ESCOLA SALESIANA SÃO JOSÉ
CENTRO PROFISSIONAL DOM BOSCO - CPDB
Av. Almeida Garret, 267 - Jardim Nossa Sra. Auxiliadora, Campinas - SP, 13087-290

Gustavo Alves Pigaiani
Murillo de Souza Germer de Lima

JOY: INTELIGÊNCIA ARTIFICIAL BASEADA NO GPT-3.5 COM INTERFACE GRÁFICA 2D

CAMPINAS
2024

Gustavo Alves Pigaiani
Murillo de Souza Germer de Lima

JOY: INTELIGÊNCIA ARTIFICIAL BASEADA NO GPT-3.5 COM INTERFACE GRÁFICA 2D

Relatório referente ao Projeto de Conclusão de Curso, apresentado ao Centro Profissional Dom Bosco da Escola Salesiana São José, como parte dos requisitos necessários à obtenção do título de Técnico em Informática.

Orientador: Galesandro Henrique Capovilla
Coorientador: Adriana Coelho

Data de início: 06/02/2024
Data final: 27/09/2024

CAMPINAS
2024

Dedicamos este trabalho aos nossos pais pelo estímulo e compreensão; a todos os professores que acreditaram no nosso trabalho; a todos os que, direta ou indiretamente, contribuíram para a conclusão de mais esta etapa.

AGRADECIMENTOS

Gostaríamos de expressar nossa sincera gratidão a todas as pessoas que contribuíram para a realização deste Trabalho de Conclusão de Curso.

Primeiramente, agradecemos aos nossos orientadores, Galesandro Henrique Capovilla e Adriana Coelho, cuja orientação, paciência e expertise foram fundamentais para o desenvolvimento desta pesquisa. Suas sugestões e críticas construtivas foram essenciais para aprimorar este trabalho.

Um agradecimento especial aos nossos amigos, que ofereceram suporte moral e motivacional durante todo o processo.

Não podemos deixar de mencionar nossa família, cuja compreensão e incentivo incondicional foram cruciais para a conclusão deste projeto.

Por fim, agradecemos a todos que, de alguma forma, contribuíram para que este trabalho fosse possível.

Muito obrigado!

“A inteligência artificial pode descobrir soluções para problemas do passado, que mudarão a nossa forma de resolver os mesmos problemas no futuro.”

Stephen Hawking - 2014

RESUMO

Este projeto visa desenvolver uma sofisticada Inteligência Artificial (IA) utilizando a avançada tecnologia do GPT-3.5 da OpenAI, em combinação com uma interface gráfica 2D altamente interativa. O objetivo principal é proporcionar uma interação altamente personalizada e eficiente entre seres humanos e computadores. A solução proposta, denominada Joy, foi criada com o propósito de enfrentar e resolver problemas comuns relacionados aos resultados de pesquisa em navegadores, que frequentemente não oferecem a eficiência ou a precisão desejada pelos usuários. Com a Joy, cada pesquisa realizada pelo usuário pode ser respondida com precisão e agilidade, garantindo que os usuários recebam as informações necessárias de forma rápida e sem obstáculos. A implementação da IA foi realizada utilizando a linguagem de programação Python, tirando proveito das capacidades avançadas de compreensão e resposta oferecidas pelo GPT-3.5. Esse processo envolveu a integração da IA GPT-3.5 em um código Python, onde foram exploradas suas capacidades avançadas de processamento de linguagem natural para interpretar comandos e fornecer respostas relevantes e apropriadas. Além disso, uma interface gráfica 2D foi desenvolvida em um código Python separado, servindo como a representação visual e interativa da IA. Esta interface foi meticulosamente projetada com um foco particular na usabilidade e na criação de uma experiência de conversação mais natural e fluida para os usuários. Elementos de design de personagem e animações foram elaborados com cuidado para garantir que a experiência seja não apenas funcional, mas também memorável e envolvente. Assim, a Joy foi desenvolvida para oferecer suporte e proporcionar uma interação harmônica e eficaz entre o usuário e a tecnologia.

Palavras-chave: Auxiliar. Interface. Interatividade.

ABSTRACT

This project aims to develop sophisticated Artificial Intelligence (AI) using OpenAI's advanced GPT-3.5 technology, in combination with a highly interactive 2D graphical interface. The main objective is to provide a highly personalized and efficient interaction between humans and computers. The proposed solution, called Joy, was created with the purpose of tackling and solving common problems related to search results in browsers, which often do not provide the efficiency or accuracy desired by users. With Joy, each user search can be answered accurately and quickly, ensuring that users receive the information they need quickly and without obstacles. The AI implementation was carried out using the Python programming language, taking advantage of the advanced understanding and response capabilities offered by GPT-3.5. This process involved integrating GPT-3.5 AI into Python code, where its advanced natural language processing capabilities were exploited to interpret commands and provide relevant and appropriate responses. Additionally, a 2D graphical interface was developed in a separate Python code, serving as the visual and interactive representation of the AI. This interface has been meticulously designed with a particular focus on usability and creating a more natural and fluid conversational experience for users. Character design elements and animations have been carefully crafted to ensure the experience is not only functional, but also memorable and engaging. Thus, Joy was developed to offer support and provide a harmonious and effective interaction between the user and technology.

Keywords: Interface. Interative. Assistant.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

CPDB – Centro Profissional Dom Bosco;
IA – Inteligência Artificial;
PLN – Processamento de Linguagem Natural;
GPT – Generative Pre-trained Transformer;
API – Application Programming Interface.

SUMÁRIO

Sumário

1. INTRODUÇÃO

Com os recentes avanços tecnológicos, a inteligência artificial tem desempenhado um papel fundamental em diversas áreas, revolucionando a maneira como as pessoas interagem com os computadores. Com o universo de possibilidades que a IA proporciona, atualmente se vê a necessidade de um atendimento personalizado e mais centrado nas dúvidas dos usuários.

Entretanto, esse projeto foi desenvolvido com base na necessidade de proporcionar ao usuário uma interface amigável de atendimento mais acessível, prestativa e personalizada. A combinação de técnicas de integração da Inteligência Artificial da OpenAI (GPT-3.5), design de interface 2D e reconhecimento de voz proporciona ao usuário uma interação mais natural e satisfatória.

Segundo as conclusões de Keith (2021, p. 23):

> Com o avanço da tecnologia, o uso de ferramentas de inteligência artificial nas empresas está cada vez mais presente em setores que necessitam de um trabalho rápido e eficiente. Diante disso, empresas estão investindo em inteligência artificial para melhorar e otimizar as atividades dos seus setores.

Cada técnica aplicada representa uma maneira de aprimorar a funcionalidade e a aceitação da inteligência artificial, visando não apenas eficiência, mas também a construção de interações mais humanizadas e centradas no usuário. Por meio da exploração dessas hipóteses, foi analisado de forma crítica o impacto dessas soluções na experiência do usuário.

1.1. OBJETIVOS

O objetivo geral deste trabalho foi desenvolver uma interface gráfica 2D avançada, baseada em inteligência artificial, projetada para se adaptar às necessidades específicas de cada usuário. O projeto visou aprimorar significativamente a interação entre o usuário e o computador, garantindo uma experiência mais intuitiva e eficiente. Para alcançar isso, a interface foi desenvolvida com foco na usabilidade e na personalização, integrando tecnologias de ponta para proporcionar respostas rápidas e precisas. Além disso, foram incorporadas

funcionalidades interativas e elementos visuais que facilitam a navegação e tornam a experiência mais envolvente e natural.

- Investigar e implementar técnicas para a implementação da IA referente a OpenAI (GPT-3.5) que será capaz de compreender e responder de maneira eficiente os comandos feitos pelos usuários.
- Explorar e aplicar técnicas avançadas de processamento de linguagem natural para aprimorar a compreensão contextual e a precisão das respostas, garantindo que a interação da inteligência artificial seja mais relevante e adaptada às necessidades do usuário.
- Desenvolver e implementar estratégias inovadoras para criar uma experiência de interação mais envolvente e personalizada, focando na usabilidade e na capacidade de resposta da interface para maximizar a satisfação e a eficiência do usuário.

1.2. JUSTIFICATIVA

Este projeto visa explorar a implementação da inteligência artificial (GPT-3.5) com interface gráfica, reconhecimento de voz e saída de áudio. Esta proposta surge em um momento em que a interação humano-computador se tornou inevitável devido à grande evolução tecnológica, cuja interação está sendo inserida em diversas áreas da sociedade, desde a assistência em tarefas cotidianas e agora, o atendimento ao usuário de maneira interativa. A implementação de uma interface 2D com características humanizadas é, portanto, uma demanda cada vez mais relevante.

A relevância deste estudo ultrapassa o âmbito acadêmico, tendo impacto direto na sociedade e na indústria. A implementação de uma inteligência artificial capaz de compreender e se comunicar de forma natural com os usuários pode melhorar significativamente a experiência do cliente em diversos setores, como comércio, saúde e educação. Além disso, a interface gráfica e a possibilidade de interação por voz com a IA podem aumentar a eficácia da comunicação e promover uma sensação de familiaridade e conforto para os usuários, resultando em interações mais positivas e satisfatórias.

Ao analisar o desenvolvimento tecnológico e as tendências de mercado, é notório que a criação de uma IA personalizada representa uma oportunidade única de inovação e diferenciação para empresas e organizações que buscam oferecer um

atendimento de excelência. O projeto busca impactar positivamente a vida das pessoas ao oferecer soluções inovadoras e acessíveis no campo da interação humano-computador.

Com isso, foi desenvolvido a Joy: uma IA personificada baseada no GPT-3.5 capaz de se comunicar com usuários de acordo com a necessidade apresentada.

2. MATERIAIS E MÉTODOS

2.1. MATERIAIS

Para alcançar os objetivos propostos, foi essencial utilizar um conjunto diversificado de ferramentas e recursos. Um notebook com acesso à internet foi fundamental para a execução e gerenciamento de todo o processo. Utilizamos o Visual Studio Code a fim de escrever e gerenciar tanto quanto o código-fonte, quanto o código da interface. O WampServer foi instalado para criar um ambiente local de servidor web e banco de dados. Para a administração e manipulação de bancos de dados, o MySQL Workbench foi a ferramenta escolhida, facilitando a gestão das bases de dados. O Canva contribuiu significativamente para a criação de teste da interface gráfica 2D. O aplicativo Krita foi fundamental na criação da interface 2D do nosso projeto, proporcionando uma ampla gama de ferramentas e recursos que facilitaram o design e a customização da Joy. Além disso, aproveitamos bibliotecas públicas e recursos dentro das plataformas de desenvolvimento para enriquecer e otimizar o processo de codificação. Essas ferramentas, combinadas, foram fundamentais para a execução bem-sucedida do projeto e o cumprimento dos objetivos estabelecidos.

2.1.1. Requisitos mínimos notebook

Para o desenvolvimento da Joy, foi necessário utilizar um notebook com acesso à internet e atender a certos requisitos mínimos para garantir o bom desempenho do código e das ferramentas de desenvolvimento. O notebook deve possuir um processador Intel Core i5, ou superior, que fornece a capacidade de processamento necessária para lidar com tarefas complexas e operações intensivas. Além disso, é importante que o dispositivo tenha pelo menos 8 GB de RAM para suportar a execução simultânea de múltiplos processos e aplicações sem comprometer o desempenho. O armazenamento de 256 GB em SSD é recomendado para garantir velocidades rápidas de leitura e gravação de dados, o que é crucial para o manejo eficiente de arquivos e a execução ágil de programas. O notebook foi utilizado ao longo de todo o

ciclo de desenvolvimento da Joy, desde a implementação e teste do código até o design e ajustes da interface do usuário, exigindo, portanto, um equipamento que ofereça robustez e confiabilidade durante todas as etapas do projeto.

2.1.2. Plataforma da OpenAI

A plataforma da OpenAI para desenvolvedores é um conjunto de ferramentas e APIs projetadas para integrar capacidades avançadas de inteligência artificial em aplicações e serviços. Esta plataforma oferece acesso a modelos de linguagem avançados, como o GPT-3.5, que podem realizar tarefas como geração de texto, compreensão de linguagem natural, tradução e até mesmo resposta a perguntas complexas. Através de uma interface de programação de aplicativos (API), os desenvolvedores podem incorporar essas funcionalidades em seus próprios aplicativos, sites e sistemas, possibilitando a criação de soluções inteligentes que entendem e interagem com os usuários de maneira mais natural e eficaz.

Para ajudar os desenvolvedores a experimentarem e começarem a utilizar a API, a OpenAI oferece um nível gratuito com alguns limites. Este nível gratuito permite que os desenvolvedores realizem um número limitado de chamadas à API sem custos, o que é ideal para testes iniciais e pequenas implementações. No entanto, dependendo da complexidade e da quantidade de uso, esses limites podem ser atingidos rapidamente.

Para esse projeto, para poder utilizar a API de forma mais abrangente e realizar diversas chamadas para testes e desenvolvimento, precisamos adicionar um saldo de 5 dólares à conta. Esse crédito adicional permitiu que superássemos as restrições do nível gratuito e trabalhasse de maneira mais extensiva com as funcionalidades da API, explorando suas capacidades e implementando soluções mais complexas conforme necessário.

2.1.3. Inicializar banco de dados localmente (WampServer)

O WampServer é uma plataforma de desenvolvimento amplamente utilizada que permite criar e gerenciar ambientes de desenvolvimento web localmente em sistemas operacionais Windows. Utilizando o WampServer, é possível instalar e configurar um banco de dados MySQL. Isso é particularmente útil para desenvolvedores que precisam configurar e testar aplicações web em um ambiente

local antes de implantá-las em um servidor remoto. Ao usar o WampServer, você pode criar e manipular bancos de dados MySQL de forma eficiente, facilitando o desenvolvimento, teste e depuração de aplicativos que dependem de uma base de dados. Com uma interface gráfica intuitiva, o WampServer simplifica a configuração e o gerenciamento do banco de dados local, proporcionando uma solução prática e eficaz para trabalhar com bancos de dados em projetos de desenvolvimento web.

Para conectar o banco de dados ao código-fonte da aplicação, foi necessária a instalação do WampServer no notebook. Uma vez instalado, o usuário deve inicializar a aplicação, o que ativa os serviços necessários para o funcionamento do servidor web e do banco de dados. Com isso, o código em Python pode se comunicar diretamente com o MySQL utilizando bibliotecas apropriadas, como o mysql-connector ou SQLAlchemy. Essa comunicação permite que os dados, como nomes de usuários e comandos fornecidos, sejam armazenados e recuperados de maneira eficiente.

2.1.4. Banco de dados (MySQL Workbench)

O MySQL Workbench é uma ferramenta robusta e intuitiva desenvolvida pela Oracle para a criação, design e administração de bancos de dados MySQL. Oferecendo uma interface gráfica de fácil uso, o Workbench permite aos desenvolvedores e administradores de banco de dados realizar uma ampla gama de tarefas de forma eficiente. Com ele, é possível criar e modificar estruturas de banco de dados, desenvolver consultas SQL, gerenciar dados e realizar operações de manutenção de forma visual e interativa. Seja para iniciantes que estão começando a trabalhar com MySQL ou para profissionais experientes que precisam de uma solução abrangente para suas necessidades de banco de dados, o MySQL Workbench oferece as funcionalidades necessárias para tornar o processo de desenvolvimento e administração eficientes.

Para o desenvolvimento do projeto, o MySQL Workbench foi utilizado como ferramenta para a administração do banco de dados MySQL que armazena e gerencia os comandos e prompts da assistente virtual. O Workbench facilitou a criação e modificação da estrutura do banco de dados, permitindo a definição das tabelas e campos necessários para armazenar as informações de entrada do usuário, como os nomes e comandos. Além disso, o Workbench foi fundamental para a execução de

consultas SQL e a manipulação dos dados armazenados. Isso foi essencial para garantir que pudéssemos armazenar, recuperar e atualizar comandos de forma eficiente, possibilitando uma interação fluida e eficaz com o banco de dados. O uso do MySQL Workbench, portanto, foi crucial para estabelecer uma base sólida e organizada para o funcionamento do sistema de assistente virtual.

Para esse projeto, foi criado um banco de dados denominado "banco_joy" utilizando o MySQL Workbench. Dentro deste banco de dados, foi estabelecida uma tabela chamada "prompts", que possui três colunas: "id", "nome" e "comando". A coluna "id" é destinada a armazenar um identificador único para cada registro, enquanto "nome" armazena o nome associado ao prompt, e "comando" contém o texto ou comando específico relacionado. Este esquema foi projetado para organizar e gerenciar dados de maneira eficiente, facilitando consultas e manutenção através da interface do Workbench.

2.1.5. Desenvolvimento do código principal

Este código foi desenvolvido para criar a assistente virtual Joy, que utiliza várias tecnologias e bibliotecas para proporcionar uma experiência interativa, baseada em inteligência artificial (IA) e processamento de linguagem natural (PLN). A Joy é projetada para receber comandos de voz ou texto, processar esses comandos com a IA GPT-3.5 da OpenAI, e responder de forma clara e interativa, tanto através de texto quanto de fala. A interface gráfica (GUI) serve como um meio de interação visual, onde o usuário pode inserir comandos. Na imagem abaixo estão todas as bibliotecas necessárias:

Figura 1 - Bibliotecas utilizadas no código fonte

```
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import openai
import speech_recognition as sr
import whisper
import pyttsx3
import os
import time
import mysql.connector
```

O código utiliza uma variedade de bibliotecas para construir uma assistente virtual multifuncional. Tkinter é a biblioteca padrão para criar interfaces gráficas em Python, permitindo a construção de janelas e componentes interativos. Messagebox é um módulo dentro do Tkinter especializado na exibição de caixas de mensagens, como alertas e notificações. OpenAI é a biblioteca responsável pela interação com a API da OpenAI, proporcionando acesso a modelos avançados de linguagem. Speech_recognition é uma biblioteca dedicada ao reconhecimento e conversão de fala em texto, essencial para a funcionalidade de entrada por voz. Whisper é outra ferramenta da OpenAI, utilizada especificamente para transcrição de áudio, convertendo falas gravadas em texto escrito. Pyttsx3 oferece funcionalidades para conversão de texto em fala, permitindo que o assistente responda verbalmente. O módulo OS é utilizado para interagir com o sistema operacional, como a execução de comandos e manipulação de arquivos. Time é uma biblioteca para manipulação de tempo, incluindo funções para pausar a execução do código. Por fim, mysql.connector é uma biblioteca que permite a conexão e interação com bancos de dados MySQL, essencial para o armazenamento e consulta de dados.

Para conectar e usar a API da OpenAI em um código Python, é necessário seguir alguns passos. Primeiro é preciso importar a biblioteca OpenAI, na qual facilita a interação com a API. Acesse a conta no site da OpenAI e navegue até a seção de API para obter a chave que autenticará as solicitações a API. No código, a “openai.api_key” é uma chave de autenticação utilizada para acessar os serviços da OpenAI, permitindo que o aplicativo interaja com modelos avançados de inteligência artificial fornecidos pela plataforma, como o GPT-3.5. Esta chave é crucial para autenticar as requisições feitas à API e garantir que o acesso aos recursos da OpenAI seja seguro e autorizado. No código Python, é preciso configurar a chave da API para que a biblioteca OpenAI possa autenticar as solicitações. A maneira mais fácil de fazer isso é definindo a chave diretamente no código. Veja o exemplo a seguir:

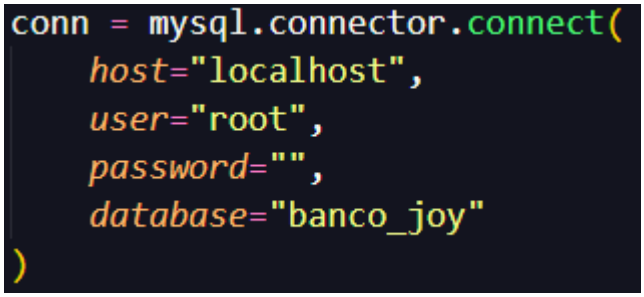
Figura 2 - Declaração da chave API

O código trabalha com arquivos de estado (state.txt) e de status (status.txt). Esses arquivos são importantes para controlar o comportamento da interface, indicando quando a assistente está em repouso (idle) ou ativa, ouvindo e processando

comandos. O programa verifica se os arquivos existem e, dependendo do resultado, seleciona o caminho correto para acessá-los. Esses arquivos servem para coordenar as mudanças de animação ou de visualização dentro da interface gráfica, como mudar a expressão da assistente quando ela está respondendo ou esperando uma entrada do usuário.

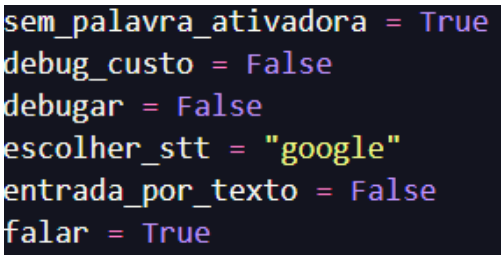
O código também se conecta a um banco de dados MySQL chamado banco_joy, que armazena uma coleção de comandos. O usuário pode adicionar novos comandos através da interface gráfica, e esses comandos são armazenados no banco de dados para serem reutilizados posteriormente. O sistema verifica se um comando com o mesmo nome já existe antes de adicioná-lo, prevenindo duplicidades. Isso é útil para criar uma biblioteca de comandos personalizados que podem ser acessados e executados durante a interação com a assistente. Observe a conexão feita com o banco de dados:

Figura 3 - Conexão com o banco de dados



Dentro do código, foram criadas algumas variáveis para melhor gerenciamento. Veja a imagem abaixo:

Figura 4 - Principais variáveis

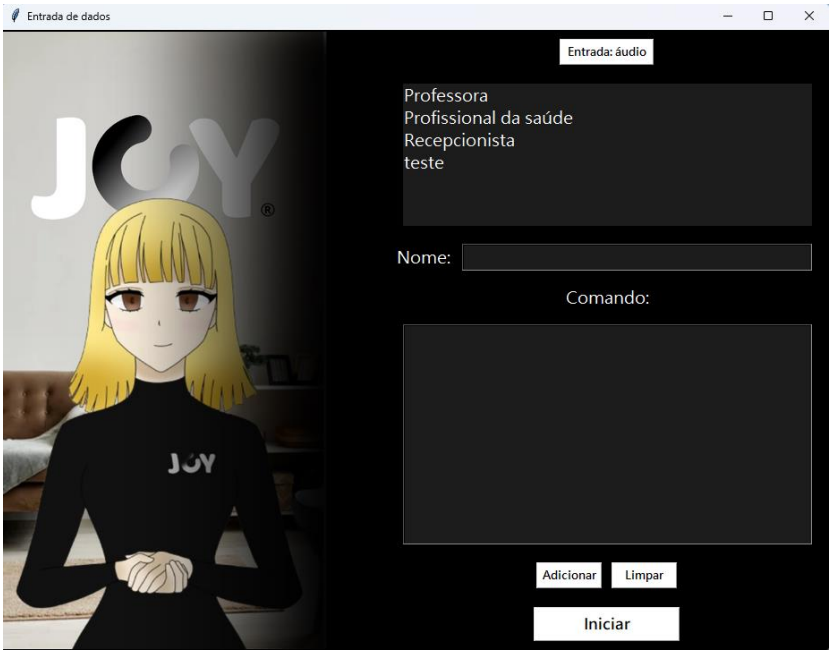


A variável “sem_palavra_ativadora” determina se o assistente virtual deve exigir uma palavra ativadora específica para iniciar o processo de escuta e resposta, permitindo um controle mais flexível sobre quando o assistente deve ativar sua

funcionalidade de reconhecimento de voz. “Debug_custo” é uma configuração que controla a exibição dos custos associados ao uso da API, ajudando na depuração ao fornecer informações sobre o consumo de recursos. “Debugar” serve como um controle geral para ativar ou desativar funcionalidades de depuração no código, facilitando o processo de identificação e resolução de problemas. “Escolher_stt” permite a seleção do serviço de reconhecimento de fala (Speech-to-Text) que será utilizado, como o Google ou outro fornecedor, influenciando a precisão e a qualidade do reconhecimento de voz. “Entrada_por_texto” define se o usuário interage com o assistente através de texto digitado, em vez de voz, oferecendo uma alternativa para diferentes cenários de uso. Finalmente, “falar” controla se o assistente deve responder verbalmente ou apenas fornecer respostas textuais, permitindo ajustes na forma como a interação com o usuário ocorre.

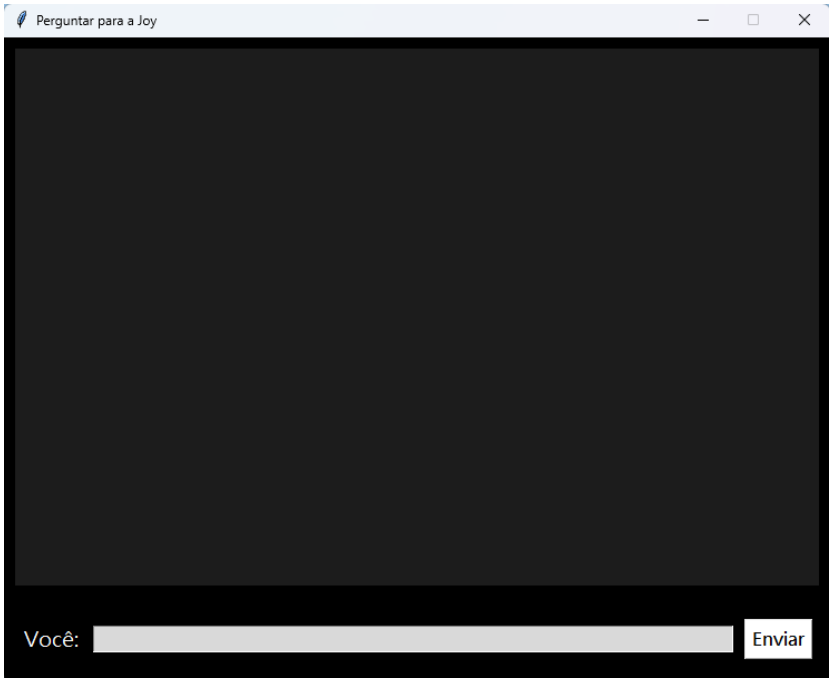
A interface gráfica, criada com o Tkinter, é a parte central da interação do usuário. Ela exibe uma janela onde o usuário pode visualizar e manipular comandos. Essa interface permite ao usuário digitar um comando personalizado e um nome associado a ele. Se estiver vazio, o sistema alerta com uma mensagem de erro; adicionar comandos ao banco de dados, o usuário pode adicioná-lo ao banco de dados, e o sistema automaticamente lista os comandos armazenados na interface; selecionar um comando existente de uma lista e ver seu conteúdo no campo de texto para edição. Isso oferece uma forma dinâmica de atualizar ou usar comandos antigos. Veja abaixo a interface:

Figura 5 - Janela do usuário



Além disso, há um botão de alternância que permite ao usuário escolher entre entrada por texto ou entrada por voz, facilitando a transição entre diferentes modos de interação. Caso o usuário escolha a opção de entrada por texto, a janela abaixo é executada:

Figura 6 - Interface de entrada de comandos



O coração deste projeto é o uso do modelo de IA GPT-3.5, que é acessado por meio de uma chave de API fornecida pela OpenAI. O código permite que o usuário envie mensagens para a IA e receba respostas baseadas no contexto da conversa. A função generate_answer() usa esse modelo para processar as mensagens enviadas pelo usuário. O sistema mantém um histórico de conversas, permitindo que a Joy responda de maneira contínua e contextualizada, simulando uma conversa fluida. Veja abaixo a função generate_answer():

Figura 7 - Função de gerar respostas

```
def generate_answer(messages):
    response = openai.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=messages,
        temperature=0.5
    )
    return [response.choices[0].message.content, response.usage]
```

Sempre que o usuário envia uma pergunta (seja por texto ou voz), a IA processa essa entrada e gera uma resposta. As respostas não são apenas exibidas na interface, mas também podem ser faladas pela Joy usando a biblioteca pyttsx3, criando uma sensação mais interativa e humanizada.

O código permite que o usuário utilize comandos de voz por meio da integração com a biblioteca speech_recognition e o modelo Whisper. Quando o usuário opta por esse modo, o microfone é ativado, e a fala do usuário é capturada e processada para ser convertida em texto. Esse texto é então tratado da mesma forma que uma entrada de texto, sendo enviado para o modelo GPT-3.5 para gerar uma resposta.

Além disso, o sistema ajusta o reconhecimento de fala ao ambiente, calibrando automaticamente o ruído de fundo, o que melhora a precisão da transcrição e a eficiência na captura de comandos de voz. Isso é importante em situações em que o ambiente pode ter interferências sonoras, garantindo que o sistema funcione de maneira eficiente.

Após gerar a resposta, o sistema utiliza a biblioteca pyttsx3 para que a Joy fale a resposta em voz alta. Isso melhora a experiência do usuário ao criar uma comunicação mais imersiva, onde a assistente virtual não só responde, mas também verbaliza suas respostas.

Durante toda a interação, o código alterna entre diferentes estados. O arquivo state.txt é constantemente atualizado para refletir o estado atual da assistente (idle ou talking), o que pode ser utilizado para alterar animações ou expressões visuais da Joy. Da mesma forma, o arquivo status.txt registra o que está acontecendo em tempo real, como "Estou ouvindo" ou "Processando", criando feedback contínuo para o usuário.

A alternância entre esses estados torna o assistente mais dinâmico e interativo, simulando uma conversa "real" onde o assistente parece estar ativamente escutando e respondendo, em vez de apenas receber comandos de forma passiva.

O código entra em um loop contínuo, onde ele fica constantemente ouvindo o usuário ou aguardando entradas via texto. Dependendo do modo selecionado (texto ou voz), o sistema ajusta seu comportamento. O loop é projetado para permanecer ativo até que o usuário solicite explicitamente o encerramento com comandos como "assistente desligar". Durante esse ciclo, o sistema processa as entradas, gera as respostas e, se habilitado, fala as respostas.

O sistema é projetado para ser flexível, permitindo ao usuário alternar entre diferentes modos de entrada e personalizar os comandos. A integração com o banco

de dados permite que o sistema seja facilmente expandido, onde o usuário pode criar novos comandos e armazená-los para uso posterior. O uso de GPT-3.5 garante que as respostas sejam contextuais, adaptando-se às necessidades do usuário com base em conversas passadas.

2.1.6. Desenvolvimento da interface de teste da Joy (Canva)

O Canva é uma plataforma online de design gráfico amplamente utilizada que facilita a criação de uma vasta gama de materiais visuais, desde apresentações e cartazes até posts para redes sociais e materiais promocionais. Com sua interface intuitiva e recursos de arrastar e soltar, o Canva permite que usuários de todos os níveis de experiência criem designs profissionais sem a necessidade de habilidades avançadas em design. Oferecendo uma vasta biblioteca de modelos, imagens, ícones e fontes, o Canva é uma ferramenta poderosa para quem busca criar visuais atraentes de maneira rápida e eficiente.

A fim de testes, utilizamos um asset gratuito obtido na Unity Asset Store como base para a criação da interface 2D. Após o download das imagens, modelamos o asset conforme as necessidades, ajustando e personalizando-o para melhor atender aos requisitos do projeto. Foram criadas 6 imagens da personagem com expressões diferentes, para que a animação fosse feita. Veja as imagens criadas abaixo:

Figura 8 - Frames de teste

Esses frames foram criados para fim de testes antes de implementarmos a interface final, feita no Krita.

2.1.7. Desenvolvimento do código da interface

O código de desenvolvimento da interface utiliza diversas bibliotecas essenciais para criar uma interface gráfica e manipular imagens em Python. Veja abaixo:

Figura 9 - Bibliotecas utilizadas no código da interface

```
import tkinter as tk
from itertools import cycle
from PIL import Image, ImageTk
import os
```

A biblioteca “tkinter” como mencionado anteriormente, é a ferramenta padrão para desenvolver interfaces gráficas, proporcionando uma forma intuitiva de criar janelas e widgets. A função “cycle” da biblioteca “itertools” é utilizada para criar um iterador que percorre uma lista de imagens de maneira infinita, facilitando a animação contínua. A biblioteca “Pillow”, com seus módulos “Image” e “ImageTk”, é empregada para manipular e converter imagens; “Image” permite abrir arquivos de imagem, enquanto “ImageTk” adapta essas imagens para o formato adequado à exibição em widgets do “tkinter”. Por fim, a biblioteca “os” é empregada para interações com o sistema operacional, como verificar a existência de arquivos, essencial para gerenciar o estado e controle da animação baseada em arquivos externos.

Este código cria uma interface gráfica para a assistente virtual Joy, com o foco em animações que refletem seu estado, como "idle" (parada) ou "talking" (falando). O objetivo principal é exibir animações fluidas na interface com base no comportamento da assistente, simulando expressões visuais conforme ela interage com o usuário.

O código começa com a definição de arquivos que armazenam o estado e o status da assistente. O estado (state.txt) determina se a Joy está falando ou parada (idle), enquanto o status (status.txt) contém informações de feedback para o usuário, como mensagens indicando que a assistente está ouvindo ou processando uma resposta. Esses arquivos são utilizados para sincronizar as animações e fornecer atualizações em tempo real sobre a interação.

O sistema verifica qual arquivo de estado ou status está disponível no computador e ajusta dinamicamente seu comportamento com base nos arquivos encontrados. Isso garante que o programa funcione corretamente em diferentes ambientes de desenvolvimento.

As funções change_state e change_status permitem alterar e salvar o estado e o status da Joy. Essas funções atualizam os arquivos correspondentes, registrando se a assistente está em modo idle ou falando e ajustando as animações de acordo. O status, por outro lado, é usado para exibir mensagens na interface, fornecendo feedback visual sobre o que a assistente está fazendo naquele momento (como "estou ouvindo" ou "processando").

O código carrega uma série de imagens que serão usadas para as animações da Joy. A função carregar_imagens é responsável por carregar e redimensionar essas imagens para que se ajustem à interface. As imagens correspondem às diferentes expressões e ações da Joy, como falar, piscar ou mudar de postura.

Essas imagens são agrupadas em três ciclos principais: Idle, quando a Joy está em repouso, suas expressões faciais são mais neutras; Talking, quando a Joy está falando, há uma série de imagens que mostram sua boca se movendo; E as transições, conjuntos de imagens que representam a transição entre os estados de idle para talking e vice-versa. Essas transições ajudam a suavizar a mudança de uma ação para outra, criando uma experiência visual mais fluida.

O código usa a função animar_imagens para exibir as animações baseadas no estado da assistente. Ele utiliza o conceito de ciclo infinito, onde as imagens são mostradas em sequência, e, quando a sequência termina, o ciclo recomeça. Isso simula a ideia de uma animação contínua e repetitiva, o que é fundamental para criar a ilusão de que a Joy está ativa. Existem dois ciclos de animações principais: ciclo idle, exibe imagens que mostram a Joy em repouso; Ciclo talking, mostra imagens da Joy movendo a boca, simulando uma fala.

Cada ciclo é controlado por uma lógica que alterna entre essas imagens de acordo com o estado atual da Joy. Se o estado mudar de idle para talking, o código realiza uma transição entre as animações de forma suave, utilizando imagens intermediárias que mostram uma mudança gradual de uma posição para outra.

O código implementa transições para quando a Joy muda de estado. As funções de transição, controladas pela função tocar_transicao, exibem uma série de imagens que fazem a transição visual entre as ações de falar e parar. Isso aumenta o realismo da animação, criando uma sensação de que a Joy está se movendo naturalmente de um estado para outro. Por exemplo, ao mudar de idle para talking, há uma sequência de imagens que simula a Joy preparando-se para falar, e o contrário também ocorre quando ela termina de falar.

A interface gráfica é construída usando Tkinter e é responsável por exibir a Joy e as informações de status. A janela da interface é centralizada na tela, e sua resolução e proporções são ajustadas para garantir que as animações e os elementos gráficos se encaixem bem no espaço disponível.

1. Janela principal: A interface gráfica abre uma janela que exibe a animação da Joy e as mensagens de status.
2. Imagem da Joy: As imagens da Joy são exibidas em um widget Label que é constantemente atualizado com novas imagens conforme o estado da animação muda.
3. Status: Uma área na interface exibe o status atual da Joy, como "Estou ouvindo" ou "Processando", que é atualizado a cada segundo, mantendo o usuário informado sobre o que está acontecendo no fundo.

O sistema usa a função janela.after(), que funciona como um temporizador para atualizar as animações e o status em intervalos regulares. O delay entre as atualizações pode ser ajustado, o que permite controlar a velocidade das animações e das transições.

1. Delay de transição: Controla a velocidade com que as animações de transição entre idle e talking são executadas.
2. Delay principal: Controla o tempo de exibição de cada frame nas animações idle e talking.

Essa estrutura de temporização permite que a animação seja fluida, com mudanças suaves de estado e transições visuais entre diferentes animações.

2.1.8. Desenvolvimento da interface final (Krita)

Com uma interface intuitiva e uma vasta gama de ferramentas e pincéis personalizáveis, o Krita oferece uma experiência poderosa e flexível para criar arte digital. Entre seus pontos positivos, destaca-se a capacidade de trabalhar com camadas e máscaras, o suporte para tablets gráficos e a vasta biblioteca de pincéis que permite uma personalização detalhada. Além disso, o Krita é gratuito, o que o torna acessível a todos, e possui uma comunidade ativa que contribui com tutoriais e plugins, enriquecendo ainda mais suas funcionalidades.

No desenvolvimento da interface final para o projeto, foi realizado o desenho da Joy, a assistente IA utilizando a funcionalidade de camadas do Krita. O processo

envolveu a criação de um modelo detalhado da personagem, aproveitando o pincel padrão para garantir consistência e qualidade na arte. Para a animação, foram exportados 20 desenhos da Joy em diferentes posições, permitindo a utilização desses frames no programa Python. Esses frames foram então integrados ao Tkinter para a criação de uma animação fluida e visualmente coesa.

O uso do Krita para este propósito demonstrou sua eficácia na gestão de projetos de arte complexos, proporcionando as ferramentas necessárias para criar e organizar múltiplas camadas de ilustrações. A capacidade de exportar imagens em diferentes formatos e a integração com outras ferramentas de programação confirmaram o Krita como uma escolha ideal para o desenvolvimento de animações e interfaces visuais. Este fluxo de trabalho não apenas facilitou a criação de animações de alta qualidade, mas também evidenciou a flexibilidade do Krita em se adaptar a diferentes etapas do processo de design.

O processo começou com o esboço da estrutura base, no qual foram definidas as proporções da personagem e o estilo desejado, mantendo uma aparência amigável e acessível para a assistente virtual. Em seguida, os contornos foram refinados, utilizando pinceis no Krita para garantir um traçado nítido e consistente.

Após a criação dos 20 desenhos que compõem os frames da animação, cada imagem foi exportada em png com alta qualidade, mantendo a fidelidade das linhas. A animação foi então ajustada no Python com o uso de Tkinter, onde se pôde animar os movimentos dos braços e a expressão facial de Joy ao falar e piscar, criando uma interação mais humanizada e dinâmica. Cada movimento foi projetado para ser suave, usando intervalos precisos entre os frames para garantir uma transição fluida. Veja os frames principais criados:

Figura 10 - Frame Joy normal

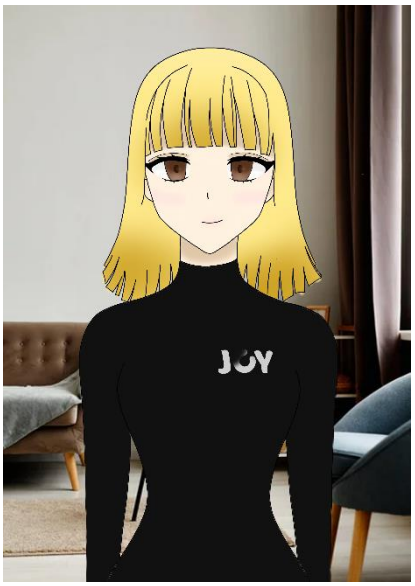


Figura 11 - Frame Joy piscando

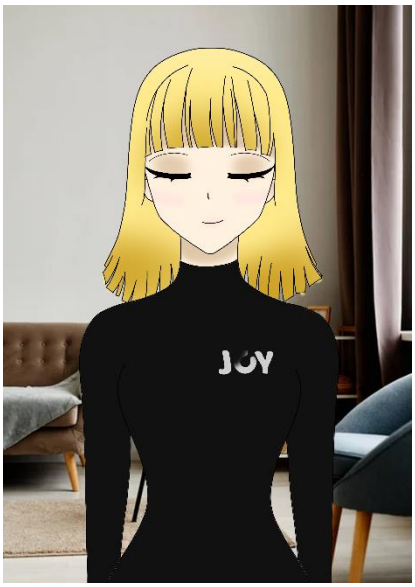
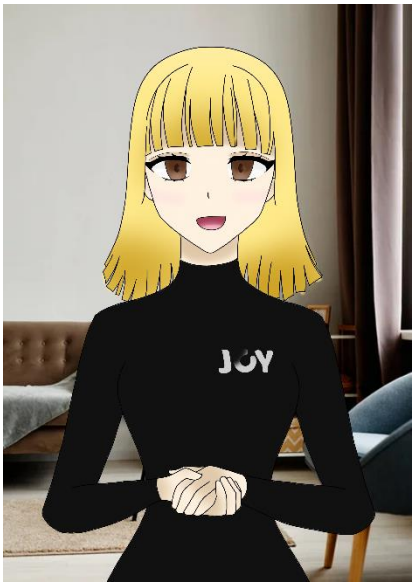


Figura 12 - Frame Joy falando



Além disso, a possibilidade de editar camadas individualmente no Krita permitiu ajustes rápidos em pequenos detalhes, como a posição das mãos e o ângulo dos olhos, sem a necessidade de redesenhar a personagem inteira. Isso otimizou o processo de animação, proporcionando um controle maior sobre os aspectos visuais sem comprometer o tempo de produção.

2.1.9. Integração do código do GPT-3.5 com o código da interface 2D

O código começa com a importação de diversas bibliotecas essenciais para o funcionamento do aplicativo. A biblioteca tkinter é importada para criar a interface gráfica, permitindo a construção de janelas, botões e outros elementos visuais. A Pillow é usada através das funções Image e ImageTk para manipular imagens, como abrir e redimensionar arquivos de imagem para exibição na interface. A biblioteca subprocess é incluída para possibilitar a execução e gerenciamento de outros scripts ou programas em processos separados. Por outro lado, as bibliotecas os e signal são importadas para interação com o sistema operacional e manipulação de sinais, embora não sejam diretamente utilizadas no código apresentado. Veja:

Figura 13 - Bibliotecas utilizadas no código de integração

```
import tkinter as tk
from PIL import Image, ImageTk
import subprocess
import os
import signal
```

Este código cria uma interface gráfica simples para controlar a execução de dois scripts Python diferentes, chamados "chat_v9.py" e "joy_v2.py", que são iniciados simultaneamente ao clicar no botão iniciar. A interface gráfica foi construída usando a biblioteca Tkinter, e o código também utiliza o módulo subprocess para gerenciar os processos desses scripts em segundo plano. O código se concentra em facilitar o início e o encerramento dos scripts, enquanto fornece uma interface visual com botões e uma imagem centralizada.

A função principal deste código é gerenciar a execução de dois scripts Python usando subprocess. A ideia central é fornecer uma forma conveniente de iniciar e encerrar os scripts diretamente através da interface gráfica.

A lista processos armazena os processos que são iniciados. Quando o usuário clica no botão de "Entrar", dois subprocessos são iniciados para rodar os scripts "chat_v9.py" e "joy_v2.py". Ambos os scripts são executados em segundo plano, permitindo que a interface gráfica permaneça ativa e funcional enquanto esses scripts são executados.

A função sair encerra os subprocessos que foram iniciados. Ela percorre a lista de processos e tenta finalizar cada um, chamando o método terminate() para garantir que os scripts sejam encerrados corretamente. Caso ocorra algum erro ao tentar terminar um processo, uma mensagem de erro é exibida no console, permitindo que o desenvolvedor saiba que algo deu errado.

A interface gráfica consiste em uma janela centralizada que contém a logo da Joy e dois botões principais: Entrar e Sair. A janela é configurada com um título ("Joy") e tem um fundo preto. A janela também é fixada em um tamanho específico, com a opção resizable(False, False), o que impede o usuário de redimensioná-la. A função on_closing() é registrada para o evento de fechamento da janela. Esta função está vazia, o que significa que, para o usuário sair da aplicação deve necessariamente clicar no botão de sair.

Existem dois botões na interface, ambos inseridos em um frame dedicado para organizá-los horizontalmente. O botão entrar quando clicado, chama a função iniciar_scripts, que inicia os dois scripts Python mencionados. Esses scripts começam a ser executados em segundo plano, permitindo que o usuário interaja com a assistente ou outro sistema implementado nesses scripts. Já o botão sair, quando clicado, chama a função sair, que finaliza todos os subprocessos que estão em execução e encerra a interface gráfica. O método root.quit() é chamado para fechar a

janela principal e parar o loop do Tkinter. Veja essa janela que é responsável pela inicialização dos dois scripts:

Figura 14 - Janela de inicialização da Joy

O uso do módulo subprocess permite que os scripts Python sejam iniciados sem bloquear a execução do código principal da interface. A função Popen do subprocess é usada para criar um novo processo que roda o script correspondente. Ao iniciar os subprocessos dessa maneira, a interface Tkinter continua responsiva, enquanto os scripts executam suas respectivas funções em segundo plano.

O código armazena os objetos de processo retornados pela função Popen na lista processos. Isso é importante porque ao encerrar a aplicação, o código precisa saber quais processos foram iniciados para garantir que todos eles sejam corretamente encerrados antes da finalização da interface.

2.2. MÉTODOS

O nosso projeto foi desenvolvido utilizando o método de engenharia, o qual estruturou todo o processo de criação e implementação de forma eficiente. Este método envolveu a definição clara dos requisitos do sistema e o planejamento detalhado das etapas de desenvolvimento. Iniciamos com uma fase de análise para entender as necessidades do usuário e os objetivos do projeto, seguida pelo design detalhado do sistema, incluindo a seleção de ferramentas e tecnologias apropriadas. Durante a fase de desenvolvimento, adotamos práticas para codificar e integrar os componentes do sistema, assegurando a conformidade com os requisitos estabelecidos. Finalmente, realizamos testes extensivos para validar o sistema e ajustar qualquer discrepância. Este método de engenharia não apenas forneceu uma

base sólida para a execução do projeto, mas também garantiu que o produto fosse robusto, eficiente atendesse às expectativas dos usuários.

3. FUNDAMENTAÇÃO TEÓRICA

3.1. Definição e Evolução da Inteligência Artificial

A Inteligência Artificial (IA) é um campo da ciência da computação dedicado ao desenvolvimento de sistemas que simulam capacidades humanas, como aprendizado, raciocínio e tomada de decisões (Russell & Norvig, 2016). Inicialmente, a IA começou com sistemas baseados em regras, que operavam segundo um conjunto fixo de instruções. Esses sistemas eram limitados em flexibilidade e adaptabilidade. A evolução da IA levou à introdução de técnicas mais avançadas, como o aprendizado de máquina (machine learning) e as redes neurais profundas.

O aprendizado de máquina, um subcampo da IA, envolve a construção de algoritmos que podem aprender com e fazer previsões baseadas em dados. As redes neurais profundas, uma extensão do aprendizado de máquina, utilizam camadas de neurônios artificiais para modelar padrões complexos em grandes volumes de dados (LeCun, Bengio, & Hinton, 2015). Essas tecnologias avançadas possibilitaram a criação de sistemas de IA mais sofisticados e capazes, como o GPT-3.5, que demonstra a capacidade da IA de gerar e compreender linguagem de forma altamente eficaz.

3.2. Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é uma área da IA que se concentra na interação entre computadores e linguagem humana. O objetivo do PLN é permitir que os sistemas computacionais compreendam, interpretem e respondam a linguagem humana de uma forma que seja tanto significativa quanto útil. A evolução do PLN tem sido marcada por avanços significativos na modelagem de linguagem, impulsionados pelo desenvolvimento de modelos baseados em redes neurais profundas, como os Modelos de Linguagem de Transformadores (Vaswani et al., 2017).

Os transformadores, uma arquitetura introduzida por Vaswani et al. (2017), revolucionaram o PLN ao permitir que os modelos capturassem relações contextuais mais complexas e de longo alcance em texto. Esses modelos, ao invés de processar

palavras individualmente, consideram a totalidade do contexto para gerar respostas mais coerentes e relevantes. A introdução dos transformadores possibilitou a criação de modelos de linguagem avançados, como o GPT-3.5, que são capazes de gerar texto com qualidade humana e realizar tarefas complexas de compreensão de linguagem.

3.3. Arquitetura dos Modelos de Transformadores

Os modelos de transformadores, descritos por Vaswani et al. (2017), introduziram uma nova abordagem para o processamento de linguagem ao substituir as redes neurais recorrentes (RNNs) e as redes neurais convolucionais (CNNs) por um mecanismo de atenção. O GPT-3.5, uma versão avançada do GPT-3, utiliza essa arquitetura com uma quantidade significativa de parâmetros (175 bilhões), permitindo que o modelo compreenda e gere texto com uma qualidade e precisão excepcionais (Brown et al., 2020).

A arquitetura dos transformadores é baseada em mecanismos de atenção que permitem ao modelo focar em diferentes partes do texto ao mesmo tempo, capturando nuances e contextos variados. Esse mecanismo melhora a capacidade do modelo de gerar texto relevante e coerente em diversos contextos e aplicações, desde chatbots até ferramentas de criação de conteúdo e assistentes virtuais.

3.4. Aplicações do GPT-3.5

O GPT-3.5 tem sido amplamente adotado em várias aplicações devido à sua habilidade de gerar texto que imita a linguagem humana de maneira convincente. Entre as suas aplicações estão os chatbots, assistentes virtuais e ferramentas de geração de conteúdo. A capacidade do GPT-3.5 de entender e gerar texto em linguagem natural o torna ideal para criar interfaces interativas que proporcionam uma experiência mais envolvente e personalizada para os usuários (Amos, 2024).

Além disso, o GPT-3.5 é utilizado em aplicações que exigem uma compreensão profunda do contexto e a capacidade de gerar respostas contextualizadas e naturais. Isso inclui a criação de sistemas de suporte ao cliente automatizados, onde o modelo pode responder a perguntas complexas e lidar com uma variedade de consultas de forma eficaz.

3.5. Importância das Interfaces Gráficas em Sistemas de IA

As interfaces gráficas são essenciais para facilitar a interação entre usuários e sistemas de IA, especialmente quando se trata de sistemas que requerem feedback visual e interação dinâmica. Uma interface gráfica bem projetada pode melhorar significativamente a usabilidade e a experiência do usuário ao fornecer uma representação visual clara e intuitiva das funcionalidades do sistema. Interfaces gráficas eficientes ajudam a tornar a interação com sistemas complexos mais acessível e menos intimidante para os usuários, promovendo uma melhor compreensão e utilização das capacidades da IA (Shneiderman & Plaisant, 2010).

3.6. Ferramentas e Tecnologias para Interfaces Gráficas 2D

Para o desenvolvimento de interfaces gráficas 2D, diversas ferramentas e tecnologias estão disponíveis. O Tkinter, por exemplo, é uma biblioteca padrão para a criação de interfaces gráficas em Python, que permite a criação de interfaces simples e funcionais. Frameworks mais avançados, como o Qt, oferecem uma gama mais ampla de funcionalidades e uma maior flexibilidade para a criação de interfaces interativas e visuais mais complexas (Dahl & Collins, 2020).

Essas ferramentas permitem a criação de interfaces que podem interagir de forma eficaz com os modelos de linguagem baseados em IA, como o GPT-3.5. A escolha da ferramenta adequada depende das necessidades específicas do projeto e das funcionalidades desejadas na interface gráfica.

3.7. Desafios e Soluções na Integração

Integrar o GPT-3.5 com uma interface gráfica 2D apresenta vários desafios, incluindo a gestão eficiente do fluxo de dados entre o modelo de linguagem e a interface visual. A comunicação entre o backend (onde o GPT-3.5 opera) e o frontend (a interface gráfica) deve ser bem coordenada para garantir uma interação fluida e responsiva. Soluções para esses desafios incluem o uso de APIs que facilitam a comunicação entre o modelo de linguagem e a interface gráfica, e a implementação de práticas de design que otimizam a experiência do usuário (Gonzalez et al., 2021).

Além disso, é crucial garantir que a interface gráfica seja intuitiva e capaz de lidar com as respostas geradas pelo GPT-3.5 de maneira eficaz. Isso pode envolver a

criação de fluxos de trabalho que permitem ao usuário interagir facilmente com o sistema e compreender as respostas fornecidas pelo modelo.

3.8. Estudos de Caso e Exemplos

Estudos de caso que exploram a integração de modelos de linguagem com interfaces gráficas demonstram a eficácia dessa abordagem em diferentes contextos. Por exemplo, chatbots interativos com interfaces gráficas têm sido implementados em ambientes corporativos e educacionais para melhorar o suporte ao cliente e a interação com os usuários. Esses estudos fornecem insights valiosos sobre as melhores práticas para a implementação bem-sucedida de sistemas que combinam modelos avançados de linguagem com interfaces visuais (Pereira, 2021; Schunk, 2020).

3.9. Conclusão

Com base na fundamentação teórica apresentada, podemos concluir que o desenvolvimento da IA Joy integra conceitos avançados de Inteligência Artificial e Processamento de Linguagem Natural (PLN) com uma interface gráfica interativa para proporcionar uma experiência de usuário eficiente e personalizada. A evolução da IA, desde os sistemas baseados em regras até o uso de redes neurais profundas e modelos de transformadores como o GPT-3.5, permitiu a criação de sistemas altamente capazes de entender e gerar linguagem natural de forma precisa e relevante. Essa capacidade é fundamental para resolver problemas comuns enfrentados pelos usuários durante pesquisas em navegadores, oferecendo uma solução que é tanto rápida quanto precisa.

Além disso, a implementação de uma interface gráfica 2D meticulosamente projetada em Python demonstra a importância das interfaces visuais na facilitação da interação entre humanos e sistemas de IA. A escolha de ferramentas como o Tkinter para a criação de uma interface gráfica intuitiva reflete a necessidade de uma experiência de usuário acessível e agradável, que complementa as avançadas capacidades de linguagem do GPT-3.5. Portanto, a fundamentação teórica sustenta a premissa de que a integração harmoniosa entre modelos de IA e interfaces gráficas pode não apenas melhorar a usabilidade, mas também transformar a forma como os

usuários interagem com a tecnologia, proporcionando interações mais naturais, fluidas e eficazes.

4. PLANILHA DE CUSTOS DO PROJETO

Tabela 1 - Planilha de custos do projeto

QTD	DESCRIÇÃO DO RECURSO	VALOR UNITÁRO (R$)	VALOR TOTAL (R$)	FONTE
1	Notebook Acer Aspire 5 Intel Core i5, 8GB RAM, SSD 256GB	R$2.879,10	R$2.879,10	https://bit.ly/48KFDNA Acesso em: 28 de agosto 2024
1	Licença Windows	R$79,00	R$79,00	https://bit.ly/4bB0F3q Acesso em: 28 de agosto 2024
1	Python 3.12	Gratuito	Gratuito	https://bit.ly/3wxC1kD Acesso em: 28 de agosto 2024
1	Visual Studio Code	Gratuito	Gratuito	https://bit.ly/3T4PrfD Acesso em: 28 de agosto 2024
1	WampServer	Gratuito	Gratuito	https://bit.ly/47d2baL Acesso em: 03 de setembro 2024
1	MySQL WorkBench	Gratuito	Gratuito	https://bit.ly/3TdinD6 Acesso em: 03 de setembro 2024
1	Conta na plataforma da OpenAI	Gratuito	Gratuito	https://bit.ly/3Ms3Wr7 Acesso em: 28 de agosto 2024
1	Depósito na plataforma da OpenAI	R$30,00	R$30,00	https://bit.ly/3SJplzq Acesso em: 13 de agosto 2024
1	Krita	Gratuito	Gratuito	https://bit.ly/3AQ9OYD Acesso em: 03 de setembro 2024
1	Asset Unity 2D	Gratuito	Gratuito	https://bit.ly/3VdXb1q Acesso em: 28 de agosto 2024
1	Canva	Gratuito	Gratuito	https://bit.ly/4go6eFx Acesso em: 03 de setembro 2024
TOTAL			R$	2988,1

Fonte: Autoria própria

*Na tabela abaixo encontra-se o valor de remuneração de dois técnicos de informática, que foram necessários para o desenvolvimento do projeto. Fizemos um cálculo aproximado e chegamos em 468 horas trabalhadas, com uma média de 3 horas por dia.

QTD	DESCRIÇÃO DO RECURSO	VALOR UNITÁRO (R$)	VALOR TOTAL (R$)	FONTE
2	Técnicos de informática júnior	R$9,05/hora	R$18,10/hora	https://bit.ly/3yBAcUSAcesso em: 27 de maio 2024
TOTAL			R$	4235,4

5. RESULTADOS E ANÁLISES DE DADOS

A análise dos dados foi realizada levando em consideração aspectos como a interação do usuário com a IA, o tempo de resposta, a eficiência do processamento de linguagem natural, e a usabilidade da interface.

Durante a fase de testes, observou-se que a Joy conseguiu interpretar os comandos fornecidos pelos usuários de maneira eficiente. O tempo médio de resposta, que variou conforme a complexidade da solicitação, foi otimizado graças à implementação do banco de dados MySQL, que armazenou e gerenciou os prompts. A integração do GPT-3.5 com o banco de dados permitiu à Joy acessar rapidamente os dados relevantes, reduzindo o tempo de processamento e fornecendo respostas ágeis e precisas.

Além do processamento eficiente das informações, um ponto de destaque foi a interface gráfica desenvolvida em Python. Ela proporcionou um ambiente amigável e intuitivo para os usuários, facilitando a personalização das interações. A interface permitiu que os usuários selecionassem comandos previamente armazenados ou criassem seus próprios, e essas funcionalidades foram avaliadas positivamente durante os testes, destacando a flexibilidade do sistema. A alternância entre as animações de "idle" e "talking", controlada por meio de um arquivo de estado, contribuiu para uma experiência mais dinâmica e envolvente, aumentando o grau de imersão do usuário durante a interação.

Os resultados também mostraram que o uso de ferramentas como o Krita para o desenvolvimento da interface final foi crucial. A animação fluida da personagem Joy, com diferentes expressões e movimentos, trouxe um aspecto visual atraente ao sistema, promovendo uma interação mais humanizada. Essa característica é

fundamental em sistemas que buscam estabelecer uma comunicação natural com os usuários.

Durante os testes, observou-se que a integração da Joy com o hardware disponível foi realizada com eficiência, demonstrando a robustez do sistema. A Joy operou com excelente desempenho em dispositivos com especificações adequadas, como notebooks com processadores modernos e capacidade de memória otimizada. Além disso, a utilização da API da OpenAI proporcionou um amplo acesso aos recursos avançados do GPT-3.5, garantindo respostas rápidas e precisas durante as interações.

Os resultados foram positivos e atingiram as expectativas. A capacidade do sistema de fornecer respostas personalizadas e ágeis reforçou a eficácia da abordagem baseada no GPT-3.5. A análise dos dados indica que a Joy é uma ferramenta promissora para auxiliar em diversas tarefas que exigem processamento de linguagem natural, especialmente quando integrada a uma interface gráfica intuitiva.

6. DISCUSSÃO

A interação entre os usuários e a Joy demonstrou a eficácia da integração do modelo GPT-3.5 com uma interface gráfica 2D, validando o pressuposto de que uma abordagem bem estruturada pode melhorar significativamente a experiência do usuário. Essa discussão envolve a análise da eficiência do sistema em oferecer respostas personalizadas e a maneira como a interface gráfica contribuiu para essa interação.

O desempenho da Joy está alinhado com estudos que destacam a importância das interfaces intuitivas em sistemas de IA. Interfaces bem projetadas são essenciais para tornar sistemas complexos mais acessíveis e melhorar a experiência do usuário. A interface gráfica desenvolvida para a Joy reforçou essa ideia, fornecendo um ambiente que simplifica a interação do usuário com a inteligência artificial. Além disso, a possibilidade de personalizar comandos e armazená-los em um banco de dados ampliou as opções de uso da Joy, tornando-a mais adaptável a diferentes necessidades e preferências dos usuários.

Embora os resultados tenham sido positivos, algumas questões importantes merecem atenção. A adoção da Joy em ambientes com computadores de desempenho mais modesto pode ser desafiadora, indicando a importância de futuras otimizações no código para possibilitar a execução eficiente em dispositivos com recursos mais limitados. Além disso, a utilização da API da OpenAI requer uma análise cuidadosa dos custos envolvidos, para garantir a viabilidade econômica da aplicação a longo prazo.

Outro aspecto relevante é a humanização da interação. Embora a Joy tenha sido projetada para oferecer respostas precisas e contextualmente relevantes, a experiência de interação pode ser aprimorada ao incorporar recursos como reconhecimento facial ou de voz, que permitiriam à IA interpretar o estado emocional do usuário e ajustar suas respostas. Esta possibilidade está alinhada com estudos recentes que enfatizam a importância de sistemas de IA em compreender e se adaptar às emoções humanas para criar interações mais empáticas e efetivas.

Além disso, é importante considerar o potencial ético do uso de uma IA como a Joy. Em um cenário onde sistemas de IA têm influência crescente nas decisões e comportamentos humanos, é fundamental refletir sobre os princípios éticos envolvidos, como a privacidade dos dados dos usuários e a transparência na comunicação. A discussão desses aspectos é essencial para garantir que a aplicação da IA se alinhe com valores éticos e legais.

7. CONCLUSÕES

O desenvolvimento e implementação da inteligência artificial Joy, baseada no modelo GPT-3.5, mostrou-se eficaz em alcançar os objetivos propostos de criar uma interface amigável, interativa e personalizada para o usuário. A combinação de processamento de linguagem natural com uma interface gráfica 2D avançada permitiu uma experiência de interação mais intuitiva e eficiente, destacando a importância das interfaces visuais em sistemas de IA.

A conclusão deste projeto evidencia a importância da integração entre a IA e o design de interface. A Joy não apenas se mostrou capaz de responder de forma relevante aos comandos dos usuários, mas também apresentou uma interação visualmente envolvente, o que é fundamental para a aceitação e usabilidade do

sistema. O uso de ferramentas como o Tkinter e Krita foi crucial para proporcionar uma interface acessível e personalizada, contribuindo para tornar o sistema mais humanizado e envolvente.

No entanto, esta conclusão também traz reflexões sobre possíveis aprimoramentos futuros. A necessidade de um hardware mais potente destaca a importância de buscar otimizações que permitam a execução eficiente do sistema em uma gama mais ampla de dispositivos. Além disso, explorar funcionalidades como reconhecimento facial e de voz pode elevar a experiência de interação, tornando a Joy ainda mais sensível às nuances do comportamento humano.

A experiência adquirida ao longo deste projeto pode servir como base para futuros desenvolvimentos na área da inteligência artificial. Os resultados reforçam a importância de interfaces amigáveis e personalizadas em sistemas de IA e abrem caminhos para novos estudos sobre como aprimorar essas interações. A aplicação da Joy pode ser expandida para diversos contextos, como atendimento ao cliente, suporte educacional e assistentes pessoais, tornando-se uma ferramenta valiosa para enfrentar os desafios atuais de interação humano-computador.

Assim, conclui-se que a Joy é um avanço significativo no campo da inteligência artificial interativa. Ela representa um passo à frente na criação de assistentes virtuais mais eficazes, envolventes e adaptáveis às necessidades dos usuários. Este projeto demonstra que a combinação de uma IA avançada com uma interface visual intuitiva é capaz de transformar a experiência do usuário, tornando-a mais natural, eficiente e prazerosa.

REFERÊNCIAS BIBLIOGRÁFICAS

AMOS, D. **The Ultimate Guide to Speech Recognition With Python**. Disponível em: https://storage.googleapis.com/kaggle-forum-message-attachments/882826/15829/Ultimate%20Guide%20To%20Speech%20Recognition%20With%20Python.pdf?GoogleAccessId=web-data@kaggle-161607.iam.gserviceaccount.com. Acesso em: 08 mar. 2024.

APICE. **Aprendizagem interativa em ciências e engenharia**. FEBRACE. 2024. Disponível em http://apice.febrace.org.br/. Acesso em abril/2024.

FONTOURA, R. V. *et al*. **Interfaces entre a Ciência da Informação e Inteligência Artificial:o uso de um Chat Inteligente**. **Ci. Inf. Rev**, Maceió, v. 9, ed. 1/3, p. 1-15, 1 abr. 2023. Disponível em: https://www.seer.ufal.br/index.php/cir/article/view/11115/10502. Acesso em: 29 fev. 2024.

JÚNIOR, C. F. C. **Chatbot: uma visão geral sobre aplicações inteligentes**. 2018. 17 p. Trabalho de conclusão de curso (Mestrado em Sistemas e Computação) - Instituto Federal de Tocantins, [S. l.], 2018. Disponível em: https://sitionovo.ifto.edu.br/index.php/sitionovo/article/view/140/86. Acesso em: 8 mar. 2024.

PEREIRA, K. A. B. **Um estudo sobre o uso da Inteligência Artificial nas empresas**. Orientador: Prof. Dr. Carlos Alberto Oliveira de Freitas. 2021. 26 p. Trabalho de conclusão de curso (Bacharel em Sistemas de Informação) - Instituto de Ciências Exatas e Tecnologia da Universidade Federal do Amazonas, Amazonas, 2021. Disponível em: https://riu.ufam.edu.br/bitstream/prefix/5989/2/TCC_KeithAnnyPereira.pdf. Acesso em: 8 mar. 2024.

POSTAL, L. C. **ROBÔ NO ATENDIMENTO AO CLIENTE: QUANTO MAIS "HUMANO" MELHOR?**. Orientador: Profa. Dra. Cristiane Pizzutti dos Santos. 2019. 94 p. Trabalho de conclusão de curso (Mestrado em Administração) - Universidade Federal do Rio Grande do Sul, [*S. l.*], 2019. Disponível em: https://lume.ufrgs.br/handle/10183/206448. Acesso em: 11 mar. 2024.

SCHUNK, L. M. **O uso de Inteligência Artificial por meio de chatbots no processo de atendimento ao cliente: um estudo sobre seus benefícios**. 2020. 135 p. Trabalho de conclusão de curso (Mestre em Administração de Empresas) - Escola de Administração de Empresas de São Paulo, da Fundação Getulio Vargas, São Paulo, 2020. Disponível em: https://repositorio.fgv.br/server/api/core/bitstreams/12b8022f-e8eb-4628-8c4d-e641e7c716b3/content. Acesso em: 8 mar. 2024.