Projeto Keila – Assistente Pessoal com IA

Keila é um dispositivo físico, baseado em um Raspberry Pi 3 B+, que atua como uma assistente virtual inteligente. O sistema recebe comandos de voz, converte para texto (STT), envia à API da OpenAI (ChatGPT), recebe a resposta em texto e converte novamente em áudio (TTS) para resposta falada. Todas as interações são exibidas em uma tela sensível ao toque, e o sistema pode ser configurado remotamente via navegador ou localmente via tela embarcada.

A Keila poderá ter diferentes ACTIONS, que são capacidades/funcionalidades que podem ser adicionadas no futuro. No momento, a única ACTION é a VoicePrompt, que engloba todo o processo de gravar áudio, enviar gpt e tocar a resposta. No futuro, poderemos ter diferentes actions como Tocar Musica, entre outras.

Existe um estado principal que define o fluxo da aplicação: INIT, CANCELING, READY ou RUNNING. Quando uma ACTION esta acontecendo, o estado global é RUNNING. Existe um endpoint que pode cancelar todas as acoes sendo feitas. Isto é implementado com um Thread Mananager, que tem métodos como cancelar todas as acoes e iniciar thread. Cada action roda em sua própria thread e pode iniciar novas threads. Mas sempre, o fluxo principal deve estar pronto a interromper mediante um comando do usuário.

Existe sempre um fluxo rodando na thread principal, com o objetivo de deixar a Keila “bem responsiva”, no sentido em que ela pode estar buscando na internet, tocando áudio, gravando áudio, qualquer comando que eu dê para interromper ela, ela deve parar o que está fazendo e reiniciar o processo de escuta de áudio. Isto envolve programação que fique ouvindo “non-stop”, algo que reconheça “silêncio / perceba que o usuário parou/terminou de falar” e algo que “reconheca alguns comandos básicos como “oi keila” ou outros.

A tela capacitiva deve mostrar um avatar da Keila, com expressões faciais simples. Esta tela é usada para exibir o avatar e informações quando necessário. A tela também é usada para configurar a Keila, de forma alternativa ao acesso as configurações via navegador. Ambas são formas válidas de interagir com a API da Keila

Deve existir um botão físico ou acessado via tela, que inicia uma ACTION pre-definida. Ex: A VoicePrompt

# Funcionalidades

* • Reconhecimento de voz
* • Integração com OpenAI (GPT-4/4o)
* • Resposta por voz (TTS)
* • Interface touchscreen embarcada
* • Painel de administração via navegador
* • Web API local para controle e configuração
* • LEDs indicadores
* • Sensor de temperatura ambiente
* • Inicialização automática ao ligar

# Hardware Utilizado

* • Raspberry Pi 3 B+: Computador principal
* • Tela Capacitiva HDMI + USB (5~7"): Interface visual com toque
* • Fonte 5V 3A (ou 4~5A): Alimentação do sistema
* • Microfone USB ou I2S: Captação de áudio
* • Caixa de som P2 ou Bluetooth: Saída de áudio
* • Sensor de Temperatura (DHT22 ou BME280): Dados de ambiente
* • LED RGB (WS2812 ou comum): Feedback visual
* • Hub USB Alimentado: Para conectar microfone, touch etc.
* • Cartão microSD 16~32GB: Sistema operacional e dados

# Software

* • Sistema operacional: Raspberry Pi OS Lite (headless) ou com interface
* • Linguagem principal: Python 3
* • Framework web/API: FastAPI
* • Bibliotecas principais: SpeechRecognition, pyaudio, gtts, requests, pygame, gpiozero, threading ou asyncio

# Arquitetura da API

* • GET /config - Ler configurações
* • POST /config - Atualizar configurações
* • GET /status - Obter status atual
* • POST /listen - Iniciar/encerrar escuta
* • GET /history - Obter histórico

A API é usada tanto pela interface web quanto pela interface embarcada.

# Estrutura de Pastas do Projeto

keila/  
├── main.py  
├── api/  
│ ├── \_\_init\_\_.py  
│ ├── routes.py  
│ ├── config\_handler.py  
│ └── gpt\_engine.py  
├── ui/  
│ ├── tela\_embarcada.py  
│ ├── avatar.py  
│ └── status\_bar.py  
├── web/  
│ ├── static/  
│ ├── templates/  
│ └── web\_ui.py  
├── voice/  
│ ├── listener.py  
│ ├── speaker.py  
│ └── processor.py  
├── hardware/  
│ ├── leds.py  
│ ├── temperature.py  
│ └── gpio\_init.py  
├── services/  
│ └── boot\_service.sh  
├── settings.json  
├── history.log  
└── requirements.txt

# Fluxo de Funcionamento

1. Ao iniciar, main.py inicializa GPIOs, API, interface e escuta.  
2. O botão ou interface inicia a captura de áudio.  
3. O áudio é enviado à API de STT.  
4. O texto é enviado ao ChatGPT.  
5. A resposta é convertida em áudio (TTS).  
6. O áudio é reproduzido.  
7. A tela mostra a conversa.

# Interface Web Admin

Acessível via navegador na rede local, permite visualizar e alterar configurações da Keila, histórico, ativar recursos e mais.

# Interface Local Embarcada

Simples e clara, com botão de escuta, avatar animado, resposta em texto e status básicos.

# Segurança

A API pode ser protegida com senha/token, acesso remoto restrito à rede local, logs com permissões adequadas.

# Inicialização Automática

Utiliza um serviço systemd para iniciar o script automaticamente no boot.

# Simulação

O projeto pode ser simulado em um PC com Python, Flask, arquivos de áudio e mocks para sensores.