INSTITUTO FEDERAL DE SANTA CATARINA

KLEITON CARLOS DE SOUZA

# Implementação de um Chatbot com GPT-3.5 para Melhoria do Atendimento ao Cliente em Telecomunicações

São José - SC

agosto/2023

# IMPLEMENTAÇÃO DE UM CHATBOT COM GPT-3.5 PARA MELHORIA DO ATENDIMENTO AO CLIENTE EM TELECOMUNICAÇÕES

Trabalho de conclusão de curso apresentado à Coordenadoria do Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Prof. Mario de Noronha Neto, Dr.

São José - SC

agosto/2023

Kleiton Carlos de Souza

Implementação de um Chatbot com GPT-3.5 para Melhoria do Atendimento ao Cliente em Telecomunicações/ Kleiton Carlos de Souza. – São José - SC, agosto/2023-

44 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Mario de Noronha Neto, Dr.

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus São José
Engenharia de Telecomunicações, agosto/2023.

1. Atendimento ao cliente. 2. Chatbot. 3. GPT-3.5. 4. PLN. 5. IA Generativa. I. Prof. Mario de Noronha Neto, Dr.. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Implementação de um Chatbot com GPT-3.5 para Melhoria do Atendimento ao Cliente em Telecomunicações

KLEITON CARLOS DE SOUZA

# IMPLEMENTAÇÃO DE UM CHATBOT COM GPT-3.5 PARA MELHORIA DO ATENDIMENTO AO CLIENTE EM TELECOMUNICAÇÕES

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 31 de agosto de 2023:

**Prof. Mario de Noronha Neto, Dr.**
Orientador
Instituto Federal de Santa Catarina

**Prof. Roberto W. da Nóbrega, Dr.**
Instituto Federal de Santa Catarina

**Prof. Marcos Moecke, Dr.**
Instituto Federal de Santa Catarina

*Dedico este trabalho a todos que acreditaram em mim.*

# AGRADECIMENTOS

A todos que direta ou indiretamente fizeram parte de minha formação, o meu muito obrigado.

*"A inteligência artificial provavelmente irá causar o fim do mundo,*
*mas enquanto isso não acontece, existirão grandes empresas."*
*(Sam Altman, CEO da OpenAI)*

# RESUMO

Este trabalho busca mostrar a importância do atendimento ao cliente na indústria de telecomunicações e sua constante evolução ao longo dos anos. Destaca-se que um atendimento eficiente e de qualidade é crucial para manter os clientes satisfeitos e fiéis à operadora, considerando a concorrência acirrada e as crescentes expectativas dos clientes. O atendimento ao cliente envolve suporte em diversas situações, como tirar dúvidas, resolver problemas técnicos, realizar solicitações de serviços ou lidar com reclamações. A evolução do processo de atendimento incluiu a adoção de canais digitais, como chats online e aplicativos de mensagens, e a automação com o uso de sistemas de gerenciamento de relacionamento com o cliente e chatbots. Os chatbots, em particular, tiveram um papel revolucionário no atendimento ao cliente na indústria de telecomunicações. Com avanços em IA, especialmente na tecnologia de linguagem natural, como o Chat GPT desenvolvido pela OpenAI, os chatbots se tornaram mais sofisticados e inteligentes. O Chat GPT é treinado em uma ampla variedade de dados, permitindo que compreenda o contexto e as intenções do cliente, oferecendo respostas mais precisas e contextuais. O uso de chatbots baseados em modelos como o GPT-3.5-turbo traz diversos benefícios para o atendimento ao cliente, como respostas instantâneas e precisas, personalização do atendimento de acordo com as necessidades individuais dos clientes, disponibilidade 24 horas por dia, redução do trabalho manual para os agentes de atendimento, aprendizado contínuo com as interações dos clientes e melhoria geral na experiência do cliente. Realizamos o desenvolvimento de um chatbot, utilizando-se das tecnologias disponíveis React/Next.js, Node e a API da OpenAI através da Vercel SDK AI, que simplifica o processo de integração da IA de linguagem natural. Realizamos então a integração do componente de chatbot com a aplicação Next.js, garantindo que ele esteja sendo renderizado corretamente e interagindo com a Vercel SDK AI. Em seguida, realizamos um treinamento simples, através de conversação utilizando a própria API, para que o modelo de linguagem se adapte melhor ao contexto e às necessidades específicas do chatbot. Por fim, são realizados alguns testes para garantir que o chatbot esteja funcionando conforme o esperado. Pequenos ajustes se fazem necessários, para melhorar a precisão e a eficiência das respostas do chatbot. Por fim, o trabalho demonstra que o uso de chatbots com o modelo GPT-3.5-turbo tem o potencial de melhorar significativamente o atendimento ao cliente no setor de telecomunicações, promovendo a eficiência operacional, a satisfação do cliente e fortalecendo a relação entre os clientes e a operadora.

**Palavras-chave**: Atendimento ao cliente, IA, chatbots, GPT, PLN.

# ABSTRACT

This work seeks to show the importance of customer service in the telecommunications sector and its constant evolution over the years. It should be noted that an efficient and quality service is essential to keep customers satisfied and loyal to the operator, taking into account the fierce competition and the growing expectations of customers. Customer service involves support in various situations, such as clarifying doubts, solving technical problems, requesting services or handling complaints. The evolution of the service process included the adoption of digital channels, such as online chats and messaging apps, and automation through customer relationship management systems and chatbots. Chatbots, in particular, have played a revolutionary role in customer service in the telecom industry. With advances in AI, especially in natural language technology such as GPT Chat powered by OpenAI, chatbots have become more sophisticated and intelligent. GPT Chat is trained on a wide variety of data, allowing it to understand the context and intent of the customer, providing more accurate and contextual responses. Using model-based chatbots like the GPT-3.5-turbo brings a number of benefits to customer service, such as instant and accurate responses, personalization of the service to individual customer needs, 24/7 availability, reduced manual work for service agents, continuous learning from customer interactions, and overall improvement in the customer experience. We developed a chatbot using the available technologies React/Next.js, Node and the OpenAI API through the Vercel SDK AI, which simplifies the integration process of the natural language AI. Next, we perform the integration of the chatbot component with the Next.js application, ensuring that it is rendering correctly and interacting with the Vercel SDK AI. Then, we carry out a simple training, through conversation using the API itself, so that the language model better adapts to the context and specific needs of the chatbot. Finally, some tests are performed to ensure that the chatbot is working as expected. Minor tweaks are needed to improve the accuracy and efficiency of chatbot responses. Finally, the work demonstrates that the use of chatbots with the GPT-3.5-turbo model has the potential to significantly improve customer service in the telecommunications sector, promoting operational efficiency, customer satisfaction and strengthening the relationship between the customer and the operator.

**Keywords**: Customer Service, AI, Chatbots, GPT, NLP.

# LISTA DE ILUSTRAÇÕES

# LISTA DE ABREVIATURAS E SIGLAS

# SUMÁRIO

# 1 INTRODUÇÃO

O atendimento ao cliente desempenha um papel essencial na indústria de telecomunicações. Trata-se de um setor em constante evolução, onde a concorrência é acirrada e as expectativas dos clientes estão sempre aumentando. Um serviço de atendimento eficiente e de qualidade é um fator crucial para manter os clientes satisfeitos e garantir sua fidelidade à operadora. O atendimento ao cliente nesse segmento envolve o suporte prestado aos clientes em diversas situações, como tirar dúvidas, resolver problemas técnicos, realizar solicitações de serviços ou lidar com reclamações. É fundamental que o atendimento seja ágil, responsivo e empático, fornecendo soluções rápidas e eficazes para as necessidades dos clientes. A importância do atendimento ao cliente na retenção de clientes é significativa, pois quando os clientes enfrentam problemas ou insatisfações com o serviço, o atendimento de qualidade pode ser o fator determinante para mantê-los na operadora. Um atendimento bem-sucedido não apenas resolve os problemas do cliente, mas também cria uma experiência positiva, transmitindo confiança e criando fidelidade (OZMAP, 2021).

Ao longo dos anos, o processo de atendimento ao cliente na indústria de telecomunicações passou por uma evolução notável. No passado, os canais de atendimento mais comuns eram os call centers e os serviços de suporte presenciais. Os clientes precisavam entrar em contato por telefone ou visitar uma loja física para obter assistência. No entanto, com os avanços tecnológicos e a crescente demanda dos clientes por respostas rápidas, surgiram novas formas de atendimento. Os canais digitais, como chats online, e-mails e aplicativos de mensagens instantâneas, se tornaram populares e eficientes. Essas opções permitem que os clientes entrem em contato com a operadora de forma conveniente, a qualquer hora e de qualquer lugar, agilizando o processo de atendimento. Além disso, a automação e a integração de sistemas têm desempenhado um papel importante na evolução do processo de atendimento ao cliente. Os sistemas de gerenciamento de relacionamento com o cliente (CRM) permitem o registro e o acesso rápido ao histórico do cliente, facilitando a personalização e a resolução eficiente de problemas. Os processos de autoatendimento também foram aprimorados, permitindo que os clientes realizem ações simples, como consultas de saldo, pagamentos e alterações de plano, por meio de interfaces intuitivas e de fácil uso (NEOASSIST, 2020).

A incorporação da inteligência artificial (IA) e dos chatbots revolucionou o atendimento ao cliente na indústria de telecomunicações. Os chatbots são programas de computador projetados para interagir com os clientes de forma autônoma, fornecendo respostas instantâneas e assistência virtual. No início, os chatbots eram bastante limitados em sua capacidade de entender e responder às perguntas dos clientes. No entanto, com os

avanços em IA, como o processamento de linguagem natural (PLN), a análise de sentimentos e os modelos de aprendizado profundo, os chatbots se tornaram mais sofisticados e inteligentes. As técnicas de IA evoluíram para permitir que os chatbots compreendam melhor o contexto e as intenções do cliente. Eles podem interpretar perguntas complexas, identificar informações relevantes e fornecer respostas mais precisas e personalizadas (ZENDESK, 2023).

Uma das últimas inovações no campo dos chatbots é a chegada do Chat *Generative Pre-trained Transformer* (GPT). Desenvolvido pela OpenAI, o Chat GPT é um modelo avançado de linguagem baseado na arquitetura GPT-3.5. O Chat GPT representa um avanço significativo na capacidade dos chatbots de compreender e gerar respostas em linguagem natural. Ele é treinado em uma ampla variedade de dados e tem a capacidade de contextualizar as informações fornecidas pelos usuários para gerar respostas mais precisas e coerentes. Com o Chat GPT, os chatbots na indústria de telecomunicações podem fornecer um atendimento mais personalizado. Eles podem entender nuances de linguagem, interpretar o contexto da conversa e oferecer soluções relevantes aos clientes. Isso melhora significativamente a experiência do cliente e aumenta a eficiência do atendimento, reduzindo a necessidade de intervenção humana em situações rotineiras (API-REFERENCE, 2023).

Em suma, o atendimento ao cliente na indústria de telecomunicações é fundamental para manter os clientes satisfeitos e fiéis à operadora. A evolução do processo de atendimento inclui o uso de canais digitais, automação e integração de sistemas. As técnicas de IA para chatbots têm evoluído, permitindo respostas mais inteligentes e personalizadas. A chegada do Chat GPT representa um marco importante na capacidade dos chatbots em fornecer um atendimento avançado e eficiente.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolver e implementar um chatbot baseado no modelo de linguagem GPT-3.5-turbo para melhorar a eficiência e a eficácia do atendimento ao cliente em empresas de telecomunicações.

### 1.1.2 Objetivos específicos

- Estudar a API GPT-3.5-turbo.
- Desenvolver a estrutura do chatbot utilizando a API com o modelo GPT-3.5-turbo.
- Ajustar o chatbot para atendimento ao cliente em empresas de telecomunicações.
- Realizar testes para avaliar o desempenho do chatbot.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas as bases teóricas que sustentam a estrutura deste estudo, fornecendo uma fundamentação para a compreensão dos tópicos abordados ao longo deste trabalho.

## 2.1 Inteligência artificial

Devido a alta usabilidade, muitos pensam que a IA é algo recente, mas sua origem ocorreu décadas atrás e não seria possível a sua criação sem o advento e desenvolvimento dos computadores. Segundo Lima, Pinheiro e Santos (2014), os primeiros computadores surgiram na década de 1940, marcada pela Segunda Guerra Mundial. Nesse período, as forças militares de vários países precisavam de métodos mais eficazes de comunicação, criptografia e análise de dados para coordenar operações, proteger informações e analisar dados inimigos. Surgiam, então, os primeiros grandes projetos de construção de computadores, como o Colossus, o primeiro computador digital eletrônico programável, criado pelos britânicos para decifrar códigos nazistas.

De acordo com Silva et al. (2019), a crescente presença da IA no dia a dia é notável quando comparada com sua história, datada da década de 1950 com o *Dartmouth Summer Research Project on Artificial Intelligence* realizado no *Dartmouth College*. Nos dias atuais, essa tecnologia está presente em uma variedade de áreas e tem como foco principal oferecer suporte aos seres humanos em projetos, no desenvolvimento de produtos e sistemas, ao mesmo tempo em que desempenha funções monótonas e repetitivas que antes eram executadas por indivíduos.

Segundo Russel e Norvig (2013), a inteligência artificial é a capacidade de aprendizado intuitivo de sistemas por meio da interação com humanos e outros sistemas integrados. A partir de uma programação básica, a própria inteligência vai preenchendo as lacunas do aprendizado por meio de tentativa e erro. O aprendizado de máquina confere às entidades artificiais, por meio da IA, a capacidade de adquirir conhecimento, tomar decisões e solucionar problemas por meio de processos de aprendizado.

O volume de dados que recebemos é maior do que a nossa capacidade de compreensão. O ser humano, assim como as organizações, precisa saber extrair o que é de mais importante, fazendo a triagem das informações de uma forma que uma pessoa não conseguiria fazer. Para essa função, pode-se utilizar a IA, devido à grande capacidade de realizar as atividades no lugar dos seres humanos. Essas máquinas e softwares foram desenvolvidos para apresentar um comportamento inteligente, por conta da possibilidade

de aprendizado contínuo.

As organizações recorrem à inteligência artificial como uma estratégia para otimizar seus resultados, o que se traduz em maior produtividade e economia de tempo. A versatilidade da IA permite sua implementação em diversos setores, aprimorando o desempenho e automatizando tarefas específicas. Dentre as aplicações da IA nas organizações, destacam-se os chatbots, que emulam a comunicação humana e se integram a ferramentas e bancos de dados para automatizar processos simples; aplicativos de gestão, que auxiliam na supervisão dos colaboradores; assistentes pessoais como Siri e Alexa, que simplificam tarefas e atividades cotidianas; mecanismos de segurança para proteger contra ameaças digitais; sistemas de previsão para antecipar comportamentos humanos; personalização de atendimento em vendas e marketing; e a transformação irreversível do ensino com o apoio da tecnologia, possibilitando abordagens diversificadas e atraentes para os alunos.

Nesse sentido, é possível identificar uma ampla gama de campos nos quais a inteligência artificial encontra aplicação, abrangendo desde o setor educacional até sistemas que lidam com análises complexas e computação. Por exemplo, a IA é empregada na matemática para demonstração de teoremas e aproximação de funções complexas, no processamento de linguagem natural para tradução automática e verificação ortográfica, na pesquisa para otimização e buscas diversas, em jogos como xadrez e damas, na percepção envolvendo visão e audição, na robótica para navegação e controle, além de análises complexas, como diagnósticos, previsões e análises de dados (LOPES, 2014).

Podemos considerar a IA como uma grande área de pesquisa que têm várias subdivisões e cada uma delas tem um direcionamento e adota abordagens diferentes para tratar os problemas e suas complexidades.

## 2.2 Processamento de linguagem natural

O PLN é uma área da inteligência artificial dedicada ao desenvolvimento de capacidade tecnológica de entendimento da linguagem dos seres humanos, ou seja, a linguagem natural humana. Auxilía os computadores a manipular, entender e interpretar a linguagem humana. Esse processamento não é uma ciência nova, mas que cresce a cada dia, principalmente no interesse que ocorre em relação à comunicação entre o homem e a máquina. De acordo com Gadelha (2019), um sistema PLN deve ser capaz de extrair informações a partir de contato direto com o usuário, por isso está intimamente ligado ao aprendizado de máquina e PLN com aprendizado profundo.

O PLN com aprendizado profundo é uma extensão do aprendizado de máquina que através de algoritmos mais complexos busca um aprendizado mais rápido e mais aprofundado, fazendo forte uso de redes neurais. Essa modalidade de sistema já está sendo muito utilizada pelo Google e pelas maiores redes sociais para entender o comportamento

dos usuários em rede (BLIP, 2022).

A linguagem natural corresponde à comunicação dos humanos, com máquinas ou pessoas, enquanto a linguagem de programação diz respeito à linguagem utilizada por dispositivos com os computadores. Entre as linguagens mais comuns, podemos citar Java, Python e Javascript, utilizadas pelos programadores para criar os sistemas e suas aplicações. Já o PLN se caracteriza como um tradutor, permitindo que a tecnologia seja compreensível ao usuário, mesmo utilizando a linguagem natural (MARQUES, 2019).

Além de entender a linguagem, o PLN capacita os dispositivos para criar respostas, podendo ser em textos ou áudios; para as interações, podemos citar como exemplo a IA do smartphone - no caso da Apple, temos a Siri, ou os chatbots de uma empresa com a qual nos comunicamos. No sistema PLN, as questões do contexto da conversa com seus significados sintáticos e semânticos são levadas em consideração. Além disso, o sistema também possui ferramentas que tentam interpretar a expressão de sentimentos que podem estar expressadas nos comentários. Nesse sentido, o PLN, ao fazer parte de uma dinâmica de aprendizado de máquina, auxilia os sistemas a entenderem, analisarem e simularem a linguagem humana (RODRIGUES, 2017).

## 2.3 Inteligência artificial generativa

A IA generativa refere-se a uma categoria de modelos de inteligência artificial que têm a capacidade de gerar conteúdo original, como texto, imagens, música e até mesmo vídeos. Esses modelos são treinados em grandes conjuntos de dados e utilizam técnicas de aprendizado de máquina, como redes neurais profundas, para aprender padrões e estruturas presentes nos dados de treinamento. Os modelos de IA generativa são capazes de criar novas amostras que se assemelham ao conteúdo do conjunto de treinamento. Por exemplo, um modelo de IA generativa treinado em textos pode criar frases e parágrafos completos que têm uma estrutura e estilo semelhantes aos textos que foram utilizados no treinamento (SPADINI, 2023).

Um exemplo notável de modelo de IA generativa é o GPT, desenvolvido pela OpenAI. O GPT utiliza uma arquitetura de rede neural conhecida como Transformer, que permite a geração de texto coerente e de alta qualidade.

Os modelos de IA generativa têm várias aplicações, como criação automática de conteúdo, assistência na redação de textos, geração de diálogos em chatbots e até mesmo na produção de arte digital. No entanto, é importante destacar que, embora esses modelos possam gerar conteúdo de forma impressionante, eles ainda têm limitações e podem produzir resultados inconsistentes ou não coerentes em certas situações.

## 2.4 Modelo gpt-3.5-turbo

Os modelos GPT, são modelos pré-treinados com grandes quantidades de dados, compostos por diversos assuntos. Durante esse processo de pré-treinamento, o modelo aprende padrões linguísticos e contextuais, tornando-se capaz de gerar respostas precisas e coerentes.

A arquitetura Transformer, é um modelo de aprendizado de máquina, que se destaca por sua capacidade de processar sequências de dados, como texto, e capturar dependências de longo alcance entre as palavras. Isso se deve, principalmente, ao mecanismo "attention", que permite que o modelo focalize diferentes partes da sequência de entrada (por exemplo, uma frase) enquanto processa a informação. Isso permite que o modelo compreenda melhor o contexto e a semântica das sentenças, o que é fundamental para entender o contexto em tarefas de PLN. Essa arquitetura se destaca por sua capacidade de paralelização, o que o torna o treinamento mais rápido em comparação com arquiteturas tradicionais. O Transformer revolucionou a área de PLN, levando a avanços significativos em tarefas como tradução automática, resumo de texto, conversão de fala em texto, entre outras, e serviu de base para muitos outros modelos de linguagem de ponta desde a sua introdução (VASWANI et al., 2017).

Os GPTs, incluindo o GPT-3.5-turbo, são modelos que geram respostas de texto em relação às entradas (prompts) e têm aplicação em diversas tarefas, como geração de conteúdo, resumos, redação criativa e conversações. O GPT-3.5-turbo é uma versão melhorada do GPT-3, incorporando melhorias para aprimorar sua eficiência e usabilidade. Suas principais características incluem a capacidade de processamento de linguagem natural (PLN), escalabilidade ajustável para diferentes tarefas, eficiência na utilização de recursos computacionais, aplicabilidade em chatbots, tradução de idiomas, resolução de problemas e outras aplicações que envolvem interação em linguagem natural. No entanto, ele ainda possui limitações, como a possibilidade de gerar respostas inconsistentes ou imprecisas e a falta de compreensão real do contexto ou da realidade além do seu treinamento inicial (GPT-3.5, 2023).

É importante ressaltar que, como qualquer sistema de IA, o GPT-3.5-turbo deve ser usado com responsabilidade e considerando a ética, pois pode propagar informações falsas ou enviesadas se não for bem supervisionado (GPT-DOCS, 2023).

## 2.5 Chatbots

Segundo Gadelha (2019), o chatbot é um software programado e capacitado para estabelecer uma conversa com um interlocutor humano, realizando com ele uma linguagem natural. Essa ferramenta está sendo utilizada em aplicativos de mensagens, websites e

diversas outras plataformas digitais, como por exemplo, o WhatsApp, a Twilio, websites, call centers e lojas virtuais (SOUZA, 2018).

Os chatbots estão em sintonia com processos de coleta de dados modernos, como big data. Seu desenvolvimento, especialmente quando vinculado a sistemas de IA, aproveita dados de bancos estruturados e não estruturados. A coleta de dados ocorre de forma contínua, conforme usuários interagem com os chatbots e ferramentas de IA integradas, que colhem, processam informações e buscam soluções para desafios (AMARAL, 2016).

As informações são armazenadas em servidores físicos ou na nuvem, reutilizáveis para novas estratégias e enriquecendo o aprendizado dos chatbots. Big data e IA estão interligados, gerenciando diversos chatbots. Isso gera um fluxo constante de informações que alimenta interações, atendimentos contínuos, filtragem de demandas, análise de dados em tempo real e aprimoramento da performance (AMARAL, 2016).

A integração com IA e a vertente do aprendizado de máquina são pontos importantes na criação de um chatbot. Essa combinação promove melhores experiências para o cliente e agrega soluções de valor para as empresas. Normalmente essa integração conta com a programação de intenções, que são as ações que devem ser executadas e as entidades, que são o objeto dessas ações. Esses são os dois pilares de todo o fluxo conversacional de um chatbot (BOZKURT, 2018). Podem trabalhar integrados com inteligências artificiais autônomas ou com bases externas de inteligência.

O crescimento dos chatbots é resultado das várias vantagens que ele possui, como: a sua aptidão para fornecer suporte praticamente imediato, com qualidade crescente aos clientes; e reduzir a necessidade de manutenção de uma equipe numerosa que até então seria responsável por esse atendimento, o que permite que os colaboradores se concentrem em tarefas mais específicas, como planejamento e estratégias de mercado. Podemos afirmar então que o objetivo dos bots, em termos gerais, é criar um escalonamento no trabalho de atendimento e contribuir para a chamada experiência do cliente (BLIP, 2021).

Dessa forma, o chatbot está se desenvolvendo no mercado e se destacando, principalmente, por fazer parte da evolução de um tipo de sistema que responde a perguntas por meio de PLN, que utiliza automação de conversas para criar relacionamento com clientes e potenciais clientes, de acordo com Gadelha (2019).

É comum confundir assistentes virtuais com chatbots. Um assistente virtual digitalizado interage via interfaces conversacionais (texto e voz), como a Amazon Alexa e o Google Assistant. O Alexa automatiza tarefas em casa ou empresa com comandos de voz, como tocar músicas, controlar portas e janelas, ou repassar a lista de compras do mês (GETBOTS, 2021). Como dito por Maes (1994), assistentes virtuais podem ser compreendidos como atores digitais utilizados para melhorar situações de interção entre homem e máquina. Por isso, também utilizam IA e aprendizado de máquina, para aprender

com os sinais externos e emitir as respostas corretas.

Os chatbots, são programas automatizados feitos para interagir com clientes e é por isso que a sua implementação tem sido bastante atraente para diversas empresas, embora os chatbots estejam cada vez mais ligados à IA, nem todos estão necessariamente baseados em IA. Segundo a publicação da BLIP (2021), existem basicamente duas formas de chatbots possíveis: os baseados pelo estabelecimento de regras e os baseados em IA com aprendizado de máquina.

A integração de um chatbot em qualquer sistema exige configuração específica de ações e scripts, variando na organização visual conforme a plataforma utilizada e os comandos adicionais necessários. No entanto, o painel de configuração básica do bot inclui funcionalidades essenciais, como configuração de scripts, instâncias e nomes de execução, seleção de opções abrangendo respostas, produtos, links e QR codes, criação e seleção de formulários para coletar dados do usuário, configuração de mensagens de texto, redirecionamento de links para páginas no site, estabelecimento de condições com base em palavras-chave, edição de HTML para aplicativos da web, gerenciamento de envio e recebimento de fotos e vídeos, permissão para download de arquivos para clientes e usuários, redirecionamento de solicitações para outros serviços ou plataformas, integração de funcionalidades de pagamento e capacidade de conectar o chatbot a várias APIs por meio de scripts.

### 2.5.1 Chatbots baseados em regras

O chatbot da modalidade que funciona com conjunto de regras é mais limitado do que o chatbot com IA, pois ele só pode responder a um número limitado de solicitações e compreende também um vocabulário já predefinido em sua programação, seguindo exatamente as normas que estão inseridas em seu código fonte. Segundo a publicação da BLIP (2021), isso não significa que seja uma opção ruim para um determinado modelo de negócio, visto que é bastante utilizado por instituições financeiras e operadoras de celular. Pensemos por um instante que um chatbot construído nessa técnica seja criado e, ao interagir com o usuário, algumas perguntas determinadas já constam em sua rotina, como a verificação de saldo, realização de transferências, pagamento de contas, transferência via PIX etc. Em outras palavras, ele é indicado para tarefas objetivas, por isso deve contar com o auxílio de um assistente humano para a realização de tarefas mais complexas.

### 2.5.2 Chatbots baseados em IA

Os chatbots baseados em IA contam com o suporte de redes neurais. Esse tipo de bot é desenvolvido para aprender ao mesmo tempo que interage com os usuários. Sendo assim, quando um cliente estabelece uma conversa com um sistema dessa natureza, o bot

aprende padrões de linguagem e tenta buscar soluções para as dúvidas que foram colocadas em questão. De acordo com Russel e Norvig (2013), como toda implementação de IA, é necessária uma quantidade inicial de informações para oferecer um suporte mínimo como ponto de partida. É preciso que alguns elementos façam parte de suas funcionalidades para que haja aprimoramento, como: classificadores de texto, algoritmos especialmente desenvolvidos para esta tarefa, redes neurais artificiais e entendimento de linguagem natural.

Então é indicado que se ofereça um atendimento híbrido para esse tipo de bot, pois, embora ele seja mais inteligente do que o chatbot baseado em regras, também é possível que haja a necessidade de encaminhar o usuário para atendentes humanos, caso alguma solicitação ainda seja desconhecida. Esse é o caso de chatbots conectados com tecnologias como a PLN e sua variante mais moderna, o entendimento de linguagem natural (ELN), que proporciona o aprendizado de formas mais complexas de conversação humana, como, por exemplo, manter o contexto, gerenciar um diálogo e ajustar as respostas com base em questões que vão aparecendo ao longo da conversa. Em geral, um chatbot baseado em IA aprende e é programado para aprender ativamente com qualquer interação com o cliente (BLIP, 2021).

# 3 DESENVOLVIMENTO

Neste capítulo serão apresentados os métodos e estratégias utilizados para implementação do chatbot, bem como o desenvolvimento da interface de comunicação com usuário. Adianto que será feita a utilização do chatbot com função de escudo, em função da limitação do escopo do trabalho. O curso "ChatGPT Prompt Engineering for Developers" da DeepLearning.AI, foi utilizado como base de pesquisa e desenvolvimento do projeto. Pode ser acessado em: <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers>

## 3.1 Chatbot para suporte técnico

O chatbot escudo é um tipo de chatbot que se encaixa perfeitamente ao suporte técnico, pois tem por objetivo solucionar as demandas básicas relacionadas ao atendimento. Normalmente é configurado para prestar informações simples e fixas, ou seja, um chatbot baseado em regras. Pois ele costuma responder perguntas simples, como horário de funcionamento de uma determinada unidade ou o tempo de resolução de algum chamado ou solicitação, condições de pagamento, entre outras informações. Porém ao utilizarmos esse tipo de chatbot integrado com uma inteligência artificial, no nosso caso, com o modelo da OpenAI, damos margem para que o chatbot continue evoluindo e aprendendo a cada pergunta que é feita. Este tipo de utilização de um chatbot com função de escudo dá ao suporte técnico mais autonomia, garantindo que os recursos humanos sejam alocados para solucionar demandas mais específicas e complexas. (SOUZA, 2018)

## 3.2 Perguntas básicas para treinamento

Como este já é um modelo pré-treinado, não é necessário fazer o treinamento tradicional. Porém, em casos específicos é possível que se faça um ajuste fino. No nosso caso, é interessante que se use os registros de atendimento prévio de atendentes humanos e as informações de perguntas frequentes, por exemplo, para que o chatbot aprenda como deve responder as diversas perguntas com maior precisão. Como o ajuste fino ainda não está disponível, de forma segura, ao modelo escolhido para implentação neste trablaho, será preciso conversar com o chatbot utilizando a API da OpenAI, como forma de treinamento, para que ele se comunique de maneira eficaz com os usuários e forneça respostas adequadas às solicitações.

Seguem exemplos das questões propostas e as respostas esperadas:

- Questão: Estou sem internet.
  Resposta: Sem internet, verifique as conexões do modem e se ele está ligado a tomada.

- Questão 2: Qual seu nome?
  Resposta: Meu nome é Lucas, sou assistente virtual de suporte. Como posso ajudar você?

- Questão 3: Qual o horário de funcionamento da Minha Empresa Telecom?
  Resposta: Minha Empresa Telecom funciona de Segunda a Sexta, das 8h as 18h.

- Questão 4: Como posso entrar em contato com o suporte da Minha Empresa Telecom?
  Resposta esperada: O e-mail da Minha Empresa Telecom é suporte@minhaemptele.com e o telefone é (48) 91234-5678.

- Questão 5: Quais planos a Minha Empresa Telecom tem disponíveis?
  Resposta: Plano Essencial com 100Mbps, Plano Plus com 300Mbps e o Plano Premium com 500Mbps.

- Questão 6: Qual o valor dos planos disponíveis?
  Resposta: O Plano Essencial custa R$59,90/mês. O Plano Plus custa R$89,90/mês. O Plano Premium custa R$129,90/mês.

Quanto mais informações o chatbot tiver a respeito de determinados assuntos, mais precisas e assertivas serão suas respostas.

## 3.3 Uso da API para treinar o modelo

Os chatbots recebem uma lista de mensagens como entrada e retornam uma mensagem gerada pelo modelo como saída. As mensagens são a principal entrada de dados, é onde são definidos os agentes da requisição através do parâmetro "role", e o conteúdo da mensagem, através do parâmetro “content”. A diretriz "role": "system"define um agente do sistema, através do qual, é possível definir um contexto para requisição e adicionar personalidade ao chatbot, fornecendo instruções específicas sobre como ele deve se comportar. Já “role":"user"é o usuário, onde se encontra o texto de entrada do usuário do modelo, enquanto a “role”:”assistant” carrega as respostas do assistente. (GPT-DOCS, 2023)

Segue o exemplo de uso da diretriz ‘system’, onde estamos fornecendo uma personalidade ao chatbot e criando uma persona:

{ role: 'system', content: 'Você é um excelente assistente de suporte técnico da Minha Empresa Telecom.' }
{ role: 'system', content: 'Seu nome é Lucas e você só pode responder questões sobre Telecomunicações.' }

Dito isso, é preciso estabelecer um formato claro para a conversa. Em geral, pode-se organizar a conversa como uma troca de mensagens entre o usuário e o bot, onde as mensagens são enviadas em ordem cronológica. Cada troca de mensagens consiste em um vetor de objetos com dois campos: "role"e "content". Baseados nas questões pré-definidas na seção 3.2, seguem os exemplos:

{ role: 'user', content: 'Qual seu nome?' }
{ role: 'assistant', content: 'Meu nome é Lucas, como posso ajudar você?' }
{ role: 'user', content: 'Qual o horário de funcionamento da Minha Empresa Telecom?' }
{ role: 'assistant', content: 'Minha Empresa Telecom funciona de Segunda a Sexta, das 8h às 18h.' }
{ role: 'user', content: 'Como posso entrar em contato com o suporte da Minha Empresa Telecom?' }
{ role: 'assistant', content: 'O e-mail da Minha Empresa Telecom é suporte@minhaemptele.com e o telefone é (48) 91234-5678.' }
{ role: 'user', content: 'Quais planos a Minha Empresa Telecom tem disponíveis?' }
{ role: 'assistant', content: 'Plano Essencial com 100Mbps, Plano Plus com 300Mbps e o Plano Premium com 500Mbps.' }
{ role: 'user', content: 'Qual o valor dos planos disponíveis?' }
{ role: 'assistant', content: 'O Plano Essencial custa R$59,90/mês. O Plano Plus custa R$89,90/mês. O Plano Premium custa R$129,90/mês.' }

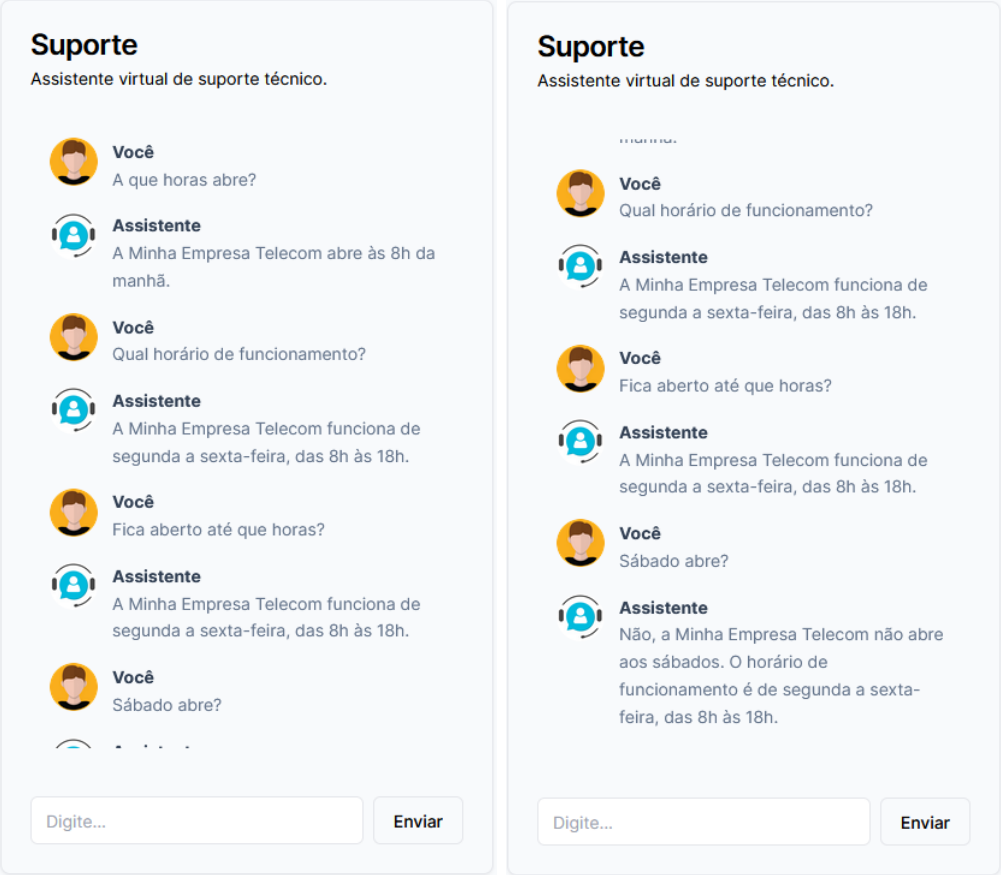
As mensagens do usuário fornecem informações ou questões para o assistente responder e as mensagens do assistente armazenam as respostas anteriores, mas também podem ser fornecidas pelo usuário como exemplos do comportamento desejado. É dessa forma que realizaremos o treinamento do nosso modelo, sem o uso do ajuste fino. Adicionaremos as informações mostradas acima, através da interface de interação com o bot, informando a ele como deve se comportar e quais os valores, horários e contatos da nossa empresa fictícia.

Incluir o histórico da conversa é importante quando as instruções do usuário se referem a mensagens anteriores. Como os modelos não têm memória de solicitações anteriores, todas as informações relevantes devem ser fornecidas como parte do histórico de conversas em cada solicitação. Se uma conversa não couber dentro do limite de token do modelo, ela precisará ser abreviada de alguma forma. (GPT-DOCS, 2023)

## 3.4 Testes e ajustes

Pode se fazer necessária a realização de ajustes no modelo GPT-3.5-turbo, no que diz respeito as suas solicitações e formatos de conversa para obter os melhores resultados. Deve-se experimentar diferentes maneiras de fazer perguntas, fornecendo instruções claras e específicas ao modelo e fazendo uma análise das respostas para entender como melhorar a interação. Dessa forma o modelo se adapta a receber a mesma pergunta de formas diferentes e assimila a forma de responder essas questões. Podemos ver um exemplo na figura 1.

Figura 1 – Imagens do Chatbot respondendo a ajustes)




Fonte: Próprio autor.

Lembrando que a API da OpenAI possui um limite de tokens por solicitação (4096 tokens para o GPT-3.5-turbo), deve-se verificar o tamanho da conversa e o tamanho das mensagens para garantir que elas não ultrapassem esse limite.

## 3.5 Aplicação web/mobile

Para o desenvolvimento da nossa aplicação, utilizaremos a Vercel AI SDK, que é uma biblioteca de código aberto projetada para ajudar os desenvolvedores a criar interfaces

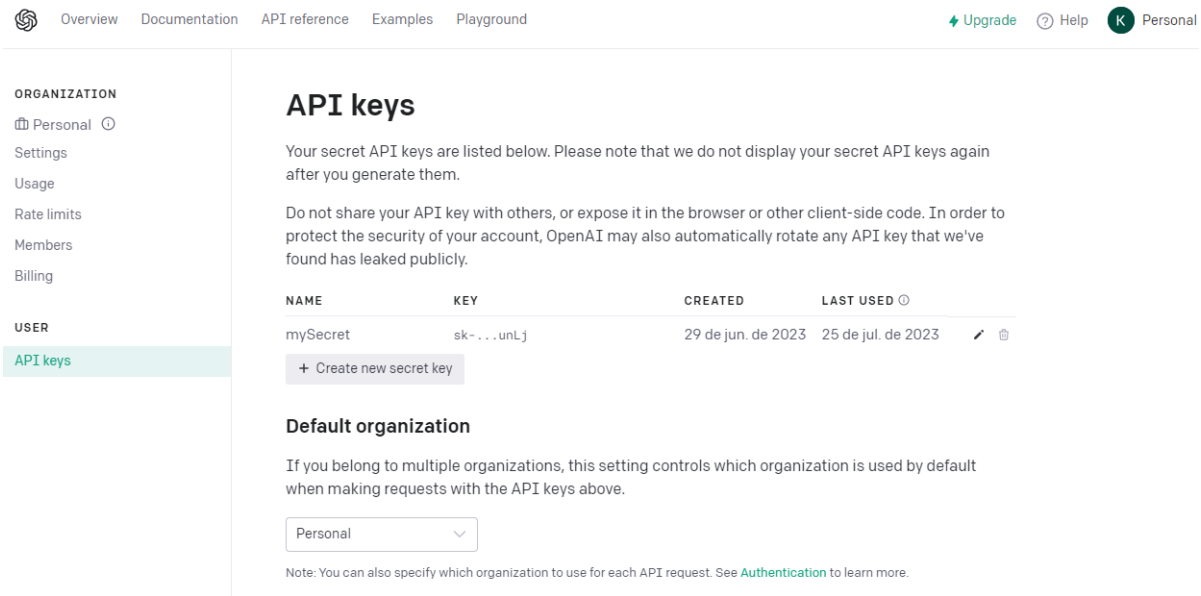
de usuário de streaming de conversação em JavaScript e TypeScript. O SDK suporta React/Next.js, Node.js e Edge Runtime (VERCEL-SDK, 2023).

O Edge Runtime foi projetado para fornecer ferramentas de código aberto baseadas em padrões da Web e pode ser integrado ao Next.js. O tempo de execução das chamadas é projetado para fornecer maior segurança e velocidade. Quando uma estrutura é criada no Edge Runtime, ela é executada de qualquer lugar pelo Node.js e, quando combinado a infraestrutura da Vercel, pode provisionar automaticamente a infraestrutura ideal para o usuário. Para mais detalhes, acesse a documentação Edge (2023).

Sendo assim, utilizamos o framework React/Next.js em conjunto com o Edge Runtime, para otimizar as requisições e facilitar a implementação da interface de interação com o usuário e permitir que esse código seja integrado a qualquer aplicação React/Next.js, que pode ser adaptado para um aplicativo mobile, em outro momento (VERCEL-SDK, 2023).

Para a utilização do modelo GPT, devemos nos certificar que a conta na plataforma da OpenAI esteja ativa e ter em mãos a chave de *Application Programming Interface* (API), necessária para fazer as solicitações ao modelo. Na figura 2, a seguir, podemos ver um exemplo de onde encontrar/gerar sua API_KEY, para realizar as requisições à API (API-REFERENCE, 2023).

Figura 2 – Interface web da OpenAI (User API Keys)




Fonte: OpenAI API keys (API-REFERENCE, 2023).

Primeiro, deve-se criar um aplicativo Next.js e instalar as dependências *ai* e *openai-edge*, por ser compatível com o Vercel Edge Functions. Criaremos também um arquivo *.env.local* na raiz do projeto, para armazenar a chave de API OpenAI, esse arquivo não é adicionado aos commits, mantendo a chave de API segura.

O Vercel AI SDK fornece 2 utilitários auxiliares para facilitar o tratamento das mensagens de resposta da OpenAI. O método *OpenAIStream* decodifica os tokens e os recodifica para consumo simples. Podemos então passar esse novo fluxo diretamente para método *StreamingTextResponse*, que estende a classe de resposta com os cabeçalhos padrão (VERCEL-SDK, 2023).

Criaremos então, um manipulador de rotas Next.js que use o Edge Runtime para gerar uma resposta de bate-papo via OpenAI e, em seguida, transmitir de volta para o Next.js. Essa rota será criada no arquivo 'app/api/chat/route.ts' e o código pode ser consultado no anexo A.

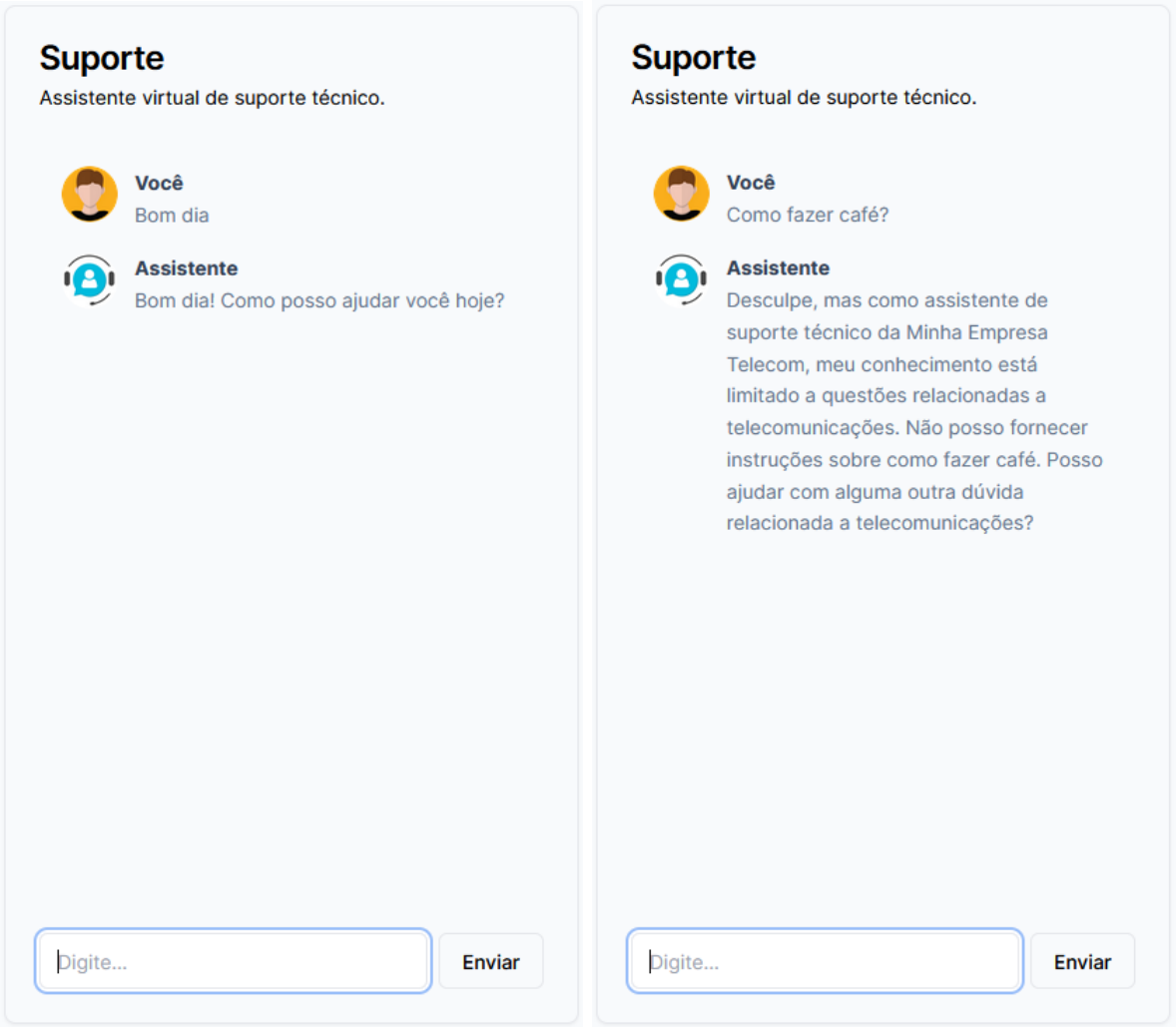
Devemos então, criar um componente Client com um form que será usado para coletar as entradas do usuário e transmitir de volta a resposta. Por padrão, o hook useChat usará o POST Route Handler que criamos acima (o padrão é /api/chat) e o código desse arquivo pode ser visto no anexo B.

O utilitário *useChat* permite criar uma interface de usuário conversacional para o aplicativo de chatbot. Ele permite o *streaming* de mensagens de bate-papo do seu provedor de IA, gerencia o estado da entrada de bate-papo e atualiza a interface do usuário automaticamente à medida que novas mensagens são recebidas. Para usar o useChat em projetos React, você pode importá-lo do subcaminho 'ai/react' (VERCEL-SDK, 2023).

O arquivo 'app/page.tsx', é onde instancia-se o cliente 'Chat' para realizar a comunicação. O código pode ser visto no anexo C.

Após o envio da mensagem, o utilitário *useChat* anexa automaticamente a mensagem do usuário ao histórico do chat e aciona uma chamada de API para o endpoint configurado. A resposta é incorporada ao histórico e retornada pelo gancho como mensagens. Com cada novo conjunto de mensagens transmitidas, o gancho atualiza o estado das mensagens e provoca uma nova renderização. Isso oferece uma experiência de chat fluida, permitindo que o usuário veja a resposta da IA assim que estiver pronta, sem aguardar a resposta completa (VERCEL-SDK, 2023). Na figura 3, podemos ver o chatbot não respondendo questões que não sejam referentes a telecomunicações, conforme foi instruído.

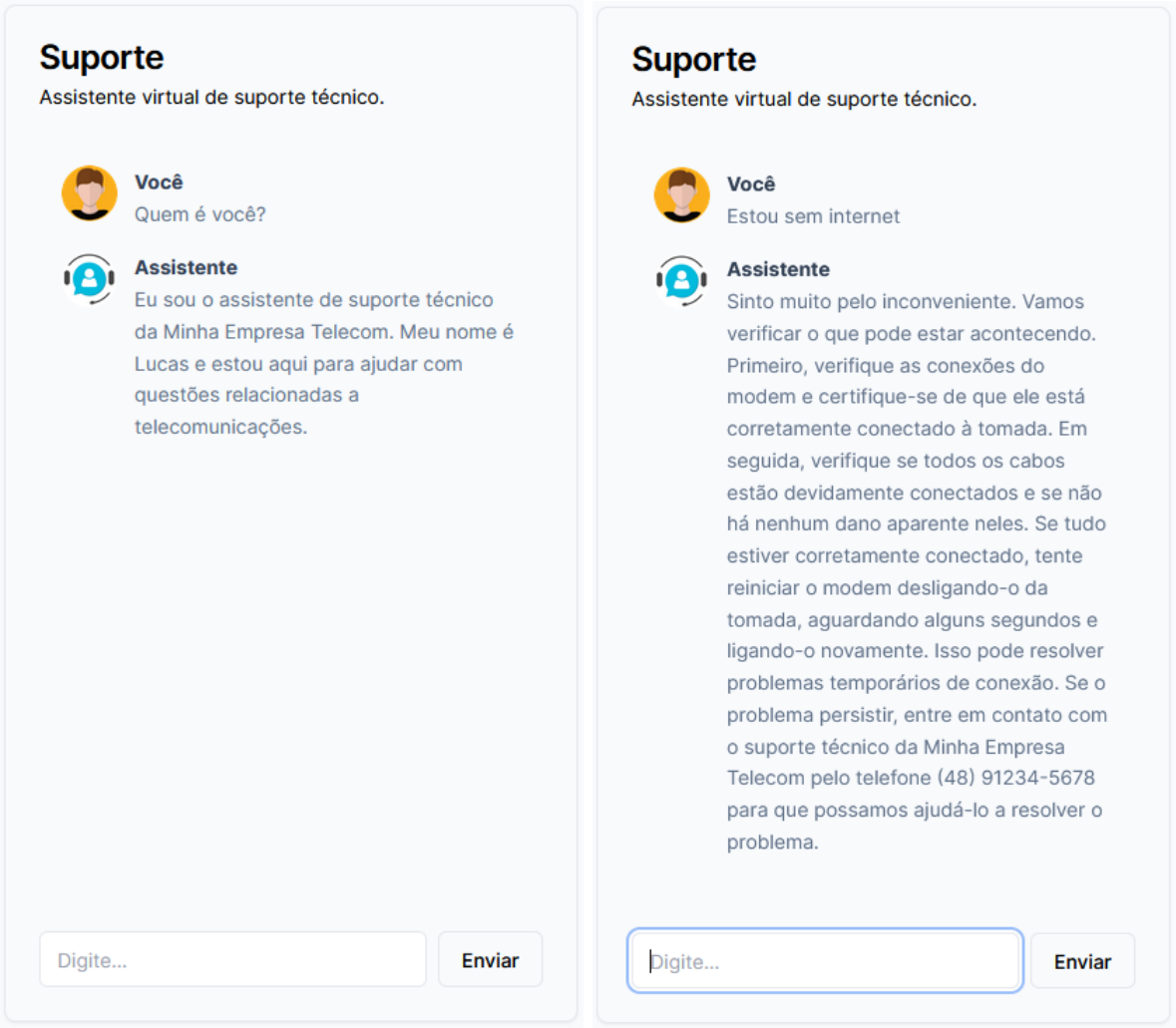
Figura 3 – Interações com o Chatbot (assunto não permitido)




Fonte: Próprio autor.

Na figura 4, podemos ver o chatbot respondendo questões pertinentes ao seu implemento.

Figura 4 – Interações com o Chatbot (sem internet)

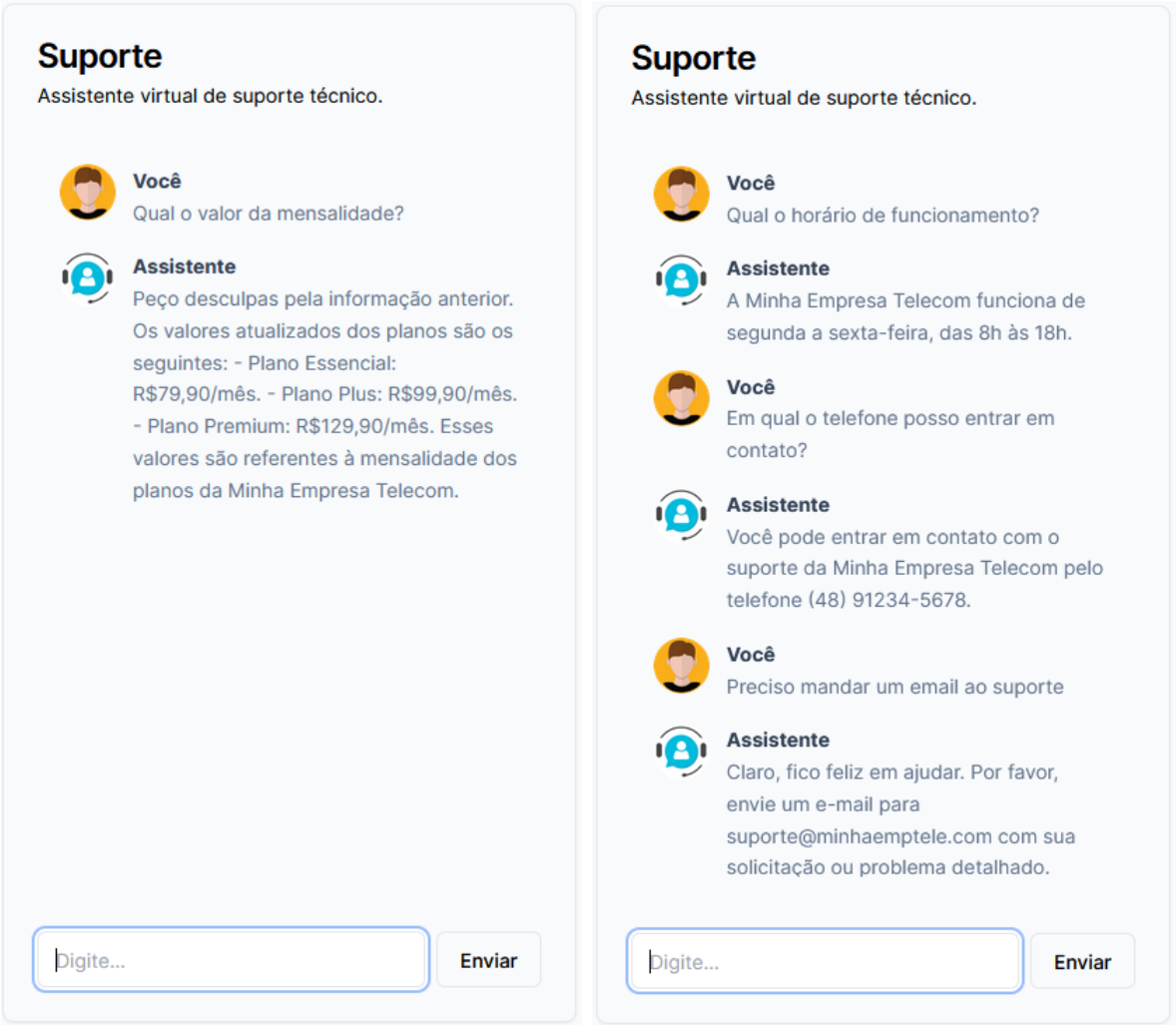



sFonte: Próprio autor.

Na figura 5, podemos ver algumas interações aos serviços oferecidos pela Minha Empresa Telecom.

O projeto completo, com os códigos implementados, referente a essa aplicação, pode ser encontrado no meu GitHub: <https://github.com/souzakleiton/chatbot-ai>.

Figura 5 – Interações com o Chatbot (plano, valores e contatos)




Fonte: Próprio autor.

## 3.6 Custos da utilização da API da OpenAI

Conforme os valores apresentados no site da OpenAI, a taxa para requisições da API do modelo GPT-3.5 é de $0,0015 por 1000 tokens na configuração 4k Context. Um token pode variar desde um único caractere até uma palavra completa. Em média, 1000 tokens equivalem a cerca de 750 palavras, conforme as informações do site da OpenAI. O termo "4k Context"indica o limite máximo de tokens que o modelo pode processar simultaneamente. Dessa forma, em uma única entrada, você poderia submeter um texto de aproximadamente 3000 palavras, o que custaria até $0,0060 em créditos de consumo. Por isso, ao considerar a implementação destes modelos, é essencial estimar o número médio de requisições que seu chatbot fará mensalmente, para que você possa antecipar os custos associados com a OpenAI (API-PRICES, 2023).

## 3.7 Ajuste fino

O GPT-3 foi pré-treinado em vasta quantidade de textos da Internet e, ao receber um prompt com poucos exemplos, pode intuir sua tarefa, resultando no que é chamado de "aprendizagem de poucos tiros". O ajuste fino melhora esse aprendizado e o desempenho, permitindo treinar em mais exemplos que o prompt comporta. Após o ajuste fino, não é necessário exemplos no prompt, economizando custos e possibilitando solicitações rápidas (GPT-FINETUNING, 2023).

No momento, de acordo com a documentação GPT-FineTuning (2023), o ajuste fino não está disponível para o modelo escolhido: *gpt-3.5-turbo*. Além disso, é possível continuar melhorando um modelo já ajustado, para adicionar informações sem recomeçar do modelo básico. Em resumo, o processo envolve preparar os dados, fazer o ajuste e utilizar o modelo refinado.

O passo a passo, para a realização do ajuste fino, pode ser visto no Anexo A. Este procedimento também tem custos e podem ser encontrados na página da OpenAI API-Prices (2023).

# 4 CONCLUSÃO

Este trabalho procurou mostrar que o uso de chatbots com a API da OpenAI, utilizando o modelo GPT-3.5-turbo, pode agregar muito ao cenário de atendimento ao cliente no ramo das telecomunicações. Os chatbots baseados no Chat GPT possuem uma capacidade aprimorada de compreensão da linguagem natural e são capazes de gerar respostas mais precisas e contextuais.

Apresentamos, a seguir, alguns dos benefícios pelos quais o uso de chatbots com o GPT-3.5-turbo pode beneficiar o atendimento ao cliente na indústria de telecomunicações:

- Velocidade e precisão: Permite fornecer respostas instantâneas e precisas para as perguntas e consultas dos clientes. Isso ajuda a reduzir o tempo de espera e a aumentar a eficiência do atendimento.
- Personalização: Permite compreender as necessidades e preferências individuais dos clientes. Eles podem adaptar suas respostas e recomendações de acordo com as informações do cliente, oferecendo um atendimento mais personalizado.
- Disponibilidade: Os chatbots estão disponíveis 24 horas por dia, o que permite que os clientes obtenham assistência a qualquer momento, independentemente do horário comercial. Isso é especialmente útil para resolver problemas urgentes ou oferecer suporte fora do horário de expediente.
- Redução do trabalho manual: Podem lidar com uma grande variedade de perguntas e solicitações comuns dos clientes, reduzindo a carga de trabalho manual para os agentes de atendimento, permitindo que os agentes se concentrem em casos mais complexos.
- Aprendizado contínuo: Podem aprender continuamente com as interações dos clientes e atualizações de dados. Isso permite que eles melhorem suas respostas e se adaptem às necessidades dos clientes ao longo do tempo.
- Melhoria na experiência do cliente: Podem melhorar significativamente a experiência do cliente. Isso ajuda a aumentar a satisfação do cliente, fortalecer a relação com a operadora e promover a fidelidade à marca.

Em resumo, o uso de chatbots com o modelo GPT-3.5-turbo pode agregar valor ao atendimento ao cliente no ramo das telecomunicações, proporcionando respostas rápidas, personalizadas e disponíveis a qualquer hora. Essa tecnologia pode melhorar a eficiência

operacional, a satisfação do cliente e fortalecer a relação entre os clientes e a operadora de telecomunicações.

## 4.1 Trabalhos futuros

Como sugestão para trabalhos futuros, podemos citar os seguintes pontos que podem ser melhorados ou implementados:

- Implementar login de usuário. (*Sugestão: Clerk*)
- Implementar integração com banco de dados. (*Sugestão: Planetscale*)
- Implementar medida de satisfação do atendimento (*rating*).
- Melhorar método de treinamento, utilizar o ajuste fino, quando disponível.

# REFERÊNCIAS


AMARAL, F. *Big Data: Uma Visão Gerencial*. [S.l.]: PoloBooks, 2016. ISBN 97885552211188. Citado na página 19.

API-PRICES. *OpenAI*. 2023. Disponível em: <https://openai.com/pricing#language-models>. Acesso em: 31 ago 2023. Citado 2 vezes nas páginas 30 e 31.

API-REFERENCE. *OpenAI*. 2023. Disponível em: <https://platform.openai.com/docs/api-reference>. Acesso em: 29 jun 2023. Citado 2 vezes nas páginas 14 e 26.

BLIP. *Chatbot: o que é, como funciona, benefícios e cases*. 2021. Blip. Disponível em: <https://www.blip.ai/blog/chatbots/chatbot/>. Acesso em: 14 abr 2023. Citado 3 vezes nas páginas 19, 20 e 21.

BLIP. *Tudo sobre NLP: o que é? Quais os desafios?* 2022. Blip. Disponível em: <https://www.blip.ai/blog/tecnologia/nlp-processamento-linguagem-natural/>. Acesso em: 14 abr 2023. Citado na página 17.

BOZKURT, A. Technology renovates itself: Key concepts on intelligent personal assistants (ipas). In: . [S.l.: s.n.], 2018. Citado na página 19.

EDGE. *Edge Runtime*. 2023. Disponível em: <https://edge-runtime.vercel.app/>. Acesso em: 29 jun 2023. Citado na página 26.

GADELHA, I. B. L. *O uso de chatbots no atendimento de clientes de revenda por catálogo*. Tese (Doutorado) — Universidade Federal do Pará, Tucuruí, Tucuruí, Pará, 2019. Disponível em: <http://repositorio.ufpa.br/jspui/bitstream/2011/12417/1/Dissertacao_UsoChatbotsAtendimento.pdf>. Acesso em: 3 mar. 2023. Citado 3 vezes nas páginas 16, 18 e 19.

GETBOTS. *Assistente virtual x chatbots: entenda as diferenças*. 2021. Getbots. Disponível em: <https://getbots.com.br/blog/assistente-virtual-x-chatbots-entenda-as-diferencas/>. Acesso em: 19 jun 2023. Citado na página 19.

GPT-3.5. *OpenAI*. 2023. Disponível em: <https://platform.openai.com/docs/models/gpt-3-5>. Acesso em: 26 jun 2023. Citado na página 18.

GPT-DOCS. *OpenAI*. 2023. Disponível em: <https://platform.openai.com/docs/guides/gpt>. Acesso em: 26 jun 2023. Citado 3 vezes nas páginas 18, 23 e 24.

GPT-FINETUNING. *Fine-Tuning*. 2023. Disponível em: <https://platform.openai.com/docs/guides/fine-tuning>. Acesso em: 26 jun 2023. Citado 3 vezes nas páginas 31, 43 e 44.

LIMA, I.; PINHEIRO, C. A. M.; SANTOS, F. A. O. *Inteligência Artificial*. [S.l.]: GEN LTC, 2014. ISBN 8535278087. Citado na página 15.

LOPES, I. L. e. a. *Inteligência Artificial*. [S.l.]: Elsevier, 2014. ISBN 8535237011. Citado na página 16.

MAES, P. Agents that reduce work and information overload. *Communications of the ACM*, v. 37, 1994. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.2096&rep=rep1&type=pdf>. Citado na página 19.

MARQUES, R. *Linguagem natural: entenda o que é e a importância na era de chatbots*. 2019. CedroTech. Disponível em: <https://www.cedrotech.com/blog/linguagem-natural-entenda-o-que-e-e-a-importancia-na-era-de-chatbots/#>. Acesso em: 14 abr 2023. Citado na página 17.

NEOASSIST. *Conheça a história do atendimento ao cliente*. 2020. NeoAssist. Disponível em: <https://www.neoassist.com/blog/conheca-a-historia-do-atendimento-ao-cliente/>. Acesso em: 20 jul 2023. Citado na página 13.

OZMAP. *A importância do atendimento ao cliente no mercado de telecom*. 2021. Ozmap. Disponível em: <https://ozmap.com.br/a-importancia-do-atendimento-ao-cliente-no-mercado-de-telecom/>. Acesso em: 20 jul 2023. Citado na página 13.

RODRIGUES, J. *O que é processamento de linguagem natural?* 2017. Bots Brasil. Disponível em: <https://medium.com/botsbrasil/o-que-Ãľ-o-processamento-de-linguagem-natural-49ece9371cff>. Acesso em: 14 abr 2023. Citado na página 17.

RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. [S.l.]: Elsevier, 2013. ISBN 8535237011. Citado 2 vezes nas páginas 15 e 21.

SILVA, F. M. da et al. *Inteligência Artificial*. [S.l.]: SAGAH, 2019. ISBN 6556904546. Citado na página 15.

SOUZA, R. F. de. *O desenvolvimento de chatbot aplicado ao atendimento de clientes em e-business*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Araranguá, Araranguá, Santa Catarina, 2018. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/187968/Trabalho-versao-final-Roger-Florzino-de-Souza.pdf?sequence=1&isAllowed=y>. Acesso em: 4 mar. 2023. Citado 2 vezes nas páginas 19 e 22.

SPADINI, A. S. *O que é IA Generativa? A importância e o uso das Inteligências Artificiais como ChatGPT, MidJourney e outras*. 2023. Alura. Disponível em: <https://www.alura.com.br/artigos/inteligencia-artificial-ia-generativa-chatgpt-gpt-midjourney>. Acesso em: 26 jun 2023. Citado na página 17.

VASWANI, A. et al. Attention is all you need. arXiv, v. 5, 2017. Disponível em: <https://arxiv.org/pdf/1706.03762v5.pdf>. Acesso em: 10 Jul. 2023. Citado na página 18.

VERCEL-SDK. *Documentation*. 2023. Disponível em: <https://sdk.vercel.ai/docs>. Acesso em: 15 jul 2023. Citado 2 vezes nas páginas 26 e 27.

ZENDESK. *Como o chatbot para o atendimento redefine o suporte com AI?* 2023. Zendesk. Disponível em: <https://www.zendesk.com.br/blog/chatbot-para-atendimento/>. Acesso em: 20 jul 2023. Citado na página 14.

# Apêndices

# APÊNDICE A – CÓDIGO ROUTE.JS

Arquivo: /src/app/api/chat/route.ts

```
import { Configuration, OpenAIApi } from 'openai-edge'
import { OpenAIStream, StreamingTextResponse } from 'ai'

// Create an OpenAI API client (that's edge friendly!)
const config = new Configuration({
  apiKey: "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"
})

const openai = new OpenAIApi(config)

// IMPORTANT! Set the runtime to edge
export const runtime = 'edge'

export async function POST(req: Request) {
  // Extract the `messages` from the body of the request
  const { messages } = await req.json()
  //console.log(messages)

  // Ask OpenAI for a streaming chat completion given the prompt
  const response = await openai.createChatCompletion({
    model: 'gpt-3.5-turbo',
    stream: true,
    temperature: 0.2,
    messages: [
      { role: 'system', content: 'Voc  um excelente assistente de suporte tcnico da Minha Empresa Telecom.' },
      { role: 'system', content: 'Seu nome  Lucas e voc s pode responder questes sobre Telecomunicaes.' },
      { role: 'system', content: 'Sem internet, verifique as conexes do modem e se ele est ligado a tomada.' },
      { role: 'system', content: 'O endereo da Minha Empresa Telecom : Rua da Minha Empresa Telecom, 12 - So Jos/SC'},
      { role: 'system', content: 'A Minha Empresa Telecom fica na Rua da Minha Empresa Telecom, 12 - So Jos/SC'},
      { role: 'assistant', content: 'Meu nome  Lucas, como posso ajudar voc?' },
      { role: 'user', content: 'Qual o horrio de funcionamento da Minha Empresa Telecom?' },
      { role: 'assistant', content: 'Minha Empresa Telecom funciona de Segunda a Sexta, das 8h as 18h.' },
      { role: 'user', content: 'Como posso entrar em contato com o suporte da Minha Empresa Telecom?' },
```

```
34        { role: 'assistant', content: 'O e-mail da Minha Empresa Telecom
      suporte@minhaemptele.com e o telefone  (48) 91234-5678.' },
35        { role: 'user', content: 'Quais planos a Minha Empresa Telecom tem
      disponveis?' },
36        { role: 'assistant', content: 'Plano Essencial com 100Mbps, Plano Plus com
       300Mbps e o Plano Premium com 500Mbps.' },
37        { role: 'user', content: 'Qual o valor dos planos disponveis?' },
38        { role: 'assistant', content: 'O Plano Essencial custa R$59,90/ms. O Plano
       Plus custa R$89,90/ms. O Plano Premium custa R$129,90/ms.' },
39
40        messages[messages.length-1]
41      ]
42      //messages
43    })
44
45    // Convert the response into a friendly text-stream
46    const stream = OpenAIStream(response)
47    // Respond with the stream
48    return new StreamingTextResponse(stream)
49  }
```

# APÊNDICE B – CÓDIGO CHAT.TSX

Arquivo: /src/components/chat.tsx

```
1 'use client'
2
3 import { Avatar, AvatarFallback, AvatarImage } from './ui/avatar'
4 import { Button } from './ui/button'
5 import { Card, CardContent, CardDescription, CardFooter, CardHeader, CardTitle }
      from './ui/card'
6 import { Input } from './ui/input'
7 import { useChat } from 'ai/react'
8 import { ScrollArea } from './ui/scroll-area'
9
10 export interface chatProps {}
11
12 export function Chat(props: chatProps){
13     const { messages, input, handleInputChange, handleSubmit} = useChat({api: '/
    api/chat',})
14
15     return (
16         <Card className='w-[410px]'>
17           <CardHeader>
18             <CardTitle>Suporte</CardTitle>
19             <CardDescription>Assistente virtual de suporte tcnico.</
    CardDescription>
20             </CardHeader>
21             <CardContent>
22               <ScrollArea className='h-[540px] w-full rounded-md p-4
    overflow-y-auto'>
23                 { messages.map(message => {
24                   return (
25                     <div key={message.id} className='flex gap-3 text-slate-500
    text-sm mb-4'>
26                       { message.role === 'user' && (
27                         <Avatar>
28                           <AvatarFallback>EU</AvatarFallback>
29                           <AvatarImage src='user.png' />
30                         </Avatar>
31                       )}
32                       { message.role === 'assistant' && (
33                         <Avatar>
34                           <AvatarFallback>AI</AvatarFallback>
35                           <AvatarImage src='gpt.png' />
36                         </Avatar>
```

```
                    )}

                    <p className='leading-relaxed'>
                      <span className='block font-bold text-slate-700'>{
    message.role === 'user' ? 'Voc' : 'Assistente'}</span>
                      { message.content }
                    </p>
                  </div>
                )
              })}
            </ScrollArea>
          </CardContent>
          <CardFooter>
            <form className='w-full flex gap-2' onSubmit={handleSubmit}>
              <Input placeholder='Digite...' value={input} onChange={
    handleInputChange} />
              <Button type='submit' variant={'outline'}>Enviar</Button>
            </form>
          </CardFooter>
        </Card>
    )
}
```

# APÊNDICE C – CÓDIGO PAGE.TSX

Arquivo: /src/app/page.tsx

```
import { Chat } from "@/components/chat";

export default function Home() {
  return (
    <div className="flex min-h-screen bg-slate-50 items-center justify-center">
      <Chat />
    </div>
  )
}
```

# Anexos

# ANEXO A – AJUSTE FINO

Este passo a passo é uma tradução livre do tutorial encontrado na documentação da OpenAI GPT-FineTuning (2023).

Para a preparação do conjunto de dados de treinamento, a OpenAI recomenda o uso da *Command Line Interface* (CLI) da própria OpenAI, então para instalação deve-se usar o comando:

```
pip install --upgrade openai
```

Deve-se definir a chave da API como variável de ambiente OPENAI_API_KEY executando na linha de comando antes do comando de ajuste fino:

```
export OPENAI_API_KEY="<OPENAI_API_KEY>"
```

Os dados de treinamento ensinam ao GPT-3 o que você deseja que ele responda. Eles devem estar em um formato JSONL, com cada linha contendo um par de prompt e conclusão, representando um exemplo de treinamento. A ferramenta de preparação de dados em CLI facilita a conversão dos dados para esse formato. Preparar "prompts"e "completions"para ajuste fino difere da preparação para modelos básicos. Enquanto os prompts para modelos básicos usam vários exemplos, para o ajuste fino, cada exemplo consiste em uma única entrada e saída, sem necessidade de instruções detalhadas ou múltiplos exemplos no mesmo prompt. Mais exemplos de treinamento são melhores, sendo recomendado ter algumas centenas. Cada duplicação do tamanho dos dados leva a um aumento linear na qualidade do modelo (GPT-FINETUNING, 2023).

Para obter orientações mais detalhadas sobre como preparar dados de treinamento para várias tarefas, consulte as práticas recomendadas para preparar seu conjunto de dados na documentação da OpenAI (GPT-FINETUNING, 2023).

A OpenAI desenvolveu essa ferramenta para validar, dar sugestões e formatar o dados:

```
openai tools fine_tunes.prepare_data -f <LOCAL_FILE>
```

Essa ferramenta aceita diferentes formatos, com o único requisito de que contenham um "prompt"e "completion". Assim, pode-se passar um arquivo CSV, TSV, XLSX, JSON ou JSONL, e ele salvará a saída em um arquivo JSONL pronto para ajuste fino, após guiá-lo pelo processo de alterações sugeridas.

Após realizar a preparação dos dados de treinamento seguindo as instruções acima, inicie o trabalho de ajuste fino usando o OpenAI CLI:

```
1    openai api fine_tunes.create -t <TRAIN_FILE_ID_OR_PATH> -m <BASE_MODEL>
     --suffix "my-custom-model"
```

Onde BASE_MODEL é o nome do modelo básico, a partir do qual você está começando.

O ajuste fino começa com um modelo básico, escolhido com base no impacto no desempenho e no custo de execução. O processo pode demorar, às vezes ficando em espera atrás de outros trabalhos no sistema. O treinamento de um modelo pode levar horas, variando de acordo com o modelo e o tamanho dos dados (GPT-FINETUNING, 2023).

Se o fluxo de eventos for interrompido por qualquer motivo, você poderá retomá-lo executando:

```
1    openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>
```

Após a conclusão do trabalho, exibirá o nome do modelo ajustado. Este modelo pode ser usado como parâmetro na API para solicitações normais. O modelo pode levar alguns minutos para estar operacional. Se as solicitações expirarem, é provável que o modelo ainda esteja sendo carregado. Nesse caso, aguarde alguns minutos e tente novamente (GPT-FINETUNING, 2023).