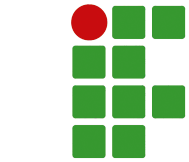


Instituto Federal de Goiás
Campus Formosa
Análise e Desenvolvimento de Sistemas
http://www.ifg.edu.br/formosa

TREMBOT: SISTEMA DE CONVERSAÇÃO AUTOMATIZADO E ESPECIALIZADO PARA O IFG *CAMPUS* FORMOSA

GIULIA CAROLINY VERA MARTINS
LETÍCIA BITTENCOURT VIEIRA

Trabalho de Conclusão de Curso

FORMOSA
2025

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[x] TCC - Graduação	[ ] Trabalho Apresentado em Evento
[ ] Produto Técnico e Educacional - Tipo: _____	

Nome Completo do Autor:
Matrícula:
Título do Trabalho:

Autorização - Marque uma das opções

1. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. (x) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data 27/01/2027 (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

( ) O documento está sujeito a registro de patente.
(x) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

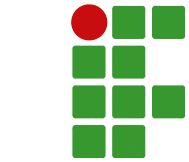
i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

gov.br Documento assinado digitalmente
GIULIA CAROLINY VERA MARTINS
Data: 04/02/2026 16:04:28-0300
Verifique em https://validar.iti.gov.br

gov.br Documento assinado digitalmente
LETICIA BITTENCOURT VIEIRA
Data: 02/02/2026 10:23:27-0300
Verifique em https://validar.iti.gov.br

Formosa , 27/01/2026.
Local Data

_____
Assinatura do Autor e/ou Detentor dos Direitos Autorais



Instituto Federal de Goiás
Campus Formosa
Análise e Desenvolvimento de Sistemas
http://www.ifg.edu.br/formosa

TREMBOT: SISTEMA DE CONVERSAÇÃO AUTOMATIZADO E ESPECIALIZADO PARA O IFG *CAMPUS* FORMOSA

Giulia Caroliny Vera Martins
Letícia Bittencourt Vieira

Trabalho de Conclusão de Curso apresentado ao Departamento de Áreas Acadêmicas do Instituto Federal de Goiás campus Formosa, como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Mario Teixeira Lemes

FORMOSA
2025

Dados Internacionais de Catalogação na Publicação

M385t

Martins, Giulia Caroliny Vera

TREMBOT: sistema de conversação automatizado e especializado para o IFG Campus Formosa / Giulia Caroliny Vera Martins, Letícia Bittencourt Vieira. — Formosa, 2026.

80 f.: il.: 30 cm.

Trabalho de conclusão de curso (Graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Câmpus Formosa, 2026.

Orientadora: Prof. Me. Mario Teixeira Lemes

1. Inteligência artificial generativa. 2. Sistemas de recuperação da informação - Educação. 3. Grupos de bate-papo pela Internet. 4. Educação – Tecnologia. I. Vieira, Letícia Bittencourt. II. Lemes, Mario Teixeira. III Título. IV. Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Câmpus Formosa.

CDD: 006.35

Bibliotecário: Jordeilson de Lana Silva CRB1-GO nº 3751

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

Formulário de Metadados para Disponibilização da Produção Técnico-Científica no ReDi IFG

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO (TCC)
MONOGRAFIA DE ESPECIALIZAÇÃO

* Preenchimento Obrigatório

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

	TCC		Monografia de Especialização

Informe o título do documento. NÃO DIGITAR EM CAIXA ALTA!	
Título: *	TREMBOT: Sistema de conversação automatizado e especializado para o IFG Campus Formosa
Informe o título alternativo. Recomenda-se preencher com a tradução do título para o inglês, para maior visibilidade do documento.	
Título Alternativo:*	TREMBOT: Automated and specialized conversation system for the IFG Campus Formosa
Permissão de acesso ao documento*	Acesso aberto (x) Acesso restrito ( ) Embargo ( )
Se o documento for de acesso restrito ou embargo, informe o motivo:	( ) O documento está sujeito a registro de patente. ( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo. ( ) Outra justificativa: _____
Caso haja restrição de acesso, indicar data para que o documento possa ser disponibilizado no ReDI.	
Data para disponibilização no ReDI:	27 /01 / 2026
Informe a data da defesa	
Data da defesa*:	14 /01 /2026

AUTOR(ES)

1	Informe o nome do(s) autores(s), conforme o formato de citação.	
	Último Nome + "Jr", Ex. Silva Último Nome: *	Vieira
	Primeiro(s) nome(s), ex. João Primeiro Nome: *	Letícia Bittencourt
	URL do Currículo Lattes:	https://lattes.cnpq.br/1191117474264635
2	Último Nome + "Jr", Ex. Silva Último Nome:	Martins
	Primeiro(s) nome(s), ex. João Primeiro Nome:	Giulia Caroliny Vera
	URL do Currículo Lattes:	

ORIENTADOR

Informe o nome do orientador, conforme o formato de citação.	
Último Nome + "Jr", Ex. Silva Último Nome: *	Lemes
Primeiro(s) nome(s), ex. João Primeiro Nome: *	Mario Teixeira
URL do Currículo Lattes: *	http://lattes.cnpq.br/4918126641251231

MEMBROS DA BANCA

1	Informe o nome do(s) membro(s) da banca, conforme o formato de citação.	
	Último Nome + "Jr", Ex. Silva Último Nome: *	Oliveira Neto
	Primeiro(s) nome(s), ex. João Primeiro Nome: *	Afrânio Furtado de
	URL do Currículo Lattes: *	http://lattes.cnpq.br/1169788371765122
2	Último Nome + "Jr", Ex. Silva Último Nome: *	Paiva
	Primeiro(s) nome(s), ex. João Primeiro Nome: *	João Ricardo Braga de
	URL do Currículo Lattes: *	http://lattes.cnpq.br/9175757340129330

DESCRIÇÃO DO TRABALHO

Informe as palavras-chave do documento descrito. Sugere-se também o uso de termos em inglês. Caso o idioma original seja inglês optar por outro idioma.	
Palavras-Chave: *	1. Generative Artificial Intelligence. 2. Conversational System. 3. Retrieval-Augmented Generation. 4. Democratization of Information.
Selecione a grande área, área do conhecimento e subárea correspondente, de acordo com tabela do CNPq.	
Áreas de conhecimento de acordo com tabela do CNPq: *	Ciências Exatas e da Terra / Ciência da Computação / Metodologia e Técnicas da Computação
Resumo do documento. Preencha o campo de acordo com o idioma do documento.	
Resumo: *	No contexto educacional, sistemas de conversação ampliam significativamente o acesso à informação ao fornecer respostas imediatas e acessíveis a toda comunidade acadêmica. Este trabalho apresenta o desenvolvimento e a avaliação do TremBot, um sistema de conversação especializado fundamentado na arquitetura de Inteligência Artificial Generativa denominada Geração Aumentada via Recuperação (RAG). O objetivo do projeto foi modernizar e democratizar o acesso à informação institucional no Instituto Federal de Goiás (IFG) Campus Formosa, oferecendo suporte dinâmico e centralizado para discentes, docentes, técnicos-administrativos e comunidade externa. A metodologia adotada compreendeu a estruturação de pipeline de Extract, Transform, Load (ETL), utilizando a ferramenta LlamaParse para ingestão de documentos oficiais em formatos complexos e o MongoDB Atlas como banco de dados vetorial para o armazenamento de embeddings. O núcleo do sistema utiliza o modelo de

	linguagem Gemini 2.5 Flash, orquestrado pelo framework LlamaIndex, com integração direta à plataforma WhatsApp para maximizar o alcance e a acessibilidade da ferramenta. Os resultados obtidos em ambiente de produção assistida, totalizando 1.059 interações, demonstraram taxa de acurácia global superior a 98% e índice de fidelidade documental de 96,8%. A análise financeira revelou viabilidade econômica da solução, com custo operacional estimado em aproximadamente US$ 22,16 para cada mil interações mensais. Conclui-se que o TremBot representa modelo eficiente de democratização da informação pública, provando que a adoção de tecnologias de Inteligência Artificial (IA) é financeiramente acessível e tecnicamente robusta no contexto educacional.

Abstract do documento. Preencha com o resumo em outro idioma.

Abstract: *	In the educational context, conversational systems significantly enhance access to information by providing immediate and accessible responses to the entire academic community. This paper presents the development and evaluation of TremBot, a specialized conversational system grounded in the generative artificial intelligence architecture known as RAG. The central objective of the project was to modernize and democratize access to institutional information at IFG Campus Formosa, offering dynamic and centralized support for students, faculty, administrative staff, and the external community. The adopted methodology involved structuring an ETL pipeline, using the LlamaParse tool to ingest official documents in complex formats, and employing MongoDB Atlas as a vector database for the storage of embeddings. The core of the system uses the Gemini 2.5 Flash language model, orchestrated by the LlamaIndex framework, with direct integration into the WhatsApp platform to maximize the reach and accessibility of the tool. Results obtained in a supervised production environment, totaling 1,059 interactions, demonstrated an overall accuracy rate above 98% and a document fidelity index of 96.8%. Financial analysis revealed the economic feasibility of the solution, with an estimated operational cost of approximately US$22.16 per thousand monthly interactions. It is concluded that TremBot represents an efficient model for the democratization of public information, demonstrating that the adoption of AI technologies is both financially accessible and technically robust in the educational context.

Referência bibliográfica do documento (como o documento deve ser citado). Use as normas de acordo com a área, por exemplo: ABNT, APA, Vancouver.

Citação: *	VIEIRA, Letícia Bittencourt; MARTINS, Giulia Caroliny Vera. *TREMBOT: Sistema de conversação automatizado e especializado para o IFG Campus Formosa*. 2026. Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Goiás, Campus Formosa, Formosa, 2026.

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Na presente data realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado **TremBot: sistema de conversação automatizado e especializado para o IFG campus Formosa**, sob orientação de Mario Teixeira Lemes , apresentado pela aluna **Giulia Caroliny Vera Martins (20231070130134)** do Curso **Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (Câmpus Formosa)**. Os trabalhos foram iniciados às **09:00** do dia **14/01/2026** pelo Professor presidente da banca examinadora, constituída pelos seguintes membros:

- **Mario Teixeira Lemes** (Presidente)
- **Afranio Furtado de Oliveira Neto** (Examinador Interno)
- **Joao Ricardo Braga de Paiva** (Examinador Interno)

A banca examinadora, tendo terminado a apresentação do conteúdo do Trabalho de Conclusão de Curso, passou à arguição da candidata. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pela aluna, tendo sido atribuído o seguinte resultado:

[X] Aprovado [ ] Reprovado Nota : 10,0

Observação / Apreciações:

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu **Mario Teixeira Lemes** lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Documento assinado eletronicamente por:

- **Mario Teixeira Lemes**, em **27/01/2026 16:20:25** com chave **3b75f24afbb511f0a1d2005056a537a4**.
- **Joao Ricardo Braga de Paiva**, em **14/01/2026 10:59:44** com chave **479d4e6af15111f09003005056a537a4**.
- **Afranio Furtado de Oliveira Neto**, em **14/01/2026 11:00:29** com chave **62513c08f15111f0b8f2005056a537a4**.

Este documento foi emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.ifg.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Ata de Projeto Final

Data da Emissão: 27/01/2026

Código de Autenticação: 91dd1f

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Na presente data realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado **TremBot: sistema de conversação automatizado e especializado para o IFG campus Formosa**, sob orientação de Mario Teixeira Lemes , apresentado pela aluna **Letícia Bittencourt Vieira (20231070130207)** do Curso **Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (Câmpus Formosa)**. Os trabalhos foram iniciados às **09:00** do dia **14/01/2026** pelo Professor presidente da banca examinadora, constituída pelos seguintes membros:

- **Mario Teixeira Lemes** (Presidente)
- **Afranio Furtado de Oliveira Neto** (Examinador Interno)
- **Joao Ricardo Braga de Paiva** (Examinador Interno)

A banca examinadora, tendo terminado a apresentação do conteúdo do Trabalho de Conclusão de Curso, passou à arguição da candidata. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pela aluna, tendo sido atribuído o seguinte resultado:

[X] Aprovado [ ] Reprovado Nota : 10,0

Observação / Apreciações:

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu **Mario Teixeira Lemes** lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Documento assinado eletronicamente por:

- **Mario Teixeira Lemes**, em **27/01/2026 16:20:16** com chave **35e9dc06fbb511f0b175005056a537a4**.
- **Joao Ricardo Braga de Paiva**, em **14/01/2026 10:59:30** com chave **3f006918f15111f083ee005056a537a4**.
- **Afranio Furtado de Oliveira Neto**, em **14/01/2026 11:00:19** com chave **5c6ebcfcf15111f09c9f005056a537a4**.

Este documento foi emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.ifg.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Ata de Projeto Final

Data da Emissão: 27/01/2026

Código de Autenticação: 838cad

Agradecimentos da Giulia

Agradeço ao meu orientador, **Prof. Me. Mario Teixeira Lemes**, pela excelência na condução deste trabalho. Agradeço pela partilha generosa de conhecimento, pelas críticas construtivas e pela confiança depositada no desenvolvimento do **`TremBot`**.

À minha mãe, **Giovana Faccin** por ter respeitado minhas vontades e sido meu porto seguro. Obrigada por me apoiar emocional e financeiramente em cada passo desta jornada universitária; você foi o alicerce indispensável para que eu pudesse concluir esta fase e sonhar com as próximas.

Aos meus tios, **Fábia e Renan Lousada** que me acolheram em sua casa e em suas vidas por tantos anos, mesmo quando o tempo superou qualquer combinado inicial. Obrigada por serem minha família cotidiana e meu refúgio enquanto estive longe da minha mãe e irmãs.

À **Marina Lousada**, minha priminha de 7 anos, que com sua pureza foi minha maior fonte de alegria e inspiração nos dias mais exaustivos.

Aos meus avós, **Terezinha e Gilberto Faccin** pelos abraços quentes e acolhedores que renovaram minhas energias e me fizeram sentir amada em cada reencontro.

Aos meus colegas e amigos, que tornaram as manhãs dos últimos três anos muito mais leves e animadas. Compartilhar essa jornada com vocês fez toda a diferença.

Ao professor **Afrânio Furtado**, por ter "segurado a barra" por mim em tantos momentos de fragilidade. Quando eu duvidava da minha própria capacidade, você acreditou no meu potencial e me incentivou a manter minha essência.

Aos meus psiquiatras e psicólogos — **Fanny Ramos, Mário Vitor e Geovanna Chaves** —, que foram fundamentais para que eu pudesse me reerguer em cada recaída. Obrigada por me ajudarem a recuperar não apenas a saúde, mas também minhas noites de sono e o equilíbrio necessário para seguir em frente.

Ao meu namorado, **Luiz Henrique**, pelo amor lindo, quente e generoso. Obrigada pela companhia constante e por saber exatamente como me animar nos momentos de tensão. Você, de tantas formas, me trouxe para a vida de novo.

E, finalmente, a mim mesma. Dedico este trabalho à **Giulia** que não desistiu, que sobreviveu aos seus invernos internos, que superou o que parecia insuperável e que, hoje, celebra a vitória de ter vencido a si mesma e aos seus desafios.

"O que a gente não entende, se resolve com amor."

— Clarice Lispector

Dedico este trabalho a mim mesma, pela resiliência e por não ter desistido de mim nos momentos de maior desafio. Aos estudantes e a toda a comunidade do IFG Campus Formosa, principais destinatários deste sistema, para que este seja apenas o início de uma jornada mais acessível e democrática à informação.

Agradecimentos da Letícia

Agradeço primeiramente a **Deus** que, em sua infinita sabedoria, guiou meus passos de forma perfeita. Hoje compreendo que nada foi por acaso.

Ao meu orientador, **Mário Lemes**, por acreditar no meu potencial mesmo quando eu mesma duvidava. Obrigada pelas horas de reuniões, pela disponibilidade constante e por me desafiar a extrair sempre o meu melhor.

Ao professor, **Afrânio Furtado**, meu eterno agradecimento. No momento mais doloroso da minha vida, quando o luto me fazia querer desistir de tudo para apoiar minha mãe e minha família, suas palavras foram o meu alicerce: "Vamos terminar essa faculdade por ele". Essa frase me deu forças nos dias mais difíceis e foi o combustível para que eu realizasse este trabalho. Obrigada por ser um mestre além da sala de aula, dividindo conosco seus conselhos e alegrando nossas manhãs com seu carisma e suas brincadeiras.

À minha mãe, **Lindalva Bittencourt**, o meu maior exemplo de força e amor. Sem o seu apoio incondicional, eu não teria chegado até aqui. Obrigada por me socorrer em todos os momentos — inclusive na produção dos doces que vendo, permitindo que eu dedicasse o tempo necessário para superar os imprevistos deste projeto. Sua ajuda foi crucial para que eu conciliasse o trabalho como bolsista, as vendas e a vida acadêmica.

Ao meu esposo, **Mathews Rezende**, pelo incentivo diário e por ser meu porto seguro. Em meio às dificuldades, você foi quem me lembrou de que eu era capaz de superar este desafio. Sou grata pelo suporte em nossa rotina e no auxílio com as vendas, o que foi essencial para que eu pudesse focar neste projeto. Sem o seu amor e dedicação, esta jornada seria muito mais árdua.

Ao meu amigo **Matheus Rodrigues**, que foi o responsável por eu estar neste curso. Obrigada por me incentivar a fazer a inscrição e pelo auxílio com as imagens me dando *feedbacks* para que elas ficassem da melhor maneira. Obrigada, acima de tudo, pela amizade genuína e por ser o ouvinte nos meus momentos de maior ansiedade e desabafo.

Ao meu amigo **Juliano Schaurich**, pela ajuda no *design* e na escolha da paleta de cores, trazendo harmonia visual a este trabalho.

Aos meus colegas de turma e à minha família, que tornaram o fardo mais leve. Vocês transformaram os dias difíceis em momentos de aprendizado e amizade, sendo a rede de apoio necessária para que eu não desistisse.

Por fim, dedico este trabalho à memória do meu pai, **Marcos Antônio**, o grande incentivador para que eu iniciasse esta jornada. Embora sua partida tenha deixado um vazio imenso, foi o seu incentivo inicial e o desejo de honrar sua história que me trouxeram até aqui. Finalizo este ciclo com a certeza de que honrei seu desejo e seu apoio.

"Nunca se esqueça de sorrir, pois o dia em que você não sorrir será um dia perdido."
— Charlie Chaplin

Dedico este trabalho à memória do meu pai, por ter sido o meu maior incentivador. À minha mãe, por todo o sacrifício e amor que tornaram este caminho possível e ao meu esposo, por caminhar ao meu lado e me lembrar constantemente do meu propósito.

Resumo

No contexto educacional, sistemas de conversação ampliam significativamente o acesso à informação ao fornecer respostas imediatas e acessíveis a toda comunidade acadêmica. Este trabalho apresenta o desenvolvimento e a avaliação do **`TremBot`**, um sistema de conversação especializado fundamentado na arquitetura de Inteligência Artificial Generativa denominada Geração Aumentada via Recuperação (RAG). O objetivo do projeto foi modernizar e democratizar o acesso à informação institucional no Instituto Federal de Goiás (IFG) *Campus* Formosa, oferecendo suporte dinâmico e centralizado para discentes, docentes, técnicos-administrativos e comunidade externa. A metodologia adotada compreendeu a estruturação de *pipeline* de *Extract, Transform, Load* (ETL), utilizando a ferramenta *LlamaParse* para ingestão de documentos oficiais em formatos complexos e o MongoDB Atlas como banco de dados vetorial para o armazenamento de *embeddings*. O núcleo do sistema utiliza o modelo de linguagem **Gemini 2.5 Flash**, orquestrado pelo *framework LlamaIndex*, com integração direta à plataforma *WhatsApp* para maximizar o alcance e a acessibilidade da ferramenta. Os resultados obtidos em ambiente de produção assistida, totalizando 1.059 interações, demonstraram taxa de acurácia global superior a 98% e índice de fidelidade documental de 96,8%. A análise financeira revelou viabilidade econômica da solução, com custo operacional estimado em aproximadamente US$ 22,16 para cada mil interações mensais. Conclui-se que o **`TremBot`** representa modelo eficiente de democratização da informação pública, provando que a adoção de tecnologias de Inteligência Artificial (IA) é financeiramente acessível e tecnicamente robusta no contexto educacional.

Palavras-chave: Inteligência Artificial Generativa. Sistema de Conversação. Geração Aumentada por Recuperação. Democratização da Informação.

Abstract

In the educational context, conversational systems significantly enhance access to information by providing immediate and accessible responses to the entire academic community. This paper presents the development and evaluation of TremBot, a specialized conversational system grounded in the generative artificial intelligence architecture known as RAG. The central objective of the project was to modernize and democratize access to institutional information at IFG Campus Formosa, offering dynamic and centralized support for students, faculty, administrative staff, and the external community. The adopted methodology involved structuring an ETL pipeline, using the LlamaParse tool to ingest official documents in complex formats, and employing MongoDB Atlas as a vector database for the storage of embeddings. The core of the system uses the Gemini 2.5 Flash language model, orchestrated by the LlamaIndex framework, with direct integration into the WhatsApp platform to maximize the reach and accessibility of the tool. Results obtained in a supervised production environment, totaling 1,059 interactions, demonstrated an overall accuracy rate above 98% and a document fidelity index of 96.8%. Financial analysis revealed the economic feasibility of the solution, with an estimated operational cost of approximately US$22.16 per thousand monthly interactions. It is concluded that TremBot represents an efficient model for the democratization of public information, demonstrating that the adoption of AI technologies is both financially accessible and technically robust in the educational context.

Keywords: Generative Artificial Intelligence. Conversational System. Retrieval-Augmented Generation. Democratization of Information.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

Sumário

1

Introdução

Os *chatbots* surgiram como solução para melhorar a interação entre humanos e máquinas, promovendo diálogos em linguagem natural que simulam conversas humanas (Tsivitanidou; Ioannou, 2020). Esses sistemas funcionam como assistentes automatizados e personalizados McTear; Callejas (2020), capazes de interagir com usuários de maneira eficiente e natural. Sua automatização é viabilizada pela capacidade de estabelecer diálogos e executar ações com base em diretrizes pré-definidas ou em técnicas de Inteligência Artificial. Seu caráter personalizável, por sua flexibilidade e capacidade de adaptação, resulta na aplicabilidade multissetorial, expressa em empresas, *e-commerce*, redes sociais, busca de informações, e educação (McTear; Callejas, 2020).

No contexto educacional, a implementação de *chatbots* demonstra grande potencial no ensino, atuando como ferramentas de apoio acadêmico e administrativo (Sousa; Fecchio; Corrêa, 2021). No entanto, muitos desses sistemas ainda se concentram exclusivamente em atender às demandas de discentes, o que limita sua aplicabilidade. Para maximizar o impacto dessa tecnologia, é fundamental expandir suas funcionalidades, incorporando soluções que atendam também às necessidades de docentes e técnicos administrativos, ampliando, assim, seu alcance e efetividade dentro das Instituições (Tsivitanidou; Ioannou, 2020).

O Instituto Federal de Goiás (IFG), enquanto Instituição pública de ensino, tem como missão não apenas oferecer educação formal, mas também promover o desenvolvimento regional por meio da integração de Ensino, Pesquisa e Extensão. Previsto no Plano de Desenvolvimento Institucional (PDI), sua estrutura baseia-se na indissociabilidade deste tripé e fundamenta-se em atender às demandas e necessidades regionais (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2018a). O ensino no IFG é pautado pela oferta de cursos técnicos de nível médio, graduação, pós-graduação lato e stricto sensu, além de formações iniciais e continuadas. Com um enfoque interdisciplinar e pluricurricular, a Instituição busca criar tempos e espaços de planejamento coletivo que fomentem o diálogo entre as áreas do conhecimento.

A pesquisa, como um dos pilares da formação, é o centro de produção de saberes, produtos ou serviços (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2018a). Promove-se a articulação ensino-pesquisa por meio da sensibilização acadêmica e pela criação

de condições físicas e materiais, onde docentes e técnicos-administrativos não são presos às salas de aula ou dependências e os alunos são fomentados a buscar oportunidades/problemas. O PDI destaca a importância de democratizar e desburocratizar o acesso à pesquisa, permitindo que toda comunidade acadêmica, docentes, técnicos-administrativos e discentes, atue como agente criador de conhecimento (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2018a). Essa integração incentiva a busca por oportunidades e soluções para problemas locais e regionais.

A extensão no IFG é o meio pelo qual saberes acadêmicos são entregues à sociedade. Ela promove interação entre Instituição e comunidade, conectando produção de conhecimento às realidades locais. Por reflexão e confronto com a realidade e suas demandas, reelabora os saberes produzidos. Extensão é o espaço onde a pesquisa e inovação, agentes transformadores de realidade, podem chegar à sociedade. Por meio de projetos, eventos e ações, a extensão fomenta a vida acadêmica, transforma demandas em inovação e contribui para a democratização do acesso ao conhecimento e à cidadania (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2018b).

Diante de sua missão de promover ensino, pesquisa e extensão, o IFG busca constantemente alinhar-se às demandas tecnológicas para aprimorar seus processos e atender à comunidade acadêmica. A implementação de um *chatbot* representa oportunidade de fortalecer a integração entre os pilares institucionais, ao oferecer suporte dinâmico e acessível. Além disso, essa ferramenta tecnológica pode contribuir para a democratização do acesso às informações, promovendo inclusão e acessibilidade (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2023).

A implementação de um sistema de conversação em contexto educacional oferece vantagens práticas significativas. Com disponibilidade ininterrupta, os *chatbots* eliminam limitações de horário de atendimento, permitindo suporte contínuo que reduz o tempo de espera e atende simultaneamente múltiplos usuários. Essa capacidade alivia a sobrecarga no antedimento ao lidar com perguntas frequentes, otimizando o trabalho interno e liberando os profissionais para exercerem outras atividades (Anaya; Braizat; Al-Ani, 2024). Adicionalmente, *chatbots* ajudam centralizar informações dispersas em diferentes plataformas, como sites e murais, tornando o acesso mais ágil e intuitivo. A integração com ferramentas populares, como o *WhatsApp*, aumenta o alcance e engajamento, uma vez que usuários tendem a preferir tecnologias com as quais já estão familiarizados. Essa acessibilidade promove não apenas eficiência operacional, mas também redução de custos, ao diminuir a necessidade de atendimentos humanos para interações simples.

De forma complementar, a introdução de *chatbot* no ambiente acadêmico reflete a busca pela inovação e modernização institucional, fortalecendo a imagem tecnológica do IFG (Instituto Federal de Educação, Ciência e Tecnologia de Goiás, 2023). A coleta de dados gerada pelo sistema, como dúvidas recorrentes e/ou *feedbacks* de usuários, fornece base valiosa para decisões estratégicas e para o aprimoramento dos serviços prestados. Assim, ao garantir o direito de acesso à informação de maneira inclusiva, o *chatbot* contribui para uma experiência mais satisfatória, rápida e eficaz para todos os usuários, consolidando o papel da tecnologia como aliada na

transformação do ensino, pesquisa e extensão.

Como parte desse esforço de modernização e alinhamento tecnológico, este Trabalho de Conclusão de Curso (TCC) aborda a concepção e construção de um sistema de conversação, denominado **`TremBot`**, para elucidar dúvidas específicas da comunidade acadêmica do IFG *campus* Formosa. A integração do sistema de perguntas e respostas a um grande modelo de linguagem, aliado ao emprego de técnicas de indexação de documentos específicos, proporcionam respostas eficientes, não somente às questões da comunidade interna, professores/as, técnicos/as administrativos/as, alunos/as, mas também à comunidade externa, a respeito de diferentes atividades executadas pelo Instituto. A pesquisa enfatiza o emprego da tecnologia de Inteligência Artificial (IA) como fator chave no suporte acadêmico e democratização de informações contidas em documentos produzidos por diferentes setores da Instituição.

Este TCC está dividido em 6 capítulos. O Capítulo 2 apresenta conceitos inerentes às tecnologias e paradigmas de Inteligência Artificial necessárias para a contextualização técnica do sistema de conversação, incluindo *Language Large Model* (LLM)s e a Arquitetura Geração Aumentada via Recuperação (RAG), e também apresenta uma discussão de trabalhos correlatos. Os materiais e métodos utilizados para execução do projeto, abrangendo o *stack* tecnológico e definição e descrição das fases de desenvolvimento são detalhados no Capítulo 3. O Capítulo 4 apresenta documentação técnica da solução, detalhando arquitetura, requisitos e modelagem do sistema. A avaliação do protótipo, que inclui testes iterativos de acurácia e avaliação com *stakeholders*, é apresentada no Capítulo 5. Por fim, o Capítulo 6 expõe a conclusão e apresenta recomendações para trabalhos futuros.

2

Fundamentação Teórica

Neste Capítulo, apresentamos os principais conceitos que embasam o desenvolvimento do *chatbot*. Primeiramente, são definidos os principais termos e princípios teóricos relacionados a rede neural artificial, inteligência artificial, aprendizado de máquina, grandes modelos de linguagem, dentre outros. Em seguida, exploramos a evolução dos *chatbots* ao longo da história, destacando marcos importantes no desenvolvimento dessas tecnologias. Finalmente, são apresentados e discutidos trabalhos correlatos à esta pesquisa, o que permite comparar os diferenciais tecnológicos providos pelo sistema de conversação automatizado e especializado para o IFG Câmpus Formosa.

2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), ilustradas na Figura 2.1, são modelos computacionais capazes de resolver problemas complexos que envolvem reconhecimento de padrões, classificação, regressão e predição de dados. São compostas por unidades fundamentais chamadas neurônios artificiais, organizadas em camadas, que são interconectadas e estruturadas em três níveis: (i) **camada de entrada**, que recebe os dados brutos (E1, E2, E3 e E4), (ii) **camada oculta**, responsável pelo processamento das informações e aplicação dos pesos (P1, P2, P3 e P4), e (iii) **camada de saída**, que gera os resultados finais (S1 e S2).

A interação entre essas camadas possibilita que a rede aprenda a partir de dados e melhore seu desempenho conforme recebe mais exemplos, ajustando os pesos sinápticos, que são coeficientes numéricos atribuídos às conexões entre os neurônios. Esses pesos determinam a influência de cada entrada na ativação do neurônio seguinte, sendo atualizados durante o processo de aprendizado para minimizar erros e otimizar as previsões do modelo (Uzair; Jamil, 2020).

O neurônio artificial é uma abstração do neurônio biológico, projetado para simular sua capacidade de processar e transmitir informações. Funciona como uma unidade básica em redes neurais artificiais, onde cada neurônio recebe múltiplos sinais de entrada, que são ponderados por coeficientes (pesos) que representam a força das conexões sinápticas em um cérebro biológico (Sharma; Rai; Dev, 2012). Esses sinais ponderados são somados e passam por uma função de ativação, que determina o sinal de saída do neurônio. Essa abordagem permite que

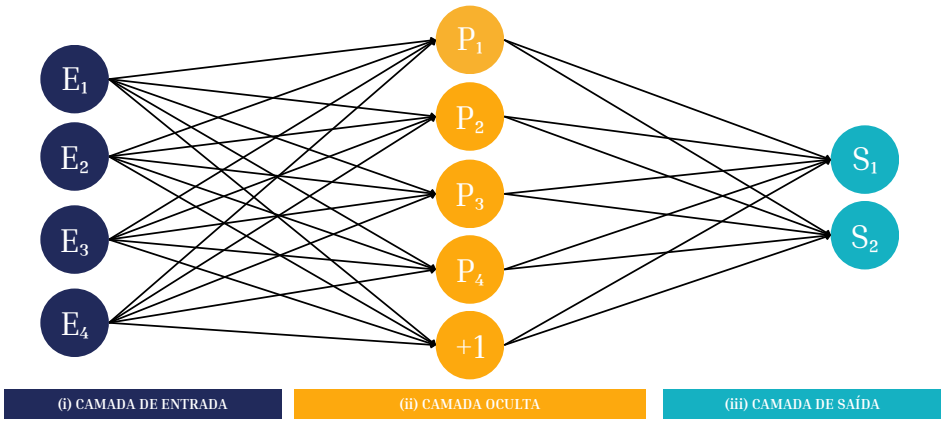


Figura 2.1: Representação de rede neural simples, composta pelas camadas de entrada, oculta e de saída.

redes neurais aprendam e tomem decisões de forma adaptativa, ajustando seus pesos sinápticos ao longo do treinamento (Kireev et al., 2022).

Os neurônios biológicos, representados na Figura 2.2, embora operem de maneira mais lenta e menos eficiente individualmente, existem aos bilhões no cérebro humano, e operam em uma rede extremamente complexa e interconectada que permite vasta gama de funcionalidades, incluindo a capacidade de aprender e adaptar-se de maneira eficiente com baixo consumo energético (Goodfellow; Bengio; Courville, 2024).

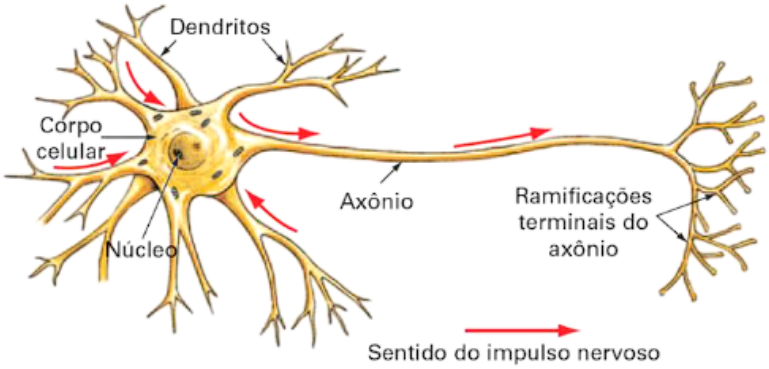


Figura 2.2: Representação de neurônio biológico.

Em contraste, neurônios artificiais, exibidos na Figura 2.3, embora mais rápidos e eficientes em tarefas específicas, possuem estrutura mais simples e limitada. Um neurônio artificial recebe múltiplas entradas (E1, E2, ..., En), cada uma associada a um peso (P1, P2, ..., Pn). Essas entradas são processadas em um somatório, que combina os valores ponderados e adiciona uma bias B para ajustar o resultado. Em seguida, a saída do somatório passa por uma função de ativação F, responsável por determinar se o neurônio será ativado, gerando um sinal de saída S.

Embora as redes neurais artificiais tenham avançado, sua estrutura ainda não se compara

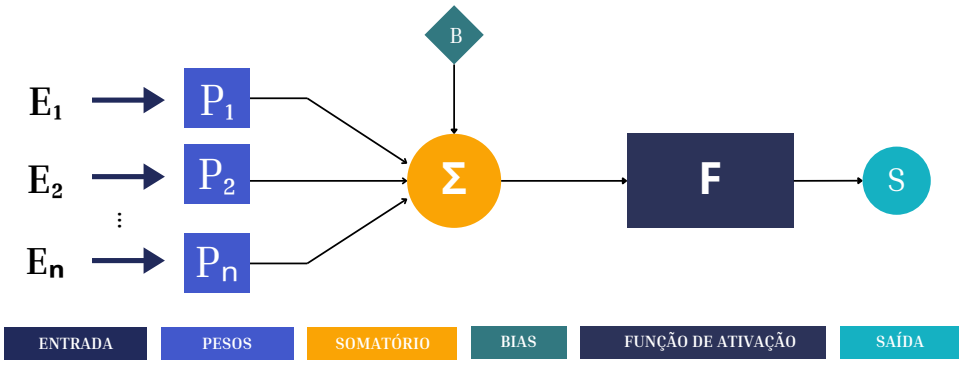


Figura 2.3: Representação de neurônio artificial.

à complexidade do cérebro humano. O alto custo computacional e energético necessário para operar e treinar milhões de neurônios artificiais torna inviável a replicação exata das funções do cérebro, reforçando as diferenças fundamentais entre os dois modelos (Tripp et al., 2024).

2.1.1 *Sequence-to-Sequence* e Arquitetura *Transformers*

O modelo *Sequence-to-Sequence* (Seq2Seq), desenvolvido em 2014, foi projetado para transformar uma **sequência de entrada** em uma **sequência de saída** de comprimento variável, sendo amplamente aplicado em tarefas como tradução automática e geração de respostas em *chatbots* (Sutskever, 2014). É composto por dois blocos: (i) **codificador**, que recebe uma sequência de palavras e a converte em um **vetor de representação compacta**, e o (ii) **decodificador**, que utiliza essa representação para gerar uma nova sequência de texto. A Figura 2.4 exibe uma representação desta arquitetura..

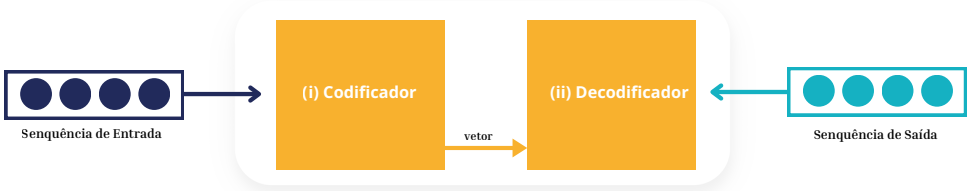


Figura 2.4: Representação do modelo *Sequence-to-Sequence.*

Além da tradução automática, o modelo Seq2Seq é amplamente empregado em *chatbots*, permitindo interações mais naturais e dinâmicas. Também é aplicado na síntese de fala, convertendo textos em áudio fluido e inteligível, contribuindo para avanços em tecnologias assistivas e a acessibilidade para pessoas com deficiência visual.

Transformers são modelos que revolucionaram o Processamento de Linguagem Natural (PLN) ao introduzirem mecanismo de "autoatenção" que permite análise simultânea de todas as palavras em uma sentença (Ramirez et al., 2024). Diferente das arquiteturas Seq2Seq, baseadas em redes recorrentes, os *Transformers* eliminam limitações de processamento sequencial, permitindo treinamento de modelos maiores com maior eficiência computacional. Esta diferença é exibida na Figura 2.5.

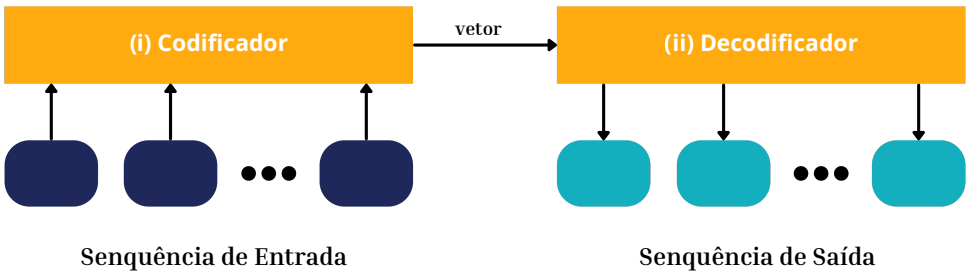


Figura 2.5: Arquitetura *Transformers* e o mecanismo de "atenção"

.

A estrutura dos *Transformers* é composta por múltiplas camadas de (i) **codificadores**, responsáveis por processar a **sequência de entrada**, e (ii) **decodificadores**, que geram a **sequência de saída**. A comunicação entre essas camadas ocorre por meio de um **vetor**, que transporta a representação da entrada para a saída.

2.2 Inteligência Artificial e Aprendizado de Máquina

Russell; Norvig (2016) definem Inteligência Artificial (IA) como subárea da Ciência da Computação dedicada a criar sistemas que realizem tarefas que, se realizadas por humanos, exigiriam inteligência. Para Nilsson (2009), a IA permite projeto de sistemas que exibem características dos comportamentos inteligentes. Esses comportamentos incluem aprendizado, raciocínio, resolução de problemas, percepção e uso da linguagem. A Figura 2.6 mostra a relação entre IA e seus principais paradigmas: **(ii) Aprendizado de Máquina**, **(iii) Aprendizado Profundo**, e **(iv) IA Generativa**.

O aprendizado de máquina ou *Machine Learning* (ML) é uma subárea da IA voltada ao desenvolvimento de técnicas computacionais capazes de adquirir conhecimento de forma automática, a partir de experiências acumuladas na solução de problemas anteriores (Monard; Baranauskas, 2003). Diferente dos métodos tradicionais, não requer programação explícita para cada tarefa, pois seus algoritmos analisam grandes volumes de dados para identificar padrões e construir modelos. Esses modelos podem ser aplicados a novos conjuntos de informações, permitindo predições, classificações ou decisões de forma automatizada.

Nesse contexto, existem três abordagens: (i) **aprendizado supervisionado** ou *Supervised Learning* (SL), que utiliza dados rotulados para ensinar os algoritmos a fazer associações entre

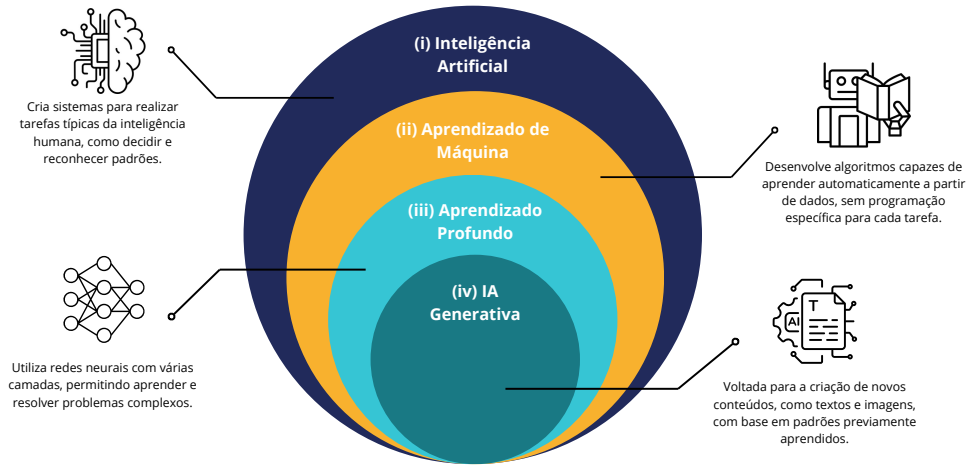


Figura 2.6: Representação das áreas que compõem o paradigma de Inteligência Artificial.

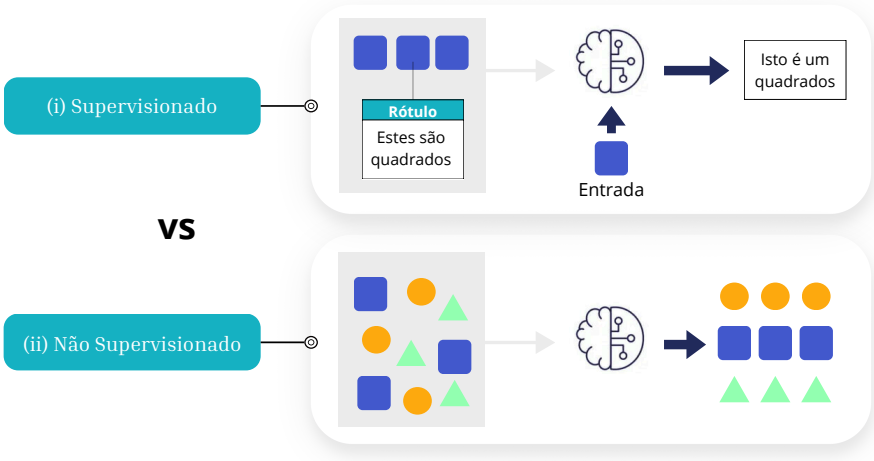


Figura 2.7: Comparação entre aprendizado supervisionado e aprendizado não supervisionado.

entradas e saídas; o (ii) **aprendizado não supervisionado** ou *Unsupervised Learning* (UL), que analisa dados não rotulados para identificar agrupamentos ou padrões ocultos; e **aprendizado por reforço** ou *Reinforcement Learning* (RL), no qual o sistema aprende por tentativa e erro, sendo recompensado por decisões corretas e penalizado por erros. A Figura 2.7 mostra a comparação entre duas dessas abordagens, o aprendizado supervisionado e o não supervisionado.

O aprendizado profundo ou *Deep Learning* (DL) é um subárea do aprendizado de máquina que utiliza redes neurais artificiais formadas por várias camadas para modelar e analisar dados complexos. Segundo LeCun; Bengio; Hinton (2015), essa tecnologia destaca-se por sua capacidade de automatizar a extração de características diretamente dos dados brutos, eliminando a necessidade de engenharia manual de atributos. Uma das arquiteturas utilizadas no aprendizado profundo são as redes neurais profundas, que possuem múltiplas camadas ocultas para capturar padrões complexos nos dados. Como ilustrado na Figura 2.8, essas redes são compostas por uma (i) **camada de entrada**, que recebe os dados iniciais (E1, E2, E3, E4), uma ou mais (ii) **camadas ocultas**, responsáveis pelo processamento dos dados e aplicação de pesos, e uma (iii) **camada de saída**, que gera os resultados finais (S1, S2).

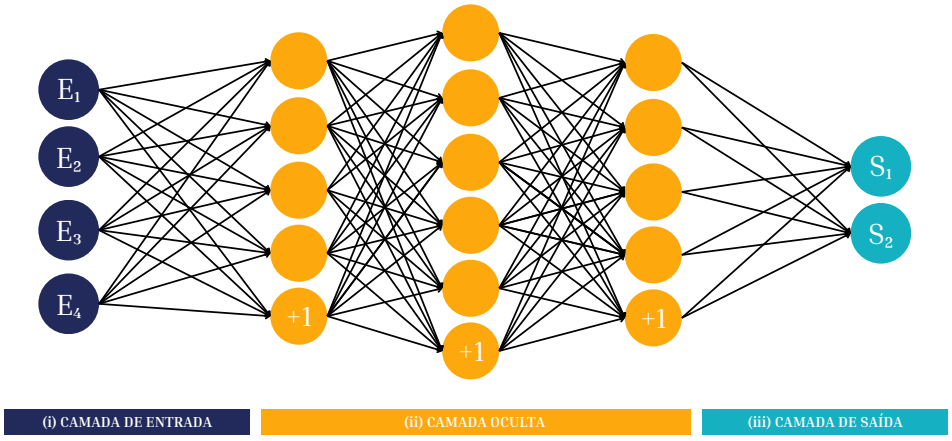


Figura 2.8: Representação de rede neural profunda, composta pelas camadas de entrada, oculta e saída.

As redes neurais profundas são capazes de discernir padrões em variados níveis de abstração, tornando-as eficientes em aplicações como reconhecimento de padrões, classificação de imagens, tradução de línguas e processamento de linguagem falada. Devido sua versatilidade em resolver problemas complexos e impulsionada por avanços significativos em *hardware* e pela disponibilidade de grandes volumes de dados, a tecnologia de aprendizado profundo tem sido empregada em diversos setores, consolidando-se como uma das principais ferramentas no desenvolvimento de sistemas inteligentes contemporâneos.

2.2.1 Inteligência Artificial Generativa e Processamento de Linguagem Natural

A Inteligência Artificial Generativa (Gen-AI) é uma subárea da IA dedicada à criação de conteúdos novos e originais, como textos, imagens, áudios e vídeos, a partir de padrões extraídos

de dados existentes. Essa tecnologia opera de forma autônoma, sem necessidade de intervenção humana direta, utilizando técnicas avançadas de aprendizado de máquina, com destaque para o aprendizado profundo (Kalota, 2024). Por meio de modelos compostos por múltiplas camadas de processamento, a Gen-AI é capaz de aprender representações complexas em diferentes níveis de abstração, permitindo a geração de novos conteúdos (Rusk, 2016). Entre suas características está a capacidade de criar *outputs* que extrapolam os padrões aprendidos, como gerar imagens inéditas ou gerar vozes sintéticas realistas para dublagem e assistentes virtuais. Suas aplicações abrangem áreas como *chatbots* avançados, geração de arte digital, síntese de fala e ferramentas de criação assistida, desempenhando papel transformador em setores como entretenimento, *marketing* e *design*.

O PLN ou *Natural Language Processing* (NLP) é uma subárea da IA que tem como objetivo principal permitir que máquinas compreendam, interpretem e gerem linguagem humana de forma significativa. Essa tecnologia combina conceitos de linguística computacional e aprendizado de máquina, possibilitando que sistemas processem e analisem grandes volumes de dados textuais e não textuais. Suas aplicações abrangem desde tradução automática e análise de sentimentos até reconhecimento de voz e extração de informações.

2.3 Grandes Modelos de Linguagem, Ajuste Fine e Indexação

Os grandes modelos de linguagem ou *Language Large Model* (LLM) são sistemas de IA projetados para processar, compreender e gerar textos em linguagem natural. Baseados em redes neurais profundas, principalmente arquiteturas baseadas em transformadores, para analisar grandes volumes de dados textuais e identificar padrões linguísticos. O treinamento dos LLMs ocorre por meio da exposição a grandes volumes de dados textuais, possibilitando a construção de representações profundas do significado e estrutura da linguagem. Esses modelos tem a capacidade de exibir habilidades emergentes, ou seja, desempenhar tarefas para as quais não foram explicitamente programados, mas que surgem como resultado do treinamento em larga escala (Zhao et al., 2024). Seu funcionamento baseia-se na predição probabilística das palavras mais adequadas em um determinado contexto, tornando as respostas mais coerentes e contextualmente relevantes.

O Ajuste Fino é um processo de refinamento de modelos de IA previamente treinados, permitindo sua adaptação para tarefas específicas. Essa técnica consiste em ajustar um modelo de linguagem de larga escala utilizando um conjunto de dados especializado, de modo a otimizar seu desempenho em domínio específico. Durante esse processo, os pesos da rede neural são atualizados de forma controlada, preservando o conhecimento adquirido no treinamento inicial, mas refinando a capacidade do modelo para responder de maneira mais precisa e contextualizada dentro da nova aplicação (Lialin; Deshpande; Rumshisky, 2023).

A indexação é um processo para organização e recuperação eficiente de informações em grandes volumes de dados. Consiste na estruturação e catalogação de conteúdos para facilitar

sua busca e acesso, tornando a recuperação mais rápida e precisa. No contexto deste trabalho, a indexação será aplicada para estruturar documentos institucionais e permitir que o *chatbot* localize informações relevantes de forma eficiente. Técnicas de indexação, como a utilização de vetores de palavras e modelos de busca semântica, podem ser empregadas para melhorar a correspondência entre as consultas dos usuários e os documentos disponíveis. Dessa forma, a implementação de um sistema de indexação adequado contribuirá para a precisão das respostas geradas pelo *chatbot*, garantindo que o conteúdo recuperado esteja alinhado com as necessidades da comunidade acadêmica.

2.4 *Chatbots*

Os *chatbots* são programas de computador desenvolvidos para simular interações humanas por meio de mensagens de texto ou voz. Segundo Adamopoulou; Moussiades (2020a), os *chatbots* desempenham papel fundamental em diversos setores, automatizando tarefas e otimizando interações humanas com sistemas computacionais. Esses sistemas podem ser implementados de diferentes maneiras, variando desde fluxos de decisão baseados em regras até modelos generativos que utilizam redes neurais avançadas para criar respostas dinâmicas e adaptáveis.

Chatbots baseados em regras figuram a mais simples estrutura de *chatbots*. Dependentes de um conjunto de regras predefinidas e uma base de conhecimento limitada construída manualmente, funcionam com base em árvores de decisão, onde cada entrada do usuário corresponde a uma resposta específica, seguindo fluxo estruturado, sem capacidade de aprender ou adaptar-se a novas situações. Embora sejam eficientes para interações simples, *chatbots* baseados em regras apresentam limitações, pois não conseguem lidar com contextos complexos ou gerar respostas novas, o que reduz sua adaptabilidade (Adamopoulou; Moussiades, 2020b).

Chatbots baseados em recuperação de informação utilizam técnicas de PLN para identificar a intenção do usuário e selecionar a resposta mais adequada a partir de um banco de dados preexistente. Diferentemente dos sistemas baseados em regras, esses *chatbots* não geram respostas novas, mas recuperam as mais relevantes com base em padrões de linguagem e contextos previamente armazenados, utilizando técnicas avançadas de busca e classificação, como aprendizado supervisionado, para selecionar respostas a partir de um conjunto pré-definido. Adamopoulou; Moussiades (2020b) destacam que esses sistemas conseguem manter contextualização ao longo de uma conversa. Contudo, sua eficácia depende da qualidade e da extensão do banco de dados, o que pode limitar sua capacidade de lidar com consultas específicas ou inéditas.

Chatbots baseados em geração utilizam modelos de aprendizado de máquina, como redes neurais profundas, para processar dados de entrada e produzir respostas originais. Esses sistemas são projetados por meio do uso de técnicas de PLN e aprendizado profundo, o que possibilita geração de linguagem natural de maneira dinâmica, considerando o contexto das interações. Adamopoulou; Moussiades (2020b) apontam que essa abordagem permite que os

chatbots compreendam e respondam a entradas de forma mais flexível, sem depender de padrões ou respostas predefinidas. Essa metodologia requer grandes volumes de dados para treinamento e utiliza algoritmos que analisam sequências textuais, como o Seq2Seq ou o *Transformers*, permitindo a construção de diálogos consistentes ao longo de múltiplas interações.

Chatbot que consideram a abordagem Geração Aumentada via Recuperação (RAG) combinam técnicas de recuperação de informação e geração de respostas por inteligência artificial, permitindo consultas a uma base de conhecimento antes de formular a resposta. Diferentemente das abordagens exclusivamente baseadas em regras ou recuperação de conteúdo pré-definido, essa abordagem aprimora a precisão e a flexibilidade do sistema ao recuperar conteúdos relevantes de documentos indexados e utilizá-los na geração de respostas contextualizadas. O processo ocorre em duas etapas: primeiro, a recuperação de informação busca conteúdos relevantes em documentos indexados, e, em seguida, a inteligência artificial generativa processa esses dados para elaborar respostas adaptadas às consultas dos usuários. Essa integração amplia a flexibilidade do sistema, tornando-o capaz de fornecer respostas embasadas em fontes verificáveis e ajustadas às necessidades específicas da interação (Lewis et al., 2020).

2.5 Contextualização Histórica

A contextualização histórica deste estudo percorre a evolução dos *chatbots*, desde a concepção da ideia de máquinas pensantes com Alan Turing até os modelos mais sofisticados da atualidade. Inicialmente, são abordados os *chatbots* pioneiros, que operavam com regras explícitas, seguidos pelos avanços proporcionados pela recuperação de informação e pelo uso de redes neurais. Por fim, explora-se a transição para os LLMs, que marcaram uma nova era na interação humano-máquina, culminando em abordagens mais seguras e refinadas para a geração de respostas.

2.5.1 Teste de Turing

A ideia mais antiga relacionada a *chatbots* é o teste de Turing e seu “jogo da imitação". Turing acreditava que, com recursos computacionais e um bom *software*, seria possível replicar comportamentos humanos. Ele inicia sua argumentação questionando a fragilidade da concepção de pensamento ao perguntar: “Máquinas podem pensar?". O jogo da imitação sugere uma dinâmica que explora essa questão. Nele, quando uma pessoa é colocada para conversar com duas entidades diferentes, um humano e uma máquina, e fosse indistinguível sua identidade, essa máquina seria passível de pensamento. Por meio do jogo, fatores emocionais e físicos são intencionalmente descartados, restando apenas o aspecto intelectual, afim de reforçar a ideia de que comportamentos humanos podem ser replicados em determinados contextos. Essa abordagem pragmática, com a intelectualidade como sinônimo de pensamento, conecta-se à concepção atual de *chatbots*, que frequentemente simulam conversas humanas para interagir de forma convincente com os usuários (Turing, 1950).

2.5.2 ELIZA

ELIZA foi criada com o propósito de estudar a comunicação em linguagem natural entre humanos e máquinas. Desenvolvida anos após o teste de Turing, ELIZA não busca demonstrar a capacidade de máquinas pensarem, mas sim explorar a simulação de entendimento em interações humanas. Atuando como terapeuta, ela recebe, processa e responde sentenças simples recebidas do usuário, analisando-as em busca de palavras-chave. Para cada palavra-chave identificada, aplica-se uma regra associada que reestrutura a frase e gera uma resposta a partir de um banco de respostas pré-definidas. Após encontrar a palavra-chave da sentença e a resposta correspondente, esta é retornada ao usuário. ELIZA se destacou pelo fato de manter a ilusão de entendimento devido ao uso de respostas genéricas e devolutivas, características do estilo terapêutico que não exige conhecimento de mundo por parte do programa. Contudo, ELIZA é essencialmente um sistema de mapeamento entre entradas e saídas baseadas em regras, sem qualquer processamento semântico real (Weizenbaum, 1966).

2.5.3 *Artificial Linguistic Internet Computer Entity*

Richard Wallace conduziu um estudo com seu *Artificial Linguistic Internet Computer Entity* (A.L.I.C.E.) para expor o que chamou de 'anatomia de A.L.I.C.E.'. Assim como ELIZA, A.L.I.C.E. é baseada em regras, mas se diferencia por ser programada em *Artificial Intelligence Markup Language* (AIML) e por contar com uma base de dados significativamente maior (Wallace, 2003a). A linguagem AIML, desenvolvida por Wallace e sua comunidade, introduziu uma abordagem mais avançada para a estruturação das respostas, permitindo que fossem organizadas em categorias, cada uma composta por um *pattern* (padrão) e um *template* (modelo). Essa abordagem estruturada possibilitou recursividade, reutilização e escalabilidade Wallace (2003b), fatores que contribuíram para a ampliação expressiva da base de conhecimento do *chatbot*. Com um cérebro baseado em AIML, A.L.I.C.E. supera o simples reconhecimento de palavras-chave utilizado por ELIZA, permitindo interações mais sofisticadas e diálogos mais elaborados (Wallace, 2003a).

2.5.4 *SmarterChild*

Um exemplo da aplicação de recuperação de informação em interfaces conversacionais é o *SmarterChild*, um *chatbot* lançado em 2001 que fornecia aos usuários informações em tempo real, como previsões meteorológicas, notícias e cotações de ações. Sua alta performance deu-se por possuir base de dados robusta e informações dos usuários que eram encontrados pela recuperação de informação (Carvalho, 2020). Desenvolvido pela *ActiveBuddy*, o *SmarterChild* integrava técnicas de processamento de linguagem natural e recuperação de informação para interagir de forma eficiente com os usuários, demonstrando o potencial da recuperação de informação em sistemas de diálogo (Adamopoulou; Moussiades, 2020b). O *SmarterChild* foi um dos primeiros exemplos de como a recuperação de informação poderia ser aplicada com

sucesso em interfaces conversacionais, abrindo caminho para o desenvolvimento de assistentes mais avançados.

2.5.5 *Google Neural Conversational Model*

Com um grande avanço tecnológico e temporal, em 2015 é lançado o *Google Neural Conversational Model*, onde este figura uma mudança de paradigma, introduzindo a capacidade de gerar respostas a partir de redes neurais, em vez de apenas recuperar ou seguir regras pré-definidas. Utilizando a arquitetura Seq2Seq, baseada em Redes Neurais Recorrentes, esse modelo demonstrou ser surpreendentemente eficaz na geração de respostas fluentes e coerentes em conversas, destacando o potencial da IA para diálogos mais naturais (Vinyals; Le, 2015). Esse avanço representa um ponto de virada na história dos *chatbots*, pois, diferentemente de abordagens anteriores, baseadas em regras ou recuperação de informação, ele permitiu a criação de *chatbots* generativos, capazes de aprender padrões linguísticos a partir de grandes conjuntos de dados. No entanto, as redes Seq2Seq ainda apresentavam limitações na retenção de informações de longo prazo e na coerência de respostas em interações mais complexas. A busca por melhorias levou à adoção de arquiteturas mais avançadas, culminando nos LLMs, que utilizam *Transformers* para lidar com longos contextos e gerar textos altamente coerentes e relevantes (Zhao et al., 2024).

2.5.6 *Generative Pre-trained Transformer* - 2

Em 2019, a OpenAI introduziu o *Generative Pre-trained Transformer* (GPT)-2, um modelo de linguagem que performou esta mudança. Com 1,5 bilhão de parâmetros, o GPT-2 demonstrou capacidades significativas em diversas tarefas de processamento de linguagem natural, como tradução automática, resumo de textos e resposta a perguntas, sem a necessidade de treinamento supervisionado específico para cada tarefa. Essa abordagem destacou o potencial dos grandes modelos de linguagem para aprender múltiplas tarefas de forma não supervisionada Radford et al. (2019). A transição para arquiteturas Seq2Seq para as baseadas em *Transformers* permitiu que modelos como o GPT-2 capturassem dependências de longo alcance no texto, gerando respostas mais coerentes e contextualmente relevantes Zhao et al. (2024).

2.5.7 *Generative Pre-trained Transformer* - 3

No ano seguinte, a OpenAI levou a tecnologia a um novo patamar com o lançamento do GPT-3, um modelo de linguagem autoregressivo que ampliou significativamente tanto a escala quanto a capacidade de retenção de contexto. Assim como seu antecessor, o GPT-3 utiliza LLMs, aprendizado não supervisionado e arquiteturas baseadas em *Transformers* para gerar respostas. No entanto, o grande diferencial do GPT-3 reside em sua base de dados com cerca de 175 bilhões de parâmetros. O modelo mostrou uma habilidade aprimorada de manter o contexto ao longo de conversas de múltiplos turnos, de forma muito mais eficiente (Brown et al., 2020).

Treinado em um conjunto massivo de dados, o GPT-3 é capaz de oferecer respostas detalhadas e contextualizadas sobre uma vasta gama de tópicos. Além disso, sua capacidade de ajustar respostas ao tom e ao estilo da conversa faz com que as interações entre humanos e máquinas se aproximem ainda mais da naturalidade de um diálogo real (Zhao et al., 2024). Esse avanço, combinado com a habilidade de manter diálogos fluidos e contextuais, não só reforça a evolução dos LLMs, como também estabeleceu um novo padrão para a interação entre humanos e IA.

2.5.8 *Sparrow*

Retornando ao aprendizado de máquina supervisionado e adicionando técnicas de ajuste fino, *Sparrow* visa maior segurança e precisão nas respostas. Apresentado em 2022, a *DeepMind* traz um modelo de *chatbot* especializado, otimizado para interações éticas, suporte técnico e atendimento ao cliente (Glaese et al., 2022). Diferentemente do GPT-3, que operava predominantemente de forma não supervisionada, o *Sparrow* foi desenvolvido com foco em segurança e precisão nas respostas. O modelo utiliza *feedback* direto e ajuste fino para aprimorar suas respostas com base em interações passadas e garantir conformidade com diretrizes éticas (Glaese et al., 2022).

2.6 Trabalhos Relacionados

Foram identificados na literatura alguns trabalhos correlatos a este, todos voltados a criação de *chatbots* aplicados ao contexto educacional. Melo; Melo et al. (2023) desenvolveram o RegBot, um assistente virtual com a finalidade de responder a dúvidas sobre o regulamento de ensino de graduação da Universidade Federal de Campina Grande (UFCG). O sistema foi projetado para responder dúvidas recorrentes como trancamento de disciplinas, reaproveitamento de matérias, reingresso, entre outras. Requisitos de acessibilidade foram devidamente observados, buscando atender alunos com deficiência visual. A solução utilizou técnicas de PLN com o uso de LLM, como o *Bidirectional Encoder Representations from Transformers* (BERT) e o *Universal Sentence Encoder*, aliadas a aprendizado supervisionado, vetorização semântica dos artigos do regulamento e validação do modelo através de verificação de acurácia. Embora os autores apresentem uma abordagem com uso de IA, a solução não possui memória de conversação, implementação de agentes autônomos ou integração com aplicativos de mensagens, restringindo-se o uso à interface *web*.

Almeida Fernandes; Vasconcelos Silveira (2024) criaram um *chatbot* voltado à orientação sobre a política de assistência estudantil no Instituto Federal do Ceará (IFCE) com o intuito de elucidar as principais dúvidas dos alunos em relação aos auxílios, tais como moradia, alimentação, transporte e outros benefícios oferecidos pela Instituição. O *chatbot* foi projetado a partir de mapeamento prévio das perguntas mais recorrentes. Em relação ao funcionamento, os autores utilizaram fluxos de diálogo estruturados, indicando ausência de técnicas de aprendizado de máquina ou uso de grandes modelos de linguagem. Em contrapartida da simplicidade da

solução desenvolvida, a aplicação foi integrada com plataformas de mensagem instantânea, como o *Telegram* e o *WhatsApp*, sendo um diferencial importante para acessibilidade. O trabalho também adotou um modelo de validação com usuários reais, aplicando a metodologia *System Usability Scale* (SUS) com 12 participantes, que apontaram um elevado índice de satisfação. Contudo, o sistema não apresenta as técnicas de vetorização, indexação semântica ou memória de conversa, sendo limitado a navegação pelos fluxos previamente definidos.

Barbosa (2024) desenvolveu um *chatbot* voltado para automatização do atendimento de dúvidas acadêmicas da Pró-Reitoria de Graduação da Universidade Federal de Ouro Preto (UFOP). A solução foi criada para responder a perguntas frequentes dos estudantes sobre temas como trancamento de disciplinas, estágios, bolsas, monitorias e outras questões institucionais. O autor usou uma abordagem baseada em LLMs, integrando a *Application Programming Interface* (API) do *ChatGPT* com técnicas de indexação semântica, por meio da biblioteca *LlamaIndex*. O *chatbot* é acessado através de uma interface *web* desenvolvida, o que garante facilidade de uso e navegação. De maneira similar a Melo; Melo et al. (2023), Barbosa (2024) não possui integração com aplicativos de mensagens, recurso considerado vantajoso por proporcionar maior familiaridade ao usuário e possibilitar a ampliação do alcance da ferramenta. Nos testes realizados, o sistema obteve 100% de acurácia em algumas intenções, mantendo acurácia mínima de 70% nas demais, demonstrando desempenho satisfatório. No entanto, o sistema não apresenta memória de conversação nem suporte a agentes autônomos, o que limita a personalização das interações.

Tabela 2.1: Comparação das funcionalidades entre trabalhos correlatos e o sistema de conversação **`TremBot`**

Funcionalidade	**REGBOT**	**IFCE**	**UFOP**	**TREMBOT**
Uso de LLM	✓		✓	✓
Aprendizado de máquina	✓		✓	✓
Documentos multissetoriais			✓	✓
Enriquecimento de contexto				✓
Vetorização do modelo	✓		✓	✓
Busca vetorial híbrida				✓
Validação do modelo	✓	✓	✓	✓
Análise de performance				✓
Feedback dos usuários	✓	✓	✓	✓
Memória da conversa				✓
Integração com *WhatsApp*		✓		✓

Legenda: REGBOT refere-se a (Melo; Melo et al. (2023)); IFCE refere-se a (Almeida Fernandes; Vasconcelos Silveira (2024)); UFOP refere-se a (Barbosa (2024)).

A Tabela 2.1 sintetiza as funcionalidades observadas nos trabalhos correlatos, evidenciando os diferenciais tecnológicos do **`TremBot`**, especificamente a busca vetorial híbrida e o enriquecimento de contexto. Além de responder as dúvidas acadêmicas baseadas em documentos multissetoriais, **`TremBot`** fornece uma interação mais flexível, baseada em linguagem natural,

sem depender de fluxogramas rígidos. A solução incorpora recursos fundamentais, como a análise de performance e mecanismos de *feedback* dos usuários, que favorecem o aprendizado ao longo das interações. Adicionalmente, o sistema de conversação engloba mecanismos de validação do modelo e memória, proporcionando uma experiência mais personalizada e contextualizada para o usuário. Diferentemente de Melo; Melo et al. (2023), existe integração com o *WhatsApp*, expandindo o alcance e a acessibilidade da ferramenta. **`TremBot`** figura como uma alternativa eficiente para apoiar toda a comunidade em dúvidas institucionais, com potencial de uso em diversos contextos acadêmicos.

3
Materiais e Métodos

Esse Capítulo detalha as fases de desenvolvimento do **`TremBot`**, desde a concepção de seus requisitos até a implantação e avaliação do protótipo. Adotou-se abordagem que combinou o ***design thinking*** na fase inicial com o **desenvolvimento iterativo e incremental**, permitindo que as fases de indexação e avaliação fossem integradas e aprimoradas em ciclos contínuos de desenvolvimento.

3.1 Fase de Elicitação e Análise de Requisitos

Esta fase teve como objetivo levantar as necessidades e definir o escopo do *chatbot*, resultando na documentação dos requisitos funcionais (RF), não funcionais (RNF) e das Regras de Negócio (RN).

Documentos oficiais do IFG *Campus* Formosa (Projetos Pedagógicos de Cursos (PPC), regulamentos de TCC, calendários, etc.) foram analisados para estruturar a base de conhecimento inicial do sistema.

Com base na elicitação, foram definidos os requisitos do sistema, com foco em:

- **Regras de Negócio (RN):** Garantir a comunicação em português (RN01), o uso exclusivo de fontes oficiais (RN02), a recusa explícita em caso de informação ausente (RN03) e o atendimento à comunidade interna e externa (RN04).
- **Requisitos Funcionais (RF):** Que descrevem a capacidade de ingestão, fragmentação (RF02), vetorização (RF03), armazenamento (RF04), busca semântica (RF06) e a arquitetura RAG (RF07).
- **Requisitos Não Funcionais (RNF):** Focados em qualidade, como a disponibilidade de 99% (RNF01), a acessibilidade via *WhatsApp* (RNF03) e a alta fidelidade às fontes (RNF02).

3.2 Fase de Desenvolvimento e Integração

Esta fase abrangeu a configuração dos componentes da solução e o desenvolvimento da lógica conversacional (núcleo do *chatbot*).

O **`TremBot`** foi desenvolvido utilizando arquitetura modular e desacoplada, fundamentada no paradigma RAG. Os componentes chave foram:

- ***Pipeline* de *Extract, Transform, Load* (ETL)**: Utiliza ferramentas de código aberto e serviços gerenciados para a extração, pré-processamento (*Optical Character Recognition* (OCR) e *Markdown*), vetorização e armazenamento de alta performance dos documentos do IFG.
- **Núcleo de Processamento**: Implementado em Python (*Microframework* Flask), utilizando o modelo **Gemini 2.5 Flash** para geração de linguagem natural e o *framework* *LlamaIndex* para orquestração RAG e gerenciamento de contexto.
- **Interface de Comunicação:** Desenvolvimento de cliente em *Node.js* com a biblioteca *whatsapp-web.js* para gerenciar sessão ativa e tráfego de mensagens via *WhatsApp*. O serviço *Ngrok* foi usado para criar um túnel seguro entre o cliente e a API *webhook*.

Em consonância com o RF10, o sistema implementa mecanismo de memória para preservar o contexto da conversação, configurado com *buffer* de 3000 *tokens*. Para cada usuário, uma instância de memória dedicada é criada e recuperada a cada nova interação, permitindo ao LLM gerar respostas coerentes com o histórico.

3.3 Coleta, Tratamento e Indexação de Dados

Ao contrário do planejamento inicial, esta fase foi executada após o desenvolvimento do Módulo de Indexação (RF01, RF02, RF03, RF04), garantindo que o *pipeline* de dados fosse integrado ao sistema.

Os documentos foram processados através de uma arquitetura ETL automatizada:

1. **Extração**: Coleta dos arquivos em formato *Portable Document Format* (PDF), *Uniform Resource Locator* (URL) e *JavaScript Object Notation* (JSON) a partir do repositório central (Google Drive).
2. **Transformação:**
    - **Pré-processamento e OCR**: Utilização do *LlamaParse* para converter PDFs em formato *markdown*, preservando a estrutura (tabelas, cabeçalhos). Quando a extração falha, o módulo de OCR é acionado condicionalmente para transcrição automática de documentos escaneados.

- **Fragmentação (*Chunking*)**: O texto foi segmentado em "*chunks*"de 2048 *tokens* com sobreposição de 200 *tokens* para otimizar a precisão da recuperação.
- **Vetorização:** Cada fragmento foi convertido em um vetor numérico (*embedding*) utilizando o modelo **text-embedding-004**.

3. **Carga**: Os vetores de *embedding* e os textos originais dos *chunks* foram armazenados na coleção `documentos_ifg` no MongoDB Atlas, conforme ilustrado no fluxo de indexação.

3.4 Fase de Testes e Avaliação

A avaliação foi realizada em duas etapas (avaliação prévia e avaliação com *stakeholders*), descritas a seguir:

A performance do modelo foi testada incrementalmente por meio de 30 ciclos de avaliação, cada um correspondente à adição de um novo documento à base de conhecimento. A avaliação se deu em dois eixos:

- **Teste de Novo Conhecimento**: Verificava a assimilação do conteúdo recém-adicionado.
- **Teste de Regressão**: Reaplicava perguntas de ciclos anteriores para garantir que o novo conteúdo não degradasse funcionalidades pré-existentes, mantendo a consistência.

As métricas focaram na Acurácia e Matriz de Confusão para analisar a capacidade do sistema em:

- Responder corretamente perguntas de “Fato Direto”, “Busca Numérica”, “Extração de Lista” e “Resumo de Conceito”.
- Recusar corretamente as “Perguntas sem Resposta”, mitigando o risco de alucinações (RN03).

A avaliação com usuários reais totalizou 1.059 interações processadas em um ambiente de produção assistido. Esta etapa buscou medir o desempenho do **`TremBot`** em um cenário real e heterogêneo.

- **Análise do Perfil de Uso**: Verificação da distribuição de usuários (Alunos, Docentes/Técnicos, Comunidade Externa) e a validação do alcance da ferramenta.
- **Análise de Desempenho em Tópicos**: Quantificação do volume de interações por tópico e análise da frequência de termos para confirmar a relevância do sistema em demandas específicas do IFG.

- **Métricas de Qualidade de Resposta**: Análise de relevância e fidelidade para garantir que as respostas geradas fossem, além de coerentes, baseadas nos documentos fonte (RN02).

Esta fase consistiu o levantamento dos recursos financeiros necessários para a transição do protótipo do **`TremBot`** para um ambiente de produção estável e escalável. A metodologia baseou-se na decomposição dos custos operacionais em quatro pilares fundamentais: processamento de IA, armazenamento vetorial, busca híbrida e infraestrutura de hospedagem.

Para garantir a fidelidade institucional e o cumprimento da RN03, optou-se por uma estratégia de precificação baseada em uso (*Pay-as-you-go*), permitindo a escalabilidade conforme a demanda da comunidade acadêmica. A análise considerou:

- **Processamento e *Tokens*:** O cálculo do consumo de *tokens* de entrada e saída do modelo **Gemini 2.5 Flash**, abrangendo tanto o fluxo de geração de respostas quanto o fluxo de avaliação automática de métricas de qualidade.
- **Persistência e Busca:** A estimativa de custos no MongoDB Atlas *Serverless*, focando na quantidade de Unidade de Processamento Reconfigurável ou *Reconfigurable Processing Unit* (RPU) necessárias para manter a latência de busca híbrida abaixo de 600ms, conforme observado nos testes de desempenho.
- **Infraestrutura:** A comparação entre modelos de hospedagem em nuvem, visando o cumprimento do **RNF01** (disponibilidade de 99%), e modelos *on-premise*.

O detalhamento dos valores e a projeção mensal baseada no volume de interações coletado durante a avaliação com os *stakeholders* são apresentados na Capítulo 5.

4

TremBot: Análise e Documentação

Este capítulo detalha o desenvolvimento técnico e a arquitetura do **`TremBot`**, sistema de conversação projetado para o IFG - *Campus* Formosa. A apresentação abrange desde a especificação dos requisitos funcionais e não funcionais até a documentação da infraestrutura tecnológica utilizada, baseada na arquitetura RAG.

4.1 Seleção dos Dados

A construção da base de conhecimento do *chatbot* foi fundamentada em fontes oficiais e públicas, assegurando confiabilidade e conformidade institucional das informações prestadas, em consonância com a Regra de Negócio 02 (RN02). O conjunto de documentos abrange os pilares de ensino, pesquisa e extensão, sendo composto por documentos extraídos do portal[1] do *Campus* Formosa. Os dados selecionados incluem:

- **Projetos Pedagógicos de Cursos:** documentos originais no formato PDF contendo matrizes curriculares, ementas e perfis de egresso dos cursos técnicos integrados ao ensino médio (Biotecnologia, Edificações, Saneamento), superiores (Engenharia Civil, Licenciaturas em Ciências Biológicas/Ciências Sociais, Tecnologia em Análise e Desenvolvimento de Sistemas) e da modalidade Educação de Jovens e Adultos (Manutenção e Suporte em Informática e Edificações);
- **Calendários Acadêmicos:** referentes aos anos letivos de 2025 e 2026, essenciais para a identificação de datas letivas, feriados e prazos acadêmicos;
- **Pilares Acadêmicos:** conteúdos extraídos do site institucional que detalham as diretrizes e funcionamentos dos eixos de ensino, pesquisa e extensão, incluindo programas específicos e projetos em andamento;
- **Estrutura Organizacional e Setores:** informações sobre o funcionamento de setores do *Campus* Formosa, abrangendo a Biblioteca, Teatro Guaiá, a Coordenação de

[1] https://ifg.edu.br/formosa

Comunicação Social, a Área Acadêmica e o Núcleo de Atendimento às Pessoas com Necessidades Específicas (NAPNE);

- **Regulamentação de TCC:** compilado contendo o regulamento geral de Trabalho de Conclusão de Curso, além de diretrizes específicas extraídas para os cursos superiores;
- **Servidores**: dados coletados referentes ao quadro de professores e técnicos-administrativos do *campus* Formosa.

4.2 Pré-processamento

A construção da base de conhecimento fundamentou-se em uma arquitetura de Engenharia de Dados conhecida como ETL conforme ilustrado na Figura 4.1. Esta *pipeline* permite converter os documentos institucionais em vetores semânticos para recuperação. A etapa de **Extração** (i) consiste na coleta dos arquivos PDF armazenados no repositório central (Google Drive). Na etapa de **Transformação** (ii), utilizou-se a ferramenta *LlamaParse* para converter os arquivos em formato *Markdown*, preservando estruturas complexas como tabelas, seguida pela higienização de metadados e fragmentação (*chunking*) do texto. Ainda nesta fase, ocorre o enriquecimento contextual e a vetorização dos fragmentos via modelos de *embedding*. Por fim, a etapa de **Carga** (iii) consolida os dados processados no banco de dados vetorial MongoDB Atlas, habilitando o sistema para consultas de alta performance.

Figura 4.1: Arquitetura ETL utilizada para extração, transformação e carregamento dos dados.

4.3 Funcionamento Geral

O funcionamento do sistema fundamenta-se na arquitetura RAG e ocorre por meio de dois fluxos distintos. O primeiro consiste na estruturação da base de conhecimento, onde os documentos institucionais são processados e indexados vetorialmente para permitir recuperação semântica. O segundo fluxo compreende o ciclo de interação com o usuário, no qual as mensagens recebidas via *WhatsApp* são confrontadas com a base indexada para geração de respostas contextualizadas pelo LLM. A orquestração entre interface de comunicação e núcleo de processamento é realizada por uma API central, responsável pelo tráfego de dados e pela gestão da sessão conversacional.

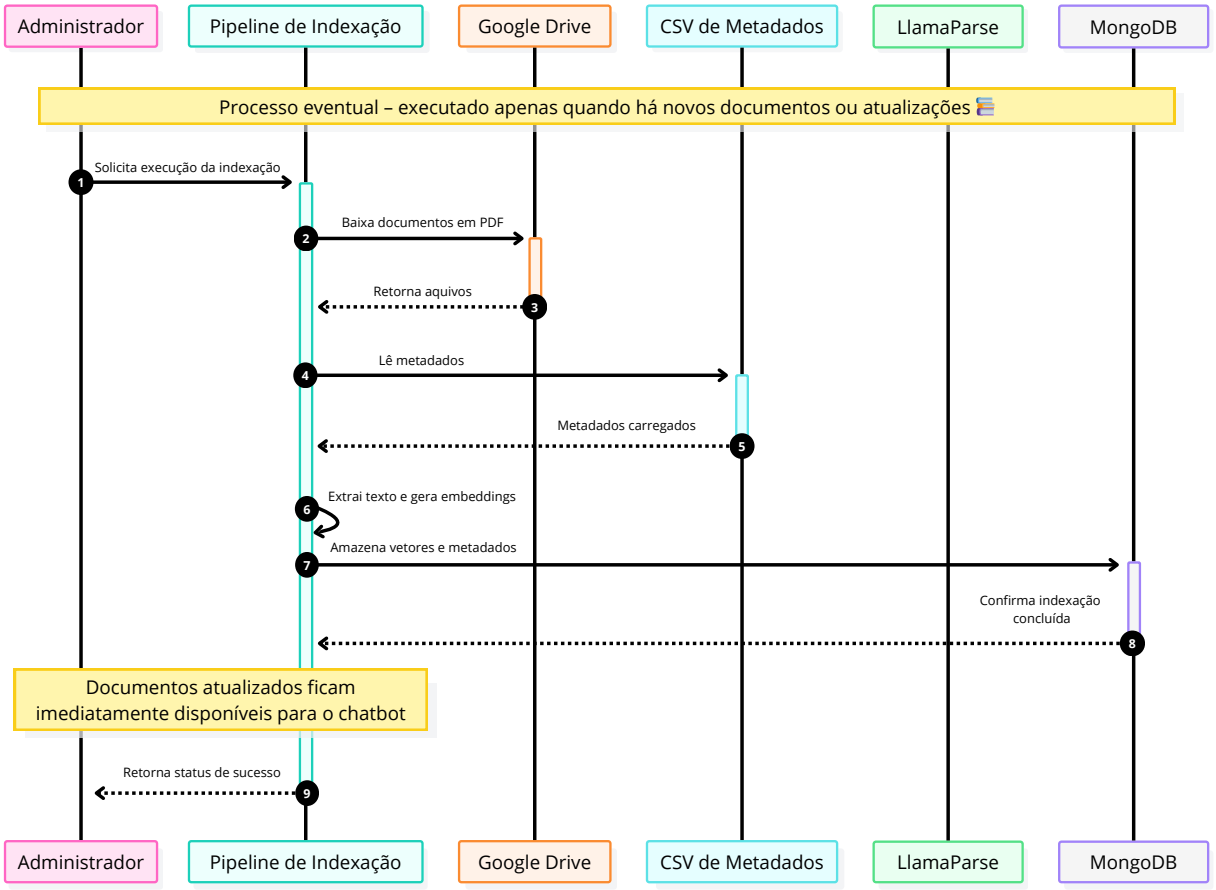


Figura 4.2: Diagrama de sequência relacionado ao processo de indexação de documentos (Fase 1).

4.3.1 Geração e Armazenamento de Contexto

A eficácia de um sistema de conversação fundamentado na arquitetura RAG está intrinsecamente associada à qualidade e organização da base de conhecimento que o sustenta. No presente trabalho, a construção dessa base envolve processo sistemático de pré-processamento e indexação dos documentos institucionais do IFG. Essa etapa constitui pré-requisito essencial, pois converte documentos brutos em formato otimizado para a recuperação ágil e semanticamente precisa de informações pelo *chatbot*. Conforme ilustra a Figura 4.5, o processo é composto por quatro fases principais: **(i) Carregamento de dados**, **(ii) Divisão de documentos**, **(iii) Geração de *embeddings***, e (iv) **Armazenamento vetorial**.

Na primeira etapa, representada pelo item (i) na Figura 4.5, realizou-se o carregamento dos documentos institucionais referentes às áreas de ensino, pesquisa e extensão em repositório no Google Drive[2].Essa escolha proporcionou flexibilidade na gestão e atualização dos arquivos, além de facilitar a integração de documentos provenientes de diferentes setores da Instituição. A diversidade das fontes contribuiu para ampliar o escopo informacional disponível ao sistema, aprimorando a capacidade do *chatbot* em responder distintos tipos de consultas.

Na etapa (ii) exibida na Figura 4.5, iniciou-se o processo de divisão e tratamento dos

[2]https://drive.google.com/drive

documentos. Arquivos com estruturas complexas foram processados utilizando o serviço *Llama-Parse*, que é uma plataforma de análise de documentos desenvolvida pela *LlamaIndex*[3], projetada para converter documentos complexos, tais como PDFs em dados estruturados e "prontos" para IA. A conversão foi executada preservando a formatação original, incluindo tabelas, seções e hierarquias textuais, com objetivo de minimizar a perda de contexto.

Com os textos devidamente estruturados, procedeu-se à segmentação do conteúdo em fragmentos menores e semanticamente coerentes, denominados "*chunks*". Essa etapa é fundamental para assegurar maior precisão na recuperação das informações, permitindo que o *chatbot* identifique trechos específicos de relevância, reduzindo a ocorrência de respostas genéricas e/ou imprecisas.

Na etapa (iii) da Figura 4.5, cada *chunk* passou pelo processo de geração de *embeddings*. Os fragmentos textuais foram transformados em vetores numéricos de alta dimensionalidade, capazes de capturar relações semânticas entre os termos. Essa representação vetorial é fundamental para comparação eficiente entre conteúdo da base de conhecimento e consultas formuladas pelos usuários.

Por fim, na etapa (iv) da Figura 4.5, os vetores resultantes do processo de *embeddings* e os textos originais dos *chunks* foram armazenados em um banco de dados vetorial no MongoDB Atlas[4]. Essa solução foi selecionada devido ao desempenho e escalabilidade na execução de buscas por similaridade semântica, permitindo que o sistema recupere, de forma rápida e precisa, os documentos mais relevantes para a geração de respostas contextualmente adequadas.

4.3.2 Processo de Pergunta e Geração de Resposta

Uma vez que a base de conhecimento foi indexada no MongoDB, o sistema está pronto para interagir com o usuário. O fluxo de conversação, ilustrado na Figura 4.3, descreve o ciclo de vida de uma pergunta, do momento em que é realizada pelo usuário até a geração de uma resposta, fundamentada nos documentos institucionais. Este processo é baseado na arquitetura RAG e ocorre em quatro etapas principais.

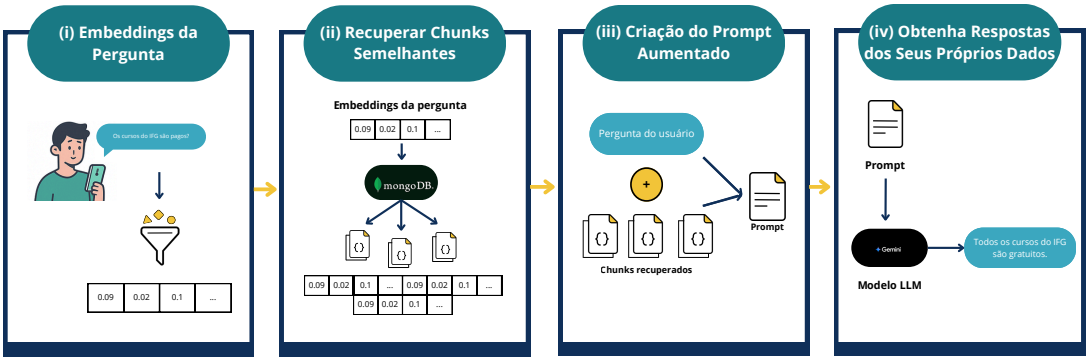


Figura 4.3: Fluxo de funcionamento do sistema de conversação baseado em RAG

[3]https://www.llamaindex.ai/
[4]https://www.mongodb.com/

O fluxo de recuperação e geração de respostas no *chatbot* segue uma sequência de etapas que integram a busca semântica e a Gen-AI, assegurando que as respostas oferecidas sejam baseadas nos dados institucionais. Conforme é exibido na Figura 4.3, o processo se inicia com a geração dos *embeddings* da pergunta. Na etapa (i), quando o usuário envia uma consulta em linguagem natural, como por exemplo: "Quando será o fim do primeiro semestre?", o sistema utiliza o modelo de *embeddings* para transformar a pergunta em um vetor numérico de alta dimensionalidade.

Com o *embeddings* resultante da pergunta, a segunda etapa consiste na recuperação dos *chunks* mais semelhantes contidos na base vetorial armazenada no MongoDB Atlas. O sistema realiza uma busca vetorial, comparando o *embeddings* da pergunta (etapa (i) da Figura 4.3) com os *embeddings* armazenados dos documentos institucionais (etapas (iii) e (iv) da Figura 4.5). Essa busca retorna os *chunks* mais semanticamente próximos da pergunta realizada.

Com esses conteúdos recuperados, é construído um *prompt* aumentado, que consiste na combinação entre a pergunta original com os *chunks* dos documentos mais relevantes, formando um contexto ampliado para o LLM. A tarefa do LLM consiste em analisar o conteúdo fornecido e gerar uma resposta precisa, coesa e fundamentada nos documentos institucionais do IFG.

Finalmente, a resposta gerada pelo LLM é encaminhada ao usuário, completando o ciclo de atendimento. Esse processo assegura que as respostas oferecidas pelo *chatbot* sejam sempre fundamentadas nas informações oficiais do IFG, garantindo precisão, confiabilidade e alinhamento institucional.

4.3.3 Ajustes de Parâmetros

Para assegurar equilíbrio entre precisão das informações recuperadas e coerência da linguagem natural gerada, foram definidos parâmetros específicos. Esses ajustes visam mitigar alucinações e garantir que as respostas sejam fundamentadas nos documentos institucionais.

Em relação ao modelo generativo, optou-se pela utilização do **Gemini 2.5 Flash** com **temperatura** configurada em $0,1$. Esse valor próximo de zero reduz a aleatoriedade do modelo, tornando as respostas mais determinísticas e factuais. Adicionalmente, definiu-se um limite de 4096 *tokens*.

Para o processo de recuperação de informação, a estratégia de indexação apoiou-se na ferramenta *LlamaParse*, configurada para estruturar os dados em formato *Markdown*, preservando cabeçalhos e tabelas. A segmentação dos documentos (*chunking*) foi ajustada com tamanho de bloco (**chunk_size**) de 2048 *tokens* e sobreposição (**chunk_overlap**) de 200 *tokens*. Essa configuração permite que cada fragmento de texto contenha contexto suficiente para ser compreendido isoladamente, enquanto a sobreposição evita perda de informações contidas nas fronteiras entre os blocos. Para a vetorização desses fragmentos, foi empregado o modelo **text-embedding-004**, otimizado para tarefas de recuperação textual.

A etapa de recuperação da informação foi configurada para operar em modo híbrido

(**hybrid search**), utilizando um parâmetro $\alpha = 0,5$. O valor de α varia em um intervalo de $[0,1]$, no qual 0 representa a busca exclusiva por palavras-chave (*keyword search*) e 1 prioriza totalmente a busca vetorial (semântica); o ajuste em $0,5$ combina ambas com pesos iguais para equilibrar termos específicos e conceitos abstratos. Adicionalmente, o parâmetro **similarity_top_k** foi definido como 8. Este valor foi determinado por meio de testes empíricos, visando maximizar a probabilidade de incluir a resposta exata no contexto enviado ao LLM sem exceder o limite de *tokens* da janela de contexto do modelo.

Por fim, para gerenciamento do fluxo de conversação optou-se pela configuração **condense_plus_context**. Este modo reformula a pergunta atual do usuário considerando o histórico da conversa antes de realizar a busca no banco de dados, garantindo que consultas dependentes de interações anteriores sejam melhor compreendidas. A memória conversacional foi configurada com um *buffer* de 3000 *tokens*, permitindo ao sistema manter o contexto de interações recentes sem comprometer o desempenho ou o custo computacional.

4.3.4 Integração com *WhatsApp*

A integração do sistema de conversação com a plataforma *WhatsApp* constituiu um dos elementos centrais deste projeto, visando ampliar o alcance e acessibilidade da ferramenta ao utilizar canal de comunicação consolidado no cotidiano da comunidade acadêmica. Para possibilitar essa integração, foi projetada uma arquitetura de comunicação desacoplada, cujo fluxo de interação é ilustrado na Figura 4.4.

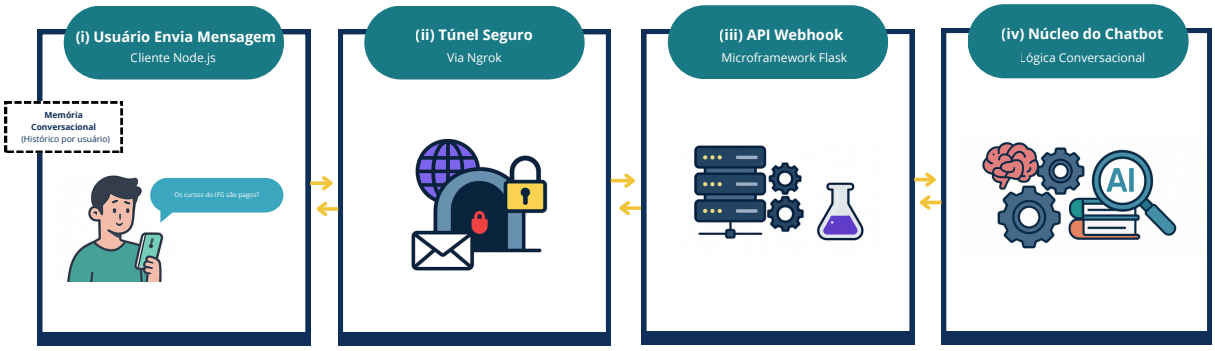


Figura 4.4: Arquitetura de integração com a plataforma *WhatsApp*

Como pode ser observado na Figura 4.4, o ciclo de vida de uma interação inicia-se quando o usuário envia uma mensagem de texto por meio do *WhatsApp* (etapa (i)). Essa mensagem é recebida por um cliente de conexão, que é uma aplicação desenvolvida em *Node.js* com a biblioteca *whatsapp-web.js*. Este componente é responsável por manter uma sessão ativa e monitorar as mensagens enviadas ao *chatbot*. Primeiro, ao detectar nova interação, a aplicação identifica o conteúdo textual e o identificador do remetente, dados essenciais para o processamento e para a gestão da memória conversacional. Em seguida, conforme exibida na

etapa (ii) da Figura 4.4, essas informações são encapsuladas e enviadas para um site público gerado pelo serviço *Ngrok*, que atua como um túnel ao expor a API de *backend* em execução local para a Internet.

Na etapa (iii) da Figura 4.4, a requisição enviada pelo cliente *Node.js*[5] é redirecionada pelo *Ngrok*[6] e chega à API *webhook*, desenvolvida em *Python*[7] com o *microframework Flask*[8]. Esta API funciona como porta de entrada, recebendo dados da requisição e orquestrando ações subsequentes. Sua principal função é validar dados recebidos e encaminhá-los ao núcleo de processamento do *chatbot* - etapa (iv) - que constitui camada responsável pela lógica conversacional. Caso a requisição esteja incompleta ou apresente inconsistências, a API retorna resposta padronizada de erro, garantindo a robustez e confiabilidade da comunicação entre os componentes.

No núcleo do sistema (etapa (iv) da Figura 4.4) ocorre o processo de RAG, responsável por combinar busca semântica em uma base de conhecimento com geração de linguagem natural. A mensagem recebida é analisada e transformada em representação vetorial, sendo comparada aos vetores armazenados no banco de dados MongoDB Atlas, que contém documentos oficiais do IFG previamente indexados. Os trechos mais relevantes são recuperados e utilizados como contexto para a geração da resposta final, garantindo coerência nas informações. apresentadas.

Um dos diferenciais desta arquitetura reside na implementação da memória conversacional, a qual habilita o sistema a preservar contexto ao longo das interações com o usuário. Para cada identificador único, é criada uma instância dedicada de memória responsável por armazenar o histórico das mensagens trocadas. Esse histórico é recuperado a cada nova interação e combinado com a pergunta atual e com fragmentos documentais relevantes, compondo o conjunto de informações que orienta o modelo de linguagem na formulação da resposta.

Essa abordagem permite que o *chatbot* compreenda questões subsequentes que dependem de diálogos anteriores, garantindo experiência natural, contextualizada e análoga à conversação humana. A resposta gerada é devolvida à API, que a transmite ao cliente de conexão e, posteriormente, ao usuário via *WhatsApp*, completando o ciclo de comunicação em tempo real.

A arquitetura modular e desacoplada, aliada ao mecanismo de memória conversacional, torna o sistema escalável, personalizável e apto à integração com outros canais de comunicação em cenários futuros.

4.4 Levantamento e Análise de Requisitos

A definição dos requisitos constitui etapa para delimitação do escopo e funcionalidades. Por meio da análise de documentos institucionais e mapeamento das demandas informacionais da comunidade acadêmica, foram identificadas necessidades prioritárias que o *chatbot* deveria

[5]https://nodejs.org/pt
[6]https://ngrok.com/
[7]https://www.python.org/
[8]https://flask.palletsprojects.com/en/stable/

atender. Para organizar o desenvolvimento e orientar a arquitetura da solução, os requisitos foram classificados em três categorias: **(i) Regras de Negócio**, que estabelecem normas e restrições operacionais do IFG, **(ii) Requisitos Funcionais**, que descrevem comportamentos e serviços que o sistema deve fornecer, e **(iii) Requisitos Não Funcionais**, que definem critérios de qualidade, desempenho e segurança.

4.4.1 Regras de Negócio (RN)

- **RN01:** o *chatbot* deve comunicar-se com o usuário em língua portuguesa.
- **RN02:** a fonte de informação do *chatbot* deve ser exclusivamente documentos institucionais do IFG fornecidos como base de conhecimento.
- **RN03:** caso a informação solicitada pelo usuário não seja encontrada nos documentos da base de conhecimento, o *chatbot* deve informar explicitamente que não pôde localizar a resposta, em vez de tentar inferir ou criar informação.
- **RN04:** o *chatbot* deve atender demandas informacionais de toda a comunidade acadêmica, inclusive da comunidade externa.

4.4.2 Requisitos Funcionais (RF)

Para melhor compreender as funcionalidades do sistema, os RFs foram elucidados em três blocos, listados a seguir.

RF - Módulo de Indexação e Gerenciamento de Dados

- **RF01:** o *chatbot* deve permitir ingestão de documentos a partir de fonte de dados centralizada (Google Drive).
- **RF02:** o *chatbot* deve segmentar os documentos em fragmentos de texto menores e semanticamente coerentes (*chunks*).
- **RF03:** o *chatbot* deve converter cada fragmento de texto em um vetor numérico utilizando modelo de *embeddings* específico.
- **RF04:** o *chatbot* deve armazenar fragmentos de texto e seus respectivos *embeddings* em base de dados vetorial MongoDB Atlas.

RF - Módulo de Processamento (*Backend*)

- **RF05:** o *chatbot* deve utilizar LLM para interpretar a semântica da pergunta do usuário e gerar as respostas.
- **RF06:** o *chatbot* deve realizar busca semântica na base de dados vetorial para encontrar trechos de documentos mais relevantes para a pergunta do usuário.

- **RF07:** o *chatbot* deve implementar arquitetura RAG, onde a pergunta do usuário é enriquecida com trechos relevantes da base de conhecimento antes de ser enviada ao LLM.

RF - Módulo de Interação com o Usuário

- **RF08:** o *chatbot* deve permitir que o usuário envie perguntas em linguagem natural.
- **RF09:** o *chatbot* deve permitir que o usuário receba respostas geradas a partir da base de conhecimento institucional.
- **RF10:** o *chatbot* deve manter histórico e contexto da conversa durante uma sessão de uso, permitindo que o usuário faça perguntas subsequentes relacionadas às anteriores.

4.4.3 Requisitos Não Funcionais (RNF)

- **RNF01:** o *chatbot* deve operar com disponibilidade de 99% do tempo, medido mensalmente, considerando a disponibilidade das APIs externas (Google Gemini) e da plataforma de hospedagem.
- **RNF02:** em avaliação com conjunto de 100 perguntas de teste, o sistema deve basear suas respostas no conteúdo dos documentos fornecidos em pelo menos 95% dos casos, verificado por avaliação humana.
- **RNF03:** o *chatbot* deve ser acessível por meio da plataforma *WhatsApp*, dispensando necessidade do usuário aprender usar nova interface, aproveitando a familiaridade com ferramenta amplamente utilizada.
- **RNF04:** a adição de novos documentos à base de conhecimento deve ser possível apenas adicionando arquivos na pasta configurada no Google Drive e re-executando o *script* de indexação, sem necessidade de alterações no código-fonte da aplicação.
- **RNF05:** as chaves de API e credenciais de acesso ao banco de dados não devem estar presentes no código-fonte. Devem ser gerenciadas por meio de variáveis de ambiente.

4.5 Modelagem do Sistema

A modelagem do sistema descreve a estrutura lógica e funcional que permite o funcionamento do *chatbot*, abrangendo desde a construção da base de conhecimento até o fluxo de interação com o usuário. A arquitetura proposta foi organizada em módulos independentes, mas integrados entre si.

A modelagem considera três componentes principais:

- *Pipeline* de ETL: Responsável por extrair, transformar e fragmentar os textos, além de gerar *embeddings* e armazenar as representações semânticas em um banco vetorial. Este componente, implementado no script `indexing.py`, operacionaliza os requisitos funcionais do **Módulo de Indexação e Gerenciamento de Dados**, garantindo a integridade da base de conhecimento;
- Módulo de Recuperação e Geração: Executa a busca híbrida e coordena a interação com o modelo generativo. Gerenciado pela lógica contida em `chatbot_logic.py`, este componente corresponde aos requisitos funcionais do **Módulo de Processamento**, sendo o núcleo de inteligência responsável pela síntese das respostas;
- Interface de Comunicação: Gerencia o fluxo de mensagens enviadas e recebidas via *WhatsApp*, conectando-se à API central. Implementado em `whatsapp_api.js`, este componente atende aos requisitos funcionais do **Módulo de Interação com o Usuário**, atuando como a ponte entre o usuário e o ecossistema do **`TremBot`**.

4.5.1 Modelagem dos Dados

A modelagem dos dados foi projetada para atender aos requisitos de recuperação semântica de documentos institucionais (RF06). Para isso, adotou-se o MongoDB Atlas como banco de dados vetorial, no qual são armazenados os textos processados e seus respectivos *embeddings* (RF04), além de informações cruciais sobre as sessões dos usuários e o histórico de interações (RF10). Essa estrutura garante não apenas a precisão na busca, mas também a persistência necessária para a continuidade das conversas e a auditoria das métricas de qualidade.

Por ser um banco de dados NoSQL, o MongoDB permite o armazenamento de estruturas flexíveis, o que o torna adequado para relacionar documentos institucionais, sessões de usuários e métricas de avaliação. Assim, foram criadas três coleções principais para atender às necessidades do projeto: `documentos_ifg`, `user_sessions` e `interacores_feedback`.

Cada documento institucional passa por um processo de fragmentação, gerando unidades menores de texto ("*chunks*") acompanhadas de metadados essenciais, como: id do documento original, vetor de *embedding*, conteúdo completo do fragmento e metadados. Essa organização permite que cada fragmento seja localizado e recuperado de forma eficiente durante o processo de busca semântica.

As sessões de usuário são registradas automaticamente a partir da primeira interação com o *chatbot*. Para cada sessão, são armazenados: id de usuário e de sessão, tipo de usuário e quantidade de mensagens. Esses dados são associados posteriormente às interações individuais, permitindo identificar quais tipos de perguntas são mais frequentes para cada perfil de usuário.

As interações registradas incluem: id da interação, id do usuário, pergunta do usuário, resposta do *chatbot*, id dos nós utilizados na geração da resposta, além de métricas de desempenho. O armazenamento dos identificadores dos nós é essencial para a avaliação do sistema, pois permitem calcular métricas de precisão e relevância com base nos trechos efetivamente utilizados

na geração da resposta. Da mesma forma, as métricas de desempenho são fundamentais para a análise apresentada no Capítulo 5.

4.5.2 Diagramas de Sequência

A Figura 4.2 e a Figura 4.6 apresentam o fluxo completo de funcionamento do sistema, dividido em duas etapas. A Fase 1 corresponde ao processo de indexação de documentos, executado de maneira pontual ou periódica, sempre que novos arquivos são inseridos ou atualizados na base institucional. A Fase 2 representa o fluxo contínuo de interação do *chatbot*, realizado sob demanda e em tempo real, a cada nova mensagem enviada por algum usuário da comunidade acadêmica.

O módulo OCR é acionado de forma condicional, apenas quando a extração textual convencional não obtém sucesso. O MongoDB atua como elemento central na arquitetura, integrando as duas fases: na Fase 1, armazena os vetores e metadados resultantes da indexação; na Fase 2, serve como base de consulta para a recuperação eficiente das informações relevantes durante as interações do *chatbot*.

Dessa forma, as duas figuras se complementam e, em conjunto, descrevem o funcionamento integral da solução proposta. A Fase 1 assegura a preparação e a atualização contínua do repositório de conhecimento, enquanto a Fase 2 explora esse conhecimento em tempo real, oferecendo respostas contextualizadas ao usuário, com base nas informações disponíveis na base de conhecimento.

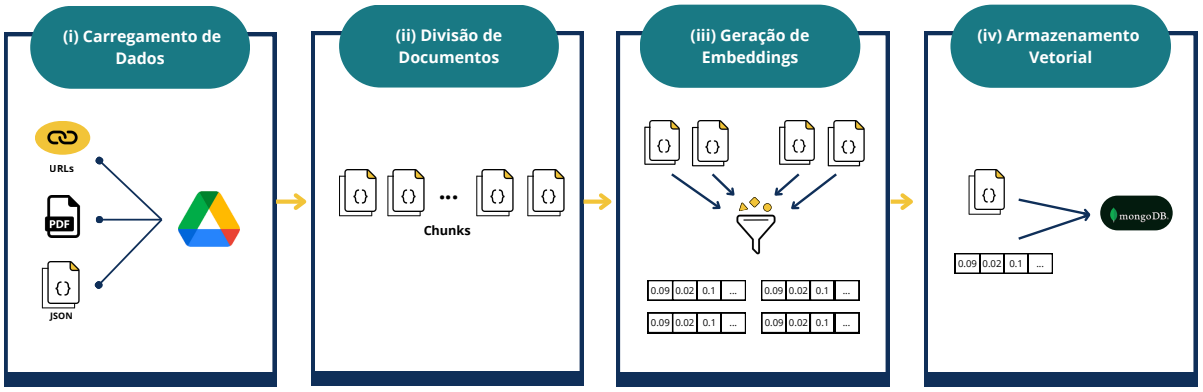


Figura 4.5: Fluxo de indexação de dados para geração e armazenamento de *embeddings*

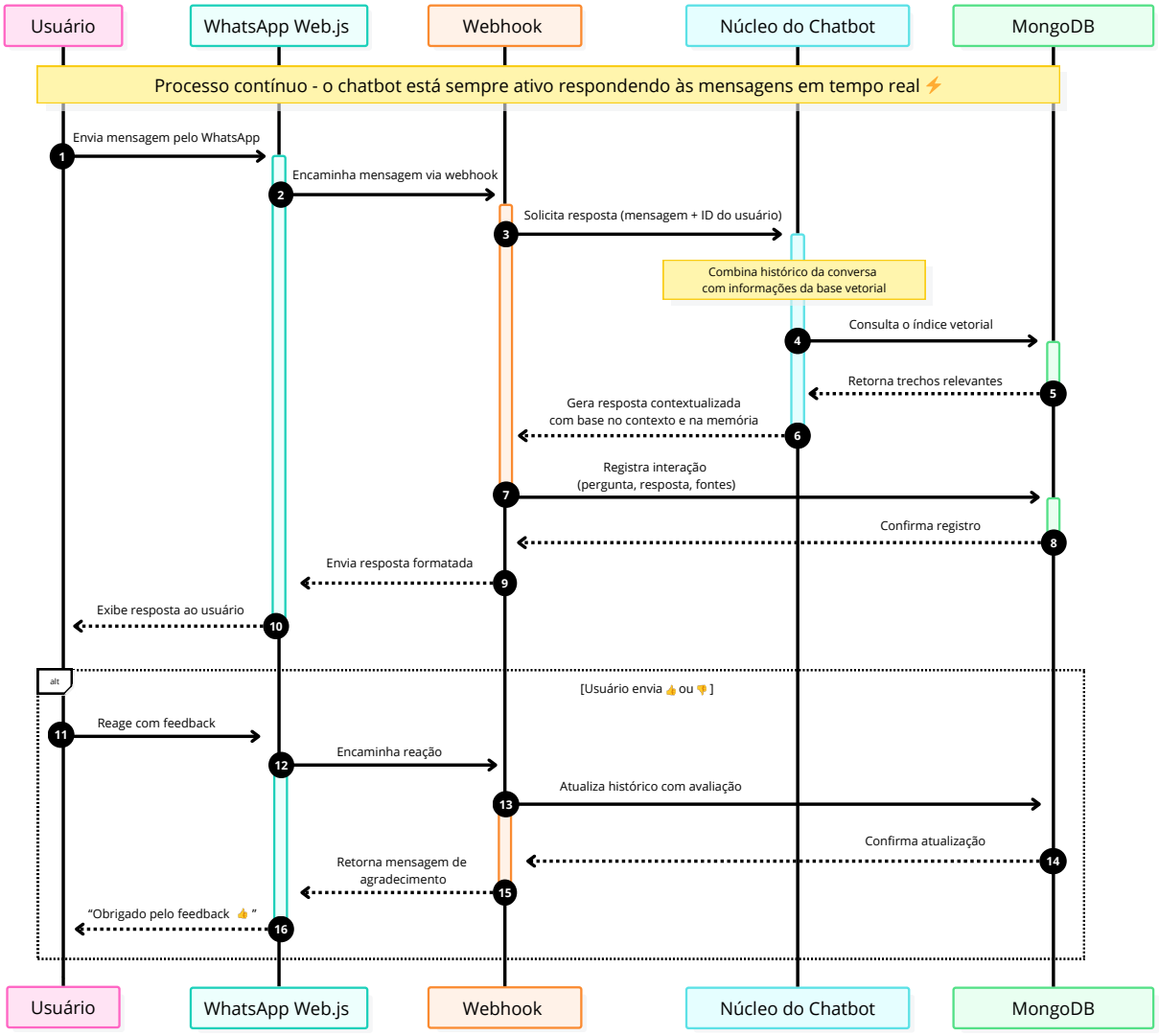


Figura 4.6: Diagrama de sequência relacionado a interação de um usuário com o *chatbot* (Fase 2).

5

Testes e Avaliação

Este capítulo apresenta os resultados das fases de teste e avaliação, detalhando o desempenho do **`TremBot`** tanto em ambiente controlado quanto em cenário real de uso. O processo de validação foi estruturado em duas etapas: (i) a avaliação prévia, focada em testes iterativos de acurácia técnica, e (ii) avaliação com os *stakeholders*, que buscou mensurar a utilidade e a percepção da comunidade acadêmica do IFG - *Campus* Formosa. Adicionalmente, são analisadas métricas para sistemas baseados em arquitetura RAG, como fidelidade, relevância e cobertura, além de latência e custos operacionais.

5.1 Avaliação Prévia

A avaliação prévia consistiu etapa de validação controlada, realizada por meio de testes incrementais organizados em 30 ciclos. O objetivo desta fase foi monitorar a evolução do desempenho do sistema à medida que novos documentos institucionais eram incorporados à base de conhecimento. Para garantir a consistência técnica, cada ciclo de teste seguiu estrutura de análise em dois eixos complementares:

- **Teste de Novo Conhecimento:** focado em verificar a correta assimilação das informações recém-adicionadas;
- **Teste de Regressão:** destinado a reaplicar perguntas de ciclos anteriores para assegurar que a expansão da base de dados não comprometeu a precisão de respostas já consolidadas.

Esta etapa é fundamental para estabelecer uma linha de base de desempenho e garantir a estabilidade do modelo, permitindo que o índice vetorial seja refinado e validado antes da exposição do sistema ao ambiente de produção assistido.

A avaliação prévia do *chatbot* foi conduzida por meio de metodologia de testes incrementais, organizada em ciclos sucessivos, cada um correspondente à adição de novo documento institucional à base de conhecimento. Essa abordagem permitiu mensurar a evolução de desempenho do *chatbot* conforme a expansão da base de dados. Considerando este cenário e os tipos de perguntas definidos na Tabela 5.1, cada ciclo foi estruturado da seguinte maneira:

- **Ciclo 1**: a base de conhecimento contém apenas um documento inicial. O objetivo deste ciclo é estabelecer uma linha de base do desempenho. Aplicam-se perguntas das categorias de **Fato Direto**, **Busca de Detalhe Numérico**, **Extração de Lista**, **Resumo de Conceito** e **Pergunta sem Resposta**, conforme descritas na Tabela 5.1.
- **Ciclo 2 e subsequentes**: um novo documento é incorporado à base de conhecimento, mantendo-se os anteriores. A avaliação é realizada em duas etapas por meio de: (i) **Teste de Novo Conhecimento**, que fortalece a assimilação do conteúdo recém-adicionado, utilizando-se de perguntas das mesmas categorias do Ciclo 1, e (ii) **Teste de Regressão**, onde são reaplicados perguntas de ciclos anteriores, utilizando as categorias de **Consistência da Resposta**, **Consistência do Erro** e **Consistência de Recusa**, para verificar se o novo conteúdo adicionado compromete o desempenho.

Tabela 5.1: Categorias de perguntas para avaliação de desempenho do **`TremBot`**

Tipo de Pergunta	Descrição e Objetivo
Fato Direto	Testar a capacidade do *chatbot* de localizar e extrair uma informação simples e explícita, contida em um único local dentro do texto. Exemplo: *"Qual a data de publicação do PDI?"*.
Busca de Detalhe Numérico	Foco específico na extração de números. É um teste de precisão para dados quantitativos. Exemplo: *"Qual a carga horária total do curso de Tecnologia em Análise e Desenvolvimento de Sistemas*".
Extração de Lista	Avaliar a habilidade do *chatbot* em coletar múltiplos itens de informação. Exemplo: *"Liste os membros que compõem o Colegiado de Curso."*.
Resumo de Conceito	Medir a capacidade de sintetizar informações para explicar um conceito ou processo, exigindo nível maior de "compreensão". Exemplo: *"Explique o que é o Eixo Tecnológico de Informação e Comunicação."*.
Pergunta sem Resposta	Avaliar a capacidade do *chatbot* de identificar os limites de seu conhecimento e se recusar a responder sobre tópicos ausentes nos documentos. Exemplo: *"Qual o valor da mensalidade do curso de Análise e Desenvolvimento de Sistemas?"*.
Consistência da Resposta	Verificar se uma pergunta que foi respondida de forma correta em ciclo anterior continua sendo respondida corretamente após a adição de novos documentos.
Consistência do Erro	Analisar se uma pergunta que foi respondida de forma incorreta em um ciclo anterior ainda resulta em erro.
Consistência de Recusa	Verificar se o *chatbot* mantém negativa em responder a uma "Pergunta sem Resposta" de um ciclo anterior.

5.1.1 Análise de Desempenho

A Figura 5.1 demonstra a evolução da acurácia do *chatbot* ao longo de 30 ciclos de avaliação incremental, totalizando 415 testes executados. Observa-se estabilidade na assertividade do modelo, que apresentou taxa global de acerto superior a 98%. A predominância de avaliações "Corretas" (407 ocorrências), tanto em testes de "Novo Conhecimento" quanto em testes de "Regressão'.

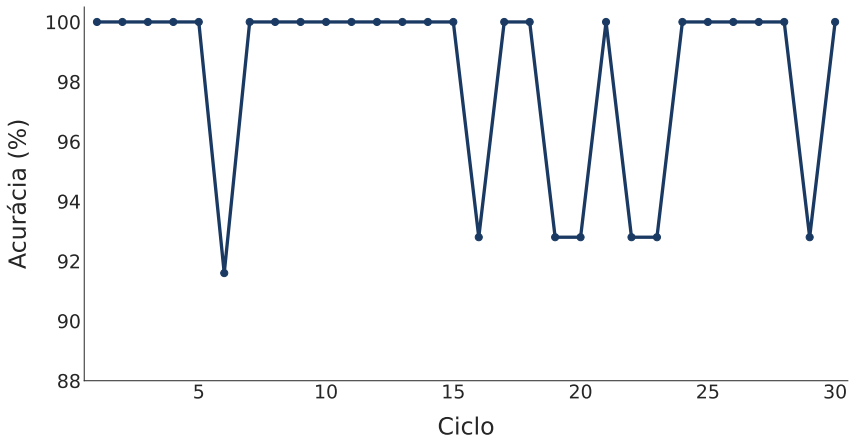


Figura 5.1: Evolução da acurácia do *chatbot* ao longo de 30 ciclos

Eventuais oscilações observadas na Figura 5.1 são compatíveis com o comportamento esperado em testes de regressão, uma vez que a inserção de novos documentos no repositório pode ocasionar ambiguidade semântica ou maior concorrência entre *chunks* relevantes, reduzindo momentaneamente a precisão da recuperação até que o índice vetorial se estabilize ou seja refinado. Ainda assim, as falhas representaram menos de 2% da amostra total e se restringiram a situações de não recuperação da informação, indicando comportamento conservador do modelo que prioriza a abstenção em detrimento da geração de respostas incorretas.

A Figura 5.2 apresenta comparação entre a acurácia obtida nas avaliações de "Novo Conhecimento" e nos testes de "Regressão". Os resultados mostram que a acurácia de regressão se manteve estável, alcançando 100% em quase todos os ciclos, com única exceção no ciclo 23, no qual ocorreu queda para 75%. Esse comportamento indica que, embora o sistema seja consistente ao preservar respostas corretas, situações isoladas podem surgir quando a inserção de novos documentos modifica a hierarquia de relevância dos trechos recuperados, impactando a manutenção de respostas consolidadas.

Em relação a análise de novo conhecimento exibido na Figura 5.2, observam-se oscilações mais frequentes (87% a 90%). Essas variações são esperadas, pois representam o processo de adaptação do modelo à entrada de novos conteúdos, exigindo redistribuição no espaço vetorial e reorganização das similaridades entre documentos. O sistema apresenta estabilidade retrospectiva ao mesmo tempo em que evidencia a importância de etapas incrementais de validação para garantir que a base continue fornecendo respostas precisas.

A Figura 5.3 apresenta a distribuição da acurácia do *chatbot* de acordo com diferentes

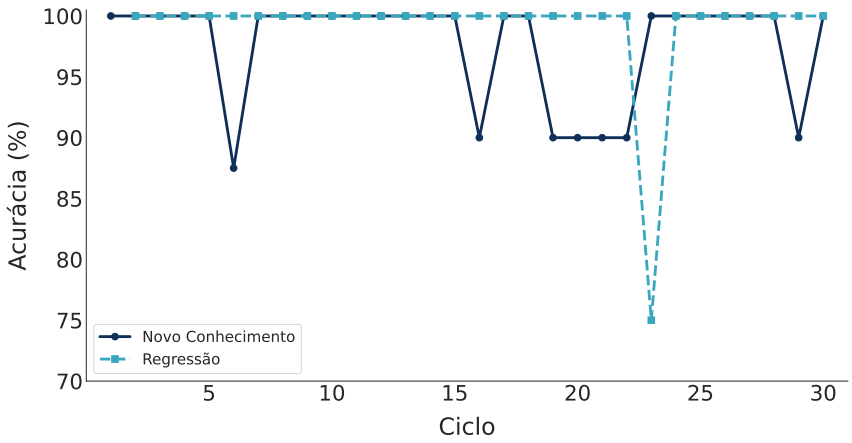


Figura 5.2: Comparação da acurácia em perguntas sobre novo conhecimento e perguntas já realizadas (regressão).

tipos de perguntas. Os resultados demonstram desempenho semelhante em todos os grupos, com pequena variação entre as categorias. Perguntas de "Busca Numérica" obtiveram o melhor resultado entre os grupos com resposta esperada, alcançando 98,1% de acurácia, seguidas por "Fato Direto" e "Extração de Lista", ambas com 96,8%. Já as questões classificadas como "Resumo de Conceito" apresentaram a menor taxa de acurácia, atingindo 96,4%, o que era esperado dada a maior complexidade semântica envolvida na síntese de informações. A categoria "Sem Resposta" atingiu 100%, indicando que o sistema foi capaz de identificar quando a informação solicitada não estava presente na base de conhecimento.

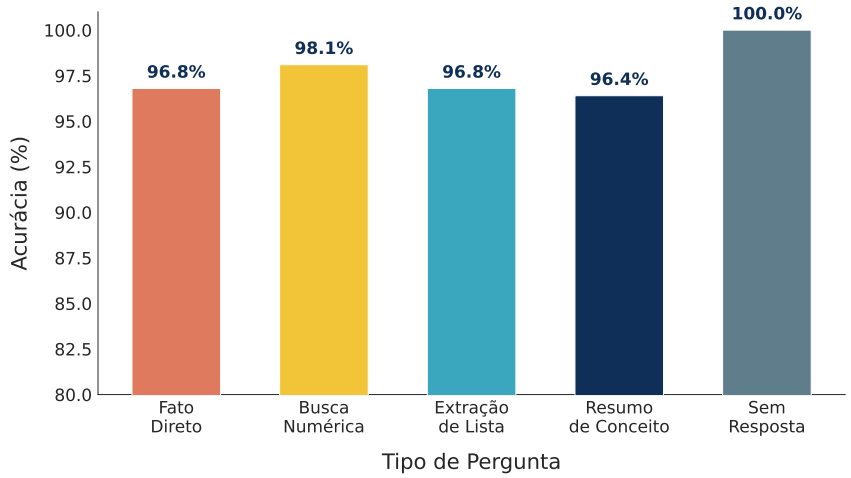


Figura 5.3: Distribuição da acurácia por tipo de pergunta

O gráfico da Figura 5.3 mostra que o modelo apresenta desempenho equilibrado entre diferentes tipos de solicitação, refletindo a qualidade da indexação e do mecanismo de recuperação e geração. Esse comportamento é devido a escolha de uma temperatura baixa no modelo de geração. Ao limitar a aleatoriedade da resposta, o sistema privilegia a fidelidade às informações indexadas, evitando a fabricação de conteúdo quando a base de conhecimento é insuficiente.

A matriz de confusão mostra a eficácia do modelo na discriminação entre domínios

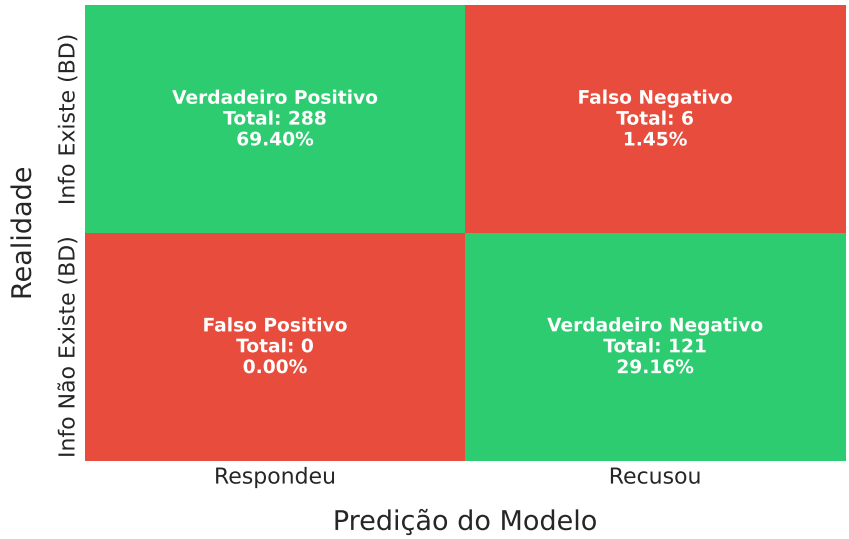


Figura 5.4: Matriz de confusão resultante da avaliação prévia.

de conhecimento dominados e desconhecidos. Como pode ser observado na Figura 5.4, a concentração de registros na diagonal principal confirma a precisão do modelo na recuperação de respostas existentes ("Verdadeiros Positivos") e na recusa correta de perguntas fora do escopo ("Verdadeiros Negativos"). Um ponto fundamental a ser observado na Figura 5.4 é a inexistência de incidência de "Falsos Positivos", que representam as alucinações, em contraste com uma taxa superior de "Falsos Negativos", que representam as omissões, ou seja, onde o sistema não foi capaz de determinar uma resposta.

5.2 Avaliação dos *Stakeholders*

Esta etapa marca a transição dos testes controlados para a validação do **`TremBot`** em ambiente de produção assistido. O objetivo foi submeter o sistema a cenário real e heterogêneo, contando com a participação de alunos, servidores e comunidade externa. Ao longo deste período, foram processadas 1.059 interações, permitindo análise sobre a utilidade prática da ferramenta e sua eficácia na democratização do acesso à informação institucional.

Para viabilizar essa análise, adotou-se abordagem quantitativa e qualitativa baseada no monitoramento sistemático das interações via *WhatsApp*. O procedimento metodológico estruturou-se nos seguintes eixos:

- **Identificação de Perfil:** O sistema registrou automaticamente a distribuição dos usuários entre alunos, servidores e comunidade externa para validar o alcance da ferramenta.
- **Categorização de Demandas:** As interações foram agrupadas por tipos de perguntas e termos mais frequentes para identificar as principais necessidades informacionais da comunidade.

- **Monitoramento de Performance:** Foram extraídas métricas de latência, distinguindo o tempo de busca na base vetorial do tempo de geração de resposta pelo LLM.
- **Avaliação de Qualidade:** Utilizou-se a autoavaliação do modelo para medir a Fidelidade (consistência com a fonte), Relevância (aderência à pergunta) e Cobertura (presença da informação na base).
- **Coleta de *Feedback*:** Após cada resposta, o usuário pode reagir com *feedback* positivo ou negativo via *WhatsApp*, permitindo mensurar o índice de satisfação direta.

5.2.1 Análise de Desempenho

A avaliação do *chatbot* com usuários reais foi conduzida em ambiente de homologação assistido, totalizando 1.059 interações. Conforme pode ser observado na Figura 5.5, a distribuição do perfil de uso revela predominância de demandas associadas ao ingresso e à vida acadêmica, sugerindo forte adesão por parte da comunidade externa e de discentes regulares. Essa característica valida o alcance do sistema de conversação, demonstrando que **`TremBot`** cumpriu seu objetivo de democratizar o acesso à informação institucional, não se restringindo apenas a usuários com conhecimento técnico prévio ou acesso facilitado, como servidores técnico-administrativos e docentes.

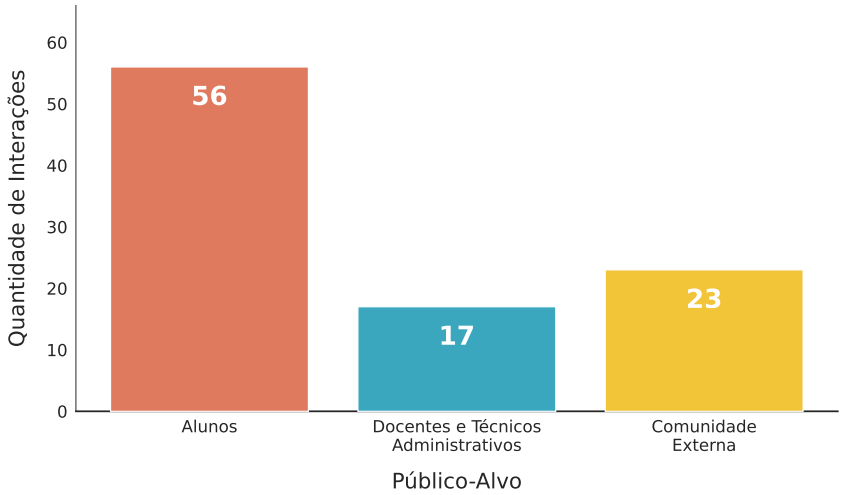


Figura 5.5: Distribuição quantitativa dos usuários por perfil.

Ao analisarmos as intenções dos usuários exibidas na Figura 5.6, observa-se que a categoria “Outros” foi a mais expressiva, concentrando 515 interações (48,6% do total), o que evidencia a heterogeneidade das demandas. Diferente de um sistema focado em um único nicho, os usuários utilizaram o *chatbot* para sanar dúvidas sobre vasta gama de tópicos institucionais. Entre os temas específicos, o tópico “Cursos” liderou com 210 ocorrências, confirmando o

interesse prioritário nos cursos oferecidos pela instituição. Em seguida, a busca por “Calendário e Horários” (121 incidências) destacou-se como demanda recorrente, típica da rotina de alunos que buscam agilidade nas informações relacionadas ao ensino.

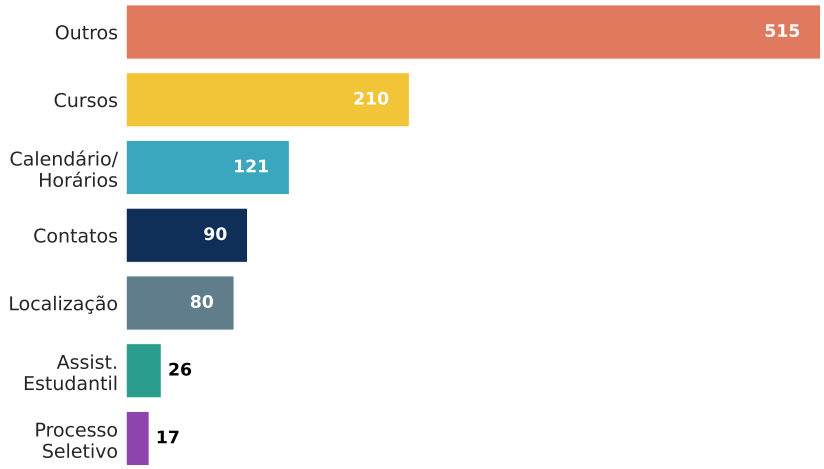


Figura 5.6: Distribuição do volume de interações por tópico

De forma complementar na compreensão das interações mais frequentes, a nuvem de palavras é uma representação visual onde termos mais buscados aparecem maiores e mais destacadas. Como pode ser observado na nuvem de palavras da Figura 5.7, a procura por termos específicos, como Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) (51 ocorrências), NAPNE (45 ocorrências), e "Horário"(37 ocorrências) mostram que as dúvidas da comunidade não são genéricas, mas sim contextualizadas e específicas.

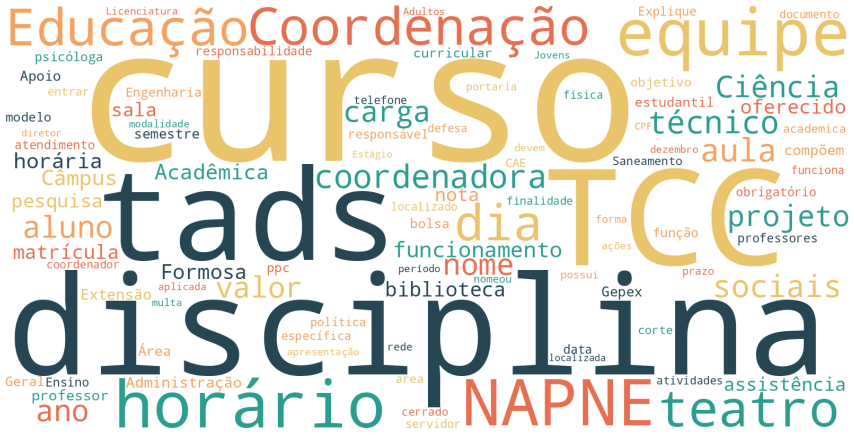


Figura 5.7: Nuvem de palavras gerada a partir dos termos mais frequentes nas interações dos *stakeholders*.

Em relação ao desempenho relacionado ao tempo de resposta, é notável que a latência em arquiteturas RAG é determinada por duas etapas: (i) o tempo de busca na base vetorial, e (ii) o tempo de geração da resposta pelo LLM. Nesse sentido, a Figura 5.8 mostra a distribuição da latência ao longo de 50 interações. Apesar de picos isolados, a maioria das interações se manteve abaixo do limite de 5 segundos. Os picos de latência observados, alguns ultrapassando

30s, são atribuídos ao tempo de geração. O tempo de busca, por sua vez, manteve-se baixo, entre 500-650 ms, demonstrando a eficiência da indexação vetorial no MongoDB Atlas e da busca híbrida implementada.

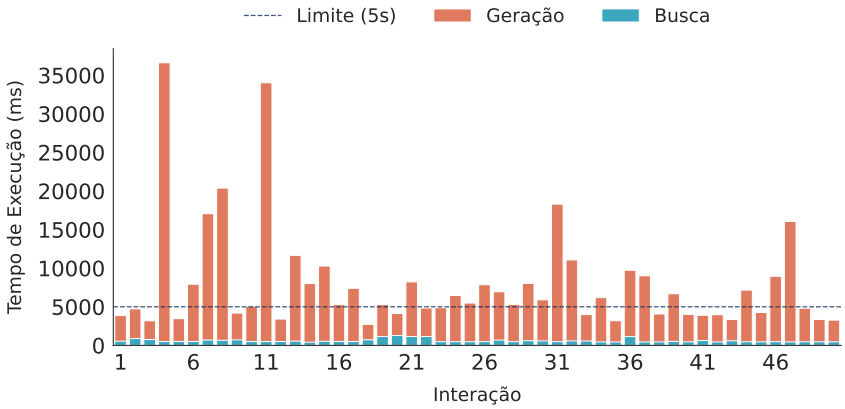


Figura 5.8: Análise dos tempos de resposta do sistema ao longo das últimas 50 (cinquenta) interações.

A análise da latência média, apresentada na Figura 5.9, permite correlacionar a latência com a complexidade semântica da consulta. O tipo de pergunta "Extração de Lista" apresentou o maior tempo médio total (10.533 ms), seguido por "Resumo de Conceito" (8.473 ms). O aumento da latência nessas categorias é explicado pelo maior esforço computacional do LLM em tarefas que exigem agregação de múltiplos *chunks* de texto ou síntese de grandes volumes de informação.

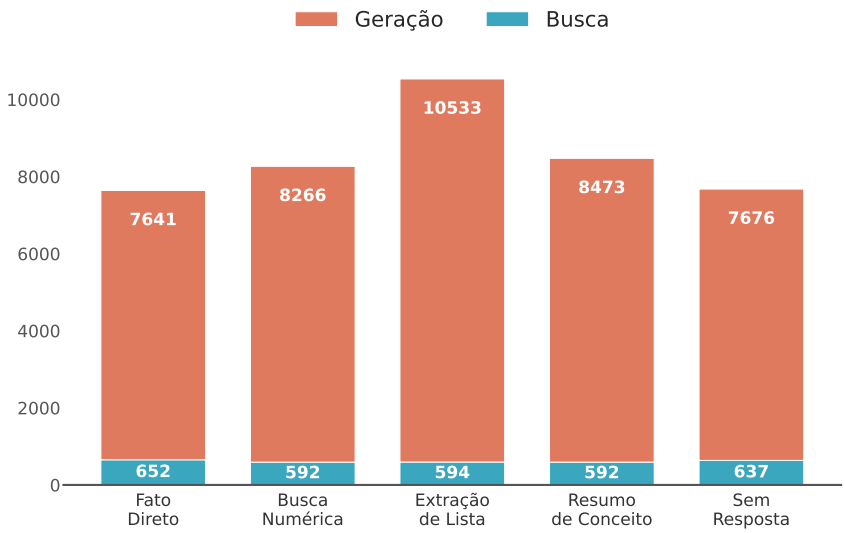


Figura 5.9: Tempo médio de resposta por tipo de pergunta.

Em relação a avaliação da recuperação de contexto, a eficácia depende da capacidade do módulo de recuperação em localizar, dentro da base vetorial, os trechos *(chunks)* que contêm a informação necessária para formulação da resposta. Para mensurar essa capacidade, analisou-se a distribuição da cobertura, conforme apresentado na Figura 5.10.

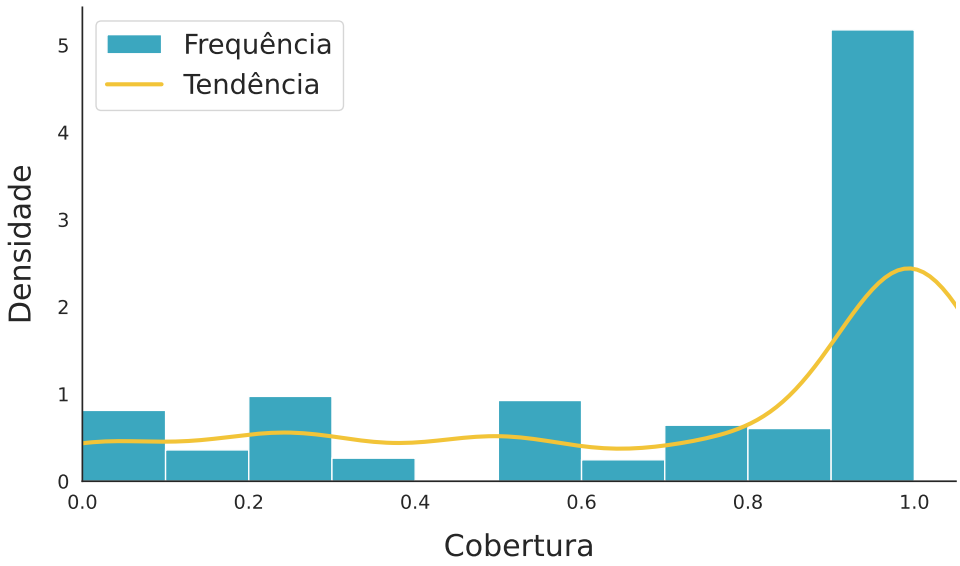


Figura 5.10: Distribuição de densidade de Cobertura nas interações dos *stakeholders*.

O gráfico da Figura 5.10 exibe o histograma de densidade combinado com uma curva de tendência, onde o eixo das abscissas representa o grau de cobertura (variando de 0 a 1) e o eixo ordenadas a frequência de ocorrência. Observa-se distribuição assimétrica à esquerda, com concentração de interações situadas no intervalo entre 0,9 e 1,0.

Esse comportamento evidencia que, para a maioria das perguntas realizadas pelos *stakeholders*, o mecanismo de busca híbrida foi capaz de recuperar contextos relevantes. A predominância de scores próximos a 1,0 corrobora a eficiência das estratégias de *chunking* e indexação detalhadas no Capítulo 4.

Ainda analisando tendências relacionadas à cobertura presentes na Figura 5.10, a existência de densidade nas faixas inferiores (entre 0,0 e 0,4) não representa falha do sistema, mas sim o cumprimento do Requisito de Negócio 03 (RN03). Esses valores correspondem às interações classificadas como "Sem Resposta" ou perguntas fora do escopo institucional, onde a baixa similaridade vetorial impediu a recuperação de documentos, levando **`TremBot`** a se abster de responder, evitando alucinações.

Além da capacidade de recuperar informações e da velocidade de resposta, é fundamental mensurar a qualidade semântica do conteúdo gerado. Para isso, utilizou-se duas métricas centrais em sistemas RAG: (i) Fidelidade, que mede o grau de consistência entre resposta gerada e os documentos recuperados, e (ii) Relevância, que avalia se a resposta atende às necessidades relacionadas a pergunta do usuário. A Figura 5.11 apresenta a visão global das métricas de fidelidade e relevância para as interações onde o sistema gerou uma resposta, ou seja, excluindo casos de recusa por ausência de informação. O modelo atingiu taxa de 96,8% de fidelidade, indicando que o *chatbot* manteve-se coerente aos documentos institucionais na maioria das interações, superando o limiar de 95% estabelecido no Requisito Não Funcional 02 (RNF02). De forma simultânea, a métrica de relevância alcançou 96,0%, demonstrando capacidade do LLM em interpretar as nuances da linguagem natural.

Figura 5.11: Qualidade das respostas geradas, através das métricas de fidelidade e relevância.

Aprofundando a análise, a Figura 5.12 e Figura 5.13 segmentam o desempenho por categoria de pergunta e perguntas frequentes. Nota-se uma consistência nas categorias que exigem recuperação factual, como aquelas presentes nas categorias de "Fato Direto" e "Busca Numérica", e raciocínio estruturado em perguntas categorizadas em "Extração de Lista", com métricas variando entre 0,88 e 0,97. Essa uniformidade também foi observada na análise por tópicos mais frequentes, onde assuntos com alta variabilidade, como "Processo Seletivo", mantiveram médias superiores a 0,92.

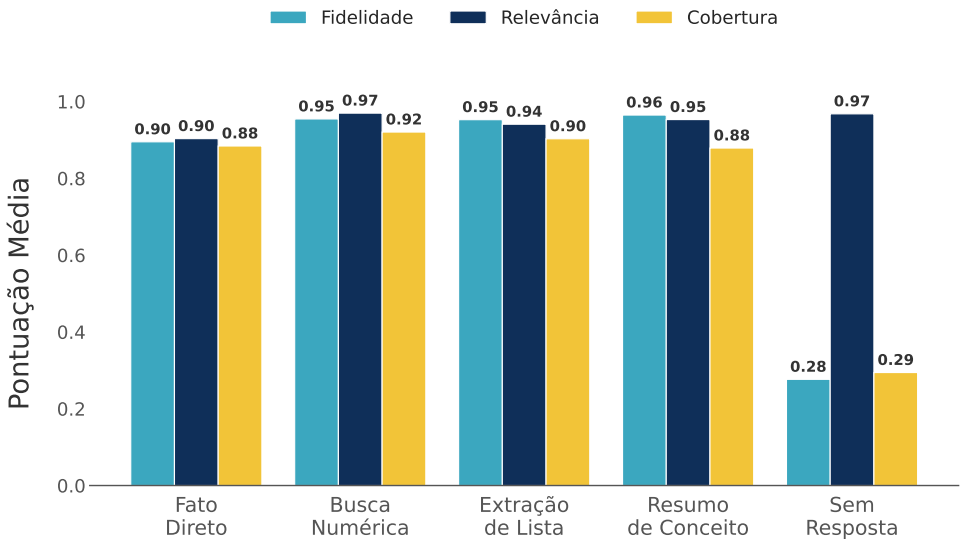


Figura 5.12: Média das métricas de desempenho por tipo de pergunta.

Fonte: autoria própria.

Um destaque crítico nesta análise reside na categoria "Sem Resposta", onde se observa uma queda nas métricas de cobertura e fidelidade, que atingiram 0,29 e 0,28, respectivamente, em contraste com a alta relevância (0,97). A baixa pontuação de fidelidade é justificável neste cenário: uma vez que a resposta gerada consiste em uma negativa orientada pelo sistema de conversação, ela não deriva textualmente dos documentos recuperados, os quais, como indicado pela baixa cobertura, foram considerados insuficientes ou irrelevantes. Portanto, esse comportamento não sinaliza falha, mas a eficácia do mecanismo de segurança que, ao não localizar evidências na base de conhecimento, opta pela recusa da resposta em detrimento da alucinação. Esse comportamento confirma o cumprimento da Regra de Negócio 03 (RN03), garantindo que o

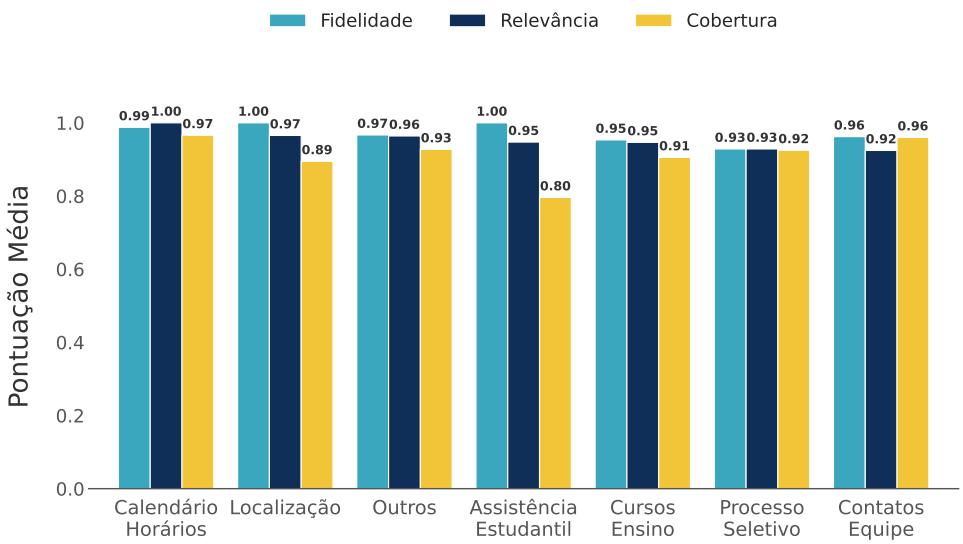


Figura 5.13: Média das métricas de desempenho por tópicos mais frequentes.

usuário seja informado quando a informação requisitada não não está presenta nos documentos institucionais.

A consolidação dessas métricas reflete-se na acurácia do sistema, apresentada na Figura 5.14. **`TremBot`** atingiu taxas de acerto superiores a 96% em todas as categorias. Ressalta-se a acurácia de 100% na categoria "Sem Resposta", evidenciando a capacidade do sistema de conversação em evitar a geração de conteúdo inexistente.

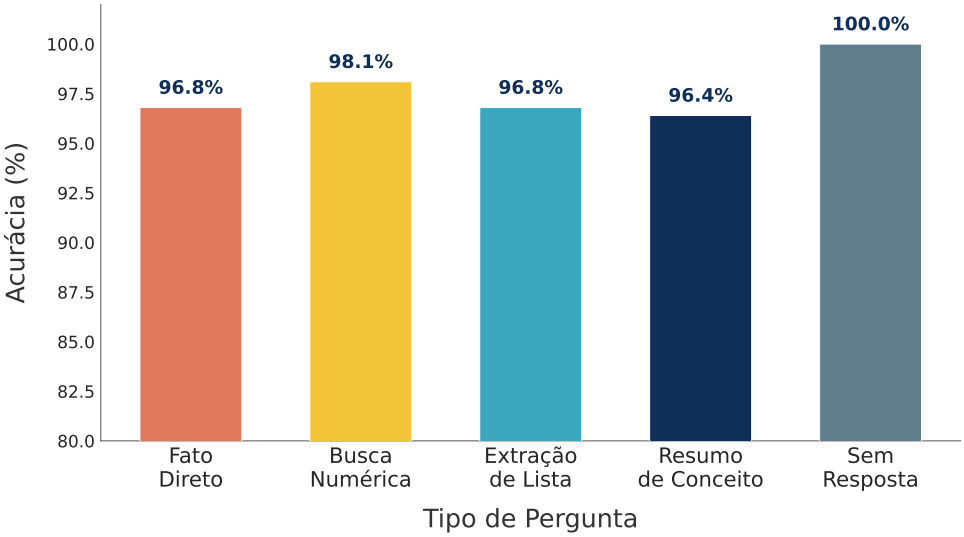


Figura 5.14: Acurácia do sistema por categoria de pergunta.

Fonte: autoria própria.

A validação final consistiu na análise da percepção subjetiva dos usuários, coletada por meio de *feedback* (reações de "positivo" ou "negativo"), disponibilizado após cada resposta gerada pelo sistema de conversação. Os dados indicam predominância de avaliações positivas (47 ocorrências) em relação às negativas (11 ocorrências). Este índice de aprovação reforça a correlação entre a alta fidelidade (96,8%) e relevância (96,0%) das respostas geradas e a utilidade

percebida pela comunidade acadêmica (satisfação positiva de 81,0%.).

Ao analisar as avaliações negativas, que representam 19% do total, é importante entender o contexto do comportamento do sistema. O sistema tende a recusar responder (RN03) quando considera que os documentos disponíveis não têm cobertura suficiente para responder com segurança. Embora essa escolha seja tecnicamente correta e ajude a evitar respostas incorretas, pode deixar o usuário frustrado por não obter a resposta que precisava, o que pode levar ao *feedback* negativo. Adicionalmente, é importante notar que o baixo volume de *feedbacks* (58 registros) em relação ao total de interações (1.059), é típico em sistemas onde o usuário é convidado a avaliar suas percepções. Ainda assim, o saldo favorável indica que a integração via *WhatsApp* (RNF03) e a precisão do modelo atenderam satisfatoriamente às expectativas da comunidade.

5.3 Custo de Implantação

Nesta Seção, apresenta-se análise financeira para implementação do sistema conversacional **`TremBot`** em ambiente de produção. Embora o desenvolvimento deste projeto tenha utilizado versões gratuitas das ferramentas, a implantação em cenário real exige a adoção de planos que garantam estabilidade, suporte e conformidade com os Requisitos Não Funcionais, descritos no Capítulo 4 (Subseção 4.4.3).

5.3.1 Processamento (Decomposição de *Tokens*)

Para calcular a viabilidade econômica, é necessário decompor o custo de uma interação. Para o núcleo de IA e o armazenamento vetorial, o sistema utiliza o modelo de cobrança por uso (*Pay-as-you-go*), o que garante que a instituição pague apenas pelo volume processado e permite escalabilidade eficiente.

A arquitetura implementada no **`TremBot`** utiliza o LLM não apenas para gerar respostas, mas também para realizar a autoavaliação de métricas de qualidade em tempo real. Cada interação do usuário desencadeia quatro processos distintos de processamento, ou seja, a geração da resposta principal e três avaliações automáticas, relacionadas a fidelidade, relevância e cobertura.

Diante desse contexto, o cálculo do custo unitário por mensagem enviada por algum *stakeholder* é dividido entre o fluxo de geração (relacionado ao RAG) e o fluxo de avaliação.

1. Fluxo de Geração (Resposta ao Usuário)

   Neste fluxo, o modelo processa o *prompt* de sistema, o histórico da conversa e os fragmentos de conhecimento recuperados.

   - *Prompt* do *chatbot*: o modelo de *prompt* estruturado enviado (instruções, persona e exemplos) possui aproximadamente 1.200 *tokens*.
   - Contexto Recuperado (*top_k=8*): considerando que cada nó possui em média 150 a 200 *tokens*, os 8 nós somam cerca de 1.400 *tokens*.

- Histórico (*Buffer*): o limite configurado de 3.000 *tokens* para a memória conversacional.
- Resposta Gerada: estimativa média de 250 *tokens*.

Total Estimado (Geração): 5.600 *tokens* de entrada e 250 de saída. Custo por Geração: US$ 0,00056 + US$ 0,00010 = US$ 0,00066.

2. Fluxo de Avaliação:

   O sistema executa três avaliadores assíncronos que processam a pergunta, a resposta gerada e os mesmos 8 nós de contexto.

   - Entrada Total (3 avaliadores): Cada avaliador processa novamente o contexto e a resposta, totalizando cerca de 4.800 *tokens* de entrada extras.
   - Saída Total (3 avaliadores): Respostas curtas de métricas somam aproximadamente 50 *tokens*.

   Custo por Avaliação: US$ 0,00048 + US$ 0,00002 = US$ 0,00050.

5.3.2 Armazenamento

A persistência de dados do **`TremBot`** é dividida em três coleções principais, cada uma com impacto específico no consumo de recursos do banco de dados vetorial. Para este cálculo, considera-se o plano MongoDB Atlas *Serverless*, que cobra por unidade de armazenamento e operações de leitura/escrita.

1. Coleção de Documentos (Base de Conhecimento)

   Cada documento (nó) armazena o texto, metadados e o vetor de *embedding*.

   - Estrutura do Vetor: O modelo **text-embedding-004** gera vetores de 768 dimensões. Como cada dimensão é um número de ponto flutuante (`float64`), o vetor ocupa cerca de 6 KB por documento.
   - Metadados e Texto: Somando o conteúdo textual e o campo **_node_content**, cada entrada ocupa em média 10 KB.
   - Total: Com uma base de 500 nós (média para o *campus*), o armazenamento total é de apenas 5 MB, o que entra na faixa mínima de custo do Atlas.

2. Coleção de Interações (*Logs* e Métricas)

   Esta é a coleção de maior expansão, pois registra cada pergunta, resposta e as métricas de avaliação. O tamanho médio de um documento de interação completo (incluindo o *array* de fontes recuperadas) ocupa cerca de 2KB.

3. Coleção de Sessões de Usuário

   Coleção que gerencia o estado do usuário e o `message_count`. Cada documento ocupa menos de 0,5 KB.

5.3.3 Busca Híbrida

O custo operacional mais relevante no banco não é o armazenamento, mas a busca híbrida executada a cada interação:

- Busca Vetorial: o Atlas processa o *embedding* da pergunta contra o índice vetorial.
- Busca de Texto (BM25): Refina os resultados por palavras-chave.

No modelo *serverless*, o custo para 1.000 buscas complexas (híbridas) com o retorno de 8 nós (**similarity_top_k=8**) é estimado em US$ 9,00 mensais. Este valor considera as RPUs necessárias para manter a latência de busca estável. Conforme observado na Figura 5.9, a latência para a operação de busca manteve-se consistentemente entre 500ms e 650ms, o que demonstra que a infraestrutura *serverless* é capaz de atender a complexidade da busca semântica sem comprometer a experiência do usuário, mesmo em períodos de maior volume de interações.

5.3.4 Infraestrutura de Hospedagem e Custo de Indexação Inicial

A hospedagem do *backend* (API Flask e cliente Node.js) representa o custo fixo mais significativo. Duas abordagens principais podem ser adotadas:

- Hospedagem em Nuvem: optar por um Servidor Virtual Privado (VPS) em provedores como *DigitalOcean*[1] ou *Amazon Web Services*[2]: uma instância básica (1 vCPU e 2GB RAM) custa cerca de US$ 12,00 mensais. Esta opção oferece alta disponibilidade e facilidade de manutenção sem necessidade de investimento inicial em *hardware*.
- Hospedagem Local: caso seja utilizado *hardware* próprio, o custo mensal de hospedagem é eliminado. Entretanto, o investimento inicial em *hardware* com requisitos mínimos (*Quad-core*, 8GB RAM) varia entre R$ 3.500,00 e R$ 5.000,00.

Para o processamento de texto (*LlamaParse*), o plano gratuito cobre até 1.000 páginas/mês. Dessa forma, para uma base institucional padrão, o custo é zero no *setup* inicial. Considerando a geração de *embeddings* (**text-embedding-004**), o custo para converter toda base de conhecimento em vetores numéricos é irrisório, estimado em menos de US$ 0,01 para toda a biblioteca de documentos do *campus*.

[1] https://www.digitalocean.com/
[2] https://aws.amazon.com/

5.3.5 Projeção de Operação Mensal

Para consolidar a viabilidade econômica do **`TremBot`**, a Tabela 5.2 apresenta a projeção de custos para uma operação de 1.000 interações mensais, baseada no volume observado durante a fase de avaliação com os *stakeholders*.

Tabela 5.2: Comparação de custos operacionais estimados: implementação base vs. implementação com monitoramento de métricas

Item de Custo	Sem Métricas	Com Métricas
Custo Unitário (IA)	$ 0,00066	$ 0,00116
Custo p/ 1.000 msgs (IA)	$ 0,66 (R$ 3,56)	$ 1,16 (R$ 6,26)
Hospedagem (VPS)	$ 12,00 (R$ 64,80)	$ 12,00 (R$ 64,80)
Banco (Atlas *Serverless*)	$ 9,00 (R$ 48,60)	$ 9,00 (R$ 48,60)
TOTAL MENSAL ESTIMADO	**R$ 116,96**	**R$ 119,66**

Nota: Estimativa baseada em um volume de 1.000 interações/mês e câmbio de referência de US$ 1,00 = R$ 5,40. O cenário "Com Métricas"inclui o processamento das métricas de fidelidade, relevância e cobertura.

A Figura 5.15 apresenta a distribuição percentual dos custos operacionais mensais. Observa-se que a infraestrutura de hospedagem e o banco de dados representam a maior parcela do investimento fixo (aproximadamente 95%), enquanto o processamento de linguagem natural via LLM, mesmo com o módulo de avaliação, representa impacto financeiro marginal no orçamento total.

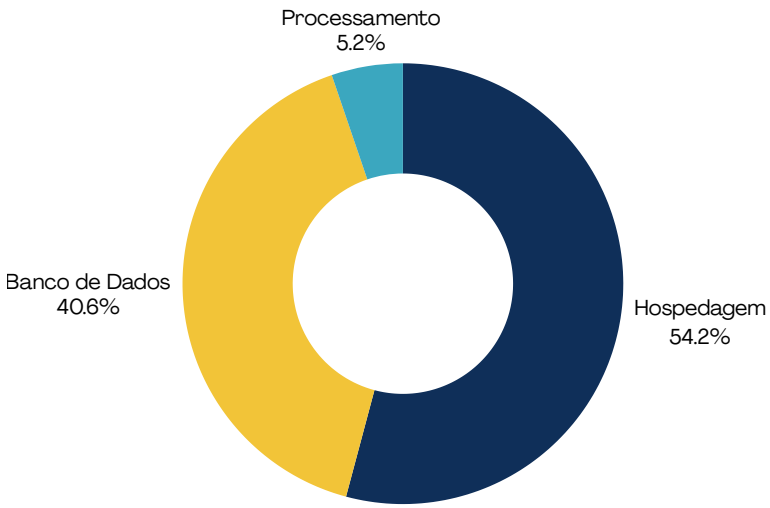


Figura 5.15: Distribuição percentual dos custos operacionais mensais estimados para implementação do sistema de conversação em ambiente de produção.

5.3.6 Considerações

A estimativa de 1.000 interações reflete o uso real documentado no mês de homologação da ferramenta. A escolha do modelo **Gemini 2.5 Flash** justifica-se pelo equilíbrio entre custo e

performance, sendo significativamente mais acessível que modelos concorrentes.

É importante destacar que as métricas de avaliação em tempo real representam diferencial técnico do sistema. Embora desativar o módulo de métricas reduza o custo de processamento de IA de US$ 1,16 para US$ 0,66, o investimento adicional de aproximadamente R$ 2,70 (convertido da diferença de US$ 0,50) é ínfimo diante da segurança institucional garantida pela mitigação de alucinações, consoante com o RN03.

Em relação à infraestrutura, a hospedagem em nuvem é recomendada por oferecer interessante custo-benefício, garantindo o requisito não funcional relacionado à disponibilidade (RNF01), aliada a não necessidade de manutenção de servidores físicos.

Por fim, os valores apresentados demonstram que a arquitetura RAG proposta é financeiramente acessível, permitindo implementação do sistema de conversação especializado com baixo custo de manutenção. Esta análise de custos, somada aos resultados de acurácia obtidos, valida a eficiência da solução desenvolvida, cujas considerações finais e perspectivas de evolução estão descritas no Capítulo 6.

6
Conclusão

O presente trabalho apresentou o desenvolvimento do **`TremBot`**, um sistema de conversação especializado baseado na arquitetura RAG, projetado para atender às demandas de informação do IFG *Campus* Formosa. Ao finalizar este ciclo, conclui-se que o projeto atingiu integralmente o objetivo de desenvolver solução capaz de mitigar alucinações e fornecer respostas precisas com baixo custo operacional.

A eficácia da implementação é evidenciada pelo cumprimento dos Requisitos Funcionais e Não Funcionais estabelecidos. O sistema não apenas realiza a recuperação de informações de forma eficiente, mas, por meio do módulo de autoavaliação em tempo real, garante a conformidade com a Regra de Negócio 03 (RN03), mantendo índices de fidelidade superiores a 95% nos testes realizados. A integração do *backend* em *Flask* com o banco de dados vetorial *MongoDB Atlas* demonstrou estabilidade, mantendo a latência dentro dos limites aceitáveis para uma experiência de usuário fluida.

Do ponto de vista financeiro, a análise de custos de implantação confirmou a viabilidade e acessibilidade do sistema de conversação. A arquitetura foi estrategicamente desenhada para utilizar recursos de escala (*pay-as-you-go*), permitindo que o custo mensal por mil interações seja de aproximadamente US$ 22,00. Ressalta-se que, embora o monitoramento de métricas represente custo adicional de processamento, este investimento é fortemente justificado pela segurança institucional e pela qualidade das respostas, requisitos indispensáveis para uso e adoção em um ambiente acadêmico.

Por fim, **`TremBot`** cumpre sua função social ao democratizar o acesso à informação institucional, removendo barreiras burocráticas no atendimento ao aluno. Mais do que um produto isolado, este projeto serve como modelo que pode ser replicável para outras instituições públicas que buscam adotar IA com responsabilidade fiscal e técnica, provando que a democratização da informação pode ser alcançada, mesmo sem grandes investimentos em infraestrutura.

Como sugestões para trabalhos futuros, visando a evolução contínua da ferramenta, propõem-se:

- **Multimodalidade:** expandir a capacidade do sistema para processar e gerar respostas a partir de imagens, áudios e vídeos.

- **Internacionalização:** implementar suporte a outros idiomas para auxiliar intercambistas e pesquisadores estrangeiros.
- **Integração Sistêmica:** conectar o sistema diretamente a bases de dados acadêmicas como o Sistema Unificado de Administração Pública (SUAP) e o *Moodle* para consultas personalizadas de notas e horários.
- **Otimização de *Retrieval*:** Explorar novos métodos de indexação e técnicas avançadas de *reranking* no *LlamaIndex* para aprimorar a precisão da busca híbrida.

A

Apêndice

A.1 Matriz de Rastreabilidade de Regras de Negócio

Tabela A.1: Matriz 3: Rastreabilidade de Regras de Negócio (RN)

ID	Descrição da Regra	Componente da Solução Implementada	Justificativa da Rastreabilidade
RF01	O *chatbot* deve comunicar-se com o usuário sempre em Português.	*Prompt* de Sistema	As instruções de comportamento do modelo de IA são definidas em um *prompt*, escrito em português, que o instrui a se comunicar exclusivamente neste idioma.
RF02	A fonte de informação do *chatbot* deve ser exclusivamente os documentos institucionais oficiais do IFG fornecidos como base de conhecimento.	Arquitetura RAG	O design do sistema, baseado em RAG, intrinsecamente limita as respostas às informações recuperadas da base de conhecimento. Não há mecanismo para consultar fontes externas.

Tabela A.1 – continuação

ID	Descrição da Regra	Componente da Solução Implementada	Justificativa da Rastreabilidade
RF03	Caso a informação solicitada pelo usuário não seja encontrada nos documentos da base de conhecimento, o *chatbot* deve informar explicitamente que não pôde localizar a resposta, em vez de tentar inferir ou criar uma informação.	*prompt* de Sistema	O *prompt* contém uma diretriz clara para o modelo de IA sobre como agir na ausência de informações, instruindo-o a comunicar a limitação ao usuário de forma transparente.
RF04	O *chatbot* deve atender às demandas informacionais de toda a comunidade acadêmica e da comunidade externa..	Canal de Acesso (*WhatsApp*)	Ao ser implementado em uma plataforma de comunicação pública e de amplo alcance como o *WhatsApp*, o sistema torna-se acessível a qualquer pessoa que possua o contato, atendendo tanto ao público interno quanto ao externo.

A.2 Matriz de Rastreabilidade de Requisitos

Tabela A.2: Matriz 1: Rastreabilidade de Requisitos Funcionais (RF)

ID	Descrição do Requisito	Componente da Solução Implementada	Justificativa da Rastreabilidade
RF01	*chatbot* deve permitir a ingestão de documentos a partir de uma fonte de dados centralizada (Google Drive).	*Pipeline* de Indexação	Um script automatizado conecta-se à pasta designada no Google Drive para baixar os documentos oficiais que servirão como base de conhecimento.

Tabela A.2 – continuação

ID	Descrição do Requisito	Componente da Solução Implementada	Justificativa da Rastreabilidade
RF02	Segmentar os documentos em fragmentos de texto menores e semanticamente coerentes (*chunks*).	Módulo de Processamento de Texto	Durante a indexação, os documentos são divididos em "*chunks*"para otimizar a busca por trechos específicos de informação, aumentando a precisão das respostas.
RF03	Converter cada fragmento de texto em um vetor numérico (*embedding*).	Módulo de Geração de *embedding*	Um modelo de IA específico transforma os *chunks* de texto em vetores numéricos, que capturam o significado semântico do conteúdo para permitir buscas por similaridade.
RF04	Armazenar os fragmentos de texto e seus respectivos *embedding* em uma base de dados vetorial (MongoDB Atlas).	Banco de Dados Vetorial	A solução utiliza o MongoDB Atlas para persistir os vetores e os textos associados, garantindo uma recuperação rápida e eficiente durante as buscas semânticas.
RF05	Utilizar um LLM para interpretar a semântica da pergunta do usuário e gerar as respostas.	Núcleo de IA (Google Gemini)	O sistema emprega um Modelo de LLM para processar as perguntas e gerar respostas coesas e naturais, com base no contexto fornecido.
RF06	Realizar uma busca semântica na base de dados vetorial para encontrar os trechos de documentos mais relevantes.	Mecanismo de Busca Vetorial	Ao receber uma pergunta, o sistema a converte em um vetor e a compara com os vetores armazenados no banco de dados para localizar os *chunks* de informação mais relevantes.

Tabela A.2 – continuação

ID	Descrição do Requisito	Componente da Solução Implementada	Justificativa da Rastreabilidade
RF07	Implementar a arquitetura RAG.	Orquestrador RAG	O fluxo principal do sistema combina a busca semântica **RF06** com a geração de texto pelo LLM **RF05**, garantindo que as respostas sejam baseadas nos documentos recuperados.
RF08	Permitir que o usuário envie perguntas em linguagem natural.	Interface de Conversação (*WhatsApp*)	O sistema é acessível via *WhatsApp*, permitindo que os usuários interajam de forma natural e intuitiva, sem a necessidade de comandos específicos.
RF09	Permitir que o usuário receba respostas geradas a partir da base de conhecimento institucional.	Interface de Conversação (*WhatsApp*)	As respostas formuladas pelo Núcleo de IA são enviadas de volta ao usuário dentro da mesma conversa no *WhatsApp*.
RF10	Manter o histórico e o contexto da conversa durante uma sessão de uso.	Módulo de Gerenciamento de Memória	Para cada usuário, o sistema mantém um histórico das trocas de mensagens, o que permite compreender perguntas subsequentes que dependem do diálogo anterior.

Tabela A.3: Matriz 2: Rastreabilidade de Requisitos Não Funcionais (RNF)

ID	Descrição do Requisito	Componente da Solução Implementada	Justificativa da Rastreabilidade
RNF01	O *chatbot* deve operar com uma disponibilidade de 99% do tempo.	Arquitetura em Nuvem e Desacoplada	...
RNF02	O sistema deve buscar suas respostas no conteúdo dos documentos fornecidos em pelo menos 95% dos casos.	Arquitetura RAG e *Prompt* de Sistema	A arquitetura RAG foi projetada para este fim. A fidelidade é reforçada por um "*Prompt* de Sistema"que instrui o modelo de IA a nunca utilizar conhecimento externo.
RNF03	O *chatbot* deve ser acessível por meio da plataforma *WhatsApp*.	Módulo de Integração com *WhatsApp*	Um componente específico da arquitetura é dedicado a gerenciar a conexão com a API do *WhatsApp*, garantindo a acessibilidade do *chatbot* por meio desta plataforma popular.
RNF04	A adição de novos documentos à base de conhecimento deve ser possível apenas adicionando os arquivos na pasta configurada no Google Drive e re-executando o *script* de indexação, sem a necessidade de alterações no código-fonte da aplicação.	*Pipeline* de Indexação Automatizado	O processo de atualização da base de conhecimento é feito pela simples adição de novos arquivos na pasta do Google Drive e a re-execução de um *script*, não exigindo modificação do código da aplicação.

Tabela A.3 – continuação

ID	Descrição do Requisito	Componente da Solução Implementada	Justificativa da Rastreabilidade
RNF05	As chaves de API e credenciais de acesso ao banco de dados não devem estar presentes no código-fonte.	Gerenciamento de Variáveis de Ambiente	Todas as informações sensíveis são carregadas a partir de variáveis de ambiente, que são externas ao código, seguindo as melhores práticas de segurança para o desenvolvimento de software.

Referências

Adamopoulou, E.; Moussiades, L. Chatbots: history, technology, and applications. **Machine Learning with applications**, [S.l.], v.2, p.100006, 2020.

Adamopoulou, E.; Moussiades, L. An Overview of Chatbot Technology. **Artificial Intelligence Applications and Innovations**, [S.l.], v.584, p.373–83, 2020.

Almeida Fernandes, L. de; Vasconcelos Silveira, F. R. de. Desenvolvimento de um Sistema Computacional para Conversação sobre a Política de Assistência Estudantil do IFCE. **Revista Eletrônica de Iniciação Científica em Computação**, [S.l.], v.22, n.1, p.128–136, 2024.

Anaya, L.; Braizat, A.; Al-Ani, R. Implementing AI-based Chatbot: benefits and challenges. **Procedia Computer Science**, [S.l.], v.239, p.1173–1179, 2024.

Barbosa, L. S. **Um chatbot especializado para o contexto da Universidade Federal de Ouro Preto**. [S.l.]: Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, 2024. Monografia (Graduação em Ciência da Computação).

Brown, T. B. et al. Language Models are Few-Shot Learners. **arXiv preprint arXiv:2005.14165**, [S.l.], 2020.

Carvalho, J. R. d. S. **Chatbot para ajuda de novos alunos**. [S.l.: s.n.], 2020.

Glaese, A. et al. **Improving alignment of dialogue agents via targeted human judgements**. 2022.

Goodfellow, I.; Bengio, Y.; Courville, A. **O Neurônio Biológico e Matemático**. Acessado em: 13 fev. 2025.

Instituto Federal de Educação, Ciência e Tecnologia de Goiás. **Plano de Desenvolvimento Institucional - PDI 2019-2023**. Aprovado pelo CONSUP/IFG em 10 de dezembro de 2018.

Instituto Federal de Educação, Ciência e Tecnologia de Goiás. **Projeto Político Pedagógico Institucional**. 2018.

Instituto Federal de Educação, Ciência e Tecnologia de Goiás. **Resolução nº 173, de 13 de setembro de 2023 – Aprova a alteração do Plano Diretor de Tecnologia da Informação e Comunicação (PDTIC) 2021/2023 do IFG**. 2023.

Kalota, F. A Primer on Generative Artificial Intelligence. **Education Sciences**, [S.l.], v.14, n.2, p.172, 2024.

Kireev, D. et al. Metaplastic and energy-efficient biocompatible graphene artificial synaptic transistors for enhanced accuracy neuromorphic computing. **Nature Communications**, [S.l.], v.13, n.1, p.4386, 2022.

LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. **nature**, [S.l.], v.521, n.7553, p.436–444, 2015.

Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. **Advances in Neural Information Processing Systems**, [S.l.], v.33, p.9459–9474, 2020.

Lialin, V.; Deshpande, V.; Rumshisky, A. Scaling down to scale up: a guide to parameter-efficient fine-tuning. **arXiv preprint arXiv:2303.15647**, [S.l.], 2023.

McTear, M.; Callejas, Z. Introduction to Conversational AI. In: McTear, M.; Callejas, Z.; Griol, D. (Ed.). **Conversational AI**: dialogue systems, conversational agents, and chatbots. [S.l.]: Springer, 2020. p.20–22.

Melo, L.; Melo, L. B. d. et al. **RegBot**: chatbot para o regulamento de ensino de graduação da universidade federal de campina grande. [S.l.: s.n.], 2023.

Monard, M. C.; Baranauskas, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, [S.l.], v.1, n.1, p.32, 2003.

Nilsson, N. J. **The quest for artificial intelligence**. [S.l.]: Cambridge University Press, 2009.

Radford, A. et al. Language Models are Unsupervised Multitask Learners. **OpenAI Blog**, [S.l.], 2019.

Ramirez, I. et al. Residual-based Attention Physics-informed Neural Networks for Efficient Spatio-Temporal Lifetime Assessment of Transformers Operated in Renewable Power Plants. **arXiv preprint arXiv:2405.06443**, [S.l.], 2024.

Rusk, N. Deep learning. **Nature Methods**, [S.l.], v.13, n.1, p.35–35, 2016.

Russell, S. J.; Norvig, P. **Artificial intelligence**: a modern approach. [S.l.]: Pearson, 2016.

Sharma, V.; Rai, S.; Dev, A. A comprehensive study of artificial neural networks. **International Journal of Advanced research in computer science and software engineering**, [S.l.], v.2, n.10, 2012.

Sousa, A. C. S. d.; Fecchio, R. L.; Corrêa, A. G. D. Chatbots no apoio à educação superior: revisão de literatura. **Revista Sítio Novo**, [S.l.], v.2, n.2, p.68–84, jul 2021.

Sutskever, I. Sequence to Sequence Learning with Neural Networks. **arXiv preprint arXiv:1409.3215**, [S.l.], 2014.

Tripp, C. E. et al. Measuring the Energy Consumption and Efficiency of Deep Neural Networks: an empirical analysis and design recommendations. **arXiv preprint arXiv:2403.08151**, [S.l.], 2024.

Tsivitanidou, O.; Ioannou, A. Users' Needs Assessment for Chatbots' Use in Higher Education. In: CENTRAL EUROPEAN CONFERENCE ON INFORMATION AND INTELLIGENT SYSTEMS, 31., Varaždin. **Proceedings...** Faculty of Organization and Informatics: University of Zagreb, 2020. p.55–62.

Turing, A. M. Computing Machinery and Intelligence. **Mind**, [S.l.], v.59, n.236, p.433–460, 1950.

Uzair, M.; Jamil, N. Effects of Hidden Layers on the Efficiency of Neural networks. In: IEEE 23RD INTERNATIONAL MULTITOPIC CONFERENCE (INMIC), 2020. **Anais...** [S.l.: s.n.], 2020. p.1–6.

Vinyals, O.; Le, Q. V. A Neural Conversational Model. In: ICML DEEP LEARNING WORKSHOP. **Anais...** [S.l.: s.n.], 2015.

Wallace, R. S. The Anatomy of A.L.I.C.E. In: INTERNATIONAL WORKSHOP ON HUMAN-COMPUTER CONVERSATION. **Anais...** [S.l.: s.n.], 2003.

Wallace, R. S. The Elements of AIML Style. In: INTERNATIONAL WORKSHOP ON HUMAN-COMPUTER CONVERSATION. **Anais...** [S.l.: s.n.], 2003. AIML: Artificial Intelligence Markup Language.

Weizenbaum, J. ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine. **Communications of the ACM**, [S.l.], v.9, n.1, p.36–45, 1966.

Zhao, W. X. et al. A Survey of Large Language Models. **arXiv preprint arXiv:2303.18223v15**, [S.l.], 2024.