Associação Propagadora Esdeva
Centro Universitário Academia – UniAcademia
Curso de Engenharia de Software
Trabalhode Conclusão de Curso _____

Papel da Inteligência Artificial na Educação: Desenvolvimento de um Chatbot para Programadores Iniciantes

Lucas Gustavo Mendes de Almeida[1]
Centro Universitário UniAcademia,JuizdeFora,MG
Romualdo Monteiro de Resende Costa[2]
Centro Universitário UniAcademia,Juizde Fora,MG

Linha de Pesquisa: Engenharia de Software

RESUMO

Este estudo objetivou explorar o desenvolvimento de um *chatbot* educacional voltado para o ensino de desenvolvimento de software, utilizando *a Generative Pre-trained Transformer* 3(API GPT-3.5 Turbo) da OpenAI. A inteligência artificial e a aprendizagem de máquina têm se tornado cada vez mais relevantes na educação, oferecendo oportunidades para melhorar a experiência de aprendizado em programação. O *chatbot* educacional é projetado para fornecer respostas contextuais e informativas a perguntas relacionadas à programação, auxiliando programadores iniciantes na busca por conhecimento e na resolução de dúvidas. Este trabalho detalha o processo de desenvolvimento do *chatbot,* bem como sua integração com a API GPT-3.5 Turbo. Devido ao foco no desenvolvimento das funcionalidades da aplicação e no estudo de modelos de integração com ferramentas de inteligência artificial, testes de usabilidade com possíveis usuários desta ferramenta estão fora do escopo deste trabalho. Como resultado final, é apresentada uma ferramenta que poderá, eventualmente, ser utilizada em conjunto com outras técnicas de aprendizado de programação.

Palavras-chave: Chatbot, chatbot na educação, Educação em programação, Openai, Python, Streamlit.

[1] Discente do Curso de Engenharia de Software do Centro Universitário UniAcademia.
[2] Docente do Curso de Engenharia de Software do Centro Universitário UniAcademia.

ABSTRACT

This study aims to explore the development of an educational chatbot focused on teaching software development, using the GPT-3.5 Turbo (Generative Pre-trained Transformer 3) API from OpenAI. Artificial intelligence and machine learning have become increasingly relevant in education, offering opportunities to enhance the programming learning experience. The educational chatbot is designed to provide contextual and informative responses to programming-related questions, assisting novice programmers in their quest for knowledge and problem-solving. This study details the chatbot development process as well as its integration with the GPT-3.5 Turbo API. Due to the focus on developing application functionalities and studying integration models with artificial intelligence tools, usability testing with potential users of this tool is beyond the scope of this study. As a final result, a tool is presented that may eventually be used in conjunction with other programming learning techniques.

Keywords: Chatbot, Programming, chatbot in education, Programming education, OpenAI, Python, Streamlit.

1. INTRODUÇÃO

Atualmente, a educação enfrenta desafios substanciais, entre eles a necessidade premente de tornar o ensino mais acessível e interativo. Dentro desse panorama desafiador, a Inteligência Artificial (IA) surge como uma ferramenta promissora, capaz de remodelar o futuro da aprendizagem (Silva, 2023). A capacidade da IA em analisar grandes volumes de dados, identificar padrões e adaptar o ensino às necessidades individuais dos alunos a torna uma ferramenta poderosa para personalizar o aprendizado e aumentar a eficácia do ensino.

Diante do cenário educacional atual, onde a demanda por habilidades digitais, como programação, está em constante crescimento, é crucial explorar como a IA pode ser aplicada para apoiar o ensino e aprendizado dessas habilidades. Por exemplo, pode-se criar um produto que, conforme "Chatbot: guia completo" (2023), são assistentes virtuais alimentados por inteligência artificial, que podem interagir com os usuários de forma natural, como se fossem humanos. Eles são programados para compreender perguntas, fornecer respostas relevantes e aprender com cada interação.

Nesse contexto a aplicação desenvolvida neste trabalho se destaca por oferecer funcionalidades abrangentes e interativas que vão além das soluções típicas de chatbots.

Diferente de outras ferramentas semelhantes, esta aplicação utiliza a API do ChatGPT de forma inovadora, incorporando um chat de áudio que permite aos usuários interagir por meio da fala, proporcionando uma comunicação mais dinâmica e acessível. Além disso, a aplicação inclui um questionário que gera perguntas automaticamente, oferecendo uma maneira eficaz para programadores testarem e aprimorarem seus conhecimentos.

Outra característica distintiva é a capacidade de análise de documentos PDF, onde o usuário pode interagir diretamente com o conteúdo do documento, facilitando a extração de informações e o esclarecimento de dúvidas de maneira prática e eficiente.

Essas funcionalidades tornam a aplicação uma ferramenta única, que se destaca pela versatilidade e pela capacidade de auxiliar os estudos de programação de múltiplas maneiras, transformando o chat em uma plataforma abrangente e interativa.

2. REFERENCIAL TEÓRICO

O referencial teórico deste trabalho foi fundamental para embasar a construção de um *chatbo*t alimentado pela Api da OpenAI.

Visando garantir a relevância dos trabalhos consultados foi utilizado bases de dados acadêmicas como google acadêmico[3], além de artigos, livros e relatórios técnicos.

O foco foi identificar autores que abordassem os temas de IA na educação, IA generativa, modelo GPT-3.5-Turbo e alguns sites para referência de ferramentas, como a API da OpenAI. Durante a pesquisa também foram analisados alguns trabalhos semelhantes, como de Wicks (2023), que propõe um chatbot para automatização de respostas para dúvidas acadêmicas utilizando a API da OpenAI.

Após a seleção, as fontes foram submetidas a uma leitura detalhada e análise crítica. Foi identificado os conceitos-chave, teorias e modelos mais relevantes para o tema. Por exemplo, a definição e potencial da IA na educação foram fundamentados em trabalhos de Spadini (2023) e Nguiraze (2023). Já a discussão sobre a IA generativa e o ChatGPT baseou-se também em Spadini (2023).

A organização do conteúdo foi dividida em 5 seções, são elas: (1) Inteligência artificial: Aborda o conceito de IA e sua aplicação na educação, (2) IA generativa: Explora

[3] Disponível em < https://scholar.google.com.br/?hl=pt

o conceito de IA generativa, (3) Modelo GPT-3.5-Turbo: Detalha as caracterizas desse modelo e seu impacto na geração de texto de alta qualidade, (4) Configuração e funcionamento da Api da OpenAI: Descreve o processo de integração Da API com a aplicação, (5) Custos: Relata os custos gerados ao utilizar a API da OpenAI.

2.1 Inteligência Artificial

A inteligência artificial é uma área da Ciência da Computação cujo objetivo é criar sistemas capazes de realizar tarefas que, até então, só poderiam ser executadas por seres humanos Spadini (2023). A utilização dessa na educação tem sido objeto de intensa investigação e aplicação, impulsionada pela busca por soluções inovadoras para melhorar o processo de ensino e aprendizagem (Nguiraze, 2023).

Nesse aspecto, a IA tem se mostrado promissora em diversos aspectos educacionais, desde a detecção de lacunas no processo de ensino e aprendizagem até a criação de ambientes de aprendizagem mais personalizados e eficazes (Nguiraze, 2023; Dores et al., 2021). Um dos principais desafios enfrentados na educação diz respeito à percepção e avaliação dos professores em relação a essas tecnologias (Parreira, Lehmann e Oliveira, 2021).

Embora haja reconhecimento do potencial da IA para auxiliar no ensino, muitos professores ainda expressam preocupações e incertezas quanto ao seu impacto na prática pedagógica e no papel do educador (Parreira, Lehmann e Oliveira, 2021).Contudo, é inegável que a pandemia de COVID-19 acelerou a adoção de soluções tecnológicas na educação, destacando o papel fundamental desse recurso no contexto do ensino híbrido e da virtualização escolar (Carius, 2021).

O surgimento de plataformas de ensino virtual e a necessidade de adaptação rápida às novas demandas educacionais evidenciaram a importância de ferramentas baseadas em IA, como os *chatbots*, na oferta de suporte e orientação aos alunos em ambientes virtuais de aprendizagem (Dores et al., 2021). Além disso, ferramentas tecnológicas têm sido aplicadas de maneira eficaz na personalização do ensino, permitindo a adaptação do conteúdo e das estratégias de ensino às necessidades individuais de cada aluno (Silveira e Vieira Júnior, 2019). Essa abordagem personalizada contribui para o engajamento dos estudantes e para a melhoria do desempenho

acadêmico, uma vez que permite que o aprendizado seja mais relevante e significativo para cada indivíduo (Silveira e Vieira Júnior, 2019).

A evolução da IA na educação representa uma oportunidade significativa para transformar a maneira como o conhecimento é transmitido e adquirido. Apesar dos desafios e das preocupações associadas à sua implementação, a IA oferece um vasto potencial para melhorar a eficácia e a qualidade do ensino, tornando-o mais acessível, personalizado e adaptado às necessidades dos alunos (Nguiraze, 2023; Dores et al., 2021).

2.2 IA Generativa

As IA generativas possuem a capacidade de criar novas informações a partir de conjuntos de dados pré-existentes. Essas tecnologias evoluem por meio de grandes bases de dados, visando adquirir padrões de construção desses dados. Com essa compreensão, tornam-se aptas a gerar novos dados, semelhantes aos utilizados para seu treinamento, mas que podem ser únicos e originais (Spadini, 2023).

Nesse contexto, surgem propostas inovadoras que buscam explorar o potencial desse recurso na função generativa para aprimorar os processos de ensino e aprendizagem. Entre essas propostas, destaca-se o desenvolvimento da plataforma OpenAI, cujo objetivo principal é criar e promover soluções baseadas em inteligência artificial para diversos campos, incluindo a educação (Spadini, 2023).

Um dos principais produtos desenvolvidos é o ChatGPT, um modelo de linguagem conversacional baseado em aprendizado de máquina que utiliza a técnica de *transformers* para gerar respostas coerentes e contextualmente relevantes em diálogos com usuários. Esse modelo é amplamente utilizado em uma variedade de aplicações educacionais, incluindo sistemas de tutoria em ambientes virtuais de aprendizagem (AVA), conforme demonstrado por Dores et al. (2021)

Esse tipo de tecnologia permite a criação de diversos conteúdos únicos, como imagens, redações, traduções e até mesmo chats educacionais capazes de ensinar diversos assuntos. Este último é o foco principal deste projeto.

2.3 Modelo GPT-3.5-turbo

Especificamente, o modelo GPT-3.5-Turbo constitui uma versão aprimorada da ferramenta GPT-3, desenvolvida pela OpenAI. Este modelo representa um marco significativo no campo da inteligência artificial devido à sua capacidade de gerar textos de qualidade próxima à humana em uma variedade de tarefas e contextos (OpenAI, 2024). De fato, essa edição é considerada um grande avanço em comparação aos seus anteriores e modelos concorrentes, visto, que sua linguagem revolucionou a forma como as máquinas entendem e geram texto, e a sua capacidade de adaptação a tarefas específicas através de ajustes finos que em um processo que um modelo de IA existente é modificado para torná-lo mais adequado para uma tarefa ou domínio específico (OpenAI, 2024).

Uma das principais vantagens dessa versão é a sua maior qualidade e consistência em relação ao seu antecessor (OpenAI, 2024).

Ao contrário do modelo anterior, a nova versão foi ajustada para oferecer desempenho mais preciso e confiável em diversas tarefas. Isso se traduz em melhor qualidade dos resultados gerados e em uma experiência mais satisfatória para os usuários. Outra vantagem notável do modelo é a redução significativa no custo do ajuste fino (OpenAI, 2024).

Na área de programação, o GPT-3.5 Turbo oferece suporte de código de alta qualidade. Os desenvolvedores podem obter sugestões e gerar código automatizado com base nas suas instruções. Isso acelera o processo de desenvolvimento, reduz erros e facilita a criação de software complexo. A assistência ao código também pode melhorar a depuração, ajudando a identificar e corrigir erros com mais eficiência (OpenAI, 2024).

2.4 Configuração e Funcionamento da Api OpenAI

Para a integração com a interface de programação desenvolvida pela OpenAI, a obtenção da chave de autenticação é um passo crucial, pois permite validar as requisições. Com a chave em mãos, é possível configurar o cliente responsável por enviar as mensagens do usuário e receber as respostas geradas pelos modelos de linguagem. O fluxo de funcionamento de um *chatbot* que utiliza essa interface segue um processo

específico. Quando o usuário deseja interagir com o *chatbot*, insere sua mensagem na interface do cliente desenvolvida neste trabalho. A interface captura a mensagem e a transforma em uma requisição, que será enviada para a API (OpenAI, 2024), tudo isto conforme apresentado na figura 1.

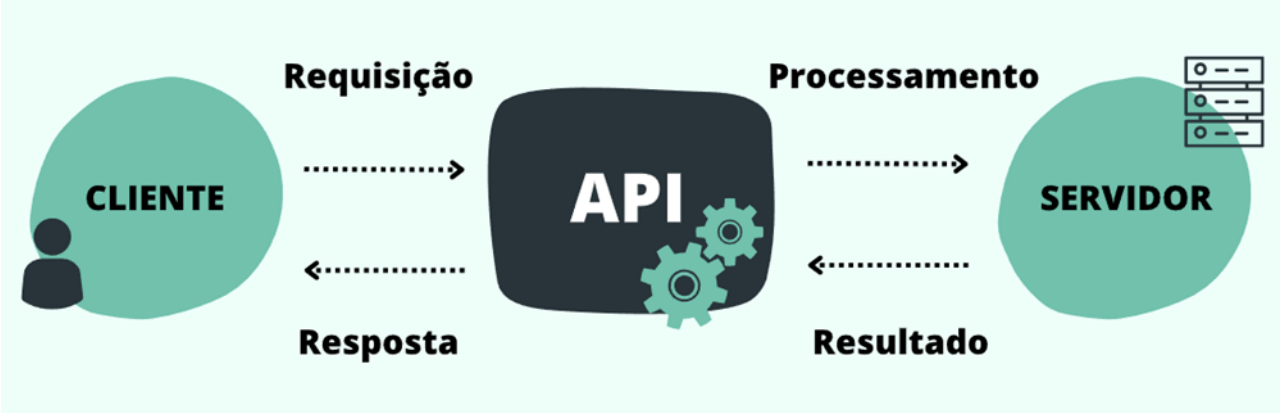

Figura 1: Fluxo de funcionamento da Api OpenAI
Fonte: https://pluga.co/blog/api-chatgpt/

Na figura 1, a requisição é uma chamada HTTP4, geralmente do tipo POST5, contendo os dados da mensagem do usuário formatados em JSON6. Essa chamada é enviada para a API, que atua como intermediária no processo de comunicação. Ao receber a requisição do cliente, essa é redirecionada para o servidor onde o modelo de linguagem está hospedado. No servidor, a mensagem do usuário é processada pelo modelo, que interpreta o conteúdo e gera uma resposta adequada ao contexto fornecido. Após o processamento, a resposta gerada é enviada de volta ao cliente, que a exibe ao usuário, completando o ciclo de interação (OpenAI, 2024).

A chave de autenticação permite controlar o acesso aos recursos, garantindo que apenas usuários autorizados possam enviar requisições e receber respostas. O processo de configuração do cliente envolve a definição de parâmetros que orientam a comunicação entre o cliente e a API. Esses parâmetros incluem o *endpoint*, o método

[4] HTTP é um protocolo de transferência

[5] Metodo HTTP utilizado para enviar dados para o servidor

[6] Formato utilizado para estruturar dados em formato de texto e permitir a troca de dados entre aplicações de forma simples

HTTP utilizado, e os cabeçalhos necessários para a autenticação e formatação das requisições. Durante o desenvolvimento do chatbot, ajustes podem ser feitos para otimizar o desempenho e garantir que as respostas geradas sejam relevantes e precisas (OpenAI, 2024).

O uso de uma chave também permite monitorar o uso do serviço, ajudando a identificar padrões de utilização e possíveis problemas. Esse monitoramento é essencial para manter a qualidade do serviço, pois permite que a OpenAI implemente melhorias contínuas e corrija eventuais falhas. O processo de envio de requisições e recebimento de respostas é repetido a cada interação do usuário com o *chatbot*, criando um ciclo dinâmico de comunicação que se adapta às necessidades e contextos apresentados pelos usuários (OpenAI, 2024).

2.5 Custo

Conforme dados divulgados pela OpenAI (2024), os valores para consumo do modelo de chat GPT-3.5 Turbo são de $0,50 de dólar por 1 milhão de tokens, conforme apresentado na Figura 3. Um token pode variar de vários caracteres a uma palavra. Em média, 1 milhão de tokens equivalem a aproximadamente 750.000 palavras.

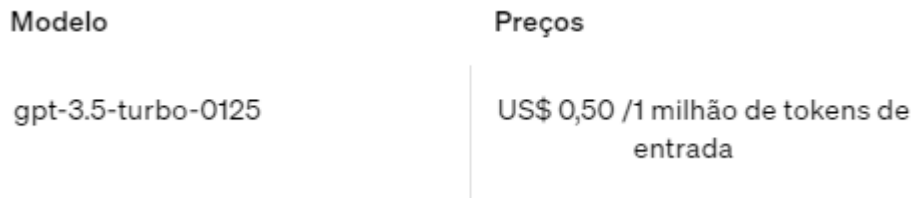

Figura 2: Custos OpenAI
Fonte: https://openai.com/pricing

O custo da utilização da inteligência artificial no contexto educacional tem sido objeto de considerável discussão entre pesquisadores e profissionais da área. Parreira, Lehmann e Oliveira (2021) investigam a percepção e avaliação dos professores em relação às tecnologias de IA na educação, destacando os desafios enfrentados e as oportunidades potenciais que essas tecnologias oferecem. A incorporação da inteligência artificial no ambiente educacional também pode ser vista em iniciativas como a proposta de Dores et al. (2021) de utilizar *chatbots* como sistemas de tutoria em AVA, visando aprimorar a experiência de aprendizagem dos alunos.

Contudo, é fundamental reconhecer que a adoção da inteligência artificial na educação não está isenta de desafios, incluindo questões relacionadas ao custo. A implementação e manutenção de sistemas de IA exigem investimentos significativos em infraestrutura, desenvolvimento de software e capacitação de pessoal. Além disso, há custos associados à aquisição de dados e algoritmos de qualidade, bem como à garantia da privacidade e segurança dos dados dos alunos (Nguiraze, 2023). No entanto, apesar dos desafios financeiros, os benefícios potenciais da IA na educação são amplamente reconhecidos.

Portanto, é crucial que instituições educacionais e formuladores de políticas considerem cuidadosamente o equilíbrio entre os custos e benefícios da integração da IA no ensino e aprendizagem (Silva e Vieira Júnior, 2019). A utilização dessa ferramenta deve ser avaliada por cada instituição, que deve verificar se os valores e os benefícios são adequados ao seu momento.

3. APLICAÇÃO

Nesta seção, são apresentadas as principais funcionalidades do sistema, focando na aplicação de filtros para otimizar interações e consultas.

Para cada funcionalidade, descritas na Tabela 1 como o chat padrão, conversa por voz, envio e análise de arquivos, e geração de questionários, foram estabelecidos filtros específicos para garantir que apenas perguntas e interações relevantes dentro do campo da tecnologia da informação sejam respondidas ou processadas pelo sistema.

FUNCIONALIDADE	DESCRIÇÃO
Chat padrão	Permite conversar e acessar perguntas relacionadas à área de TI.
Pergunta por áudio	Permite fazer perguntas utilizando áudio como entrada. O sistema transcreve o áudio em texto para processamento.
Envio e análise de arquivos	Os usuários podem enviar arquivos relacionados a questões de TI para análise ou fazer perguntas específicas sobre o conteúdo do arquivo enviado.
Geração automática de questionário	Utilizando a tecnologia da OpenAI, o sistema gera automaticamente questionários sobre diversos tópicos de TI, proporcionando uma forma interativa de aprendizado.

Tabela 1: Funcionalidades da inteligência artificial na educação e suas descrições
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

A primeira funcionalidade é o chat padrão, que permite aos usuários iniciar conversas e formular perguntas relacionadas à tecnologia da informação. Essa funcionalidade serve como base para interações gerais e consultas sobre diversos tópicos dentro do campo da TI

A Figura 3 ilustra diversas telas com exemplo dessa funcionalidade.

Figura 3: Tela do chat e funcionamento do filtro
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Na Figura 3, a interface do chat padrão é mostrada, destacando como as conversas são exibidas e como o chat inicia com uma pergunta padrão do assistente. O filtro redireciona perguntas fora do tópico de ensino para o assunto correto.

A segunda funcionalidade é o chat por voz, permitindo conversas sobre tecnologia da informação via fala, com o aplicativo convertendo fala em texto. A Figura 4 mostra a interface desta funcionalidade, usando o mesmo sistema de filtro, mas com o microfone na parte superior e as mensagens na parte inferior, garantindo que a última mensagem seja sempre visível.superior da tela.

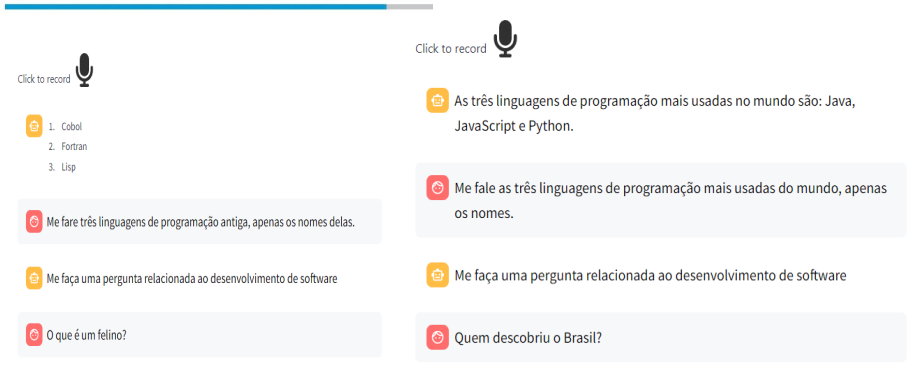

Figura 4: Utilizando chat por fala.
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

A terceira funcionalidade permite o envio e análise de arquivos, permitindo aos usuários compartilhar documentos relevantes para suas perguntas. Podem ser feitas perguntas específicas sobre o conteúdo desses arquivos, proporcionando uma interação detalhada e contextualizada.

A Figura 5 mostra a interface dessa funcionalidade, com o local de upload na parte superior. Após o upload, o conteúdo do arquivo é exibido, permitindo a troca de mensagens entre o assistente e o usuário sobre o conteúdo submetido.

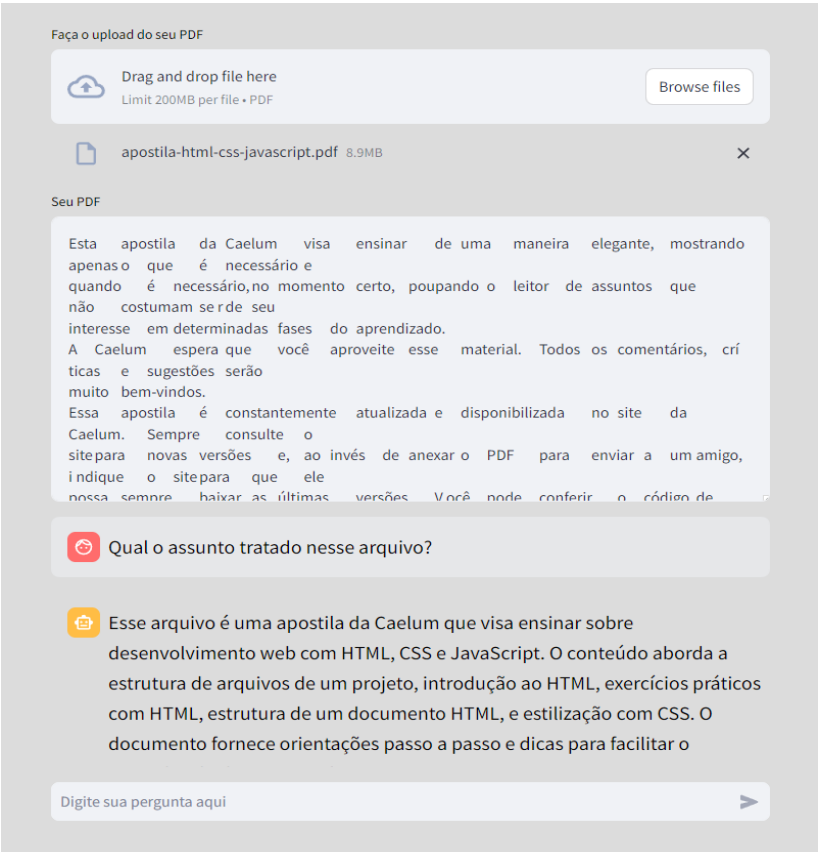

Figura 5: Envio de arquivos

Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

A ultima funcionalidade é mostrada na Figura 6, sua interface é baseada em uma sidebar com alternativas para tipos de questionários, depois que o usuario seleciona a opção desejada, a interface muda diretamente para perguntas geradas automaticamente pela openai, o usuario marca a resposta que ele considera verdadeira e depois clica em enviar resposta, logo apos o clique, será gerado uma nova pergunta, formando assim um quiz de 5 perguntas, se o usuario acertar 3 questões no minimo, ele receberá uma mensagm de sucesso, caso a nota seja menor que 3 receberá uma mensagem de fracasso.

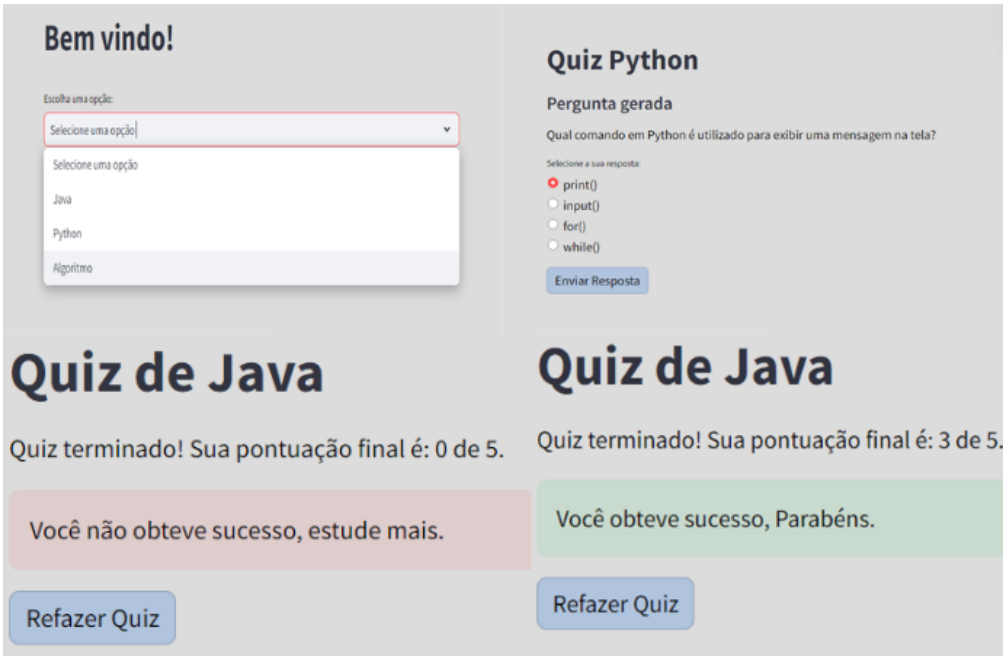

Figura 6: Software automatizado para o uso de questionários
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

4. METODOLOGIA

A Figura 7 apresenta os processos que envolveram a metodologia de desenvolvimento do *chatbot* alimentado pela API da OpenAI, destacando de forma abrangente as principais etapas envolvidas. Inicialmente, foi discutida a definição dos objetivos do projeto, seguida pela escolha da tecnologia mais apropriada. Em seguida, foi explorada a configuração da OpenAI para integração ao sistema, culminando na

implementação do *chatbot*. Cada uma dessas etapas foi detalhada posteriormente, demonstrando processo completo de desenvolvimento.

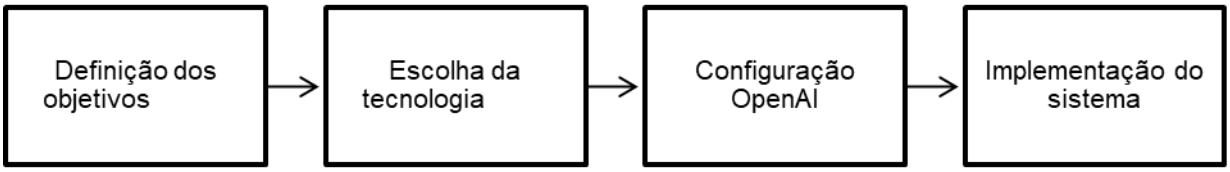

Figura 7: Diagrama de fluxo de trabalho
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

4.1 Procedimentos

A definição dos objetivos, incluíram fornecer suporte a alunos iniciantes em programação, facilitando seu aprendizado por meio de uma ferramenta cada vez mais presente em nossa sociedade. Outro objetivo foi demonstrar o poder da OpenAI e como ela é capaz de simplificar a criação de aplicações que antes seriam muito complexas.

No levantamento e escolha das tecnologias, optou-se pelas linguagens Python[7] e Streamlit[8]. A primeira possui uma sintaxe básica semelhante ao inglês, o que facilita a leitura e entendimento dos códigos pelos desenvolvedores e uma comunidade ativa com milhões de desenvolvedores de suporte em todo o mundo, conforme descrito em “aws: o que é python?” (2023).

A segunda permite a criação rápida de interfaces Web interativas, sendo ideal para prototipagem e desenvolvimento ágil. Suas funcionalidades permitem especificar uma atualização contínua da aplicação sem a necessidade de recarregar a página.

A configuração da API foi tratada na seção 2.4 de forma completa, por isso o foco a partir desse ponto será na implementação.

Na implementação o chat padrão foi o primeiro a ser desenvolvido. O processo inicia com a verificação e inicialização do histórico de mensagens, que é armazenado no estado da sessão através de *st.session_state*. Caso não exista uma chave "*messages*"

[7] Linguagem de programação amplamente usada em aplicações da Web, desenvolvimento de software, ciência de dados e machine learning. Disponível em <https://python.org>

[8] Ferramenta que transforma scripts python em aplicativos web com poucas linhas de código. Disponível em <https://streamlit.io>

nesse estado, ela é criada com uma mensagem inicial do assistente, estabelecendo o ponto de partida da interação, como mostrado nas linhas 97 e 98 da figura 8. Em seguida, um laço percorre cada mensagem no histórico, conforme apresentado nas linhas 101 e 102 da mesma figura, para exibir seu conteúdo. Esse método garante que cada mensagem seja exibida conforme o papel de quem a enviou, diferenciando entre mensagens do usuário e do assistente.

```
96  # Inicialização do histórico de mensagens no estado da sessão
97  if "messages" not in st.session_state:
98      st.session_state["messages"] = [{"role": "assistant", "content": "Como posso lhe ajudar com desenvolvimento de software?"}]
99
100 # Exibição das mensagens do histórico
101 for msg in st.session_state.messages:
102     st.chat_message(msg["role"]).write(msg["content"])
```

Figura 8: Código inicialização de mensagens
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Após a inicialização, o sistema pode receber novas entradas do usuário. O código dessa funcionalidade é mostrado na Figura 9, linhas 105 a 107. Cada texto informado é adicionado ao histórico de mensagens (linhas 110 e 111), permitindo um registro contínuo das conversas para fornecer contexto às respostas. A mensagem do usuário é exibida imediatamente, proporcionando feedback visual instantâneo.

```
104 # Captura da entrada do usuário
105 if prompt := st.chat_input("Digite sua mensagem aqui..."):
106     if not openai_api_key:
107         st.stop()  # Interrompe a execução se a chave da API não estiver definida
109     # Adiciona a mensagem do usuário ao histórico
110     st.session_state.messages.append({"role": "user", "content": prompt})
111     st.chat_message("user").write(prompt)
112
```

Figura 9: Entrada do usuário e adição ao histórico de mensagens
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

É criada uma mensagem de controle que informa ao assistente sua especialização e limita as perguntas ao desenvolvimento de software. A mensagem do usuário é concatenada com esta mensagem de controle e adicionada ao histórico a ser enviado para a API da OpenAI (messages_to_send), como mostrado na Figura 10.

```
113     # Criação da mensagem de controle para envio à API
114     chat = (
115         "Eu sou um assistente especializado em desenvolvimento de software. "
116         "Eu respondo a perguntas sobre programação, sistemas de informação, engenharia de software, design de software,"
117         " arquitetura de software, "
118         "testes de software, metodologias ágeis, DevOps, segurança de software, bancos de dados, integração de sistemas, "
119         "gerenciamento de projetos de software e qualquer outro tópico relacionado ao desenvolvimento de software. "
120         "Por favor, me faça perguntas apenas sobre esses tópicos. Se sua pergunta não estiver relacionada a esses tópicos, "
121         "eu responderei com: 'Me faça uma pergunta relacionada ao desenvolvimento de software.'"
122     )
123     messages_to_send = st.session_state.messages + [{"role": "user", "content": chat + "\n" + prompt}]
124
```

Figura 10: Criação do filtro de perguntas
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Uma vez fornecida capturada a mensagem do usuário e configurado o filtro, uma parte crucial do processo é a solicitação à API. A figura 11, nas linhas 126 a 129 mostra a utilização do comando da API openai.ChatCompletion.create, que envia a mensagem para o modelo GPT-3.5 Turbo e solicita uma nova resposta.

```
125     # Faz a solicitação à API da OpenAI para gerar uma resposta
126     response = openai.ChatCompletion.create(
127         model="gpt-3.5-turbo",
128         messages=messages_to_send
129     )
```

Figura 11: Criação do filtro de perguntas
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Feito isso a resposta gerada pelo assistente é adicionada ao histórico de mensagens sob o papel de “assistant”, e então exibida ao usuário como mostrado na figura 12. Esse ciclose repete para cada nova entrada, mantendo o fluxo de interação continuo.

```
134     # Adiciona a resposta do assistente ao histórico de mensagens
135     st.session_state.messages.append({"role": "assistant", "content": msg})
136     st.chat_message("assistant").write(msg)
```

Figura 12: Adiciona a resposta do assistente ao histórico
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024)

No chat por voz, algumas das mesmas técnicas foram usadas, como o filtro de perguntas e a exibição das mensagens. No entanto, foi necessário adicionar outras

especificidades, como um gravador de áudio. Esse gravador é chamado pela função *audio_recorder*(), que captura os bytes do áudio gravado e, logo depois, verifica se há bytes capturados. Se houver, os bytes são salvos em um arquivo WAV. Em seguida, o modelo Whisper[9] da OpenAI é carregado e utilizado para transcrever o áudio, armazenando o texto transcrito na variável text conforme mostrado nas linhas 132 até 144 da Figura 13.

```
131    # Inicializando o gravador de áudio
132    audio_bytes = audio_recorder()
133
134    if audio_bytes:
135        # Salvar os bytes do áudio em um arquivo
136        with open("audio_gravado.wav", "wb") as f:
137            f.write(audio_bytes)
138
139        # Carregar o modelo Whisper
140        modelo = whisper.load_model("base")
141
142        # Transcrever o áudio
143        resposta = modelo.transcribe("audio_gravado.wav")
144        text = resposta['text']
```

Figura 13: Captura e transcrição de áudio
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Por sua vez, na análise da Figura 14, linhas 81 a 86, é mostrada a análise de um arquivo PDF. Primeiramente, foram definidas duas funções principais: ler_pdf(arquivo) e perguntar (conteudo, pergunta, chave_api). A função ler_pdf(arquivo) é responsável por extrair o texto de um arquivo PDF. Utiliza a biblioteca PyPDF2[10] para ler o conteúdo do PDF e, em seguida, iterar sobre cada página, extraindo o texto de cada uma delas. O texto é então concatenado em uma única string e retornado pela função.

[9] Modelo de conversão de fala, usado para transcrever arquivos de áudio. Disponível em < https://openai.com/index/whisper/>

[10] Disponível em < https://pypi.org/project/PyPDF2/>

```python
def ler_pdf(arquivo):
    leitor_pdf = PyPDF2.PdfReader(arquivo)
    texto = ''
    for pagina in leitor_pdf.pages:
        texto += pagina.extract_text()
    return texto
```

Figura 14: Função para ler o arquivo
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

A função "perguntar (conteudo, pergunta, chave_api)" está na Figura 15. A linha 89 especifica a chave da API, a linha 92 define o tamanho máximo do arquivo e, nas linhas 95 a 98, as mensagens são preparadas para a API. Na linha 97, a variável "conteudo truncado" contém uma versão reduzida do texto para não exceder o limite da OpenAI. Nas linhas 101 a 106, a requisição é feita para a API. Na linha 108, o conteúdo da resposta é extraído e os espaços em branco são removidos; na linha 109, a resposta final é apresentada.

```python
def perguntar(conteudo, pergunta, chave_api):
    openai.api_key = chave_api

    # Limita o conteúdo do PDF a ser enviado para a API
    tamanho_maximo_conteudo = 2000  # Ajuste conforme necessário
    conteudo_truncado = conteudo[:tamanho_maximo_conteudo]

    mensagens = [
        {"role": "system", "content": "Você é um assistente útil."},
        {"role": "user", "content": f"O seguinte é o conteúdo de um documento:\n\n{conteudo_truncado}"},
        {"role": "user", "content": pergunta},
    ]

    resposta = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=mensagens,
        max_tokens=150,
        temperature=0.7,
    )

    resposta_final = resposta.choices[0].message['content'].strip()
    return resposta_final
```

Figura 15: Função perguntar
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Um ponto central desse estudo, se estabelece na análise do desenvolvimento do quiz, foram definidas variáveis no estado da sessão utilizando st.session_state. Esse objeto permite armazenar variáveis que persistem entre diferentes execuções do aplicativo. As variáveis incluem listas para perguntas, alternativas, respostas corretas, a pontuação do usuário, o número de perguntas respondidas, a resposta selecionada pelo usuário e uma flag para indicar se a resposta correta deve ser mostrada. Se alguma

dessas variáveis não existir no estado da sessão, ela é inicializada com um valor padrão. Como mostra a Figura 16, da linha 86 até 99.

```
85      # Inicializando variáveis no estado da sessão
86      if 'perguntas' not in st.session_state:
87          st.session_state['perguntas'] = []
88      if 'alternativas' not in st.session_state:
89          st.session_state['alternativas'] = []
90      if 'respostas' not in st.session_state:
91          st.session_state['respostas'] = []
92      if 'pontuacao' not in st.session_state:
93          st.session_state['pontuacao'] = 0
94      if 'perguntas_respondidas' not in st.session_state:
95          st.session_state['perguntas_respondidas'] = 0
96      if 'resposta_selecionada' not in st.session_state:
97          st.session_state['resposta_selecionada'] = None
98      if 'mostrar_resposta' not in st.session_state:
99          st.session_state['mostrar_resposta'] = False
```

Figura 16: Inicialização de variáveis no estado de sessão
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Contudo, para gerar uma nova pergunta, a função gerar_pergunta é chamada. Essa função cria uma pergunta específica sobre o assunto escolhido pelo usuário, com quatro alternativas e uma resposta correta, enviando essa solicitação para a OpenAI através da função enviar_pergunta_para_openai, a ser visto na Figura 17, linhas 113 até 116.

```
112     # Função para gerar uma nova pergunta
113     def gerar_pergunta():
114         pergunta = ("Crie uma pergunta sobre Java com 4 alternativas, identificando qual é a correta. Formato: "
115         "Pergunta: ... A: ... B: ... C: ... D: ... Resposta: ...")
116         pergunta_gerada = enviar_pergunta_para_openai(pergunta)
117
118             # Extrair pergunta, alternativas e resposta correta da string gerada
119             linhas = pergunta_gerada.split('\n')
120             pergunta_texto = ""
121             alternativa_a = ""
122             alternativa_b = ""
123             alternativa_c = ""
124             alternativa_d = ""
125             resposta_correta = ""
126
```

Figura 17: Função gerar pergunta
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Logo depois a resposta gerada é processada para extrair a pergunta, as alternativas e a resposta correta. Esse processamento envolve dividir a resposta em

linhas e analisar cada linha para identificar a pergunta e as alternativas, que são então armazenadas nas listas correspondentes no estado da sessão.

O controle do fluxo do quiz permite que o usuário selecione uma resposta utilizando botões de rádio (st.radio). Se o usuário ainda não enviou uma resposta, ele pode selecionar uma alternativa e clicar em um botão para enviar sua escolha. Quando o botão é clicado, a resposta selecionada é comparada com a correta. Se a resposta estiver correta, a pontuação do usuário é incrementada. Essa constatação é verificada através do código presente na Figura 18:

```
127         for linha in linhas:
128             if linha.startswith("Pergunta:"):
129                 pergunta_texto = linha.replace("Pergunta: ", "").strip()
130             elif linha.startswith("A:"):
131                 alternativa_a = linha.replace("A: ", "").strip()
132             elif linha.startswith("B:"):
133                 alternativa_b = linha.replace("B: ", "").strip()
134             elif linha.startswith("C:"):
135                 alternativa_c = linha.replace("C: ", "").strip()
136             elif linha.startswith("D:"):
137                 alternativa_d = linha.replace("D: ", "").strip()
138             elif linha.startswith("Resposta:"):
139                 resposta_correta = linha.replace("Resposta: ", "").strip()
140
141     # Adicionar a pergunta gerada e as alternativas à lista de perguntas
142     st.session_state['perguntas'].append(pergunta_texto)
143     st.session_state['alternativas'].append([alternativa_a, alternativa_b, alternativa_c, alternativa_d])
144     st.session_state['respostas'].append(resposta_correta)
145
```

Figura 18: Divisão da resposta em linhas
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

Analisando a Figura 19 percebe-se que na linha 151 se obtém o índice da pergunta atual com base no número de perguntas respondidas, logo depois nas linhas 152 a 156 é verificado se o índice atual é menor que o total de perguntas disponíveis. Caso a verificação seja verdadeira a pergunta atual é exibida. Da linha 157 até a 163 é criado um grupo de botões do tipo radio com as alternativas da pergunta.

A partir da linha 164 até a 170 é verificado se o botão enviar resposta foi ativado, quando isso acontece a resposta selecionada é armazenada no session_state, logo depois é obtido a resposta correta da pergunta atual, quando a resposta correta for igual a resposta selecionada, o usuário ganhara um ponto, em seguida será mostrada a resposta correta na tela e por fim será forçada a atualização da interface.

```
    # Exibir a pergunta atual
    pergunta_atual_index = st.session_state['perguntas_respondidas']
    if pergunta_atual_index < len(st.session_state['perguntas']):
        st.subheader("Pergunta gerada")
        st.write(st.session_state['perguntas'][pergunta_atual_index])
        alternativas = st.session_state['alternativas'][pergunta_atual_index]

        if not st.session_state['mostrar_resposta']:
            resposta_selecionada = st.radio(
                "Selecione a sua resposta:",
                alternativas,
                index=0,
                key=f"radio_{pergunta_atual_index}"
            )
            if st.button("Enviar Resposta"):
                st.session_state['resposta_selecionada'] = resposta_selecionada
                resposta_correta = st.session_state['respostas'][pergunta_atual_index]
                if resposta_selecionada.strip() == resposta_correta.strip():
                    st.session_state['pontuacao'] += 1
                st.session_state['mostrar_resposta'] = True
                st.experimental_rerun()
```

Figura 19: Exibição e Verificação de Perguntas e Respostas no Streamlit
Fonte: Elaborada pelo Autor (2024) a partir de Dados da Pesquisa (2024).

5. CONSIDERAÇÕES FINAIS

O projeto fez uso da API da OpenAI para simplificar seu desenvolvimento, evitando a necessidade de criar um modelo complexo como o GPT-3.5 Turbo, o que seria um processo extremamente complicado e demorado.

Com base nisso, o desenvolvimento deste projeto visou demonstrar, ainda que de forma básica, o poder da inteligência artificial e seu impacto na redefinição do conceito de educação. Foi ilustrado como a utilização da API da OpenAI facilita a criação de chatbots educacionais, quebrando as barreiras tecnológicas e tornando o acesso à educação mais acessível e inovador.

Futuramente, estão previstas diversas atualizações para aprimorar o sistema e possibilitar sua comercialização. Entre as melhorias planejadas estão a criação de telas de cadastro e login, a implementação de um banco de dados para armazenar informações dos usuários e o aprimoramento do quiz, transformando-o em um jogo competitivo entre os alunos, tornando a aplicação cada vez mais interessante.

Além disso, será incorporado um filtro dinâmico que permitirá aos professores ajustarem o filtro para direcionar o chat a falar sobre assuntos específicos, facilitando o foco em tópicos de interesse ou de estudo prioritário. Essas atualizações visam tornar a

aplicação mais robusta, personalizada e engajante para os usuários.O código-fonte do projeto está disponível para download em https://github.com/Lucasgma/TCC.

REFERÊNCIAS

AWS**. O que é python?** [S. l.], 2023. Disponível em: https://aws.amazon.com/pt/what-is/python/. Acesso em: 17 jun. 2024.

CARIUS, A. C. COVID-19 post pandemic, blended learning and artificial intelligence: Is it the school virtualization? **Research, Society and Development**, v. 10, n. 7, p. e55610716834, 2021. DOI: 10.33448/rsd-v10i7.16834. Disponível em: https://rsdjournal.org/index.php/rsd/article/view/16834. Acesso em: 13 jun. 2024.

Chatbot. **Chatbot: guia completo**. [S. l.], 2021. Disponível em: https://getbots.com.br/blog/guia-completo-chatbot/. Acesso em: 15 jun. 2024.

OPENAI. OpenAI API. Disponível em: https://www.openai.com/api/. Acesso em: 24 jun 2024.

Chatbot. **Chatbot: tudo sobre o que é, como funciona e benefícios + cases**. [S. l.], 2021. Disponível em: https://www.blip.ai/blog/chatbots/chatbot/. Acesso em: 12 jun. 2024.

DORES, A. R. das et al. Aplicação da IA na educação: proposta de um projeto ou utilização de chatbot como sistema de tutorial aplicado em um AVA. **RevistaInovaEduc**, n. 7, p. 1–16, 2021.

WICKS, Gerald Hamilton. CHATBOT-POLI: uma proposta de MVP para automatização de respostas para dúvidas acadêmicas com OpenAI. 2023. 42 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de controle e Automação) – Escola Politécnica, Universidade da Bahia, Bahia, 2023.

NGUIRAZE, J. A. O papel da inteligência artificial na detecção de lacunas no processo de ensino e aprendizagem. **RevistaMultidisciplinar do NordesteMineiro**, v. 8, n. 1, p. 1-14, 2023.

PARREIRA, A.; LEHMANN, L.; OLIVEIRA, M. The challenge of artificial intelligence technologies in education: teachers' perception and evaluation. **Ensaio: Aval. Pol. Públ. Educ.**, v. 29, n. 113, [S.p.], out./dez 2021. Disponível em: https://doi.org/10.1590/S0104-40362020002803115. Acesso em: 11 jun. 2024.

PLUGA. **Chat GPT**. [S. l.], 2024. Disponível em: https://pluga.co/blog/api-chatgpt/. Acesso em: 06 jun. 2024.

SILVA, H. P. R. et al. Desenvolvimento de Jogos 2D de Plataforma: Explorando Unity e Chat GPT para Criação de Códigos Dinâmicos. **Sociedade Brasileira de Computação**, 2023.

SILVEIRA, A.; VIEIRA JÚNIOR, N. A inteligência artificial na educação: utilizações e possibilidades. **Revista Interritórios**

, v. 5, n. 8, p. 206-217, 2019.

SPADINI, A. S. **O que é IA generativa?** GPT, ChatGPT e Midjourney. [S. l.], 2023. Disponível em: https://www.alura.com.br/artigos/inteligencia-artificial-ia-generativa-chatgpt-gpt-midjourney. Acesso em: 10 jun. 2024.

SUNAGA, A. Ensino híbrido - diretrizes para planos de aula de qualidade. (2023) [S. l.: s. d.; s. n.], Disponível em: https://issuu.com/alesunaga/docs/ebook_ensino_h_brido_-_diretrizes_p. Acesso em: 09 jun. 2024. Ebook.