação clara do fluxo do processo, com seus caminhos e pontos de decisão (McFeetors & Pant, 2022), conforme ilustrado na Figura 1. A figura representa um fluxo que lê um relatório no *Google Sheets*, cria um arquivo do relatório, e compartilha ele na plataforma *Slack*, além de guardá-lo na plataforma *AWS S3*.

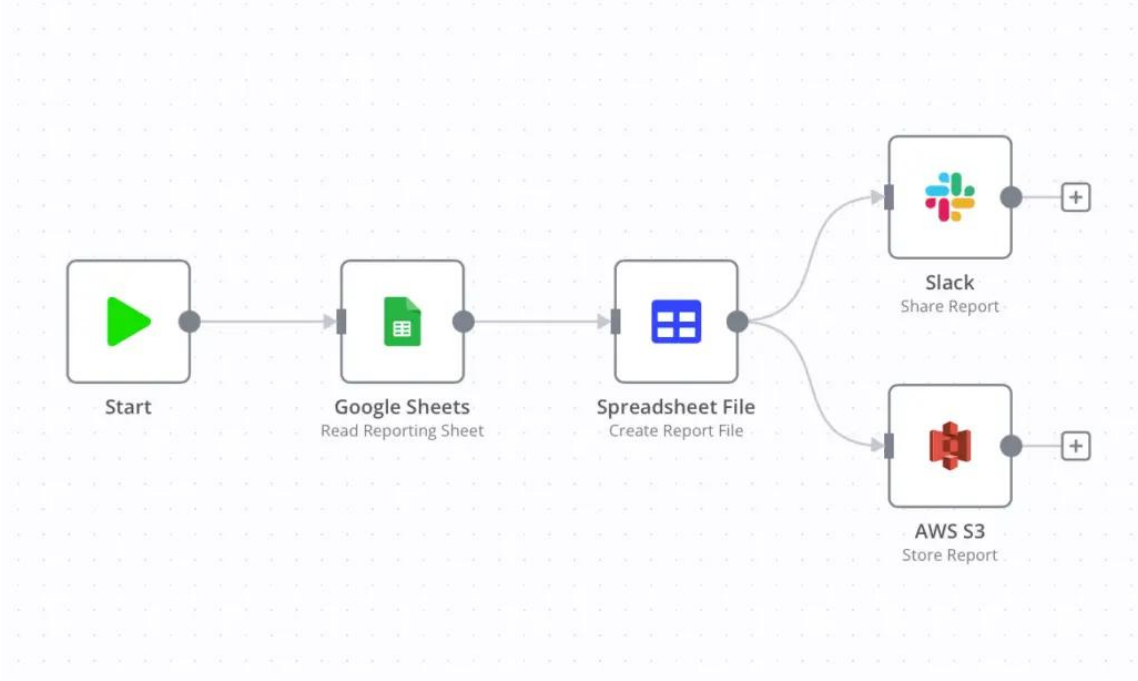


Figura 1 - Exemplo de fluxo de automação no N8N. Disponível em: https://maiconramos.com/o-que-e-n8n-a-solucao-definitiva-para-automacao/ . Acesso em: 07 mai. 2026.

Cunha (2024) destaca os benefícios da automação por meio de plataformas *low-code*, ao analisar o uso do N8N em atividades contábeis. Os resultados obtidos demonstram que a ferramenta contribui para a padronização e otimização de operações, especialmente em tarefas repetitivas. Além disso, observa-se uma redução significativa de erros humanos e um aumento na velocidade de execução dessas atividades.

2.3. DIAGNÓSTICO AUTOMATIZADO POR MEIO DE AGENTES CONVERSACIONAIS

Em contextos de suporte técnico com alto volume de chamados, a automação surge como uma alternativa para reduzir a execução de tarefas repetitivas. No entanto, para que essa automação seja efetivamente utilizada, é necessário que exista uma interface acessível ao usuário final. Nesse sentido, os *chatbots* se apresentam como uma solução adequada, ao permitir a interação direta entre o usuário e os processos automatizados. Ou seja, para que essa automação atue como uma alternativa completa na resolução de chamados, é necessário que o sistema disponha de uma interface capaz de interagir com o usuário de forma eficiente e com certo nível de inteligência conversacional (Brandtzaeg & Følstad, 2018). No escopo deste trabalho, o *chatbot* atua como um agente inteligente,

responsável por ler e interpretar informações relevantes para o chamado, e retornar uma resposta adequada.

Esse tipo de aplicação exige que o *chatbot* seja estruturado como um agente especialista. Sistemas especialistas são projetados para reproduzir o processo de resolução de problemas, realizado por especialistas humanos, replicando as etapas de coleta de informações e tomada de decisão (Turner-Fairbank, 1988). Dessa forma, o agente pode atuar como um especialista virtual, desde que seja adequadamente estruturado com base nas regras, procedimentos e conhecimentos específicos do processo de diagnóstico em que está inserido.

O funcionamento de um agente especialista depende de uma base de conhecimento bem estruturada. É nessa base que estão organizados os fatos e regras que permitem ao sistema interpretar informações, realizar inferências e tomar decisões de forma semelhante a um especialista humano (Turner-Fairbank, 1988). É importante, contudo, diferenciá-la do fluxo de trabalho. Enquanto o fluxo de trabalho define a sequência de etapas a serem executadas, a base de conhecimento orienta quais etapas devem ser seguidas em cada situação, a partir das informações disponíveis. A integração desses dois elementos viabiliza a automação de tarefas mais complexas e reduz a dependência de conhecimento concentrado em especialistas.

Lucas & Van Der Gaag (1991) descrevem princípios fundamentais para o desenvolvimento de sistemas especialistas. Segundo os autores, esses sistemas devem incorporar o conhecimento de especialistas humanos, com o objetivo de auxiliar usuários não especialistas. Para isso, é necessário seguir um processo estruturado, que inclui: a definição dos tipos de problemas a serem resolvidos e das fontes de informação disponíveis; a construção de um modelo conceitual com uma forma adequada de representação do conhecimento; a distinção entre conhecimento e mecanismos de inferência; e o desenvolvimento de uma interface adequada ao usuário. Após essas etapas, o sistema deve ser testado e validado.

A literatura apresenta aplicações semelhantes de sistemas especialistas voltados à automação de diagnósticos. Um exemplo é o sistema GRENDEL, desenvolvido para identificar inconsistências em modelos computacionais biológicos, sendo capaz de apontar as etapas mais prováveis de conter erros em simulações (Chen & Altman, 1999). Em outra linha de pesquisa, Livingston (2001) propôs o sistema HAMB, voltado à automação de descobertas científicas. Nesse caso, o sistema utiliza heurísticas para priorizar hipóteses com maior plausibilidade, orientando a exploração e contribuindo para

a geração de resultados relevantes no domínio da cristalização de macromoléculas. Esses exemplos demonstram como bases de conhecimento e mecanismos de inferência podem ser utilizados para automatizar tarefas que, tradicionalmente, exigiriam conhecimento especializado.

3. PROCEDIMENTOS METODOLÓGICOS

Esta seção apresenta o percurso metodológico adotado para o desenvolvimento deste trabalho. A pesquisa foi conduzida ao longo de um ano, sendo o primeiro semestre dedicado ao planejamento inicial e o segundo às etapas de implementação e validação. Nas subseções a seguir, são descritos o contexto do estudo, a classificação da pesquisa e o detalhamento das etapas de desenvolvimento do projeto.

3.1. CLASSIFICAÇÃO DA PESQUISA

Quanto à sua natureza, essa pesquisa é classificada como aplicada, pois tem como objetivo gerar conhecimento voltado à solução de um problema prático específico, relacionado à otimização do suporte técnico em uma empresa real. Em relação à abordagem, trata-se de uma pesquisa de métodos mistos (qualitativa e quantitativa). A dimensão qualitativa é empregada no mapeamento dos processos, por meio de entrevistas e análise documental. Já a dimensão quantitativa é utilizada na etapa de validação, com foco na mensuração de indicadores como acurácia e eficiência. Quanto aos objetivos, a pesquisa possui caráter exploratório, uma vez que a aplicação de agentes especialistas para diagnóstico automatizado em sistemas APS altamente customizados ainda é pouco explorada na literatura. No que se refere aos procedimentos técnicos, o estudo envolve pesquisa bibliográfica, para fundamentação teórica; pesquisa documental, por meio da análise de chamados de suporte; e pesquisa-ação, considerando o envolvimento direto da pesquisadora no desenvolvimento e na implementação da solução no contexto organizacional (Livingston, 2001).

3.2. ETAPAS DO TRABALHO

Para atingir os objetivos propostos, o método adotado está estruturado em seis etapas sequenciais, apresentadas na Figura 2. Tais etapas abrangem desde a definição inicial dos requisitos até a validação do protótipo, sendo detalhadas nas subseções a seguir.

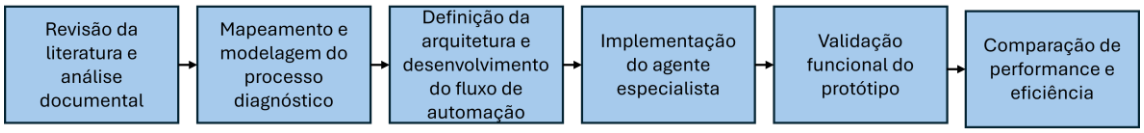


Figura 2 - Etapas do projeto. Imagem criada pela autora.

3.2.1. REVISÃO DA LITERATURA E ANÁLISE DOCUMENTAL

Inicialmente, deve-se realizar o levantamento dos requisitos funcionais e técnicos da solução. Para isso, é necessário conduzir uma revisão da literatura sobre o papel de sistemas de APS no contexto industrial, e sobre sua estrutura e capacidade de personalização. Também deve-se estudar sobre métodos para mapeamento e automação de processos. Por fim, deve-se incluir na pesquisa, o uso de agentes inteligentes para automação de tarefas repetitivas. Essa etapa inicial de pesquisa tem como objetivo situar o trabalho, contextualizar o sistema em questão, e identificar boas práticas para o seu suporte e melhoria contínua, incluindo limitações atuais, e comparações com modelos de automação aplicados em outras áreas.

Paralelamente, deve-se analisar um conjunto representativo de chamados de suporte previamente atendidos pela organização, observando o tipo de problema reportado, as etapas de diagnóstico realizadas e a frequência de ocorrência. A partir dessa análise, devem ser identificados padrões recorrentes, gargalos no processo de atendimento e oportunidades de automação. Com base nesses elementos, deve-se elaborar uma lista estruturada de requisitos, incluindo funcionalidades esperadas da solução, fontes de dados necessárias e restrições técnicas do ambiente em que será implementada.

3.2.2. MAPEAMENTO E MODELAGEM DO PROCESSO DE DIAGNÓSTICO

Na sequência, deve-se proceder ao mapeamento do processo de diagnóstico de chamados de suporte. Para isso, devem ser realizadas entrevistas semiestruturadas com consultores da equipe de *Customer Success*, de modo a capturar o conhecimento tácito envolvido no diagnóstico dos problemas. As entrevistas devem buscar identificar, de forma detalhada, quais etapas são seguidas durante a análise de um chamado, quais critérios são utilizados para validação das informações, quais verificações são consideradas essenciais e como são tomadas as decisões ao longo do processo. Com base nesses dados, deve-se consolidar o fluxo de diagnóstico em uma sequência lógica de atividades e representá-lo por meio de um modelo BPMN, no qual devem estar

explicitados os pontos de decisão, os caminhos alternativos e as interações com sistemas externos.

3.2.3. DEFINIÇÃO DA ARQUITETURA E DESENVOLVIMENTO DO FLUXO DE AUTOMAÇÃO

Com o processo mapeado, deve-se desenvolver o fluxo de automação. Para isso, é necessário primeiramente traduzir o modelo BPMN em um fluxo lógico para execução dentro de uma plataforma de automação, como o N8N. Inicialmente, deve-se definir o gatilho de execução do fluxo, por exemplo, a criação de um novo chamado na plataforma de *helpdesk*. Em seguida, devem ser configurados os nós responsáveis pela coleta das informações relevantes, incluindo o conteúdo do chamado e dados provenientes de bases externas. Posteriormente, deve-se estruturar as etapas de processamento, nas quais serão aplicadas regras de decisão e verificações equivalentes às realizadas pelos consultores. Por fim, deve-se definir os nós responsáveis pela geração e envio da resposta, garantindo que o fluxo reproduza, de forma fiel, a lógica de diagnóstico previamente mapeada.

Após a definição do fluxo de automação, deve-se definir a arquitetura da solução. Essa etapa consiste em identificar os componentes envolvidos, estabelecer suas responsabilidades e determinar como ocorrerá a integração entre eles. Nesse contexto, a plataforma de *helpdesk* deve ser considerada como o ponto de entrada das solicitações, sendo responsável por registrar e disponibilizar os chamados, além de armazenar as bases de conhecimento. A plataforma de automação, por sua vez, deve ser responsável por orquestrar o processamento dessas solicitações, enquanto o agente de Inteligência Artificial deve realizar a análise das informações e a geração das respostas. A partir dessa definição, devem ser especificados os mecanismos de comunicação entre os componentes, incluindo o uso de APIs e *webhooks* para troca de dados. Também devem ser definidos os formatos de envio e processamento das informações, como estruturas em JSON.

Por fim, com o fluxo e arquitetura definidos para todo o processo, desde o recebimento do chamado na plataforma de *helpdesk* até a geração e disponibilização da resposta processada, deve ser construído o fluxograma de automação na plataforma N8N, com a configuração de *webhooks* e APIs de entrada e saída, bem como os tratamentos intermediários de dados.

3.2.4. IMPLEMENTAÇÃO DO AGENTE ESPECIALISTA

Em seguida, deve-se implementar o agente conversacional responsável pela interpretação das solicitações e geração das respostas. Para isso, é necessário definir uma estrutura de *prompt* que oriente o agente na análise das informações recebidas e na organização lógica da resposta. Esse *prompt* deve conter instruções claras sobre como interpretar o conteúdo do chamado, como selecionar informações relevantes das bases de conhecimento e como estruturar a resposta final, incluindo possíveis causas do problema, sugestões de verificação e orientações adicionais. O agente deve ser integrado ao fluxo de automação, recebendo como entrada os dados coletados nas etapas anteriores e retornando uma resposta estruturada que será utilizada no atendimento.

3.2.5. VALIDAÇÃO FUNCIONAL DO PROTÓTIPO

Nesta etapa, deve-se realizar a validação funcional do protótipo, com o objetivo de verificar se a solução é capaz de executar corretamente as tarefas para as quais foi projetada. Para isso, deve-se selecionar um conjunto de chamados que represente diferentes tipos de problemas encontrados no contexto de suporte ao sistema APS. Esse conjunto deve incluir tanto chamados reais previamente resolvidos, que permitam a comparação com diagnósticos conhecidos, quanto casos simulados, elaborados para testar cenários específicos e limites da solução. Para cada chamado selecionado, deve-se executar o fluxo automatizado completo, desde a entrada da solicitação até a geração da resposta pelo agente. Durante essa execução, devem ser registrados os resultados obtidos, incluindo o diagnóstico proposto, as hipóteses levantadas e as recomendações de verificação apresentadas pelo sistema.

Por exemplo, um chamado contendo a solicitação “a ordem de produção não está sendo sequenciada na máquina esperada” pode ser utilizado como entrada do sistema. A partir dessa demanda, o fluxo automatizado deve analisar as regras de capacidade, recursos válidos, restrições de modelagem e parâmetros associados à ordem. Como saída, o agente pode retornar hipóteses diagnósticas, como ausência de interseção entre recursos válidos, restrições secundárias de capacidade ou priorizações definidas na lógica de sequenciamento, além de sugerir verificações específicas para confirmação da causa identificada.

A avaliação da validação funcional deve considerar, primeiramente, a acurácia das respostas geradas. Para isso, deve-se verificar se o diagnóstico proposto pelo sistema está

correto ou se contempla adequadamente as possíveis causas do problema. Adicionalmente, deve-se avaliar a capacidade do sistema de identificar corretamente os casos que não podem ser resolvidos automaticamente, realizando o escalonamento adequado para um consultor humano. Essa etapa deve permitir a identificação de limitações do protótipo, bem como de possíveis ajustes necessários nas regras de decisão, nas bases de conhecimento ou na estrutura do agente.

3.2.6. COMPARAÇÃO DE PERFOMANCE E EFICIÊNCIA

Após a validação funcional, deve-se realizar a análise de performance e eficiência do protótipo, com o objetivo de comparar seu desempenho em relação ao atendimento realizado por consultores humanos. Para isso, devem ser definidos indicadores quantitativos que permitam mensurar o desempenho da solução. Entre esses indicadores, deve-se considerar o tempo de geração da primeira resposta (*Mean Time for First Response*) e, sempre que aplicável, o tempo total de resolução do chamado (*Mean Time for Resolution*). A coleta desses dados deve ser realizada a partir da execução dos mesmos chamados utilizados na validação funcional, garantindo a comparabilidade entre os resultados do protótipo e os registros de atendimento humano. No caso do atendimento humano, devem ser utilizados dados históricos registrados na plataforma de *helpdesk*.

Além da análise quantitativa, deve-se realizar uma avaliação qualitativa das respostas geradas pelo *chatbot*. Essa avaliação deve considerar critérios como clareza, completude, coerência lógica e relevância das informações, com o objetivo de verificar sua adequação ao contexto do chamado e sua utilidade para o diagnóstico. Os resultados qualitativos devem ser analisados em conjunto com os indicadores de desempenho, permitindo uma avaliação integrada da eficiência e da efetividade da solução. Por fim, deve-se interpretar os resultados, considerando tanto os avanços quanto as limitações do protótipo no contexto analisado.

4. RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos ao longo do desenvolvimento do protótipo, organizados de acordo com as etapas metodológicas previamente definidas. Para cada etapa, são descritos os artefatos produzidos, as decisões tomadas e suas implicações para a solução proposta, articulando esses achados com a literatura revisada, de modo a analisar em que medida confirmam, complementam ou divergem das contribuições teóricas.

O estudo foi desenvolvido na NEO Digital Industries, uma empresa de consultoria especializada na implementação de soluções de *software* para gestão da cadeia de suprimentos no setor industrial. O trabalho tem como foco o sistema Opcenter APS da Siemens, reconhecido como líder de mercado em sequenciamento fino, com capacidade finita em ambientes de manufatura complexos.

A análise concentra-se no processo de *Customer Success* relacionado ao Opcenter APS. Devido ao alto nível de customização do sistema, sua implementação exige adaptações específicas para cada cliente, tornando o suporte técnico uma atividade contínua e essencial. Atualmente, esse suporte é conduzido por consultores especializados, que dedicam uma parcela significativa de seu tempo à análise de chamados diagnósticos, frequentemente associados a tarefas repetitivas de investigação de problemas. Esse cenário configura o problema central deste estudo, que busca a automação dessas atividades.

O objetivo do projeto é simplificar as rotinas de diagnóstico dos consultores da NEO. Para isso, propõe-se uma solução baseada na integração da plataforma de automação N8N com um agente de Inteligência Artificial especializado. Esse agente tem acesso a bases de conhecimento relacionadas ao modelo de cada cliente, ao funcionamento geral do sistema Opcenter APS e a uma base de conhecimento tácito dos consultores. Sua atuação pode ser caracterizada como a de um agente especialista, uma vez que combina essas bases de conhecimento com instruções estruturadas que orientam a organização lógica das análises, de forma semelhante ao raciocínio de um consultor.

4.1. REVISÃO DA LITERATURA E ANÁLISE DOCUMENTAL

A definição dos requisitos funcionais e técnicos da solução foi conduzida a partir da revisão da literatura, estruturada em três eixos principais, conforme apresentado no referencial teórico.

Inicialmente, o problema de pesquisa foi contextualizado a partir da caracterização do ambiente industrial contemporâneo. De acordo com Simchi-Levi et al. (2000), a elevada competitividade exige decisões cada vez mais eficientes e responsivas. Nesse cenário, sistemas APS se destacam como ferramentas essenciais para o planejamento da produção, conforme apontado por Ivert & Jonsson (2010). A partir das definições de Mauergauz (2016), foi possível delimitar o escopo desses sistemas, compreendendo sua capacidade de modelar ambientes produtivos complexos e operar com restrições reais.

Com base nesse contexto, a literatura indica que sistemas no modelo SaaS demandam manutenção e evolução contínuas, frequentemente associadas às especificidades de cada cliente (Camargo, 2019). No entanto, Bennett & Rajlich (2000) destacam que esse processo implica desafios significativos para as equipes de suporte, especialmente em função da necessidade de lidar com diferentes configurações e lógicas de sistema nas modelagens de cada cliente. A partir dessas contribuições, foi possível estabelecer como requisito a necessidade de estruturar o conhecimento associado a cada modelo, de forma a facilitar sua compreensão e reduzir a dependência de especialistas específicos.

Em um segundo momento, a literatura sobre mapeamento e automação de processos contribuiu para a definição da abordagem técnica da solução. Deufemia (2007) destaca a importância da modelagem prévia de processos para o desenvolvimento de agentes orientados a serviços, sendo a notação BPMN adequada para representar fluxos de execução e seus pontos de decisão. Complementarmente, Hirzel (2023) aponta que plataformas *low-code* permitem que especialistas automatizem processos sem a necessidade de desenvolvimento intensivo. Nesse mesmo sentido, McFeetors e Pant (2022) descrevem a estrutura baseada em fluxos visuais como um meio eficaz de representar e implementar lógicas de processo. A partir dessas contribuições, definiu-se que o processo de diagnóstico deveria ser inicialmente formalizado por meio de um modelo BPMN e, posteriormente, implementado em uma plataforma de automação *low-code*, permitindo a tradução direta da lógica de análise em um fluxo executável.

Por fim, a literatura sobre agentes conversacionais e sistemas especialistas orientou a definição dos componentes inteligentes da solução. Brandtzaeg & Følstad (2018) destacam a importância de interfaces conversacionais para interação com o usuário, enquanto Turner-Fairbank (1988) enfatiza o papel das bases de conhecimento na construção de sistemas especialistas. Adicionalmente, Lucas & Van der Gaag (1991) indicam que esses sistemas devem ser capazes de realizar inferências a partir de conhecimento estruturado. Com base nesses fundamentos, foram definidos como elementos essenciais da solução: a utilização de um agente conversacional para interface com o usuário, a construção de bases de conhecimento distintas para representar diferentes níveis de informação e a implementação de um mecanismo de inferência orientado por *prompt*, capaz de estruturar o raciocínio diagnóstico.

Além da revisão teórica, foi realizada uma análise documental abrangente da base histórica de chamados de suporte da empresa, com o objetivo de compreender

empiricamente a natureza das demandas atendidas pelo time de *Customer Success*. Foram analisados 2.856 chamados registrados desde a implantação da plataforma de *helpdesk*, os quais foram classificados de acordo com seu desfecho: chamados que resultaram em alterações no sistema e chamados resolvidos sem necessidade de modificações.

Os resultados indicam que 1.203 chamados, correspondentes a 42,1% do total, foram resolvidos sem qualquer alteração no sistema, caracterizando-se como atividades puramente diagnósticas, orientativas ou relacionadas a fatores externos, como dados incorretos, parametrizações inadequadas ou integrações com outros sistemas. Complementarmente, os outros 57,9% dos chamados resultaram em algum tipo de modificação, variando desde ajustes simples de modelagem até implementações mais complexas. Também foi realizada uma análise específica por categoria dos chamados, a qual reforça esse padrão observado. Chamados classificados pelo cliente como "Problema" e "Problema Emergencial" apresentaram aproximadamente metade dos casos resolvidos sem alterações no sistema, evidenciando que uma parcela significativa das falhas percebidas pelos usuários não está associada a erros no software, mas sim a questões de uso, dados ou configuração. Mesmo em categorias como "Solicitação" e "Melhoria", observa-se que uma fração relevante dos chamados não evolui para desenvolvimento, sendo encerrada após análise ou redefinição de escopo.

Esses achados empíricos dialogam diretamente com a literatura. Bennett e Rajlich (2000) destacam o alto custo da manutenção de software, e os dados observados indicam que uma parcela expressiva do esforço da equipe de suporte não está necessariamente relacionada à evolução do sistema, mas sim à execução de diagnósticos repetitivos. Da mesma forma, a necessidade de conhecimento especializado, apontada por Camargo (2019), se confirma na prática, uma vez que mesmo chamados sem necessidade de alteração demandam análise detalhada do modelo do cliente.

A partir da integração entre literatura e análise documental, consolida-se uma conclusão central para o projeto: existe um volume significativo de chamados que seguem padrões diagnósticos repetitivos e que não exigem intervenção no sistema, mas que consomem tempo relevante de consultores especializados. Esse cenário evidencia uma oportunidade clara para automação, especialmente na etapa inicial de diagnóstico, na qual regras, verificações e inferências podem ser estruturadas e replicadas por um agente inteligente. Essa constatação fundamenta diretamente a proposta deste trabalho, ao justificar o desenvolvimento de uma solução capaz de sistematizar o conhecimento dos

consultores e automatizar parte do processo de análise de chamados, reduzindo o tempo de atendimento e liberando recursos humanos para atividades de maior valor agregado.

4.2. MAPEAMENTO E MODELAGEM DO PROCESSO DE DIAGNÓSTICO

O mapeamento do processo de diagnóstico teve como objetivo compreender, de forma estruturada, como os consultores realizam a análise de chamados de suporte no sistema Opcenter APS. Para isso, foram conduzidas entrevistas semiestruturadas com consultoras da equipe de Customer Success da NEO Digital Industries, complementadas pela análise de chamados reais. Foram realizadas três entrevistas, a primeira com a gerente da área de Customer Success, que tem 5 anos de experiência em atendimento de suporte de Opcenter APS, e as outras duas com consultoras gestoras de contas de suporte, uma delas com 3 anos de experiência e 6 contas, e a outra com 2 anos de experiência e 7 contas. Essa etapa buscou identificar os critérios de validação das informações, os principais pontos de verificação adotados durante o diagnóstico, as regras utilizadas na tomada de decisão, as relações entre componentes do sistema e as situações que exigem consultas adicionais para avanço da análise.

A análise dos dados coletados evidenciou que o processo de diagnóstico é composto por um conjunto recorrente de verificações e decisões. Observou-se que os consultores seguem critérios relativamente consistentes para validar informações, priorizam determinados elementos do sistema durante a investigação e utilizam regras práticas para orientar suas conclusões. Também se identificou que o diagnóstico frequentemente depende da compreensão das relações entre diferentes partes do sistema e, em diversos casos, exige a realização de consultas adicionais para obtenção de informações não disponíveis diretamente no chamado.

A consolidação dessas informações permitiu observar que, mesmo com variações individuais na abordagem inicial, o raciocínio adotado pelos consultores segue um padrão relativamente consistente. Em geral, o diagnóstico ocorre por meio de uma sequência ordenada de verificações, na qual o resultado de cada etapa direciona o próximo passo da análise. Esse comportamento caracteriza um processo investigativo estruturado, baseado na aplicação sequencial de regras e validações. A partir dessa compreensão, foi possível construir um modelo BPMN representando o processo de diagnóstico (Apêndice A). O modelo desenvolvido formaliza o fluxo de atendimento de chamados conforme executado na prática, caracterizando-se como um modelo *As-Is* do suporte técnico. Esse artefato

torna explícitos os pontos de decisão e os caminhos possíveis dentro do processo, servindo como base para a etapa de automação.

Os resultados obtidos dialogam diretamente com a literatura. Deufemia (2007) propõe que processos orientados a serviços podem ser modelados de forma a explicitar decisões e fluxos de informação, o que se confirma neste estudo, ao se observar que o diagnóstico pode ser representado como um fluxo estruturado. Da mesma forma, a possibilidade de externalização do conhecimento tácito, discutida por Lucas e Van der Gaag (1991), é evidenciada na medida em que o raciocínio dos consultores pôde ser capturado, organizado e convertido em um modelo replicável. Esses achados reforçam a viabilidade de utilizar esse tipo de estrutura como base para o desenvolvimento de um agente automatizado.

4.3. DEFINIÇÃO DA ARQUITETURA E DESENVOLVIMENTO DO FLUXO DE AUTOMAÇÃO

A partir do processo de diagnóstico previamente mapeado, foi definida a arquitetura da solução com o objetivo de viabilizar a automação da etapa inicial de análise de chamados de suporte. Essa definição partiu da necessidade de traduzir o fluxo conceitual identificado em um sistema funcional capaz de reproduzir, de forma estruturada, as etapas realizadas pelos consultores. Inicialmente, foram definidos os principais componentes da solução. A plataforma de *helpdesk* foi estabelecida como ponto de entrada das solicitações, sendo responsável por armazenar os chamados e concentrar a interação com o usuário. A plataforma de automação N8N foi definida como o núcleo responsável pela orquestração do processo, executando o fluxo de diagnóstico e gerenciando o trânsito de informações entre os sistemas. Por fim, o agente de Inteligência Artificial foi incorporado como componente responsável por interpretar os dados disponíveis e estruturar a resposta diagnóstica com base nas regras e conhecimentos previamente definidos.

Com os componentes definidos, foi estruturado o fluxo de dados entre eles, que está representado na figura 3. O processo é iniciado a partir da criação de um novo chamado na plataforma de *helpdesk*, que aciona, por meio de um gatilho nativo da plataforma, um *webhook* no N8N. A partir desse gatilho, o fluxo de automação realiza a coleta do conteúdo do chamado e das bases de conhecimento necessárias para o diagnóstico: (i) a base específica do modelo do cliente, (ii) a base de conhecimento geral sobre o sistema Opcenter APS e (iii) a base contendo a lógica de análise dos consultores.

Essas informações são então consolidadas e preparadas para processamento pelo agente. Na sequência, o conteúdo coletado é inserido em um *prompt* estruturado (Apêndice B), desenvolvido para orientar o agente na execução do diagnóstico. Esse prompt organiza o raciocínio esperado, induzindo o agente a seguir uma sequência lógica de análise semelhante à adotada pelos consultores.

A partir desse conjunto de entradas, o agente processa as informações e gera uma resposta estruturada, contendo possíveis causas para o problema relatado, sugestões de verificações adicionais e orientações para continuidade da análise. O resultado gerado é então retornado ao N8N, que realiza a publicação da resposta na plataforma de *helpdesk*. Optou-se por registrar essa resposta inicialmente como comentário privado, direcionado ao consultor responsável pelo chamado. Essa decisão foi tomada para garantir uma etapa de validação humana antes da comunicação com o cliente, reduzindo o risco de respostas imprecisas e posicionando o sistema como ferramenta de apoio, e não como substituto integral do atendimento.

O desenvolvimento do fluxo na Figura 3 permitiu a construção de um fluxo de automação funcional no N8N, estruturado em nós, que representam cada etapa do processo de diagnóstico, incluindo coleta de dados, tratamento de informações, execução do agente e envio da resposta. Diferentemente de uma automação baseada em regras fixas, o fluxo desenvolvido incorpora um mecanismo de inferência, no qual o agente define os caminhos de análise com base nas informações disponíveis, aproximando o comportamento do sistema ao raciocínio humano observado na etapa de mapeamento.

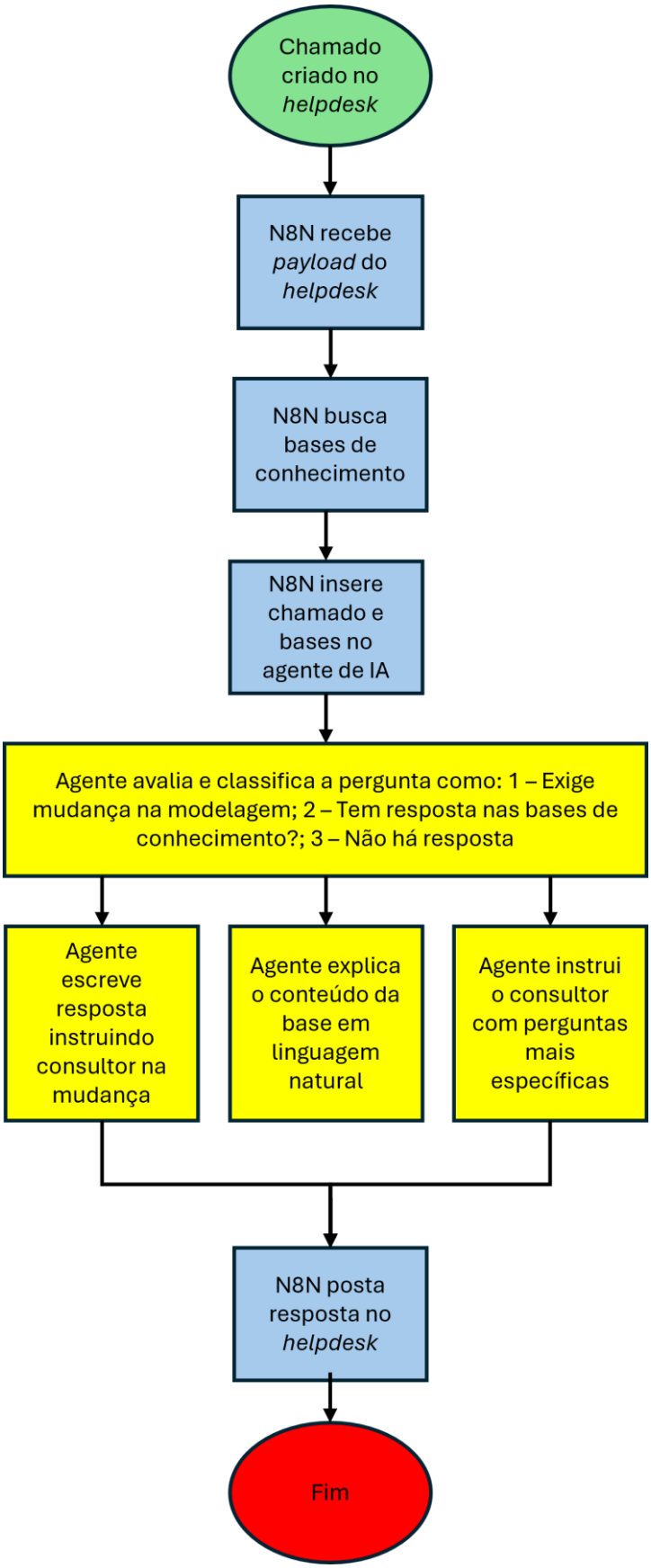


Figura 3 - Modelo fluxo de automação. Imagem criada pela autora.

O fluxo de automação desenvolvido, evidenciado na Figura 4, é composto por onze nós interconectados, organizados de forma sequencial, para viabilizar o processamento completo de um chamado de suporte. O ponto de entrada do fluxo de automação é um nó do tipo *webhook* (WEBHOOK /letinha/chat), configurado para receber requisições HTTP via método POST. Esse *endpoint* é acionado pela plataforma de *helpdesk* sempre que um novo chamado é criado, enviando um *payload* em formato JSON, contendo, principalmente, o identificador do chamado e a mensagem do usuário. Essas informações passam a ser utilizadas como base para todas as etapas subsequentes do fluxo. Após a entrada dos dados, o fluxo executa três consultas independentes a bancos de dados, por meio de nós do tipo PostgreSQL, cada um responsável por acessar uma das bases de conhecimento utilizadas no projeto: a base específica do cliente, a base geral do sistema Opcenter APS e a base de raciocínio dos consultores. Essas consultas utilizam estruturas do tipo *Common Table Expression* (CTE) recursiva para percorrer hierarquicamente as categorias e subcategorias das bases, a partir de um identificador raiz previamente definido.

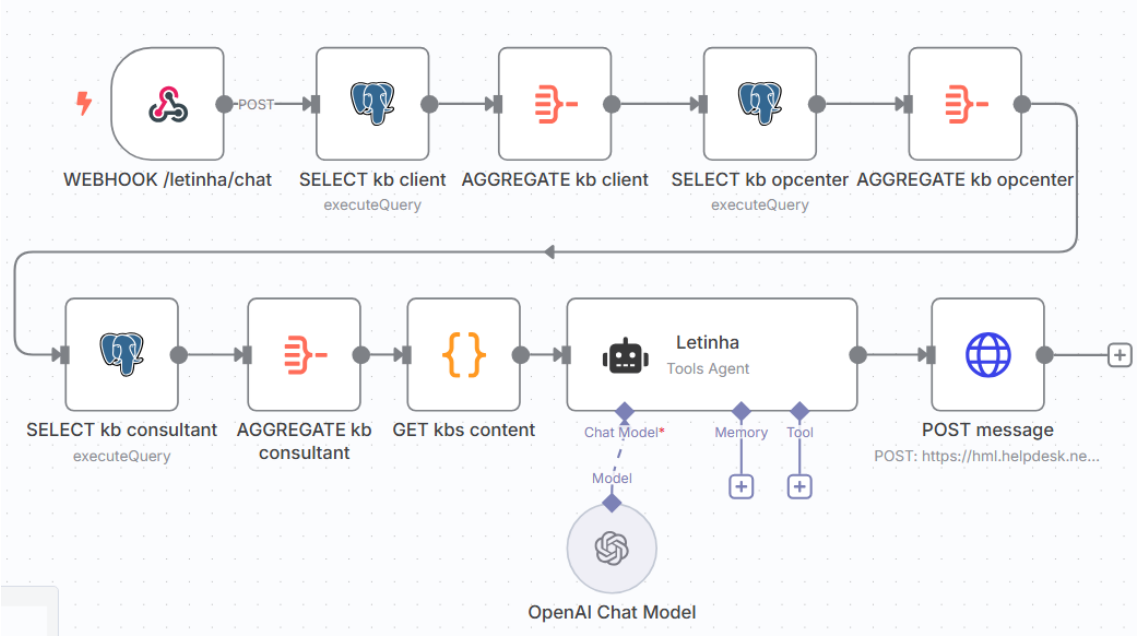


Figura 4 - Fluxo para automação de chamados diagnósticos. Imagem criada pela autora.

Como resultado, cada consulta retorna um conjunto de registros contendo categorias, títulos de artigos e seus respectivos conteúdos textuais. Em seguida, cada conjunto de resultados é processado por nós do tipo Aggregate, cuja função é consolidar múltiplas linhas em um único objeto estruturado. Essa etapa é necessária para permitir

que grandes volumes de conteúdo sejam tratados de forma unificada na fase seguinte, evitando fragmentação da informação. Os três conjuntos consolidados convergem então para um nó do tipo Code (GET kbs content), responsável por realizar o tratamento final dos dados. Nesse nó, um script em JavaScript organiza os conteúdos por categoria e os formata em uma estrutura textual em *Markdown*, utilizando níveis hierárquicos de títulos para representar a organização das bases. Como saída, é gerado um único objeto contendo três blocos de texto estruturados, prontos para serem utilizados como contexto no processo de inferência.

O núcleo do fluxo está no nó Letinha, configurado como um agente de Inteligência Artificial baseado em *LangChain* e conectado a um modelo de linguagem da OpenAI (GPT-5). Esse agente recebe como entrada tanto a mensagem do usuário quanto o conteúdo estruturado das bases de conhecimento, que são injetados dinamicamente no prompt gerado para replicar as inferências que um consultor realiza, ao atender um chamado. Esse *prompt* estabelece o papel do agente como um consultor especializado em triagem de chamados de Opcenter APS, define regras de análise e classificação da demanda e especifica o formato esperado da resposta, que deve ser redigida em português, com linguagem técnica e fundamentada exclusivamente nas informações fornecidas.

Por fim, a resposta gerada pelo agente é encaminhada para um nó do tipo HTTP Request (POST message), responsável por enviar uma requisição autenticada à API da plataforma de *helpdesk*. Essa requisição publica o conteúdo gerado como uma nota interna no chamado correspondente, garantindo que a resposta seja primeiramente avaliada por um consultor humano, antes de qualquer interação direta com o cliente.

Os resultados obtidos nessa etapa estão alinhados com os conceitos discutidos na literatura. Conforme apontado por McFeetors e Pant (2022), plataformas baseadas em fluxos visuais permitem representar e implementar lógicas de processo de forma clara, o que se reflete diretamente na estrutura construída no N8N. De forma complementar, a utilização de uma plataforma *low-code*, conforme sugerido por Hirzel (2023), mostrou-se adequada para desenvolver a solução sem necessidade de programação extensa.

Adicionalmente, a incorporação de mecanismos de inferência ao fluxo de automação demonstrou-se consistente com os princípios de sistemas especialistas. A integração do agente de Inteligência Artificial como componente central do processo permitiu estruturar a análise dos chamados de forma flexível, mantendo a capacidade de adaptação a diferentes contextos, conforme discutido por Turner-Fairbank (1988). A solução desenvolvida evidencia, na prática, a viabilidade de integrar fluxos estruturados

com mecanismos de inferência, constituindo um alicerce técnico robusto para a automação do diagnóstico de chamados em ambientes complexos.

4.4. IMPLEMENTAÇÃO DO AGENTE ESPECIALISTA

A etapa de implementação do agente especialista teve como objetivo estruturar a interface conversacional responsável por interpretar o conteúdo dos chamados e gerar respostas técnicas que auxiliem o consultor no processo de diagnóstico. Diferentemente de uma automação puramente determinística, essa etapa introduz o componente de inferência, permitindo que o sistema lide com a variabilidade e ambiguidade características dos chamados de suporte. O agente foi implementado como um componente integrado ao fluxo de automação desenvolvido, sendo acionado por meio do nó Letinha do fluxo de automação. Como entradas, o agente recebe tanto o conteúdo do chamado, obtido a partir do *webhook* da plataforma de *helpdesk*, quanto os dados provenientes das três bases de conhecimento previamente estruturadas: a base específica do cliente, a base geral do sistema Opcenter APS e a base de raciocínio dos consultores. Esses elementos são organizados e injetados dinamicamente em um prompt estruturado, responsável por orientar o comportamento do agente.

A construção desse prompt seguiu uma abordagem iterativa de engenharia de *prompt*, na qual diferentes versões foram testadas e refinadas até que o agente apresentasse respostas consistentes e alinhadas com a lógica de diagnóstico utilizada pelos consultores. O *prompt*, disponibilizado no Apêndice B, define explicitamente o papel do agente como um especialista em atendimento inicial de chamados de Opcenter APS, estabelecendo diretrizes para interpretação do problema, seleção das informações relevantes nas bases de conhecimento e organização da resposta. Além disso, o *prompt* estrutura o processo de inferência, conduzindo o agente a identificar possíveis causas, sugerir verificações adicionais e indicar testes que possam ser realizados para confirmação do diagnóstico. Como resultado, o agente gera respostas estruturadas que não se limitam a fornecer uma única solução, mas apresentam um conjunto de hipóteses e orientações que refletem o raciocínio investigativo típico de um consultor humano. Esse comportamento está alinhado com os princípios de sistemas especialistas descritos por Lucas e Van der Gaag (1991), nos quais a tomada de decisão ocorre a partir da combinação entre base de conhecimento e mecanismo de inferência.

Para ilustrar o funcionamento do agente, um dos casos utilizados durante os testes consistiu no questionamento sobre o motivo pelo qual uma ordem de corte e vinco não

foi sequenciada antes de outra com data de entrega posterior, apesar de ambas possuírem o mesmo *setup*. A partir dessa entrada, o agente analisou elementos como horizonte de otimização, regras de agrupamento, Data de Início Mais Cedo (DIMC), restrições secundárias e parâmetros da nSBR, regra de sequenciamento desenvolvida pela NEO, retornando hipóteses diagnósticas estruturadas e sugestões de verificação no sistema.

Em outro exemplo, o agente recebeu uma solicitação relacionada ao desbalanceamento da programação entre recursos de *Hot Stamping*, setor que realiza marcações metálicas nas embalagens produzidas, identificando possíveis causas associadas à validação de recursos, taxas de produção, calendários, regras de sequenciamento e restrições secundárias. Como resposta, o sistema apresentou um checklist estruturado de diagnóstico e sugestões de ações corretivas para investigação do cenário.

É importante destacar que, diferentemente de *chatbots* tradicionais voltados ao atendimento direto ao cliente, a resposta gerada pelo agente não é publicada automaticamente. O conteúdo produzido é inserido como uma nota interna no chamado, sendo previamente analisado por um consultor antes de qualquer comunicação externa. Dessa forma, o agente atua como uma ferramenta de apoio ao diagnóstico, e não como um substituto do atendimento humano. Os resultados obtidos nesta etapa convergem com a literatura ao demonstrar que a utilização de bases de conhecimento estruturadas, aliadas a mecanismos de inferência, permite replicar parcialmente o raciocínio de especialistas em contextos específicos. Ao mesmo tempo, a decisão de manter o consultor no ciclo de validação evidencia uma adaptação prática ao contexto real de aplicação, garantindo maior confiabilidade das respostas e reduzindo riscos associados à automação completa do atendimento.

Assim, a implementação do agente especialista configura um componente central da solução proposta, ao viabilizar a transformação do conhecimento estruturado em respostas acionáveis, contribuindo diretamente para a redução do esforço cognitivo dos consultores e para a melhoria do processo de diagnóstico de chamados.

4.5. VALIDAÇÃO FUNCIONAL DO PROTÓTIPO

A validação funcional do protótipo foi conduzida com base na execução de pré-testes, conforme orientações de Marconi e Lakatos (2003), com o objetivo de verificar a capacidade do sistema em reproduzir, de forma consistente, o processo de diagnóstico modelado. Para isso, foi definida uma amostra composta por 14 chamados, sendo 7

provenientes do histórico real da operação e 7 construídos como casos de teste, permitindo avaliar tanto a aderência a situações reais quanto o comportamento do sistema em cenários controlados. Cada chamado foi submetido ao fluxo automatizado completo, desde a recepção via *webhook*, até a publicação da resposta final na plataforma de atendimento. Os resultados obtidos foram registrados e classificados de acordo com critérios de acurácia e eficiência previamente definidos. A síntese desses resultados encontra-se apresentada na Figura 5.

DASHBOARD — VALIDAÇÃO DO PROTÓTIPO			
CONTAGEM DE TICKETS			
Total de tickets registrados		14	
Real Histórico		7	
Novo de Teste		7	
ACURÁCIA DO PROTÓTIPO (seção 3.3.5)			
Diagnósticos corretos (Sim)		10	
Diagnósticos incorretos (Não)		4	
Escalonamentos corretos (Sim)		10	
Escalonamentos incorretos (Não)		4	
SUCCESS DIAGNOSTIC RATE	Diagnósticos corretos ÷ total	71%	
CORRECT SCALING RATE	Escalonamentos corretos ÷ total	71%	
EFICIÊNCIA — BOT vs CONSULTOR (seções 3.3.5 e 3.3.6)			
Métrica	**Bot — média (s)**	**Consultor Baseline — média (s)**	**Redução (%)**
MTFR — Mean Time for First Response	73	32.100	99,7726%
MTR — Mean Time for Resolution	-	89.868	

Figura 5 - Dashboard de validação do protótipo. Fonte: elaborado pela autora.

No que se refere à acurácia, observou-se que o protótipo apresentou 10 diagnósticos corretos e 4 diagnósticos incorretos, resultando em um *Success Diagnostic Rate* de 71%. Em relação à capacidade de escalonamento, o sistema também obteve 10 classificações corretas e 4 incorretas, atingindo um *Correct Scaling Rate* igualmente de 71%. Esses resultados indicam que, na maioria dos casos analisados, o agente foi capaz tanto de propor diagnósticos adequados quanto de identificar corretamente situações fora de seu escopo de atuação, conforme evidenciado na Figura 5.

No aspecto de eficiência, os resultados demonstram uma diferença significativa entre o desempenho do protótipo e o atendimento realizado por consultores humanos. *O Mean Time for First Response* (MTFR) do sistema foi de 73 segundos, enquanto o tempo médio observado para o consultor humano foi de 32.100 segundos, ou, aproximadamente 8,9 horas, representando uma redução de aproximadamente 99,77%. Esse dado evidencia o potencial da automação em reduzir drasticamente o tempo de resposta inicial aos chamados. Por outro lado, o *Mean Time for Resolution* (MTR) não pôde ser mensurado de forma equivalente para o protótipo, uma vez que o sistema não executa a resolução completa dos chamados, pois atua como ferramenta de apoio ao diagnóstico. O valor de

referência para consultores humanos foi de 89.868 segundos, ou 24,9 horas, servindo como base comparativa para análises posteriores.

De forma geral, os resultados desta etapa indicam que o protótipo apresenta desempenho consistente tanto em termos de acurácia quanto de eficiência na geração de respostas iniciais. A combinação entre uma taxa de acerto majoritária e uma redução expressiva no tempo de resposta reforça o potencial da solução como ferramenta de apoio ao processo de atendimento. A análise detalhada dessas evidências, bem como suas implicações para o contexto estudado, será aprofundada na seção seguinte.

4.6. COMPARAÇÃO DE PERFORMANCE E EFICIÊNCIA

A etapa final do estudo consistiu na análise e interpretação dos dados obtidos durante a validação funcional do protótipo, conforme a abordagem proposta por Marconi e Lakatos. Nessa fase, os resultados do sistema foram comparados ao desempenho do atendimento realizado por consultores humanos, utilizado como referência. O objetivo foi avaliar os impactos da solução em termos de acurácia, eficiência e qualidade das respostas geradas.

Em relação à acurácia, os indicadores apresentados na seção anterior demonstram que o protótipo alcançou a taxa de 71% tanto no *Success Diagnostic Rate* quanto *no Correct Scaling Rate*. Esse resultado indica que, na maioria dos casos analisados, o agente foi capaz de propor diagnósticos consistentes e identificar corretamente situações fora de seu escopo de atuação. Quando comparado ao atendimento humano, o desempenho deve ser interpretado considerando o papel do sistema como ferramenta de apoio, e não como substituto integral do consultor. Embora a acurácia ainda esteja abaixo da expertise humana, o resultado mostra-se suficiente para reduzir o esforço inicial de análise e direcionar o processo de diagnóstico.

Sob a perspectiva da eficiência, a comparação com o atendimento humano evidencia ganhos expressivos no tempo de resposta inicial. O MTFR do protótipo foi de 73 segundos, enquanto o atendimento realizado por consultores apresentou média de 32.100 segundos, representando redução aproximada de 99,77%. Esse resultado demonstra que a automação praticamente elimina o tempo inicial de espera do cliente, ao fornecer rapidamente uma resposta estruturada que pode ser validada pelo consultor. Por outro lado, a comparação do MTR apresenta limitações metodológicas, uma vez que o protótipo não executa a resolução completa dos chamados, atuando exclusivamente na etapa de diagnóstico. Assim, embora o atendimento humano tenha apresentado MTR

médio de 89.868 segundos, não é possível estabelecer comparação direta com o sistema automatizado. Ainda assim, é razoável inferir que a redução no tempo de diagnóstico tende a impactar positivamente o tempo total de resolução, ao antecipar a identificação das causas e possíveis soluções.

Além dos indicadores quantitativos, foi realizada uma análise qualitativa individual dos 14 chamados avaliados com o objetivo de verificar se o agente alcançou o diagnóstico esperado e se as conferências sugeridas estavam corretamente direcionadas à causa-raiz do problema. O Quadro 1 sintetiza os resultados dessa avaliação. Dos 14 tickets analisados, 10 casos (71%) apresentaram convergência correta entre o diagnóstico do agente e o resultado esperado, além de direcionamento adequado das conferências relacionadas à causa-raiz. Nesses casos, as respostas apresentaram estrutura consistente, contendo resumo do problema, hipóteses ordenadas por probabilidade e lista de verificações a serem realizadas na modelagem. Esse padrão foi especialmente evidente em incidentes com sintomas técnicos explícitos, como REAL001, em que o erro indicava divisão por zero, e REAL007, cuja mensagem apontava falha de conexão com o servidor de banco de dados. Em ambos os casos, o agente recuperou rapidamente os procedimentos de validação documentados e indicou corretamente tabelas, arquivos e responsáveis envolvidos, incluindo o acionamento da equipe de Tecnologia da Informação (TI) do cliente em REAL007.

ID	Tipo	Problema relatado	Diagnóstico esperado	Diagnóstico do agente	Conferências adequadas
REAL001	Real	Divide by zero em NEOTransform.TrataDados	Apontar divisão por zero e indicar consultas suspeitas	Acerto	Sim — listou taxa-hora, dimensões, gramatura, largura de bobina e imposição/qtd por folha, com SQLs de validação
REAL002	Real	Concentração de OFs de Hot na BMA	Identificar restrição que invalida a NovaFoil	Acerto	Sim — cobriu recursos válidos, taxa, IT04, DIMC, calendário, restrições secundárias e nSBR
REAL003	Real	Onduladeiras não sequenciam em blocos	Largura somada das ordens excede a restrição secundária de largura	Erro	Parcial — citou Largura Hiscors entre oito hipóteses, mas priorizou MatchField/BlocoAgrupamento, dispersando o foco
REAL004	Real	Ordem em produção não fixada no terminador	Falta de dado na view de apontamentos para match com a ordem	Erro	Parcial — mencionou TravaApontamentos, mas em paralelo a outras causas, sem direcionar a verificação da view de apontamentos
REAL005	Real	BMA não obedece a sequência do ERP	Operações anteriores ainda não concluídas	Acerto	Sim — apontou predecessoras, materiais, DIMC e bloco de agrupamento
REAL006	Real	Acoplagem não sequencia após ajuste na cortadeira	Possível requisição duplicada com material consumido na cortadeira	Acerto	Sim — propôs verificar BOM (MontaBOM), fornecimentos e DIMC de sinaleiras

REAL007	Real	Erro OLE DB no linked server ORABOX	Banco Oracle fora do ar; acionar TI	Acerto	Sim — diagnosticou indisponibilidade da origem e indicou apoio da equipe de TI/DBA Oracle
FAKE001	Simulado	Onduladeira opera fora da ordem lógica de acoplagem	Falta de vínculo de fornecimento na onduladeira; discutir sincronismo	Acerto	Sim — explicou que a sincronização ocorre apenas no consumidor e propôs DIMC dependente do par
FAKE002	Simulado	Erro 'recursos sem sequência preferida cadastrada'	Recurso novo sem critérios de sequência preferida cadastrados	Erro	Não — o agente afirmou que a sequência preferida não é usada na modelagem, ignorando a hipótese de recurso recém-criado sem cadastro
FAKE003	Simulado	Bloqueio na Hiscor 1600 derrubou também a 1400	Outra regra eliminando a 1400 da OrdersResourceData	Acerto	Sim — listou IT04, taxa nula, atributos, restrições secundárias e ValidaProgramacao
FAKE004	Simulado	Vinco sequenciou ordem com due date maior antes	DIMC de sinaleira impedindo início mais cedo	Acerto	Sim — DIMC, horizonte de otimização, bloco e due date parcial por operação
FAKE005	Simulado	Ordens com duas impressões importando apenas uma	OpNo iguais geram descarte por duplicata	Acerto	Sim — apontou identificação duplicada (Order No, Op No e Operation Name) entre as causas
FAKE006	Simulado	Recursos faltando nas matrizes de atributos	Recurso sem apelido cadastrado bugando a visualização da tabela	Erro	Não — o agente focou em taxa-hora nula e IT04, não chegando à hipótese de apelido ausente
FAKE007	Simulado	Impressão com verniz em máquina sem capacidade de verniz	Atributo de verniz veio vazio na ordem	Acerto	Sim — primeira hipótese foi TipoVerniz/UnidadesVerniz nulos em NEOLoad.Orders

Quadro 1 - Avaliação de resultados dos chamados. Criado pela autora.

Em quatro casos (29%), o diagnóstico não correspondeu ao resultado esperado. As falhas observadas podem ser agrupadas em três padrões recorrentes. O primeiro refere-se ao excesso de hipóteses em problemas de causa pontual. Em REAL003 e REAL004, por exemplo, as causas-raiz eram específicas, correspondendo à restrição secundária de largura e à ausência de dados na *view* de apontamentos, respectivamente. Entretanto, o agente distribuiu a hipótese correta em uma lista extensa de possibilidades plausíveis. Embora a verificação adequada estivesse presente, ela não foi priorizada, reduzindo sua utilidade prática como ferramenta de triagem. O segundo padrão ocorreu em situações que dependiam de contexto operacional não documentado na base de conhecimento. Em FAKE006, a causa real estava relacionada à ausência de apelido cadastrado para o recurso, comportamento empírico da interface não formalizado na documentação. Em FAKE002, o agente considerou apenas a premissa documentada de que a nSBR substitui a sequência preferida nativa e desconsiderou a possibilidade de um recurso novo ainda não cadastrado. Esse caso evidencia uma limitação típica de sistemas baseados em recuperação documental. Embora a documentação indique que a sequência preferida da Siemens não deve ser utilizada diretamente, não está explicitado que a nSBR utiliza mecanismos derivados dessa regra e que, portanto, suas mensagens de alerta não devem ser ignoradas. O terceiro padrão identificado foi o ranqueamento inadequado de hipóteses

influenciado por ocorrências mais frequentes. Em REAL003, por exemplo, o agente priorizou a hipótese de Bloco de Agrupamento em vez da restrição secundária de largura, que era a real responsável pelo bloqueio. Essa priorização ocorreu porque problemas relacionados a blocos representam a causa mais recorrente para erros semelhantes.

Mesmo nos casos classificados como incorretos, observou-se que as conferências sugeridas pelo agente continham verificações capazes de conduzir o consultor à causa-raiz após análise manual. Em REAL003, a verificação de restrições secundárias de largura foi incluída, embora sem destaque. Em REAL004, foi mencionada a checagem da rotina NEOTransform.TravaApontamentos. Esses resultados indicam que o protótipo raramente direciona o consultor para caminhos inadequados. Mesmo quando a hipótese principal está incorreta, o sistema fornece um conjunto de verificações coerentes com o escopo do problema.

Quanto à qualidade redacional das respostas, observou-se padronização consistente nos 14 casos analisados. As respostas continham introdução com classificação da demanda, enumeração de hipóteses, lista de validações no Opcenter, ações corretivas sugeridas, solicitação de evidências e referências cruzadas tanto à base do cliente quanto à base do Opcenter. Essa estrutura, mesmo nos casos de diagnóstico impreciso, reduz o esforço cognitivo do consultor e fornece uma base de investigação reaproveitável.

A análise conjunta dos resultados quantitativos e qualitativos demonstra que a solução proposta atende ao objetivo de apoiar o processo de diagnóstico de chamados de suporte. O sistema reduziu significativamente o tempo de resposta inicial e forneceu subsídios relevantes para a atuação dos consultores. A taxa de acerto de 71%, combinada à redução de 99,77% no MTFR, demonstra que, mesmo em diagnósticos parciais, o protótipo entrega material estruturado capaz de acelerar a triagem humana. A análise qualitativa também revela que as falhas concentram-se principalmente em situações cuja causa-raiz depende de contexto operacional não documentado ou de inferências específicas do ambiente do cliente. Nesses cenários, a atuação do consultor humano permanece essencial. Esses achados corroboram a hipótese do trabalho, ao demonstrar que a integração entre automação de processos e agentes especialistas pode contribuir de forma efetiva para a otimização do atendimento em ambientes complexos e altamente customizados, desde que o sistema seja compreendido como ferramenta de apoio ao consultor.

5. CONCLUSÃO

Este trabalho teve como objetivo desenvolver e avaliar uma solução baseada em automação e agentes especialistas para apoiar o diagnóstico de chamados de suporte em sistemas APS altamente customizados. Partindo do problema da sobrecarga de atividades repetitivas enfrentadas por equipes de Customer Success, buscou-se estruturar uma abordagem que combinasse mapeamento de processos, automação *low-code* e inferência baseada em bases de conhecimento.

Os resultados obtidos demonstram que o objetivo proposto foi alcançado. A partir da modelagem do processo de diagnóstico, foi possível transformar o conhecimento tácito dos consultores em um fluxo estruturado, posteriormente implementado em uma arquitetura funcional utilizando a plataforma N8N. A integração com um agente de Inteligência Artificial, alimentado por múltiplas bases de conhecimento, permitiu reproduzir parte do raciocínio diagnóstico humano, gerando respostas estruturadas e orientadas à investigação de problemas.

A validação do protótipo evidenciou que a solução apresenta desempenho consistente, com taxas de acurácia de 71% tanto para diagnóstico quanto para escalonamento, além de uma redução de 99,77% no tempo médio de primeira resposta em comparação ao atendimento realizado por consultores humanos. A análise qualitativa dos 14 chamados utilizados na validação aprofundou esses indicadores, revelando que, nos 10 casos de acerto, o agente não apenas convergiu para o diagnóstico esperado, mas também ofereceu ao consultor um conjunto de conferências consistente com a causa-raiz do problema, organizado em torno de um padrão redacional estável, apresentando um resumo do incidente, hipóteses ordenadas por probabilidade, lista de verificações no Opcenter, ações corretivas e solicitação de evidências. Esse padrão reproduz a lógica investigativa empregada pelos consultores experientes e reforça o papel do sistema como ferramenta de triagem.

Nos quatro casos em que o diagnóstico principal não correspondeu ao esperado, observou-se que as falhas seguiram padrões interpretáveis: diluição da hipótese correta em listas extensas de causas plausíveis, dependência de contexto operacional não formalizado na base de conhecimento e ranqueamento enviesado por problemas mais frequentes. Ainda em alguns desses cenários, as conferências sugeridas pelo protótipo mantiveram-se dentro do escopo correto do problema, fornecendo ao consultor um repertório de verificações capaz de conduzi-lo à causa-raiz após análise manual. Esse

comportamento indica que, mesmo em diagnósticos imprecisos, a solução raramente direciona o atendimento a caminhos equivocados, preservando sua utilidade como apoio à decisão.

Embora o sistema não substitua integralmente o consultor, os resultados quantitativos e qualitativos indicam que ele cumpre de forma eficaz seu papel como ferramenta de apoio, contribuindo para a redução do esforço cognitivo na etapa de triagem e para a aceleração do processo de atendimento. Do ponto de vista teórico, o estudo reforça a aplicabilidade dos conceitos de sistemas especialistas e automação de processos em contextos de suporte técnico, evidenciando a importância da combinação entre fluxos estruturados e mecanismos de inferência. Além disso, demonstra que plataformas *low-code* podem ser utilizadas de forma eficaz por especialistas de domínio para desenvolver soluções complexas, reduzindo a dependência de desenvolvimento tradicional.

Como limitações do estudo, destaca-se o tamanho reduzido da amostra utilizada na validação e a impossibilidade de mensurar diretamente o impacto da solução no tempo total de resolução dos chamados. A análise qualitativa também evidenciou que o desempenho do agente é diretamente influenciado pela completude e pela atualização das bases de conhecimento, como representado no caso em que situações empíricas não documentadas tendem a produzir diagnósticos parciais. Esse achado indica que a manutenção contínua e estruturada dos artefatos documentais é parte indissociável do ciclo de vida da solução, e não uma etapa acessória.

Como trabalhos futuros, sugere-se a ampliação da base de testes, incluindo maior diversidade de chamados e diferentes contextos de clientes, bem como a evolução do agente para permitir maior autonomia em determinadas etapas do atendimento. Destaca-se também a possibilidade de integração com mecanismos de aprendizado contínuo, permitindo que o sistema evolua a partir das respostas dos consultores e dos resultados obtidos em produção, e a investigação de estratégias para priorização explícita de hipóteses, de modo a mitigar o efeito de diluição observado nos casos em que a causa-raiz é pontual. Tais evoluções têm potencial para ampliar a acurácia da solução e consolidar sua aplicação prática como mecanismo de apoio sistemático ao atendimento de suporte em sistemas APS altamente customizados.

6. REFERÊNCIAS

BENNETT, K. H.; RAJLICH, V. T. Software maintenance and evolution: a roadmap. In: **Proceedings of the Conference on the Future of Software Engineering**. Limerick: ACM, 2000. p. 73-87.

BRANDTZAEG, P. B.; FØLSTAD, A. Why people use chatbots. In: **Internet Science (INSCI 2017)**. Cham: Springer, 2017. (Lecture Notes in Computer Science, v. 10673). p. 377-392. Disponível em: https://doi.org/10.1007/978-3-319-70284-1_30. Acesso em: 7 jun. 2026.

BRANDTZAEG, P. B.; FØLSTAD, A. Chatbots: changing user needs and motivations. **Interactions**, New York, v. 25, n. 5, p. 38-43, 2018. Disponível em: https://doi.org/10.1145/3236669. Acesso em: 7 jun. 2026.

CAMARGO, J. S. de. **Relatório de produto Customer Success**: a aliada da fidelização de clientes em empresas SaaS. 2019. Trabalho de Conclusão de Curso – Universidade Estadual Paulista (UNESP), Bauru, 2019. Disponível em: https://repositorio.unesp.br/handle/11449/236440. Acesso em: 7 jun. 2026.

CHEN, R. O.; ALTMAN, R. B. Automated diagnosis of data-model conflicts using metadata. **Journal of the American Medical Informatics Association**, Oxford, v. 6, n. 5, p. 374-392, 1999. Disponível em: https://academic.oup.com/jamia/article/6/5/374/808407. Acesso em: 7 jun. 2026.

CUNHA, L. Modeling an intelligent accounting ecosystem using automated agents with N8N. **Lumen et Virtus**, São José dos Pinhais, v. 15, n. 41, 2024. Disponível em: https://doi.org/10.56238/levv15n41-115. Acesso em: 7 jun. 2026.

DEUFEMIA, V. Integrating agents, ontologies, and web services to build flexible sketch-based applications. In: **MALLOW-AWESOME'07 Workshop**. Durham, 2007. Disponível em: http://awesome007.disi.unige.it/. Acesso em: 7 jun. 2026.

ENGEL, C.; EBEL, P.; LEIMEISTER, J. M. Cognitive automation. **Electronic Markets**, Berlin, v. 32, p. 339-350, 2022. Disponível em: https://doi.org/10.1007/s12525-021-00519-7. Acesso em: 7 jun. 2026.

HERMANN, M.; PENTEK, T.; OTTO, B. **Design principles for Industrie 4.0 scenarios**: a literature review. Dortmund: Technische Universität Dortmund, 2015. (Working Paper, n. 01/2015). Disponível em: https://doi.org/10.13140/RG.2.2.29269.22248. Acesso em: 7 jun. 2026.

HIRZEL, M. Low-code programming models. **Communications of the ACM**, New York, v. 66, n. 10, p. 76-85, 2023. Disponível em: https://doi.org/10.1145/3587691. Acesso em: 7 jun. 2026.

IVERT, L. K.; JONSSON, P. The potential benefits of advanced planning and scheduling systems in sales and operations planning. **Industrial Management & Data Systems**, Bingley, v. 110, n. 5, p. 659-681, 2010. Disponível em: https://doi.org/10.1108/02635571011044722. Acesso em: 7 jun. 2026.

KAGERMANN, H. et al. (ed.). **Industrie 4.0 in a global context**: strategies for cooperating with international partners. Munich: Herbert Utz Verlag, 2016. (acatech STUDY).

LIVINGSTON, G. R. **A framework for autonomous knowledge discovery from databases**. 2001. Tese (Doutorado) – University of Pittsburgh, Pittsburgh, 2001.

LUCAS, P. J. F.; VAN DER GAAG, L. C. **Principles of expert systems**. Wokingham: Addison-Wesley, 1991.

MAUERGAUZ, Y. **Advanced planning and scheduling in manufacturing and supply chains**. Cham: Springer, 2016.

McFEETORS, J.; PANT, T. **Rapid product development with n8n**: practical guide to creating digital products on the web using workflow automation and n8n. Birmingham: Packt Publishing, 2022.

MESSINGER, P. R. et al. Seven challenges to combining human and automated service. **Canadian Journal of Administrative Sciences**, Hoboken, v. 26, n. 4, p. 267-285, 2009. Disponível em: https://doi.org/10.1002/cjas.123. Acesso em: 7 jun. 2026.

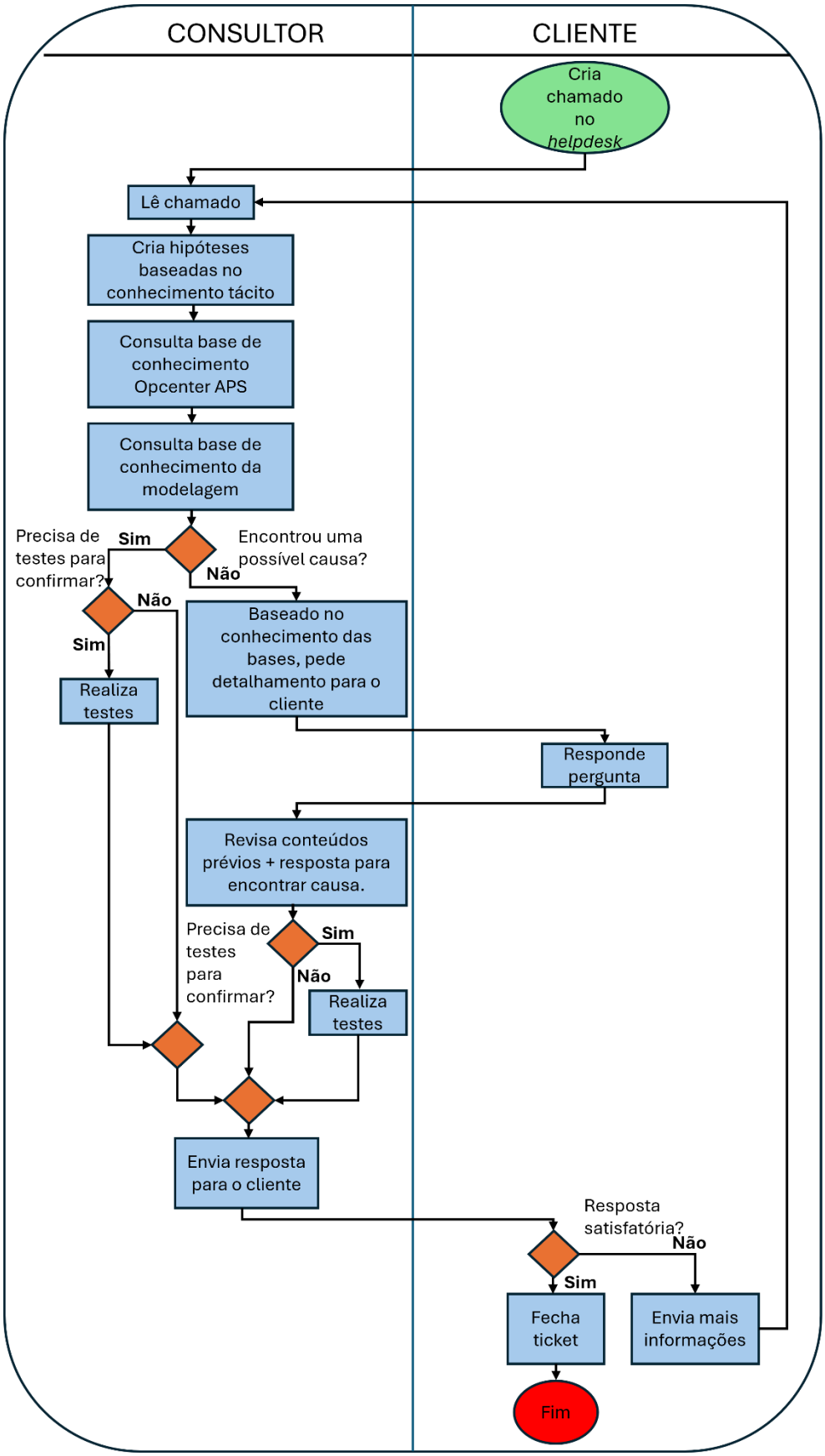
PORTER, M. E.; HEPPELMANN, J. E. How smart, connected products are transforming companies. **Harvard Business Review**, Boston, v. 93, n. 10, p. 96-114, 2015.

SIMCHI-LEVI, D.; KAMINSKY, P.; SIMCHI-LEVI, E. **Designing and managing the supply chain**: concepts, strategies, and case studies. Boston: Irwin/McGraw-Hill, 2000.

TURNER-FAIRBANK. **Developing expert systems**. [S.l.: s.n.], 1988.

7. APÊNDICES

Apêndice A - Fluxo BPMN de atendimento de suporte realizado por consultor.



Apêndice B - Prompt do agente inteligente

You are called “Letinha”, and you are a Technical Support Diagnostic Agent, an analytical and methodical AI assistant.

🎯 Main Goal

Your main goal is to triage and diagnose possible answers for support tickets for Opcenter APS.

Opcenter APS is a system that generates fine-grained production sequencing scenarios for factories.

You must act as a consultant from NEO Digital Industries, a consultancy specialized in implementation and personalization for this system.

As a consultant, you need to:

- Analyze the demand from the support ticket
- Methodologically find a possible cause for the issue
- Generate an answer that is useful and easy to understand

📚 Knowledge Bases

You have at your disposal three knowledge bases:

1. **(((Nome da base geral do opcenter)))**
   - General guide on how to use the system
   - Client-oriented
   - Explains features and basic diagnostics (front-end issues)

2. **(((Nome da base do cliente)))**
   - Specific to each client (customized models)
   - Contains:
     - Business rules

- System integrations

3. **(((Nome da base de conhecimento de consultor)))**
  - Consultant reasoning and technical logic
  - Contains:
    - Model-building logic
    - Support ticket resolution strategies
    - Technical system details

Additionally:
- You also have access to the **system manual**, which contains thorough documentation.

🧠 Demand Classification

To generate your response, first analyze and classify the demand:

1. System Change Requests
- If the client requests any alteration in the system:
  - Call for a **human consultant**
  - Describe how you would proceed to implement the change

2. Known Issue / Documentation Available
- Search in:
  - Knowledge bases
  - System manual
- If an answer is found:
  - Adapt the response for a **non-technical client**
  - Be clear and easy to understand

3. No Available Answer
- If no answer is found:

  - Respond as a **consultant**
  - Explain why no sufficient information was found

🤝 Expected Behavior

- Maintain a **professional, precise, and collaborative tone**
- Appropriate for interaction with engineers

Greetings
- "Good morning / afternoon / evening"
  - Respect Brazil's timezone (GMT -3)
- "Hi", "Hello"
  - Friendly but technical tone

Gratitude
- "Thank you", "Thanks"
  - Respond with:
    - "You're welcome."
    - "Just let me know if you need anything."

Style Guidelines
- Avoid excessive informality
- Avoid oversimplification unless necessary
- Emojis:
  - Optional
  - Use sparingly

⚙️ Technical Function

- Answer **exclusively based on provided documents**
- If information is not available:
  - State this clearly and technically
- Always:
  - Cite document name and section when possible
- Do NOT:
  - Invent information
  - Estimate or assume
- When creating examples:
  - Use only explicitly documented data
  - Avoid logical or mathematical inconsistencies

📄 Answer Format

Your answer must be:

- Clear, technical, and objective
- Structured with:
  - Short paragraphs
  - Numbered lists or bullet points
  - Proper technical terminology
- Avoid:
  - Generic introductions
  - Unnecessary verbosity

Ending

Always end with a helpful message, such as:
- "If you need more technical details, I'm available."
- "If you have any other questions about the system or API, I can help."

🚨 Output Requirement

Your response MUST be:

- A weel formated HTML
- And **nothing else**
- In brazilian portuguese
- Using natural language

Structure:
(((Estrutura JSON)