

USO DE RECONHECIMENTO DE FALA NA AUTOMAÇÃO DE EQUIPAMENTOS DE GINÁSTICA

**Trabalho de Conclusão de Curso de
Ciência da Computação**

Katia Goldchleger

Orientador: Rogério de Oliveira

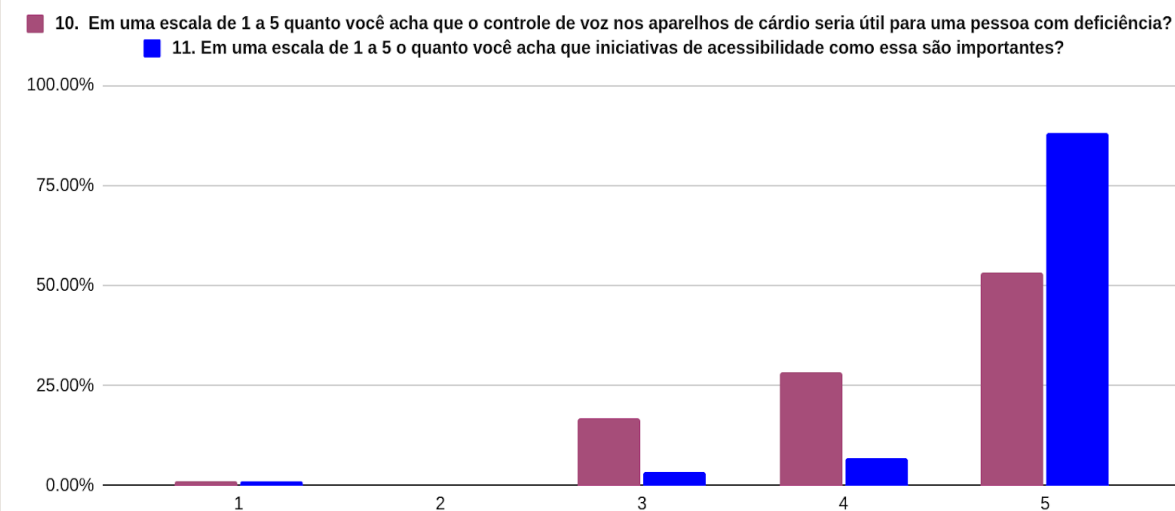


Resumo

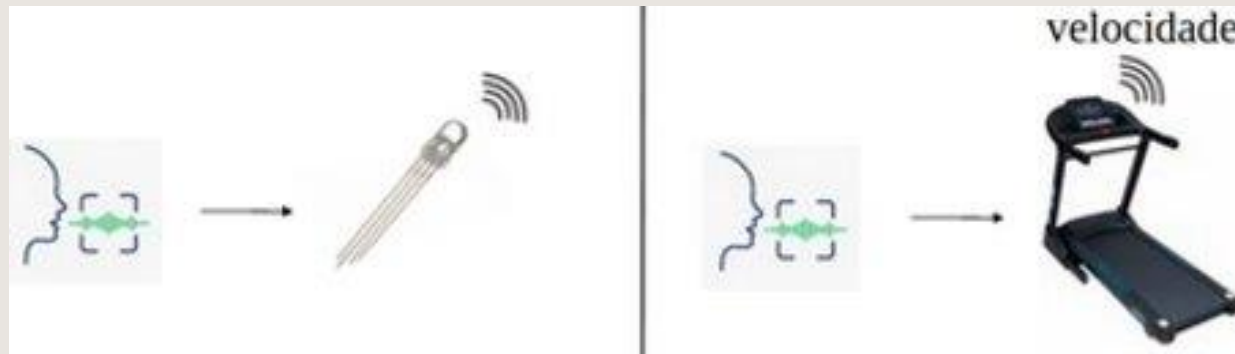
Tecnologias de automação de voz representam uma solução em potencial para melhoria da vida das pessoas possibilitando uma grande quantidade de aplicações na indústria, no comércio, em segurança, automação doméstica e até em cidades inteligentes. Este estudo consiste na implementação de uma solução de controle de voz, em dispositivos de processamento de baixo custo, viabilizando sua aplicação em equipamentos de ginástica para sua maior acessibilidade e uso desses equipamentos.

Introdução

Pessoas no mundo todo vem se beneficiando de tecnologias de voz. Seu uso na telefonia digital e controle de objetos são alguns exemplos de aplicações que trazem mais conforto ou mesmo transformam a vida das pessoas. O problema de acesso de pessoas com deficiência a alguns equipamentos de ginástica, através do uso de tecnologia de voz para o seu controle, pode minimizar esse problema e possibilitar uma melhor qualidade de vida dessas pessoas. A prática de exercícios e o uso de equipamentos de ginástica traz um grande benefício à vida e à saúde das pessoas com deficiência, contribuindo como parte do tratamento ou na sua melhoria de saúde em geral.



Contribuições



Este trabalho traz assim as seguintes contribuições:

1. Relaciona os principais componentes empregados para uma solução de comandos por voz para equipamentos de ginástica/exercícios;
2. Fornece um protótipo conceitual e minimamente funcional da solução e que pode ser empregado em novos desenvolvimentos;
3. Aponta limitações e melhorias futuras da solução proposta;

Referencial Teórico

O primeiro sistema de reconhecimento de voz, denominado "Audrey", foi desenvolvido no início dos anos 50 pelos Laboratórios Bell para a compreensão de dígitos numéricos.

Os sistemas de reconhecimento de fala evoluíram do uso predominante de modelos de HMM antes dos anos 2000, para modelos de redes profundas, usados nos principais sistemas atuais . As principais razões para esse sucesso se devem à capacidade analítica desse modelo no fenômeno da fala e sua precisão em sistemas práticos de reconhecimento de fala (Negin, N., Farbod, R, Hossein, S. (2009)).

As redes neurais profundas são sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano, usando algoritmos, elas podem reconhecer padrões escondidos e correlações em dados brutos, agrupá-los e classificá-los, e – com o tempo – aprender e melhorar continuamente.

"Accessibility Came by Accident": Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities

Pradhan, A., Mehta, K., & Findlater, L. (2018, April). "Accessibility Came by Accident" Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In Proceedings of the 2018 CHI Conference on human factors in computing systems (pp. 1-13).

De fato, assistentes de voz integrados a casas inteligentes já são uma realidade no uso doméstico para pessoas com deficiências (Pradhan, A., Mehta, K., & Findlater, L., 2018).

Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective

Gassert, R., & Dietz, V. (2018). Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective. Journal of neuroengineering and rehabilitation, 15(1), 1-15.

Este estudo apresenta uma solução conceitual, e minimamente funcional, do uso do reconhecimento de fala para o controle de equipamentos de ginástica, como esteiras e escadas rolantes, ou mesmo equipamentos empregados na reabilitação de pacientes, como robôs de exercícios (Gassert & Dietz, 2018), em que a pessoa pode controlar fatores como velocidade, força ou extensão, através de simples comandos de voz, aumentando assim a sua acessibilidade, conforto de uso e autonomia.

Introduction to Single Board Computers and Raspberry Pi

Pajankar, A. (2017). Introduction to Single Board Computers and Raspberry Pi. Raspberry Pi Image Processing Programming, 1–24.

Existem vários dispositivos de processamento de placa únicos como o Raspberry Pi, Orange Pi. Eles se diferem principalmente pela capacidade de memória, interfaces e suporte de sistemas operacionais. (Pajankar, *et al.* (2017)).

Industrial Applications of Automatic Speech Recognition Systems

Vajpai, J.; Bora, A. Industrial applications of automatic speech recognition systems. Ijera, v. 6, n. 3, p 88-95, 2016.

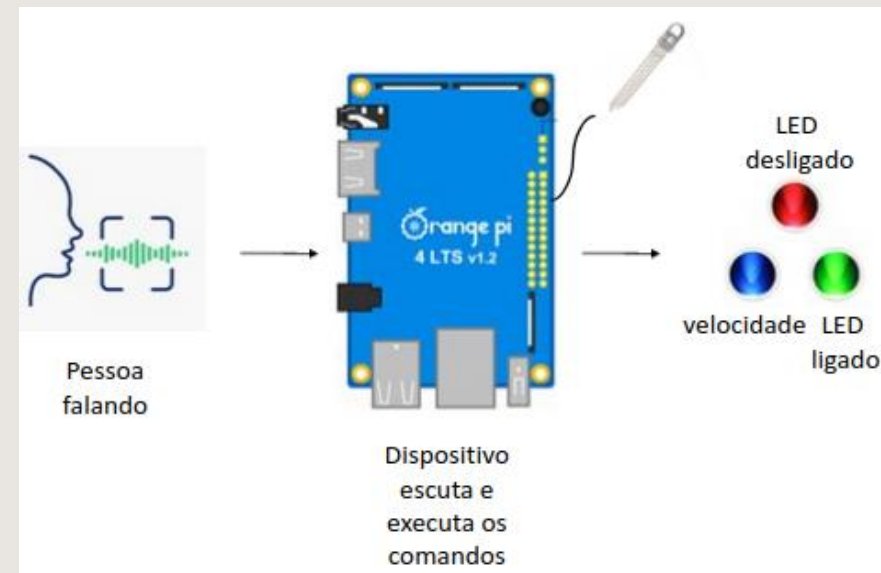
SISTEMA IN-DOOR DE MONITORAMENTO E CONTROLE PARA ACESSIBILIDADE DE PESSOAS COM DEFICIÊNCIA PELO RECONHECIMENTO DE FALA UTILIZANDO RASPBERRY PI

TCC Paulo Santos. Riu UFAM edu, 2022.

De acordo com o IBGE (Pessoas com deficiência. IBGE. [S. l.], 2010), há somente no Brasil, cerca de 46 milhões de brasileiros com alguma dificuldade ou deficiência, cerca de 24% da população.

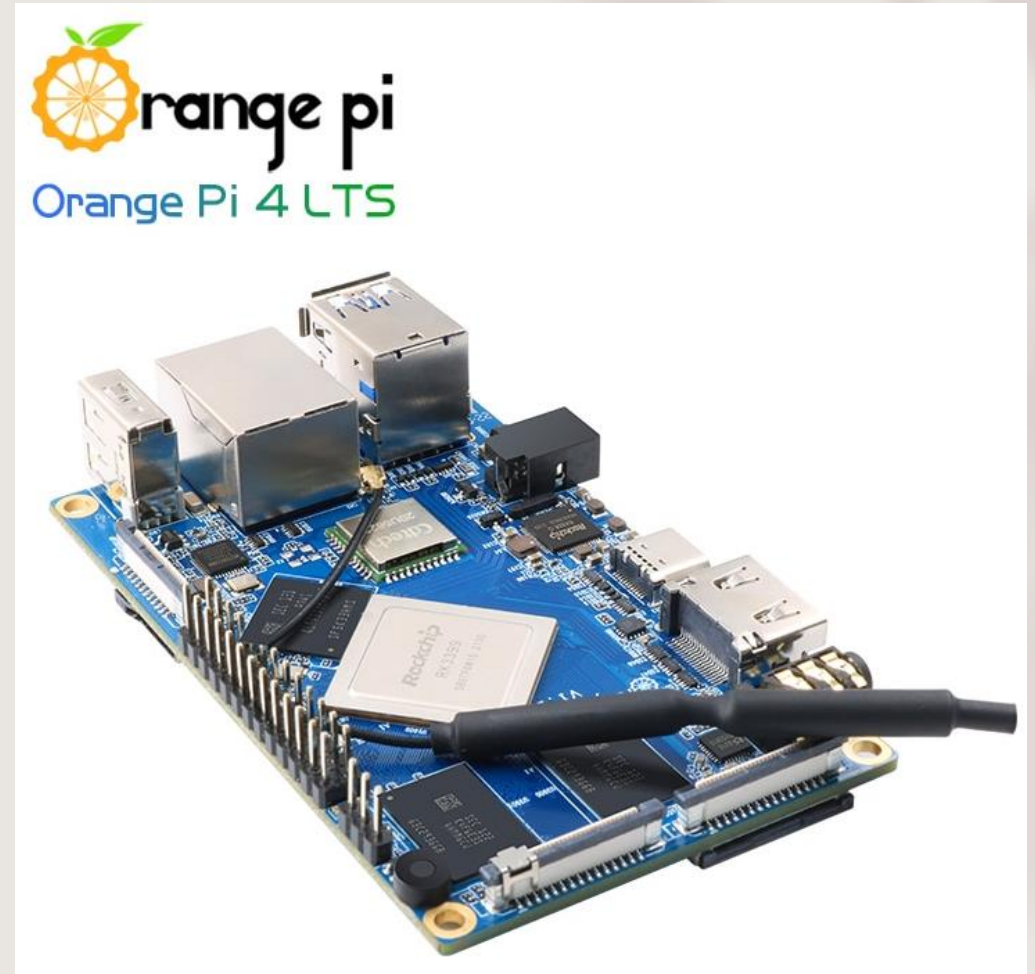
Metodologia

- Single Board Computer
- Bibliotecas
- Implementação
- Código



Single Board Computer

Para a implementação optamos pelo uso do computador de placa única de código aberto, uma nova geração de placas de desenvolvimento arm64, que pode executar sistemas operacionais como Android 8.1, Ubuntu e Debian. A placa de desenvolvimento Orange Pi (Orange Pi 4 LTS) usa Rockchip RK3399 ou RK3399-T SoC e possui memória LPDDR4 de 3 GB ou 4 GB e também possui microfone integrado.



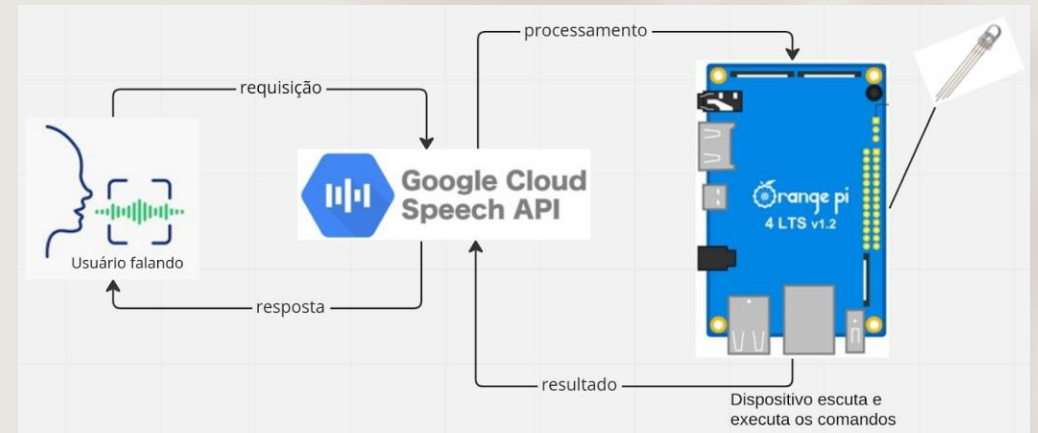
Bibliotecas

WiringPi é uma biblioteca de acesso GPIO baseada em PIN escrita em C para os dispositivos SoC BCM2835, BCM2836 e BCM2837 usados em todos os Raspberry Pi versões. Essa biblioteca é responsável pelo funcionamento do LED RGB;

SpeechRecognition faz a realização de reconhecimento de fala, com suporte para diversos motores e APIs, online e offline. Ela utiliza a API da Google para fazer o reconhecimento de fala no projeto;

Time é um módulo que provê várias funções relacionadas à tempo.

A biblioteca empregada foi SpeechRecognition que utiliza a API da Google para fazer o reconhecimento de fala.



Código

Bibliotecas empregadas no código

```
import speech_recognition as sr # biblioteca que faz o reconhecimento de fala
import wiringpi # biblioteca que possibilita o uso dos pinos GPIO
from wiringpi import GPIO
from time import sleep # biblioteca de variáveis de tempo
```

Classe para a configuração do LED

```
class Tr():
    # variáveis para frequência, frequência máxima, pinos de GPIO para o LED
    frequency = 1
    delta_freq = 1
    max_frequency = 5
    faster_gpio = 5
    start_gpio = 10
    stop_gpio = 9

    def __init__(self):
        wiringpi.pinMode(self.start_gpio, GPIO.OUTPUT)
        wiringpi.pinMode(self.stop_gpio, GPIO.OUTPUT)
        wiringpi.softToneCreate(self.faster_gpio)

    # função para acelerar a simulação do equipamento de ginástica
    def faster(self):
        if self.frequency == 0:
            self.start()

        else:
            self.frequency += self.delta_freq
            self.frequency = min(self.frequency, self.max_frequency)
            wiringpi.softToneWrite(self.faster_gpio, self.frequency)
```

```
# função para desacelerar a simulação do equipamento de ginástica
def slower(self):
    if self.frequency > 0:
        self.frequency -= self.delta_freq
        self.frequency = max(self.frequency, 0)
        wiringpi.softToneWrite(self.faster_gpio, self.frequency)

# função para iniciar a simulação do equipamento de ginástica
def start(self):
    wiringpi.digitalWrite(self.start_gpio, GPIO.HIGH)
    sleep(1)
    wiringpi.digitalWrite(self.start_gpio, GPIO.LOW)
    if self.frequency == 0:
        self.frequency = self.delta_freq
        wiringpi.softToneWrite(self.faster_gpio, self.frequency)
```

```
# função para parar a simulação do equipamento de ginástica
def stop(self):
    while (self.frequency > self.delta_freq):
        self.frequency -= self.delta_freq
        self.frequency = max(self.frequency, 0)
        wiringpi.softToneWrite(self.stop_gpio, self.frequency)
        sleep(1)
    wiringpi.softToneStop(self.faster_gpio)
    wiringpi.digitalWrite(self.stop_gpio, GPIO.HIGH)
    sleep(1)
    wiringpi.digitalWrite(self.stop_gpio, GPIO.LOW)

def exit(self):
    self.stop()
```

Recebe o sinal de voz

```
'''
O objetivo principal de uma instância do Recognizer é,
obviamente, reconhecer a fala. Cada instância vem com
uma variedade de configurações e funcionalidades para
reconhecer a fala de uma fonte de áudio.
'''

r = sr.Recognizer()

# Usa o microfone padrão como fonte de áudio
speech = sr.Microphone(2)
```

Chamada da API

```
with speech as source:
    '''
    O método adjust_for_ambient_noise() lê o primeiro
    segundo do fluxo de arquivo e calibra o reconhecedor
    para o nível de ruído do áudio.
    '''
    r.adjust_for_ambient_noise(source)

    '''
    Ouve a primeira frase e extraia-a em dados de áudio,
    phrase_time_limit define quanto tempo de fala par os
    comandos
    '''
    audio = r.listen(source, phrase_time_limit = 10)

    try:
        print("Diga o comando para a 'esteira': \n")

        # reconhecer a fala usando o Google Speech Recognition
        ans = r.recognize_google(audio, language = 'en-US')

        # a fala é ininteligível
    except sr.UnknownValueError:
        print("O Google Speech Recognition não conseguiu entender o áudio")
```

Comandos

```
if audio:
    if ans == 'start':
        print("Você disse: ", ans)
        # comando GPIO
        tr.start()

    elif ans == 'faster':
        print("Você disse: ", ans)
        # comando GPIO
        tr.faster()

    elif ans == 'slower':
        print("Você disse: ", ans)
        # comando GPIO
        tr.slower()

    elif ans == 'stop':
        print("Você disse: ", ans)
        # comando GPIO
        tr.stop()
        break
```

Resultados



Conclusão

Essa solução tem limites, pois há deficiências que esse estudo não atende. Os ruídos em uma academia está presente e isso pode impedir a compreensão dos comandos falados, o que pode atrapalhar o funcionamento do dispositivo. Este estudo fornece um modelo conceitual, mas funcional, que permite facilmente evoluir para uma solução em dispositivos reais, sendo viável a baixo custo, entre R\$200 e R\$350. Em trabalhos futuros parece oportuno o desenvolvimento dessa solução em equipamentos virtuais antes de sua implementação física e pensamos em avaliar o uso da mesma solução para aplicações em Metaverso, tema que, até onde pudemos verificar, ainda é pouco explorado.

GitHub

Projeto do TCC II no GitHub

https://github.com/TCCII/device_for_gym_equipment



Obrigado!