

Audição_cognitiva_google

March 1, 2024

```
[ ]: !pip install gtts
!pip install SpeechRecognition
!pip install pygame
!pip install pyaudio
```

```
[ ]: import sys
import tempfile
from gtts import gTTS
import speech_recognition as sr
import pygame
import os
import time

# Criar um diretório temporário para armazenar os arquivos temporários
temp_dir = tempfile.mkdtemp(prefix='tts_temp_', dir=tempfile.gettempdir())

recognizer = sr.Recognizer()

# Função de nome do cliente
def welcome_message():
    welcome_text = "Seja bem-vindo à Quantum Finance! Diga seu nome de cliente."
    print(welcome_text)
    tts = gTTS(text=welcome_text, lang='pt-br', slow=False)
    tts.save(f"{temp_dir}/welcome_message.mp3")
    play_audio(f"{temp_dir}/welcome_message.mp3")

def repeat_welcome_message():
    repeat_text = "Não foi possível entender."
    print(repeat_text)
    tts = gTTS(text=repeat_text, lang='pt-br', slow=False)
    tts.save(f"{temp_dir}/repeat_welcome_message.mp3")
    play_audio(f"{temp_dir}/repeat_welcome_message.mp3")

def play_audio(file_path):
    pygame.mixer.init()
```

```

pygame.mixer.music.load(file_path)
pygame.mixer.music.play()
while pygame.mixer.music.get_busy():
    pygame.time.Clock().tick(5)
pygame.mixer.quit()

def get_client_name():
    while True:
        welcome_message()

        with sr.Microphone() as source:
            print("Aguardando o nome do cliente...")
            try:
                audio = recognizer.listen(source, 10, 5)
                client_name = recognizer.recognize_google(
                    audio, language='pt-BR')
                print(f"Cliente disse: {client_name}")

                welcome_back_text = f"Ficamos felizes em lhe atender, \u
↪{client_name}."
                print(welcome_back_text)
                tts = gTTS(text=welcome_back_text, lang='pt-br', slow=False)
                tts.save(f"{temp_dir}/welcome_back_message.mp3")
                play_audio(f"{temp_dir}/welcome_back_message.mp3")

                return client_name

            except sr.UnknownValueError:
                repeat_welcome_message()
            except Exception as e:
                print(f"Vou pedir novamente {e}")

client_name = get_client_name()
print(f"Nome do cliente: {client_name}")

# Função das opções
def handle_user_choice():
    while True:
        intro_text = "Aqui estão as opções possíveis."
        print(intro_text)
        tts = gTTS(text=intro_text, lang='pt-br', slow=False)
        tts.save(f"{temp_dir}/intro_options.mp3")
        play_audio(f"{temp_dir}/intro_options.mp3")

```

```

for key in options:
    option_text = options[key]
    tts = gTTS(text=option_text, lang='pt-br', slow=False)
    tts.save(f"{temp_dir}/option_{key}.mp3")
    play_audio(f"{temp_dir}/option_{key}.mp3")

with sr.Microphone() as source:
    print("Aguardando sua escolha...")
    try:
        audio = recognizer.listen(source, 10, 5)
        print("Processando sua resposta...")
        user_choice = recognizer.recognize_google(
            audio, language='pt-BR').lower()
        print(f"Você disse: {user_choice}")

        # Verificar se o usuário respondeu com um número de 1 a 4
        if user_choice.isdigit() and 1 <= int(user_choice) <= 4:
            if user_choice == "1":
                choice_text = "Você escolheu a opção 1. Vou seguir com
↪sua solicitação. Seu saldo é de um milhão quinhentos e setenta e nove mil
↪dólares."

                elif user_choice == "2":
                    choice_text = "Você escolheu a opção 2. Vou seguir com
↪sua solicitação. Devido ao fuso horário, não podemos executar essa simulação
↪no momento."

                    elif user_choice == "3":
                        choice_text = "Você escolheu a opção 3. Vou seguir com
↪sua solicitação. Infelizmente, todos estão ocupados assistindo BBB. Por
↪favor, escolha essa opção mais tarde."

                        elif user_choice == "4":
                            choice_text = "Muito obrigado e conte sempre conosco."
                            print(choice_text)
                            tts = gTTS(text=choice_text, lang='pt-br', slow=False)
                            tts.save(f"{temp_dir}/choice_response.mp3")
                            play_audio(f"{temp_dir}/choice_response.mp3")
                            break # Encerrar o loop se a escolha for 4

                        print(f"Bot: {choice_text}") # Imprimir a resposta do bot
                        tts = gTTS(text=choice_text, lang='pt-br', slow=False)
                        tts.save(f"{temp_dir}/choice_response.mp3")
                        play_audio(f"{temp_dir}/choice_response.mp3")

                    else:
                        retry_text = "Desculpe, responda somente com a alternativa
↪desejada."

                        print(retry_text)
                        tts = gTTS(text=retry_text, lang='pt-br', slow=False)

```

```

        tts.save(f"{temp_dir}/retry_question.mp3")
        play_audio(f"{temp_dir}/retry_question.mp3")
        time.sleep(1)

    except sr.WaitTimeoutError:
        print("Nenhum comando recebido. Reiniciando...")
    except Exception as e:
        print(f"Vou pedir novamente {e}")

# Definir opções
options = {
    "1": "Um, Consultar saldo",
    "2": "Dois, Simulação de empréstimo",
    "3": "Três, Falar com um atendente",
    "4": "Quatro, Encerrar atendimento"
}

# Chamar a função para lidar com a escolha do usuário
handle_user_choice()

```