

SISTEMA DE RECONHECIMENTO DE VOZ APLICADO À AUTOMAÇÃO RESIDENCIAL PARA AUXÍLIO DE DEFICIENTES VISUAIS

Ivando Diniz¹, Luana Ferrão Álvares de Sousa², Rodrigo de Souza Jacomini³

¹Instituto de Ciência e Tecnologia de Sorocaba - ICST- GASI, PGEE – Pós graduação em Engenharia Elétrica Sorocaba - SP, ivando.diniz@unesp.br - Brasil

²Instituto de Ciêncics e Tecnologia de Sorocaba – ICST – Sorocaba – SP - Brasil

³Instituto de Ciêncics e Tecnologia de Sorocaba ICST- GASI, PGEE – Pós-graduação em Engenharia Elétrica - Sorocaba – SP- Brasil

Resumo: Com base no potencial da tecnologia para promover acessibilidade para pessoas com deficiência, o objetivo deste projeto foi desenvolver um sistema de reconhecimento de fala que permite monitorar e controlar pontos estratégicos da residência de um portador de deficiência visual por meio do celular. Esses pontos incluíam iluminação, portas, janelas, alarmes, entre outros, visando facilitar as atividades diárias do indivíduo. Durante o desenvolvimento, houve uma preocupação em garantir que o sistema fosse acessível, de baixo custo e simples o suficiente para realizar o reconhecimento das atividades desejadas com poucos recursos de processamento, além de ser customizável para não exigir grandes modificações no programa. Para isso, foram utilizadas APIs disponíveis para criação, modificação e geração de banco de dados SQLite, geração de áudio a partir de texto e reconhecimento de fala fornecidas pela Google. O trabalho abrange todas as etapas do desenvolvimento, desde a seleção dos dispositivos até o desenvolvimento dos códigos e a realização dos testes.

Palavras Chaves: Reconhecimento de voz, Mobile, Tecnologia Assistiva, Automação Residencial

INTRODUÇÃO

A automação residencial não deve ser vista somente sob o ponto de vista de luxo ou excentricidade, disponível àqueles com mais recursos financeiros. Como qualquer novidade, a automação residencial, inicialmente, é vista como um símbolo de status e não como algo de fato útil (Eiras,2023). No momento seguinte, o conforto e a conveniência proporcionados passam a ser decisivos. E, por fim, ela se tornará uma necessidade e um fator de economia. O mercado da automação residencial vem crescendo muito rapidamente, a cada dia surgem novos equipamentos, novas tecnologias, tudo para interagir com o nosso dia-a-dia (Carolina,2023).

A tecnologia é algo absolutamente presente na vida de qualquer ser humano atualmente pelo fato de que ela está cada vez mais transparente ao uso, deixando de ser algo assustador ou mesmo complexo de se operar. A automação em edifícios e empresas é bastante comum já há algum tempo, é possível notar os sistemas existentes nestes ambientes, como o sistema de detectar e combater incêndios, as centrais de alarmes, as câmeras de segurança, as portas giratórias, os

sensores de presença, entre outros (Pinto ,2019). O fator interessante é que estes sistemas estão migrando também para as residências, dando origem assim, aos termos Automação Residencial, Casa Automática, Domótica, Residência Inteligente e assim por diante. Os equipamentos devem centralizar os controles e processos tornando tudo mais simples e automático, mas é o desejo do usuário que deve prevalecer. A automação das residências deve, certamente, ajudar de forma eficiente e nas atividades diárias, que custe alto tempo de execução ou mesmo proporcionar uma sensação de conforto e segurança pelo fato de que algumas tarefas serão realizadas sem a necessidade de ordens diretas, como por exemplo, esquecer janelas abertas havendo uma previsão de chuva, o sistema se encarrega de fechar e trancar evitando possíveis furtos e danos à moradia (Alves, 2022).

Embora a automação seja um conceito bastante conhecido, ainda enfrenta alguns obstáculos, seja pelo custo que ainda é elevado ou por dificuldades infraestruturais de residência ou prédios. Mas essa visão já está sendo mudada, pois as pessoas estão buscando cada vez mais segurança e economia, benefícios esses que a automação atende (Feitosa, 2022).

A forte evolução da chamada “Internet das Coisas” abre uma enorme gama de possibilidades para este tipo de automação, uma vez que começam a surgir dispositivos padronizados para comunicação e controle, que utilizam protocolos de comunicação amplamente difundidos como WiFi e Bluetooth, facilitando a integração de módulos de automação com aplicativos para smartphones e tablet que podem ser desenvolvidos gratuitamente na plataforma Android e se comunicarem com dispositivos programáveis e plataformas de desenvolvimento, como o Arduino, reduzindo o custo de aquisição e instalação dos sistemas (Molina, 2015).

REVISÃO DE LEITURA

Internet das Coisas

A Internet das Coisas representa a integração e interação de objetos físicos, reais, através de uma conexão de Internet. No mundo virtual, o que ocorre são interações entre um indivíduo e outro e/ou informações eletrônicas por meio de hyperlinks. Uma das propostas da IoT é permitir, com o uso de tecnologias de rastreamento, identificação e troca de informações, que numerosos objetos comuniquem-se automaticamente e à distância. Este tipo de estudo poderá permitir diversas utilizações (Molina, 2015).

Android

O Android tem como características principais os seguintes itens, que demonstra a adaptação da plataforma aos dispositivos para os quais foi desenvolvido.

- Layouts para celulares
- Armazenamento (SQLite e MySQL)
- Conectividade (Wifi, Bluetooth, NFC)
- Suporte a múltiplas línguas
- Navegador Web

O Android suporta diversos recursos que podem estar presentes em um dispositivo móvel como câmeras, telas sensíveis ao toque, acelerômetros, giroscópios, barômetros, magnetômetros, controle de games dedicados, sensores de proximidade, movimento, pressão, termômetro, aceleradores 2D e 3D.

Arduino

Arduino é uma plataforma de prototipagem eletrônica, criada por Massimo Banzi e David Cuartielles em 2005 com objetivo de permitir o desenvolvimento de controle de sistemas interativos, de baixo custo e acessível a todos. Com o Arduino é possível também

enviar e receber informações de praticamente qualquer outro sistema eletrônico. Desta forma é possível construir por exemplo, um sistema de captação de dados de sensores, como temperatura, iluminação, processar e enviar esses dados para um sistema remoto por exemplo. Outra característica importante é que todo material (software, bibliotecas, hardware) é open-source, ou seja, pode ser reproduzido e usado por qualquer pessoa sem a necessidade de pagamento de royalties ou direitos autorais (ARDUINO, 2015).

Ethernet Shield

Existe uma grande diversidade de placas compatíveis com a plataforma Arduino, as quais agregam funções e periféricos adicionais, permitindo o desenvolvimento de soluções para os mais diversos setores e ambientes. Estas placas conectam-se diretamente ao Arduino e são chamadas de “Shields”. No projeto proposto foi utilizado o Ethernet Shield, que permite que o dispositivo se conecte à uma rede ethernet e até mesmo à internet, permitindo a troca de informações e possibilitando a execução de comandos como ler sinais de entradas e acionar saídas (ARDUINO, 2015).

Características:

- Requer uma placa Arduino
- Alimentação de 5V proveniente do Arduino
- Conector RJ-45
- Controlador ethernet W5100 com buffer interno de 16K
- Velocidade 10/100MB
- Conecta-se ao Arduino pela porta SPI

O Shield Ethernet possui ainda uma entrada para cartão microSD, que pode ser utilizado para armazenar arquivos e disponibilizá-los na rede.

PROPOSIÇÃO

Este trabalho apresenta um sistema de reconhecimento de fala aliado à automação residencial focado no auxílio a deficientes visuais. Dessa forma, as solicitações ao sistema como consulta do status de dispositivos e alteração do status de dispositivos deve ser feito através do reconhecimento de fala e a resposta do sistema às entradas dos usuários deve ser apresentada em formato de áudio. Dessa forma, o deficiente visual poderá acionar dispositivos ou consultar diversos status da residência como: “As janelas estão fechadas?”, “O portão está fechado?”, “A luz da cozinha está acesa?”, acionados através de comandos de voz apenas. O sistema também permite a adição de novos dispositivos físicos e de novas palavras no banco de dados.

Para a implementação do sistema é utilizado um conjunto Arduino Uno/Ethernet Shield, um roteador e um dispositivo Android, podendo ser um smartphone ou um tablet. Uma vez que, esta é uma das maneiras mais utilizadas de implementação de redes WiFi, esta configuração é ideal para a implementação em residências, pois permite que o conjunto Arduino e a WiFi do dispositivo Android se conectem à mesma rede que a rede provedora de Internet, necessário para o funcionamento do sistema.

Acesso à Internet pelo dispositivo Android é imprescindível, pois a API de reconhecimento de voz que utilizamos requer processamento em nuvem para identificar as palavras (GOOGLE, 2015). Por isso, optamos por utilizar o Ethernet Shield para conectar o Arduino à rede, garantindo assim a conectividade do dispositivo Android e do Arduino na mesma rede.

O aplicativo foi construído em inglês, uma vez que, durante seu desenvolvimento, a API de reconhecimento de voz em português não se encontrava com a precisão necessária para o desenvolvimento de sistemas como o descrito neste trabalho.

MATERIAIS E MÉTODOS

Para a construção do sistema proposto neste trabalho, foram utilizados os dispositivos apresentados abaixo:

- Microcontrolador Arduino/Genuíno Uno;
- Arduino Ethernet Shield;
- Smartphone Sony XperiaTM M2;
- Roteador onde possa-se conectar o conjunto Arduino/Shield à internet e gerar uma rede wifi onde o smartphone possa se conectar à internet e à rede local;
- leds para simular os dispositivos instalados na residência;
- resistores de 200k Ω ;

Dessa forma, são criadas as seguintes conexões (Monk, 2011):

- Arduino Uno / Ethernet Shield – Residência: através das saídas digitais presentes na placa.
- Arduino Uno / Ethernet Shield – Roteador: através do protocolo Ethernet TCP/IP e do cabo de rede conectado a ambos.
- Smartphone-Roteador WiFi: através do protocolo WiFi 802.11n.

Esta configuração foi escolhida pois, dessa forma, é possível que o usuário acesse a internet e controle sua residência através da mesma rede criada localmente através do roteador, como é usualmente realizado na maioria das residências, gerando, dessa forma, uma experiência de usuário mais intuitiva e confortável.

As próximas seções discutem características de cada um desses elementos para a construção do dispositivo.

Hardware

O Hardware, ou parte física do trabalho, consiste em um roteador wi-fi, o Arduino Uno em conjunto com uma Shield Ethernet e os componentes já existentes no dispositivo Android para a comunicação Wireless. Para simular o controle dos sensores e atuadores instalados na residência, foi montado um sistema elétrico ligado ao conjunto Arduino/Ethernet Shield utilizando LED's comuns para a sua sinalização.

O sistema de simulação dos dispositivos presentes na residência consiste em uma montagem elétrica simples na protoboard, alimentada pela placa do arduino e consiste em seis leds, ligados à resistores e ao conjunto Arduino Uno/Ethernet Shield apresentado na Figura 1.

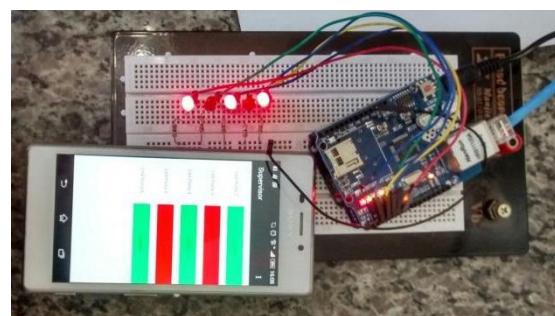


Figura 1. Protótipo desenvolvido. Fonte(Autoria Própria).

Dessa forma, é possível para o Arduino monitorar as portas e mudar seus status. A comunicação entre o conjunto Arduino/Ethernet Shield com os leds é direta, sendo que o valor definido para a porta I/O utilizada será o mesmo valor apresentado nos leds.

A conexão TCP/IP com o roteador é realizada pelo Ethernet Shield, conectando os dois dispositivos através de um simples cabo de rede, onde o roteador agritará o conjunto Arduino Uno/Ethernet Shield aos dispositivos pertencentes à rede local.

A conexão entre o dispositivo Android e o conjunto Arduino Uno/Ethernet Shield é realizada através de comunicação WiFi, intermediada pelo roteador. A conexão entre o dispositivo Android e o Roteador se dá através da rede Wi-Fi 802.11n. Todos os dispositivos utilizados para a atuação (Arduino Uno e dispositivo Android) devem estar configurados utilizando uma variação do IP do roteador. Por exemplo, para um roteador cujo IP é 192.168.1.1 o último parâmetro deve ser diferente do IP do roteador, mas comum ao IP de todos os componentes do sistema. Por exemplo, para o mesmo roteador citado

anteriormente, os dispositivos poderiam ser configurados com o IP 192.168.1.210.

Uma vez que dispositivos Android suportam apenas uma conexão Wireless por vez, é importante, para o conforto do usuário que a rede de controle doméstico seja a mesma rede por onde o dispositivo se conecta à internet, a fim de evitar desconforto e dificuldades para o usuário.

Software

A parte do desenvolvimento do projeto descrita nesta seção refere-se aos softwares utilizados para o desenvolvimento deste projeto. Foram utilizados softwares para escrever e compilar o código fonte tanto do Arduino Uno quanto do dispositivo Android. Ambos os dispositivos já possuem drivers para gravação automática através de seus programas de escrita e compilação.

Eclipse

Eclipse é uma IDE para desenvolvimento Java, podendo aceitar outras linguagens de programação através de plug-ins.

As aplicações para o Android podem ser desenvolvidas através da linguagem Java utilizando o Android Software Development Kit, SDK. No entanto existem outras ferramentas para se desenvolver aplicações, como por exemplo, o Native Development Kit que permite a criação de aplicativos com extensão C ou C++, além do Google App Inventor, um ambiente visual para programadores iniciantes e várias aplicações cruzadas em plataforma móvel e web frameworks.

Para desenvolvedores, o Android SDK proporciona um rico conjunto de ferramentas, incluindo depuradores, bibliotecas, simuladores para dispositivos, documentação detalhada, amostras de códigos e tutoriais. As aplicações para Android podem ser desenvolvidas utilizando o Eclipse com a ajuda de um plug-in chamado Android Development Tools, ADT. Este ajuda a alavancar os recursos presentes no Eclipse, como o conteúdo assistido, busca Java, recursos abertos, integração com o JUnit, e diferentes formas de aparência e perspectivas para a criação de aplicativos Android.

Plataforma Arduíno

Arduino é uma IDE para desenvolvimento do programa de qualquer placa da marca Arduíno, utilizando programação em Java. A programação na plataforma de desenvolvimento Arduino é bem

simples, os dados são transmitidos para o microcontrolador através de um cabo USB – SERIAL. Existem diversas outras plataformas para a programação de placas Arduino em outras linguagens, essas plataformas podem ser encontradas no site do Arduíno na url: <https://www.arduino.cc/en/Main/Software>.

Descrição do método de identificação gramatical

O programa FOXY, carregado no dispositivo Android tem como objetivo receber uma entrada do usuário em forma de voz, realizar o reconhecimento da mesma e gerar uma resposta à solicitação. Para isso, o resultado da entrada de voz do usuário é separado em palavras isoladas e, a partir da análise das palavras e das suas respectivas posições na frase, a identificação é realizada.

Uma vez que o aplicativo tem como foco somente a automação residencial, um número limitado de palavras serão necessárias para a realização das ações, dessa forma, foi criada uma base de dados padrão com as possíveis palavras necessárias para a realização das ações. A interpretação das palavras utilizadas e a sua posição na frase definirão se a entrada do usuário é uma pergunta ou uma solicitação, o dispositivo procurado e sua localização (“table_word”).

Os dispositivos alvo, ou seja, os sensores e atuadores presentes na residência, são localizados através da interpretação das palavras inseridas pelo usuário e seu endereço no hardware (Tabela “table_hard”). Assim, são usadas duas tabelas no banco de dados, uma com as palavras em geral e suas características necessárias para sua classificação e outra com o nome do dispositivo alvo e seu respectivo endereço no arduino.

As palavras de uso geral são definidas em 2 parâmetros: tipo e status. Dessa forma, cada frase gerará 3 vetores, cada um com o seu tipo.

O parâmetro TYPE pode receber qualquer uma das seguintes classificações:

- VERB: Indica que a palavra é um verbo.
- DUAL: Indica que a palavra pode ser um verbo ou um status. Todas as palavras do tipo dual tem o parâmetro STATUS como ON quando são utilizadas como status.
- SPECIAL: indica um tipo de pergunta ou ação referente à mais de um dispositivo diferente. (ex: no caso de “turn off all lights ” all é um special).
- LOCATION: Identificação do local físico da residência onde o dispositivo alvo se encontra.
- ID: Classificação do Location, caso existam dois cômodos com o mesmo nome. Por exemplo: kids room e main room

- TARGET: Nome do dispositivo que será monitorado/controlado.
 - STATUS: Estado do dispositivo (ligado, desligado, aberto, fechado, etc.)
 - NAV_COMMAND: Comandos de navegação para trocar de tela.
- O parâmetro STATUS pode receber qualquer uma das seguintes classificações:
- QUESTION: Define que o verbo utilizado indica uma pergunta.
 - ACTION: Define que o verbo utilizado indica uma solicitação.
 - ON: Define o estado procurado como ligado, aberto, etc.
 - OFF: Define o estado procurado como desligado, fechado, etc.
 - NONE: O parâmetro STATUS é irrelevante para a palavra procurada.
 - SINGLE: Indica que o dispositivo alvo é um único elemento.
 - PLURAL: Indica que o dispositivo alvo é um conjunto de elementos.

A tabela previamente gerada pelo aplicativo está apresentada na Tabela 1.

Tabela 1. : Tabela "WORDS" 'de palavras de uso geral.

KEY_WORD	KEY_TYPE	KEY_STATUS
IS	VERB	QUESTION
ARE	VERB	QUESTION
TURN	VERB	ACTION
CLOSE	VERB	ACTION
OPEN	DUAL	ON
SET	DUAL	ON
GARAGE	LOCATION	NONE
ROOM	LOCATION	NONE
SUITE	LOCATION	NONE
KIDS	ID	NONE
ONE	ID	NONE
TWO	ID	NONE
TREE	ID	NONE
LIGHT	TARGET	SINGLE
LIGHTS	TARGET	PLURAL
WINDOW	TARGET	SINGLE
WINDOWS	TARGET	PLURAL
ALARM	TARGET	SINGLE
ALARMS	TARGET	PLURAL
GATE	TARGET	SINGLE
GATES	TARGET	PLURAL
ON	STATUS	ON
OFF	STATUS	OFF
CLOSED	STATUS	OFF
ANY	SPECIAL	NULL
WITCH	SPECIAL	NULL
ALL	SPECIAL	NULL
WORD	ID	INSERT
DEVICE	ID	INSERT

SUPERVISOR	NAV_COMMAND	NOID
INSERT	NAV_COMMAND	ID
MAIN	NAV_COMMAND	NOID

Os dispositivos instalados na residência são identificados com a combinação do seu tipo, localização e id e inseridos em uma tabela que os relaciona com o endereço da porta em que estão ligados no dispositivo físico, como apresentado na Tabela 2.

Tabela 2 : Tabela "HARDWARE" de sentenças identificadoras de dispositivos e seus endereços.

KEY_LOCATION	KEY_HARD_TYPE
my room light	0
kids room window	1
garage gate	2
kids room light	3
alarm	4

A partir dessas informações, a identificação da frase inserida pelo usuário é encontrada da seguinte forma:

Irá verificar se a entrada é um COMANDO DE NAVEGAÇÃO. Para isto, caso a palavra inicial (antes da separação) seja igual a um dos comandos pré programados apresentados nas Tabela 1 e 2.

- "view database" Abre uma activity que mostra todos os dados inseridos no banco de dados
- "insert word" Abre uma activity que insere uma nova palavra de uso geral na tabela Words juntamente com suas classificações (type e status)
- "insert device" Abre uma activity que insere um novo dispositivo na tabela Hardware juntamente com suas classificações (KEY_LOCATION e porta relacionada)
- "supervisor" Abre uma activity que mostra em botões os status dos dispositivos e simular seu funcionamento.
- "finish activity" Encerra o aplicativo.

Caso não seja um Comando de Navegação, a frase inserida é dividida em palavras separadas e os vetores do TIPO e do STATUS são criados. A partir dessas informações são identificados:

- O dispositivo alvo (luz, janela, alarme, etc.);
- A localização do dispositivo (sala, quarto das crianças, garagem, etc.);

Caso as informações encontradas sejam suficientes para formar uma frase e existam no banco de dados, são procuradas as portas referentes aos dispositivos alvo. As condições para que a frase seja tratável são:

- i. É necessário que exista um TARGET
- ii. Caso não exista o um dispositivo nomeado simplesmente com um TARGET, é necessário que exista um LOCATION + ID + TARGET identificável.
- iii. Caso exista um DUAL, é necessário que exista um VERB ou STATUS.
- iv. É necessário que exista um VERB, ou, caso não exista status, é necessário que exista um DUAL.

Então o tipo do verbo é analisado e tratado diferentemente para os casos Pergunta e Solicitação. No caso de haver uma palavra com o tipo DUAL na frase, é analisada a presença ou não de verbos. Caso exista verbo, o DUAL é tratado como status, caso contrário é tratado como verbo.

Caso Pergunta:

- v. Caso a pergunta apresente um SPECIAL ela é dividida em três casos.
 - “all”: Retorna “yes” se todos os dispositivos que possuam o TARGET procurado dentro de seu nome (KEY_LOCATION da tabela TABLE_HARD.) apresentarem o STATUS inserido pelo usuário igual ao encontrado no HARDWARE. Retorna “No” caso contrário.
 - “any”: Retorna “yes” se algum dos os dispositivos que possuam o TARGET procurado dentro de seu nome (KEY_LOCATION da tabela TABLE_HARD.) apresentarem o STATUS inserido pelo usuário igual ao encontrado no HARDWARE. Retorna “No” caso contrário.
 - “witch”: Procura por todos os dispositivos que possuam o TARGET procurado dentro de seu nome (KEY_LOCATION da tabela TABLE_HARD) e que tenham o STATUS inserido pelo usuário igual ao encontrado no HARDWARE e retorna seus nomes. Caso nenhum cumpra os critérios desejados, é retornado que nenhum está no estado procurado.
- vi. Caso não exista o tipo SPECIAL na frase, é verificado se o status do dispositivo identificado inserido pelo usuário é igual ao status do dispositivo identificado lido no Hardware e é gerada a resposta para o usuário. Para resposta afirmativa ou negativa:

Caso Solicitação:

Caso a solicitação apresente um SPECIAL ela somente será possível para o caso all.

- “all”: Verifica, como explicado para o caso “Pergunta” se todos os dispositivos que possuam o TARGET procurado dentro de seu nome (KEY_LOCATION da tabela TABLE_HARD.) apresentam o STATUS inserido pelo usuário igual ao encontrado no HARDWARE. Caso sim, é retornada a resposta, caso contrário, o comando para inverter o estado de todos os dispositivos que apresentarem o STATUS inserido pelo usuário diferente do encontrado no HARDWARE.

Caso não exista o tipo SPECIAL na frase, é verificado se o status do dispositivo identificado inserido pelo usuário é igual ao status do dispositivo identificado lido no Hardware e é gerada a resposta ou ação e resposta para o usuário.

- Caso o STATUS do HARDWARE esteja igual ao STATUS inserido pelo USUÁRIO ele retorna a resposta de forma semelhante ao caso da pergunta.
- Caso o STATUS do HARDWARE esteja diferente do STATUS inserido pelo USUÁRIO ele inverte o STATUS do HARDWARE e informa ao usuário da ação tomada de forma semelhante ao caso pergunta.

Descrição do Código dos Programas Desenvolvidos

Para o desenvolvimento deste trabalho, foram criados dois programas diferentes, sendo um para o dispositivo Android, e outro para o Arduino.

O programa carregado no Arduíno Uno/Ethernet Shield é utilizado para monitorar e controlar os leds ligados ao mesmo de acordo com os comandos recebidos do dispositivo Android e enviar uma resposta ao Smartphone quando solicitado.

O programa desenvolvido para o dispositivo Android, visa realizar o reconhecimento de fala e estruturas de banco de dados. Para tanto, foram utilizadas API's disponibilizadas pela Google para o desenvolvimento de aplicativos Android. Este aplicativo é responsável por:

- Comunicar com o Arduíno Uno/Ethernet Shield solicitando os status das portas monitoradas.

- Comunicar com o Arduino Uno/Ethernet Shield solicitando a mudança do status das portas monitoradas.
- Geração, acesso e monitoração de um banco de dados baseado em SQLite.
- Realização do reconhecimento de fala.
- Interpretação da fala reconhecida através de análise gramatical.
- Geração da resposta ao pedido do usuário e realização da ação desejada nas portas verificadas.

Foxy – programa carregado no Android

O programa FOXY utilizado no dispositivo Android utiliza a API de reconhecimento de voz disponibilizada pela Google, que utiliza processamento em nuvem. Portanto, para que o reconhecimento seja realizado, é necessário que o dispositivo andróide e o Arduíno estejam conectados à mesma rede e que o dispositivo Android esteja conectado à Internet.

O código para o smartphone foi desenvolvido em ambiente Eclipse e dividido em 7 diferentes classes:

- LANGUAGE HANDLER: Auxilia as outras classes a lidar com os processos ligados ao banco de dados e não possui layout vinculado. As responsabilidades da classe são:

1. Realizar a criação do banco de dados na linguagem SQLite (Mednieks, 2014).
2. Criar as tabelas necessárias para a interpretação gramatical das frases inseridas pelo usuário e os endereços de cada dispositivo inserido para a realização das ações solicitadas pelo usuário.
3. Adicionar as entradas pré-programadas no banco de dados.
4. Executar os métodos necessários para os diferentes tipos de pesquisa no Banco de Dados. É utilizado como um objeto.

- CUSTOM ADAPTER: É a classe objeto criada para customizar as formas como as informações inseridas pelo usuário e geradas pelo programa serão apresentadas no elemento ListView criado na main activity (Google).

- SUPERVISOR: Monitora e simula as ações realizadas nos leds. É possível mudar os status dos Leds a partir do supervisor e também ver o seu status caso o mesmo tenha sido mudado por outra classe.

- VIEW DATABASE: Apresenta as tabelas inseridas no banco de dados
- INSERT DATA WORD: Insere novas palavras de uso geral na tabela WORDS do banco de dados do aplicativo.

- INSERT DEVICE: Insere novos dispositivos na tabela HARDWARE do banco de dados do aplicativo.

- MAIN ACTIVITY: É a principal tela do aplicativo. Recebe automaticamente os comandos do usuário e gera ou uma resposta ou uma ação e uma resposta, dependendo da entrada. As atividades realizadas por esta tela são as principais do programa – receber as solicitações sobre a residência e gerar as respostas. Uma vez que o aplicativo é construído para receber comandos de voz, o layout não é muito importante, porém uma entrada para a entrada manual.

As atividades realizadas pela classe são:

1. Receber e separar as mensagens do usuário.
2. Analisar gramaticalmente as frases inseridas pelo usuário.
3. Definir se o pedido do usuário é uma pergunta ou uma solicitação de mudança do status de porta.
4. Organizar as palavras chave para cada tipo de solicitação e procurar no banco de dados a porta do dispositivo procurado.
5. Enviar os comandos para consultar o status portas monitoradas pelo Arduíno Uno/Ethernet Shield.
6. Se necessário, enviar os comandos para alterar o status portas monitoradas pelo Arduíno Uno/Ethernet Shield.
7. Gerar a resposta à pergunta/solicitação realizada pelo usuário.

Foxy – programa carregado no Arduino Uno

O programa carregado no Arduino se comunica com o roteador através do cabo de rede pela porta 80 e controla as saídas digitais de 2 a 9. A comunicação entre o aplicativo Android e o Arduíno se dá pelo envio de “strings” ao roteador que está conectado à placa Ethernet Shield e ao dispositivo Android através da WiFi. O Arduíno recebe os sinais enviados pelo dispositivo Android e também envia strings com mensagens de monitoramento indicando os status das portas. As saídas digitais do Arduíno podem ser ligadas a uma placa com relés para realizar o controle de dispositivos de alta/média potência.

RESULTADOS

Para realizar o teste do aplicativo, cada led foi considerado como um dispositivo, sendo:

- LED 01: my room light
- LED 02: kids room window
- LED 03: garage gate

- LED 04: kids room light
- LED 05: alarm

Nesta configuração temos os seguintes tipos de dispositivos:

- Luzes: light
- Janelas: Windows
- Portão: Gate
- Alarme: Alarm

Foram realizados testes para todas as combinações possíveis de estado de dispositivo e pergunta, ou seja, foram realizadas todas as perguntas e solicitações possíveis para cada dispositivo em cada um dos seus estados.

Para isso, primeiramente todos os dispositivos foram setados para o estado ACIONADOS como condição inicial e as seguintes perguntas/solicitações foram feitas para cada endereço considerando os status para os respectivos tipos de alvo (ex: on para light e open para gate) e foram recebidas as seguintes respostas:

TODOS OS DISPOSITIVOS ACIONADOS

1. Is [ENDEREÇO] on/open?
Yes, he [ENDEREÇO] is on/open.
2. Is [ENDEREÇO] off/closed?
No, the [ENDEREÇO] is not off/closed.
3. Turn on/open [ENDEREÇO]?
The [ENDEREÇO] is already on/open.
4. Turn off/close [ENDEREÇO].
The [ENDEREÇO] is now off/closed.

Em seguida, todos os dispositivos foram setados para o estado DESATIVADOS como condição inicial e as seguintes perguntas/solicitações foram feitas para cada endereço considerando os status para os respectivos tipos de alvo (ex: on para light e open para gate) e recebidas as respectivas respostas:

TODOS OS DISPOSITIVOS DESATIVADOS

1. Is [ENDEREÇO] on/open?
No, the [ENDEREÇO] is not on/open.
2. Is [ENDEREÇO] off/closed?
Yes, he [ENDEREÇO] is off/closed.
3. Turn off/close [ENDEREÇO]
The [ENDEREÇO] is already off/closed.
4. Turn on/open [ENDEREÇO]?
The [ENDEREÇO] is now on/open.

Após estes testes, foram feitas perguntas para todos os dispositivos alvo (TARGETS) apresentados na Tabela 4.1 com as seguintes configurações:

TODOS OS DISPOSITIVOS ATIVADOS:

1. Whitch [TARGET] is on?
The [TARGET] are on/open.
2. Is there any [TARGET] on?
YES
3. Is there all [TARGET] on?
YES
4. Turn on all [TARGET].
All the [TARGET] are already on/open.
5. Whitch [TARGET] is off?
There is no [TARGET] off.
6. Is there any [TARGET] off?
NO
7. Is there all [TARGET] off?
NO
8. Turn off all [TARGET].
All the [TARGET] are now off/closed.

TODOS OS DISPOSITIVOS DESATIVADOS:

1. Whitch [TARGET] is on?
There is no [TARGET] on/open.
2. Is there any [TARGET] on?
NO
3. Is there all [TARGET] on?
NO
4. Turn on all [TARGET].
All the [TARGET] are now on/open.
5. Whitch [TARGET] is off?
The [TARGET] are already off/closed.
6. Is there any [TARGET] off?
YES
7. Is there all [TARGET] off?
YES
8. Turn off all [TARGET].
All the [TARGET] are already off/closed.

Uma vez que há apenas um tipo de TARGET em dois endereços, foram feitas as seguintes perguntas para a configuração “my room light” =ON e “kids room light” = OFF.

1. Whitch [LIGHT] is on?
The my room light is on
2. Is there any [LIGHT] on?
Yes
3. Is there all [LIGHT] on?

- NO
4. Turn on all [TARGET].
All lights are now on.
 5. Whitch [LIGHT] is off?
The kids room light is off.
 6. Is there any [LIGHT] off?
YES
 7. Is there all [LIGHT] off?
NO
 8. Turn off all [LIGHT].
All lights are now off.

Os resultados obtidos para as mesmas perguntas com a configuração “my room light” = OFF e “kids room light” = ON foram o contrário das obtidas com a condição anterior, como esperado.

Todos os LEDS foram ACIONADOS ou DESATIVADOS quando solicitado.

CONCLUSÕES

O projeto aqui proposto alcançou seu objetivo de gerar um sistema de reconhecimento de fala capaz de monitorar e alterar pontos automatizados de uma residência utilizando um smartphone e uma placa de controle dos dispositivos formada por um Arduino Uno concatenado a um “Ethernet Shield” comunicando-se através de uma rede Wi-Fi local visando maior conforto e segurança para indivíduos com deficiência visual.

Foi possível a geração de um sistema flexível, e adaptável, capaz de lidar com um grande número de informações a partir de um pequeno número de entradas. O modo como o programa lida com as entradas do banco de dados possibilita uma fácil integração de novos dispositivos e palavras, o que torna o sistema altamente personalizável e ideal para ser utilizado em aplicações residenciais onde cada casa possui uma planta diferente.

A interface para pessoas com deficiência visual também mostrou-se eficiente, uma vez que é possível realizar tanto a consulta quanto solicitações aos dispositivos através da fala, sendo necessário auxílio de uma segunda pessoa apenas para instalação. O que mostra o poder da tecnologia e sua influência positiva na vida das pessoas com deficiência, onde mesmo uma aplicação simples pode trazer um grande impacto e aumento da qualidade de vida.

Um ponto importante a ser considerado é o reconhecimento de fala, que depende da capacidade do hardware captar e tratar os sinais de áudio recebidos. Por isso é importante que o microfone utilizado para a captação da fala seja de qualidade.

Uma sugestão para trabalhos posteriores seria a implementação em uma residência real, controlando os dispositivos da residência, expandir para outras

plataformas como iOS e traduzir o programa para outras linguagens, a fim de atingir um público maior, beneficiando mais pessoas.

REFERÊNCIA BIBLIOGRÁFICA

ALVES, Adriano Ferreira et al. Automação residencial: evolução da segurança e conforto na sua moradia. RCMOS-Revista Científica Multidisciplinar O Saber, v. 2, n. 2, p. 383-388, 2022.

Arduino. (s.d.). *Download the Arduino Software*. Acesso em 01 de 03 de 2024, disponível em Arduino:
<https://www.arduino.cc/en/Main/Software>

CAROLINO, Suênia Fernandes. Histórico da automação residencial no Brasil: exemplos de aplicações. 2023. Trabalho de Conclusão de Curso.

EIRAS, Caio Viana Silva; DOS SANTOS SILVA, Leandro; PIRES, Rachel Cristina Santos. Processos de Automação Residencial na Construção Civil e Suas Diretrizes nas Etapas Construtivas. Epitaya E-books, v. 1, n. 32, p. 179-191, 2023.

FEITOSA, Manoel Evandro de Carvalho et al. Uma análise de implantação de projetos de automação residencial no mercado de São Luis-MA. 2022.

Google. (s.d.). *Adapter*. Acesso em 01 de 04 de 2024, disponível em Android Developers:
<http://developer.android.com/reference/android/widget/Adapter.html>

Jonathan, S. (2011). *Head First Android Development*. Sebastopol: O'Reilly.

Lavrador, M. (s.d.). *Automação residencial com comando de voz. Chegou a hora?* Acesso em 02 de 04 de 2024, disponível em AURESIDE:
<http://www.aureside.org.br/temastec/default.asp?file=softwares02.asp>

Meddieks, Z. (2011). *Programming Android*. Sebastopol: O'Reilly.

Mednieks, Z. (2014). *Enterprise Android*. Indianapolis: Wrox.

Molina, F. (s.d.). *USP desenvolve projetos ligados à “Internet das Coisas”*. Acesso em 30 de 04

de 2024, disponível em USP Brasil:
<http://www5.usp.br/14645/USP->

DESENVOLVE-PROJETOS-LIGADOS-A-
INTERNET-DAS-COISAS/

PINTO, Jéssica Ferreira. Tecnologias de automação de edifícios para melhoria do conforto e usabilidade de um edifício empresarial. 2019. Tese de Doutorado.

Six, J. (2012). *Application Security*. Sebastopol: O'Reilly.

Stark, J. (2010). *Building Android Apps*. Sebastopol: O'Reilly Media, Inc.

W. FRANK ABLESON. (2012). *Android in Action*. SHELTER ISLAND: MANNING.

COSTA, R. L. **SQL Guia Prático**. Rio de Janeiro: Brasport.