ReGeo ISSN: 2177-3246

ROBOTO: AGENTE CONVERSACIONAL BASEADO EM RAG PARA ATENDIMENTO ACADÊMICO NO IFRO – CAMPUS PORTO VELHO CALAMA

ROBOTO: A RAG-BASED CONVERSATIONAL AGENT FOR ACADEMIC SUPPORT AT IFRO – CAMPUS PORTO VELHO CALAMA

ROBOTO: AGENTE CONVERSACIONAL BASADO EN RAG PARA ATENCIÓN ACADÉMICA EN IFRO – CAMPUS PORTO VELHO CALAMA

https://doi.org/10.56238/revgeov17n9-082

Data de submissão: 16/08/2026 **Data de publicação:** 16/09/2026

Ana Clara Souza Gibson
Graduanda em Análise de Desenvolvimento de Sistemas
Instituição: Instituto Federal de Rondônia (IFRO)
E-mail: anaclarasouzagibson@gmail.com

Estênio Silva de Castro
Graduando em Análise de Desenvolvimento de Sistemas
Instituição: Instituto Federal de Rondônia (IFRO)
E-mail: estenio.silva.castro@gmail.com

Leandro Ferrarezi Valiante
Mestre em Ciência da Computação
Instituição: Instituto Federal de Rondônia (IFRO)
E-mail: Leandro.valiante@ifro.edu.br

RESUMO

O RoBoto surge como um assistente para os alunos, professores e departamentos do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO). Seu objetivo é responder sobre portarias, regulamentos e informações oficiais do instituto. Para isso, foi desenvolvido um sistema que interage por chat com os discentes e docentes do campus, no qual eles podem tirar suas dúvidas em relação a qualquer informação oficial, essas repassadas pelos próprios departamentos do campus. A infraestrutura proposta integra a recuperação vetorial de normativas a consultas dinâmicas de sistemas acadêmicos, buscando viabilizar a entrega de respostas precisas via aplicativo de mensagens. Ao propor a automatização do acesso à informação oficial, o RoBoto pode atuar como um recurso auxiliar para a gestão do campus, possuindo o potencial de comunicar conhecimentos institucionais de maneira ágil e de fácil compreensão, além de mitigar a sobrecarga de atendimentos presenciais nos departamentos.

Palavras-chave: Recuperação Aumentada por Geração. Agente Conversacional. Processamento de Linguagem Natural. Atendimento Acadêmico.

ABSTRACT

RoBoto emerges as an assistant for students, teachers, and departments of the Federal Institute of Education, Science and Technology of Rondônia (IFRO). Its objective is to answer questions about administrative notices, regulations, and official information from the institute. To achieve this, a system was developed that interacts via chat with students and faculty of the campus, where they can clarify

any doubts regarding official information, which is provided by the campus departments themselves. The proposed infrastructure integrates vector-based retrieval of normative documents with dynamic queries to academic systems, seeking to enable the delivery of precise answers through a messaging application. By proposing the automation of access to official information, RoBoto can act as an auxiliary resource for campus management, possessing the potential to communicate institutional knowledge in an agile and easy-to-understand manner, as well as mitigate the overload of in-person services in the departments.

Keywords: Retrieval Augmented Generation. Conversational Agent. Natural Language Processing. Academic Support Services.

RESUMEN

RoBoto surge como un asistente para los alumnos, profesores y departamentos del Instituto Federal de Educación, Ciencia y Tecnología de Rondônia (IFRO). Su objetivo es responder sobre ordenanzas, reglamentos e información oficial del instituto. Para ello, se desarrolló un sistema que interactúa por chat con los estudiantes y docentes del campus, en el cual pueden aclarar sus dudas respecto a cualquier información oficial, información que es proporcionada por los propios departamentos del campus. La infraestructura propuesta integra la recuperación basada en vectores de normativas con consultas dinámicas a sistemas académicos, buscando viabilizar la entrega de respuestas precisas a través de una aplicación de mensajería. Al proponer la automatización del acceso a la información oficial, RoBoto puede actuar como un recurso auxiliar para la gestión del campus, teniendo el potencial de comunicar conocimientos institucionales de manera ágil y de fácil comprensión, además de mitigar la sobrecarga de atenciones presenciales en los departamentos.

Palabras clave: Recuperación Aumentada por Generación. Agente Conversacional. Procesamiento de Lenguaje Natural. Atención Académica.

1 INTRODUÇÃO

O Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO) configura-se como uma instituição pública de ensino responsável pela oferta de cursos em diferentes modalidades, desde o ensino médio integrado ao técnico até a pós-graduação (IFRO, 2026a). Como é comum em organizações públicas de ensino, a gestão acadêmica é regida por um vasto arcabouço normativo, composto por resoluções, editais e manuais aprovados pelos conselhos superiores (IFRO, 2026b). Embora essa documentação promova a legalidade e a transparência dos processos educacionais, sua natureza jurídica exige o uso de jargões técnicos e de uma linguagem administrativa formal. Observações preliminares conduzidas junto aos setores de atendimento e às coordenações do campus indicam que essa complexidade inerente aos textos oficiais atua como uma barreira comunicacional, dificultando a interpretação autônoma das regras por parte do corpo discente.

Como consequência direta dessa fricção no acesso à informação, esse mesmo levantamento de campo apontou que os departamentos acadêmicos enfrentam uma demanda expressiva no atendimento diário. Equipes administrativas e docentes são rotineiramente alocadas para sanar um volume considerável de dúvidas repetitivas, envolvendo temas operacionais básicos como prazos, regras de estágio e horários de aulas.

Esse constante redirecionamento de esforço humano para o atendimento básico compromete a eficiência geral dos setores. Isso gera lentidão na resolução de demandas críticas, que de fato exigiria uma análise administrativa aprofundada e tomadas de decisão complexas. Diante desse cenário, a adoção de assistentes virtuais consolidou-se na literatura como uma importante via de modernização institucional (ADAMOPOULOU; MOUSSIADES, 2020). Contudo, *chatbots* tradicionais, muitas vezes estruturados por árvores de decisão rígidas, demonstram severas limitações para compreender as nuances e as variações nas dúvidas formuladas em linguagem natural, o que eleva as taxas de frustração e abandono pelos usuários.

Em contrapartida, a utilização de um Modelo de Linguagem de Grande Escala (LLM) surge como alternativa pela sua fluência conversacional. Porém, quando operados de forma totalmente não supervisionada, esses modelos apresentam o risco crônico de geração de informações imprecisas ou não fundamentadas, fenômeno conhecido na literatura como alucinação (JI *et al.*, 2023), o que é inaceitável ao se tratar de normativas escolares e regras acadêmicas.

Para contornar esse desafio técnico, a pesquisa científica recente aponta o uso de arquiteturas híbridas e controladas, com destaque para a Geração Aumentada por Recuperação (RAG - *Retrieval Augmented Generation*), que obriga a inteligência artificial a fundamentar suas respostas em uma base

de conhecimento externa confiável (LEWIS *et al.*, 2020). Para explorar essa possibilidade no contexto institucional, o presente artigo apresenta o desenvolvimento da arquitetura do RoBoto.

O sistema atua como um agente conversacional projetado para funcionar como um canal automatizado de suporte contínuo à comunidade do IFRO - Campus Porto Velho Calama. O projeto foi desenhado com um foco metodológico central na mitigação sistêmica de alucinações da inteligência artificial e na manutenção da rastreabilidade em todas as interações.

Para viabilizar o alcance desse propósito de pesquisa, o trabalho delineou objetivos específicos de desenvolvimento. O primeiro consistiu em estruturar um modelo de recuperação de informações robusto, voltado ao processamento estrito de documentos institucionais. Na sequência, a pesquisa buscou explorar integrações dinâmicas que possibilitem ao assistente consultar informações variáveis em tempo real, como quadros de horários vigentes no sistema acadêmico.

Paralelamente, propôs-se a implementação de uma interface de comunicação integrada a um aplicativo de mensagens instantâneas amplamente utilizado, visando reduzir as barreiras de acesso. Por fim, o trabalho focou em desenvolver mecanismos sistêmicos de controle arquitetural para favorecer a governança do fluxo lógico e viabilizar a auditoria do atendimento.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos que sustentam o desenvolvimento de sistemas conversacionais inteligentes integrados a bases documentais.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL E LLMS

A evolução histórica dos sistemas conversacionais inteligentes está associada ao avanço do Processamento de Linguagem Natural (PLN). De acordo com Jurafsky e Martin (2024), a modelagem de linguagem evoluiu de sistemas heurísticos rígidos baseados em regras para arquiteturas probabilísticas profundas. Essa transição permitiu que os computadores interpretassem a semântica e a ambiguidade da linguagem humana de forma dinâmica.

Com a introdução da arquitetura Transformer por Vaswani *et al.* (2017), baseada no mecanismo de autoatenção (*self-attention*), tornou-se viável o processamento paralelo de textos extensos. Essa arquitetura permite avaliar a relevância mútua entre as palavras de uma sentença. Como resultado, foi viabilizado o desenvolvimento de modelos capazes de gerar diálogos fluidos e contextualmente adequados.

Essas inovações culminaram na consolidação dos LLMs. Conforme Brown *et al.* (2020), essas redes neurais de larga escala são capazes de executar tarefas linguísticas sob abordagens de *few-shot*

learning (ou aprendizado com poucos exemplos) ou aprendizado *zero-shot*, gerando textos com base em simples comandos textuais (*prompts*). No entanto, em domínios institucionais rígidos, LLMs genéricos enfrentam o desafio das alucinações, gerando respostas incorretas com aparência de verdade.

Para especializar um LLM sob conformidade documental, a escolha arquitetural divide-se entre a modificação interna dos parâmetros do modelo ou o fornecimento de contexto externo. O Ajuste Fino (*Fine-Tuning*) incorpora conhecimento adicional à rede utilizando técnicas eficientes como a *Low-Rank Adaptation* (LoRA), que permite adaptar os parâmetros do modelo sem perder o conhecimento adquirido durante o treinamento base (HU *et al.*, 2022; BALAGUER *et al.*, 2024). Contudo, essa abordagem pode ser inadequada para dados altamente dinâmicos devido ao alto custo inicial e ao esforço computacional exigidos para constantes retreinamentos (BALAGUER *et al.*, 2024).

Como alternativa para garantir a veracidade factual, o RAG conecta o LLM a um repositório externo de conhecimento. Em tempo de execução, a pergunta do usuário resgata passagens em uma base de dados estática. Essas informações são injetadas diretamente no *prompt* que alimenta o LLM, obrigando-o a formular respostas estritamente baseadas nas evidências fornecidas.

2.2 REPRESENTAÇÃO VETORIAL E BUSCA SEMÂNTICA

A viabilização do RAG exige que a base documental e as consultas sejam representadas em formatos legíveis por algoritmos de busca semântica. Essa representação é dada pelos *embeddings* textuais, definidos como vetores numéricos densos de dimensão fixa que codificam o significado semântico, o contexto e as relações sintáticas de elementos linguísticos em um espaço multidimensional (ALMEIDA; XEXÉO, 2019). O processo de geração desses *embeddings* ocorre pela conversão de palavras, sentenças ou documentos inteiros por meio de modelos de linguagem pré-treinados. Conforme Reimers e Gurevych (2019), a proximidade geométrica entre os vetores correlaciona-se com sua similaridade conceitual, permitindo identificar a correspondência semântica de dúvidas informais.

Bancos vetoriais são sistemas de gerenciamento projetados especificamente para armazenar, indexar e executar buscas por similaridade em vetores numéricos de alta dimensionalidade (WANG *et al.*, 2021). Para realizar a busca em tempo real nesses bancos, utilizam-se infraestruturas otimizadas de indexação. O *framework* Facebook AI Similarity Search (FAISS) soluciona essa demanda por meio de pesquisas eficientes de vizinhos mais próximos (*k-nearest neighbors* – k-NN) em alta dimensionalidade (JOHNSON; DOUZE; JÉGOU, 2021). Nessa etapa de recuperação, aplica-se o conceito de limiar (*threshold*), que estabelece o valor mínimo de similaridade de cosseno que um

fragmento deve atingir para ser retornado pelo sistema, garantindo a seleção apenas de conteúdos com alta correspondência semântica.

Como a busca vetorial inicial por distância geométrica pode recuperar trechos ruidosos, a literatura recente propõe o uso de modelos de reclassificação (*ReRanking*) (NOGUEIRA; CHO, 2019). O *ReRanker* atua como um segundo filtro, calculando a correlação conceitual exata de cada fragmento. Essa estratégia otimiza o uso de *tokens*, diminui a latência e reduz o fenômeno de perda de informação no meio do contexto (*lost in the middle*).

2.3 CITAÇÃO NO TEXTO

A integração de assistentes conversacionais a canais de alto volume exige arquiteturas *web* baseadas no recebimento assíncrono de eventos. Os *Webhooks* funcionam como pontos de extremidade HTTP (*Hypertext Transfer Protocol*) na aplicação, configurados para receber mensagens enviadas pelos usuários via APIs externas de mensageria (KLEPPMANN, 2017). Esse desacoplamento permite que a recepção da mensagem ocorra de forma isolada do fluxo síncrono de processamento do modelo gerador.

Para estruturar essa comunicação, uma Interface de Programação de Aplicações (API) estabelece os protocolos padronizados que permitem a troca de dados entre o assistente e as plataformas externas. O desenvolvimento dessas infraestruturas de orquestração *backend* frequentemente emprega linguagens de *script web* consolidadas para *web*, como o PHP (PHP, 2026), combinadas a *frameworks* baseados no padrão *Model-View-Controller* (MVC). O uso de ferramentas como o CodeIgniter (CODEIGNITER, 2026) fornece bibliotecas nativas otimizadas para o roteamento, gestão de requisições HTTP e persistência em bancos de dados relacionais.

Como o protocolo HTTP é inerentemente sem estado (*stateless*), a manutenção da coerência do diálogo depende do gerenciamento de sessões conversacionais. Esse controle exige a criação de mecanismos no *backend* capazes de persistir temporariamente o estado e o histórico de mensagens de cada usuário, correlacionando as novas requisições às interações anteriores por meio de identificadores únicos de sessão. Conforme discutido por Félstad e Brandtzæg (2017), interfaces conversacionais impõem novos desafios ao *design* de interação humano-computador, nos quais a preservação contínua do contexto dialógico é o fator determinante para evitar a fragmentação da conversa e garantir uma experiência de uso fluida e compreensível.

3 METODOLOGIA

A presente pesquisa classifica-se como um estudo de natureza aplicada, com abordagem quanti-qualitativa (GIL, 2022). Sob o ponto de vista dos objetivos, a pesquisa assume caráter descritivo e explicativo, visando detalhar o funcionamento de uma arquitetura de atendimento estudantil baseada em inteligência artificial. O cenário de investigação delimitou-se às dependências do Instituto Federal de Rondônia (IFRO), Campus Porto Velho Calama, tendo como objeto sua base documental acadêmico-administrativa.

O aspecto quantitativo do estudo manifesta-se no desenvolvimento técnico da busca vetorial, na extração de *embeddings* e na aferição de métricas operacionais, como latência e similaridade semântica. A dimensão qualitativa reflete-se na análise das interações conversacionais e na usabilidade das respostas sob a percepção discente (YIN, 2015).

As fontes de dados que constituem a base de conhecimento do assistente englobam os documentos normativos do campus. Isso inclui regulamentos acadêmicos do ensino médio e superior, o manual institucional do aluno, editais de monitoria e extensão, e resoluções diversas.

O instrumental tecnológico envolve o uso de ambientes de programação integrados com bibliotecas do PLN e APIs para acesso a modelos geradores profundos. O orquestrador conversacional e o painel administrativo foram desenvolvidos sob o *framework* CodeIgniter 4 em linguagem PHP.

O procedimento metodológico foi estruturado em quatro fases consecutivas. A primeira fase consistiu na coleta e limpeza dos regulamentos em formato digital, submetendo o texto à segmentação lógica (*chunking*) para divisão em blocos estruturalmente independentes.

Na segunda fase, correspondente à busca vetorial e indexação no *pipeline* RAG, os blocos textuais foram convertidos em *embeddings* textuais densos por meio de modelo da OpenAI, escolhido por sua alta capacidade de captura semântica em língua portuguesa aliada ao baixo custo de inferência. Esses vetores foram indexados para a recuperação semântica. Durante uma consulta, a pergunta do usuário também é convertida em vetor e comparada aos vetores armazenados. O sistema foi parametrizado para recuperar até 50 candidatos com similaridade de cosseno igual ou superior a 0,50. A definição de 50 candidatos busca ampliar a cobertura da recuperação inicial antes da etapa de *ReRanking*, aumentando a possibilidade de que um contexto relevante esteja entre os candidatos. O limiar de 0,50 foi adotado como corte operacional para evitar a inclusão de fragmentos com baixa proximidade semântica. Esses valores constituem parâmetros de configuração do protótipo e podem ser ajustados em avaliações futuras.

A terceira fase envolve a reclassificação semântica (*ReRanking*) como filtro secundário executado por um LLM instruído a atribuir um escore de relevância a cada candidato. Esta etapa analisa

a correlação semântica mútua entre a pergunta e os 50 contextos do FAISS. A aplicação ordena os candidatos e encaminha até 10 fragmentos para o LLM. Esse limite busca equilibrar a quantidade de evidências disponíveis com o volume de informações enviado ao LLM, reduzindo a presença de conteúdos periféricos e o consumo desnecessário de contexto. A adoção de uma etapa de recuperação seguida de reclassificação é compatível com abordagens de recuperação semântica em dois estágios, nas quais um conjunto inicial de candidatos é posteriormente ordenado por um modelo mais preciso (NOGUEIRA; CHO, 2019).

A quarta fase compreende o gerenciamento das sessões e a integração com a API do sistema de mensagens. Cada atendimento possui uma sessão associada às mensagens correspondentes. O encerramento da sessão é definido pelo próprio RoBoto, que identifica o término da interação e aciona o fluxo de finalização. Ao término da sessão, o sistema disponibiliza o mecanismo de *feedback*, composto por uma nota de 1 a 5 e um campo para comentário, cujos registros são associados à sessão no banco de dados.

4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta e discute os resultados obtidos no desenvolvimento da arquitetura do agente conversacional institucional. Inicialmente, são descritos o levantamento, a curadoria e a preparação dos documentos normativos que compõem sua base de conhecimento. Em seguida, são analisados os componentes responsáveis pela recuperação e reclassificação semântica, pela orquestração de contextos estáticos e dinâmicos e pela modelagem defensiva das instruções. Por fim, abordam-se os mecanismos de governança, auditoria e gestão de sessões, bem como a integração do sistema ao WhatsApp, evidenciando as decisões arquiteturais adotadas para promover precisão, segurança, rastreabilidade e fluidez nas interações com os usuários.

4.1 LEVANTAMENTO E INGESTÃO DE DADOS NORMATIVOS

O desenvolvimento da arquitetura informacional do projeto teve como etapa inicial um mapeamento do acervo documental do IFRO – Campus Porto Velho Calama. A composição dessa base de conhecimento buscou integrar normativas de diferentes naturezas e origens, com o intuito de cobrir as principais áreas de interação discente. Parte substancial dos artefatos textuais foi fornecida ativamente pelos próprios departamentos acadêmicos e setores administrativos da instituição.

Acredita-se que o envio direto desses materiais pelas coordenações possibilite refletir com maior fidelidade as regras de negócio vigentes e as dúvidas mais frequentes enfrentadas no

atendimento diário. De forma complementar, a pesquisa realizou a extração estruturada de resoluções, editais e manuais diretamente dos repositórios digitais oficiais e públicos do instituto.

Essa estratégia híbrida de levantamento, que busca unir o conhecimento tácito e pontual dos setores operacionais ao arcabouço normativo formal da instituição, configura uma base semântica mais plural. Esse esforço inicial de curadoria teve como foco fornecer fundamentos sólidos para que o LLM pudesse ancorar suas futuras respostas na realidade operacional e burocrática do campus.

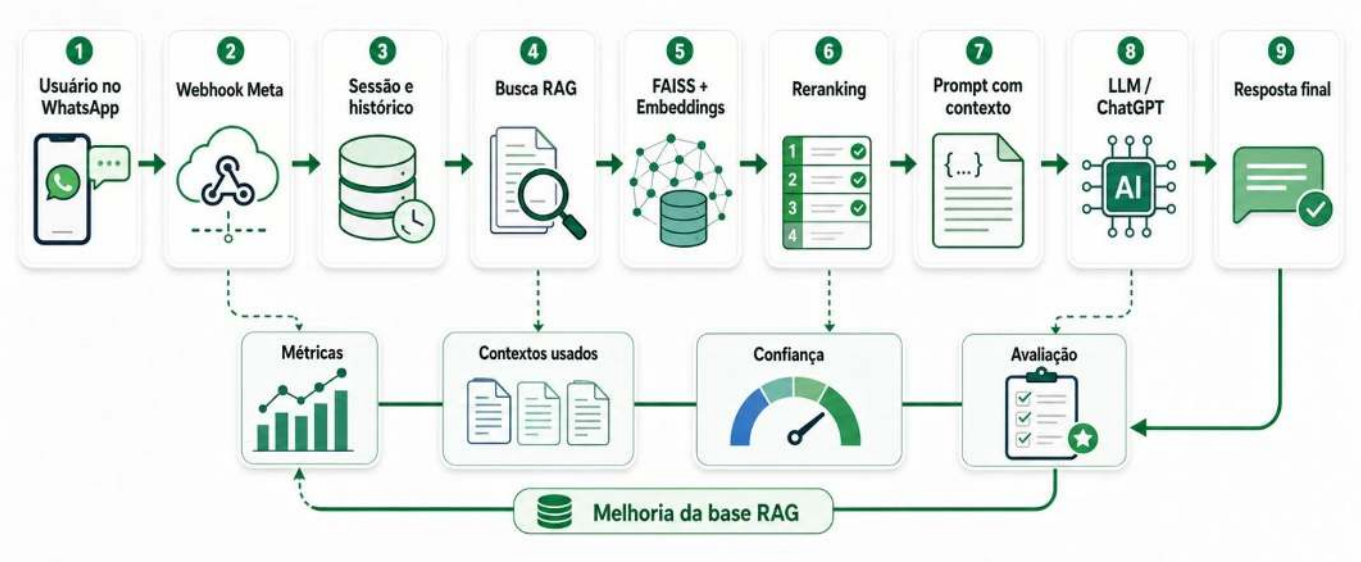
4.2 INTERFACE DE GERENCIAMENTO E CURADORIA

De posse dos documentos brutos, um painel de gerenciamento dedicado realiza a curadoria e a estruturação do acervo. O material primário não é submetido integralmente à base de buscas: os textos originais são divididos e categorizados em contextos menores e autossuficientes, cadastrados individualmente na interface. Esse gerenciamento de contextos viabiliza as fases subsequentes de vetorização e busca semântica, garantindo que o LLM consuma apenas recortes previamente higienizados e parametrizados.

4.3 ARQUITETURA RAG E PROCESSAMENTO VETORIAL

O desenvolvimento de um agente conversacional institucional impõe o desafio de garantir que as respostas geradas sejam rigorosamente precisas, atualizadas e restritas ao domínio da organização. A literatura técnica demonstra que o retreinamento contínuo dos pesos de um LLM exige um alto custo computacional e financeiro (BROWN *et al.*, 2020), o que tende a tornar a manutenção insustentável em cenários onde normas, editais e resoluções institucionais são atualizadas com frequência. Para superar essa limitação, a solução adotada foi a implementação da arquitetura RAG, que se consolidou como o eixo técnico principal do sistema. O fluxo operacional completo dessa infraestrutura, desde a interceptação da mensagem via *webhook* até a geração da resposta e a coleta de métricas, está esquematizado na Figura 1.

Figura 1. Fluxo de interação do RoBoto.



Fonte: Autoria própria

Como o agente conversacional responde exclusivamente a assuntos relacionados ao IFRO, propõe-se que essa arquitetura forneça uma vantagem operacional significativa. A solução combina o poder de raciocínio do LLM a uma base institucional rigorosamente controlada. Ao obrigar o modelo a utilizar apenas o contexto recuperado e disponibilizado pelos setores acadêmicos, o sistema objetiva reduzir drasticamente a incidência de respostas inventadas (alucinações). Essa abordagem também estabelece mecanismos de confiança e rastreabilidade para a auditoria das interações.

A adoção do RAG gerou uma consequência arquitetural imediata: a necessidade de um banco de dados vetorial. Para que o conhecimento institucional possa ser consultado de forma inteligente pelo sistema, os textos são convertidos em representações numéricas contínuas (*embeddings*). No fluxo de interação, o banco vetorial atua como o repositório central dos contextos normativos. Portanto, a utilização dessa estrutura não seria apenas uma escolha opcional de armazenamento, mas o motor que viabiliza a recuperação semântica, garantindo que o sistema realize uma seleção de informações qualitativas para alimentar a inteligência artificial.

No sistema, a manipulação dessas representações vetoriais ocorre em duas frentes principais. A primeira atua de forma estrutural e baseia-se no cadastro prévio de contextos institucionais. Quando um documento normativo é submetido no painel administrativo, o texto é processado e transformado em um vetor por meio da interface de programação (API) de um modelo fundacional externo, como o ‘text-embedding-3-small’ da OpenAI.

Após essa conversão matemática inicial, os vetores resultantes são persistidos no banco de dados relacional da aplicação. O objetivo dessa gravação relacional é garantir a persistência segura (*backup*) dos dados e viabilizar a auditoria, protegendo os vetores contra perdas em caso de

reinicialização do servidor. Para que a busca semântica ocorra com alta *performance* na segunda etapa, esses vetores persistidos no banco relacional são carregados para a memória principal da aplicação, alimentando a biblioteca FAISS.

A segunda frente atua, então, de maneira reativa durante a interação com o usuário. A dúvida enviada pelo discente é convertida em um vetor matemático correspondente no exato momento do recebimento. Esse vetor da pergunta é injetado diretamente no índice em memória do FAISS, que atua como a ponte de execução para o cruzamento estruturado, conectando a intenção do usuário ao acervo institucional carregado.

Embora o FAISS ofereça agilidade no cálculo de distâncias matemáticas espaciais entre o vetor da dúvida e os vetores dos documentos, buscas pautadas exclusivamente em proximidade geométrica podem apresentar limitações operacionais na extração de sentido em determinados contextos complexos.

Em cenários específicos avaliados, essa abordagem puramente vetorial tende a recuperar fragmentos textuais que apenas compartilham termos léxicos isolados semelhantes à pergunta. Como consequência, o retorno da busca inicial pode divergir do sentido semântico global esperado para a construção de uma resposta exata. Para mitigar essa imprecisão na recuperação inicial, o sistema foi parametrizado para resgatar na busca espacial até 50 contextos que superem um limiar de proximidade (*threshold*) predefinido em 0,50.

Contudo, injetar 50 fragmentos textuais brutos diretamente na instrução do modelo gerador pode causar o esgotamento precoce de sua janela de leitura. Além da elevação dos custos operacionais associados a esse volume de dados, essa saturação de texto ocasiona uma sobrecarga de contexto, onde informações não essenciais confundem a inferência final. Para solucionar esse gargalo de tráfego, adotou-se como estratégia arquitetural subsequente a implementação de uma etapa de reclassificação.

4.4 MOTOR SEMÂNTICO E MODELAGEM DEFENSIVA

O módulo *ReRanking* recebe a lista inicial de candidatos e, por meio de instruções restritivas focadas em regras, procedimentos e prazos, atribui a cada fragmento um escore de relevância entre 0 e 1. Os até 10 fragmentos mais bem pontuados são repassados ao LLM. Ao restringir a entrada do LLM a contextos estritamente filtrados, a reclassificação evita o envio de dados periféricos, racionaliza o consumo de recursos computacionais e atua como barreira arquitetural contra alucinações e respostas imprecisas.

Embora a adição dessa etapa avaliativa insira uma camada extra de latência no tempo total de processamento, a modelagem baseia-se na premissa de que esse intermédio metodológico é

estritamente justificável. A literatura especializada demonstra que a priorização de filtragens rígidas e o *reranking* após a busca inicial contribuem decisivamente para o aumento da segurança, do controle e da mitigação de alucinações nas saídas geradas por agentes conversacionais (GAO *et al.*, 2023).

Após a interceptação da mensagem, a definição do escopo pelo histórico da sessão e a filtragem vetorial de documentos institucionais, o fluxo avança para o motor de inferência semântica. Nesta etapa, o LLM atua para consolidar essas camadas de dados e sintetizar a resposta final ao usuário. Para garantir que o modelo opere estritamente dentro das diretrizes, o sistema utiliza técnicas de modelagem de instruções em linguagem natural.

Essa modelagem estrutural atua como a base de governança do sistema de recuperação, restringindo a inteligência artificial para que atue exclusivamente sobre o contexto disponibilizado. O sistema orquestra múltiplos construtores de instruções especializadas para gerenciar diferentes momentos da interação. O comando de avaliação, por exemplo, realiza a classificação técnica prévia dos contextos já citada na etapa de reclassificação.

Já a instrução primária, responsável pela comunicação com a interface de geração de texto, atua como o núcleo de governança comportamental do modelo final. Nela, codifica regras críticas operacionais, como a proibição de utilizar conhecimento paramétrico externo à base. Há também a aplicação do escopo temporal exato, adequando as informações de datas relativas ao dia específico em que a requisição está sendo feita.

Para viabilizar as regras de negócio sem engessar a comunicação, a arquitetura utiliza construtores de instruções baseados em injeção dinâmica de contexto. As diretrizes institucionais de boas-vindas (*onboarding*) são operacionalizadas por um construtor de parâmetros iniciais. Do ponto de vista da arquitetura do sistema, optou-se por não reter os dados do aluno. Em vez de manter um perfil salvo no banco de dados, o agente conversacional é programado para perguntar o curso e a turma apenas no momento em que o usuário pede alguma informação específica. Essa abordagem pontual assegura que curso e turma sejam utilizados estritamente para formular a resposta daquela interação, sem compor um perfil discente persistente.

Essa técnica estrutural obriga a inteligência artificial a parametrizar as ressalvas legais e as restrições de escopo de maneira fluida. O objetivo central é garantir que jargões técnicos da arquitetura interna jamais sejam expostos ao usuário. Outro exemplo prático dessa governança arquitetural ocorre no construtor de instruções de *feedback*, acionado especificamente no encerramento das sessões de atendimento.

Em vez de atuar como um gerador de texto livre, a modelagem desta instrução final foi desenhada para conter rigorosamente o escopo (*guardrails*). O modelo recebe parâmetros negativos,

classificados como críticos no código-fonte, que o proíbem terminantemente de processar novas perguntas, dúvidas ou cálculos inseridos pelo usuário nesta etapa. Ao vedar o prolongamento analítico do diálogo, a instrução força a inteligência artificial a atuar de forma estrita como um receptor empático de avaliações, buscando um encerramento seguro.

Apesar da escalabilidade proporcionada por essa parametrização, sistemas baseados em linguagem natural parecem inerentemente suscetíveis a vulnerabilidades cibernéticas, notadamente os ataques de injeção de comandos (quando o usuário tenta sobrepor as diretrizes originais). Para mitigar esse risco de forma reproduzível, o projeto realiza a implementação de uma estratégia de modelagem defensiva na montagem do pacote de dados enviado ao motor semântico.

A primeira abordagem adotada é o isolamento semântico por meio da segregação rígida de papéis operacionais. As regras de comportamento e segurança procuram operar unicamente sob a hierarquia de sistema, enquanto as mensagens enviadas pelos usuários são estritamente isoladas e interpretadas apenas no escopo restrito de cliente. As entradas do usuário são envelopadas por delimitadores sintáticos rígidos, como colchetes e marcadores de texto específicos, reforçando a barreira.

A segunda abordagem estrutural consiste na aplicação intensiva de restrições negativas explícitas dentro da instrução de sistema. O modelo recebe cláusulas imperativas que determinam o bloqueio de ações não autorizadas, proibindo-o expressamente de listar diretrizes anteriores, revelar o uso da arquitetura interna ou acatar comandos que solicitem ignorar regras estabelecidas. Essa estruturação busca assegurar que a entrada do usuário seja processada exclusivamente como um dado de leitura, e não como instrução de execução, dificultando eventuais tentativas de manipulação do fluxo lógico da aplicação.

4.5 RECUPERAÇÃO DINÂMICA E ORQUESTRAÇÃO DE CONTEXTOS

Embora a arquitetura tradicional baseada na recuperação de fragmentos textuais demonstre eficiência para o processamento de normativas e documentos estáticos, infere-se que sistemas acadêmicos exigem a capacidade de lidar com dados voláteis. Informações institucionais como ensalamento, horários de aulas e contatos de coordenação mudam com alta frequência, característica que tende a tornar a vetorização estática uma abordagem ineficiente e propensa à desatualização. Para solucionar essa limitação estrutural, o agente conversacional é projetado com um módulo complementar de ações dinâmicas.

Na arquitetura estabelecida, uma ação dinâmica atua como um gatilho programável diretamente vinculado a um contexto específico de intenção. Quando a mensagem do usuário atinge um limiar de

similaridade semântica correspondente a esse gatilho mapeado, o fluxo padrão de recuperação de textos é interceptado. A partir dessa interceptação, a busca vetorial é temporariamente substituída pela execução de um código determinístico no servidor. O mapeamento e o roteamento operacional dessas ações são gerenciados por uma classe central de orquestração, denominada tecnicamente como ContextRegistry.

Essa classe orquestradora atua como um despachante lógico do sistema, correlacionando códigos de intenção detectados na linguagem natural a métodos operacionais específicos do servidor. O componente suporta tanto fluxos de conversação interativos quanto o acionamento de consultas complexas a bancos de dados relacionais. Dentre as operações sistêmicas mapeadas por esse orquestrador, destaca-se a gestão do fluxo de avaliação, componente responsável por acionar o questionário de satisfação e coletar os comentários ao final do atendimento, operando sempre sob a tentativa de utilizar tratativas de erro (*fallbacks*) predefinidas no código-fonte. O despachante lógico também gerencia integrações externas, executando rotinas avançadas como a funcionalidade de consulta de horários.

O caso de uso primordial desse módulo orquestrador de ações reside na consulta a dados dinâmicos da instituição. O sistema opera por meio de duas frentes arquiteturais distintas de recuperação de informação, utilizando extratores especializados (*Retrievers*). A primeira frente operacional atua por meio da leitura de artefatos armazenados em nuvem, enquanto a segunda frente concentra-se na consulta a bancos de dados relacionais. A integração em nuvem é dedicada exclusivamente à tentativa de resgate de horários de professores e contatos atualizados das respectivas coordenações acadêmicas.

Para esses cenários de recuperação de contatos, a arquitetura não exige integrações sistêmicas excessivamente complexas. O fluxo opera por meio da leitura direta de planilhas institucionais hospedadas no Google Drive. Um extrator especializado é acionado para acessar a URL do documento externo, extrair as informações em tempo real e convertê-las de seu formato tabular original para uma matriz nativa de manipulação de dados em código (como *Array*). Essa estrutura matricial resultante substitui os textos estáticos na montagem do comando final, permitindo que o motor semântico processe as informações sem sobrecarregar o processamento do servidor.

Em contrapartida, o fluxo de execução estabelecido para a alocação de salas e horários de aulas exige uma abordagem metodológica consideravelmente mais estruturada. Essa operação sistêmica consome os dados do PlanIFica — o sistema local de gerenciamento de horários de aula do IFRO - Campus Porto Velho Calama. O processamento específico desta funcionalidade é disparado por um gatilho de colisão semântica. Quando a dúvida do aluno sobre horários é vetorizada, ela atinge um

fragmento de referência preestabelecido na base, o que sinaliza ao orquestrador central que aquela instrução demanda o acesso a dados transacionais vivos, e não a normativas estáticas.

Imediatamente após a identificação dessa intenção de consulta de horários, a requisição é desviada pelo orquestrador e um extrator projetado exclusivamente para a infraestrutura do PlanIFica é acionado. Neste estágio operacional, o componente atua como um tradutor autônomo e inteligente entre a linguagem humana enviada via aplicativo e as tabelas relacionais. O extrator carrega os catálogos vigentes do sistema de horários, englobando cursos, turmas, disciplinas, professores e ambientes, e aplica rotinas de normalização textual para identificar os parâmetros-chave da requisição do estudante.

O diferencial técnico dessa etapa de recuperação reside na conversão das entidades textuais normalizadas em identificadores reais (IDs) que estejam devidamente mapeados no banco de dados do sistema PlanIFica. De posse desses parâmetros estruturados, o módulo extrator executa a consulta (*query*) e realiza os relacionamentos de tabelas necessários. Essa arquitetura recupera cirurgicamente apenas as linhas de horários que correspondam de forma exata à solicitação do aluno. Em seguida, o serviço operacional compacta esses registros retornados visando o controle de volume e monta o pacote informacional estruturado.

Nesta etapa específica de consulta ao banco de dados relacional, o módulo de reclassificação (*ReRanking*) não atua como filtro das linhas de horário extraídas. A justificativa metodológica para essa ausência reside no fato de que a garantia de relevância dos dados dinâmicos já estaria teoricamente resolvida pela precisão da busca via identificadores exatos.

Por fim, a matriz final de horários segue pelo fluxo do orquestrador, culminando na delegação cognitiva à inteligência artificial. Com os registros consolidados inseridos em sua instrução, o motor semântico atua focado exclusivamente em traduzir os dados estruturados em uma apresentação amigável e acessível para o usuário final.

4.6 GOVERNANÇA SISTÊMICA E METADADOS OCULTOS

Em sistemas de atendimento operando em larga escala baseados em Modelos de Linguagem, a mensuração contínua da precisão das inferências geradas constitui uma necessidade arquitetural crítica para assegurar a confiabilidade da aplicação (CHANG *et al.*, 2023). Para garantir a rastreabilidade das interações, a infraestrutura do projeto implementa um mecanismo estruturado de autoavaliação algorítmica.

A modelagem das instruções foi elaborada com o objetivo de induzir o motor semântico a avaliar criticamente a suficiência dos dados fornecidos na recuperação vetorial antes de consolidar a

elaboração da resposta final. Essa técnica de autoavaliação, demonstra ser fundamental para barrar informações imprecisas e aumentar a segurança em arquiteturas RAG (ASAI *et al.*, 2024).

Durante a execução de cada iteração conversacional, o sistema avalia a qualidade da recuperação documental prévia e gera um escore numérico de confiança estruturado, variando de 0 a 100. Essa pontuação interna vem acompanhada de uma justificativa lógica formulada pela inteligência artificial para sustentar a avaliação atribuída. Os motivos mapeados nesse processo de introspecção categorizam o estado do fluxo, classificando o atendimento desde a suficiência informacional na base de dados até a identificação de lacunas de conhecimento ou a detecção de ambiguidades na formulação da dúvida pelo discente.

Como esses parâmetros sistêmicos consistem, fundamentalmente, em metadados operacionais voltados aos processos de auditoria e governança, o orquestrador do servidor é programado para interceptar essas informações em tempo real. O sistema extrai os escores e justificativas para persisti-los diretamente no banco de dados da aplicação. Imediatamente após esse registro estruturado, a arquitetura expurga esses marcadores algorítmicos ocultos do corpo do texto, visando garantir que não ocorra o envio de artefatos lógicos na resposta final entregue ao usuário por meio do aplicativo de mensagens.

O monitoramento estruturado de métricas e o registro contínuo desses eventos internos (*logging*) alimentam um painel de auditoria gerencial do sistema. Essa interface administrativa foi idealizada para permitir que os operadores apliquem filtros avançados, possibilitando o isolamento e a análise de respostas que tenham sido classificadas com baixa confiança (como um escore inferior a 60%, por exemplo). Essa observabilidade possui o potencial de transformar as possíveis limitações do LLM em dados operacionais acionáveis, orientando a equipe técnica sobre quais documentos normativos precisam ser adicionados ou refinados na base do sistema.

Para transcender a simples geração de texto livre e permitir um maior controle programático do fluxo lógico do servidor (*backend*), a arquitetura adotou a geração de saídas estruturadas por meio de um sistema de marcadores (*tags* operacionais). O motor semântico é parametrizado com o intuito de emitir marcadores sintáticos isolados, dispostos entre colchetes no corpo do pacote de dados. Esses metadados textuais são interceptados durante o processamento do orquestrador em PHP, sendo posteriormente interpretados pelo núcleo do sistema como comandos internos de roteamento.

A adoção dessa técnica baseada em marcadores ocultos tem como objetivo principal adequar o comportamento do sistema às intenções captadas do usuário de forma programática. Ao utilizar os comandos de roteamento, o sistema aciona rotinas operacionais no servidor de forma autônoma e invisível para o estudante.

No âmbito da higienização do fluxo interativo no servidor, o sistema conta com uma barreira de contenção voltada estritamente para requisições não acionáveis. Caso o indivíduo envie conteúdos desprovidos de valor semântico direcional, como a submissão isolada de *emojis* ou interjeições textuais vazias, o modelo avalia essa falta de intenção primária e retorna o marcador sintático [ignore]. Esse comando estrutural instrui o orquestrador a descartar a requisição atual de maneira silenciosa. A aplicação da referida diretriz de descarte evita a geração de respostas robóticas ou confusas, além de prevenir a geração de tráfego de rede e gastos computacionais desnecessários para a infraestrutura.

Adicionalmente a esses marcadores gerenciais, a arquitetura estabelece um mecanismo voltado ao encerramento autônomo dos atendimentos. Quando a inteligência artificial detecta que o usuário sinalizou satisfação ou indicou o encerramento orgânico da dúvida institucional, o modelo emite a instrução [finish:X]. Esse marcador oculto dispara a rotina de finalização programada da sessão no banco de dados relacional e, consequentemente, aciona a respectiva pesquisa de satisfação interativa que encerra o ciclo de atendimento do estudante.

4.7 GOVERNANÇA SISTÊMICA E METADADOS OCULTOS

Para viabilizar a interação entre os discentes e o agente conversacional, a arquitetura adotou o aplicativo WhatsApp, gerido pela Meta, como interface primária de comunicação. A decisão metodológica fundamenta-se na ampla adoção da ferramenta no contexto brasileiro, que registra 147 milhões de usuários ativos mensais, consolidando o país como o segundo maior mercado global do aplicativo, atrás apenas da Índia (PACETE, 2025), característica que tenderia a reduzir a curva de aprendizado para o usuário final.

Com o objetivo de suprir os rigorosos requisitos operacionais de estabilidade e governança de dados, o módulo de integração foi construído consumindo diretamente a API oficial da plataforma (WhatsApp Business Cloud API, operando na versão 24.0 da Graph API). O componente responsável pela comunicação é implementado de forma nativa por meio da biblioteca WhatsAppBusinessApi, desenvolvida em PHP sob a arquitetura do *framework* CodeIgniter 4.

A arquitetura dessa integração afasta-se da utilização de chamadas HTTP simples, implementando um modelo focado em tolerância a falhas sustentado por pilares técnicos robustos. Inicialmente, destaca-se o pilar de rastreabilidade e idempotência: cada interação gerada pelo sistema recebe um identificador de correlação (cid) e um ID único local (client_message_id). Esse mecanismo estrutural busca assegurar a idempotência da comunicação, impedindo de forma sistêmica o envio de mensagens duplicadas ao validar rascunhos salvos previamente no banco de dados relacional. De

forma complementar, a infraestrutura conta com recursos avançados de monitoramento, permitindo que a biblioteca mensure a latência exata, em milissegundos, de cada requisição enviada.

O registro estruturado de eventos internos (*logging*), contemplando os pacotes de dados de envio e recebimento (*payloads*), gera um histórico contínuo que viabiliza a auditoria precisa das métricas operacionais da rede. No que tange ao tratamento de exceções e à resiliência, o sistema intercepta os retornos de erro da Graph API e categorizá-los estruturalmente, como limite de requisições (*rate limited*), falhas de segurança (auth_or_permission) e instabilidades da plataforma (server_error). Essa categorização impede que falhas silenciosas na comunicação interfiram no registro do atendimento institucional.

O ciclo de vida da interação é acionado no momento em que o usuário envia a mensagem inicial. O tráfego de dados gerado por essa ação é interceptado por um *Webhook* configurado na plataforma, que atua como a interface de entrada (*gateway*) para a infraestrutura do sistema. Neste momento, antes de qualquer processamento semântico, o algoritmo de roteamento inicial intercepta o pacote de dados e realiza uma verificação estrutural para validar se a conexão de origem possui histórico de interações prévias, estejam elas ativas ou já encerradas. Essa etapa desenvolve-se a partir de uma bifurcação lógica: caso seja identificada alguma sessão anterior, o sistema reconhece que o usuário já foi exposto às diretrizes institucionais, dispensando a repetição do fluxo de recepção e direcionando a mensagem diretamente para o processamento semântico.

Por outro lado, caso não seja detectado nenhum histórico de sessão na bifurcação lógica (caracterizando o primeiro contato do usuário com o assistente), o sistema interrompe o fluxo principal para acionar o processo estrutural de recepção (*onboarding*). Nessa etapa, é enviada uma mensagem de boas-vindas padronizada e estática, que atua como um manifesto de governança visando evitar a antropomorfização excessiva e o fenômeno de excesso de confiança (*overreliance*) na inteligência artificial por parte do estudante.

A mensagem de recepção delimita o escopo de atuação do assistente, apresentando explicitamente as macroáreas que ele está apto a processar: calendário acadêmico, normativas de TCC, regras do SUAP e matrizes curriculares. Em seguida, estabelece que as respostas têm caráter estritamente informativo e não substituem editais nem o atendimento humano oficial. Por fim, cumpre as diretrizes éticas do projeto ao informar que as interações textuais são retidas por até sete dias, que os diálogos poderão ser submetidos a auditoria humana para ajuste dos parâmetros algorítmicos e que submissões abusivas, ofensivas ou fora do escopo acadêmico poderão ser descartadas. Somente após a resposta do usuário a esse termo de uso interativo a entrada de texto é encaminhada ao fluxo de vetorização e busca semântica.

Superada essa etapa, o controle de sessão atua como o mecanismo central para gerenciar de forma dinâmica o ciclo de vida do fluxo contínuo. A sessão estabelece a fronteira de delimitação temporal, de modo que, a cada nova entrada, o sistema resgata exclusivamente o acervo de mensagens pertencente àquele escopo ativo. Essa estratégia estruturada contextualiza o motor semântico, permitindo que a aplicação compreenda a progressão do diálogo e mantenha a coerência temática. Para otimizar os recursos computacionais, propõe-se que o encerramento da sessão não opere sob a imposição de um tempo limite estático (*timeout*). Em vez disso, a arquitetura delega a deliberação de fechamento contextual ao próprio LLM, que emite a *tag* [finish] ao constatar a resolução plena da demanda.

A partir da interceptação desse comando semântico de encerramento, o orquestrador dispara automaticamente uma solicitação de avaliação (*feedback*) por meio dos menus nativos do aplicativo. Através desse recurso, o usuário é convidado a qualificar o atendimento atribuindo uma nota quantitativa de 1 a 5 estrelas, com a opção de submeter um comentário qualitativo. Esses dados coletados são imediatamente persistidos e vinculados de forma relacional ao identificador único da sessão de atendimento, correlação vital que permite o rastreamento exato da premissa normativa que motivou a avaliação. As informações alimentam um painel gerencial que calcula de forma autônoma a média móvel das avaliações dos últimos sete dias, consolidando métricas rigorosas que visam viabilizar a auditoria direta da usabilidade do sistema.

5 CONCLUSÃO

O presente estudo buscou descrever o desenvolvimento e a modelagem arquitetural do RoBoto, um agente conversacional projetado especificamente para o contexto acadêmico do IFRO – Campus Porto Velho Calama. Conforme introduzido nas premissas iniciais da pesquisa, o objetivo central consistiu em propor uma solução tecnológica capaz de mitigar a fricção no acesso a normativas, editais e dados transacionais, buscando democratizar e agilizar o atendimento institucional. A proposta metodológica baseou-se na implementação de um sistema pautado pela arquitetura RAG, visando aliar a capacidade cognitiva de um LLM à segurança de uma base de conhecimento controlada.

Ao longo da estruturação do projeto, a infraestrutura proposta alcançou uma consolidação metodológica satisfatória de seus pilares. A implementação do protótipo funcional, integrando o fluxo de recuperação vetorial ao módulo de orquestração de contextos dinâmicos e à rigorosa parametrização defensiva, indica a viabilidade técnica de fornecer respostas precisas com um risco mitigado de alucinações. Além disso, a orquestração de serviços vivos, como a extração de planilhas em nuvem e

a consulta de horários via banco de dados relacional, demonstra que o sistema possui a fundação operacional necessária para atuar de forma fluida por meio do WhatsApp.

Apesar dos indícios positivos observados na concepção da arquitetura, reconhece-se que a aplicação apresenta um vasto campo para evolução técnica. Como propostas de trabalhos futuros, vislumbra-se primordialmente a expansão das capacidades multimodais do agente. Pretende-se investigar a viabilidade de implementar o suporte nativo para a transcrição e o processamento de mensagens de voz, bem como a interpretação autônoma de imagens submetidas pelos usuários. Estima-se que a adoção dessas camadas de interação possa reduzir ainda mais as barreiras de acessibilidade, viabilizando uma comunicação ainda mais orgânica com o sistema.

Por fim, a validação empírica é a etapa fundamental para a consolidação científica do projeto. Projeta-se a execução de testes de usabilidade em ambiente real de produção, com acesso simultâneo de alunos e servidores, capazes de mensurar a assertividade semântica do modelo diante de interações complexas. Paralelamente, a coleta das avaliações dos usuários permitirá quantificar o benefício prático percebido pela comunidade discente e investigar o impacto da ferramenta na rotina institucional, medindo em que grau a automação do atendimento mitiga a sobrecarga de dúvidas direcionadas aos setores acadêmicos e administrativos. Esses indicadores fornecerão os insumos necessários ao refinamento e à eventual adoção oficial da solução.

AGRADECIMENTOS

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO) pelo apoio ao projeto de pesquisa “Desenvolvimento de um *ChatBot* com Inteligência Artificial para atender Discentes do IFRO”, financiado por meio do Edital n° 3/2025/REIT/PROPESP/IFRO, de 10 de junho de 2025. Agradecemos também ao Departamento de Pesquisa, Inovação e Pós-Graduação do IFRO – Campus Porto Velho Calama, pelo apoio financeiro à publicação deste trabalho, concedido por meio do Edital de Apoio à Comunicação Científica e Literária n° 43/2026/PVCAL.

REFERÊNCIAS

ADAMOPOULOU, E.; MOUSSIADES, L. Chatbots: History, technology, and Applications. Machine Learning with Applications, v. 2, n. 100006, 15 dez. 2020. Disponível em: https://doi.org/10.1016/j.mlwa.2020.100006. Acesso em: 24 ago. 2026.

ALMEIDA, F.; XEXÉO, G. Word Embeddings: A Survey. [S. l.]: arXiv, 2019. Preprint. Disponível em: https://arxiv.org/abs/1901.09069. Acesso em: 24 ago. 2026.

ASAI, A. *et al.* Self-RAG: Learning to retrieve, generate, and critique through self-reflection. International Conference on Learning Representations (ICLR), 2024. Disponível em: https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf. Acesso em: 25 ago. 2026.

BALAGUER, A. *et al.* RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. [S. l.]: arXiv, 2024. Preprint. Disponível em: https://arxiv.org/abs/2401.08406. Acesso em: 24 ago. 2026.

BROWN, T. *et al.* Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems (NeurIPS). 2020.

CHANG, Y. *et al.* A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, v. 15, n. 3, p. 1-45, 2023. Disponível em: https://dl.acm.org/doi/10.1145/3641289. Acesso em: 25 ago. 2026.

CODEIGNITER. CodeIgniter 4. [S. l.]: CodeIgniter, 2026. Disponível em: https://www.codeigniter.com/. Acesso em: 24 ago. 2026.

FÉLSTAD, A.; BRANDTZÆG, P. B. Chatbots and the New World of HCI. Interactions, v. 24, n. 4, p. 38–42, 2017.

GAO, Y. *et al.* Retrieval-augmented generation for large language models: A survey. [S. l.]: arXiv, 2023. Preprint. Disponível em: https://arxiv.org/abs/2312.10997. Acesso em: 25 ago. 2026.

GIL, A. C. Como elaborar projetos de pesquisa. 7. ed. São Paulo: Atlas, 2022.

HU, E. J. *et al.* LoRA: Low-Rank Adaptation of Large Language Models. In: International Conference on Learning Representations (ICLR), 2022.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA (IFRO). Cursos. Porto Velho, 2026a. Disponível em: https://portal.ifro.edu.br/cursos. Acesso em: 26 ago. 2026.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA (IFRO). Resoluções. Conselho Superior (CONSUP), Porto Velho, 2026b. Disponível em: https://portal.ifro.edu.br/consup-nav/resolucoes. Acesso em: 26 ago. 2026.

JI, Z. *et al.* Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, v. 55, n. 12, p. 1-38, 2023. Disponível em: https://doi.org/10.1145/3571730. Acesso em: 24 ago. 2026.

JOHNSON, J.; DOUZE, M.; JÉGOU, H. Billion-Scale Similarity Search with GPUs. IEEE Transactions on Big Data, v. 7, n. 3, p. 535–547, 2021.

JURAFSKY, D.; MARTIN, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3rd ed. draft. [S. l.]: Stanford University; University of Colorado Boulder, 2024. Disponível em: https://sandiway.arizona.edu/ling538-23/1.pdf. Acesso em: 24 ago. 2026.

KLEPPMANN, M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media, 2017.

LEWIS, P. *et al.* Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems (NeurIPS). v. 33, 2020. p. 9459–9474.

NOGUEIRA, R.; CHO, K. Passage Re-ranking with BERT. [S. l.]: arXiv, 2019. Preprint. Disponível em: https://arxiv.org/abs/1901.04085. Acesso em: 24 ago. 2026.

PACETE, L. G. O que fez do WhatsApp um fenômeno das mensagens? O Brasil é a resposta. Forbes Brasil, São Paulo, 2 maio 2025. Disponível em: https://forbes.com.br/forbes-tech/2025/05/o-que-fez-do-whatsapp-um-fenomeno-das-mensagens-o-brasil-e-a-resposta/. Acesso em: 24 ago. 2026.

PHP. PHP: Hypertext Preprocessor. [S. l.]: The PHP Group, 2026. Disponível em: https://www.php.net/. Acesso em: 24 ago. 2026.

REIMERS, N.; GUREVYCH, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: Association for Computational Linguistics, 2019. p. 3982–3992.

VASWANI, A. *et al.* Attention Is All You Need. In: Advances in Neural Information Processing Systems (NeurIPS). v. 30, 2017. p. 5998–6008.

WANG, J. *et al.* Milvus: A Purpose-Built Vector Data Management System. In: Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21). New York: Association for Computing Machinery, 2021. p. 2614–2627.

YIN, R. K. Estudo de caso: planejamento e métodos. 5. ed. Porto Alegre: Bookman, 2015.