



**UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA QUÍMICA**

GERALD HAMILTON WICKS

**CHATBOT-POLI: UMA PROPOSTA DE MVP PARA
AUTOMATIZAÇÃO DE RESPOSTAS PARA DÚVIDAS
ACADÊMICAS COM OPENAI**

GERALD HAMILTON WICKS

CHATBOT-POLI: UMA PROPOSTA DE MVP PARA AUTOMATIZAÇÃO DE
RESPOSTAS PARA DÚVIDAS ACADÊMICAS COM OPENAI

Documento apresentado como trabalho de conclusão de curso na modalidade software, no curso de Engenharia de Controle e Automação, cursado na Universidade Federal da Bahia - UFBA.

Orientador: Raony Fontes

UNIVERSIDADE FEDERAL DA BAHIA
CURSO DE GRADUAÇÃO ENGENHARIA DE CONTROLE E
AUTOMAÇÃO DE PROCESSOS

FOLHA DE APROVAÇÃO

GERALD HAMILTON WICKS

CHATBOT-POLI: UMA PROPOSTA DE MVP PARA AUTOMATIZAÇÃO DE
RESPOSTAS PARA DÚVIDAS ACADÊMICAS COM OPENAI

Trabalho final de conclusão de curso apresentado ao curso de Engenharia de Controle e Automação de Processos da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação de Processos

Orientador: Dr. Raony Maia Fontes

Aprovado em: 14 de dezembro de 2023.

Banca Examinadora

Prof. Dr. Raony Maia Fontes

Prof Dr Daniel Diniz Santana

Prof Dr Joao Thiago de Guimarães Anchieta e Araújo Campos

RESUMO

Este trabalho aborda o desenvolvimento de um chatbot baseado na tecnologia OpenAI, destinado a auxiliar estudantes da nossa instituição na resolução de dúvidas. O chatbot busca respostas em documentos em PDF, incluindo resoluções do colegiado. Utilizando um modelo de linguagem avançado, o chatbot compreende e responde às perguntas dos alunos, oferecendo respostas precisas e oportunas.

Palavras-Chave: Chatbot, ChatGPT, Inteligência artificial, OpenAI, PLN, LLM's.

LISTA DE FIGURAS

| | |
|--|----|
| FIGURA 1 - Interação do PLN..... | 13 |
| FIGURA 2 - Lógica do funcionamento do chatbot..... | 17 |
| FIGURA 3 - Página inicial do sistema..... | 25 |
| FIGURA 4 - Página do chatbot..... | 26 |
| FIGURA 5 - Pergunta ao chatbot..... | 27 |
| FIGURA 6 - Validação de respostas, pergunta 1..... | 38 |
| FIGURA 7 - Validação de respostas, pergunta 2..... | 38 |
| FIGURA 8 - Validação de respostas, pergunta 3..... | 39 |
| FIGURA 9 - Validação de respostas, pergunta 4..... | 39 |
| FIGURA 10 - Validação de respostas, pergunta 5..... | 40 |
| FIGURA 11 - Validação de respostas, pergunta 6..... | 40 |
| FIGURA 12 - Validação de respostas, pergunta 7..... | 41 |
| FIGURA 13 - Validação de respostas, pergunta 8..... | 41 |
| FIGURA 14 - Validação de respostas, pergunta 9..... | 42 |
| FIGURA 15 - Validação de respostas, pergunta 10..... | 42 |

LISTA DE TABELAS

| | |
|---|----|
| TABELA 1 - Avaliando as respostas do chatbot. Fonte: Autor..... | 28 |
|---|----|

LISTA DE ABREVIATURAS E SIGLAS

PLN - Processamento de linguagem natural

LLM's - Modelo de linguagem grande

GPT - Generative Pre-Trained Transformer

PDF - Portable Document Format

UFBA - Universidade Federal da Bahia

CCECA - Colegiado do Curso de Engenharia de Controle e Automação

CCEQ - Colegiado do Curso de Engenharia Química

Art. - Artigo

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO | 10 |
| 2. FUNDAMENTAÇÃO TEÓRICA | 11 |
| 2.1. Python | 11 |
| 2.2. Processamento de Linguagem Natural (PLN) | 12 |
| 2.3. Modelo de Linguagem Grande (LLM's) | 13 |
| 2.4. TypeScript | 14 |
| 2.5. React | 15 |
| 3. MATERIAIS E MÉTODOS | 15 |
| 3.1 Arquitetura do backend | 15 |
| 3.2 Arquitetura do frontend | 16 |
| 3.3 Geração de respostas | 16 |
| 3.3.1 PDFs utilizados (Fonte de conhecimento externo) | 17 |
| 3.3.2 Processamento de documentos PDFs | 18 |
| 3.3.3 Divisão de texto em trechos | 18 |
| 3.3.4 Representação vetorial de texto | 19 |
| 3.3.5 Modelagem de conversação | 19 |
| 3.3.6 Memória de conversação | 20 |
| 4. MANUAL DO USUÁRIO | 20 |
| 4.1. Pré-requisitos | 21 |
| 4.2. Clonando o projeto | 21 |
| 4.3. Configurando e rodando o backend | 22 |
| 4.4. Configurando e rodando o frontend | 23 |
| 4.5. Acessando a aplicação | 24 |
| 5. USO DO SOFTWARE | 24 |
| 5.1 Navegando pelo sistema | 24 |
| 5.2 Validação das respostas | 27 |
| 6. MANUTENÇÃO DA BASE DE DADOS | 33 |
| 7. INVESTIMENTO | 33 |
| 8. CONCLUSÃO | 34 |
| 8.1. Limitações | 35 |
| 8.2. Próximos passos | 35 |
| REFERÊNCIAS | 36 |
| APÊNDICE | 38 |

1. INTRODUÇÃO

No contexto acadêmico, é recorrente que os estudantes enfrentem dúvidas referentes às normas e regulamentos do curso, muitas vezes recorrendo a e-mails ou consultas presenciais com coordenadores ou professores. Porém, esta abordagem demanda tempo tanto dos alunos quanto dos coordenadores, que poderiam ser mais eficientemente empregados. Em diversas situações, a resposta para tais dúvidas encontra-se em resoluções do colegiado, tornando a comunicação mais acessível e direta. Entretanto, essa busca por informações pode ser uma tarefa árdua para os alunos, que frequentemente enfrentam dificuldades para acessar e compreender as informações necessárias para o esclarecimento de suas dúvidas.

A complexidade aumenta consideravelmente devido ao grande volume de informações presentes nas resoluções do colegiado. O excesso de dados disponíveis pode sobrecarregar os alunos, dificultando a localização rápida e eficaz das respostas que procuram.

Esse elevado volume de informações não apenas torna o processo de busca mais demorado, mas também contribui para outros problemas, como falta de clareza na comunicação, possíveis interpretações equivocadas das normas e até mesmo desmotivação dos alunos diante da complexidade burocrática. Dessa forma, a sobrecarga de informações não apenas dificulta o acesso à informação, mas também impacta negativamente a experiência acadêmica dos estudantes.

Com o objetivo de otimizar o processo, desenvolvemos o "chatbot-poli", um sistema para agilizar o esclarecimento de dúvidas dos estudantes. Ao usar o chatbot, os coordenadores do curso não precisarão dedicar longos períodos para lidar com questões comuns, permitindo respostas mais rápidas aos alunos. O chatbot permite que os alunos expressem suas dúvidas, recebendo respostas geradas a partir de documentos, principalmente resoluções do colegiado, das áreas de engenharia de controle e automação e engenharia química. Isso proporciona soluções ágeis e eficientes.

2. FUNDAMENTAÇÃO TEÓRICA

Para o funcionamento do "chatbot-poli", é crucial compreender as tecnologias subjacentes que contribuem para o seu desempenho. O sistema pode ser dividido em duas partes principais: o backend, responsável pelo processamento e geração de respostas para o chatbot, e o frontend, onde é desenvolvida a interface direta com o usuário.

No lado do backend, a compreensão da linguagem de programação Python é essencial. Além disso, são fundamentais o conhecimento sobre Processamento de Linguagem Natural (PLN) e os Modelos de Linguagem de Grande Escala (LLMs). O Python é a linguagem principal para a implementação do backend, enquanto o PLN e os LLMs desempenham um papel crucial no processamento avançado da linguagem para proporcionar respostas precisas e relevantes.

No frontend, é imprescindível compreender a linguagem de programação TypeScript e o React. TypeScript é a linguagem utilizada para desenvolver o frontend, enquanto o React, uma biblioteca que facilita a construção de interfaces com componentes reutilizáveis, desempenha um papel vital na criação de uma experiência de usuário intuitiva e eficiente.

De forma simplificada, quando um usuário faz uma pergunta, ela é capturada pelo frontend e enviada para o backend, passando por duas etapas de Processamento de Linguagem Natural (PLN). Inicialmente, a pergunta em linguagem humana é convertida para um formato compreensível pela máquina. Em seguida, a pergunta processada é encaminhada a um Modelo de Linguagem de Grande Escala (LLMs), treinado em grandes volumes de texto, capaz de buscar a melhor resposta em sua base de dados (neste caso, os PDFs do colegiado). A resposta gerada pelo LLMs passa por uma segunda etapa de PLN, revertendo a linguagem de máquina para a linguagem humana. Finalmente, a resposta é enviada do backend de volta para o frontend, que a apresenta ao usuário.

2.1. Python

Conforme Diego Melo (2020), "Python é uma linguagem de programação de alto nível, com sintaxe simplificada e próxima da linguagem humana, utilizada em

diversas aplicações, como desktop, web, servidores e ciência de dados. Este guia para iniciantes fornece mais informações sobre a origem e as principais vantagens da linguagem."

Em resumo, a escolha do Python como backend foi motivada por três razões principais: a simplicidade da linguagem, a vasta comunidade que facilita o acesso a tutoriais e materiais, e a ampla gama de bibliotecas disponíveis que facilitam o desenvolvimento.

2.2. Processamento de Linguagem Natural (PLN)

Segundo Williane Magalhães (2022), o processamento de linguagem natural (PLN) é uma subárea da inteligência artificial. Ele estuda a capacidade e as limitações de uma máquina para compreender a linguagem dos seres humanos. Seu processo se dá na criação de programas capazes de interpretar mensagens codificadas na linguagem natural, a fim de decifrá-las para uma linguagem da máquina. Esse desafio é o que move o avanço de aprendizado do PLN.

A tecnologia tem avançado rapidamente, graças ao interesse de intensificar a comunicação seres humanos-máquina. O objetivo é agilizar processos e obter uma relíquia valiosa para os tempos modernos: dados e informações das pessoas.

A linguagem dos computadores não é semelhante à dos seres humanos. Enquanto as pessoas utilizam signos pautados por idiomas, como inglês, espanhol, português ou chinês, as máquinas utilizam uma comunicação binária. Ela é pautada em milhões de 0 e 1, dentro de sequências lógicas. Se antigamente a interação entre seres humanos e máquinas necessitava de cartões furados, atualmente é possível conversar. Ao dizer "Alexa, apague a luz", por exemplo, o sistema reage com uma resposta compreensível ou ação. Toda essa evolução se dá por meio do processamento de linguagem natural.

Junto a outras tecnologias de inteligência artificial, como machine learning e deep learning, o Processamento de Linguagem Natural (PLN) converte nossa linguagem em uma comunicação compreensível para os computadores. Em última análise, o PLN atua como um intermediário na interação entre máquinas e seres humanos, conforme ilustrado na Figura 1 abaixo.

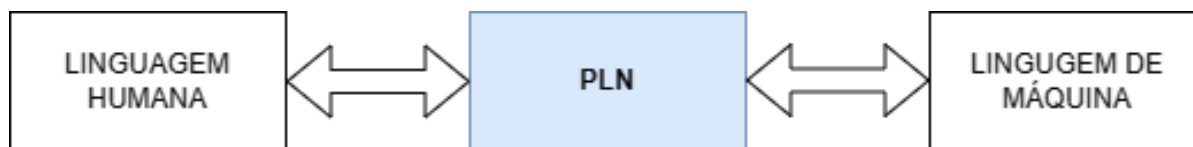


Figura 1 - Interação do PLN. Fonte: Autor

2.3. Modelo de Linguagem Grande (LLM's)

Os Large Language Models (LLMs), como GPT-3 e GPT-4, são modelos de aprendizado profundo (Deep Learning) que foram treinados em grandes quantidades de texto. Eles são capazes de gerar respostas coerentes e contextualmente relevantes para várias solicitações, incluindo responder perguntas, gerar textos, resumir textos, traduzir idiomas e muito mais.

Segundo LAYA NEELAKANDAN (2023) , 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). Também 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.

LLMs são normalmente construídos usando um tipo de rede neural chamada transformador . Os transformadores podem aprender dependências de longo alcance entre palavras, essenciais para compreender e gerar linguagem natural. Os modelos de transformadores compreendem múltiplas camadas , cada uma das quais executa uma tarefa diferente. Por exemplo, a camada de autoatenção permite que o modelo aprenda as relações entre diferentes palavras em uma frase.

Um LLM opera inicialmente aprendendo as relações estatísticas entre palavras e frases em um idioma, treinando em um vasto conjunto de dados de texto e código. Este conjunto abrange desde livros e artigos até repositórios de código e postagens em mídias sociais.

O LLM, uma vez treinado, compreende as relações estatísticas entre palavras e frases, identificando quais palavras têm maior probabilidade de aparecer juntas e como a ordem das palavras influencia o significado de uma frase. Posteriormente, o LLM pode gerar novo texto iniciando com um texto base, como algumas palavras ou uma frase. Utiliza as relações estatísticas aprendidas para prever a próxima palavra na sequência, repetindo esse processo até formar uma nova frase. Exemplificando, dado o texto inicial "O gato sentou no tapete", o LLM pode prever: "O gato sentou no tapete e encarou o pássaro", utilizando suas aprendizagens sobre quais palavras frequentemente seguem outras.

O LLM é composto por uma rede neural, um modelo matemático complexo que aprende a reconhecer padrões nos dados. Nesse contexto, a rede neural é treinada em um amplo conjunto de dados de texto e código, sendo alimentada palavra por palavra para prever a próxima na sequência. A precisão das previsões é constantemente avaliada e a rede neural é ajustada para melhorar sua exatidão, em um processo repetido até que a rede aprenda a prever com precisão a próxima palavra na sequência.

2.4. TypeScript

De acordo com Ugo Roveda (2023), "O TypeScript foi criado para adicionar recursos ausentes no JS, permitindo a definição de tipagem estática, parâmetros e retorno de funções. Além de ser uma ferramenta orientada a objetos, fortemente tipada e utilizável em qualquer ambiente de desenvolvimento, o TypeScript, quando instalado via gerenciador de pacotes JS, possibilita a verificação de erros e o uso de compiladores que suportam esse mecanismo."

A escolha do TypeScript para a linguagem do frontend baseou-se principalmente na sua tipagem estática, proporcionando maior segurança em comparação com o JavaScript. No TypeScript, é possível identificar possíveis erros de tipagem sem executar o sistema, apenas de forma estática. Além disso, como o TypeScript é

construído sobre o JavaScript, ele também possui a vantagem de ter acesso a uma ampla gama de bibliotecas disponíveis e ser popular, facilitando o acesso a tutoriais e materiais.

2.5. React

Conforme Andrei Longen (2023), “O ReactJS é uma das bibliotecas de JavaScript mais populares para desenvolvimento de aplicativos web ou para dispositivos móveis. Criado pelo Facebook, o React contém uma coleção de trechos de código de JavaScript reutilizáveis chamados componentes, usados para a construção da interface do usuário (UI).”

A escolha do ReactJS foi essencial para a construção do sistema devido à sua sintaxe simplificada e à funcionalidade de construção de componentes. Essa característica facilita a criação de códigos reutilizáveis, demandando menos esforço durante o desenvolvimento.

3. MATERIAIS E MÉTODOS

O sistema "chatbot-poli" foi desenvolvido com o propósito de fornecer respostas a perguntas de estudantes, gerando respostas com base nas informações contidas nos documentos PDFs do colegiado da faculdade. Para uma compreensão mais aprofundada do sistema, é crucial analisar a arquitetura, tanto do backend quanto do frontend. Por fim, será detalhado de forma mais aprofundada como o chatbot consegue gerar as respostas.

3.1 Arquitetura do backend

A estrutura do backend é explicada com a seguinte notação: quando uso "/" (parênteses)", refiro-me a uma pasta, não a um arquivo. Assim, a arquitetura do backend pode ser representada da seguinte maneira:

/backend: pasta raiz, é onde é carregado todas as rotas do backend.

/backend/endpoints: onde a lógica de todas as rotas são criadas.

/backend/pdfs: onde todos os pdfs utilizados no chatbot são colocados.

3.2 Arquitetura do frontend

Utilizando a mesma notação já utilizada anteriormente, podemos representar a arquitetura do frontend da seguinte maneira:

/frontend: pasta raiz que contém a maioria das configurações e onde os pacotes externos (bibliotecas de terceiros) são instalados.

/frontend/public: o conteúdo desta pasta é acessível diretamente pelo navegador e não passa por processos de construção ou empacotamento, em outras palavras, não é alterada pelo react, portanto não é muito utilizada durante o desenvolvimento.

/frontend/src: pasta principal para o uso do React, onde é escrito a maior parte do código da aplicação.

/frontend/src/pages: onde fica cada página do sistema, exemplo: página inicial, página de termos de uso, etc.

/frontend/src/components: esta pasta contém todos os componentes reutilizáveis do seu aplicativo.

/frontend/src/helpers: onde contém pedaços de código que podem ajudar nos componentes e páginas. Pode incluir funções com lógica ou tipagens que são usadas em mais de um lugar no projeto.

3.3 Geração de respostas

Para permitir que o chatbot gere respostas de maneira satisfatória, integramos duas bibliotecas cruciais no backend. A primeira é o ChatOpenAI, responsável pelo papel do LLM's (Modelo de Linguagem Grande), e a segunda é o Langchain, que desempenha o papel do PLN (Processamento de Linguagem Natural). Em resumo, o chatbot realiza o processamento de documentos PDFs, a divisão de texto, a criação de representações vetoriais, a recuperação de informações relevantes e mantém uma memória de conversação. Dessa forma, o chatbot fornece respostas contextuais ao utilizar dados do colegiado da faculdade. A lógica operacional do chatbot é esquematizada na Figura 2 abaixo.

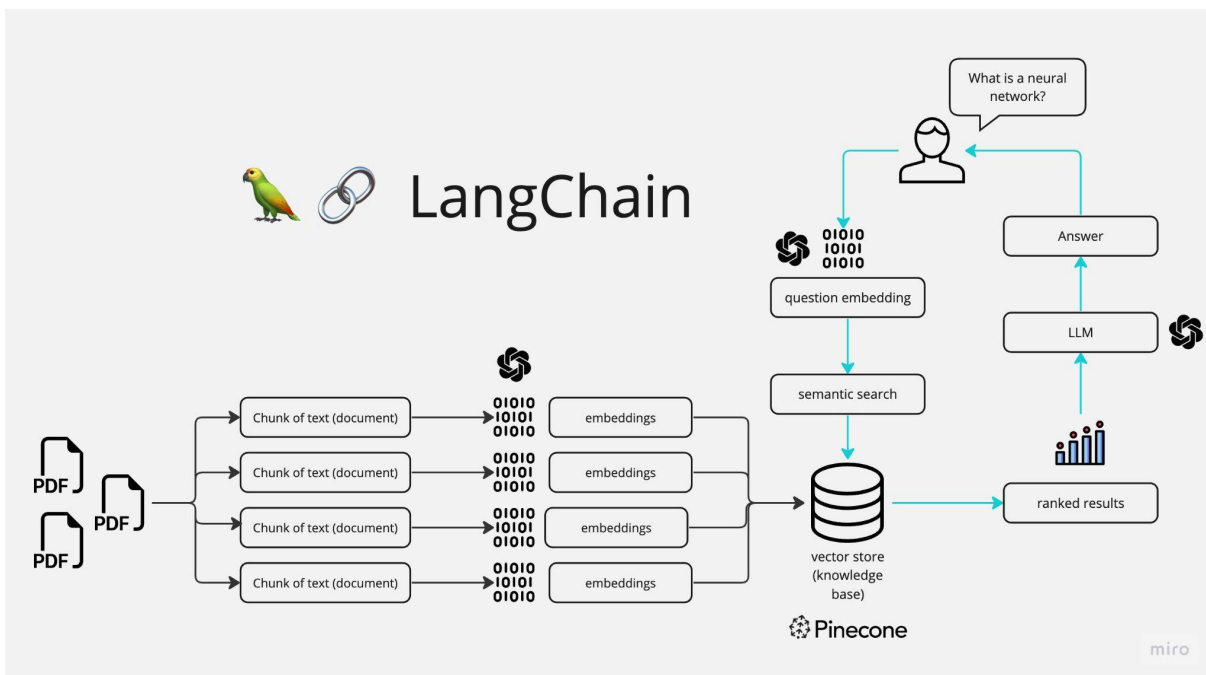


Figura 2 - Lógica do funcionamento do chatbot. Fonte: Alejandro (2023)

3.3.1 PDFs utilizados (Fonte de conhecimento externo)

Para fornecer uma base de conhecimento que possa gerar perguntas para as dúvidas estudantis, foram utilizados os seguintes PDFs.

- REGULAMENTO DE ENSINO DE GRADUAÇÃO E PÓS-GRADUAÇÃO (Atualizado de acordo com a Resolução CAE nº 03/2017)
- GUIA DO ESTUDANTE SUPAC
- RESOLUÇÃO Nº 01/2023 (Aprovada pelo CCECA em 30 de março de 2023)
- RESOLUÇÃO Nº 01/09 (Aprovada em reunião plenária de 15/05/2009, Engenharia de controle e automação)
- RESOLUÇÃO Nº 01/17 (Aprovada pelo CCECA em 09 de fevereiro de 2017)
- RESOLUÇÃO Nº 01/19 (Aprovada pelo CCECA em 26 de junho de 2019)
- RESOLUÇÃO Nº 02/17 (Aprovada pelo CCECA em 21 de novembro de 2017)

- RESOLUÇÃO N° 02/2019 (Aprovada pelo CCECA em 24 de setembro de 2019)
- RESOLUÇÃO N° 03/2019 (Aprovada pelo CCECA em 24 de setembro de 2019)
- ITENS OBRIGATÓRIOS PARA UM PROJETO DE INOVAÇÃO TECNOLÓGICA (Engenharia de controle e automação)
- RESOLUÇÃO N° 01/2012 (Aprovada pelo CCEQ em 19 de abril de 2012)
- RESOLUÇÃO N° 01/05 (Aprovada pelo CCEQ em 15 de abril de 2005 e retificada/ratificada em 10 de maio de 2005 e 15 de junho de 2005)
- RESOLUÇÃO N° 02/08 (Aprovada em reunião plenária de 09/09/2008)
- RESOLUÇÃO N. 01/2015-CCEQ (Aprovada em reunião plenária de 01 de abril de 2015)

Ou seja, em resumo, em grande maioria, os PDFs utilizados foram resoluções ou documentos dos colegiados, além disso, vale ressaltar que diante dos documentos fornecidos como base de conhecimento para o chatbot, os cursos englobados são o de engenharia de controle e automação e engenharia química.

3.3.2 Processamento de documentos PDFs

O primeiro passo na operação do sistema envolve a extração de informações dos documentos em PDF do colegiado. Isso é realizado por meio da biblioteca "PyPDF2", que permite acessar o texto contido nas páginas dos documentos e transformá-lo em uma representação textual utilizável. Essa etapa é essencial para que o sistema possa buscar informações relevantes a partir dos documentos fornecidos.

3.3.3 Divisão de texto em trechos

Após a extração do texto dos documentos PDFs, ocorre a divisão do texto em porções menores chamadas "chunks". Esse processo é realizado pelo componente "CharacterTextSplitter", proveniente da biblioteca langchain. Este componente segmenta o texto com base em critérios específicos, como um separador

(geralmente "\n" para quebra de linha) e tamanho dos chunks. A subdivisão em chunks é crucial para simplificar a manipulação e a busca de informações.

3.3.4 Representação vetorial de texto

No processo de Representação Vetorial de Texto, o sistema busca uma compreensão profunda e eficaz do conteúdo textual. Essa representação é vital para possibilitar que o sistema interprete e processe o significado inerente às palavras e frases. No âmbito desse sistema, a escolha recai sobre o uso de "embeddings" da OpenAI.

Esses "embeddings" desempenham um papel fundamental ao transformar o texto em vetores numéricos. Essa transformação permite não apenas uma representação eficiente, mas também capacita o sistema a compreender a semântica subjacente das palavras e frases. Em essência, o sistema adquire uma capacidade intrínseca de entender o significado contextual das informações contidas nos documentos PDF.

Ao traduzir o texto em uma forma numérica compreensível, o sistema está melhor equipado para buscar e recuperar dados relevantes dos documentos PDF, proporcionando uma base sólida para a geração de respostas precisas e contextualmente relevantes às perguntas dos usuários. Essa integração eficaz da representação vetorial de texto é essencial para o desempenho coerente e inteligente do sistema em todo o seu espectro de funcionalidades.

3.3.5 Modelagem de conversação

Na vertente da modelagem de conversação, o sistema adota uma abordagem dinâmica e interativa, possibilitando uma experiência mais personalizada aos usuários. O componente central, o "ChatOpenAI", desempenha um papel crucial nesse processo, atuando como um modelo especializado em interpretar e responder às perguntas dos estudantes.

Ao interagir com o "vectorstore", que representa de forma vetorial as informações extraídas dos documentos PDF subdivididos em "chunks", o sistema não apenas recupera dados relevantes, mas também modela a conversação de maneira adaptativa. Isso significa que a resposta gerada não é apenas uma

correspondência direta, mas é moldada pelo contexto da interação, proporcionando respostas mais fluidas e contextuais.

Em síntese, a modelagem de conversação vai além da mera recuperação de informações, proporcionando uma interação mais natural e contextualmente rica. Essa abordagem refinada é fundamental para criar uma experiência de chatbot mais humanizada e eficaz na resolução das dúvidas dos estudantes.

3.3.6 Memória de conversação

No contexto da Memória de Conversação, o sistema não se limita à mera recuperação de informações, mas também incorpora a capacidade de reter um histórico das interações com os usuários. Essa funcionalidade é habilmente implementada por meio do componente "ConversationBufferMemory". Essa memória não apenas registra diálogos passados, mas desempenha um papel crucial na manutenção do contexto das conversas. Ao recordar interações anteriores, o sistema pode oferecer respostas mais contextualizadas, adaptando-se de forma eficiente às nuances específicas de cada conversa. Essa abordagem não apenas aprimora a coerência das respostas ao longo do tempo, mas também contribui para uma experiência de conversação mais natural e adaptativa para os usuários.

Em resumo, o "chatbot-poli" é uma aplicação que combina várias tecnologias, desde o processamento de documentos PDFs até a representação vetorial de texto e a modelagem de conversação. Isso permite que o sistema forneça respostas precisas e relevantes aos estudantes, aproveitando as informações contidas nos documentos do colegiado da faculdade.

4. MANUAL DO USUÁRIO

Antes de explorar o funcionamento do sistema e compreender como ele opera, é crucial iniciar executando o software localmente. Isso permitirá interagir com o sistema e obter respostas de forma eficaz. Neste manual do usuário, apresentaremos um conjunto de instruções detalhadas, passo a passo, especialmente pensadas para quem deseja rodar o projeto "chatbot-poli". Este

projeto consiste em um frontend desenvolvido em React e um backend em Python. Para garantir o funcionamento correto do projeto, é fundamental seguir atentamente todas as etapas aqui descritas. No entanto, é importante notar que algum conhecimento básico em programação é necessário para a execução bem sucedida do sistema.

4.1. Pré-requisitos

Para começarmos, certifique-se de que você possui as seguintes ferramentas instaladas no seu computador:

VSCode: Um editor de código altamente recomendado. Você pode baixá-lo em <https://code.visualstudio.com>.

Python 3.11.0: A versão 3.11.0 do Python deve estar instalada em seu sistema. Você pode baixá-la em <https://www.python.org/downloads/>.

Node.js 18.12.1: O Node.js é necessário para executar o ambiente de desenvolvimento do frontend. Baixe a versão 18.12.1 em <https://nodejs.org/en/download>.

4.2. Clonando o projeto

1. Abra o VSCode após a instalação.
2. Abra o terminal no VSCode. Para fazer isso, clique em "Terminal" no menu superior e selecione "New Terminal."
3. No terminal, cole o seguinte comando para clonar o projeto a partir do repositório no GitHub:

```
git clone https://github.com/eng-ufba/chatgpt-poli.git
```

Observe que utilizamos o terminal para baixar o código do chatbot-poli, mas você também pode encontrar todo o código do chatbot-poli no seguinte website:

<https://github.com/eng-ufba/chatbot-poli>.

4.3. Configurando e rodando o backend

Antes de rodar o backend, você precisa configurar a variável de ambiente **OPENAI_API_KEY**.

1. Acesse o [site da OpenAI](#).
2. Crie uma conta ou faça login em sua conta existente.
3. Navegue para a seção de chaves de API em <https://platform.openai.com/account/api-keys>.
4. Clique em "Create new secret key" para gerar uma nova chave de API.
5. Copie a chave API gerada.
6. Abra o terminal do VSCode.
7. Navegue até a pasta do backend usando o comando "cd":

```
cd chatgpt-poli/backend
```

8. Vá até a pasta do backend, e crie um arquivo chamado **.env**
9. Abra o arquivo **.env** com um editor de texto, como o VSCode:
10. Cole a chave API copiada no arquivo `.env` no seguinte formato:

```
OPENAI_API_KEY=sua-chave-aqui
```

Agora, vamos configurar e rodar o backend .

11. Instale os pacotes do Python listadas no arquivo `requirements.txt` com o comando:

```
pip install -r requirements.txt
```

12. Após a instalação bem-sucedida das dependências, inicie o servidor Flask com o seguinte comando:

```
python app.py
```

Caso o backend esteja rodando corretamente, a seguinte mensagem irá aparecer no terminal:

```
Running on http://127.0.0.1:5000
```

4.4. Configurando e rodando o frontend

Agora, vamos configurar e executar o frontend do projeto.

1. Abra um novo terminal no VSCode, e navegue para a pasta frontend, com os comandos:

```
cd chatgpt-poli  
cd frontend
```

2. Instale as dependências do Node.js executando o comando:

```
npm ci
```

3. Após a instalação das dependências do frontend, inicie o servidor de desenvolvimento com o seguinte comando:

```
npm start
```

Caso o frontend tenha rodado corretamente, ele deve abrir automaticamente no navegador o chatbot-poli, através da url <http://localhost:3000/>.

4.5. Acessando a aplicação

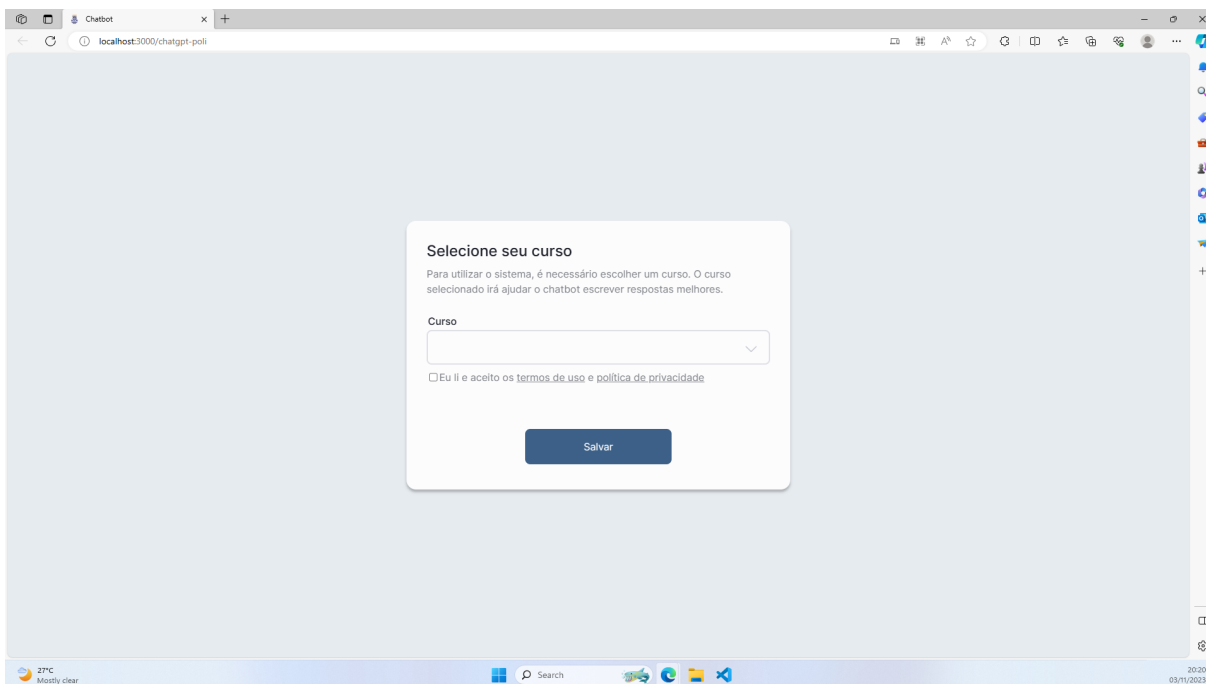
Abra o seu navegador da web e insira o seguinte endereço na barra de endereços: <http://localhost:3000/>. Agora, você deverá ser capaz de acessar a aplicação que combina o frontend React e o backend Python com Flask. Por fim, se por algum motivo você não conseguir seguir o passo a passo descrito anteriormente, pode acompanhar os passos neste vídeo: <https://youtu.be/0t4817rhC0Y>, criado para ajudar neste processo.

5. USO DO SOFTWARE

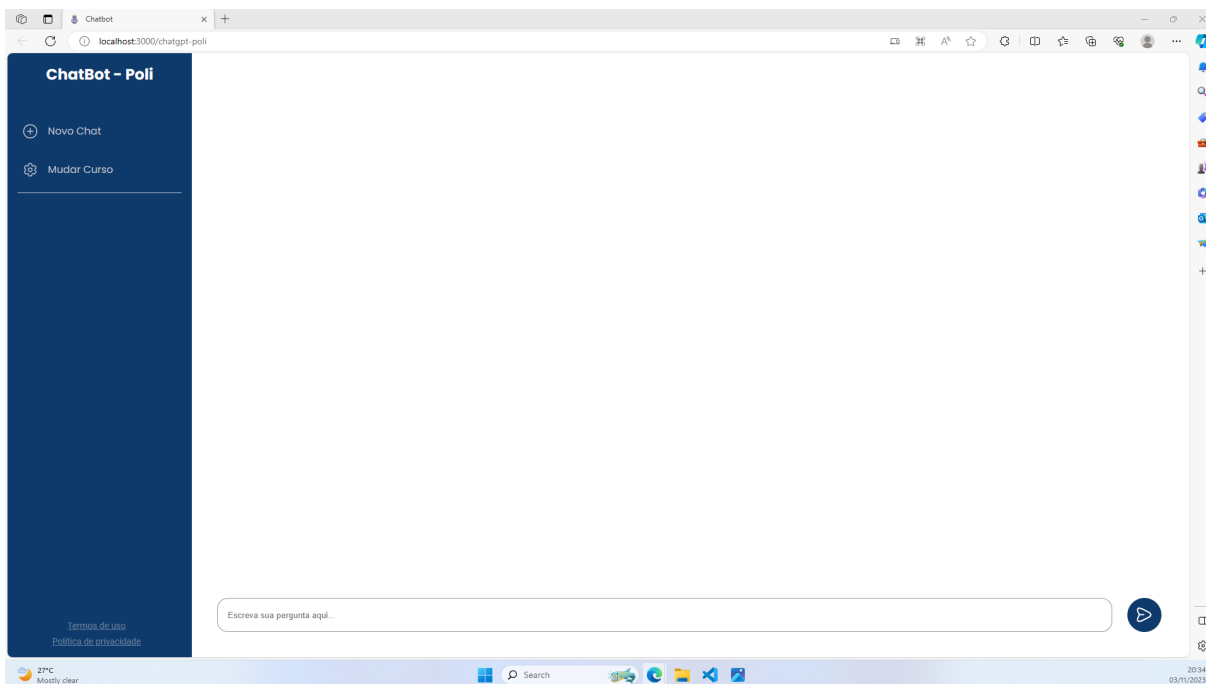
Aqui, será apresentado um guia prático que demonstrará como utilizar o "chatbot-poli". Serão fornecidas instruções detalhadas e imagens do sistema para orientar os usuários, desde o processo de configuração até a interação com o chatbot. A abordagem visual e passo a passo permitirá que os usuários compreendam facilmente como utilizar eficazmente essa ferramenta inovadora no contexto acadêmico. Além disso, serão realizadas algumas perguntas com o sistema para validar a precisão e a qualidade das respostas fornecidas, assegurando assim a eficácia do chatbot como fonte confiável de informações acadêmicas

5.1 Navegando pelo sistema

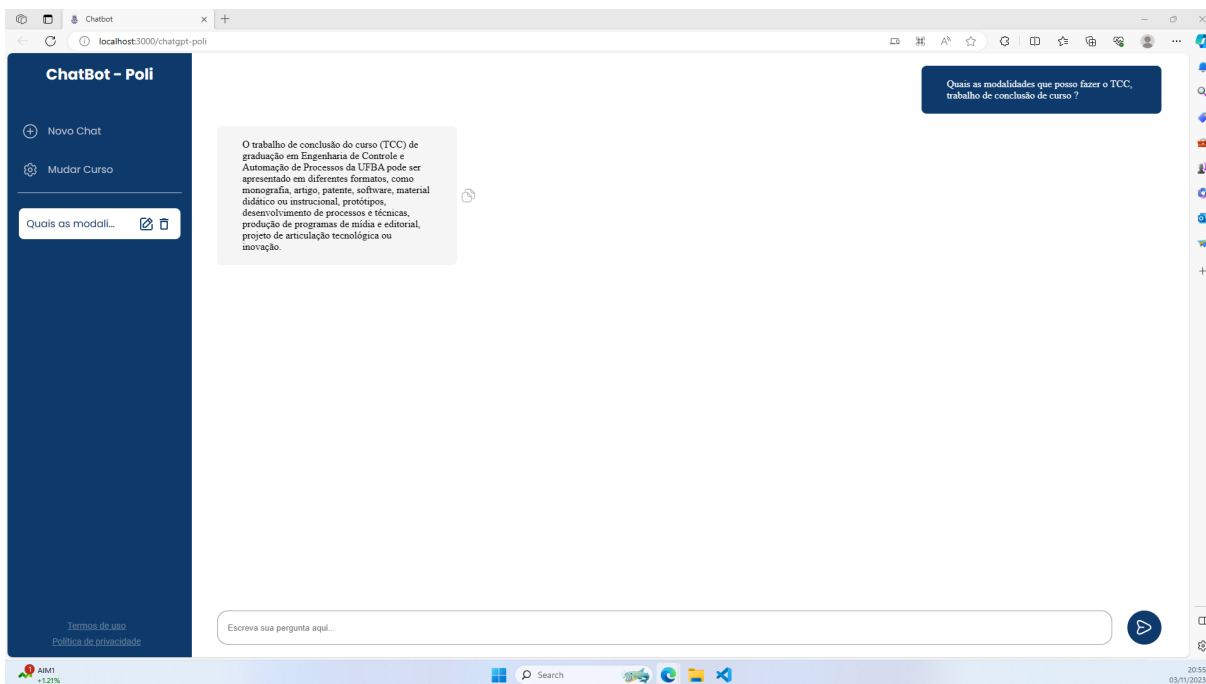
Quando acessar o sistema pela primeira vez, encontrará a tela mostrada na figura 3 abaixo. Nessa página, é preciso selecionar um curso (Engenharia de Controle e Automação ou Engenharia Química). Em seguida, marque a caixa de seleção que indica "Eu li e aceito os termos de uso e política de privacidade" e clique em "Salvar". A escolha do curso é fundamental para orientar as respostas que o estudante receberá. Além disso, a informação sobre o curso será armazenada localmente no navegador, o que significa que, se o estudante utilizar o mesmo dispositivo, a escolha do curso permanecerá registrada.

Figura 3 - Página inicial do sistema. Fonte: Autor

Após clicar em "Salvar", você será automaticamente redirecionado para a página do chatbot, apresentada na figura 4 a seguir. Nessa página, você encontrará algumas opções de interação. No canto esquerdo, existem dois botões: "Novo chat" e "Mudar Curso". O primeiro botão cria novas conversas, funcionando de maneira semelhante ao ChatGPT. Por outro lado, se você clicar no segundo botão, será direcionado de volta à página anterior, onde poderá escolher outro curso. Além disso, na parte inferior da página, há uma caixa de texto que diz "Escreva sua pergunta aqui...". Nessa caixa, você pode digitar suas perguntas para o chatbot.

Figura 4 - Página do chatbot. Fonte: Autor

Agora que estamos familiarizados com a página do chatbot, vamos fazer nossa primeira pergunta. Para começar, basta digitar a pergunta na caixa de texto localizada no canto inferior. Em seguida, clique no botão azul ao lado da caixa de texto. Após fazer isso, um ícone de carregamento aparecerá e ficará visível enquanto o sistema processa a pergunta. Logo depois, a pergunta aparecerá no lado direito, e a resposta será exibida no lado esquerdo, conforme mostrado na figura 5 abaixo.

Figura 5 - Pergunta ao chatbot. Fonte: Autor

Por fim, há algumas interações disponíveis para o usuário. Primeiramente, observe que surgiu um campo relacionado à conversa da pergunta anterior. Nesse campo, é possível alterar o nome da conversa clicando no ícone de lápis, ou excluir a conversa clicando no ícone de lixeira. É importante destacar que todas as conversas, perguntas e respostas são salvas localmente no navegador. Portanto, se sair do sistema e entrar novamente com o mesmo dispositivo, suas conversas e interações não serão perdidas.

5.2 Validação das respostas

A validação das respostas fornecidas pelo "chatbot-poli" é uma etapa crítica para assegurar a precisão e confiabilidade do sistema na transmissão de informações acadêmicas. A validação das respostas é essencial para garantir que o sistema atenda às expectativas dos estudantes e forneça informações corretas e atualizadas.

O método de validação de respostas seguirá o seguinte processo: serão formuladas perguntas ao chatbot-poli, as respostas geradas por ele serão registradas e, em seguida, avaliadas quanto à conformidade com os documentos em PDF das resoluções do colegiado. Com base nessa avaliação, será atribuída uma

nota de 0 (indicando que a resposta gerada pelo chatbot não está em conformidade com as resoluções) a 10 (indicando que a resposta gerada pelo chatbot está altamente em conformidade com as resoluções). Esse método de avaliação fornecerá uma medida objetiva da qualidade e precisão das respostas fornecidas pelo chatbot. Dessa forma, chegamos ao seguinte resumo da avaliação, apresentada na tabela 1 a seguir (as figuras que comprovam as respostas do chatbot estão disponíveis no "APÊNDICE", figuras 6 a 15):

| | |
|-----------------|---|
| Nº. da pergunta | 1 |
| Curso | Engenharia de controle e automação |
| Pergunta | Quais as modalidades que posso fazer o TCC, trabalho de conclusão de curso ? |
| Resposta | O trabalho de conclusão do curso (TCC) de graduação em Engenharia de Controle e Automação de Processos da UFBA pode ser apresentado em diferentes formatos, como monografia, artigo, patente, software, material didático ou instrucional, protótipos, desenvolvimento de processos e técnicas, produção de programas de mídia e editorial, projeto de articulação tecnológica ou inovação. |
| Nota | 10 |
| Nº. da pergunta | 2 |
| Curso | Engenharia de controle e automação |
| Pergunta | Quais atividades complementares são consideradas passíveis de aproveitamento como componente curricular ? |
| Resposta | As atividades complementares passíveis de aproveitamento como componente curricular no âmbito do CCECA-UFBA são: pesquisa científica e tecnológica, monitorias, participações em congressos e similares, visitas técnicas, estágios industriais, participações em empresas juniores, participações em centros e diretórios acadêmicos e aprovação em disciplinas optativas complementares. |
| Nota | 10 |
| Nº. da pergunta | 3 |
| Curso | Engenharia de controle e automação |
| Pergunta | Como deve ser feito para solicitar o aproveitamento de atividades complementares ? |

| | |
|-----------------|--|
| Resposta | <p>Para solicitar o aproveitamento de atividades complementares na UFBA, você deve enviar um email para aproveitamento.nae@ufba.br e seguir os seguintes passos:</p> <ol style="list-style-type: none"> 1. Baixe, preencha e assine o documento "Requerimento Escolar para Aproveitamento de Estudos", disponível em https://supac.ufba.br/formularios -menu-aluno ou clicando no link fornecido. - A assinatura deve ser manuscrita ou eletrônica, não pode ser o nome digitado. 2. Envie o documento preenchido e assinado juntamente com a documentação comprobatória da atividade realizada para o email mencionado acima. 3. Aguarde o julgamento e análise do Colegiado do curso, que irá avaliar o mérito, carga horária e outros aspectos conforme as disposições da Resolução. <p>Caso tenha mais dúvidas, você pode enviar sua pergunta para engautomacao@ufba.br.</p> |
| Nota | 0 |
| Nº. da pergunta | 4 |
| Curso | Engenharia de controle e automação |
| Pergunta | Quais motivos são considerados para trancamento do curso ? |
| Resposta | De acordo com as informações fornecidas, os motivos para trancamento do curso podem ser variados, desde motivos de saúde até outros motivos pessoais. No entanto, não há especificações sobre quais motivos são considerados válidos para o trancamento do curso. Recomendo entrar em contato com a universidade através do e-mail engautomacao@ufba.br para obter mais informações sobre os motivos aceitos para o trancamento do curso. |
| Nota | 8 |
| Nº. da pergunta | 5 |
| Curso | Engenharia de controle e automação |
| Pergunta | Quais itens deve ter um projeto de inovação tecnológica ? |
| Resposta | <p>Os itens que devem estar presentes em um projeto de inovação tecnológica são:</p> <ol style="list-style-type: none"> 1. Identificação do problema ou oportunidade: descrever de forma objetiva e sintética o foco do problema ou oportunidade, sua relevância no contexto da área de conhecimento e a motivação para o desenvolvimento de uma solução inovadora. 2. Solução proposta: caracterizar os meios, ações e instrumentos que compõem a solução proposta, destacando as razões pelas quais ela tem potencial de gerar contribuições originais. 3. Tecnologias necessárias: descrever de forma sucinta as tecnologias necessárias para o desenvolvimento da solução proposta. <p>Caso não haja uma resposta satisfatória nos textos, recomenda-se enviar a dúvida para o e-mail engautomacao@ufba.br.</p> |
| Nota | 8 |

| | |
|-----------------|--|
| Nº. da pergunta | 6 |
| Curso | Engenharia de controle e automação |
| Pergunta | Qual é o procedimento para a escolha e formalização do orientador para o trabalho de conclusão de curso (TCC)? |
| Resposta | O procedimento para a escolha e formalização do orientador para o TCC é informar ao CCECA-UFBA quem será o professor orientador, com a sua aquiescência formal, dentro de um período anterior ao semestre no qual deseja se matricular no componente curricular associado ao TCC. A coordenação então solicitará ao Departamento de Engenharia Química a criação de uma turma de ENGG05 em nome do professor orientador. |
| Nota | 10 |
| Nº. da pergunta | 7 |
| Curso | Engenharia de controle e automação |
| Pergunta | Quando pode ser concedida uma quebra de pré-requisito para um aluno ? |
| Resposta | Uma quebra de pré-requisito pode ser concedida para um aluno nas seguintes situações: - Quando o aluno está em processo de aproveitamento de estudo em andamento e o componente curricular que ele deseja se matricular é um pré-requisito para o aproveitamento de estudo. - Quando o aluno é concluinte do curso de Engenharia Controle e Automação de Processos da UFBA, no semestre em questão, e deseja se matricular com apenas uma quebra de pré-requisito. |
| Nota | 10 |
| Nº. da pergunta | 8 |
| Curso | Engenharia de controle e automação |
| Pergunta | Existe limite para o aproveitamento de atividades complementares ? |
| Resposta | Sim, existe limite para o aproveitamento de atividades complementares. No contexto fornecido, o aproveitamento de atividades complementares está limitado a um máximo de 136 horas para cursos de extensão ou extracurriculares na área ou fora da área de engenharia. Além disso, o aproveitamento de atividades de extensão está limitado a um máximo de 120 pontos, conforme especificado nos textos. |
| Nota | 7 |
| Nº. da pergunta | 9 |
| Curso | Engenharia Química |
| Pergunta | Quais motivos são considerados justos para concessão de trancamento parcial ou total ? |

| | |
|-----------------|---|
| Resposta | <p>Os motivos considerados justos para concessão de trancamento parcial ou total são:</p> <p>1. Saúde do aluno: Deve ser comprovada por atestado médico, o qual deve explicitar a incapacidade total ou parcial de acompanhamento das atividades escolares durante o período do trancamento requerido.</p> <p>2. Gestação: Deve ser comprovada por atestado médico.</p> <p>Esses são os motivos mencionados nos textos fornecidos. Se você tiver dúvidas adicionais, é recomendado entrar em contato com o e-mail engquim@ufba.br para obter informações mais precisas.</p> |
| Nota | 5 |
| Nº. da pergunta | 10 |
| Curso | Engenharia Química |
| Pergunta | No trabalho final de curso, quais as formas de apresentação da produção escrita ? |
| Resposta | <p>As formas de apresentação da produção escrita no trabalho final de curso podem ser:</p> <p>I. Monografia;</p> <p>II. Relatório técnico;</p> <p>III. Projeto de engenharia;</p> <p>IV. Protótipo acrescido de descrição técnica detalhada.</p> |
| Nota | 8 |
| Legenda | Nota 0-3: indicando que a resposta gerada pelo chatbot não está em conformidade com as resoluções |
| | Nota 4-6: indicando que a resposta gerada pelo chatbot apresenta algo correto, mas ainda faltam muitos itens para estar em conformidade com as resoluções. |
| | Nota 7-9: indicando que a resposta gerada pelo chatbot é boa e bem parecida com a resoluções, porém falta poucos itens para ser igual as resoluções |
| | Nota 10: indicando que a resposta gerada pelo chatbot está altamente em conformidade com as resoluções |

Tabela 1 - Avaliando as respostas do chatbot. Fonte: Autor

Após compilar perguntas e respostas, atribuímos uma nota qualitativa a cada resposta do chatbot-poli. A média das notas foi de 7,6, indicando eficiência geral nas respostas fornecidas pelo chatbot.

Analisando alguns casos de forma mais detalhada, é possível entender melhor a atribuição da nota de forma qualitativa, a análise é feita de forma

comparativa, comparando a resposta do chatbot com a resolução do colegiado da universidade. A resposta do chatbot referente a primeira pergunta foi:

“O trabalho de conclusão do curso (TCC) de graduação em Engenharia de Controle e Automação de Processos da UFBA pode ser apresentado em diferentes formatos, como monografia, artigo, patente, software, material didático ou instrucional, protótipos, desenvolvimento de processos e técnicas, produção de programas de mídia e editorial, projeto de articulação tecnológica ou inovação.”

Por outro lado, comparando com a resposta tirada diretamente da resolução do colegiado, podemos extrair a seguinte resposta da resolução Nº 02/17 (Aprovada pelo CCECA em 21 de novembro de 2017):

“Art. 1º O trabalho de conclusão do curso (TCC) de graduação em Engenharia de Controle e Automação de Processos da UFBA poderá ser apresentado em diferentes formatos, tais como monografia, artigo, patente, software, material didático ou instrucional, protótipos, desenvolvimento de processos e técnicas, produção de programas de mídia e editorial, projeto de articulação tecnológica ou inovação.”

Como a resposta do chatbot é idêntica à resposta da resolução do colegiado, ela foi avaliada com nota 10. Analogamente, fazendo a mesma análise para a décima pergunta, a resposta do chatbot é dada por:

“As formas de apresentação da produção escrita no trabalho final de curso podem ser: I. Monografia; II. Relatório técnico; III. Projeto de engenharia; IV. Protótipo acrescido de descrição técnica detalhada.”

Por outro lado, comparando com a resposta tirada diretamente da resolução do colegiado, podemos extrair a seguinte resposta da resolução N. 01/2015-CCEQ:

“As formas de apresentação da produção escrita podem ser: I. Monografia; II. Relatório técnico; III. Projeto de engenharia; IV. Protótipo acrescido de descrição

técnica detalhada; V. Artigo técnico-científico aceito em congresso ou revista; VI. Patente depositada.”

Como a resposta do chatbot não incluiu os itens V e VI, ela foi avaliada com nota 8. Por fim, a análise do texto mostrou que a resposta do chatbot é, em geral, precisa e completa. No entanto, existem alguns casos em que a resposta não é totalmente compatível com a resolução do colegiado.

6. MANUTENÇÃO DA BASE DE DADOS

Ao longo do tempo, a base de dados fornecida ao chatbot pode tornar-se desatualizada, exigindo a substituição dos PDFs das resoluções do colegiado. Para realizar essa atualização, é crucial compreender o local específico onde as alterações devem ser feitas. Como previamente detalhado no capítulo de "Materiais e Métodos", os PDFs carregados pelo sistema estão localizados na pasta "/backend/pdfs", que, por sua vez, possui as seguintes subpastas:

/backend/pdfs/automacao: contém todos os PDFs relacionados à Engenharia de Controle e Automação, carregados pelo chatbot.

/backend/pdfs/quimica: contém todos os PDFs relacionados à Engenharia Química, carregados pelo chatbot.

Portanto, se houver a necessidade de adicionar, editar ou excluir algum PDF, basta acessar as subpastas mencionadas e efetuar as alterações desejadas.

7. INVESTIMENTO

Embora o "chatbot-poli" seja uma solução automatizada para fornecer respostas aos alunos, é fundamental analisar os custos envolvidos no seu funcionamento, especialmente considerando o uso da tecnologia OpenAI. A pergunta que surge naturalmente é se o investimento no sistema vale a pena, e quanto custa para utilizar o sistema.

Ao realizar uma análise detalhada de custos, é possível obter uma visão clara da eficiência econômica do "chatbot-poli" em relação a outras abordagens. Primeiramente, é importante considerar o custo por pergunta feita ao sistema. Após um exame de 24 perguntas houve um gasto total de R\$ 0,15 (com a cotação do dólar de aproximadamente R\$ 5,00), com isso, chegamos a um custo de aproximadamente R\$ 0,00625 por pergunta ao utilizar o "chatbot-poli". Em outras palavras, para 1600 perguntas, o investimento seria de apenas R\$ 10,00!

Além disso, um fator intangível a ser considerado é que a implementação do "chatbot-poli" não apenas economiza recursos financeiros para a instituição, mas também libera o tempo dos professores e coordenadores para tarefas mais estratégicas, melhorando a eficiência administrativa.

Portanto, ao ponderar a relação custo-benefício, o "chatbot-poli" se destaca como uma opção altamente econômica e eficiente para fornecer respostas aos estudantes. Garante o acesso rápido e econômico a informações essenciais, ao mesmo tempo em que alivia a carga de trabalho do corpo docente e dos coordenadores. Essa abordagem contribui para uma experiência acadêmica mais produtiva e satisfatória para todos os envolvidos.

8. CONCLUSÃO

O presente trabalho apresenta o desenvolvimento do sistema "chatbot-poli", uma aplicação que se destina a auxiliar os estudantes de nossa instituição de ensino na busca por respostas a suas dúvidas acadêmicas, com foco na obtenção de respostas através de informações contidas em documentos PDFs do colegiado. Este sistema combina uma série de componentes e conceitos de vanguarda na área de processamento de linguagem natural e recuperação de informações para atingir seu propósito. O "chatbot-poli" exemplifica como a tecnologia pode ser aplicada de maneira inovadora para aprimorar a experiência educacional dos alunos e tornar as informações acadêmicas mais acessíveis. Além disso, a capacidade do sistema de manter um histórico de conversações contribui para respostas mais contextuais.

Em resumo, o "chatbot-poli" representa uma abordagem inovadora na interação entre estudantes e informações acadêmicas, aproveitando tecnologias avançadas. Este trabalho abre caminho para o uso da IA no campo da educação e

aponta para um futuro promissor na integração de sistemas de recuperação de informações em ambientes acadêmicos, tornando a busca de informações mais eficiente e eficaz para todos os estudantes.

8.1. Limitações

Embora o chatbot-poli demonstre eficiência em sua função principal, é importante reconhecer algumas limitações que podem impactar sua performance e usabilidade. Durante a validação das respostas, observou-se que, em alguns casos, a resposta do chatbot pode não ser completamente satisfatória. Essa limitação decorre da fonte de informações restrita às resoluções do colegiado, o que implica que a capacidade de resposta está diretamente ligada ao conteúdo presente nos documentos processados. Se a pergunta do aluno envolver informações não contempladas nesses documentos, o chatbot pode não oferecer uma resposta ideal.

É relevante observar que a utilização do chatbot-poli envolve custos, uma vez que faz uso da API paga OpenAI. Essa consideração financeira deve ser ponderada pelos usuários e implementadores do sistema, uma vez que pode impactar a viabilidade econômica do projeto.

8.2. Próximos passos

O chatbot-poli representa uma inovação significativa no suporte aos estudantes de Engenharia de Controle e Automação e Engenharia Química em nossa instituição de ensino. No entanto, seu potencial vai muito além desses cursos específicos. Os próximos passos na evolução desse sistema promissor incluem a expansão para abranger todas as engenharias oferecidas pela UFBA e, posteriormente, considerar sua implementação em todos os cursos da universidade.

A visão é transformar o chatbot-poli em um recurso de amplo alcance, capaz de atender às necessidades de todos os alunos da UFBA, independentemente do curso. Isso exigirá um esforço de coleta e integração de uma variedade de documentos e informações acadêmicas em diferentes formatos e estruturas. Além disso, a personalização do sistema para atender às peculiaridades de cada curso será fundamental, permitindo que os estudantes acessem respostas precisas e relevantes.

Para melhorar a validação das respostas, uma solução seria a inclusão de mais documentos do Colegiado, abrangendo um número maior de dúvidas. Além disso, a criação de um PDF compilado com diversas perguntas e respostas poderia ampliar a base de conhecimento do chatbot, tornando-o mais robusto na oferta de respostas contextualizadas.

Além da expansão de conteúdo, os próximos passos envolvem tornar o sistema ainda mais acessível. Uma maneira de alcançar isso é disponibilizar o chatbot-poli na web, eliminando a necessidade de instalação e configuração local. Ao disponibilizar o sistema por meio de um website, o acesso às informações acadêmicas se torna mais simples e direto para os estudantes, independentemente de sua experiência em programação ou configuração de software.

Para contornar a limitação relacionada ao custo, seria ideal que a instituição, a UFBA, assumisse não apenas os custos associados à utilização da API OpenAI, mas também todos os gastos necessários para estabelecer e manter o ambiente requerido para disponibilizar o sistema por meio de um website. Isso inclui a hospedagem tanto do frontend quanto do backend. Essa abordagem poderia ser implementada como parte de um investimento estratégico, visando aprimorar a experiência acadêmica e maximizar a eficácia do chatbot-poli.

REFERÊNCIAS

Longen, Andrei. **React Descomplicado: O Que é e Como Começar Já!**. setembro, 2023. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript/>>. Acesso em: 20 de dezembro de 2023.

Roved, Ugo. **TYPESCRIPT: O QUE É, COMO COMEÇAR E QUAIS SÃO AS VANTAGENS?**. março, 2023. Disponível em: <<https://kenzie.com.br/blog/typescript/>>. Acesso em: 20 de dezembro de 2023.

Melo, Diego. **O que é Python? [Guia para iniciantes]**. 2020. Disponível em: <<https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/>>. Acesso em: 20 de dezembro de 2023.

Magalhães, Williane. **Processamento de linguagem natural: entenda o que é PLN.** julho, 2022. Disponível em: <<https://www.remessaonline.com.br/blog/processamento-de-linguagem-natural/>>. Acesso em: 23 de novembro de 2023.

NEELAKANDAN, LAYA. **O que são grandes modelos de linguagem (LLMs) e para que servem.** abril, 2023. Disponível em: <<https://fastcompanybrasil.com/tech/o-que-sao-grandes-modelos-de-linguagem-llms-e-para-que-servem/>>. Acesso em: 23 de novembro de 2023.

AO, Alejandro. **Create a ChatGPT clone using Streamlit and LangChain.** Youtube, 2023. Disponível em <<https://www.youtube.com/watch?v=laTiyQ2oYUQ>>. Acesso em: 11 de agosto de 2023.

AO, Alejandro. **Chat with Multiple PDFs | LangChain App Tutorial in Python (Free LLMs and Embeddings).** Youtube, 2023. Disponível em <<https://www.youtube.com/watch?v=dXxQ0LR-3Hg>>. Acesso em: 11 de agosto de 2023.

AO, Alejandro. **Langchain + Qdrant Cloud | Pinecone FREE Alternative (20GB) | Tutorial.** Youtube, 2023. Disponível em <<https://www.youtube.com/watch?v=VL6MAAgwSDM>>. Acesso em: 11 de agosto de 2023.

APÊNDICE

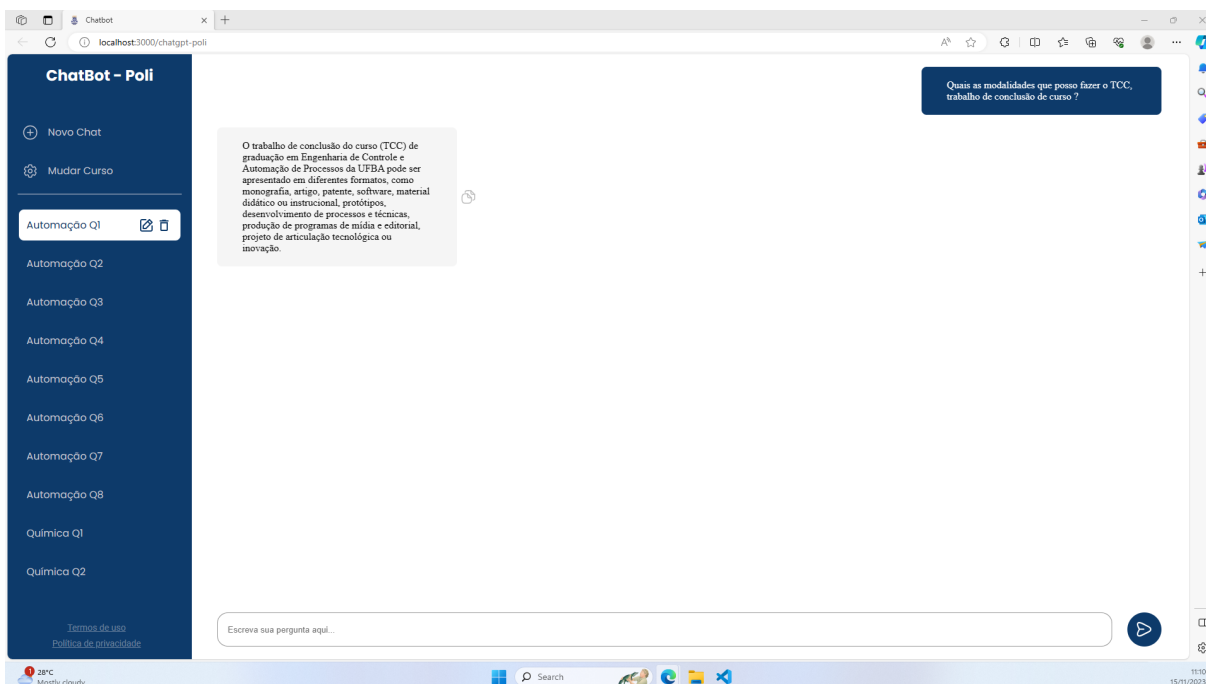


Figura 6 - Validação de respostas, pergunta 1. Fonte: Autor

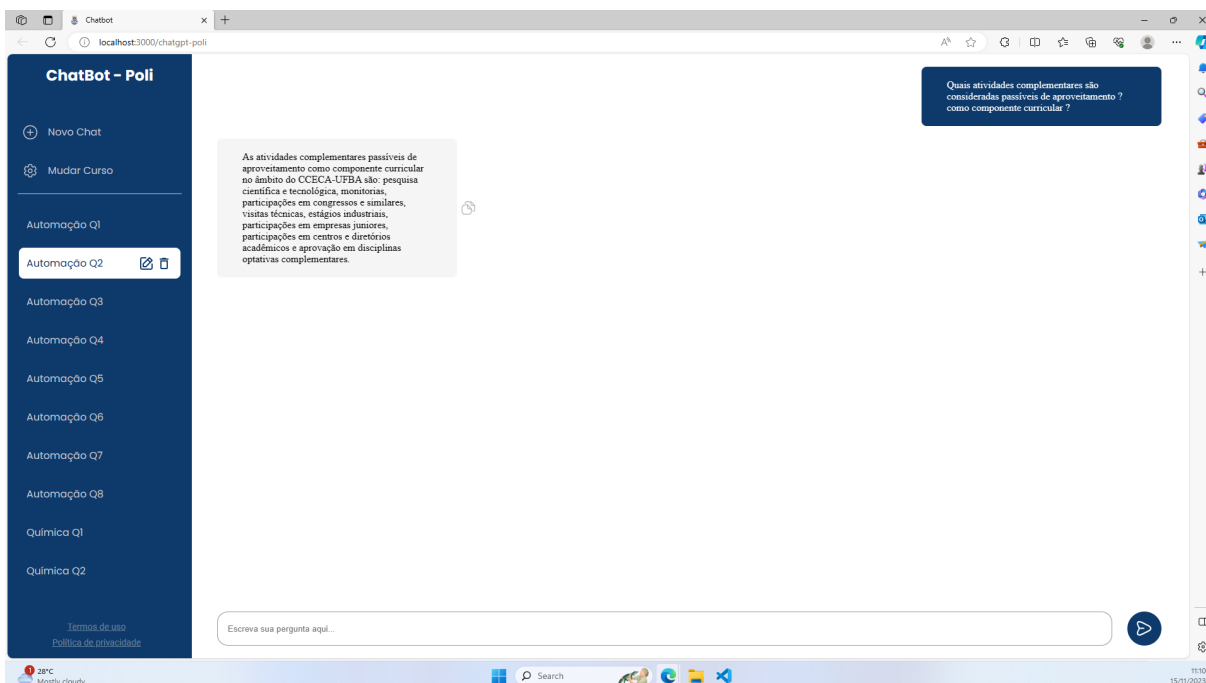


Figura 7 - Validação de respostas, pergunta 2. Fonte: Autor



Figura 8 - Validação de respostas, pergunta 3. Fonte: Autor

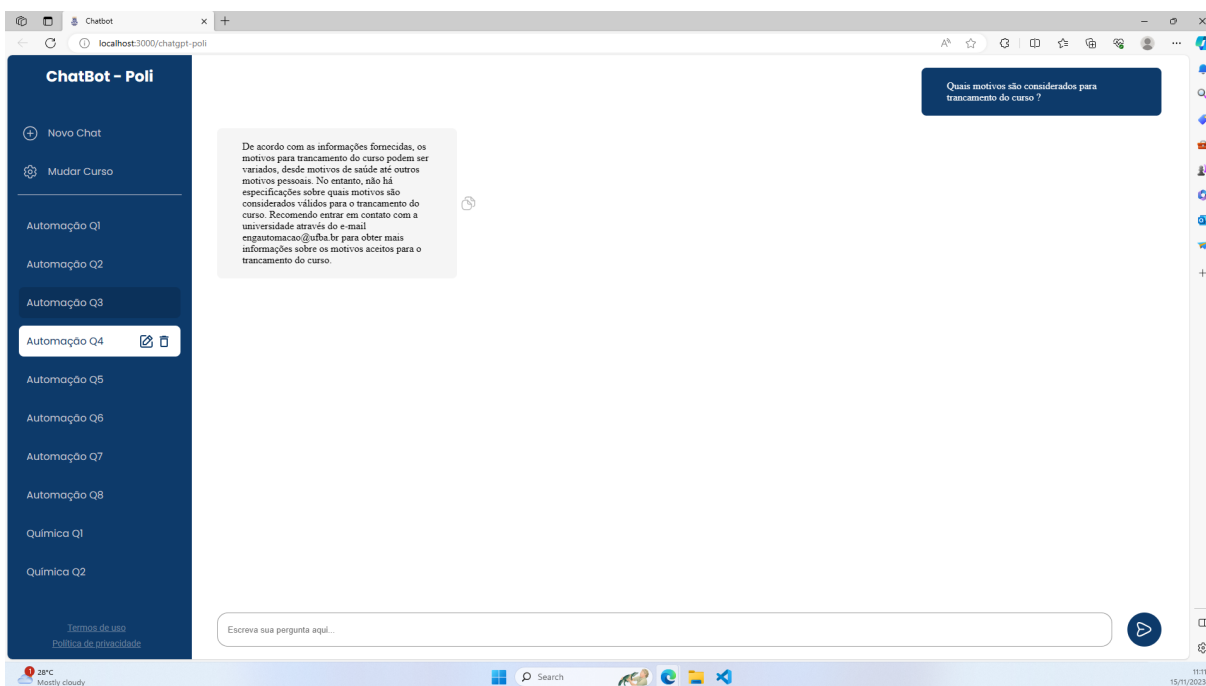


Figura 9 - Validação de respostas, pergunta 4. Fonte: Autor



Figura 10 - Validação de respostas, pergunta 5. Fonte: Autor

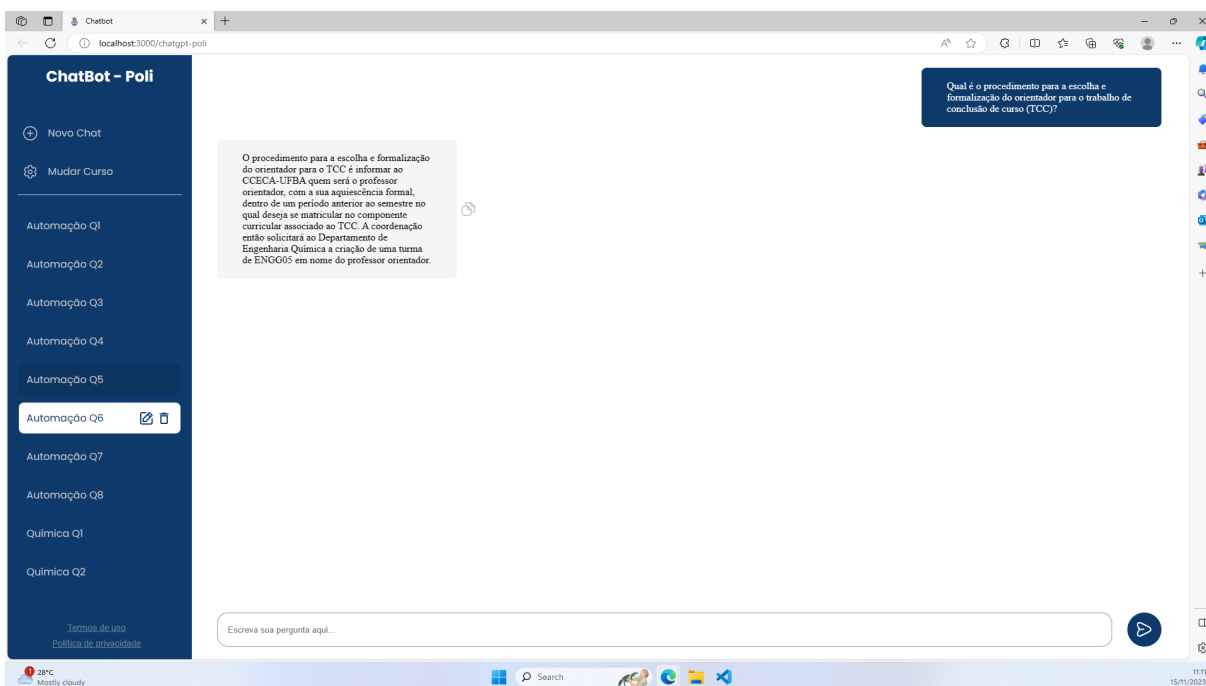


Figura 11 - Validação de respostas, pergunta 6. Fonte: Autor

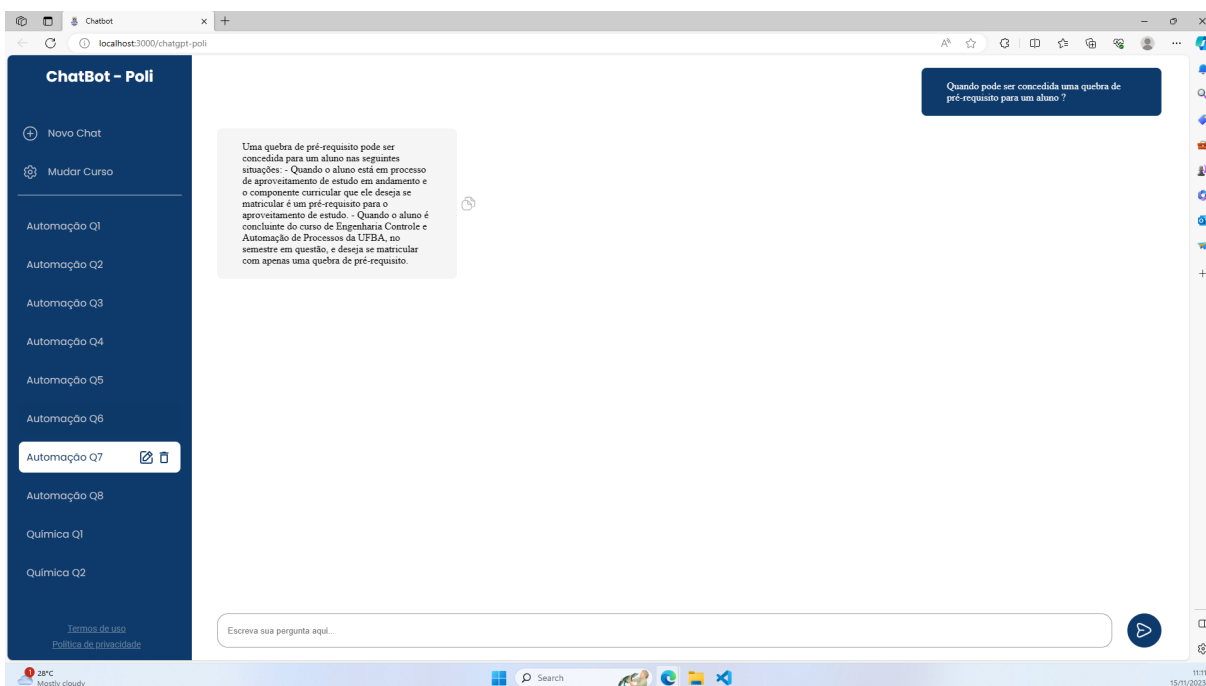


Figura 12 - Validação de respostas, pergunta 7. Fonte: Autor

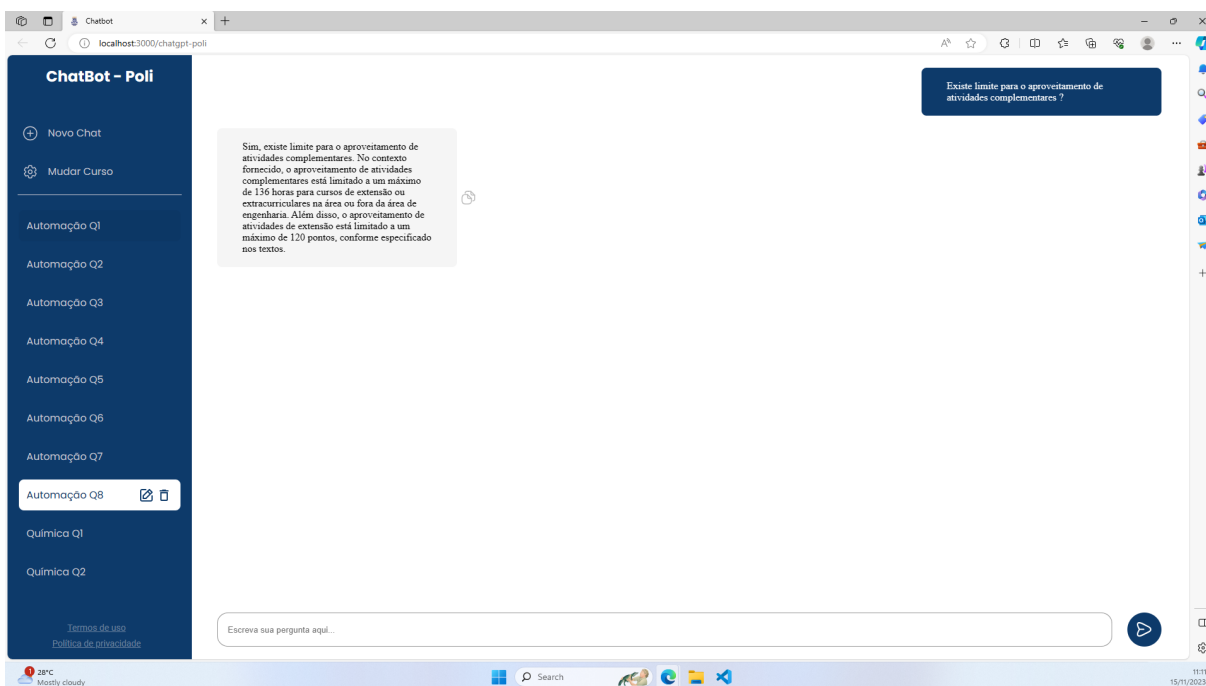


Figura 13 - Validação de respostas, pergunta 8. Fonte: Autor

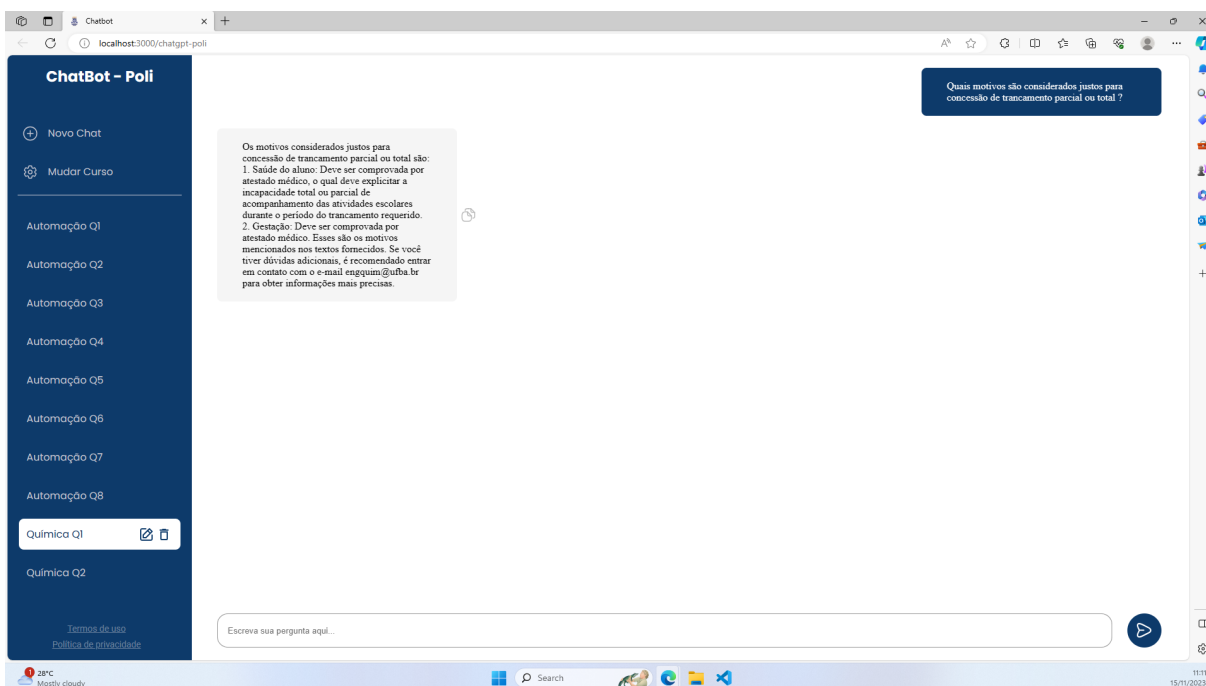


Figura 14 - Validação de respostas, pergunta 9. Fonte: Autor

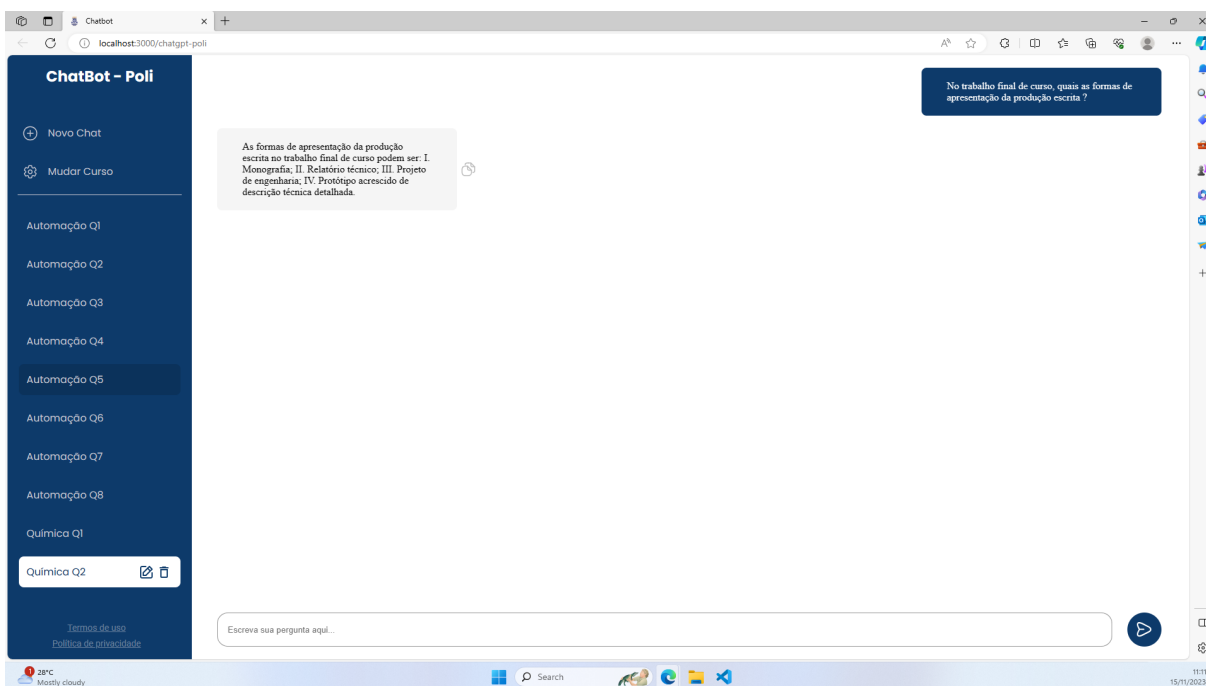


Figura 15 - Validação de respostas, pergunta 10. Fonte: Autor