INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

JOÃO VITOR TRINDADE

CHATBOT RAG NO SUPORTE TÉCNICO: EFICIÊNCIA NO AUTOATENDIMENTO COM INTELIGÊNCIA ARTIFICIAL

CAÇADOR, 2025.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS CAÇADOR
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

JOÃO VITOR TRINDADE

CHATBOT RAG NO SUPORTE TÉCNICO: EFICIÊNCIA NO AUTOATENDIMENTO COM INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador:
Prof. Paulo Roberto Córdova, Dr.

Coorientador:
Cristiano Mesquita Garcia, Me.

CAÇADOR, 2025.

Trindade, João Vitor.
T833c Chatbot Rag no suporte técnico : eficiência no autoatendimento com Inteligência Artificial / João Vitor Trindade ; orientador: Paulo Roberto Córdova, coorientador: Cristiano Mesquita Garcia. -- 2025.
49 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Caçador, 2025.
Inclui bibliografias.

1. Suporte técnico. 2. Chatbots. 3. Retrieval-augmented generation. 4. Large language models. I. Córdova, Paulo Roberto. II. Garcia, Cristiano Mesquita. III. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina – Graduação em Sistemas de Informação. IV. Título.

CDD 600

Ficha catalográfica elaborada pela Bibliotecária
Janice Moser Corrêa – CRB-14/1865

CHATBOT RAG NO SUPORTE TÉCNICO: EFICIÊNCIA NO AUTOATENDIMENTO COM INTELIGÊNCIA ARTIFICIAL

JOÃO VITOR TRINDADE

Este Trabalho foi julgado adequado para obtenção do Título de Bacharel em Sistemas de Informação e aprovado pela banca examinadora do Curso de Sistemas de Informação do Instituto Federal de Educação, Ciência, e Tecnologia de Santa Catarina.

CAÇADOR, 10 de fevereiro de 2025.

Banca Examinadora:

PAULO ROBERTO CORDOVA: 00830004963

Digitally signed by PAULO ROBERTO CORDOVA: 00830004963
DN: CN=PAULO ROBERTO CORDOVA: 00830004963, OU=IFSC - Instituto Federal de Santa Catarina, O=ICPEdu, C=BR
Reason: I am approving this document
Location: your signing location here
Date: 2025.07.29 10:30:32-03'00'
Foxit PhantomPDF Version: 10.1.1

Paulo Roberto Córdova, Dr.

Documento assinado digitalmente
gov.br
CRISTIANO MESQUITA GARCIA
Data: 29/07/2025 13:36:03-0300
Verifique em https://validar.iti.gov.br

Cristiano Mesquita Garcia, Me.

Documento assinado digitalmente
gov.br
CLARIANNE NATALI DE CAMPOS
Data: 29/07/2025 14:16:55-0300
Verifique em https://validar.iti.gov.br

Clarianne Natali Campos, Me.

Documento assinado digitalmente
gov.br
JAISON SCHINAIDER
Data: 29/07/2025 17:56:38-0300
Verifique em https://validar.iti.gov.br

Jaison Schinaider, Dr.

A Deus, por me permitir chegar até aqui,
e à minha família e amigos, pelo apoio
e compreensão nas horas de ausência.

AGRADECIMENTOS

Acima de tudo a Deus, que me permitiu concluir este trabalho, me dando forças para conciliar as adversidades, a rotina e o cansaço dos dias de trabalho.

Ao meu pai de criação e coração, Valdir Piacentini Trindade, por sempre se doar a me apoiar nos estudos acima de qualquer coisa, e me fazer entender desde muito novo que através do conhecimento eu poderia mudar o meu destino.

A minha mãe, Roseli de Fátima Trindade, e minha Avó, Maria Piacentini Trindade, por serem a minha base, de amor, cuidado e carinho, sem elas eu não teria a segurança para novos desafios.

A minha irmã de coração, Viviane Aparecida Trindade, por ser primeiramente exemplo de pessoa, profissional e estudante para mim, por sempre acreditar, e dar a oportunidade para meus novos passos.

A minha família e a quem acredita em mim, e de alguma forma me apoiou tanto em toda a jornada quanto neste trabalho.

Por fim, mas não menos importante, aos meus amigos, por me apoiarem e serem o alívio e a recuperação de energia a essa aventura cansativa que é graduar.

Nada seria possível sem a participação de cada um, agradeço novamente a Deus por me permitir ter á todos e tudo que preciso.

A fé é dar o primeiro passo,
mesmo quando você não vê
a escada toda.
Martin Luther King Jr.

RESUMO

O uso de sistemas como ERPs (*Enterprise Resource Planning*) e ECMs (*Enterprise Content Management*) é essencial para a gestão de processos e a produtividade nas empresas modernas. Entretanto, a crescente complexidade desses sistemas e a dependência tecnológica, geram desafios para o suporte técnico, especialmente devido à frequência de incidentes repetitivos com soluções já documentadas, mas de difícil acesso pelos usuários. Essa situação sobrecarrega as equipes de suporte e reduz a eficiência. Este projeto explora a aplicação de *chatbots* integrados a *Large Language Models* (LLMs) combinados com a técnica de *Retrieval-Augmented Generation* (RAG) para reduzir a interação humana do suporte técnico em incidentes de TI com soluções já catalogadas, diminuindo assim a sobrecarga no setor. A solução buscou utilizar bases de conhecimento da empresa para proporcionar respostas precisas e contextuais a perguntas feitas ao *chatbot*. A pesquisa analisou a efetividade do *chatbot*, utilizando critérios estabelecidos, que foram avaliados pela equipe de suporte técnico da empresa, com foco em resolver os incidentes descritos. Ao combinar a flexibilidade dos LLMs com o acesso dinâmico aos dados proporcionados pelo RAG, o estudo demonstra como essa abordagem pode ser usada para reduzir a carga de trabalho do suporte humano, melhorar os tempos de resposta e diminuir custos operacionais, representando uma evolução no uso da inteligência artificial para atendimento corporativo.

Palavras-chave: suporte técnico; chatbots; *retrieval-augmented generation; large language models.*

ABSTRACT

The use of systems such as ERPs (Enterprise Resource Planning) and ECMs (Enterprise Content Management) is essential for process management and productivity in modern companies. However, the growing complexity of these systems and the increasing technological dependency pose challenges for technical support, especially due to the frequent recurrence of incidents with already documented solutions that are difficult for users to access. This situation overloads support teams and reduces overall efficiency. This project explores the application of chatbots integrated with Large Language Models (LLMs), combined with the Retrieval-Augmented Generation (RAG) technique, to reduce human involvement in IT support for incidents with previously cataloged solutions, thereby relieving pressure on the support sector. The solution aimed to leverage the company's knowledge bases to provide accurate and contextual responses to questions posed to the chatbot. The research assessed the chatbot's effectiveness using predefined criteria, which were evaluated by the company's technical support team, focusing on resolving the reported incidents. By combining the flexibility of LLMs with the dynamic data access enabled by RAG, the study demonstrates how this approach can be used to reduce human support workload, improve response times, and lower operational costs, representing an evolution in the use of artificial intelligence for corporate support.

Keywords: technical support; chatbots; retrieval-augmented generation; large language models.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BPM	Business Process Management
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning
IA	Inteligência Artificial
IFSC	Instituto Federal de Santa Catarina
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO

Nas empresas modernas, o uso de *softwares* como ERPs (Enterprise Resource Planning) e CRMs (Customer Relationship Management) é essencial. O impacto positivo desses sistemas é evidente na melhoria da produtividade, na redução de custos e no aprimoramento da comunicação interna e externa (Chaffey; Edmundson-Bird; Hemphill, 2019). Para auxiliar na resolução de possíveis problemas do uso cotidiano desses softwares, surge a necessidade do suporte técnico.

O suporte técnico é o serviço oferecido aos usuários para ajudá-los a resolver problemas relacionados ao uso de um *software*, garantindo a funcionalidade do sistema e a satisfação do cliente (Bordoloi; Fitzsimmons; Fitzsimmons, 2019). Normalmente, o atendimento é feito diretamente por técnicos capacitados. Um suporte técnico eficiente pode atuar como um diferencial competitivo, especialmente em mercados de alta tecnologia (Haksever; Render, 2017).

Com o aumento da complexidade dos *softwares* empresariais e a dependência cada vez maior das tecnologias digitais, muitos incidentes de TI recebidos diariamente acabam sendo repetitivos e possuem soluções já documentadas, mas a dificuldade dos usuários em navegar por essas bases de conhecimento e a comodidade faz com que o suporte humano seja acionado mesmo que desnecessariamente.

Segundo AWS (2024) os *chatbots* podem ajudar a resolver chamados de clientes via aplicações de atendimento, e diminuir a carga de trabalho dos funcionários do suporte. Eles realizam desde respostas simples a tarefas complexas, como configuração de serviços ou resolução de problemas, dependendo do contexto utilizado como base, sendo amplamente utilizados em setores como saúde e telecomunicações (Adamopoulou; Moussiades, 2020). Um exemplo de implementação bem sucedida é o *Watson Assistant* [1] da IBM, que tem sido utilizado por grandes empresas para otimizar o atendimento ao cliente, mostrando resultados expressivos, especialmente no suporte a clientes do setor bancário, reduzindo os custos, aumentando a eficiência e obtendo melhora na satisfação do cliente (Brodowicz, 2025).

O uso de *Retrieval-Augmented Generation* (RAG) para o desenvolvimento de *chatbots*, representa uma evolução significativa no campo da inteligência artificial. Conforme descrito por Lewis *et al.* (2020), essa combinação permite acesso a dados atualizados e contextuais, reduzindo erros comuns em modelos treinados exclusivamente com dados estáticos.

Esta pesquisa investiga a viabiliade da utilização de um *chatbot* desenvolvido com Retrieval-Augmented Generation e Large Language Model para fornecer respostas precisas para resolução de perguntas sobre incidentes de TI do suporte técnico de uma empresa de *software*. As respostas foram construídas com base no contexto

[1] https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started

extraído das bases de conhecimento da empresa. A avaliação do *chatbot* foi realizada diretamente pela equipe de suporte, que respondeu uma série de questões avaliativas, como a sua utilidade, a velocidade da resposta, clareza e se ela resolve o problema descrito.

1.1 Justificativa

Segundo a renomada empresa de consultoria Gartner (2023), até 2026 80% das empresas terão automatizado seus processos em algum nível utilizando IA, o que pode reduzir custos e aumentar a eficiência operacional, principalmente no atendimento do suporte técnico aos clientes. Uma das principais maneiras da automatização do atendimento ao cliente, é o uso de *chatbots*. Atualmente, grandes empresas de tecnologia, como a Microsoft e o Google, têm investido fortemente em assistentes virtuais. De acordo com a Microsoft, o uso de seu serviço *"Power Virtual Agents"*, conseguiu reduzir em 60% o volume de chamadas de suporte nas empresas que o adotaram, permitindo que os técnicos se concentrem em questões mais complexas e estratégicas (Microsoft, 2023).

Os *chatbots* podem ser classificados em duas categorias, uma delas são os *chatbots* por regras, que funcionam com base em gatilhos ou palavras-chave, e apresentam usabilidade de atendimento mais limitada. A outra forma seriam *chatbots* com inteligência artificial (IA), que são mais dinâmicos, reconhecem padrões e entendem linguagem natural, oferecendo soluções de forma autônoma sem depender de regras específicas. Esse por sua vez é fruto de avanços de diversas áreas da ciência, como a própria inteligência artificial, processamento de linguagem natural, bancos de dados e redes de comunicação de dados (Cruz; Alencar; Schmitz, 2018).

Utilizar *Large Language Model* (LLM) com *Retrieval-Augmented Generation* (RAG)[2] pode ser uma boa escolha no desenvolvimento de *chatbots*. Destacado como uma abordagem poderosa para responder a perguntas complexas, essa técnica melhora a precisão em cenários que exigem dados específicos, como consultas em grandes bases de conhecimento (Lewis *et al.*, 2020). Outro estudo relevante de Gao *et al.* (2023) mostrou que essa combinação reduz a dependência de atualizações frequentes dos modelos, pois as informações são recuperadas dinamicamente.

Considerando o problema de desnecessária demanda da intervenção humana no suporte de incidentes de TI já catalogados e as demonstrações recentes de sucesso no uso de *Retrieval-Augmented Generation* (RAG), este trabalho propõe o desenvolvimento de um chatbot com técnicas de RAG combinadas com *Large Language Model* (LLM) como solução para automatizar o atendimento a incidentes de TI já catalogados. Esse *chatbot* deve fornecer respostas precisas diretamente de uma base de conhecimento

[2] Os conceitos de LLMs e RAG serão detalhados nos capítulos 2.3 e 2.4.

personalizada, reduzindo assim a carga sobre as equipes de suporte e agilizando o processo.

1.2 Definição do Problema

O setor de suporte técnico das empresas de *software* diariamente lida com incidentes de TI com problemas já documentados, que poderiam ser resolvidos pelo próprio usuário por meio de uma consulta a base de conhecimento, evitando o contato com o suporte. Essa situação sobrecarrega as equipes de suporte técnico, que perdem tempo em tarefas simples, comprometendo a produtividade e aumentando os tempos de resposta para problemas mais críticos. Além disso, a dependência do suporte humano eleva os custos operacionais, à medida que as empresas precisam aumentar suas equipes para atender à crescente demanda.

Conforme relatado pela equipe de suporte de uma empresa de *software* de Santa Catarina, muitos clientes não fazem essa busca por conta própria, seja por falta de praticidade em navegar nas bases de conhecimento extensas ou por preferência da agilidade de atendimento que um funcionário do suporte pode lhes prestar. Segundo o levantamento da equipe consultada, 4 a cada 10 incidentes que diariamente passam pela triagem e chegam ao suporte poderiam ser resolvidos com consulta a documentação. A intervenção humana acaba ocorrendo no atendimentos a incidentes já catalogados, que consomem entre 15 minutos a 1 hora como relatado em pesquisa feita ao suporte técnico da empresa de software. Essa sobrecarga, mesmo que parcialmente evitável, afeta a eficiência da equipe e aumenta os tempos de resposta.

Deste modo, considerando o contexto apresentado, seria viável a utilização de um *chatbot* desenvolvido com *Retrieval-Augmented Generation* e *Large Language Model* para fornecer respostas precisas para a solução de incidentes de TI já catalogados de uma empresa de *software*?

1.3 Objetivo Geral

Este trabalho tem como objetivo avaliar a viabilidade de utilizar um *chatbot*, desenvolvido com a técnica *Retrieval-Augmented Generation* em um *Large Language Model*, para fornecer respostas precisas na resolução de incidentes de TI no suporte de uma empresa de *software*.

1.4 Objetivos Específicos

a) Definir o escopo e as tecnologias;

b) Desenvolver o *chatbot*;

c) Aplicar o *chatbot* para os técnicos do suporte avaliarem as respostas;

d) Analisar as avaliações respondidas.

1.5 Estrutura do Trabalho

O presente trabalho está organizado em cinco capítulos.

O capítulo 2 aborda a fundamentação teórica, essencial para a contextualização e compreensão do tema. Inicialmente, explicamos o conceito de suporte técnico, área foco da aplicação da pesquisa. Em seguida, tratamos sobre chatbots, explicando o que são e como podem ser utilizados. Posteriormente, exploramos o Large Language Model (LLM), modelo de IA empregado na construção do chatbot, descrevendo seus conceitos, principais aplicações e arquitetura. Por fim, discutimos o RAG (Retrieval-Augmented Generation), explicando sua técnica, utilidade, arquitetura, funcionamento de cada camada, casos de uso e realizando uma breve comparação com a técnica de fine-tuning.

O capítulo 3 apresenta a metodologia de pesquisa e a metodologia adotada para o desenvolvimento, detalhando como as tecnologias serão implementadas para alcançar os objetivos descritos na introdução. No capítulo 4, são apresentados os resultados preliminares ou esperados. Já o capítulo 5, por sua vez, trás as considerações finais do trabalho, mostrando um resumo geral de todo o processo, além de uma discussão sobre os resultados obtidos e possíveis melhorias futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será apresentado o conhecimento necessário para o entendimento deste trabalho de pesquisa. Abordaremos os conceitos que envolvem a problemática a ser resolvida, os *chatbots* e as tecnologias que serão utilizadas para o desenvolvimento.

2.1 Suporte Técnico

Segundo Meyrelles (2019), o suporte técnico é um serviço que presta assistência intelectual, tecnológica e material a um determinado usuário, com o fim de solucionar problemas técnicos, para assim garantir o funcionamento adequado. Esse serviço pode abranger atividades como resolução de erros, instalação de atualizações, orientação no uso de funcionalidades, e até treinamento para equipes.

Conforme apontado por Laudon e Laudon (2004), o suporte técnico é essencial dentro da gestão de sistemas de informação, garantindo que os usuários finais usufruam de tecnologias de maneira eficiente e confiável. Ele desempenha um papel crucial no ciclo de vida do *software*, estendendo sua utilidade e alinhando-se às necessidades dinâmicas dos usuários (Pressman, 2010).

Além disso, de acordo com Sommerville (2016), a qualidade do suporte técnico pode ser determinante na percepção de valor de um *software*, influenciando diretamente na fidelização de clientes. Nesse contexto, o suporte não apenas resolve problemas, mas também atua como um elo de comunicação entre os desenvolvedores e os usuários finais, promovendo melhorias contínuas no produto (McConnell, 2004).

Segundo Laudon e Laudon (2004), o suporte técnico é tipicamente dividido em vários níveis para otimizar a eficiência operacional. O suporte de primeiro nível trata das solicitações mais comuns e de baixa complexidade, enquanto os níveis subsequentes são projetados para lidar com problemas de maior complexidade e, frequentemente, colaboração entre diferentes equipes técnicas.

2.2 Chatbots

Segundo Schlicht (2016), um *chatbot* é um serviço de interação com o usuário que pode funcionar com respostas pré-programadas baseadas em regras, ou de forma mais avançada, utilizando inteligência artificial para compreender e responder de maneira dinâmica às necessidades dos usuários.

Eles são frequentemente empregados em serviços de atendimento ao cliente, aplicativos de mensagens e sistemas de automação, com o objetivo de oferecer respostas rápidas e precisas às solicitações dos usuários (Russell; Norvig, 2016).

Além de seu uso comercial, Følstad e Brandtzæg (2017) destacam que *chatbots* podem atuar como assistentes educacionais, ajudando estudantes com tarefas mais

específicas para seu aprendizado, o que é cada vez mais notável desde o fundamental ao ensino superior. Além disso, Dale (2016) observa que esses sistemas são projetados para serem intuitivos e interativos, facilitando o autoatendimento, cada vez tornando o usuário mais independente.

A partir do desenvolvimento dos LLMs, avanços na tecnologia de IA têm permitido a criação de *chatbots* mais sofisticados, capazes de entender contextos complexos e oferecer interações personalizadas (Jurafsky, 2000).

2.3 Large Language Model

Large Language Models (LLMs) são modelos de aprendizado de máquina especializados em compreender, gerar e processar linguagem natural com alta precisão. Em outras palavras, eles são projetados para entender e gerar texto como um humano. Baseados em redes neurais profundas, esses modelos necessitam ser treinados em vastas quantidades de dados textuais e utilizam a arquitetura *Transformer*, introduzida por Vaswani *et al.* (2017), que revolucionou o processamento de linguagem natural. Essa arquitetura é fundamentada no mecanismo de atenção, que permite ao modelo identificar relações contextuais entre palavras em uma frase, independentemente de sua posição. Segundo Fröhlich e Soares (2018), eles são desenvolvidos com o objetivo de proporcionar ao usuário a impressão de estar interagindo com outra pessoa, simulando a naturalidade e fluidez de uma conversa agradável.

A principal característica dos LLMs é a sua capacidade de aprendizado contextual, permitindo-lhes realizar tarefas como tradução, resumo de textos e geração de conteúdo criativo, além de conseguir manter uma sequência de respostas contínua com o usuário, semelhante a uma conversa. Esses modelos são construídos com bilhões ou até trilhões de parâmetros, que representam os pesos aprendidos durante o treinamento, o que os torna melhores em entender ambiguidades e detalhes da linguagem (Brown, 2020).

Os LMMs representam um marco no campo do processamento de linguagem natural (NLP), revolucionando a área da interação humano-computador, consolidando-se como ferramentas essenciais para inovação tecnológica (Bommasani *et al.*, 2021). Uma das suas aplicações mais comuns são em assistentes virtuais e chatbots, que hoje em dia são facilmente acessíveis ao público por meio de interfaces como por exemplo o ChatGPT[1], podendo auxiliar em praticamente todas as áreas de segmento e até em tarefas do dia a dia. Outras utilizações mais profissionais são nos sistemas de atendimento ao cliente que simulam interações humanas com alto grau de naturalidade (Brown, 2020). Esses sistemas interpretam entradas com o objetivo de produzir determinados comportamentos, como por exemplo de um atendente, esse processo é conhecido como "*Prompt Engineering*".

[1] http://chatgpt.com

O *Prompt Engineering* é a prática de formular instruções de forma adequada para guiar o comportamento de modelos de linguagem. Por exemplo, em vez de simplesmente perguntar “Como configuro a assinatura de um documento?”, um *prompt* bem estruturado poderia ser: “Baseado na documentação da plataforma, explique passo a passo como configurar a assinatura em um tipo de documento específico. Se a informação não estiver disponível, informe isso.” Essa formulação reduz ambiguidades e melhora a precisão da resposta do modelo. Esta prática tem sido amplamente estudada como uma forma de alinhar os modelos às intenções humanas sem necessidade de ajuste nos pesos do modelo.

2.3.1 Arquitetura Transformer

A arquitetura *Transformer*, proposta por Vaswani *et al.* (2017) é composta principalmente por dois componentes fundamentais: o *Encoder* e o *Decoder*. Esses componentes desempenham o papel do processamento de sequências de dados, como textos, sem a necessidade de redes recorrentes ou convolucionais[2], permitindo maior eficiência e paralelismo.

[2] Redes neurais recorrentes (RNNs) processam dados sequenciais, como texto ou séries temporais, utilizando conexões cíclicas para manter informações ao longo do tempo, sendo úteis em tarefas como tradução automática (Graves, 2012). Já as redes convolucionais (CNNs) são especializadas em dados com estrutura em grade, como imagens, extraindo características através de filtros convolucionais, também conhecidos como kernels, que são matrizes pequenas, sendo amplamente usadas em visão computacional (LeCun *et al.*, 1998). Ambas influenciaram o avanço da IA antes da ascensão da arquitetura *Transformer*.

Figura 1 – Arquitetura Transformer.

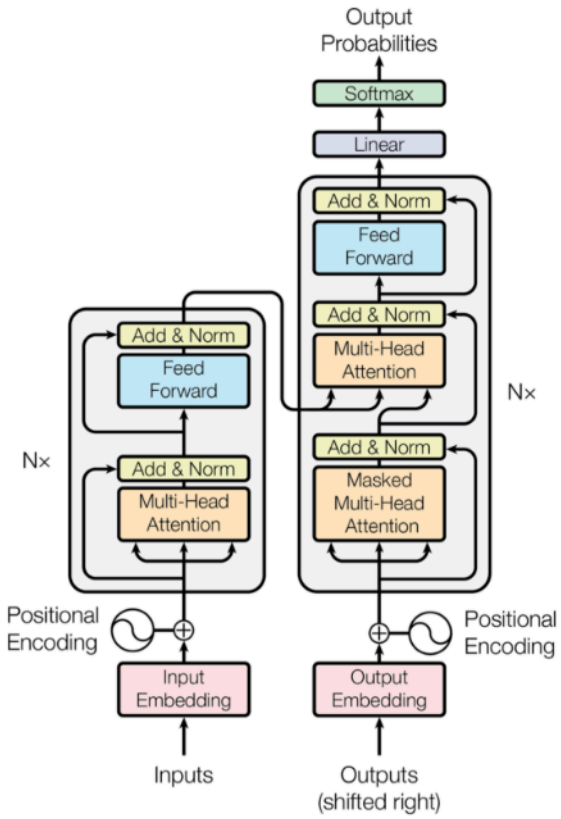


Figure 1: The Transformer - model architecture.

Fonte: Alves (2022).

O *encoder* inicia o processo convertendo os *tokens* de entrada em representações vetoriais através da camada de *embedding*. Em seguida, o processo de *Positional Encoding* para incorporar as informações sobre a posição dos *tokens*, visto que o modelo não possui uma estrutura sequencial. O componente chave do *encoder* é o *Self-Attention*, que permite que cada *token* preste atenção em todos os outros *tokens* na sequência de entrada, capturando as relações contextuais (Vaswani *et al.*, 2017). A saída dessa atenção é refinada por uma *Feed-Forward Neural Network*, e processos de Normalização e *Dropout* são aplicados para estabilizar o treinamento e reduzir o risco de *overfitting*.

Por outro lado, o *Decoder* é responsável pela geração da sequência de saída. Assim como o *encoder*, o *decoder* começa com camadas de *Embedding* e *Positional Encoding*. Contudo, o *Masked Self-Attention* é utilizado para garantir que a geração da saída seja autoregressiva, ou seja, o modelo não pode acessar informações sobre *tokens* futuros durante o treinamento. Além disso, o *Cross-Attention* permite que o *decoder* utilize as representações geradas pelo *encoder*, melhorando o contexto durante a geração da saída. Após a atenção, a informação é refinada por outra *Feed-Forward Neural Network*, e, como no *encoder*, também são aplicadas técnicas de

Normalização e *Dropout* (Vaswani *et al.*, 2017).

2.4 Retrieval-Augmented Generation

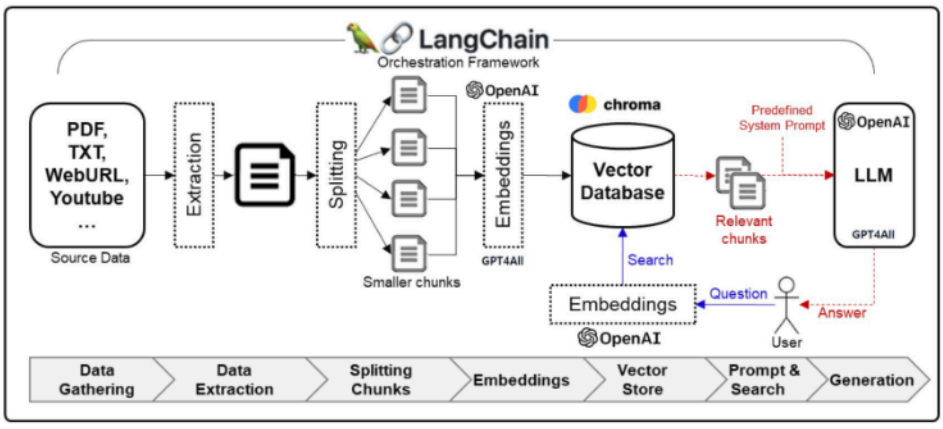
Segundo AWS (2024), *Retrieval-Augmented Generation* (RAG) é um processo que aprimora as respostas fornecidas por um grande modelo de linguagem, permitindo que ele utilize informações provenientes de uma base de conhecimento armazenada em um banco de dados vetorial contendo informações externas às suas fontes de treinamento. Em RAG, o modelo generativo utiliza informações recuperadas de uma base de conhecimento ou banco de dados externo para fundamentar suas respostas, promovendo resultados mais contextuais e confiáveis (Lewis *et al.*, 2020).

Ao integrar recuperação de dados atuais para geração de respostas com contexto personalizados, RAG supera limitações dos modelos tradicionais de geração de texto, que frequentemente dependem exclusivamente de dados pré-treinados. Essa abordagem permite que o sistema acesse informações atualizadas, evitando problemas de alucinação gerativa, em que o modelo inventa informações incorretas (Guu *et al.*, 2020).

2.4.1 Arquitetura RAG

A rotina de funcionamento RAG consiste basicamente na recuperação de informações e geração de texto assistida por LLMs. Essa rotina é dividida em várias etapas que formam sua arquitetura, conforme a Figura 2.

Figura 2 – Arquitetura RAG.



Fonte: Jeong (2023).

2.4.1.1 Tratamento dos dados

A primeira etapa deve ser a de tratamento dos dados que servirão de base para contexto, esses dados podem incluir: documentos, tabelas de banco de dados ou

APIs e fontes externas. Após coletados os dados, eles devem passar por um processo chamado *splitting*, onde são separados em fragmentos com tamanhos delimitados chamados de *chunks*. Segundo Lewis *et al.* (2020), dividir documentos em *chunks* reduz a complexidade do modelo ao permitir o processamento em trechos menores e aumenta a relevância ao indexar porções específicas de informações.

Para exemplificar o conceito de *splitting*, vamos fragmentar uma frase em três pedaços, que seriam os *chunks*. Temos a frase original: "A inteligência artificial revolucionou diversas indústrias, sendo aplicada em áreas como saúde, finanças, educação e transporte, promovendo eficiência e inovação.". Após passar pelo processo de *splitting* a frase se transformaria em três partes, que seriam "A inteligência artificial revolucionou diversas indústrias.", "Sendo aplicada em áreas como saúde, finanças, educação e transporte.", "Promovendo eficiência e inovação.".

Com os fragmentos de informação prontos, cada "*chunk*" é transformado em um vetor, utilizando modelos de *embeddings*, como Word2Vec para palavras e *OpenAI Embeddings* para palavras e documentos. Os vetores, são associados a metadados que facilitam a identificação e a busca, como no exemplo anterior, o metadado poderia ser o capítulo do artigo em que as frases aparecem.

Após gerados os *embeddings*, eles são armazenados em um banco de dados vetorial que cria índices otimizados para facilitar a recuperação de dados por similaridade. Modelos como *Hierarchical Navigable Small World* (HNSW) ou *Inverted File Index* (IVF) são comuns para essa tarefa.

2.4.1.2 Banco de dados vetorial

Bancos de dados vetoriais são sistemas especializados no armazenamento e recuperação de dados em formato vetorial, frequentemente gerados por técnicas de *embeddings*. Esses bancos são projetados para lidar com grandes volumes de dados, tornando possível a busca eficiente de dados complexos.

A principal função dos *VectorDBs* é a realização de buscas por similaridade, como *nearest neighbor search*, que permite localizar dados com base na proximidade de seus vetores em um espaço multidimensional. Essas buscas de dados similares geram contextos que serão utilizados para alimentar o LLM que fará a geração da resposta.

Alguns banco de dados vetoriais mais populares são o ChromaDB [3] e FAISS [4], que oferecem operações eficientes para recuperação de *embeddings* e podem ser integrados a *frameworks* RAG como o Langchain [5].

[3] https://www.trychroma.com/
[4] https://ai.meta.com/tools/faiss/
[5] https://www.langchain.com/

2.4.1.3 Arquitetura

Após os dados vetorizados que serão utilizados para contexto ja estarem inseridos no banco de dados vetorial, o fluxo de funcionamento do RAG começa com a entrada do usuário, que insere uma consulta ou pergunta no *software* que será enviado para a estrutura RAG. Essa consulta é então vetorizada, ou seja, transformada em um vetor usando modelos de *embeddings* assim como foi feito com os dados da base de conhecimento.

Após a vetorização, o vetor da consulta é comparado aos vetores armazenados no banco de dados vetorial. Utilizando algoritmos de busca, como o *Nearest Neighbor Search*, o sistema recupera os trechos mais relevantes que correspondem à consulta do usuário. Por exemplo, se a consulta for sobre um chamado de *bug* em uma parte específica do sistema, o RAG pode recuperar trechos de artigos da documentação ou de chamados anteriores que tenham relação com esse *bug*. Esses trechos são então combinados com a consulta original do usuário, criando um contexto. Esse contexto é enviado para um LLM, como o GPT-4 por exemplo, que utiliza as informações recuperadas para gerar uma resposta, que é apresentada ao usuário.

2.4.2 Casos de uso de RAG

Ao manter sistemas de processamento de linguagem natural atualizados, sem necessitar de treinamento, uma de suas principais aplicações é em sistemas de perguntas e respostas. A atualização de contexto que o RAG oferece, permite não ser necessário a criação de uma IA ou o retreinamento da mesma, pois o modelo pode gerar respostas com base em dados relevantes recuperados de grandes repositórios personalizados (Karpukhin *et al.*, 2020). Uma dessas aplicações de perguntas e respostas podem ser *chatbots*, especialmente em domínios especializados como assistência médica, jurídico e suporte técnico, onde respostas confiáveis e específicas de cada setor são necessárias (Lewis *et al.*, 2020).

Em educação, RAG é usada para criar sistemas de tutoria que acessam bases de conhecimento para fornecer explicações detalhadas (Izacard; Grave, 2020). Outra aplicação relevante é em pesquisas científicas como esta, onde RAG ajuda a navegar por bases de dados acadêmicas, fornecendo resumos de artigos e dados relevantes (Thakur *et al.*, 2021). Também é usada em *e-commerce*, para gerar descrições de produtos e respostas personalizadas a perguntas dos clientes, e em *marketing* de conteúdo, para criar textos fundamentados e engajantes (Guu *et al.*, 2020).

Com base no que foi apresentado, sistemas baseados em RAG destacam-se por combinar recuperação de informações de um contexto específico com geração de texto otimizada, oferecendo soluções personalizadas com linguagem natural. Essa abordagem reduz falhas, como alucinações gerativas, e demonstra grande potencial

para aplicações diversas, inclusive as propostas neste trabalho.

2.4.3 Retrieval-Augmented Generation *versus* Fine-tuning

Segundo Silva (2018), *fine-tuning* é um processo de adaptação de modelos de aprendizado pré-treinados a tarefas específicas. Nessa técnica, congelam-se algumas camadas do modelo, preservando o conhecimento adquirido em um grande conjunto de dados, e treinam-se as camadas restantes com um conjunto de dados mais específico. Essa abordagem é particularmente útil quando os novos dados apresentam características distintas dos dados originais, permitindo que o modelo aproveite o conhecimento prévio e aprenda a lidar com as novas informações de forma eficiente.

A principal vantagem do *fine-tuning* é que ele reduz o tempo e os recursos computacionais necessários para treinar modelos complexos. Em vez de iniciar o treinamento de um modelo do zero, o modelo pré-treinado serve como um ponto de partida, economizando tempo e recursos, além de melhorar a precisão do modelo em tarefas específicas (Devlin, 2018).

Como dito anteriormente, em RAG, o modelo generativo utiliza informações recuperadas de uma base de conhecimento ou banco de dados para fundamentar suas respostas em tempo real, diferente do *fine-tuning*, que busca melhorar a forma que o LLM compreende novas tarefas. O RAG foca em empregar um contexto específico ao LLM ja treinado sem precisar treinar o modelo novamente, combinando a recuperação de dados e a geração de texto (Lewis *et al.*, 2020).

Apesar de ambos poderem ser usados para adaptação de um modelo de linguagem, a principal diferença entre as duas tecnologias é que o RAG não precisa alterar o aprendizado do modelo de linguagem, ele apenas adiciona contexto externo para a construção da resposta atual, sendo mais rápido e eficiente para atualizações constantes e específicas. Isso é muito interessante para empresas por exemplo, onde um acontecimento do dia anterior, como um incidente de TI, se registrado em uma base de possível acesso, pode ser usado para contexto no dia atual.

3 METODOLOGIA

A metodologia presente nesta pesquisa é de natureza aplicada e quantitativa, com o objetivo de analisar a precisão das respostas fornecidas por um *chatbot* que utiliza *Large Language Models* e *Retrieval-Augmented Generation* para a resolução de chamados de TI em uma empresa localizada em Santa Catarina. A pesquisa aplicada, conforme descrito por Gil (2002), busca desenvolver conhecimentos com propósito prático, sendo especialmente útil neste estudo, onde o foco é a implementação e avaliação do uso de um software que tem o objetivo de auxiliar diretamente no suporte técnico da empresa.

Essa pesquisa possui caráter descritivo e exploratório pois, segundo Lakatos e Marconi (2017), a pesquisa descritiva permite detalhar aspectos específicos de um fenômeno, enquanto a pesquisa exploratória é ideal para investigar novas tecnologias ou metodologias de uso delas, o que se aplica ao desenvolvimento de um *chatbot* que será utilizado no suporte técnico.

Para o desenvolvimento do *chatbot*, foram utilizados métodos de coleta de dados a partir de um levantamento bibliográfico focado nas tecnologias de linguagem natural presentes no momento e nos modelos de implementação RAG, além de estudos de casos sobre assistentes virtuais em cenários de suporte técnico. Essa revisão teórica forneceu a base para desenvolvimento do *chatbot*, alinhando a construção da solução as melhores práticas e inovações na área de Inteligência Artificial.

Por fim, a pesquisa inclui a avaliação das respostas pelos profissionais da área de TI da própria empresa, com foco na acurácia e relevância das respostas geradas. Esse método quantitativo permite avaliar a precisão e qualidade do atendimento.

3.1 Tratamento dos dados

Na etapa inicial do projeto, foi realizado o contato com a empresa parceira para identificar os dados relevantes que formaram a base de conhecimento do *chatbot*, incluindo a documentação do produto, bases de suporte técnico e documentos úteis. Esses dados foram utilizados para criar uma base sólida, que foi vetorizada por meio da geração de *embeddings* utilizando o modelo de embedding intfloat/multilingual-e5-large e armazenada em um banco de dados vetorial ChromaDB [1]. Segundo Mikolov (2013) representações distribuídas de palavras em um espaço vetorial ajudam os algoritmos de aprendizado a alcançar melhor os contextos que serão usados para respostas, agrupando palavras semelhantes.

[1] https://www.trychroma.com/

3.1.1 *Embeddings*

Após definida a base de conhecimento que seria utilizada para contexto do *chatbot*, foi feita a vetorização desses dados, para que o LLM possa consumi-los em seu contexto. Esse processo de vetorização pode ser chamado de geração de *embeddings*.

Embeddings são representações numéricas poderosas utilizadas em recuperação aumentada de informações. Eles transformam palavras, frases ou objetos em vetores multidimensionais, refletindo semelhanças semânticas por meio da proximidade entre os vetores em um espaço vetorial. Essa abordagem é fundamental em sistemas de recuperação de informação, como evidenciado por Deerwester *et al.* (1990), que utilizam a técnica para melhorar a precisão na recuperação de documentos em grandes bases de dados. Por exemplo, *embeddings* podem identificar que "rei" e "rainha" estão semanticamente relacionados, assim como "cachorro" e "*dog*" aparecem próximos em *embeddings* multilinguísticos, facilitando a tradução.

Existem diferentes tipos de *embeddings*, como os *Word Embeddings*, que mapeiam palavras individuais para vetores de alta dimensionalidade e são amplamente utilizados em modelos como o Word2Vec (Mikolov, 2013) e o GloVe (Pennington; Socher; Manning, 2014), capturando relações semânticas entre palavras para tarefas como tradução e análise de sentimentos. Além dos *Word Embeddings*, existem os *Document Embeddings*, que processam textos maiores e representam documentos completos em um único vetor. Por outro lado, os *Contextual Embeddings*, como os utilizados no BERT (Devlin, 2018), capturam o significado dinâmico das palavras, considerando o contexto em que estão inseridas.

No desenvolvimento deste projeto, o modelo de *embedding* utilizado foi o *intfloat/multilingual-e5-large*, que gera embeddings contextuais. Ele é capaz de gerar representações vetoriais otimizadas para múltiplos idiomas, sendo ideal para lidar com consultas e documentos multilíngues de forma eficiente. A escolha desse modelo ocorreu após um processo de avaliação comparativa com outros dois modelos: o *sentencetransformers/LaBSE* e o *BAAI/bge-m3*.

A métrica utilizada para essa avaliação se chama *Top-K Accuracy*, a qual consiste em verificar se a resposta correta está presente entre as primeiras posições retornadas pelo modelo em uma busca por similaridade. Essa abordagem está relacionada às métricas de avaliação de sistemas de recuperação de informação baseadas em ranking, como o *Precision at K*, descrito por Christopher, Prabhakar e Hinrich (2008). No contexto do experimento, foi utilizado um Top-3 Accuracy, ou seja, considerou-se como acerto quando o modelo conseguiu recuperar o trecho relevante dentro dos três primeiros resultados. Esse tipo de métrica é bastante utilizado em tarefas de recuperação de informação e sistemas de busca, pois reflete a capacidade do modelo

em posicionar os documentos mais relevantes entre os primeiros resultados, mesmo que não necessariamente em primeira posição. Assim, o uso do Top-K permite uma avaliação mais flexível e aderente ao comportamento esperado em aplicações práticas de sistemas baseados em embeddings. Para mais informações sobre a avaliação que definiu a escolha do modelo de embedding, consultar a sessão "Escolha do Modelo de *Embedding*", no capítulo "Resultados e Discussões".

Interessante informar que o uso de LangChain [2] facilitou a implementação desses modelos para os testes além do desenvolvimento do *chatbot* em geral. Ele oferece integração com bancos de dados vetoriais e simplifica o armazenamento e recuperação de *embeddings*. Após transformados os dados em vetores, foram armazenados em um banco vetorial, conhecidos como *vectorDB*. Neste projeto o banco de dados escolhido foi o ChromaDB [3].

3.2 Similaridade

Antes de encontrar a similaridade, o *prompt* inserido pelo usuário também passa pelo processo de vetorização com o modelo de embedding. Ou seja, o texto inserido pelo usuário é transformado em um vetor utilizando o modelo de *embedding intfloat/multilingual-e5-large*, da mesma forma que os dados da base de conhecimento. Esse vetor gerado a partir do *prompt* é, então, utilizado para fazer busca por similaridade nos dados vetorizados armazenados no banco de dados, da mesma forma como foi feito com as perguntas no teste para escolha do modelo.

A similaridade é medida utilizando um modelo de recuperação conhecido como modelo vetorial de busca, onde documentos ou textos são representados como vetores em um espaço multidimensional. Quando o usuário faz uma consulta, a busca é realizada para encontrar os documentos cujos vetores estão mais próximos do vetor gerado pela consulta. Essa similaridade é comumente medida utilizando métricas como a distância do cosseno ou a distância euclidiana (Salton; Wong; Yang, 1975).

Neste projeto, foi definido que cada consulta por similaridade deve retornar sete trechos, cada trecho possui até 1500 caracteres. A medida de similaridade utilizada é a similaridade cosseno, que é a métrica padrão adotada pelo ChromaDB via LangChain, utilizando o método similarity_search. Após a recuperação desses trechos, ocorre a concatenação entre o prompt do usuário, os trechos retornados e uma instrução adicional de prompt engineering que orienta o modelo: "Baseado nesta parte retirada da documentação do Fusion Platform, responda à pergunta abaixo. Se você não encontrar a resposta no contexto disponibilizado, diga que não foi possível responder.".

Essa instrução tem como objetivo evitar que o modelo de linguagem (LLM) produza respostas inventadas (*hallucinations*), incentivando-o a assumir a ausência de

[2] https://www.langchain.com/
[3] https://www.trychroma.com/

informação quando o contexto fornecido não for suficiente para responder à pergunta.

3.3 Integração com a API do LLM escolhido

Após a concatenação do contexto recuperado, da pergunta do usuário e da instrução, o texto é enviado via requisição de API para o LLM. O modelo, então, processa todas as informações fornecidas e retorna uma resposta com base nos dados disponibilizados.

O LLM escolhido foi o GPT-4 Turbo, da OpenAI. Essa decisão foi tomada com base em uma série de testes comparativos realizados com outros modelos disponíveis no mercado escolhidos, conforme detalhado no subcapítulo “Escolha do Modelo de Linguagem”, localizado no capítulo "Resultados". O GPT-4 Turbo destacou-se pela sua capacidade superior da compreensão contextual, interpretando com precisão as instruções e selecionando de forma eficiente os trechos mais relevantes dos contextos fornecidos para formular a resposta. Além disso, demonstrou elevada clareza na escolha das palavras e estrutura da resposta, que foram coesas, organizadas e com linguagem acessível ao usuário, o que é muito importante em aplicações que envolvem explicações passo a passo ou tutoriais. Apesar de apresentar um tempo de resposta intermediário (cerca de 20 segundos), seu desempenho interpretativo e a robustez das respostas compensaram amplamente essa latência. Dessa forma, o GPT-4 Turbo mostrou ser o modelo mais adequado para atender aos objetivos da aplicação.

3.4 Desenvolvimento da Interface do usuário

A interface do *chatbot* foi desenvolvida utilizando a biblioteca Streamlit do Python. O Streamlit é ideal para protótipos e testes, onde não é necessária uma interface profissional e sofisticada e sim perfeitamente funcional e prática de ser implementada (Peres, 2023). A interface é composta por um campo de entrada na parte superior da tela, onde o usuário poderá digitar seu *prompt* e um campo de visualização de respostas, no qual as respostas geradas pelo LLM serão exibidas, no formato de um *chat* com *scroll* lateral. Ao ser feita uma entrada, é mostrado um item de carregamento instruindo ao usuário que o chat está processando seu *prompt*, até que a resposta seja totalmente carregada.

Figura 3 – Tela do *chatbot*.

Fonte: Elaborada pelo autor (2025)

3.5 Hospedagem

Para hospedagem do *chatbot* foi utilizada a plataforma Streamlit Community Cloud, uma solução gratuita, prática e eficiente para publicação de aplicações que foram desenvolvidas com a biblioteca Streamlit em Python. Essa plataforma permite que aplicações sejam executadas diretamente na nuvem, sem a necessidade de configurar servidores ou contratar serviços de hospedagem pagos. Para utilizar a plataforma, basta que o repositório do projeto esteja disponível e público no GitHub, dessa forma ele ja estará sendo automaticamente integrado à infraestrutura em nuvem do Streamlit. A implantação foi simples, exigindo apenas a definição de dependências em um arquivo requirements.txt e a seleção do *script* principal da aplicação, que tem a interface do Streamlit. Durante os testes e avaliação dos usuários, a hospedagem demonstrou ser estável, permitindo o acesso ao *chatbot* por qualquer navegador *web*, sem necessidade de qualquer instalação local. Dessa forma, a escolha pela hospedagem via Streamlit se mostrou uma solução adequada tanto em termos de custo-benefício (100% gratuita) quanto de facilidade de implantação e manutenção.

3.6 Avaliação

A avaliação das respostas geradas pelo *chatbot* foi realizada por meio de um formulário do Google descrito no Apêndice A, composto por seis campos de avaliação,

e aplicado aos técnicos de suporte da empresa parceira. Cinco desses campos correspondem a perguntas de múltipla escolha, baseadas em uma escala de 1 a 5, onde 1 representa "muito insatisfatório" e 5, "muito satisfatório". Esses campos foram elaborados com o objetivo de mensurar quantitativamente cinco critérios principais: rapidez da resposta, utilidade, clareza, precisão e satisfação geral. O sexto campo consiste em uma pergunta aberta, permitindo que o avaliador expresse livremente sua opinião sobre o projeto. Essa avaliação qualitativa possibilita uma compreensão mais aprofundada das dores, sugestões ou pontos positivos identificados por cada usuário.

A avaliação da solução desenvolvida foi realizada com 15 colaboradores da empresa, sendo todos os membros do setor de atendimento, além de alguns profissionais do setor de desenvolvimento, selecionados com base em seu conhecimento técnico e envolvimento direto com o produto da empresa. A escolha dos participantes seguiu a técnica de amostragem autoritativa, também conhecida como amostragem intencional, na qual o pesquisador seleciona, de forma deliberada, os indivíduos que julga mais adequados para fornecer informações relevantes ao objeto de estudo. Segundo Gil (2008), nesse tipo de amostragem, o pesquisador utiliza seu julgamento para escolher elementos que considera representativos ou informativos para os fins da pesquisa.

A adoção dessa abordagem justificou-se pelo fato de que os colaboradores do setor de atendimento são os que mais lidam diretamente com as dúvidas dos clientes, possuindo, portanto, amplo conhecimento sobre as principais dificuldades enfrentadas pelos usuários finais. Essa experiência prática os torna especialmente qualificados para avaliar a utilidade do *chatbot*, sobretudo no que se refere à clareza e eficácia das respostas às perguntas frequentes. Embora o público-alvo do sistema seja composto pelos clientes da empresa, considera-se que a amostra selecionada representa adequadamente os principais fluxos e contextos de uso, permitindo validar de forma consistente a funcionalidade, relevância e usabilidade da solução RAG proposta.

4 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados e analisados os resultados obtidos com a realização do trabalho. As informações obtidas a partir dos experimentos e análises são discutidas com base nos objetivos propostos, permitindo avaliar o desempenho da solução desenvolvida, suas contribuições e eventuais limitações. A discussão dos resultados busca contextualizar os achados em relação à literatura existente e às práticas observadas no mercado.

4.1 Escolha do modelo de *embedding*

Os resultados da recuperação de informação por similaridade foram adequados, foi possível mensurar esses resultados com os testes comparativos feitos para definir o modelo de *embedding* com melhor desempenho a ser utilizado no projeto, que atingiu 90% de acurácia Top-K.

Foi elaborado um processo de avaliação comparativa entre três modelos: (a) LaBSE (*sentencetransformers/LaBSE*), (b) E5 (*intfloat/multilingual-e5-large*), e (c) bge-m3 (*BAAI/bge-m3*). Para o experimento, foram criadas três bases distintas no banco vetorial ChromaDB, cada uma populada com a mesma documentação da empresa parceira, mas vetorizada por um dos modelos mencionados.

Em seguida, foram realizadas buscas por similaridade utilizando um conjunto de 30 perguntas elaboradas com base na documentação. A métrica de avaliação utilizado foi *Top-K Accuracy*, que consiste na quantidade de vezes que cada modelo é capaz de recuperar corretamente o contexto mais relevante, considerando até três trechos por consulta, ou seja acurácia Top-K com k=3.

O modelo E5 apresentou o melhor desempenho, recuperando corretamente o trecho adequado em 27 das 30 tentativas. Já os modelos LaBSE e bge-m3 obtiveram, respectivamente, 18 e 23 acertos. Esse resultado foi decisivo para a escolha do modelo de *embedding*, além de demonstrar que o processo de recuperação de contexto estava funcionando adequadamente.

Como explicado no capítulo de metodologia, a métrica utilizada para avaliação dos modelos de *embedding* foi a *Top-K Accuracy*, a qual se mostrou adequada para o objetivo de verificar se os documentos relevantes estavam presentes entre os K primeiros itens retornados pelo sistema. Essa abordagem é baseada nos conceitos de avaliação de ranking em recuperação de informação, como o *Precision at K*, descrito por Christopher, Prabhakar e Hinrich (2008).

Tabela 1 – Desempenho dos modelos de *embedding* na tarefa de recuperação de contexto

Modelo de Embedding	Acertos (de 30)	Acurácia Top-K (k=3)
intfloat/multilingual-e5-large	27	90%
BAAI/bge-m3	23	76,7%
sentencetransformers/LaBSE	18	60%

Fonte: Dados do experimento realizado pelo autor.

Vale ressaltar que embora o modelo *BERTimbau*, desenvolvido especificamente para a língua portuguesa pelo Centro de Pesquisa e Desenvolvimento em Inteligência Artificial (C4AI) da Universidade de São Paulo (USP), em parceria com a IBM Research (Souza; Nogueira; Lotufo, 2020), seja amplamente reconhecido por seu desempenho em tarefas de processamento de linguagem natural em português, ele não foi utilizado nos testes deste projeto. A principal razão para essa decisão foi a necessidade de um modelo de embedding multilíngue, visto que a base documental da empresa parceira contém não apenas textos em português, mas também uma quantidade significativa de conteúdos em inglês. Assim, optou-se por avaliar apenas modelos com suporte nativo a múltiplos idiomas, garantindo uma vetorização mais coerente e eficiente de toda a documentação, independentemente do idioma em que foi escrita.

4.2 Escolha do Modelo de Linguagem

Após a definição do modelo de *embedding*, com o objetivo de selecionar o LLM mais adequado para integrar a solução proposta, foram realizados testes comparativos entre os seguintes modelos: Sabiá-3, GPT-3.5, GPT-4 Turbo e Gemini 1.5 Flash. Todos os modelos foram avaliados com base em um mesmo *prompt* executado cinco vezes, composto por um conjunto de contextos recuperados automaticamente pelo algoritmo de similaridade cosseno no banco de dados vetorial utilizando o modelo de *embedding* escolhido, uma pergunta e instruções explícitas sobre como estruturar a resposta. Os critérios de avaliação incluíram tempo de resposta, qualidade interpretativa, clareza textual e aderência ao contexto.

O primeiro modelo avaliado foi o Sabiá-3, um LLM desenvolvido pela Maritaca AI — um projeto brasileiro criado na Unicamp, sob a liderança do engenheiro Rodrigo Nogueira (Abonizio *et al.*, 2024). Seu desempenho foi surpreendente: respondeu com profundidade, demonstrando boa capacidade de interpretação e identificação precisa dos trechos relevantes do contexto. A resposta apresentou um bom nível de detalhamento e respeito à estrutura esperada. No entanto, esse desempenho veio acompanhado de uma latência elevada, com tempos de resposta que chegavam a ultrapassar os 35 segundos mesmo ao utilizar o mesmo *prompt* mais de uma vez, o que compromete a experiência do usuário em contextos mais dinâmicos como o dia a

dia empresarial.

Em seguida, foi testado o Gemini 1.5 Flash, da Google, que possui como principais atrativos sua alta janela de contexto (até 1 milhão de *tokens*) e baixo custo por *token* processado. O tempo de resposta foi razoável, em torno de 15 segundos, contudo, sua principal limitação esteve na qualidade da resposta: foram observados erros de escolha do contexto correto para basear a resposta, além de construções textuais pouco claras para um "tutorial". Erros pequenos, mas que podem dificultar a compreensão em etapas instrucionais ou explicativas, especialmente em fluxos que exigem precisão.

Na sequência, avaliou-se o GPT-3.5, da OpenAI, que se destacou pelo seu tempo de resposta extremamente rápido, com média de apenas 6 segundos. Apesar disso, o modelo apresentou desempenho insatisfatório em termos de interpretação e construção da resposta. Em todas as tentativas de envio do mesmo *prompt* ele não conseguiu apresentar uma solução adequada ou completa para os problemas propostos, evidenciando limitações na capacidade de contextualização e raciocínio exigido para o projeto.

Por fim, foi testado o GPT-4 Turbo, também da OpenAI, que acabou sendo o modelo selecionado para compor a solução final. Embora seu tempo de resposta tenha sido superior ao do Gemini (cerca de 20 segundos), destacou-se pela elevada qualidade das respostas geradas. Demonstrou excelente capacidade de compreensão do problema, uso preciso dos contextos fornecidos e clareza na resposta. Além disso, apresentou consistência e precisão em todos os testes realizados, mesmo diante de instruções mais complexas. Esses fatores, aliados a confiabilidade da plataforma, justificaram sua escolha, mesmo com um custo financeiro superior em relação aos demais modelos testados.

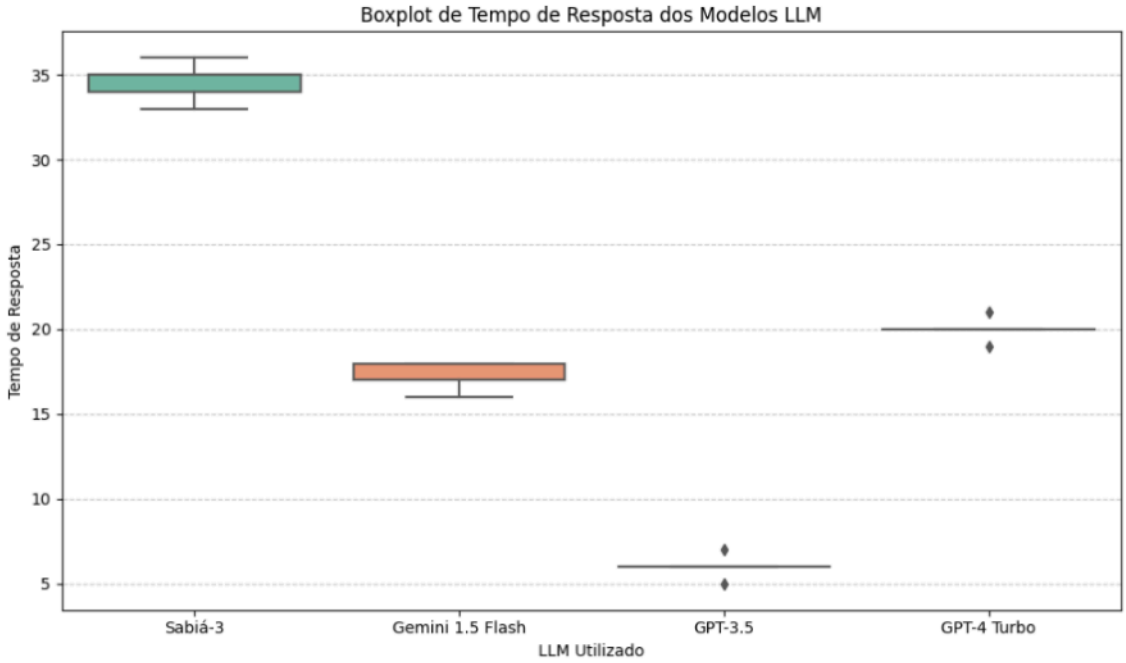
A Figura 4 mostra o exemplo em tela da resposta dada pelo modelo GPT-4 Turbo, escolhido para o projeto, onde é possível perceber a clareza em sua resposta, onde ele descreve os passos precisamente com base na documentação como um tutorial:

Figura 4 – Exemplo pergunta e resposta chatbot.

Fonte: Elaborada pelo autor (2025)

Abaixo no Gráfico 1, observa-se a comparação do tempo de resposta entre os quatro modelos de LLM avaliados: Sabiá-3, Gemini 1.5 Flash, GPT-3.5 e GPT-4 Turbo. O modelo GPT-3.5 demonstrou o melhor desempenho, com tempos de resposta bastante baixos e consistentes, concentrando-se em torno de 6 segundos. Em contrapartida, o modelo Sabiá-3 apresentou os maiores tempos de resposta, variando aproximadamente entre 33 e 36 segundos, embora com pouca variação entre as respostas. O Gemini 1.5 Flash e o GPT-4 Turbo tiveram desempenhos intermediários.

Gráfico 1: Boxplot comparativo dos tempos de resposta dos modelos LLM testados.



Fonte: Elaborada pelo autor (2025)

4.3 Avaliação quantitativa do *chatbot*

A fim de mensurar a percepção dos avaliadores em relação ao desempenho do *chatbot*, foi realizada uma análise quantitativa baseada nas respostas obtidas através de um formulário Google descrito no Apêndice A. Cada critério de avaliação contava com 15 respostas, totalizando 75 respostas ao todo. Para transformar essas respostas em uma média percentual, foram atribuídos pesos numéricos a cada categoria de resposta, de acordo com seu grau de satisfação: "Muito insatisfatório" = 0%, "Insatisfatório" = 25%, "Regular" = 50%, "Satisfatório" = 75% e "Muito satisfatório" = 100%. Em seguida, foi realizada uma média ponderada para cada critério, multiplicando o percentual de respostas em cada categoria pelo seu respectivo peso, e somando os resultados.

Os Gráficos 2 a 6 mostram o resultado do questionário aplicado para avaliação do *chatbot* aos 15 colabores da empresa:

O Gráfico 2 apresenta a satisfação no critério "Precisão das respostas", 80% dos avaliadores classificaram como "Muito satisfatório" e 20% como "Satisfatório". Foi obtido uma média de 95%, o que evidencia uma percepção bastante positiva quanto à exatidão das informações contidas nas respostas do *chatbot*.

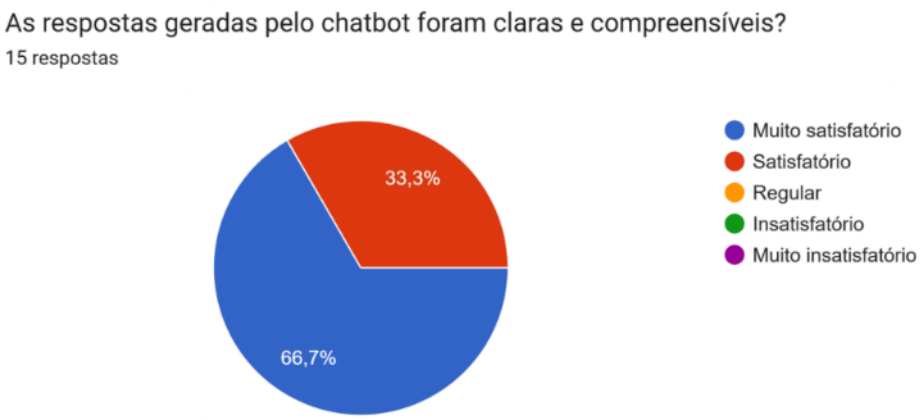
Gráfico 2: Resultado da 1ª pergunta do questionário.

Fonte: Elaborada pelo autor (2025)

Isso demonstra a qualidade na recuperação do contexto por similaridade feito no RAG, que tem exatamente o objetivo de buscar informações mais recentes e relevantes no momento da pergunta. Essas informações são muito bem interpretadas pelo modelo de linguagem escolhido, GPT-4 Turbo, que como LLM é capaz de compreender e gerar linguagem natural com alta precisão, por conta de ser treinado em grandes volumes de dados textuais e ao uso da arquitetura Transformer (Vaswani *et al.*, 2017).

No Gráfico 3, pode-se observar a avaliação em relação à "Clareza das respostas", 66,7% dos avaliadores atribuíram a nota máxima e os outros 33,3% consideraram as respostas satisfatórias. A média ponderada resultou em 91,7%, indicando que a forma como as respostas foram formuladas pelo *chatbot* foram bem compreendidas pela maioria dos avaliadores.

Gráfico 3: Resultado da 2ª pergunta do questionário.



Fonte: Elaborada pelo autor (2025)

O grande mérito desse resultado é a capacidade dos LLMs para gerar linguagem natural fluida e compreensível, simulando conversas humanas (Fröhlich; Soares, 2018). Além da capacidade dos LLMs de receber instruções para estruturação da resposta, técnica chamada de *prompt engineering*, o GPT-4 Turbo demonstra alto entendimento e adaptação as instruções dadas a ele para a geração da resposta adequada.

Verifica-se no Gráfico 4 que no quesito "Utilidade do *chatbot*", 73,3% consideraram a experiência muito satisfatória e 26,7% satisfatória. A média ponderada ficou em 93,3%, o que mostra que os usuários perceberam o *chatbot* como uma ferramenta efetiva e funcional tanto para os funcionários quanto para os clientes da empresa.

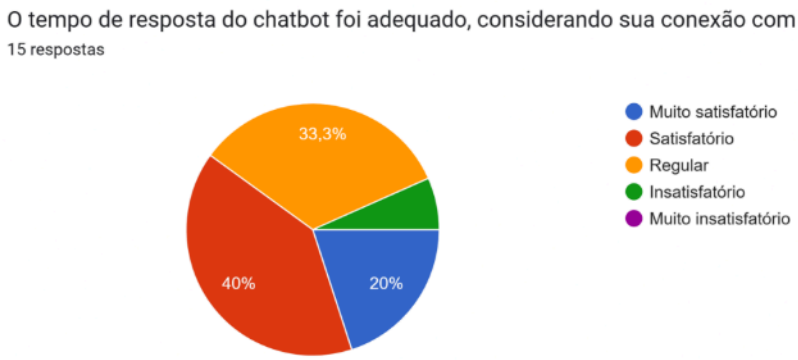
Gráfico 4: Resultado da 3ª pergunta do questionário.

Fonte: Elaborada pelo autor (2025)

Conforme Schlicht (2016) e Russell e Norvig (2016), os *chatbots* são projetados para oferecer respostas rápidas e automatizadas, o que facilita o autoatendimento, e isso para funcionários do suporte técnico e do setor de tecnologia da empresa é muito importante, pois significa uma diminuição nas demandas vindas dos clientes e também uma consulta mais rápida para solucionar problemas de incidentes de TI e tirar duvidas pessoais sobre o produto.

Conforme o Gráfico 5, o critério "Tempo de resposta" apresentou uma distribuição mais variada, 20% muito satisfatório, 40% satisfatório, 33,3% regular e 6,7% insatisfatório. Com essa diferença de avaliações, a média ponderada resultou em 68,33%, o que sugere que ainda há espaço para melhorias no tempo de entrega das respostas.

Gráfico 5: Resultado da 4ª pergunta do questionário.



Fonte: Elaborada pelo autor (2025)

Conforme avaliado e descrito na sessão de "Escolha do Modelo de Linguagem", esse resultado se atribui em grande parte pela velocidade de processamento da API do LLM utilizado. A latência na resposta de APIs de LLMs pode variar por diversos fatores técnicos e operacionais. Segundo Vaswani *et al.* (2017), modelos baseados em Transformers, como o utilizado neste trabalho, exigem grande capacidade computacional para processar grandes volumes de dados com atenção contextual. Isso significa que, quanto maior o modelo e mais complexa a entrada, ou seja, quanto mais dados complexos para ele processar, maior será o tempo de resposta. Como o *chatbot* desenvolvido neste trabalho faz utilização da técnica de Retrieval-Augmented Generation (RAG), que envolve o envio de grandes quantidades de dados como instruções de *prompt engineering*, contextos e o *prompt* do usuário para o LLM processar, ele acaba exigindo uma taxa de processamento elevada que o modelo de linguagem utilizado não tem. Uma forma de melhorar esse cenário é a escolha de um LLM com maior capacidade de processamento e infraestrutura mais robusta.

Por fim, o critério "Satisfação geral com a usabilidade" foi avaliado como muito satisfatório por 60% dos participantes e satisfatório por 40%, conforme o Gráfico 6. A média ponderada foi de 90%, evidenciando uma boa aceitação geral da experiência dos usuários na utilização do *chatbot*.

Gráfico 6: Resultado da 5ª pergunta do questionário.

Fonte: Elaborada pelo autor (2025)

Com base nas médias individuais, foi calculada uma média geral de 87,67% de satisfação entre os cinco critérios. Esse resultado demonstra que, de forma ampla, o *chatbot* foi bem avaliado pelos técnicos de suporte, com destaque para a precisão, clareza e utilidade das respostas. O único critério que apresentou desempenho menor comparado aos outros foi o tempo de resposta, o que pode direcionar futuras melhorias.

4.4 Avaliação qualitativa do *chatbot*

Além das avaliações quantitativas, o formulário incluiu um campo texto com o objetivo de obter avaliações qualitativas sobre o uso do *chatbot*. Esse campo texto permitiu que os avaliadores expressassem livremente suas opiniões, sendo elas suas experiências, elogios, sugestões ou críticas em relação ao projeto. A análise desse conteúdo qualitativo é interessante para apontar pontos fortes não capturados pelos números e evidenciar aspectos que ainda podem ser aprimorados. Portanto, são opiniões complementares que enriquecem a avaliação geral do *chatbot*, dando "voz" aos usuários.

Entre as 15 respostas, houve duas respostas neutras, onde foi respondido "N/A" e "Nada a acrescentar", que serão ignoradas nessa análise pois não enriquecem a

pesquisa. Abaixo, estão disponibilizadas as 13 respostas separadas em duas categorias, "críticas e sugestões" e "pontos fortes". Pode-se notar que a categoria de críticas e sugestões mostra que muitas vezes as críticas vieram acompanhadas de uma sugestão ou um elogio, mostrando que a avaliação dos usuários foi sincera e buscando acrescentar ao *chatbot* e não apenas a reclamação. Já os pontos fortes foram os elogios e apontamentos de qualidade ao *chatbot*, onde os usuários acharam que não era necessário acrescentar nada além de ressaltar sua satisfação.

4.4.1 Críticas e sugestões

Quadro 1: Críticas e sugestões apontadas pelos usuários

1. Utilizei para entender melhor como configurar certos pontos do Fusion, como relatórios. Seria interessante se o chatbot referenciasse qual ponto da documentação ele se baseou.
2. O chatbot não me deu um passo a passo 100% preciso, entretanto me orientou muito bem e me ajudou a localizar dentro da plataforma, que é importante. E também me passou a fórmula que eu precisava, que é o mais importante. Muito bom!
3. Um dos questionamentos que fiz a plataforma não conseguiu responder (como configurar uma tarefa de um processo para que ela caia na caixa de aprovação da Central de Tarefas). Mas ainda assim, eu acho que essa plataforma seria extremamente importante para que os clientes utilizem mais a nossa documentação...
4. Conseguiu explicar bem de acordo com a documentação e localizar corretamente o módulo, entretanto, ele demorou um pouquinho para encontrar a resposta. Como próximo passo, acredito que otimizar esse tempo seria o ideal para que esse chatbot ficasse perfeito.
5. Ficou muito bom, poderia ser padrão do manual, inclusive usando alguma LLM rodando internamente, para funcionar dentro das infraestruturas mais "LACRADAS", sem acesso à internet.
6. As respostas foram compreensíveis levando em consideração uma noção prévia do sistema. Algumas respostas ficaram um pouco genéricas... Caso seja possível esta melhoria, seria interessante que a resposta fosse mais específica.

Fonte: Elaborada pelo autor (2025)

As sugestões de melhoria destacam pontos importantes, principalmente aspectos de técnicos, como a necessidade de referências diretas à documentação, ser mais específico em algumas situações, velocidade na entrega da resposta e possibilidades de implementação local para contextos sem acesso a internet, o que é interessante se

a empresa tem mais restrições de segurança. Essas observações demonstram que os avaliadores não apenas testaram, mas também refletiram sobre como o *chatbot* poderia evoluir para atender melhor as rotinas da empresa, ajudando muito nos resultados e nas próximas melhorias.

4.4.2 Pontos Fortes

Quadro 2: Pontos fortes destacados pelos usuários.

1. Consegui usar e consultar conteúdos relacionados à documentação do Fusion.
2. O chatbot respondeu corretamente minhas perguntas, tirando todas as dúvidas necessárias e de acordo com a documentação do Fusion.
3. Muito eficaz, parabéns pelo projeto.
4. Achei o funcionamento do chatbot muito bom.
5. Ficou ótimo! Parabéns!
6. Muito bom! Facilitou muito a tirar dúvidas sobre a documentação ou alguma dúvida técnica do Fusion. Já salvei a ferramenta e pretendo usar mais vezes!
7. Sinceramente, achei sensacional. O nível de detalhe das respostas foi muito bom. Superando até algumas documentações oficiais.

Fonte: Elaborada pelo autor (2025)

Os pontos fortes evidenciam que o *chatbot* foi bem aceito e trouxe resultados positivos para os usuários, as respostas se concentraram na eficácia prática do *chatbot*, principalmente na clareza, utilidade e aderência à documentação oficial. Muitos relataram que conseguiram resolver suas dúvidas com facilidade, o que mostra que o sistema já oferece valor real ao usuário. Além disso, foi destacado que o *chatbot* já é útil no dia a dia e até melhor do que a própria documentação em alguns casos, demonstrando o impacto positivo que a ferramenta gerou no ambiente de suporte técnico.

5 Considerações finais

5.1 Considerações gerais

O presente projeto teve como objetivo principal avaliar a viabilidade de um *chatbot* desenvolvido com a técnica de RAG em um utilizando um LLM, em auxiliar clientes e técnicos de suporte no atendimento a incidentes de TI de uma empresa parceira, fornecendo respostas precisas para soluções de incidentes de TI. Isso demandou o desenvolvimento, implantação e avaliação do *chatbot* no suporte técnico da empresa parceira. Todos os objetivos específicos foram plenamente alcançados, incluindo a definição do escopo e das tecnologias a serem empregadas. Para isso, foram realizadas pesquisas, testes e análises que embasaram a escolha das ferramentas mais adequadas ao desenvolvimento do *chatbot*, garantindo uma solução robusta e eficiente. O desenvolvimento foi concluído com sucesso, resultando em um sistema construído com boas práticas de engenharia de software e plenamente alinhado com o objetivo proposto. O *chatbot* foi submetido à avaliação por 15 profissionais altamente capacitados do suporte técnico da empresa parceira. E por fim, os resultados das avaliações foram analisados e apresentados, gerando discussão e sendo de grande importância para o enriquecimento do trabalho. Todos esses objetivos foram plenamente alcançados, demonstrando a eficácia do planejamento e da execução do projeto.

Entre os resultados mais relevantes do projeto, estão os testes comparativos com diferentes modelos de *embeddings* e modelos de linguagem, que forneceram *insights* valiosos para a definição da arquitetura final do sistema e contribuíram para o fortalecimento da qualidade técnica do projeto. As avaliações, tanto quantitativas quanto qualitativas, indicaram alta aceitação da solução. Os técnicos destacaram a precisão, clareza e eficiência das respostas fornecidas pelo *chatbot*, evidenciando sua utilidade prática no apoio às atividades diárias de suporte. As sugestões de melhorias apresentadas pelos avaliadores, como a otimização do tempo de resposta e a possibilidade de funcionamento *offline*, ajudam a melhorar o projeto em trabalhos futuros.

O projeto está diretamente alinhado à área de Sistemas de Informação, com ênfase em inteligência artificial e processamento de linguagem natural, campos de grande relevância e crescimento no mercado atual. A crescente demanda por soluções automatizadas, como *chatbots*, em setores de suporte ao cliente e gestão de serviços, reforça não apenas a atualidade do tema, mas também sua aplicabilidade prática. Desse modo, o trabalho não apenas tem competências técnicas na área de formação, como também atende a uma necessidade real do mercado, mostrando que pode ser útil para um problema atual nas empresas, que é a alta demanda do suporte e a não consulta as documentações existentes.

O sucesso obtido durante o desenvolvimento e os testes do *chatbot* despertou

grande interesse por parte da empresa parceira. Como resultado, está prevista a sua produtização e implantação em ambiente de produção a partir de julho, quando passará a ser utilizado efetivamente no atendimento aos usuários e para consultas internas. Essa decisão demonstra a confiança da organização na solução desenvolvida e reforça a relevância prática do projeto, evidenciando seu potencial de gerar benefícios concretos no suporte técnico e nas operações cotidianas da empresa.

5.2 Aprendizados

O desenvolvimento do *chatbot* proporcionou um grande conhecimento, especialmente na área de inteligência artificial, que é algo que eu já tinha vontade de me aprofundar. Durante o projeto, tive a oportunidade de aprofundar meu conhecimento em técnicas de *embeddings* para representação de texto, utilização de LMM e integração de APIs. Além disso, adquiri experiência em RAG, uma abordagem atualmente bastante utilizada em diversos setores principalmente empresarial, pois permite a criação de *chatbots* inteligentes sem a necessidade de treinar um modelo de linguagem do zero, o que reduz significativamente o custo e o tempo de desenvolvimento. Também desenvolvi habilidades em *deploy* de aplicações, aprendendo a configurar ambientes para disponibilizar o *chatbot* em um contexto operacional, o que incluiu ajustes para escalabilidade da aplicação.

Os desafios enfrentados foram inúmeros e contribuíram diretamente para meu crescimento pessoal e profissional. No quesito do desenvolvimento do *chatbot*, se destacam o estudo das tecnologias para escolher as mais adequadas, os testes e ajustes feitos para definir o melhor funcionamento da aplicação, o que demandou muito tempo e busca por conhecimento.

Isso exigiu ajustes finos nos *embeddings* e testes exaustivos, o que me ensinou a importância da paciência e da experimentação iterativa. Outro desafio foi interpretar e incorporar o *feedback* qualitativo dos técnicos de suporte, que demandou habilidades de comunicação e empatia para alinhar as expectativas dos usuários com as capacidades técnicas do *chatbot*. Superar esses desafios me tornou mais resiliente e confiante na resolução de problemas complexos.

Como bagagem para a vida profissional, levo uma compreensão mais profunda sobre o ciclo completo de desenvolvimento de soluções de IA, desde a concepção da ideia e o planejamento, até a implementação e avaliação. Aprendi a importância de equilibrar a capacidade técnica com a usabilidade prática que é o objetivo principal, garantindo que a tecnologia atenda às necessidades reais dos usuários. Além disso, desenvolvi uma mentalidade analítica apurada, capaz de lidar com dados quantitativos e qualitativos. Essas competências, aliadas à experiência de trabalhar em um projeto com impacto real, serão fundamentais para enfrentar desafios futuros no mercado de

tecnologia.

5.3 Trabalhos futuros

O *chatbot* mostra que tem amplo potencial para expansão e aprimoramento, tanto em suas funcionalidades quanto na sua aplicabilidade em cenários organizacionais. Uma das melhorias potenciais seria a integração de recursos de processamento de linguagem falada, permitindo que o *chatbot* receba e interprete comandos por voz. Essa funcionalidade pode facilitar ainda mais o atendimento, proporcionando uma experiência mais natural e acessível aos usuários, além de ser um grande produto para o marketing aumentar as vendas da empresa.

Outra evolução importante seria a adoção de modelos de linguagem e técnicas de *embeddings* executados localmente. Essa abordagem permitiria que o sistema operasse mesmo em situações com conectividade de internet limitada ou instável, reduzindo a dependência de APIs externas e aumentando a autonomia da solução.

Além disso, há a possibilidade de integrar novas bases de conhecimento ao *chatbot*, como bases de registros de processos, publicações de documentos e outros repositórios institucionais. Com isso, seria possível obter informações mais detalhadas, por exemplo, identificar quais usuários abriram determinado processo, quais ações foram realizadas ou mesmo oferecer relatórios gerenciais baseados em interações históricas.

Outro aprimoramento relevante seria a utilização do *chatbot* na criação de fluxos de processos de negócios (BPM), aproveitando fluxos já existentes no sistema como base. Dessa forma, o *chatbot* poderia atuar não apenas como um assistente de suporte, mas também como um facilitador na modelagem e automatização de processos internos.

Atualmente, as principais limitações do projeto incluem a necessidade de conexão contínua com a internet para acessar os modelos e APIs externas, bem como a limitada personalização para cenários organizacionais muito específicos. Superar essas restrições envolve o desenvolvimento de uma versão robusta que opere offline e a criação de interfaces de configuração intuitivas, que permitam aos administradores personalizar o comportamento do *chatbot* sem necessidade de conhecimento técnico aprofundado.

Com a implementação dessas melhorias, o projeto tem potencial de se consolidar como uma solução ainda mais escalável, flexível e impactante, não apenas para o suporte técnico, mas também para outras áreas organizacionais que demandam atendimento inteligente, automação e apoio à decisão.

REFERÊNCIAS

ABONIZIO, H. *et al.* **Sabiá-3 Technical Report**. *arXiv preprint arXiv:2410.12049*, 2024.

ADAMOPOULOU, E.; MOUSSIADES, L. **Chatbots: History, Technology, and Applications**. *Machine Learning with Applications*, Elsevier, v. 2, p. 100006, 2020.

ALVES, E. **Um Breve Estudo dos Transformers**. 2022. https://erika-gl-alves.medium.com/um-breve-estudo-dos-transformers-6abbf1b77512. Acesso em: 21 nov. 2024.

AWS. **O que é um chatbot?** Disponível em https://aws.amazon.com/pt/what-is/chatbot/, 2024. Acesso em: 26 nov. 2024.

BOMMASANI, R. *et al.* **On the Opportunities and Risks of Foundation Models**. *arXiv preprint arXiv:2108.07258*, 2021.

BORDOLOI, S.; FITZSIMMONS, J. A.; FITZSIMMONS, M. J. **Service Management: Operations, Strategy, Information Technology**. New York: McGraw-Hill, 2019.

BRODOWICZ, M. **O Assistente Watson da IBM automatizando o suporte ao cliente para clientes bancários**. 2025. Disponível em: https://aithor.com/essay-examples/o-assistente-watson-da-ibm-automatizando-o-suporte-ao-cliente-para-clientes-bancarios. Acesso em: 5 abr. 2024.

BROWN, T. B. **Language Models are Few-shot Learners**. *arXiv preprint arXiv:2005.14165*, 2020.

CHAFFEY, D.; EDMUNDSON-BIRD, D.; HEMPHILL, T. **Digital Business and E-commerce Management**. UK: Pearson, 2019.

CHRISTOPHER, D. M.; PRABHAKAR, R.; HINRICH, S. **Introduction to Information Retrieval**. [S.l.]: Cambridge University Press, 2008.

CRUZ, L. T.; ALENCAR, A. J.; SCHMITZ, E. A. **Assistentes Virtuais Inteligentes e Chatbots**: **Um guia prático e teórico sobre como criar experiências e recordações encantadoras para os clientes da sua empresa**. Rio de Janeiro: Brasport, 2018.

DALE, R. **The Return of the Chatbots**. *Natural Language Engineering*, Cambridge University Press, v. 22, n. 5, p. 811–817, 2016.

DEERWESTER, S. *et al.* **Indexing by Latent Semantic Analysis**. *Journal of the American Society for Information Science*, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990.

DEVLIN, J. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. *arXiv preprint arXiv:1810.04805*, 2018.

FRöHLICH, L. F. G.; SOARES, V. D. **Robotização nos relacionamentos: um estudo sobre o uso de chatbots**. *Fólio – Revista Científica Digital – Jornalismo, Publicidade e Turismo*, 2018.

FøLSTAD, A.; BRANDTZæG, P. B. **Chatbots and the New World of HCI**. *Interactions*, ACM, v. 24, n. 4, p. 38–42, 2017.

GAO, Y. *et al.* **Retrieval-augmented generation for large language models: A survey**. *arXiv preprint arXiv:2312.10997*, 2023.

GARTNER. **Gartner Says More Than 80% of Enterprises Will Have Used Generative AI APIs or Deployed Generative AI-Enabled Applications by 2026**. Disponível em https://www.gartner.com/en/newsroom/press-releases/2023-10-11-gartner-says-more-than-80-percent-of-enterprises-will-have-used-generative-ai-apis-2023. Acesso em: 17 nov. 2024.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Editora Atlas, 2002.

GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. São Paulo: 6. ed. Editora Atlas SA, 2008.

GRAVES, A. **Supervised Sequence Labelling with Recurrent Neural Networks**. Berlin: Springer, 2012.

GUU, K. *et al.* **Retrieval Augmented Language Model Pre-training**. In: *Proceedings of the 37th International Conference on Machine Learning*. Online: PMLR, 2020. p. 3929–3938. Trabalho apresentado no 37th International Conference on Machine Learning – ICML, 2020.

HAKSEVER, C.; RENDER, B. **Service and Operations Management**. New Jersey: World Scientific Publishing Company, 2017.

IZACARD, G.; GRAVE, E. **Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering**. *arXiv preprint arXiv:2007.01282*, 2020.

JEONG, C. **A Study on the Implementation of Generative AI Services Using an Enterprise Data-based LLM Application Architecture**. *arXiv preprint arXiv:2309.01105*, 2023.

JURAFSKY, D. **Speech and Language Processing**. Hoboken, New Jersey: Prentice-Hall, 2000.

KARPUKHIN, V. *et al.* **Dense Passage Retrieval for Open-domain Question Answering**. *arXiv preprint arXiv:2004.04906*, 2020.

LAKATOS, E. M.; MARCONI, M. A. **Fundamentos de metodologia científica**. São Paulo: Atlas, 2017.

LAUDON, K. C.; LAUDON, J. P. **Management Information Systems: Managing the Digital Firm**. London, England: Pearson Educación, 2004.

LECUN, Y. *et al.* **Gradient-based Learning Applied to Document Recognition**. *Proceedings of the IEEE*, IEEE, New York, v. 86, n. 11, p. 2278–2324, 1998.

LEWIS, P. *et al.* **Retrieval-augmented generation for knowledge-intensive NLP tasks**. *Advances in Neural Information Processing Systems*, v. 33, p. 9459–9474, 2020.

MCCONNELL, S. **Code Complete**. London, England: Pearson Education, 2004.

MEYRELLES, T. **O que é suporte em TI?** 2019. https://www.migalhas.com.br/depeso/306640/o-que-e-suporte-em-ti. Acesso em: 11 nov. 2024.

MICROSOFT. **Insurance agency Nsure.com leverages Microsoft Power Platform and generative AI to reduce manual processes by 60%+**. 2023. https://www.microsoft.com/en-us/power-platform/blog/power-automate/insurance-agency-nsure-com-leverages-microsoft-power-platform-and-generative-ai-to-reduce-ma Acesso em: 2 nov. 2024.

MIKOLOV, T. **Efficient Estimation of Word Representations in Vector Space**. *arXiv preprint arXiv:1301.3781*, v. 3781, 2013.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. **GloVe: Global Vectors for Word Representation**. In: *Anais do Conference on Empirical Methods in Natural Language Processing*. Doha: Association for Computational Linguistics, 2014. p. 1532–1543. Trabalho apresentado no Conference on Empirical Methods in Natural Language Processing, 2014.

PERES, C. **Simplificando o Desenvolvimento de Aplicações Interativas com Streamlit na Ciência de Dados**. 2023. https://medium.com/@christianoDS/simplificando-o-desenvolvimento-de-aplica%C3%A7%C3%B5es-interativas-com-streamlit-na-ci%C3%AAncia-de-dados-a1717518f674. Acesso em: 20 fev. 2025.

PRESSMAN, R. S. **Software Engineering: a practitioner's approach**. *McGraw-Hill*, 2010.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence**: **A Modern Approach**. London, England: Pearson, 2016.

SALTON, G.; WONG, A.; YANG, C.-S. **A Vector Space Model for Automatic Indexing**. *Communications of the ACM*, ACM New York, NY, USA, v. 18, n. 11, p. 613–620, 1975.

SCHLICHT, M. **The Complete Beginner's Guide To Chatbots**. 2016. https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca. Acesso em: 22 nov. 2024.

SILVA, L. B. D. **Classificação de Personagens Animados usando Redes Neurais Convolucionais Profundas Pré-treinadas e Fine-tuning**. 2018. https://repositorio.ufc.br/handle/riufc/39481. Acesso em: 15 nov. 2024.

SOMMERVILLE, I. **IEEE Software and Professional Development**. *IEEE Software*, IEEE, v. 33, n. 2, p. 90–92, 2016.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. **BERTimbau: Pretrained BERT Models for Brazilian Portuguese**. In: SPRINGER. *Brazilian Conference on Intelligent Systems*. [S.l.], 2020. p. 403–417.

THAKUR, N. *et al.* **BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models**. *arXiv preprint arXiv:2104.08663*, 2021.

VASWANI, A. *et al.* **Attention Is All You Need**. *Advances in Neural Information Processing Systems*, v. 30, p. 6000–6010, 2017.

APÊNDICES

APÊNDICE A – QUESTIONÁRIO APLICADO AOS AVALIADORES

Questionário de Avaliação do Chatbot

Esse é um questionário avaliativo que deve ser respondido após a utilização do chatbot para fornecer um feedback sobre o projeto. Nele, deve ser levado em consideração apenas a usabilidade do chatbot, sem focar na interface de teste que está sendo utilizada.

Tabela 2 – Perguntas com escala de avaliação

Pergunta	Opções de Resposta
O chatbot respondeu corretamente às perguntas com base na documentação oficial?	Muito satisfatório Satisfatório Regular Insatisfatório Muito insatisfatório
As respostas geradas pelo chatbot foram claras e compreensíveis?	Muito satisfatório Satisfatório Regular Insatisfatório Muito insatisfatório
O chatbot demonstrou potencial para ser útil, tanto para funcionários da Neomind quanto para seus clientes?	Muito satisfatório Satisfatório Regular Insatisfatório Muito insatisfatório
O tempo de resposta do chatbot foi adequado, considerando sua conexão com a internet?	Muito satisfatório Satisfatório Regular Insatisfatório Muito insatisfatório
De forma geral, quão satisfeito(a) você está com a usabilidade do chatbot para uso diário?	Muito satisfatório Satisfatório Regular Insatisfatório Muito insatisfatório

Comentário aberto: Se desejar, descreva brevemente sua experiência com o chatbot. Fique à vontade para sugerir melhorias ou destacar pontos positivos.