Desenvolvimento de um Chatbot para resolução de dúvidas acadêmicas no IFSUL Campus Passo Fundo

Vinícius Dal Prá[1]
Prof. Dr. João Mário Lopes Brezolin[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia IFSUL-rio-grandense - Campus Passo Fundo

viniciuspra.pf033@academico.ifsul.edu.br

Abstract. *This work presents the development of an academic chatbot aimed at students of the Computer Science program at IFSUL - Passo Fundo Campus. The goal is to provide a virtual assistant capable of answering common questions related to the course, such as workload, regulations, subjects, faculty contacts, and other institutional information. To achieve this, an architecture based on the Retrieval-Augmented Generation (RAG) technique was implemented, using official institutional documents in PDF format as the knowledge base. The solution aims to optimize access to academic information, promoting greater student autonomy and contributing to improved communication between students and the institution. User testing indicated good usability and response accuracy. Participants reported satisfaction with the system, highlighting its practicality and ease of use.*

Resumo. *Este trabalho apresenta o desenvolvimento de um chatbot acadêmico voltado aos alunos do curso de Ciência da Computação do IFSUL - Campus Passo Fundo. O objetivo é oferecer um assistente virtual capaz de responder dúvidas recorrentes relacionadas ao curso, como carga horária, regulamentos, disciplinas, contatos de professores e demais informações institucionais. Para isso, foi implementada uma arquitetura baseada na técnica de Recuperação Aumentada por Geração (RAG), utilizando como base de conhecimento documentos oficiais da instituição em formato PDF. A solução visa otimizar o acesso às informações acadêmicas, proporcionando maior autonomia aos estudantes e contribuindo para a melhoria da comunicação entre discentes e a instituição. Os testes realizados com usuários indicaram boa usabilidade e eficiência nas respostas. Os participantes demonstraram satisfação com o sistema, destacando sua praticidade e facilidade de uso.*

1. Introdução

É comum que alunos do Campus Passo Fundo do Instituto Federal de Educação Ciência e Tecnologia Sul-Riograndense (IFSUL) terem dúvidas sobre diversos processos acadêmicos, como: como encaminhar a documentação para validação das horas complementares, quais são os documentos exigidos, ou ainda quais os requisitos para cursar a disciplina de Trabalho de Conclusão de Curso (TCC). Essas e outras dúvidas fazem parte do cotidiano dos discentes levando-os a buscar orientação junto a coordenadores, professores e demais membros da instituição. Segundo Bulhões (2020), “a agilidade no retorno

aos questionamentos dos estudantes evitará a desmotivação, podendo contribuir para a redução do número de evasões" (BULHOES et al., 2020).

Com o avanço das tecnologias ligadas à Inteligência Artificial (IA), tornou-se possível desenvolver ferramentas capazes de aprimorar a gestão de processos acadêmicos, facilitando o trabalho de coordenadores, professores e tutores (BULHOES et al., 2020). Nesse contexto, observa-se que a criação de um chatbot institucional voltado ao suporte acadêmico pode simplificar o processo de busca por informações, otimizando o tempo tanto de alunos quanto de servidores do campus.

Dessa forma, este trabalho propõe o desenvolvimento de um chatbot para o curso de Ciência da Computação. O sistema será alimentado por informações e documentos disponibilizados no site institucional, incluindo regras sobre horas complementares, estrutura curricular (versões antiga e nova), contatos de professores e coordenadores, editais, eventos, além de outras informações relevantes sobre o curso e o Instituto. Para a construção do chatbot, será utilizado o *framework* LangChain, em conjunto com o modelo GPT-4o da OpenAI, implementado em Python. O sistema será capaz de responder às dúvidas dos estudantes de forma contextualizada, rápida e precisa, utilizando técnicas de Recuperação de Informação com base nos documentos institucionais. Com essa aplicação, busca-se democratizar o acesso às informações acadêmicas, simplificar o processo de consulta, otimizar o atendimento institucional e, sobretudo, contribuir para a melhoria da experiência acadêmica dos estudantes.

2. Revisão Bibliográfica

Nesta seção, será realizada uma revisão da literatura, abordando os principais conceitos, ferramentas e tecnologias que fundamentam o desenvolvimento de chatbots baseados em inteligência artificial, com foco especial na aplicação dessas ferramentas no contexto educacional. Essa revisão tem como objetivo fornecer base teórica para a metodologia proposta neste trabalho.

2.1. Chatbot

Um chatbot é um programa ou aplicativo com o qual os usuários podem conversar por voz ou texto. Eles foram desenvolvidos pela primeira vez na década de 1960, e a tecnologia que os impulsiona mudou com o tempo. Tradicionalmente, os chatbots usam regras predefinidas para conversar com os usuários e fornecer respostas com script. Os chatbots contemporâneos utilizam o Processamento de Linguagem Natural (PLN) para entender os usuários e responder a perguntas complexas com grande profundidade e precisão. As organizações podem usar chatbots para escalar, personalizar e melhorar a comunicação em áreas como fluxos de trabalho de atendimento ao cliente (Amazon Web Services, 2024a).

2.2. Processamento de Linguagem Natural

O Processamento de Linguagem Natural é um conjunto de tecnologias que permite aos sistemas captar, compreender e manipular a linguagem humana. O PLN está fortemente ligado à Inteligência Artificial (IA) e depende de conceitos fundamentais como o aprendizado de máquina (machine learning), que capacitam a IA a melhorar suas respostas e interpretações de forma contínua. O PLN teve suas raízes no desenvolvimento de

sistemas baseados em regras nos anos 1950 e 1960, como o ELIZA, um dos primeiros programas de conversação. Com o passar do tempo, os métodos baseados em estatística substituíram as abordagens manuais, permitindo maior escalabilidade e precisão. A partir da década de 2010, o surgimento de redes neurais profundas e, mais especificamente, dos modelos transformers, como BERT e GPT, revolucionou o PLN, possibilitando avanços significativos na compreensão e geração de linguagem natural. Os transformers são um tipo avançado de rede neural projetado para processar dados textuais de forma eficiente, permitindo que modelos como GPT-4 compreendam o contexto completo de uma frase ou parágrafo. Por meio de um mecanismo de atenção, eles conseguem identificar as partes mais relevantes do texto para analisar ou gerar respostas, mesmo em conteúdos complexos ou de longa extensão.

Os modelos de linguagem modernos, baseados em redes neurais profundas, utilizam arquiteturas como os transformers, que aproveitam mecanismos de atenção para processar texto de maneira mais eficiente e contextualizada. Modelos como GPT-3 e GPT-4 são pré-treinados em grandes conjuntos de dados e podem ser ajustados para tarefas específicas. Esses modelos conseguem gerar texto coerente, responder perguntas e realizar tarefas complexas, sendo fundamentais para sistemas como o LangChain, que aproveitam essa capacidade para criar aplicações robustas de PLN. ”O processamento de linguagem natural usa machine learning para revelar a estrutura e o significado do texto. Com aplicativos de processamento de linguagem natural, as organizações podem analisar textos e extrair informações.”(Google Cloud, 2024). Em chatbots, o PLN desempenha um papel essencial ao interpretar as entradas dos usuários, identificar intenções e extrair entidades relevantes. Com isso, é possível gerar respostas precisas e contextualizadas, personalizando a interação. Modelos como GPT utilizam o PLN para analisar o significado das perguntas, mesmo quando são feitas de forma ambígua ou informal, oferecendo soluções rápidas e eficientes. Além disso, técnicas como busca semântica e ajuste fino permitem que os chatbots sejam treinados em bases de dados específicas, como no caso deste projeto, que visa fornecer informações acadêmicas de forma acessível e relevante.

2.3. Grandes Modelos de Linguagem (LLM’s)

“Os grandes modelos de linguagem são compostos por múltiplas camadas de redes neurais, que trabalham em conjunto para analisar textos e prever o que vem em seguida (como na busca do Google). Os Large Language Models (LLM’s) são treinados com transformadores bidirecionais, que atuam para maximizar a probabilidade de acertar quais as palavras que antecedem e precedem determinados termos dentro de um contexto – da mesma forma que os humanos conseguem “adivinhar” quais palavras virão a seguir em uma frase. LLMs contam ainda com um mecanismo de atenção que permite a eles focar seletivamente em partes do texto, de modo a identificar os trechos mais relevantes para fazer resumos, por exemplo.”(NEELAKANDAN, 2023). Data Science Academy (2024) destaca que, Os LLMs têm visto uma série de avanços significativos nos últimos anos. Por exemplo, o GPT-3 da OpenAI, lançado em 2020, tem 175 bilhões de parâmetros e ficou famoso ao gerar texto preciso a partir de entradas feitas no ChatGPT. Outras melhorias incluem avanços na compreensão de contexto de longo alcance, a capacidade de gerar respostas mais coerentes e relevantes e a capacidade de entender e responder a uma variedade maior de entradas de texto.

Há uma série de aplicações potenciais para LLMs. Eles são frequentemente usa-

dos para tarefas como responder perguntas, escrever redações, traduzir texto, resumir documentos, gerar código em linguagem de programação e muito mais. Eles também são usados em chatbots, assistentes digitais e em muitas outras aplicações onde a geração ou compreensão de texto é necessária.

2.4. Recuperação Aumentada por Geração (RAG)

A técnica conhecida como RAG *(Retrieval-Augmented Generation)* busca melhorar a qualidade e a precisão das respostas geradas por grandes modelos de linguagem (LLMs) ao combinar sua capacidade de geração com mecanismos externos de recuperação de informações. Com isso, o modelo pode acessar bases de dados específicas ou documentos institucionais durante o processo de geração, sem a necessidade de retreinamento. Essa abordagem permite que o modelo forneça respostas mais atualizadas e contextualizadas, mesmo em domínios muito específicos (ALURA, 2024).

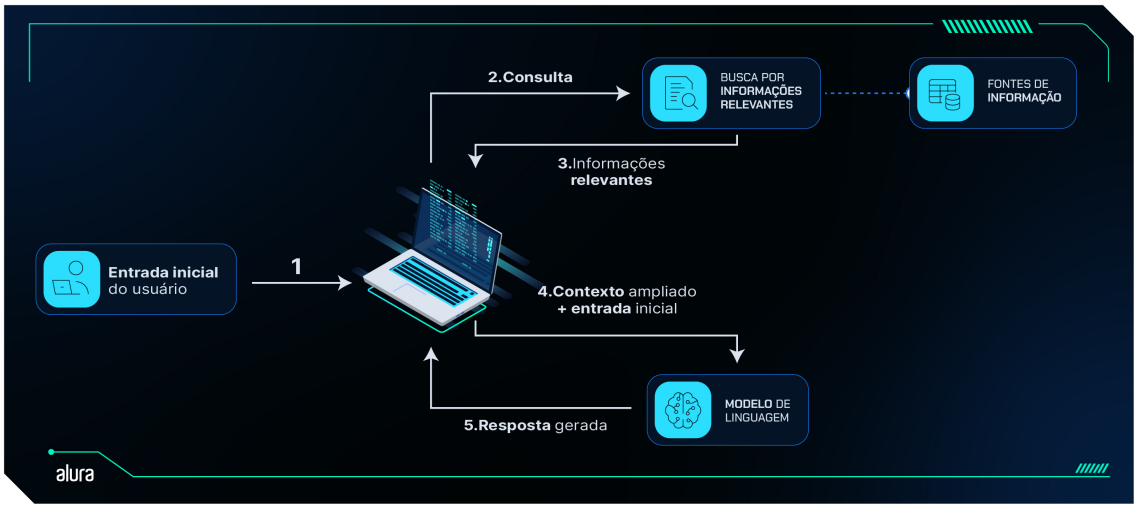

Figura 1. Fluxo de trabalho de um chatbot com RAG. Fonte: Alura (2024).

3. Tecnologias utilizadas no desenvolvimento do chatbot

Para o desenvolvimento do chatbot proposto, foram selecionadas tecnologias específicas que possibilitam a implementação das funcionalidades requeridas e garantem desempenho e escalabilidade. Abaixo, são descritas as principais ferramentas e frameworks que serão utilizados no projeto.

3.1. Python

A linguagem Python se tornou uma das linguagens de programação mais populares do mundo nos últimos anos. Isso se deve, principalmente, à sua versatilidade: ele funciona para o aprendizado de máquinas, construção de sites e até para automação de tarefas e testes de softwares. Além disso, sua proximidade com a linguagem humana faz com que muitas pessoas o utilizem, tanto quem desenvolve quanto quem necessariamente faz parte desse mercado. Em resumo, a escolha do Python foi motivada pela familiaridade com a linguagem e pela sua simplicidade, que facilita o desenvolvimento de aplicações baseadas em modelos de linguagem de grande escala (CARVALHO, 2024). A utilização

do Python no desenvolvimento deste chatbot é fundamental devido à sua ampla adoção na área de inteligência artificial e processamento de linguagem natural. Python oferece integração com bibliotecas robustas que são essenciais para trabalhar com modelos de linguagem avançados. Sua sintaxe simples facilita o desenvolvimento e a manipulação de dados.

3.2. LangChain

O LangChain é uma estrutura de código aberto para criar aplicações baseadas em grandes modelos de linguagem (LLMs). O LangChain fornece ferramentas e abstrações para melhorar a personalização, a precisão e a relevância das informações que os modelos geram. Por exemplo, os desenvolvedores podem usar componentes do LangChain para criar novas correntes de prompts ou personalizar modelos existentes. O LangChain também inclui componentes que permitem que os LLMs acessem novos conjuntos de dados sem retreinamento(Amazon Web Services, 2024b).

O LangChain facilita a construção de chatbots e outras aplicações que utilizam modelos de linguagem como o GPT. Ele permite a criação de fluxos de conversação dinâmicos e inteligentes, integrando modelos de linguagem com outras fontes de dados, ferramentas e APIs externas. Uma das principais vantagens do LangChain é o suporte a memória, o que permite que o chatbot mantenha o contexto das conversas passadas, tornando as interações mais naturais e personalizadas.

3.3. Streamlit

Segundo a Asimov Academy (Asimov Academy, 2023), ”Streamlit é uma biblioteca *open-source* em Python que permite a criação de aplicativos web para análise de dados de forma extremamente rápida”. A ferramenta permite transformar scripts de dados em web apps com poucas linhas de código, sendo especialmente útil para cientistas de dados, analistas e desenvolvedores que desejam interagir com dados de forma dinâmica, sem a necessidade de conhecimento avançado em desenvolvimento web.

O Streamlit facilita a integração com outras bibliotecas Python, proporcionando uma maneira ágil de exibir resultados, como o histórico de chat e as respostas do modelo de linguagem. Sua capacidade de atualização dinâmica e renderização em tempo real é fundamental para manter a interação fluida entre o usuário e o chatbot. Além disso, a facilidade de deploy e a integração direta com o código Python tornam o Streamlit uma escolha fundamental para o desenvolvimento desse chatbot.

3.4. GPT-4o

Segundo a IBM (IBM, 2024), o GPT-4o é um modelo multimodal e multilíngue da OpenAI, lançado em maio de 2024, capaz de processar texto, imagem, áudio e vídeo em sua entrada e gerar também imagens como saída. Esse modelo é mais poderoso do que seus antecessores, as versões GPT 3.5 e GPT 4, entregando respostas mais rápidas e precisas.

No caso específico deste projeto, o modelo GPT-4o foi escolhido por sua capacidade de gerar respostas assertivas e naturais, utilizando modelos treinados em grandes volumes de dados. A utilização desse modelo possibilita que o sistema forneça respostas mais contextuais, precisas e com a maior velocidade possível, baseadas nos dados recuperados, sem a necessidade de treinamento adicional ou configuração complexa.

3.5. FAISS e Embeddings

Segundo a documentação oficial do Faiss (2025), o FAISS é uma biblioteca para busca eficiente de similaridade e agrupamento de vetores densos. Ela contém algoritmos que buscam em conjuntos de vetores de qualquer tamanho, até aqueles que possivelmente não cabem na RAM. Além disso, oferece código de suporte para avaliação e ajuste de parâmetros (Facebook AI Research, 2025).

Sua principal aplicação é em sistemas de recuperação de informações, como a busca semântica, onde é necessário comparar e localizar documentos ou trechos de texto que são semanticamente semelhantes a uma consulta. Para este projeto, o FAISS é utilizado para indexar e buscar informações relevantes nos documentos PDF carregados pelo sistema, utilizando embeddings (representações vetoriais) de texto. Essas representações são geradas a partir de modelos como o HuggingFace, o que permite capturar o significado semântico dos textos.

O uso do FAISS torna a busca nos documentos mais eficiente e escalável, permitindo que o chatbot forneça respostas rápidas e precisas, mesmo quando o volume de dados é grande.

3.6. HuggingFace

Segundo Rebelo (2025), o Hugging Face é uma plataforma online dedicada à ciência de dados e aprendizado de máquina. Ela permite que os usuários naveguem por conjuntos de dados, treinem modelos de IA, compartilhem seu trabalho e colaborem com outras pessoas. Além disso, oferece a biblioteca Transformer, uma coleção de APIs prontas para uso para todos os modelos de IA da plataforma, facilmente acessíveis com poucas linhas de código (REBELO, 2025).

Neste projeto, o HuggingFace é utilizado principalmente para a geração de embeddings, que são representações numéricas dos textos que permitem que o modelo capture o significado semântico e realize a comparação eficiente entre as consultas do usuário e os documentos armazenados. O modelo específico de embeddings utilizado é o ”BAAI/bge-m3”, que é altamente eficaz na captura de representações semânticas precisas de textos e na busca por correspondências relevantes.

A biblioteca também oferece ferramentas para trabalhar com modelos de linguagem e realizar o ajuste fino para tarefas específicas, como a personalização das respostas do chatbot. Com a integração do HuggingFace ao LangChain, o projeto ganha em performance e flexibilidade na manipulação de dados textuais.

3.7. PyPDFLoader

”O pypdf é uma biblioteca PDF em Python puro, gratuita e de código aberto, capaz de dividir, mesclar, recortar e transformar as páginas de arquivos PDF. Ele também pode adicionar dados personalizados, opções de visualização e senhas a arquivos PDF. O pypdf também pode recuperar texto e metadados de PDFs”(pypdf, 2025).

O uso do PyPDFLoader é fundamental para o projeto, pois ele permite que o chatbot interaja com documentos de maneira transparente, extraindo o texto de maneira eficiente e sem a necessidade de intervenção manual. A flexibilidade dessa ferramenta garante que o sistema possa lidar com diferentes tipos de PDFs, permitindo a expansão do chatbot conforme mais documentos são adicionados.

4. Trabalhos Relacionados

A área de chatbots acadêmicos e automação de respostas para dúvidas de estudantes tem se expandido rapidamente nos últimos anos, especialmente com o avanço dos Modelos de Linguagem Grande (LLMs) e do Processamento de Linguagem Natural (PLN). Esses sistemas permitem que as universidades ofereçam atendimento rápido e eficiente para as dúvidas comuns dos alunos, reduzindo a carga sobre o corpo administrativo e melhorando a experiência do usuário.

Wicks (2023), desenvolveu, na Universidade Federal da Bahia (UFBA), um sistema denominado CHATBOT-POLI com o objetivo de automatizar respostas a perguntas acadêmicas utilizando a tecnologia da OpenAI. O projeto foi implementado com Python para o backend e React para o frontend, integrando técnicas de PLN e LLMs para fornecer respostas precisas e contextuais. Diferenciando-se de sistemas anteriores, o CHATBOT-POLI consulta documentos específicos, como resoluções do colegiado, diretamente em arquivos PDF. Isso permite que o chatbot ofereça respostas baseadas em fontes oficiais e atualizadas, uma característica importante para garantir a confiabilidade das informações fornecidas.

Para validar o modelo, Wicks realizou testes baseados em perguntas reais de estudantes e comparou as respostas do chatbot com as resoluções do colegiado. Os resultados foram promissores, apresentando uma média de 7,6 em conformidade com as resoluções oficiais, o que reforça a precisão do sistema na tarefa proposta. Esse trabalho contribui para o avanço na área ao demonstrar uma aplicação prática de LLMs e PLN em um contexto acadêmico, oferecendo um modelo escalável e de fácil atualização, que facilita o acesso à informação por parte dos alunos da UFBA.

Neves (2024), da Universidade Federal do Rio Grande do Norte (UFRN), desenvolveu uma plataforma inovadora de IA Generativa voltada para otimizar o acesso dos estudantes a regulamentos, históricos acadêmicos e outros documentos institucionais. A plataforma integra diversas tecnologias, incluindo o runtime Bun do JavaScript, RabbitMQ para controle de mensagens, e o Google Vertex AI para garantir a escalabilidade com modelos como Claude 3.5 e Llama 3.1. O projeto é único por utilizar a técnica de Recuperação Aumentada de Geração (RAG), que combina geração de respostas com a recuperação de informações de bases de dados, fornecendo respostas mais precisas e relevantes aos usuários. A plataforma foi testada com interações baseadas em texto e áudio, usando documentos oficiais da UFRN, como regulamentos institucionais. Os resultados indicaram que o uso do RAG aumentou a precisão das respostas, possibilitando uma interação mais eficiente para os estudantes. A arquitetura da plataforma foi projetada para suportar alta demanda, com ênfase na segurança dos dados dos usuários, uma característica essencial para aplicações em ambientes acadêmicos de grande escala.

A evolução dos agentes virtuais para atendimento ao cliente é uma área em constante desenvolvimento, especialmente com a crescente demanda por interações mais humanizadas e eficientes. Diversos estudos buscam aprimorar a capacidade dos chatbots de entender e responder com naturalidade às solicitações dos usuários, contribuindo para um atendimento mais satisfatório e reduzindo a necessidade de intervenção humana em demandas rotineiras.

Lara et al. (2023), da Universidade Federal de Itajubá, desenvolveram um agente

virtual chamado Alice para atendimento bancário, utilizando o framework LangChain e o modelo GPT-3.5 da OpenAI. Esse projeto foi concebido para que o agente possa compreender e responder de forma humanizada às demandas dos clientes, sendo capaz de identificar solicitações por meio do modelo BERT-Banking77. Diferente de abordagens anteriores, Alice foi projetada para atender a uma ampla gama de pedidos bancários comuns, respondendo com maior precisão e naturalidade. O agente foi avaliado em simulações de casos de uso reais, incluindo questões como a chegada de um cartão novo. Os resultados mostraram que Alice foi eficaz em identificar e responder corretamente a solicitações diretas, mas encontrou dificuldades para manter a continuidade do atendimento em interações mais complexas. A avaliação qualitativa realizada destacou os sucessos do agente em cenários simples e a necessidade de melhorias em sua lógica de atendimento para aprimorar a experiência do usuário em situações que exigem um entendimento mais profundo do contexto. Esse trabalho contribui para o desenvolvimento de agentes mais avançados, ao demonstrar o potencial do GPT-3.5 e do BERT-Banking77 para a criação de um atendimento bancário mais humanizado, ao mesmo tempo que identifica as limitações que ainda precisam ser superadas na área de agentes virtuais para atendimento ao cliente.

Almeida, Almeida e Araujo, do Instituto de Ciências Exatas e Naturais (ICEN) da Universidade Federal do Pará (UFPA), desenvolveram o AnneBot, um chatbot destinado a apoiar a formação de docentes, especialmente no ensino de pensamento computacional. Projetado para facilitar o aprendizado e auxiliar professores, o AnneBot utiliza interações automatizadas para guiar usuários em conceitos fundamentais de computação. Testado com estudantes de licenciatura, o chatbot demonstrou eficácia na resolução de dúvidas e no estímulo ao aprendizado autônomo, aumentando o engajamento, embora melhorias no sistema de interação tenham sido sugeridas.

Santos et al. (2023) desenvolveram o chatbot ”Alfa”, implementado no Ambiente Virtual de Aprendizagem (AVA) Moodle. O objetivo do chatbot é automatizar a interação com os alunos, respondendo a perguntas frequentes sobre conteúdo do curso, políticas acadêmicas e outros temas. A implementação foi baseada em um sistema de perguntas e respostas, utilizando uma abordagem simples e eficaz para fornecer respostas rápidas. A eficácia do chatbot foi testada com usuários reais, que indicaram melhorias na acessibilidade às informações e na otimização do tempo gasto com dúvidas.

5. Desenvolvimento do Sistema Proposto

Para viabilizar o funcionamento do chatbot acadêmico, os documentos institucionais foram inicialmente coletados a partir do site oficial do Campus Passo Fundo. Esses documentos incluem regulamentos, diretrizes do curso, informações sobre professores e demais conteúdos de interesse dos alunos. Todo o material foi organizado em uma pasta local denominada docs/, em formato PDF. Para otimizar a extração das informações, conteúdos dispersos foram reunidos e estruturados em novos arquivos, como, por exemplo, um documento consolidado contendo os nomes dos professores, suas áreas de atuação e informações de contato.

O sistema é composto por três etapas principais. A primeira é o pré-processamento dos documentos, no qual cada PDF é carregado por meio da biblioteca PyPDFLoader. Em seguida, o conteúdo textual é segmentado em blocos de até 512 caracteres, com sobreposição de 64, utilizando o RecursiveCharacterTextSplitter. Esses blocos são trans-

formados em vetores semânticos por meio do modelo de embeddings BAAI/bge-m3, que permite a representação vetorial do conteúdo textual.

Na segunda etapa, ocorre a indexação e busca semântica. Os vetores gerados são armazenados utilizando a biblioteca FAISS, que permite realizar buscas eficientes com base em similaridade vetorial. A recuperação das informações é feita com a estratégia de MMR *Maximal Marginal Relevance*, que equilibra a relevância e diversidade dos resultados, retornando os blocos mais adequados à pergunta do usuário.

A terceira etapa consiste na geração da resposta com base no contexto. O sistema utiliza um retriever configurado para levar em conta o histórico das interações anteriores. A partir dos blocos recuperados, o modelo GPT-4o, da OpenAI, é responsável por gerar as respostas. Para isso, foram elaborados prompts específicos que instruem o modelo a utilizar somente as informações contidas nos documentos como base para suas respostas, garantindo precisão e confiabilidade.

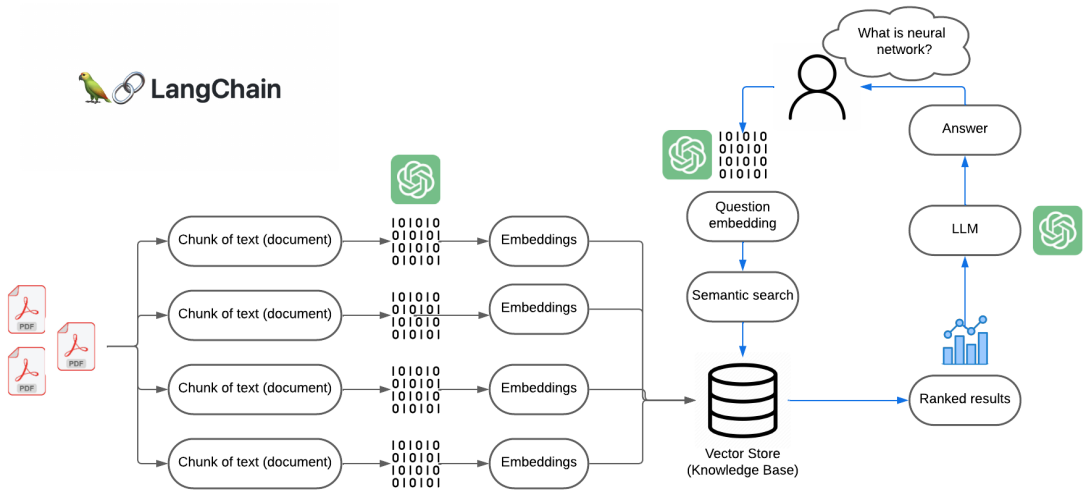

Figura 2. Lógica do funcionamento do chatbot. Fonte: Autor (2024)

A interface do usuário foi desenvolvida com a biblioteca Streamlit, permitindo uma experiência interativa por meio de um chat simples e funcional. O usuário pode digitar perguntas, visualizar respostas e acompanhar todo o histórico da conversa de maneira contínua.

Além disso, foi implementado um sistema de persistência de sessão, que permite ao sistema manter o contexto das interações entre usuário e assistente ao longo da sessão, garantindo que as respostas considerem o histórico completo das perguntas anteriores.

Por fim, diversos parâmetros foram ajustados para garantir a eficácia do sistema. O modelo de linguagem utilizado foi o GPT-4o da OpenAI, com temperatura configurada para 0.1, a fim de gerar respostas mais objetivas. O modelo de embeddings escolhido foi o BAAI/bge-m3, com normalização ativada. O tamanho dos blocos de texto *chunks* foi definido como 512, com sobreposição de 64.

Figura 3. Interface do Chatbot Fonte: Autor (2025)

6. Resultados e Discussões

6.1. Avaliação com os usuários

Após a conclusão do sistema proposto, com objetivo de avaliar a precisão das respostas, a facilidade de uso e o potencial de adoção do sistema por outros estudantes, foi realizado um teste com dois alunos do curso de Ciência da Computação do IFSUL – Câmpus Passo Fundo. Cada aluno teve acesso ao chatbot e pôde interagir livremente com o sistema, formulando perguntas de interesse pessoal relacionadas ao curso. Em seguida, foi aplicado um questionário qualitativo com as seguintes perguntas:

Nº	Questão
1	O sistema respondeu corretamente aos seus questionamentos?
2	O que você achou da usabilidade do sistema? Achou prático/intuitivo?
3	Você recomendaria que esse sistema fosse utilizado por outros alunos? Explique.

Em relação à precisão das respostas, ambos os participantes indicaram que o sistema respondeu corretamente aos questionamentos realizados. Esse retorno positivo demonstra que o modelo foi capaz de recuperar informações relevantes dos documentos institucionais e fornecer respostas satisfatórias, condizentes com o esperado.

Com base nas respostas fornecidas pelos dois participantes, é possível observar uma aceitação positiva quanto ao funcionamento e à usabilidade do chatbot proposto. Segundo o aluno 1, “o sistema é fácil e intuitivo de utilizar, não deixando dúvidas em seu funcionamento”. Já o aluno 2 destaca que “o sistema é de fácil entendimento e extremamente intuitivo e prático”. Por fim, ambos os participantes afirmaram que recomendariam o sistema para outros alunos, mencionando o potencial de facilitar o acesso às informações institucionais e de servir como uma ferramenta útil para a resolução de dúvidas.

6.2. Discussão

Os resultados obtidos por meio da avaliação inicial sugerem que o sistema atende adequadamente aos seus objetivos principais: agilizar o acesso às informações do curso e reduzir a dependência direta de professores ou coordenadores para dúvidas recorrentes.

A percepção positiva quanto à precisão das respostas valida a eficácia da abordagem baseada na técnica de Recuperação Aumentada por Geração (RAG) com uso de embeddings vetoriais. A combinação do modelo *BAAI/bge-m3* para embeddings e do modelo *GPT-4o da OpenAI* para geração de respostas demonstrou ser adequada para o domínio de informações acadêmicas específicas.

No entanto, por se tratar de uma avaliação com um número reduzido de participantes, entende-se que novas etapas de validação com uma amostra maior poderão oferecer percepções mais profundas sobre limitações, casos de uso extremos e eventuais ajustes de parâmetros, documentos e interface.

7. Considerações finais

A aplicação demonstrou-se funcional e eficaz, fornecendo respostas contextualizadas com base nas informações extraídas dos documentos institucionais. Os testes realizados com alunos do curso evidenciaram que o sistema é capaz de entregar respostas corretas, com boa usabilidade e com potencial de adoção por parte da comunidade acadêmica. A interface desenvolvida com o framework Streamlit proporcionou uma experiência prática, intuitiva e acessível, permitindo uma interação fluida entre usuário e sistema.

Os resultados indicam que o chatbot possui grande potencial como ferramenta de apoio acadêmico, contribuindo para a autonomia dos estudantes na busca por informações e para a redução da sobrecarga de atendimentos por parte de professores e coordenadores. A abordagem adotada permite, ainda, a fácil atualização da base de conhecimento, tornando o sistema escalável para outros cursos ou instituições.

Como continuidade deste trabalho, sugere-se a ampliação da base de documentos, a realização de testes com um número maior de alunos e a inclusão de novos recursos, como categorização de dúvidas e suporte a múltiplos cursos. Essas melhorias poderão refinar ainda mais a experiência do usuário e aumentar a qualidade das respostas geradas.

Dessa forma, conclui-se que o uso de tecnologias de inteligência artificial generativa, como os modelos de linguagem combinados com técnicas de busca semântica, representa uma solução viável e eficiente para otimizar o acesso às informações no ambiente acadêmico.

Referências

ALURA. *O que é RAG e como funciona a Recuperação Aumentada por Geração*. 2024. Acesso em: 03 maio 2025. Disponível em: ⟨https://www.alura.com.br/artigos/o-que-e-rag⟩.

Amazon Web Services. *O que é chatbot?* 2024. Acesso em: 2 nov. 2024. Disponível em: ⟨https://aws.amazon.com/pt/what-is/chatbot/⟩.

Amazon Web Services. *O que é LangChain?* 2024. Acesso em: 2 nov. 2024. Disponível em: ⟨https://aws.amazon.com/pt/what-is/langchain/⟩.

Asimov Academy. *Streamlit: Guia completo para iniciantes e profissionais*. 2023. Acesso em: 9 maio 2025. Disponível em: ⟨https://hub.asimov.academy/blog/streamlit-guia-completo/⟩.

BULHOES, D. B. et al. Professora vitória: um chatbot para o ensino da leitura. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 2020. p. 451–460. Acesso em: 7 dez. 2024. Disponível em: ⟨https://sol.sbc.org.br/index.php/sbie/article/view/12801/12655⟩.

CARVALHO, C. *Python: o que é, como aprender e quais são as áreas de atuação*. 2024. ⟨https://www.alura.com.br/artigos/python⟩. Acesso em: 4 nov. 2024.

Data Science Academy. *Os avanços dos LLMs e suas aplicações*. 2024. Acesso em: 21 nov. 2024. Disponível em: ⟨https://www.datascienceacademy.com.br⟩.

Facebook AI Research. *Faiss: A library for efficient similarity search and clustering of dense vectors*. 2025. Acesso em: 9 maio 2025. Disponível em: ⟨https://faiss.ai/⟩.

Google Cloud. *O que é o processamento de linguagem natural?* 2024. Acesso em: 4 nov. 2024. Disponível em: ⟨https://cloud.google.com/learn/what-is-natural-language-processing?hl=pt-BR⟩.

IBM. *O que é o GPT-4o?* 2024. ⟨https://www.ibm.com/br-pt/think/topics/gpt-4o⟩. Acesso em: 24 maio 2025.

LARA, D. F. et al. Atendente artificial humanizada usando langchain para manipulação de modelos de linguagem em larga escala. In: *Anais do VI Simpósio de Iniciação Científica da UNIFEI*. [s.n.], 2023. Acesso em: 7 dez. 2024. Disponível em: ⟨https://periodicos.unifei.edu.br/index.php/rtic/article/view/590⟩.

NEELAKANDAN, L. *O que são grandes modelos de linguagem (LLMs) e para que servem*. 2023. Acesso em: 03 maio 2025. Disponível em: ⟨https://fastcompanybrasil.com/tech/o-que-sao-grandes-modelos-de-linguagem-llms-e-para-que-servem/⟩.

NEVES, C. M. F. R. *CampusHubAI: Uma integração inovadora do aluno à universidade com GenAI*. Monografia (Bacharelado em Ciências e Tecnologia) — Universidade Federal do Rio Grande do Norte, Escola de Ciências e Tecnologia, Natal, 2024. Acesso em: 7 dez. 2024. Disponível em: ⟨https://repositorio.ufrn.br/handle/123456789/59957⟩.

pypdf. *pypdf Documentation*. 2025. Acesso em: 9 maio 2025. Disponível em: ⟨https://pypdf.readthedocs.io/en/stable/⟩.

REBELO, M. *What is Hugging Face?* 2025. Acesso em: 9 maio 2025. Disponível em: ⟨https://zapier.com/blog/hugging-face/⟩.

SANTOS, V. et al. Alfa - um chatbot do tipo perguntas e respostas como assistente virtual no ava moodle. In: *Simpósio Brasileiro de Informática na Educação (SBIE), 2023, Online*. Porto Alegre: SBC, 2023. Acesso em: 7 dez. 2024. Disponível em: ⟨https://sol.sbc.org.br/index.php/sbie/article/view/26760⟩.

WICKS, G. H. *Chatbot-Poli: uma proposta de MVP para automatização de respostas para dúvidas acadêmicas com OpenAI*. Trabalho de Conclusão de Curso (Engenharia

de Controle e Automação) — Universidade Federal da Bahia, Escola Politécnica, Departamento de Engenharia Química, 2023. Acesso em: 2 nov. 2024. Disponível em: ⟨https://repositorio.ufba.br/bitstream/ri/38954/1/TCC_versao_final.pdf⟩.