



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
RONDÔNIA CAMPUS ARIQUEMES
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUCAS DE OLIVEIRA

ASSISTENTE VIRTUAL INTELIGENTE: OTIMIZANDO O ACESSO À
INFORMAÇÃO PARA ESTUDANTES DE ADS DO IFRO - CAMPUS ARIQUEMES

Ariquemes
2025

LUCAS DE OLIVEIRA

ASSISTENTE VIRTUAL INTELIGENTE: OTIMIZANDO O ACESSO À
INFORMAÇÃO PARA ESTUDANTES DE ADS DO IFRO - CAMPUS ARIQUEMES

Relatório técnico entregue como Trabalho Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – (IFRO), Campus Ariquemes, como requisito parcial para obtenção do grau de tecnólogo junto ao curso de Análise e Desenvolvimento de Sistemas.

Orientador: Andrey Alencar Quadros

Ariquemes

2025

LUCAS DE OLIVEIRA

ASSISTENTE VIRTUAL INTELIGENTE: OTIMIZANDO O ACESSO À
INFORMAÇÃO PARA ESTUDANTES DE ADS DO IFRO - CAMPUS ARIQUEMES

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de
“Analista e Desenvolvedor de Sistemas” e aprovado em sua forma final pelo Curso de
Tecnologia em Análise e Desenvolvimento de Sistemas.

Ariquemes, 03 de fevereiro de 2025.

Banca Examinadora:

Prof. Mestre Andrey Alencar Quadros

(orientador)

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO - Campus
Ariquemes)

Prof. Mestre Luciano Topolniak

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO - Campus
Ariquemes)

Prof. Especialista Marcos Alves Faino

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO - Campus
Ariquemes)

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO,
com dados informados pelo(a) próprio(a) autor(a).

Oliveira, Lucas de.

Assistente virtual inteligente: otimizando o acesso à informação para
estudantes de ADS do IFRO - Campus Ariquemes / Lucas de Oliveira,
Ariquemes-RO, 2025.

35 f. : il.

Orientador(a): Prof. Me. Andrey Alencar Quadros.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e
Tecnologia de Rondônia - IFRO, Ariquemes-RO, 2025.

1. Inteligência artificial. 2. Processamento de linguagem natural. 3.
Assistente Virtual. 4. Ensino superior. I. Quadros, Andrey Alencar (orient.).
II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO.
III. Título.

Bibliotecário(a) Responsável: Renilce Silva Moraes, CRB-11/906 (Campus Ariquemes)

AGRADECIMENTOS

Inicialmente, agradeço à minha família pelo apoio incondicional ao longo desses anos e pelo incentivo, que foi uma parte essencial da minha motivação para chegar até este momento. Sou grato também por me estimularem a dedicar-me aos estudos, pois, sem eles, eu não teria chegado a lugar nenhum.

Agradeço igualmente ao meu orientador, Me. Andrey Alencar Quadros, pela orientação e pelo incentivo para que eu concluísse o curso até o último minuto. Minha gratidão se estende também aos professores das demais disciplinas pelo conhecimento transmitido, pelo tempo dedicado ao ensino e pela educação de qualidade que me proporcionaram.

Por fim, expresso meu agradecimento ao Instituto por oferecer oportunidades que permitem aos alunos o acesso a professores de altíssimo nível no que diz respeito ao conhecimento.

RESUMO

O avanço da IA (Inteligência Artificial) e do PLN (Processamento de Linguagem Natural) tem viabilizado o desenvolvimento de assistentes virtuais que aprimoram o acesso à informação em diversos contextos. No ambiente acadêmico, a dificuldade dos estudantes em obter respostas rápidas e precisas sobre regulamentos, prazos e procedimentos institucionais evidencia a necessidade de soluções inovadoras. Este trabalho propõe a criação de um assistente virtual utilizando a API da OpenAI, integrado a um sistema de busca em arquivos institucionais, para responder automaticamente às perguntas mais frequentes dos alunos do curso de Análise e Desenvolvimento de Sistemas. A aplicação foi desenvolvida com Streamlit para a interface gráfica, utilizando a plataforma Streamlit Community Cloud para hospedagem. Foram realizados testes com 100 perguntas baseadas nos documentos institucionais, analisando a precisão e eficiência das respostas. Os resultados demonstraram uma taxa de resposta de 92%, ou seja, consegue responder em 92% dos casos, e consegue responder corretamente em 84,78% dos casos com respostas. Conclui-se que a ferramenta pode otimizar a comunicação acadêmica, reduzir a sobrecarga da coordenação do curso e proporcionar maior autonomia aos estudantes na obtenção de informações institucionais.

Palavras-chave: Inteligência Artificial; Processamento de Linguagem Natural; Assistente Virtual; Ensino Superior.

ABSTRACT

The advancement of AI (Artificial Intelligence) and NLP (Natural Language Processing) has enabled the development of virtual assistants that enhance information access in various contexts. In the academic environment, students often struggle to obtain quick and accurate answers regarding regulations, deadlines, and institutional procedures, highlighting the need for innovative solutions. This study proposes the creation of a virtual assistant using OpenAI's API, integrated with a search system for institutional files, to automatically answer frequently asked questions from students in the Systems Analysis and Development course. The application was developed using Streamlit for the graphical interface and hosted on the Streamlit Community Cloud platform. Tests were conducted with 100 questions based on institutional documents, analyzing the accuracy and efficiency of the responses. The results showed a response rate of 92%, meaning it can provide answers in 92% of cases, with an accuracy of 84.78% for the given responses. It is concluded that the tool can optimize academic communication, reduce the workload of the course coordination, and provide students with greater autonomy in obtaining institutional information.

Keywords: Artificial Intelligence; Natural Language Processing; Virtual Assistant; Higher Education.

LISTA DE FIGURAS

Figura 1 – Resposta à questão da quantidade de faltas.	27
Figura 2 – Resposta à questão das disciplinas do semestre.	27
Figura 3 – Assistente respondendo a mesma questão escrita de duas formas.	29

LISTA DE ABREVIATURAS E SIGLAS

ADS	Análise e Desenvolvimento de Sistemas
AI	Artificial Intelligence
API	Application Programming Interface
AVA	Ambiente Virtual de Aprendizagem
CPU	Central Processing Unit
IA	Inteligência Artificial
IFRO	Instituto Federal de Rondônia
NLP	Natural Language Processing
PLN	Processamento de Linguagem Natural
TIC	Tecnologia de Informação e Comunicação

SUMÁRIO

1	INTRODUÇÃO	11
2	OBJETIVOS	12
2.1	OBJETIVO GERAL	12
2.2	OBJETIVOS ESPECÍFICOS	12
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	INTELIGÊNCIA ARTIFICIAL E PROCESSAMENTO DE LINGUAGEM NATURAL	13
3.2	SISTEMAS DE ATENDIMENTO AUTOMATIZADOS	13
3.3	CHATGPT E FILE SEARCH	14
3.4	O USO DE IA NA EDUCAÇÃO	14
4	ACESSO À INFORMAÇÃO NO ENSINO SUPERIOR	16
4.1	O DIREITO DE ACESSO À INFORMAÇÃO NA CONSTITUIÇÃO FEDERAL	16
4.2	BARREIRAS NA COMUNICAÇÃO ACADÊMICA	16
4.3	SOLUÇÕES TECNOLÓGICAS PARA DEMOCRATIZAÇÃO DA INFORMAÇÃO	16
5	DESENVOLVIMENTO DO ASSISTENTE VIRTUAL	18
5.1	TECNOLOGIAS UTILIZADAS	18
5.1.1	Streamlit para Interface Gráfica	18
5.1.2	Python para Manipulação de Threads	18
5.1.3	API da OpenAI	19
5.2	IMPLEMENTAÇÃO DA SOLUÇÃO	19
5.2.1	Requisitos do Sistema	19
5.2.1.1	<i>Download e Instalação do Python</i>	20
5.3	HOSPEDAGEM NO STREAMLIT COMMUNITY CLOUD	21
5.3.1	Passo a Passo para Hospedagem no Streamlit Community Cloud	21
5.3.2	Configuração da API da OpenAI com Assistente e Contexto	22
5.3.2.1	<i>Tratamento das Threads</i>	23
5.3.2.2	<i>Contexto Utilizado</i>	25
6	FUNCIONALIDADES DA APLICAÇÃO E RESULTADOS	26
6.1	METODOLOGIA DOS TESTES	26
6.2	AVALIAÇÃO DE EFICIÊNCIA DO ASSISTENTE	27
6.2.1	Cálculos de Eficiência	28
6.2.1.1	<i>Respostas não retornadas</i>	28
6.2.2	Interpretação dos Resultados	28
7	CONCLUSÃO	30
7.1	CONTRIBUIÇÕES DO TRABALHO	30

7.2	LIMITAÇÕES ENCONTRADAS	30
7.3	SUGESTÕES PARA MELHORIAS FUTURAS	31
7.4	CONTINUIDADE E EXPANSÃO DO PROJETO	31
7.4.1	Documentação Técnica e Organizacional	32
7.5	CONSIDERAÇÕES FINAIS	32
	REFERÊNCIAS	33

1 INTRODUÇÃO

O acesso à informação acadêmica é um fator determinante para o sucesso dos estudantes no ensino superior. Entretanto, muitos alunos enfrentam dificuldades para obter respostas rápidas e precisas sobre normas, prazos e regulamentos institucionais. A fragmentação das informações em diferentes plataformas e documentos institucionais, aliada à alta demanda de atendimento por parte das coordenações de curso, resulta em processos burocráticos que comprometem a eficiência da comunicação acadêmica. Esse cenário pode gerar dúvidas recorrentes, atrasos no cumprimento de exigências acadêmicas e um impacto negativo na experiência estudantil.

Nos últimos anos, a IA (Inteligência Artificial) tem desempenhado um papel fundamental na transformação digital de diversos setores, incluindo a educação. Soluções baseadas em IA vêm sendo utilizadas para aprimorar o acesso à informação, automatizar processos administrativos e melhorar a interação entre estudantes e instituições de ensino. Dentre essas soluções, destacam-se os assistentes virtuais inteligentes, que utilizam técnicas de PLN (Processamento de Linguagem Natural) para interpretar perguntas e fornecer respostas contextuais de forma automatizada.

Neste contexto, este estudo propõe o desenvolvimento de um **assistente virtual** baseado na API da OpenAI, integrado a um sistema de pesquisa em arquivos institucionais. O objetivo é fornecer respostas precisas e instantâneas às dúvidas acadêmicas mais frequentes dos alunos do curso de ADS (Análise e Desenvolvimento de Sistemas). A ferramenta foi desenvolvida utilizando a plataforma *Streamlit* para a construção da interface gráfica e hospedada na *Streamlit Community Cloud*, garantindo acessibilidade sem a necessidade de instalação local. O assistente processa e analisa documentos oficiais do curso, aplicando técnicas de PLN (Processamento de Linguagem Natural) para interpretar e responder às perguntas dos usuários de maneira eficiente e contextualizada.

A implementação deste assistente virtual tem o potencial de otimizar significativamente a comunicação acadêmica, reduzindo a sobrecarga da coordenação do curso e proporcionando maior autonomia aos estudantes na busca por informações institucionais. Além disso, ao consolidar dados essenciais em um único ambiente digital, a ferramenta contribui para a padronização das respostas e para a democratização do acesso às informações acadêmicas.

Dessa forma, este estudo busca avaliar a eficiência do assistente virtual, analisando sua capacidade de fornecer respostas corretas e úteis aos estudantes. Também discute os impactos da automação na comunicação acadêmica, bem como os desafios e limitações envolvidos na implementação de assistentes virtuais em contextos educativos. Os resultados obtidos indicam que a solução desenvolvida pode representar um avanço significativo na modernização do ensino superior, promovendo maior transparência, acessibilidade e eficiência no fornecimento de informações institucionais.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver um assistente virtual com a API da OpenAI para responder automaticamente às dúvidas mais frequentes dos estudantes de Análise e Desenvolvimento de Sistemas, facilitando o acesso a informações oficiais do curso de forma ágil e eficiente.

2.2 OBJETIVOS ESPECÍFICOS

1. Desenvolver uma interface intuitiva com Streamlit para facilitar a interação dos estudantes com o assistente virtual.
2. Integrar a API do OpenAI com um sistema de busca em documentos institucionais, garantindo respostas precisas e baseadas em dados oficiais do curso.
3. Implementar o assistente na Streamlit Community Cloud, assegurando disponibilidade e acesso remoto sem necessidade de instalação local.
4. Automatizar a gestão de interações e manter o acervo atualizado, reduzindo a sobrecarga da coordenação e garantindo informações sempre relevantes.

Espera-se que o assistente virtual minimize as dúvidas recorrentes à coordenação, oferecendo respostas rápidas e precisas, além de otimizar o tempo dos gestores acadêmicos ao facilitar o acesso eficiente às informações do curso.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos de Inteligência Artificial, destacando os sistemas de atendimento automatizados e as ferramentas que facilitam o acesso à informações na educação.

3.1 INTELIGÊNCIA ARTIFICIAL E PROCESSAMENTO DE LINGUAGEM NATURAL

“A IA (Inteligência Artificial) é uma área da computação voltada a desenvolver algoritmos e sistemas capazes de realizar tarefas que demandam habilidades associadas à inteligência humana” (GARCIA, 2020, p. 15). (IBM, s.d.) define inteligência artificial como “uma tecnologia que permite que computadores e máquinas simulem a capacidade de resolução de problemas e a inteligência humana”. Ademais, sobre IA:

A inteligência artificial é um conjunto de tecnologias que permitem aos computadores executar uma variedade de funções avançadas, incluindo a capacidade de ver, entender e traduzir idiomas falados e escritos, analisar dados, fazer recomendações e muito mais. (CLOUD, s.d.).

A IA é uma tecnologia que simula a inteligência humana, possibilitando que máquinas realizem tarefas complexas de maneira independente. A sua utilização vai do processamento de linguagem à análise de dados, permitindo progressos notáveis na automação e na tomada de decisões. Portanto, a Inteligência Artificial torna-se um componente crucial para aprimorar processos e fomentar inovações em várias áreas.

Já o PLN que é uma espécie de *machine learning* que é responsável pelo treinamento e melhorias de tecnologias de Inteligência Artificial fazendo com que o algoritmo se familiarize com o contexto da situação, deixando-o apto para atingir o objetivo específico para o qual o algoritmo foi treinado (TRINDADE; OLIVEIRA, 2024). Ainda sobre os tipos de aprendizado de máquina:

Dentre os tipos de aprendizado de máquina apresentados, destaca-se o aprendizado profundo, pois está associado ao reconhecimento de padrões e contribuiu significativamente para o avanço da visão computacional, do Processamento de Linguagem Natural (PLN) e dos sistemas de recomendação (TRINDADE; OLIVEIRA, 2024).

Resumidamente PLN (Processamento de Linguagem Natural) “é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana” (CASELI; NUNES, 2024).

3.2 SISTEMAS DE ATENDIMENTO AUTOMATIZADOS

Segundo (BALDIN *et al.*, s.d.), “A automatização de processos trata-se de transformar em processos computacionais, as atividades dependentes de intervenção exclusivamente humana”.

Corroborando com isso Baldin, Schambeck, Matos e Crescencio (BALDIN *et al.*, s.d.) deixam claro que uma das grandes vantagens da automatização de processos são a conformidade e agilidade na execução do processo, uma menor possibilidade de erros, custos mais baixos e alguns outros.

3.3 CHATGPT E FILE SEARCH

Segundo (CASELI; NUNES, 2024) o ChatGPT é um dos “exemplos de aplicações de agentes de conversação (*chatbots*) baseados em modelos de linguagem gerativos (ou generativos)”.

Alguns autores usam o termo “agente de conversação” para denominar um sistema de conversação que se comuniquem com os seus utilizadores usando a linguagem humana e os dividem em duas classes: agentes orientados a tarefas, que possui a finalidade de resolver problemas ou tarefas que lhe foram passadas através do diálogo, como calcular o custo de um financiamento veicular, e a outra categoria é a de *chatbots* que são agentes que possuem a finalidade de conversar com os humanos e tentar simular os seus diálogos (CASELI; NUNES, 2024).

File Search ou Buscador de Arquivos é uma ferramenta de IA (Inteligência Artificial) que, segundo (BERNO; PEIXE; BALSAN, s.d.), oferece soluções para captura, classificação, recuperação e tratamento de dados.

Segundo (BERNO; PEIXE; BALSAN, s.d.), “Os sistemas de Inteligência Artificial têm a capacidade de extrair informações essenciais de documentos, reconhecer padrões, identificar anomalias e até mesmo antecipar tendências futuras por meio da análise de dados”.

Os sistemas de conversação são ferramentas criadas para se comunicar com os usuários por meio de linguagem natural. Eles podem ser divididos em agentes focados em tarefas, que ajudam na solução de problemas específicos, e *chatbots*, que tentam reproduzir conversas humanas. Neste cenário, instrumentos como o *File Search* são fundamentais, pois permitem a coleta, categorização e recuperação de dados, auxiliando na eficácia da análise de informações. Ademais, os sistemas de Inteligência Artificial têm a habilidade de reconhecer padrões, identificar anomalias e prever tendências, tornando-se essenciais para a melhoria de processos que utilizam grandes quantidades de dados.

3.4 O USO DE IA NA EDUCAÇÃO

Segundo (GONÇALEZ; MOREIRA, R. S.; TORRES, s.d.) a evolução dos *chatbots* está tão avançada que está sendo possível utilizá-los na educação.

Além disso, (GANDRA, 2023) diz que a IA

Para os professores, facilita no ensino das disciplinas e eles podem se concentrar mais na parte crítica, desenvolvendo o conhecimento crítico dos alunos. A nova

tecnologia pode ajudar a promover a criatividade e curiosidade e a fornecer conteúdos personalizados aos alunos, entre outras estratégias, com o objetivo de reduzir o abandono escolar.

A Inteligência Artificial na educação possibilita aos alunos o desenvolvimento do seu pensamento crítico de forma a fazê-los criar uma curiosidade maior sobre os assuntos disponibilizados dentro do ambiente escolar.

4 ACESSO À INFORMAÇÃO NO ENSINO SUPERIOR

4.1 O DIREITO DE ACESSO À INFORMAÇÃO NA CONSTITUIÇÃO FEDERAL

A Constituição Federal do Brasil no Art. 5º, inciso XXXIII, diz o seguinte sobre o acesso a informação:

XXXIII - todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidade, ressalvadas aquelas cujo sigilo seja imprescindível à segurança da sociedade e do Estado (BRASIL, 1988).

Diante disso, torna-se clara a necessidade de deixar a informação disponível e acessível à todos.

4.2 BARREIRAS NA COMUNICAÇÃO ACADÊMICA

A comunicação em organizações costumam não ser tão eficientes quanto se imagina e nem sempre é tratada da maneira que realmente deveria, isso acontece devido a inúmeros acontecimentos inesperados que prejudicam ou sujam o trajeto entre o emissor e o receptor (GELLERTH, s.d.).

Normalmente, a recepção, a interpretação, a codificação e decodificação da mensagem, devem ser levados em consideração de maneira que incentivem os agentes da mensagem a pensarem e analisarem sobre a oportunidade de manterem um controle mais assíduo sobre a qualidade da mensagem de maneira que ela não vá se atenuando ou se alterando a partir do momento em que passar por vários comunicadores (GELLERTH, s.d.).

Na academia, cada pessoa, discente ou docente, acadêmico ou não, possui uma cultura única e que são divergentes por virem de locais diferentes com ensinamentos e conceitos diferentes, o que, por sua vez, dificulta ainda mais o processo de comunicação aumentando assim as barreiras ou problemas de comunicação (GELLERTH, s.d.)

Portanto, os discentes, como integrantes da sociedade acadêmica, acabam sofrendo com as dificuldades e barreiras encontradas na obtenção de informações.

4.3 SOLUÇÕES TECNOLÓGICAS PARA DEMOCRATIZAÇÃO DA INFORMAÇÃO

Segundo (MOREIRA, P. A.; RANZATTI; ROSINI, 2020), “As soluções de TIC (Tecnologia de Informação e Comunicação) estão cada vez mais presentes nas relações, sejam pessoais, comerciais ou sociais”. Corroborando com isso (MOREIRA, P. A.; RANZATTI; ROSINI, 2020) diz que devido a mudança e evolução dos meios de comunicação com foco, principalmente, na Internet, houve uma mudança significativa entre a relação Estado e Sociedade.

Diante desse contexto, torna-se evidente a necessidade de adotar soluções tecnológicas que facilitem e democratizem o acesso à informação no ambiente acadêmico. Segundo (FUNDAÇÃO GETULIO VARGAS, 2024), a Inteligência Artificial tem o potencial de revolucionar a comunicação interna e externa das instituições, promovendo maior eficiência na disseminação do conhecimento e na resolução de demandas administrativas.

Com o avanço dos *chatbots*, os alunos passaram a contar com um meio interativo e acessível para obter informações relevantes de maneira autônoma. Essa tecnologia não apenas agiliza o atendimento, reduzindo a sobrecarga da coordenação e do corpo docente, mas também possibilita uma ampliação significativa do campo de visão dos estudantes, permitindo o acesso a conteúdos que antes recebiam pouca atenção ou tinham menor visibilidade. Dessa forma, os assistentes virtuais baseados em IA tornam-se ferramentas estratégicas para o ensino superior, contribuindo para a modernização da comunicação acadêmica e a melhoria da experiência educacional.

5 DESENVOLVIMENTO DO ASSISTENTE VIRTUAL

5.1 TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas fazem referência à arquitetura do sistema. Conforme (TRT9, s.d.) podemos dizer que a arquitetura do sistema representa as estruturas nas quais o sistema está embasado, consistindo em componentes de software, propriedades externamente visíveis do componentes e os relacionamentos entre eles.

5.1.1 Streamlit para Interface Gráfica

Segundo (SOARES *et al.*, 2024), “*Streamlit* é uma ferramenta de código aberto para criação de aplicativos web interativos em linguagem *Python* e amplamente utilizada em ciência de dados”. Com ele podem ser criadas interfaces interativas e atrativas sem a necessidade de um grande conhecimento em desenvolvimento web, e sua escolha se dá por ser extremamente fácil e adaptável. Ele também fornece uma grande variedade de componentes e *widgets* previamente criados que rapidamente podem ser personalizados para o projeto (SOARES *et al.*, 2024).

5.1.2 Python para Manipulação de Threads

De acordo com (SOARES *et al.*, 2024), “Python é uma linguagem de programação popular e amplamente utilizada em pesquisa científica e educação”. Como a linguagem Python é uma linguagem extremamente flexível, possuir certa facilidade e com uma grande compatibilidade com várias bibliotecas acabam tornando-a a escolha essencial para o projeto (SOARES *et al.*, 2024).

Já (SOUTO, 2024) diz que *Thread* “refere-se a um fluxo sequencial de execução de instruções, seja de código das aplicações ou de sistema operacional. Um processo pode ter múltiplas threads, cada uma representando uma sequência independente de execução”. Uma *thread* pode estar em um dos seus vários estados, que são: em execução, bloqueado, pronto, ou concluído. Para efeito de comparação, uma *thread* que esteja com estado em execução está utilizando a *CPU (Central Processing Unit)*, em português, Unidade Central de Processamento, e está ativo. Já um *thread* com estado bloqueado está esperando algum evento que o desbloqueie, um *thread* pronto está programado para ser executado assim que chegar a sua vez (SOUTO, 2024).

O uso de Python para gerenciar threads possibilita aprimorar a resposta às solicitações, assegurando que a aplicação permaneça responsiva enquanto aguarda respostas da API (Application Programming Interface). Como cada *thread* representa um fluxo de execução separado, é possível administrar várias requisições ao mesmo tempo, prevenindo bloqueios no sistema. Este processo é crucial para aprimorar a performance do assistente virtual, garantindo uma interação suave e eficaz com os utilizadores.

5.1.3 API da OpenAI

Segundo (ARAÚJO, 2021), “Uma API é um conjunto de código de programação que possibilita a transmissão de dados entre diferentes softwares, além de integrar serviços, empresas, negócios e parceiros”. Com base em (ARAÚJO, 2021), podemos dizer que uma API bem construída depende de uma documentação clara, que descreva seus *endpoints*, parâmetros e respostas. Isso facilita a implementação, reduz erros e melhora a experiência dos desenvolvedores, garantindo uma integração eficiente entre sistemas.

A API da OpenAI permitirá retornar uma conclusão de texto, com base no contexto, que é o conceito ambiental dos exemplos de respostas, para a qual ela foi programada. O sucesso da resposta depende da complexidade da pergunta ou da tarefa que foi dada a ela. Caso a API seja treinada com reforços positivos para tarefas que ela fizer corretamente ou com reforços negativos para o caso contrário, serão retornadas respostas mais coerentes com o tema e com as definições reforçadas positivamente (OPENAI, 2020).

5.2 IMPLEMENTAÇÃO DA SOLUÇÃO

5.2.1 Requisitos do Sistema

Os requisitos mínimos para rodar o *Python* com o *Streamlit* incluem:

- Sistema Operacional: *Linux*, *Windows* ou *MacOS*;
- Python: Versão 3.7 ou superior;
- Navegador: Qualquer navegador moderno, como *Google Chrome*, *Mozilla Firefox*, *Microsoft Edge* ou *Safari*;
- Pacotes Necessários:
 - `streamlit==1.34.0`: Framework para criação da interface gráfica interativa.
 - `openai==1.28.0`: SDK para integração com a API do ChatGPT.
 - `pandas==2.2.2`: Manipulação e análise de dados tabulares.
 - `numpy==1.26.4`: Biblioteca para operações matemáticas e manipulação de arrays multidimensionais.
 - `python-dateutil==2.9.0.post0`: Manipulação avançada de datas e horários.
 - `pytz==2024.1`: Suporte a fusos horários em Python.
 - `pyarrow==16.0.0`: Processamento de dados em memória e suporte a formatos como Parquet.
 - `watchdog==4.0.0`: Monitoramento de alterações em arquivos e diretórios.

- `rich==13.7.1`: Biblioteca para formatação e exibição avançada de texto no terminal.
 - `tqdm==4.66.4`: Exibição de barras de progresso para operações demoradas.
 - `tenacity==8.3.0`: Gerenciamento de tentativas automáticas para chamadas à API.
 - `anyio==4.3.0`: Gerenciamento de entrada e saída assíncrona.
 - `requests==2.31.0`: Biblioteca HTTP simples e eficiente para chamadas de API.
 - `httpx==0.27.0`: Cliente HTTP assíncrono com suporte a HTTP/2.
 - `cryptography==41.0.4`: Biblioteca para segurança e criptografia de dados.
 - `attrs==23.2.0`: Facilita a criação de classes baseadas em atributos.
 - `toolz==0.12.1`: Conjunto de ferramentas para manipulação funcional de dados.
 - `pydantic==2.7.1`: Validação de dados e criação de modelos com tipagem forte.
 - `jsonschema==4.22.0`: Validação de dados no formato JSON.
 - `platformdirs==3.11.0`: Determina diretórios específicos de configuração e cache.
 - `virtualenv==20.26.1`: Criação de ambientes virtuais isolados para Python.
- Conexão com a Internet: Necessária para instalação de pacotes e integração com APIs externas.

5.2.1.1 Download e Instalação do Python

O procedimento para realizar o download e instalação do *Python* é descrito a seguir:

1. Acesse o site oficial: Abra o navegador de sua preferência e acesse o site oficial do Python: <https://www.python.org>.
2. Faça o download do instalador:
 - a) Clique na aba *Downloads* e selecione a versão recomendada para o seu sistema operacional (*Windows*, *macOS* ou *Linux*).
 - b) Certifique-se de baixar a versão 3.7 ou superior, conforme a necessidade do projeto.
3. Execute o instalador:
 - a) Após o download, localize o arquivo instalador e execute-o.

- b) Durante a instalação, marque a opção *Add Python to PATH* para facilitar o uso do Python no terminal.
4. Siga as instruções da instalação:
 - a) Clique em *Customize installation* se desejar configurar componentes adicionais, ou *Install Now* para uma instalação padrão.
 - b) Aguarde a conclusão do processo de instalação.
5. Verifique a instalação:
 - a) Abra o terminal ou prompt de comando e digite o seguinte comando para verificar se o Python foi instalado corretamente:

```
python --version
```
 - b) O terminal exibirá a versão instalada do Python, confirmando que a instalação foi bem-sucedida.
6. Instale o *pip* (caso necessário):
 - a) O *pip*, gerenciador de pacotes do Python, geralmente é instalado automaticamente. Para verificar, execute o comando:

```
pip --version
```
 - b) Se não estiver instalado, siga as instruções no site oficial do Python para configurar o *pip*.

5.3 HOSPEDAGEM NO STREAMLIT COMMUNITY CLOUD

Utilizou-se a plataforma *Streamlit Community Cloud* para a realização da hospedagem do sistema.

5.3.1 Passo a Passo para Hospedagem no Streamlit Community Cloud

Para realizar a hospedagem do projeto na plataforma *Streamlit Community Cloud*, devem ser seguidos os seguintes passos:

1. **Acessar a plataforma e realizar o login**
 - a) Abra o navegador de sua preferência.
 - b) Acesse o site oficial: <https://streamlit.io/cloud>.
 - c) Clique na opção *Sign in* e faça login utilizando uma conta *GitHub*, *Google* ou *email*.
2. **Criar um novo aplicativo**

- a) No canto superior direito da tela, clique no botão *Create App*.
- b) Escolha a origem do repositório onde o código do projeto está armazenado.

3. Configurar as chaves da API

- a) No painel de configurações do *Streamlit*, vá até a seção de variáveis de ambiente.
- b) Insira a chave da API do ChatGPT no campo correspondente (*OPENAIAPIKEY*).
- c) Salve as configurações para garantir a autenticação correta da API.

4. Publicar o projeto

- a) Revise todas as configurações da aplicação.
- b) Clique em *Deploy* para finalizar a publicação do projeto.
- c) Aguarde a finalização do processo e verifique se a aplicação foi implantada corretamente.

5.3.2 Configuração da API da OpenAI com Assistente e Contexto

Para configurar a API da OpenAI utilizando um assistente e definir um contexto adequado, devem ser seguidos os seguintes passos:

1. Acessar a plataforma da OpenAI e realizar o login

- a) Abra o navegador de sua preferência.
- b) Acesse o site oficial: <https://platform.openai.com>.
- c) Clique em *Sign in* e faça login utilizando uma conta previamente cadastrada.

2. Criar um novo assistente

- a) No menu lateral esquerdo, clique em *Assistants*.
- b) Selecione a opção *Create Assistant*.
- c) Defina um nome para o assistente e escolha o modelo de IA apropriado, como `gpt-3.5-turbo-0125`.

3. Definir o contexto do assistente

- a) No campo de descrição, insira um contexto detalhado sobre a função do assistente.
- b) Exemplo: *"Este assistente responde dúvidas acadêmicas de estudantes do curso de Análise e Desenvolvimento de Sistemas com base nos documentos institucionais."*
- c) Salve as configurações para aplicar o contexto.

4. Configurar arquivos de referência (*File Search*)

- a) Na aba *Knowledge* do assistente, clique em *Upload Files*.
- b) Faça o upload dos documentos institucionais relevantes (ex.: regulamentos, ementas de disciplinas).
- c) Confirme que os arquivos foram indexados corretamente para permitir buscas eficientes.

5. Gerar e copiar a chave da API

- a) No menu lateral, clique em *API Keys*.
- b) Selecione a opção *Create API Key*.
- c) Copie a chave gerada e guarde-a em local seguro, pois ela não será exibida novamente.

6. Configurar permissões e salvar

- a) No menu de configurações do assistente, revise as permissões de acesso.
- b) Certifique-se de que a API está habilitada para integração com aplicações externas.
- c) Clique em *Save* para finalizar a configuração.

5.3.2.1 Tratamento das Threads

Segue abaixo o código utilizado para tratamento das threads e utilização do *File Search* para busca no arquivo contexto:

```
import streamlit as st
import logging
from openai import OpenAI

client = OpenAI(api_key=st.secrets['OPENAI_API_KEY'])

class GetAnswer:
    def __init__(self):
        pass

    @staticmethod
    def create_thread(message_file_id, question):
        try:
            thread = client.beta.threads.create(
                messages=[
                    {
                        "role": "user",
                        "content": question,
```

```
        "attachments": [
            {"file_id": message_file_id, "tools":
             [{"type": "file_search"}]}
        ],
    }
]
)
logging.info(f"Thread created with ID: {thread.id}")
return thread
except Exception as e:
    logging.error(f"Erro ao criar thread: {e}")
    return None

@staticmethod
def get_answer_with_assistant(thread_id, assistant_id):
    if not thread_id:
        logging.error("Invalid thread ID.")
        return None

    try:
        run = client.beta.threads.runs.create_and_poll(
            thread_id=thread_id, assistant_id=assistant_id
        )

        messages =
        list(
            client.beta.threads.messages.list(
                thread_id=thread_id, run_id=run.id
            )
        )

        if not messages:
            logging.error("No messages found in the thread.")
            return None

        message_content = messages[0].content[0].text
        annotations = getattr(message_content, "annotations", [])
        citations = []

        for index, annotation in enumerate(annotations):
```

```
message_content.value =
message_content.value.replace(annotation.text,
f"[{index}]" )
if file_citation := getattr(
    annotation, "file_citation", None):
    cited_file = client.files.retrieve(
        file_citation.file_id)
    citations.append(f"[{index}] {cited_file.filename}")

return f'{message_content.value}\n\n{" ".join(citations)}'

except Exception as e:
    logging.error(f"Erro ao gerar resposta: {e}")
    return None
```

5.3.2.2 Contexto Utilizado

Para que fosse realizado a busca através do *File Search*, foram utilizados alguns arquivos e resoluções que norteiam o Curso de ADS (Análise e Desenvolvimento de Sistemas) do IFRO (Instituto Federal de Rondônia) - *Campus Ariquemes*. São eles:

1. **Horas Complementares 2022:** Resolução nº 24 do CONSUP.
2. **Projeto Pedagógico de Curso:** *Projeto Pedagógico de Curso ADS 2022*.
3. **Requerimento de Aproveitamento de Estudos:** *Requerimento de Aproveitamento de Estudos ADS Ariquemes*.
4. **Regulamento de TCC:** *Resolução nº 11 - Regulamento TCC - SEI*.
5. **Regimento Interno do Campus:** *Resolução nº 51 - Regimento Interno do Campus Ariquemes*.
6. **Regulamentação PSU:** *Resolução nº 71 - Regulamentação PSU corrigido*.
7. **Regulamento de Estágio:** *Resolução nº 79 - Regulamento de Estágio do IFRO - SEI*.
8. **ROA Graduação:** *Resolução nº 87 - ROA Graduação - SEI*.

Todos os arquivos citados estão disponíveis no site do Curso de ADS (Análise e Desenvolvimento de Sistemas) através da url <https://tads.arquemes.ifro.edu.br/documentos/>.

6 FUNCIONALIDADES DA APLICAÇÃO E RESULTADOS

Detalhes das funcionalidades da aplicação e os resultados obtidos com a aplicação serão apresentados neste capítulo. Serão utilizadas imagens para demonstrar o comportamento da aplicação quando realizado algum questionamento e as respostas que são retornadas.

6.1 METODOLOGIA DOS TESTES

Os testes foram conduzidos por meio de questionamentos estruturados sobre os conteúdos presentes nos documentos utilizados como contexto pelo assistente virtual. A metodologia adotada consistiu em enviar perguntas à IA e analisar sua capacidade de interpretar e processar as informações contidas nos documentos fornecidos. O objetivo principal foi verificar se a Inteligência Artificial conseguia acessar, analisar e extrair respostas relevantes dos arquivos em sua base de conhecimento.

Como o PLN do Assistente precisa interpretar uma ampla diversidade cultural em relação ao modo de escrita e fala dos alunos, tornou-se necessário repetir a mesma pergunta em diferentes formas, permitindo que o sistema se adaptasse à variedade linguística. Algumas dessas variações resultaram em respostas corretas, enquanto outras geraram respostas incorretas para a mesma pergunta formulada de maneira distinta. Além disso, em alguns casos, nenhuma resposta foi retornada.

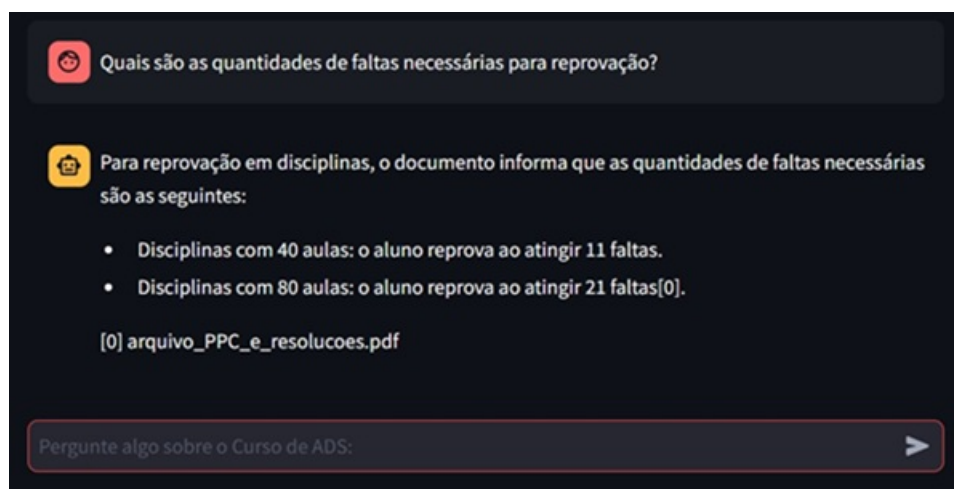
O processo de validação ocorreu em duas etapas. Primeiramente, ao receber uma pergunta, a IA tentava interpretar e localizar a informação relevante dentro dos documentos. Se a análise fosse bem-sucedida, o assistente retornava uma resposta fundamentada nos textos disponíveis. Caso contrário, o sistema identificava uma falha na recuperação da informação, o que poderia resultar em um erro na criação da *thread* responsável pelo processamento da resposta.

Para avaliar a precisão e confiabilidade das respostas, foi adotado um método de reforço supervisionado. Sempre que a resposta retornada pela IA estivesse correta, um reforço positivo era aplicado, indicando que aquela informação estava adequada. Por outro lado, caso a resposta estivesse incorreta ou imprecisa, era realizado um reforço negativo, sinalizando à Inteligência Artificial que a resposta precisava ser ajustada. Esse processo permitiu refinar progressivamente o desempenho do modelo, aprimorando sua capacidade de compreensão e resposta ao longo dos testes.

Dessa forma, a metodologia empregada possibilitou mensurar a eficácia da Inteligência na consulta a documentos institucionais, garantindo que as respostas fossem coerentes e alinhadas ao contexto acadêmico. Além disso, o uso de reforço supervisionado contribuiu para a melhoria contínua do sistema, tornando-o mais preciso e confiável para os usuários.

Abaixo a figura 1 que mostra o assistente respondendo ao questionamento da quantidade de faltas necessárias para a reprovação.

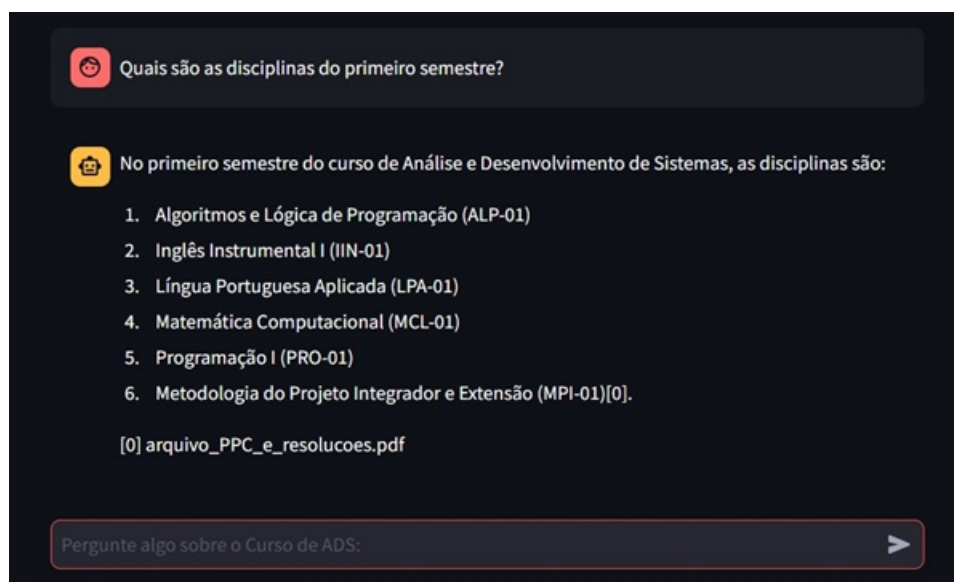
Figura 1 – Resposta à questão da quantidade de faltas.



Fonte: Elaborado pelo autor, 2025.

Temos também a figura 2 que mostra quando o assistente de IA (Inteligência Artificial) retorna corretamente à pergunta feita.

Figura 2 – Resposta à questão das disciplinas do semestre.



Fonte: Elaborado pelo autor, 2025.

6.2 AVALIAÇÃO DE EFICIÊNCIA DO ASSISTENTE

Com os testes realizados no assistente, foi possível obter os seguintes resultados:

- Total de testes realizados: **100**.

- Testes que retornaram respostas: **92 (92%** de eficiência).
- Testes que não retornaram respostas: **8 (8%** de ineficiência).
- Testes que retornaram respostas corretas: **78 (84,78%** de eficiência em relação aos testes que retornaram respostas).
- Testes que retornaram respostas incorretas: **14 (15,22%** em relação aos testes que retornaram respostas).

6.2.1 Cálculos de Eficiência

Os índices de eficiência foram calculados da seguinte forma:

1. Eficiência dos testes que retornaram respostas:

$$\text{Eficiência} = \frac{\text{Testes que retornaram respostas}}{\text{Total de testes realizados}} \times 100$$

Substituindo os valores:

$$\text{Eficiência} = \frac{92}{100} \times 100 = 92\%$$

2. Eficiência dos testes que retornaram respostas corretas:

$$\text{Eficiência} = \frac{\text{Testes que retornaram respostas corretas}}{\text{Testes que retornaram respostas}} \times 100$$

Substituindo os valores:

$$\text{Eficiência} = \frac{78}{92} \times 100 \approx 84,78\%$$

6.2.1.1 Respostas não retornadas

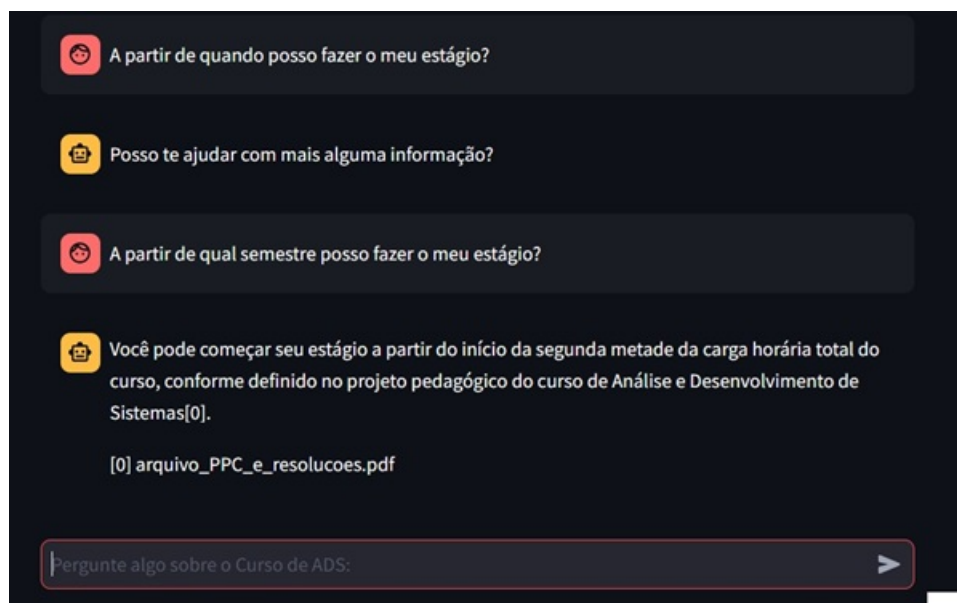
Alguns testes não retornaram respostas, pois a API não conseguiu converter a *thread* para um formato compatível com o *File Search*. Como resultado, o mecanismo de busca não pôde interpretar corretamente a solicitação e localizar a resposta adequada no arquivo correspondente.

6.2.2 Interpretação dos Resultados

Os dados demonstram que o assistente obteve uma eficiência geral de **92%** no retorno de respostas aos testes realizados. Entre as respostas fornecidas, **84,78%** foram corretas, evidenciando a capacidade do assistente de responder adequadamente à maior parte das solicitações. No entanto, **15,22%** das respostas retornadas apresentaram erros, indicando oportunidades de melhoria na precisão das respostas fornecidas.

Na figura 3 podemos ver o assistente respondendo à questão, mas respondendo incorretamente. Após ser questionado novamente, ele responde corretamente.

Figura 3 – Assistente respondendo a mesma questão escrita de duas formas.



Fonte: Elaborado pelo autor, 2025.

7 CONCLUSÃO

A aplicação do assistente fundamentado em IA se mostrou uma alternativa promissora para aprimorar o acesso a informações acadêmicas, lidando com um dos maiores desafios dos estudantes: a demanda por respostas ágeis e exatas sem a intervenção direta de docentes ou coordenadores. Este estudo utiliza a tecnologia para automatizar processos repetitivos, aprimorar a comunicação institucional e unificar as informações em um único ambiente digital, simplificando o acesso aos documentos institucionais e diminuindo a pressão sobre a equipe acadêmica.

7.1 CONTRIBUIÇÕES DO TRABALHO

O projeto revelou um efeito notável na dinâmica de acesso a informações acadêmicas. Com a introdução do assistente, os estudantes ganham um meio rápido, acessível e disponível 24 horas por dia para tirar dúvidas sobre o curso, prazos institucionais e normas, eliminando a necessidade de consultar a coordenação com frequência. Isso não só maximiza o tempo dos administradores acadêmicos, possibilitando-lhes focar em atividades mais estratégicas, mas também fomenta a independência dos alunos, estimulando-os a procurar informações por conta própria.

Outra característica notável é o agrupamento das informações institucionais, prevenindo a dispersão de dados e simplificando a procura por documentos cruciais, tais como regulamentos, normas e diretrizes acadêmicas. O assistente atua como um apoio inteligente, assegurando que todas as respostas estejam em conformidade com os documentos oficiais do curso, o que aumenta a transparência e a confiabilidade das informações compartilhadas.

Ademais, o estudo destaca a relevância da Inteligência Artificial na educação, evidenciando que instrumentos fundamentados em IA podem ser fortes aliados na modernização do ensino universitário. O emprego de PLN e ferramentas de pesquisa em documentos potencializa a habilidade do assistente em entender e responder às questões dos estudantes de forma mais natural e contextual, tornando a interação mais fluida e eficaz.

7.2 LIMITAÇÕES ENCONTRADAS

Embora tenham sido feitas contribuições significativas, algumas restrições foram detectadas durante a criação e execução do assistente virtual. A principal dificuldade reside no custo da API empregada, que, mesmo não sendo alta, constitui um gasto constante para a manutenção do sistema. Ademais, a necessidade de uma API externa pode resultar em possíveis limitações de uso, atualizações e ajustes necessários para se adequar às transformações tecnológicas.

Outro aspecto a levar em conta é a necessidade de formação contínua do assistente,

já que, sem um processo apropriado de aprimoramento e atualização, existe a possibilidade de respostas imprecisas ou fora de contexto. A qualidade do serviço está diretamente relacionada à calibração do modelo, demandando esforços contínuos para fornecer ao assistente novas informações e corrigir eventuais erros na interpretação das questões.

7.3 SUGESTÕES PARA MELHORIAS FUTURAS

Para melhorar a solução criada e torná-la ainda mais eficaz, é possível aplicar algumas melhorias. Uma opção viável seria substituir o *File Search* externo por um sistema interno de pesquisa de arquivos, diminuindo a necessidade de APIs externas e reduzindo os gastos operacionais. Esta estratégia proporcionaria maior controle sobre as informações e a capacidade de customizar os algoritmos de pesquisa de acordo com as demandas particulares da instituição.

Ademais, o investimento em métodos de aprendizagem contínua pode melhorar a habilidade do assistente de se desenvolver de forma autônoma, diminuindo a demanda por intervenções manuais na formação do modelo. É possível explorar métodos de aprendizado supervisionado e não supervisionado para que o sistema aprenda com as interações dos estudantes e se ajuste a novas necessidades ao longo do tempo.

Outra proposta seria estender a funcionalidade do assistente para diversos meios de comunicação, integrando-o a aplicativos de mensagens, redes sociais e plataformas acadêmicas, como o AVA (Ambiente Virtual de Aprendizagem) ou ao aplicativo IFRO Mobile, aumentando sua acessibilidade e visibilidade. Assim, os estudantes poderiam obter informações de forma ainda mais prática, sem a necessidade de utilizar apenas a interface do *Streamlit*.

Além do Assistente Virtual voltado para os alunos, há também a possibilidade de desenvolver uma versão específica para atender exclusivamente os servidores da instituição. Esse assistente teria como objetivo esclarecer dúvidas recorrentes, padronizar a disseminação de informações institucionais e facilitar o acesso a procedimentos administrativos. Dessa forma, seria possível otimizar a comunicação interna, reduzir a sobrecarga dos setores responsáveis pelo suporte e garantir maior eficiência na execução dos processos dentro do ambiente acadêmico.

7.4 CONTINUIDADE E EXPANSÃO DO PROJETO

Para garantir que este projeto possa ser continuado e aprimorado por futuros alunos, é essencial documentar e estruturar o desenvolvimento de forma acessível e organizada. A seguir, são apresentadas diretrizes e recomendações para a manutenção, expansão e possível evolução do Assistente Virtual Inteligente.

7.4.1 Documentação Técnica e Organizacional

A continuidade do projeto depende de uma documentação clara e bem estruturada. Recomenda-se que os futuros desenvolvedores mantenham e atualizem os seguintes documentos:

- **Guia de Instalação e Configuração:** Um passo a passo detalhado sobre a configuração do ambiente de desenvolvimento, incluindo a instalação das dependências, configuração da API da OpenAI e hospedagem no Streamlit Community Cloud.
- **Manual de Código:** Comentários no código-fonte explicando a função de cada módulo, facilitando a compreensão e modificação futura.
- **Registro de Testes e Resultados:** Armazenamento das métricas de eficiência, incluindo taxas de resposta correta, falhas e possíveis melhorias.
- **Lista de Melhorias Futuras:** Sugestões para aprimoramento do assistente, como novas funcionalidades, otimização de desempenho e ampliação para outros públicos dentro da instituição.

7.5 CONSIDERAÇÕES FINAIS

A introdução do assistente virtual evidenciou a capacidade da Inteligência Artificial como instrumento estratégico na educação superior, fomentando um acesso mais igualitário e eficaz à informação. O projeto enfatizou a relevância da atualização dos meios de comunicação acadêmicos, diminuindo a pressão sobre a coordenação e fortalecendo a independência dos alunos.

Apesar de obstáculos como custos operacionais e a exigência de formação contínua ainda necessitarem de aprimoramento, as propostas apresentadas apontam para estratégias viáveis para tornar a ferramenta ainda mais sólida e sustentável. Com essas modificações, o assistente poderá progredir para um sistema mais inteligente, acessível e unificado, estabelecendo-se como uma solução crucial para aprimorar a experiência no ambiente acadêmico e atualizar a administração educacional.

O projeto está disponível através do repositório no *GitHub*: <https://github.com/lucasaludo/cstads.git>.

REFERÊNCIAS

ARAÚJO, Samuel Gomes Bezerra. **Um estudo de caso sobre avaliação de uma API de sistema Web baseado em Microserviços**. [S.l.: s.n.], jul. 2021. Trabalho de Conclusão de Curso (Graduação) – Universidade Federal da Paraíba, João Pessoa, 2021. Acesso em: 1 fev. 2025. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/31736>.

BALDIN, Cleison Pinter; SCHAMBECK, Marcelo M.; MATOS, Samuel Duarte; CRESCENCIO, Wendel. **A Inteligência Artificial na Automatização de Processos**. [S.l.: s.n.], s.d. Acesso em: 1 fev. 2025. Disponível em: https://egov.ufsc.br/portal/sites/default/files/a_inteligencia_artificial_na_automatizacao_de_processos.pdf.

BERNO, Adriana; PEIXE, Adriana Maria Miguel; BALSAN, Jorge. O uso da inteligência artificial na gestão de documentos e de dados. **Revista IBICT**, s.d. Acesso em: 1 fev. 2025.

BRASIL. **Constituição da República Federativa do Brasil de 1988**. [S.l.: s.n.], 1988. Art. 5º, inciso XXXIII. Acesso em: 1 fev. 2025. Disponível em: https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm.

CASELI, H. M.; NUNES, M. G. V. (Ed.). **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. 2. ed. [S.l.]: BPLN, 2024. ISBN 978-65-00-95750-1.

CLOUD, Google. **O que é inteligência artificial?** [S.l.: s.n.], s.d. Acesso em: 1 fev. 2025. Disponível em: <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=pt-BR>.

FUNDAÇÃO GETULIO VARGAS. **Estudo analisa o impacto da inteligência artificial na comunicação corporativa nas redes sociais**. [S.l.: s.n.], jun. 2024. Acesso em: 1 fev. 2025. Disponível em: <https://portal.fgv.br/noticias/estudo-analisa-impacto-inteligencia-artificial-comunicacao-corporativa-redes-sociais>.

GANDRA, Alana. **Cientistas veem riscos e benefícios da inteligência artificial**. [S.l.: s.n.], nov. 2023. Entrevista publicada em 9 nov. 2023. Acesso em: 1 fev. 2025. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2023-11/cientistas-veem-riscos-e-benef%C3%ADcios-da-intelig%C3%A4ncia-artificial>.

GARCIA, Ana Cristina Bicharra. Ética e Inteligência Artificial. **Computação Brasil**, n. 43, p. 14–22, nov. 2020.

GELLERTH, Karin. **As Barreiras na Comunicação Organizacional**. [S.l.: s.n.], s.d. Acesso em: 1 fev. 2025. Disponível em: <https://esic.br/wp-content/uploads/2023/04/AS-BARREIRAS-DA-COMUNICACAO-ORGANIZACIONAL-220414.pdf>.

GONÇALEZ, Fernando Feliu; MOREIRA, Rui Silva; TORRES, Jose Manual. **UMA PROPOSTA DE SOLUÇÃO ARQUITETURAL PARA ATENDIMENTO AUTOMATIZADO UTILIZANDO CHATBOTS**. [S.l.: s.n.], s.d. Acesso em: 1 fev. 2025. Disponível em: https://www.researchgate.net/profile/Fernando-Gonzalez/publication/339069899_UMA_PROPOSTA_DE_SOLUCAO_ARQUITETURAL_PARA_ATENDIMENTO_AUTOMATIZADO_UTILIZANDO_CHATBOTS/links/5e3bf856299bf1cdb9117722/UMA-PROPOSTA-DE-SOLUCAO-ARQUITETURAL-PARA-ATENDIMENTO-AUTOMATIZADO-UTILIZANDO-CHATBOTS.pdf.

IBM. **Inteligência Artificial (IA)**. [S.l.: s.n.], s.d. <https://www.ibm.com/br-pt/topics/artificial-intelligence>. Acesso em: 1 fev. 2025.

MOREIRA, Patrícia Alves; RANZATTI, Mario Augusto; ROSINI, Alessandro Marco. As Soluções de TIC na Política Pública de Acesso à Informação e Transparência da União – O caso do e-SIC / ICT Solutions in the Public Policy of Access to Information and Transparency of the Union - The case of e-SIC. **Brazilian Journal of Business**, v. 2, n. 4, p. 4042–4055, dez. 2020.

OPENAI. **API OpenAI**. [S.l.: s.n.], jun. 2020. Acesso em: 1 fev. 2025. Disponível em: <https://openai.com/index/openai-api/>.

SOARES, Raquel Solares; SANTOS, Samara Prass dos; MARQUES, Cibele Cruz; CARVALHO, Gustavo de; SANTOS, Felipe Fernando Muller dos; YOSHIZUMI, Maíra Tiemi; ALVA, Thatiane Alves Pianoschi; TROMBINI, Henrique. Implementação de um laboratório virtual de física médica. **Revista Brasileira de Ensino de Física**, jul. 2024. Acesso em: 1 fev. 2025. ISSN 1806-9126.

SOUTO, Hiarly Fernandes de. **Uma análise comparativa entre threads e green threads no Java**. [S.l.: s.n.], maio 2024. Trabalho de Conclusão de Curso (Graduação) – Universidade Federal de Campina Grande, Campina Grande, 2024. Acesso em: 1 fev. 2025. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/>

38147/HIARLY%20FERNANDES%20DE%20SOUTO-ARTIGO-CI%c3%8aNCIA%20DA%20COMPUTA%c3%87%c3%83O-CEEI%20%282024%29.pdf?sequence=1&isAllowed=y.

TRINDADE, Alessandra Stefane Cândido Elias da; OLIVEIRA, Henry Poncio Cruz de. Inteligência Artificial (IA) Generativa e Competência em Informação: Habilidades Informacionais Necessárias ao Uso de Ferramentas de IA Generativa em Demandas Informacionais de Natureza Acadêmica-Científica. **SCIELO Brasil**, maio 2024. Acesso em: 1 fev. 2025.

TRT9. **Conceito: Arquitetura de Software**. [*S.l.: s.n.*], s.d. Acesso em: 1 fev. 2025. Disponível em: https://www.trt9.jus.br/pds/pdstrt9/guidances/concepts/software_architecture_59A08DE0.html.