



**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO” -
UNESP**

CÂMPUS DE BAURU - FEB

Rafael Esperança Rother

**AUTOMAÇÃO RESIDENCIAL: ANÁLISE SOBRE A
IMPLEMENTAÇÃO DE UM SISTEMA DE VOZ**

Bauru, 2019

Rafael Esperança Rother

AUTOMAÇÃO RESIDENCIAL: ANÁLISE SOBRE A IMPLEMENTAÇÃO DE UM SISTEMA DE VOZ

Trabalho de Graduação do Curso de
Engenharia Elétrica apresentado à
Faculdade de Engenharia de
Bauru/UNESP.

Orientador: Prof. Dr. Marcelo Nicoletti
Franchin

Bauru, 2019

Rother, Rafael Esperança.

Automação Residencial: Análise sobre a
implementação de um sistema de voz / Rafael
Esperança Rother, 2019

47 f. : il.

Orientador: Marcelo Nicholetti Franchin

Monografia (Graduação)-Universidade Estadual
Paulista. Faculdade de Engenharia de Bauru,
Bauru, 2019

1. Automação Residencial. 2. Domótica. 3.
Reconhecimento de voz. I. Universidade Estadual
Paulista. Faculdade de Engenharia de Bauru. II.
Título.

FOLHA DE APROVAÇÃO

Aluno: **Rafael Esperança Rother**

Título: **Automação Residencial: Análise sobre a Implementação de um Sistema de Voz**

Trabalho de Graduação defendido e aprovado em **27 de November de 2019**, com

MÉDIA: 7,0 (SETE INTEIROS), pela comissão julgadora:

M. L. F. Franchin

Prof. Dr. Marcelo Nicoletti Franchin

A. L. Andreoli

Prof. Dr. Andre Luiz Andreoli

P. S. da Silva

Prof. Dr. Paulo Sérgio da Silva

Mário Eduardo Bordon

Dr. Mário Eduardo Bordon

Coordenador do Curso de Graduação em
Engenharia Elétrica

AGRADECIMENTOS

Agradeço a Deus por seu infinito amor e à Maria, Nossa Senhora, por sua intercessão e proteção.

Agradeço aos meus pais, Rosemary Aparecida Esperança Rother e José Eduardo Rother, por batalharem sempre por mim, me ensinarem tantas coisas boas e me trazerem até aqui. Motivados sempre pelo grande amor por mim.

Agradeço a todos meus amigos, que me apoiaram de alguma forma e em especial a Raphael Gusman Caccia e a Marina Pierangelli Murilha, que são como irmãos para mim. Marina é uma das maiores responsáveis pelo meu crescimento pessoal e pela colossal ajuda durante toda faculdade, mesmo estando longe. Seu suporte sempre foi de vital importância para mim, inclusive nesse trabalho de graduação.

Agradeço também ao professor José Angêlo Cagnon pela orientação na primeira parte do trabalho de graduação e ao professor Marcelo Nicoletti Franchin pela orientação nesta segunda e última parte.

*“O sol só vem depois
É o astro rei, okay, mas vem depois”*
Emicida

LISTA DE FIGURAS

Figura 1 - Áreas da domótica	5
Figura 2 - Exemplo de casa domótica	7
Figura 3 - Gatebox	8
Figura 4 - Exemplo de sinal de voz gravado	10
Figura 5 - Processo de amostragem	11
Figura 6 - Processos de quantificação e codificação	11
Figura 7 - Transformada de Fourier janelada	12
Figura 8 - Módulos do BitVoicer Server	15
Figura 9 - Nível Mínimo de Áudio	16
Figura 10 - Google Home	18
Figura 11 - Google Home desmontado	20
Figura 12 - Visão geral do Google Home	21
Figura 13 - Esquema de montagem	22
Figura 14 - Esquemático do Exemplo 8.1	23
Figura 15 - Configuração do dispositivo no BitVoicer Server	24
Figura 16 - Circuito do Exemplo 8.1 montado e funcionando	24
Figura 17 - Esquemático do Exemplo 8.2	25
Figura 18 - Configuração da guia <i>Cues</i> para o Exemplo 8.2	25
Figura 19 - Comando adicionado ao anagrama no Exemplo 8.2	26
Figura 20 - Circuito do Exemplo 8.2 montado e funcionando	27
Figura 21 - Circuito do Exemplo 8.2 ao reconhecer a palavra de ativação	27
Figura 22 - Resultado do Exemplo 8.4	28
Figura 23 - Configuração para saída de áudio do sintetizador de voz	29
Figura 24 - Configuração do dispositivo	30
Figura 25 - Configuração de comunicação	30
Figura 26 - Configuração da aba <i>Cues</i>	31
Figura 27 - Sentença para piscar LED	35
Figura 28 - Adição de comandos para piscar LED	35
Figura 29 - Esquemático da primeira montagem	36
Figura 30 - Adição de novas sentenças	37
Figura 31 - Comando para acender o LED	37
Figura 32 - Comando para apagar o LED	37

Figura 33 - Esquemático da montagem com o LED inserido	38
Figura 34 - Sentenças para o Servo Motor	39
Figura 35 - Comandos para o Servo Motor	39
Figura 36 - Esquemático da montagem com a inclusão do Servo Motor	40
Figura 37 - Sentenças para o acendimento da lâmpada	41
Figura 38 - Esquemático da montagem adicionando o relé e a lâmpada	41

RESUMO

A integração de dispositivos eletrônicos e métodos computacionais, permite que uma residência ou um espaço comercial tenha total controle e gestão de equipamentos. Isso possibilita maior conforto, acessibilidade, segurança e economia de recursos. A integração de um sistema de voz em um sistema automatizado é um passo a mais na melhoria destes benefícios, já que permite, por exemplo, que um deficiente visual possa usar esse sistema, pois ele não necessita o uso de smartphones ou displays. Este trabalho propõe integrar um programa já existente de interpretação e reconhecimento de voz, com a plataforma Arduino que será responsável pelo controle e automação de dispositivos.

Ao final do desenvolvimento, houve a obtenção de resultados satisfatórios ao conseguir a integração dos componentes propostos. Com um comando de voz “acender a lâmpada”, por exemplo, um relé foi acionado, acendendo uma lâmpada.

Palavras-chave: Automação; Arduino; Domótica; Automação Residencial; Microcontrolador; Reconhecimento de voz; Acessibilidade.

ABSTRACT

The electronical devices integration and computational methods may fully manage and control homes and comercial places. It enables people to have better life comfort, accessibility, safety and resource saving. A voice system integration in an automatized system improves a person lifestyle while using that, not requiring smartphones or displays for. This work integrates a pre-concept interpretation and voice program recognition with an Arduino microcontroler that will be responsible for the dispositive control and automation.

At the end of the development, satisfactory results were obtained by achieving the integration of the proposed components. With a voice command “turn on the lamp”, for example, a relay was activated, turning on a lamp.

Key Words: Automation; Arduino; Domotics; Residencial Automation; Microcontroler; Voice Recognition; Accessibility.

SUMÁRIO

1. INTRODUÇÃO	1
2. REFERENCIAL TEÓRICO	4
2.2. AUTOMAÇÃO	4
2.3. DOMÓTICA	5
2.4. ARDUINO	8
2.5. RECONHECIMENTO DE VOZ	9
3. MATERIAIS	14
3.1. BITVOICER SERVER	15
3.1.1. FUNCIONAMENTO DO GOOGLE HOME	17
4. DESENVOLVIMENTO	22
4.1. DOS EXEMPLOS DO BITVOICER SERVER	23
4.2. DA REALIZAÇÃO DA DOMÓTICA	29
5. CONCLUSÃO	42
6. REFERÊNCIAS	44

1. INTRODUÇÃO

O ser humano sempre está em busca de novas tecnologias, com objetivo de facilitar seu dia-a-dia, organizar, simplificar e, principalmente, controlar suas tarefas e o meio em que vive. Para esse propósito, a computação é seu maior aliado, já que tem se tornado comum às pessoas estarem rodeadas por dispositivos com algum tipo de processamento de informações não só na engenharia, mas também na medicina, no comércio, no cotidiano, etc.

Com o convívio cada vez mais comum com novas tecnologias de informação e a facilidade para se controlar objetos, não há razão para não estender esse conhecimento para a própria casa, para torná-la mais confortável, mais segura, mais flexível para atender as necessidades e até gerar economias. Disso vem o interesse na automação residencial, conhecida como domótica.

Na época da Revolução Industrial na Inglaterra, a chegada do gás, da água corrente e posteriormente da eletricidade nas casas, se deu uma revolução doméstica, em consonância com intensas mudanças em várias outras áreas. A introdução destas tecnologias domésticas significou o começo da racionalização e da mecanização da casa. O fato de que este processo foi produzido de forma lenta foi devido ao desinteresse da maioria dos arquitetos da época pelos avanços tecnológicos, que, no geral, se preocupavam mais com a aparência dos edifícios do que com a funcionalidade (Domínguez e Vacas, 2006).

O início de uma nova revolução doméstica, é datada no final da década de 1970, nos Estados Unidos, que, segundo Muratori e Dal Bó (2011), foi quando surgiram os primeiros módulos “inteligentes”, cujos comandos eram enviados pela própria rede elétrica da residência, no conceito de PLC (*Power Line Carrier*). Tratava-se de soluções simples, praticamente não integradas e que resolviam situações pontuais, como ligar remotamente algum equipamento ou luzes (Muratori e Del Bó, 2011).

Com a criação do controlador lógico programável (CLP), um dispositivo eletrônico digital, que pode ser programado e utilizado para controle através de suas entradas e saídas, novas possibilidades se abriram para a automação industrial, predial e residencial. O CLP passou a ter grande utilização devido aos avanços em sua tecnologia, como o surgimento dos circuitos integrados, uso microprocessadores e microcontroladores, melhoras no poder computacional e possibilidade de integração em rede (Parede e Gomes, 2011).

Há um grande crescimento no mercado da domótica, pois interessa aos consumidores, cada vez mais personalizar, controlar e interagir com o ambiente em que se vive, deixando-o mais prático e confortável. As possibilidades são variadas: desde controle de iluminação, até monitoramento a distância, passando por outras infinitas possibilidades. Em entrevista, Medrado diz que: “Em meio à correria é muito bom contar com a ajuda dos recursos tecnológicos e ainda poder ter uma casa mais bonita e valorizada” (Medrado, 2008).

Dentro dos itens típicos de um trabalho de automação residencial, encontramos sensores, softwares, controladores, atuadores e interfaces homem/máquina (Muratori e Del Bó, 2011).

Eliminar uma interface física e implementando comandos de voz em um sistema automatizado, resultaria em muitos benefícios aos portadores de alguma deficiência, principalmente a visual. Poderiam realizar atividades até então consideradas difíceis ou impossíveis, levando uma maior autonomia e independência. Em alguns países da Europa, é comum que pessoas com alguma limitação física busquem isso. Acessibilidade é a palavra-chave (Ferreira, 2010).

Com vista nisso, o desenvolvimento de sistemas por controle de voz, tem ganhado força. Os lançamentos de mais destaque nessa área, são o Amazon Echo, lançado em 2015 e o Google Home, lançado em 2016. Ambos são alto-falantes inteligentes, que possuem suporte integrado à automação residencial e são controlados pelos assistentes e permitem interações através dos seus assistentes virtuais Amazon Alexa e Google Assistente.

Neste sentido, objetivo deste trabalho é desenvolver um modelo de automação residencial utilizando a plataforma Arduino e implementação de controle por voz. O reconhecimento de voz, será feito através do software BitVoicer Server. O resultado final deverá permitir o controle de vários componentes, como lâmpadas, motores e etc., quando um certo padrão de fala é reconhecido, sendo assim, o modelo de uma residência automatizada.

No capítulo 2, será apresentado o retrospectivo histórico e o referencial teórico sobre domótica, Arduino e o reconhecimento de voz. Também no capítulo 2, será mostrada uma das opções já disponíveis no mercado para a implementação de um sistema de voz em uma residência automatizada, o Google Home.

O capítulo 3 mostrará os materiais utilizados no desenvolvimento do trabalho.

No capítulo 4, serão descritos os passos do desenvolvimento do projeto, as montagens dos circuitos e os códigos necessários para o funcionamento. Também serão exibidos os resultados obtidos em cada etapa.

O capítulo 5 será dedicado às considerações finais, conclusões, dificuldades encontradas e possíveis melhorias para projetos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo é reservado para as referências teóricas utilizadas no desenvolvimento e estudo desse trabalho.

2.1. AUTOMAÇÃO

Dorf e Bishop (2011) definem a automação como uma tecnologia que utiliza comandos programados para operar um dado processo, combinados com retroação de informação para determinar que os comandos sejam executados corretamente, frequentemente utilizada em processos antes operados por seres humanos

Máquinas substituindo humanos em tarefas reduzem erros, aumentam a produção, tem maior eficiência, controlam melhor os recursos, sendo uma grande vantagem sua aplicação industrial, abaixando os custos e aumentando os lucros de uma linha de produção (Silveira, 2003). Contudo, para Bill Gates, a linha já deve seguir um modelo eficiente: “A primeira regra de qualquer tecnologia utilizada nos negócios é que automação aplicada a uma operação eficiente aumentará a eficiência. A segunda é que a automação aplicada a uma operação ineficiente aumentará a ineficiência.” (Bill Gates, 1995).

A automação também pode ser aplicada em outras áreas diferentes da industrial. Também é possível automatizar veículos, prédios, residências, procedimentos cirúrgicos.

Os avanços tecnológicos nos permitem ter cada vez mais objetos automatizados, que geram dados de forma autônoma, que podem ser interconectados através da internet. Este tipo de automação, podemos chamar de Internet das Coisas (do inglês: *Internet of Things – IoT*) (Ferrasi, 2016).

O que Lemos (2013) define como Internet das Coisas é praticamente a mesma definição de automação: conjunto de redes, sensores, atuadores, objetos ligados por sistemas informatizados que ampliam a comunicação entre pessoas e objetos e entre objetos de forma autônoma, automática e sensível ao contexto. Objetos passam a “sentir” a presença de outros a trocar informações e a mediar ações entre eles e entre humanos.

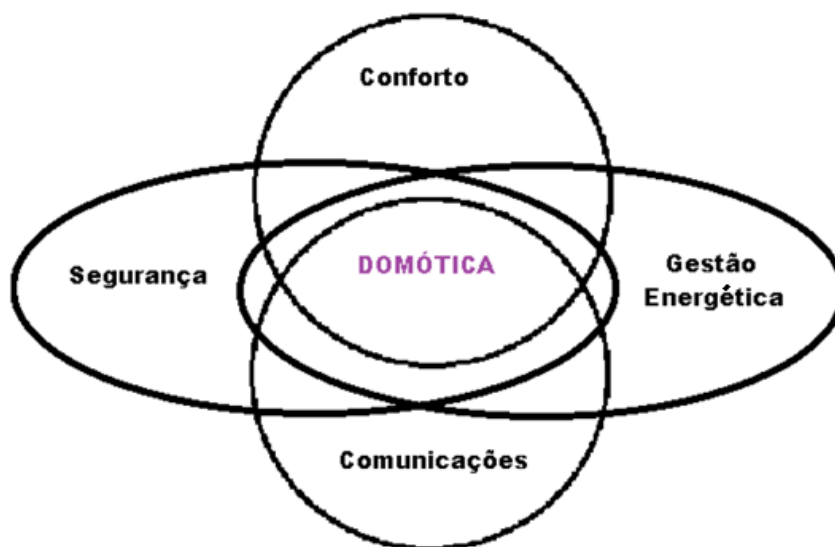
Partindo desta possibilidade, é que se tratará o presente trabalho, criando um dispositivo que poderia ser utilizado enquanto automação residencial ou, como é conhecida, domótica.

2.2. DOMÓTICA

A automação residencial, também é conhecida como domótica (do latim, domus, que significa “casa” e do grego, automática, que significa “funciona autonomamente”) (Domínguez e Vacas, 2006).

Segundo Ferreira (2010), o conceito de domótica, introduzido na França nos anos 1980, refere-se à integração de diversas tecnologias no ambiente doméstico mediante o uso simultâneo de eletricidade, eletrônica, informática e telecomunicações, buscando como resultado melhorar aspectos como segurança, conforto, gestão energética, flexibilidade de uso dos espaços, e, conseqüentemente, a qualidade de vida de seus moradores. Estas áreas que a domótica atua visando melhorias é ilustrada na Figura 1.

Figura 1 - Áreas da domótica



Fonte: Ferreira, 2010

A automação residencial tem sido um grande desafio para as engenharias que se relacionam com tecnologia. O mercado, cada vez mais, tende a desenvolver dispositivos inteligentes das mais variadas formas contidas no ambiente residencial. Atualmente existem dispositivos que interagem com sistemas de segurança,

monitoramento, sistemas de áudio e vídeo, entre outras opções que visam o conforto do usuário no ambiente (Ribeiro, 2018).

Para Lucena (2006) as diversas aplicações da domótica dependem do limite da tecnologia, da necessidade do morador e da criatividade do desenvolvedor.

A diferença de gastos financeiros na parte elétrica entre uma residência convencional e uma automatizada não está tanto na parte central da automação, que vem como o grande diferencial, e sim no que é necessário para que ela possa ser bem aproveitada, como na utilização de luminárias para a criação de vários tipos de ambiente, por exemplo (Guimarães Filho, Souza e Júnior, 2017).

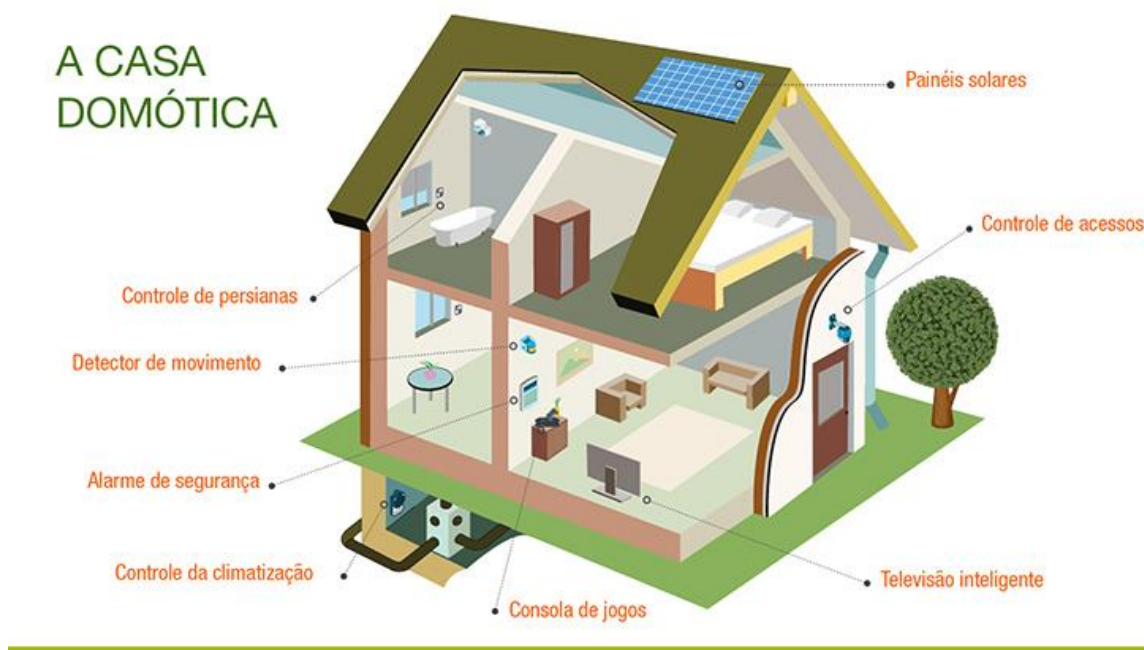
O investimento em automação residencial compensa pois permite maior conforto e valorização do imóvel, mesmo elevando o custo da construção, já que casas desse tipo são vendidas com mais facilidade (Guimarães Filho, Souza e Júnior, 2017).

Como qualquer novidade, a domótica inicialmente é percebida como um símbolo de status e modernidade. Porém, em um segundo momento, os benefícios proporcionados por essa tecnologia superam qualquer preconceito existente. Em um futuro próximo, a tendência é que essa tecnologia se torne uma necessidade vital e um fator de economia, assim como ocorreu com os celulares (Souza, 2016).

Existem alguns itens que são indispensáveis em um de um trabalho de automação residencial: sensores, softwares, controladores, interfaces homem/máquina e um meio para que haja comunicação entre os componentes. É recomendado que para esse tipo de instalação tenha uma central de automação no centro da residência, localização estratégica, onde serão recebidos, processados e enviados os dados (controlador) e tenha uma interface para interação (normalmente um display *touchscreen* nos sistemas atuais) (Silveira, 2003).

A Figura 2 exemplifica uma casa domótica, onde são mostradas várias possibilidades de controle e obtenção de informações.

Figura 2 - Exemplo de casa domótica



Fonte: Iberdrola, 2019

Sensores são dispositivos capazes de “sentir” um estímulo físico/químico. A resposta de um sensor será traduzida por um transdutor, em algo capaz de ser compreendido como um sinal detectável (elétrico ou mecânico). O sensor faz parte do transdutor e assim será possível a medição de uma grandeza. Todavia, o conjunto é comumente chamado de sensor (Balbinot e Brusamarello, 2006).

Alguns exemplos de sensores: termômetros, microfones, piezoelétricos, contadores Geiger, acelerômetros, voltímetros, sensores de proximidade, sensores de presença, sensores de luminosidade, sensores de umidade.

Os atuadores são dispositivos que quando respondem um comando, respondem com um movimento. Por exemplo: atuadores pneumáticos, atuadores hidráulicos e motores elétricos, todos muito utilizados na domótica.

Os dispositivos necessitam comunicar-se entre si, por isso deve existir um meio para isso. Cabos e fios são usados desde o começo dos estudos da domótica, mas para economia, comodidade e evitar excesso de fios, tecnologias como o *wi-fi* e o *bluetooth* começaram a ser empregadas (Lemos, 2007).

Além da interface da central de automação, outras interfaces poderão atuar no sistema como controles remotos, controles fixos, displays, smartphones. Esta interface é necessária para que ocorra a interação entre homem e máquina, para que eles possam “conversar”. O uso de uma interface por voz, dá a sensação de

uma conversa literal, pois mecanismos de inteligência artificial estão sendo desenvolvidos para ficar o mais natural possível, reconhecendo melhor a fala e dando respostas mais convincentes (Garbin, 2010). Existe uma interface japonesa chamada *Gatebox*, que além da inteligência artificial, tem a projeção de uma personagem em holograma. A Figura 3 apresenta um homem interagindo com o holograma da *Gatebox*.

Figura 3 - Gatebox



Fonte: Site Oficial Gatebox

2.3. ARDUINO

Nos anos 60, o acesso a computadores era muito restrito, devido ao custo do equipamento e falta de pessoas com conhecimento técnico em utilizar a tecnologia. Apenas grandes empresas e universidades possuíam computadores. Porém com os avanços na área de tecnologia de semicondutores, foram desenvolvidos diferentes tipos de computadores, com diferentes propósitos e diferentes tamanhos. A palavra “computador” passou a fazer parte do cotidiano (Gaonkar, 2002).

Todos computadores possuem os mesmos componentes básicos: microprocessador, memória, sistema I/O (*input/output*, que são as entradas e saídas), barramento de dados (Gaonkar, 2002).

Computadores completos que podem ser construídos em apenas um chip, são chamados de microcontroladores. São usados primariamente com a função de controle. São exemplos de microcontroladores: Zilog Z8, Intel MCS 51 series, Microchip PIC. (Gaonkar, 2002).

A plataforma computacional usada nesse trabalho será o Arduino, devido ao seu baixo custo, versatilidade e simplicidade.

O Arduino foi criado em 2005 por um grupo de pesquisadores no Ivrea Interaction Design Institute, com o objetivo de elaborar um dispositivo barato, funcional e prático, sendo acessível para estudantes e projetistas amadores. Assim, foi criada uma placa composta por um microcontrolador ATMEL ATmega328, circuitos de entrada e saída e desenvolvida uma IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) para se programar o Arduino, conectando-o a um computador (Thomsen, 2014).

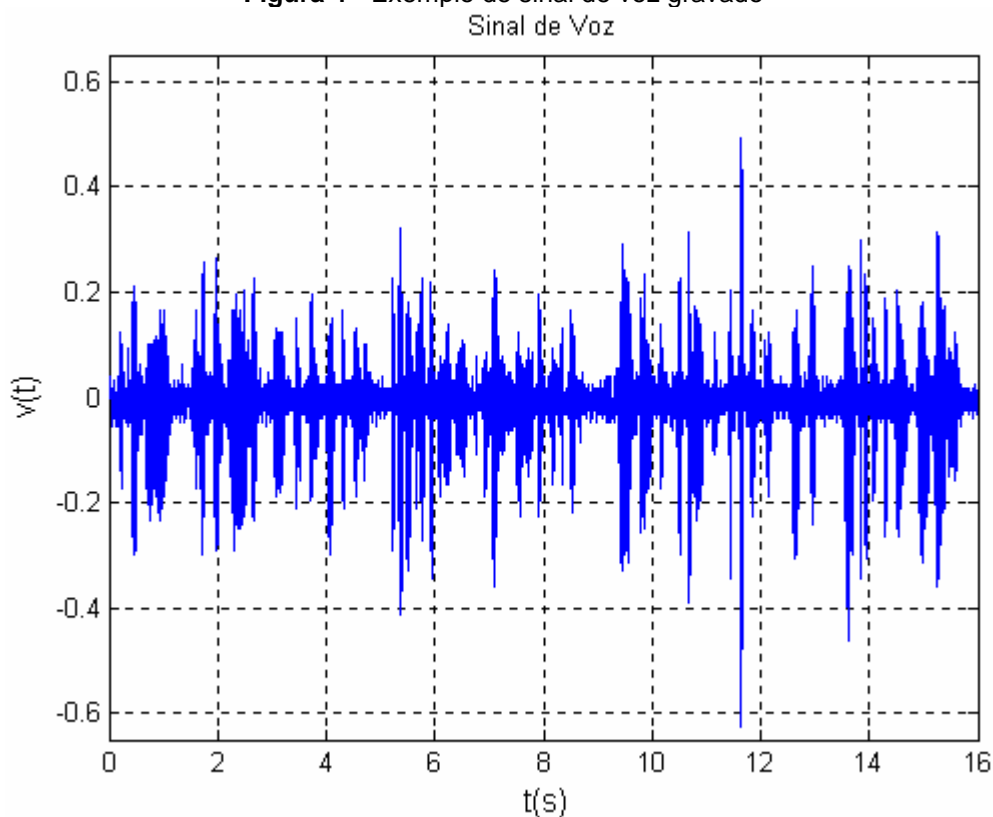
O Arduino possui vasta quantidade de sensores, módulos (placas que contém sensores e componentes auxiliares) e Shields (placas que podem ser acopladas ao Arduino), expandindo suas funcionalidades, dando a ele possibilidades de uso praticamente infinitas (Thomsen, 2014).

2.4. RECONHECIMENTO DE VOZ

A expressão “reconhecimento de voz” tem mais de um sentido podendo se referir a tecnologias distintas. O processamento de voz pode ser aplicado em quatro principais áreas:

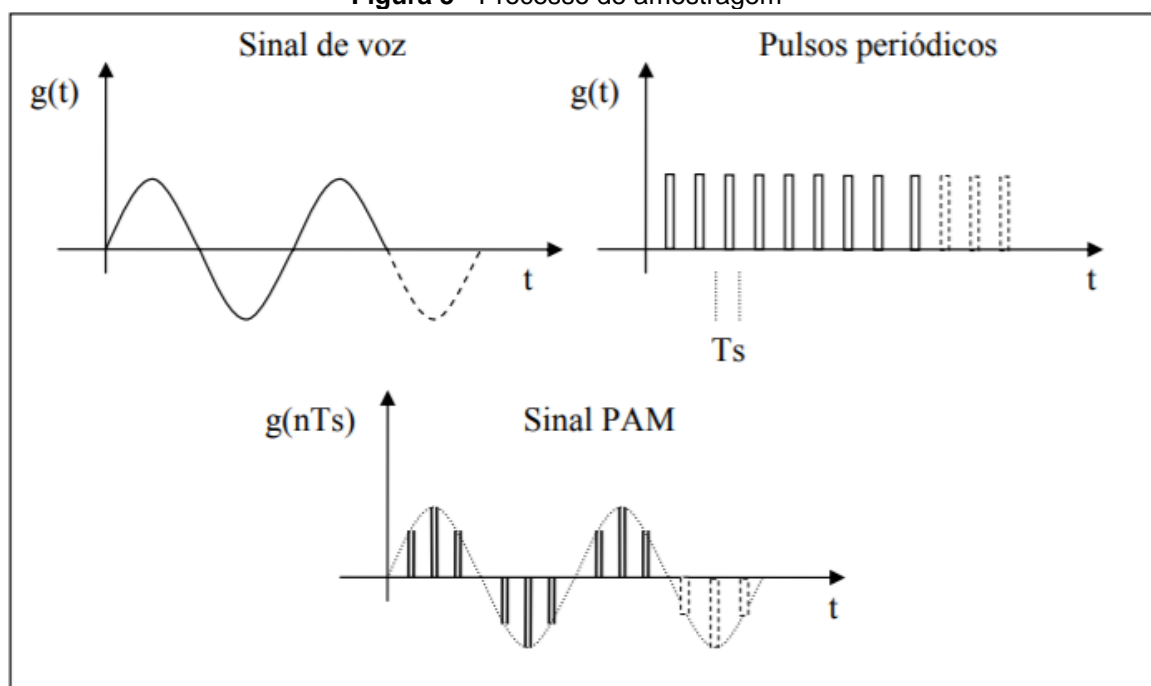
- Reconhecimento de palavras, usadas para comandos de voz mais simples, já que eles serão comparados com comandos que o sistema já conhece de antemão. Exemplo: centrais de atendimento telefônico.
- Reconhecimento de fala natural, envolvendo frases mais complexas, isto é, um conjunto de palavras com um sentido semântico. A fala reconhecida é então transformada em texto; Exemplo: Ditado de documentos
- A síntese de voz, que é o procedimento inverso do reconhecimento de fala. O sintetizador recebe um texto na forma digital e o transforma em ondas sonoras.
- Autenticação, que se baseia em que a voz é única para cada pessoa e pode ser utilizada para identificação (Guilhoto e Rosa, 2001).

O primeiro passo para o reconhecimento de voz é a captação das ondas da vibração da voz (que tem um exemplo mostrado na Figura 4), por meio de microfones e transmitido para uma placa de som, que fará a amostragem do sinal analógico. Este será digitalizado através de uma técnica chamada *Pulse Code Modulation* (Guilhoto e Rosa, 2001).

Figura 4 - Exemplo de sinal de voz gravado

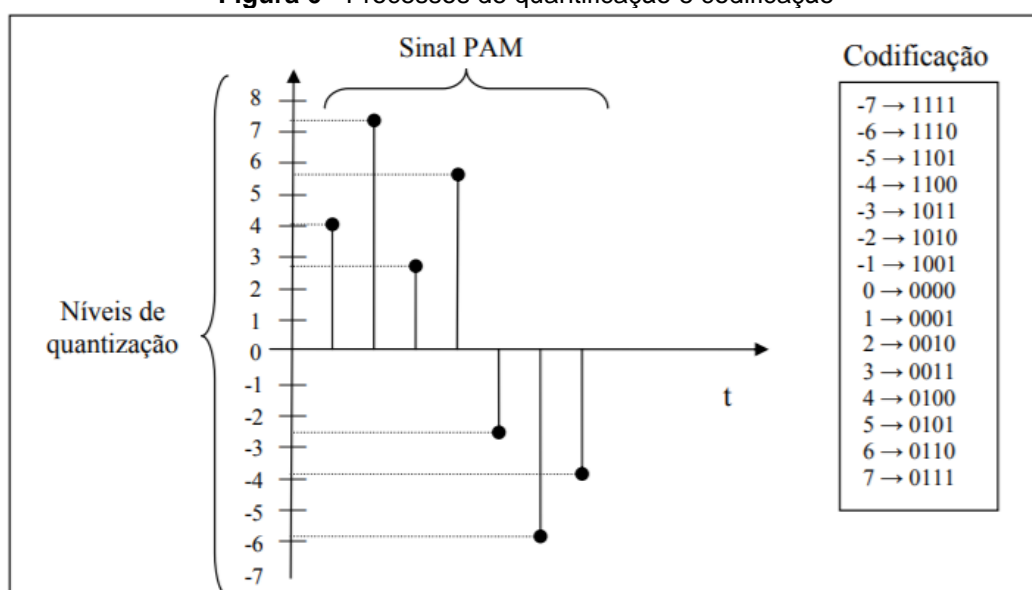
Fonte: ResearchGate

Tal técnica, para Santos (2006), consiste na representação de um sinal por meio de uma sequência de pulsos codificados, obtido através da sua representação na forma discreta, no tempo e na amplitude. O processo de amostragem, resulta em uma representação chamada *Pulse Amplitude Modulation* (PAM), que matematicamente, consiste no resultado da multiplicação de um trem de pulsos de amplitude 1, pelo sinal original a ser amostrado. O processo é exemplificado na Figura 5.

Figura 5 - Processo de amostragem

Fonte: Santos, 2006

Após o processo de amostragem, Santos diz que o resultado obtido deverá ser quantizado, seguindo uma lei uniforme, onde a separação entre os níveis do sinal é espaçada igualmente, ou uma lei variável. Para a representação dos valores que foram amostrados, utiliza-se de um processo de codificação, onde cada evento discreto de um código é chamado de símbolo. Cada símbolo representa um valor distinto do conjunto de valores discretos (Santos, 2006). Os processos de quantização e codificação são ilustrados na Figura 6.

Figura 6 - Processos de quantificação e codificação

Fonte: Santos, 2006

O sinal amostrado é filtrado para separá-lo de ruídos e interferências. Esta forma do sinal é uma *stream* de amplitudes que representa o sinal analógico. No entanto, um software de reconhecimento de voz não pode trabalhar diretamente baseado nesta *stream* pois é muito complicado procurar padrões que possam ser relacionados com o que está sendo dito. É neste sentido que esta *stream* é transformada em um conjunto de bandas de frequência discreta usando a transformada rápida de Fourier janelada (*Windowed Fast Fourier Transform - WFFT*) (Guilhoto e Rosa, 2001).

Souza afirma que a transformada rápida de Fourier janelada é a seleção de uma parte do sinal através da multiplicação dele por uma janela com posição variável no tempo, ou seja, obtendo diversos segmentos do sinal no tempo. Assim, cada espectro de frequência mostra um conteúdo de frequência durante curto período e a totalidade dos espectros contém a evolução do conteúdo de frequência com o tempo de todo sinal em análise (Souza, 2004).

Assim, é possível mapear o sinal original em uma função bidimensional de tempo e frequência, podendo ser feita uma análise da frequência de um sinal localmente em uma janela de tempo, como ilustrado pela Figura 7 (Souza, 2004).

Figura 7 - Transformada de Fourier janelada



Fonte: Souza, 2004

A *stream* de entrada é agora representada por um conjunto que se aproxima do que é audível pelo ser humano. Nessa etapa do processo, o som pode necessitar ser sincronizado, pois as diferentes vozes não costumam utilizar o mesmo tom e nem sempre possuem a mesma velocidade (Guilhoto e Rosa, 2001 e Pereira, 2009).

Desta maneira é possível obter separação da digitalização em frações, que segundo Pereira, representam sons fonéticos, não maiores que uma sílaba. Em

seguida, o programa compara os sons captados com fonemas conhecidos e presentes em seu banco de dados que correspondam ao idioma que o locutor tenha falado. Em outras palavras, é aplicado um método de busca para associar as saídas com padrões de palavras e da voz de quem as emitiu (Pereira, 2009).

Se o sistema for apenas para reconhecimento de comandos (reconhecimento de palavras), o modelo linguístico a ser gravado no banco de dados é simplificado, necessitando apenas os comandos necessários, com diferentes formas de pronúncia. Caso seja um sistema para reconhecimento de fala natural, é preciso um banco de dados com um dicionário com todas as palavras da língua (ou grande parte). Também é preciso construir um modelo gramatical com as construções possíveis e a probabilidade de ocorrência de cada tipo de construção (Guilhoto e Rosa, 2001 e Pereira, 2009).

Mesmo com uma boa base, o sistema pode apresentar erros devido a diferenças no jeito de falar de pessoas de diferentes regiões, pela falta de uma palavra na sua base de dados, por não conseguir diferenciar palavras homônimas, etc. Por isso um caminho que se segue no desenvolvimento de novas interfaces de reconhecimento de voz, é o treinamento do sistema, com várias pessoas diferentes falando com ele, para que ele possa aprender a entender melhor (Pereira, 2009).

Reconhecimento de voz, aliado a sintetizadores de voz e inteligência artificial, permitem que a automação de uma residência seja mais prática e interativa. É possível encontrar alguns dispositivos para compra que já possuem assistente de voz e são preparados para integração com sistemas automatizados, sendo possível, por comandos de voz, ligar luzes, abrir cortinas, tocar música e até pedir para o assistente contar piadas. As principais opções disponíveis no mercado possuem um custo mais elevado e entre elas estão o Amazon Echo, Apple Homepod e Google Home.

3. MATERIAIS

Para o desenvolvimento do trabalho, serão usados materiais para controle, reconhecimento de voz e simulação de uma residência automatizada.

Foi usada uma placa Arduino UNO R3 que contém o microcontrolador ATMEL ATmega328. Um cabo USB A/B foi usado como conexão serial entre o Arduino e o computador.

Foi essencial o uso de um computador para operar os softwares IDE Arduino e BitVoicer Server. A IDE Arduino, é um aplicativo gratuito e OpenSource, que torna possível a programação da placa Arduino, que fará o controle do trabalho. Para o reconhecimento e síntese de voz será usado o BitVoicer Server, desenvolvido pela empresa BitSophia. Os dois softwares são compatíveis e existem bibliotecas na IDE Arduino que possibilitam o uso do BitVoicer Server em conjunto com a plataforma.

Ainda para o reconhecimento de voz, foi usado o sensor de som fabricado pela empresa RoboCore que consiste na junção de um microfone de eletreto e um amplificador operacional de ganho 100 em uma mesma placa.

Foram utilizados ainda componentes que comumente se encontram em uma residência automatizada, para simular uma: 3 LEDs de 5mm, sendo dois vermelhos e um amarelo; 3 resistores de 330 Ω ; Micro Servo 9g SG90 TowerPro de tamanho 23x12,1x21,4mm, tensão de operação de 4,8V à 6V e torque de 1,5kg.cm (4,8V), 1,7kg.cm (6V); módulo relé da RoboCore de carga nominal 10A, que contém um relé miniatura da Metaltex modelo AX1RC-5V (carga nominal de 12A/152VAC – 7A/250VAC e tempo de acionamento de contatos de 10ms) e todo seu circuito de driver; lâmpada LED Ouralux 9W modelo 20031 e soquete de porcelana para lâmpada E27.

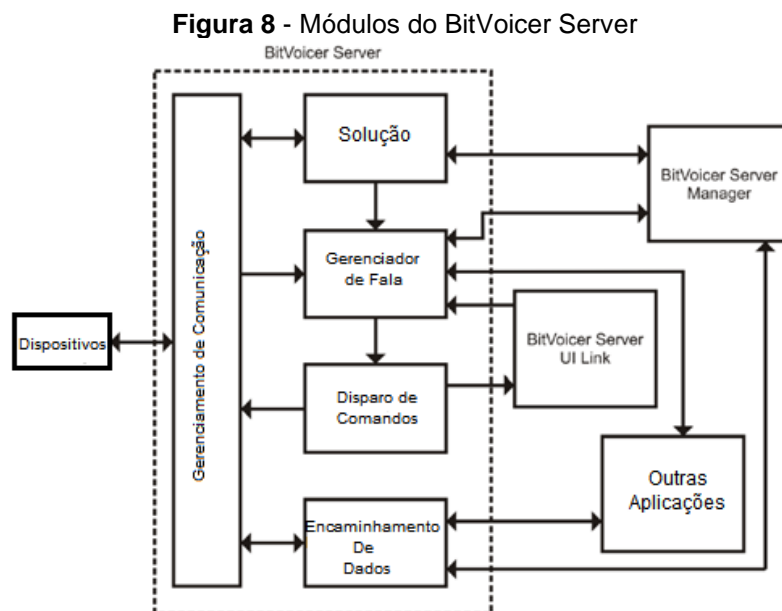
Também foi usada uma protoboard de 400 pontos espaçados pelo padrão de 0,1" e com dimensões de 83.5 x 54.5 x 8.5 mm.

Também foram utilizados diversos jumpers que consistem em pontas quadradas conectadas por pequenos fios coloridos de 10 ou 20 cm, dois metros de fio de cobre paralelo de 4mm, com isolamento de PVC branco e um plug de tomada da marca FAME de dois polos para correntes de até 10A.

3.1. BITVOICER SERVER

Para o desenvolvimento deste trabalho, foi usado o software pago BitVoicer Server, que é um servidor de reconhecimento e síntese de fala para automação por voz e que sua licença custa US\$8,90, sendo necessário comprar uma licença para cada dispositivo de entrada (dispositivos de saída são ilimitados). De maneira geral, microcontroladores não possuem poder de processamento e memória suficientes para realizar reconhecimento e síntese de fala avançados. O BitVoicer Server elimina os efeitos dessas limitações realizando o trabalho pesado. Dessa forma, o microcontrolador pode destinar a maioria de seus recursos à sua funcionalidade principal (BitSophia, 2015).

O BitVoicer Server, segundo a sua desenvolvedora BitSophia no manual do usuário, é composto de cinco módulos interdependentes, sendo eles: *Solution* (Solução), *Communication Manager* (Gerenciador de Comunicação), *Speech Manager* (Gerenciador de Fala), *Command Shooter* (Disparador de Comandos) e *Data Forwarding* (Encaminhamento de Dados). A interação entre os módulos e os softwares BitVoicer Server Manager e BitVoicer Server UI Link são apresentados na Figura 8.



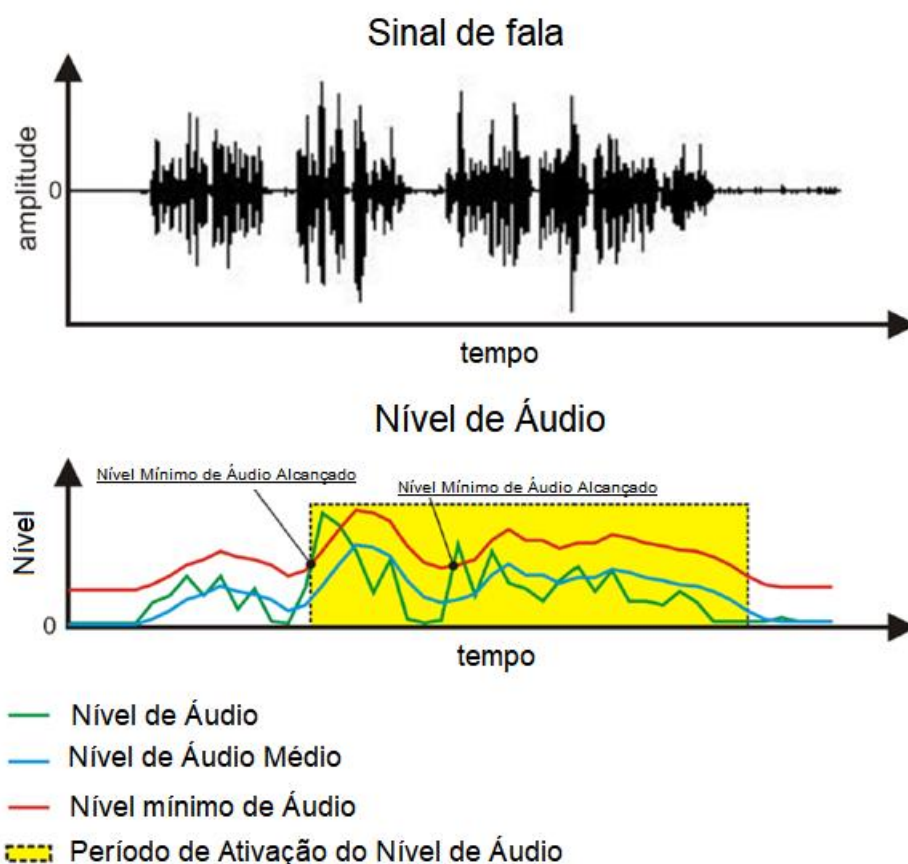
Fonte: Traduzido de BitSophia (2015)

A BitSophia define o módulo *Solution* como o responsável pelo armazenamento, gerenciamento, e validação de todas as operações realizadas em objetos de solução do BitVoicer Server. O módulo *Communication Manager* tem dentro suas

principais funções implementar o protocolo do BitVoicer Server, manter as conexões seriais ou TCP/IP e gerenciar o fluxo de dados. Já o módulo *Speech Manager* é o responsável pelo processamento dos fluxos de áudio para reconhecimento de fala e pela síntese de voz (BitSophia, 2015).

Para melhorar a precisão do reconhecimento de voz, o BitVoicer Server emprega uma série de filtros de validação definidos pelo desenvolvedor, onde somente o sinal que satisfaça o critério destes filtros será considerado válido. Um desses filtros é o Nível Mínimo de Áudio, que durante o processamento de fluxo de áudio, o volume captado é medido periodicamente. Tendo esta média, acrescida do valor definido no filtro, constituem o Nível Mínimo de Áudio. Ao atingir este nível, será iniciado o período ativado por nível de áudio. Esse filtro é exemplificado na Figura 9 ao mostrar no primeiro gráfico um sinal de fala e no segundo, os níveis de áudio alcançados por esta fala. O retângulo amarelo se refere ao período de ativação do nível de áudio, ou seja, só é ativado quando se obter determinado nível mínimo. (BitSophia, 2015).

Figura 9 - Nível Mínimo de Áudio



Fonte: Traduzido de BitSophia (2015)

Command Shooter é o módulo responsável pela execução de comandos que serão definidos ao se criar um esquema de voz, ou seja, quando haver correspondência em uma frase, o comando determinado para ela será executado por este módulo. E, por fim, *Data Forwarding* é o módulo que permite a troca de dados entre o BitVoicer Server e aplicações externas (BitSophia, 2015).

Pelo fato do BitVoicer Server ser compatível com Arduino e possuir uma biblioteca no seu pacote de instalação pronta para ser adicionada a IDE Arduino, a escolha deste software mostra-se uma opção viável para o desenvolvimento do trabalho.

Outro motivo da escolha do BitVoicer Server em relação a outros dispositivos, como o Google Home, é a possibilidade de maior personalização quanto aos comandos de áudio recebidos e as respostas sintetizada, o preço menor e a facilidade de aquisição, pois ainda não é possível adquirir um Google Home de maneira oficial no Brasil.

Apenas a título de comparação, para entender um pouco como um dispositivo comercial funciona, é apresentado o Google Home na subseção 3.1.1.

3.1.1. FUNCIONAMENTO DO GOOGLE HOME

Os assistentes de voz podem ser encontrados em smartphones, tablets, computadores e dispositivos próprios, além de outras opções de dispositivos que se conectam a internet.

O Google Home é uma das opções existente hoje em dia de dispositivos com assistente de voz, que podem ser usados para a implementação de um sistema de automação residencial controlada por voz. O Google Home, apresentado na Figura 10, é um alto-falante com acesso à internet, que funciona com o sistema de reconhecimento de voz do Google Assistant, que ao ser ativado pela palavra-chave “Ok, Google”, reconhece comandos e perguntas. Assim é possível obter informações, clima ou qual foi o resultado de uma partida de futebol, por exemplo, programar alarmes, marcar datas no calendário e controlar dispositivos de uma casa automatizada como lâmpadas, tomadas, geladeiras e câmeras (Svetlik, 2019) (CNET Home).

Figura 10 - Google Home

Fonte: Cool Blue, 2019

A escolha do Google Home para esta análise ao invés de modelos de outras empresas disponíveis no mercado, se deu pela maior familiaridade com o Google Assistant, que pode ser facilmente encontrado em outros dispositivos, como smartphones (principalmente os que utilizam sistema operacional Android). Porém, ele possui menos opções de línguas suportadas, que até o momento são:

Tabela 1 – Idiomas disponíveis do Google Home

País	Língua Suportada
Austrália	Inglês
Áustria	Alemão
Canadá	Inglês / Francês CA
Dinamarca	Dinamarquês
França	Francês
Alemanha	Alemão
Índia	Inglês (Índia) / Inglês (US) / Hindi
Irlanda	Inglês (UK)
Itália	Italiano
Japão	Japonês
Coreia do Sul	Coreano
México	Espanhol

Holanda	Holandês
Noruega	Norueguês
Cingapura	Inglês (Cingapura) / Inglês (US)
Espanha	Espanhol
Suécia	Sueco
Reino Unido	Inglês
Estados Unidos	Inglês

Fonte: Google

O Google Home ainda não possui suporte e nem é comercializado no Brasil. O preço deste dispositivo nos Estados Unidos é de aproximadamente US\$99,90.

Um assistente digital pessoal pode ser implementado em um hardware ou ser uma combinação de hardware e software. Sendo que em um dispositivo, comumente, encontram-se unidade de processamento central (CPU), interfaces (áudio, por exemplo) e barramentos de conexão, com a possibilidade de comunicação com outros dispositivos através de uma rede de comunicações (Gruber, 2010).

O hardware de um Google Home é muito semelhante ao encontrado em um Google Chromecast, segundo apontamento por Scott Harvard, que desmontou um e encontrou os seguintes componentes eletrônicos: Atmel ATSAMD21 32-bit ARM Cortex-M0+microcontroller; NXP PCA9956BTW LED drivers; InvenSense INMP621 MEMS microfones; Peerless PLS-50N25AL07-04; Marvell 88DE3006 Armada 1500 Mini Plus dual-core ARM Cortex-A7 media processor; Toshiba TC58NVG1S3HBA16 256 MB NAND flash; Marvell Avastar 88W8887 WLAN/BT/NFC SoC; amplificador de áudio Texas Instruments TAS5720; Samsung K4B4G16 512 MB B-Die DDR3 SDRAM (Harvard, 2016). A Figura 11 mostra um Google Home desmontado.

Figura 11 - Google Home desmontado



Fonte: iFixit

Li (2017) explica que o Google Home, teve diversos avanços no reconhecimento de voz, até seu lançamento, em 2016, usando técnicas de uma *front-end* de *dereverberation* (termo em inglês, referente ao processo inverso de reverberação), que é o processo que remove reverberações de um som captado, de modelos de redes neurais que executam processamento multicanal em conjunto com uma modelagem acústica e de Grid-LSTMs para modelar variações de frequência, diminuindo a taxa de erro de palavras.

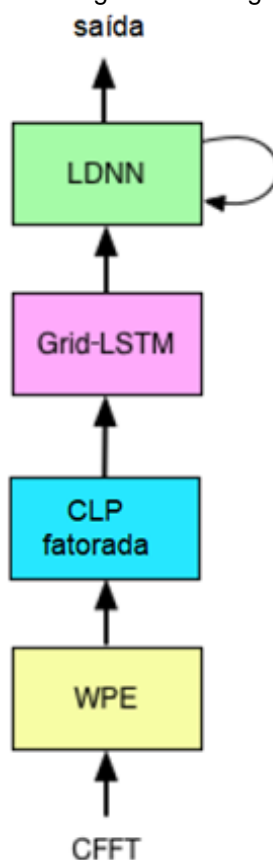
Grid-LSTM é uma arquitetura artificial de rede neural recorrente, usada no processo de *deep learning*, que possui conexões de feedback, sendo um tipo bidimensional que modela através do tempo e da frequência (Li, 2017).

O Google Home utiliza para a análise de fala um modelo chamado Projeção Linear Complexa Fatorada (*factored Complex Linear Projection – fCLP*) tendo a transformada rápida complexa de Fourier (CFFT) como entrada e agindo como um filtro de soma. O processo do *front-end* de *dereverberation* propicia ao sistema uma maior robustez, por um algoritmo adaptativo de quadrados mínimos recursivos

mútuos (RLS), baseado em um algoritmo de previsão de erro ponderada (WPE) (Li, 2017).

Um diagrama de blocos, apresentado na Figura 12, mostra o sistema utilizado pelo Google Home. O sinal de voz é captado e convertido pela CFFT e é aplicado o *dereverberation* pela WPE adaptável. Então, será obtida uma representação de frequência e tempo, que terá sua modelação feita pela Grid-LSTM e assim, o resultado será enviado para um modelo acústico padrão para o reconhecimento.

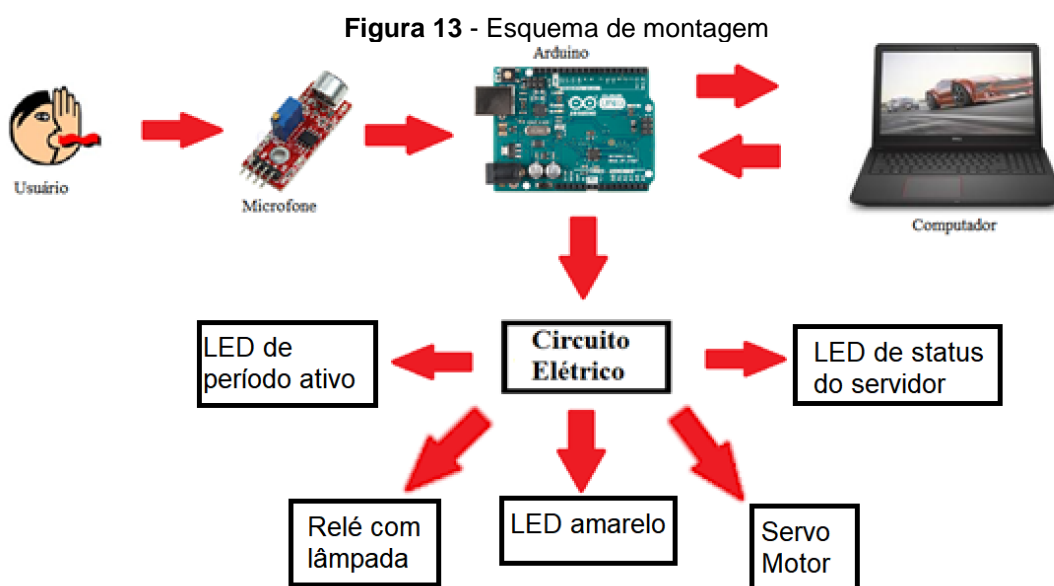
Figura 12 - Visão geral do Google Home



Fonte: Traduzido de Li, 2017

4. DESENVOLVIMENTO

O sistema desenvolvido é exemplificado na Figura 13. O usuário irá enviar comandos de voz, que serão captados por um microfone conectado ao Arduino, que por sua vez, irá processar e enviar a informação ao computador, responsável por traduzir os dados recebidos e comparar com aqueles presentes na biblioteca do software de reconhecimento de voz, BitVoicer Server. Caso haja correspondência, será enviado ao Arduino um dado específico, que será reconhecido por ele e atuará em algum dos componentes elétricos do circuito. Serão implementados: um LED vermelho de status do servidor, um LED vermelho que indicará o período de reconhecimento de fala, um LED amarelo para representar a iluminação de um cômodo da residência, por exemplo, um servo motor agindo como uma cortina automatizada e um relé que acionará uma lâmpada.



Fonte: Montagem do Autor com imagens de BitSophia, Arduino e Dell

Antes de iniciar a construção do sistema, foram reproduzidos os exemplos disponíveis no manual do BitVoicer Server, para melhor entendimento do sistema, como está descrito na seção 4.1. A seção 4.2, mostra como foi feita a configuração do software e os processos de montagem e programação do modelo desenvolvido. Microfone e LEDs de status foram os primeiros componentes a serem implementados. Antes da adição de cada novo elemento, este foi testado

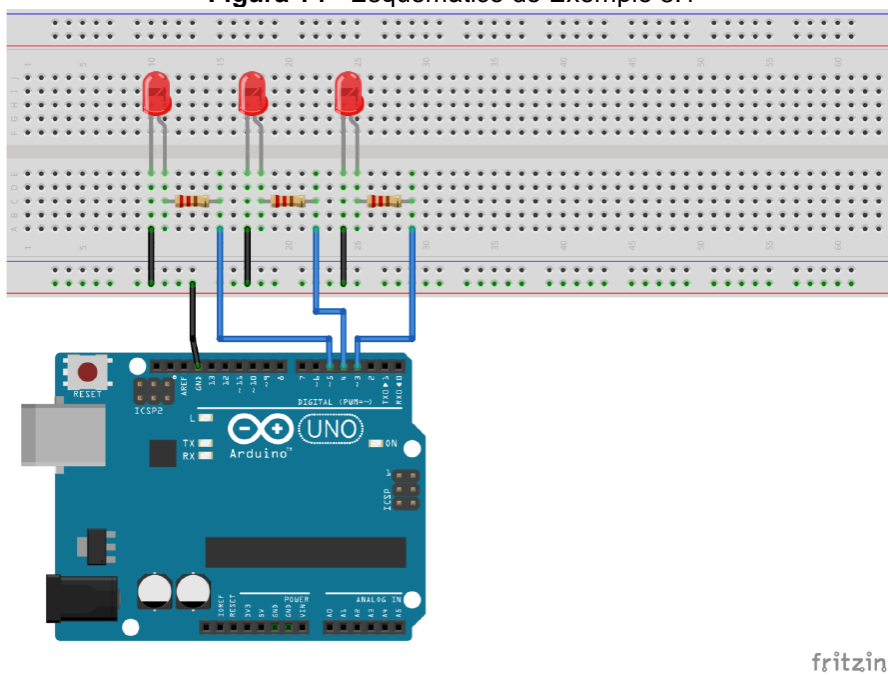
isoladamente, para só depois da confirmação de funcionamento e do entendimento do funcionamento, ele pudesse ser acrescentado ao projeto completo.

4.1. DOS EXEMPLOS DO BITVOICER SERVER

Para entender o funcionamento do BitVoicer Server, seu manual oferece diversos exemplos práticos de implementação de algumas funcionalidades do BitVoicer Server. Portanto, testar esses exemplos foi o ponto de partida deste trabalho.

O primeiro exemplo proposto pelo manual (Exemplo 8.1), propõe a obtenção do estado do servidor. Verificar se ele está operando, se há motor de reconhecimento de fala (Speech Recognition Engine – SRE) e se o módulo Data Forwarding está operante. Como não foi atribuído nenhum SRE ainda, apenas dois dos três LEDs usados irão acender. O esquemático é apresentado na Figura 14 e o código usado é obtido diretamente na IDE do Arduino, abrindo o sketch BVSSStatus (Arquivos > Exemplos > BVSP Examples > BVSPStatus). Com o código compilado, o Arduino já montado, sendo ligado a PC através da entrada USB. O código foi carregado sem nenhum erro. (BitSophia, 2015)

Figura 14 - Esquemático do Exemplo 8.1



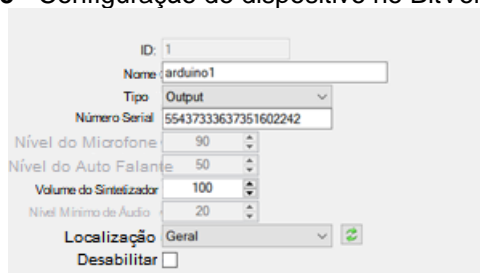
Fonte: BitSophia, 2015

O BitVoicer Server foi então aberto e dentro dele criado um novo Local e um novo dispositivo. Ao configurar o dispositivo, é necessário o Número Serial, para

que o programa reconheça o Arduino e consiga se comunicar com ele. É possível obter este número através da IDE do Arduino (Ferramentas > Obter Informação de Placa).

A Figura 15 mostra uma captura de tela do programa na etapa de configuração do dispositivo, foi escolhido “arduino1” como nome do dispositivo, escolhido o tipo “output”, ou seja, saída, colocado o número de série da placa e escolhido o local “Geral”. Os outros parâmetros foram mantidos os padrões do programa.

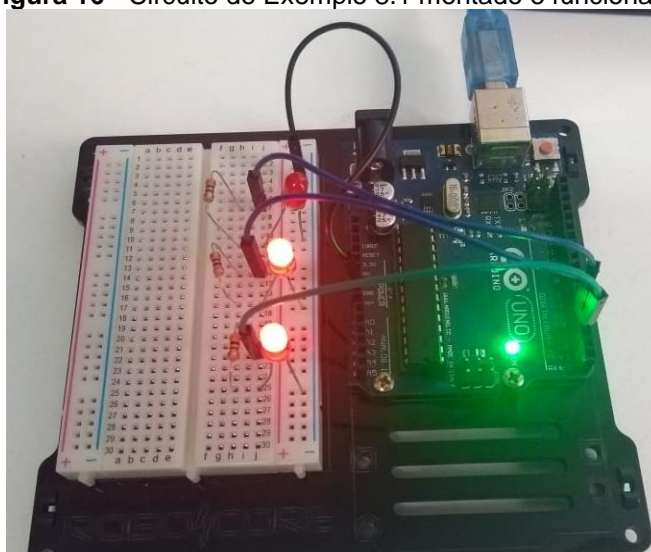
Figura 15 - Configuração do dispositivo no BitVoicer Server



Fonte: O autor

O resultado apresentado foi o esperado: o circuito funcionou corretamente, dois LEDs acesos (ligados a porta 3 e 4), indicando que o servidor e o módulo de Data Forwarding estão operantes e um apagado (ligado a porta 5), demonstrando que não há SRE, como mostrado na Figura 16. Os LEDs RX e TX da própria placa, piscando a cada 2 segundos, apontam que o Arduino está fazendo uma requisição de estado ao servidor e está recebendo uma resposta (BitSophia, 2015).

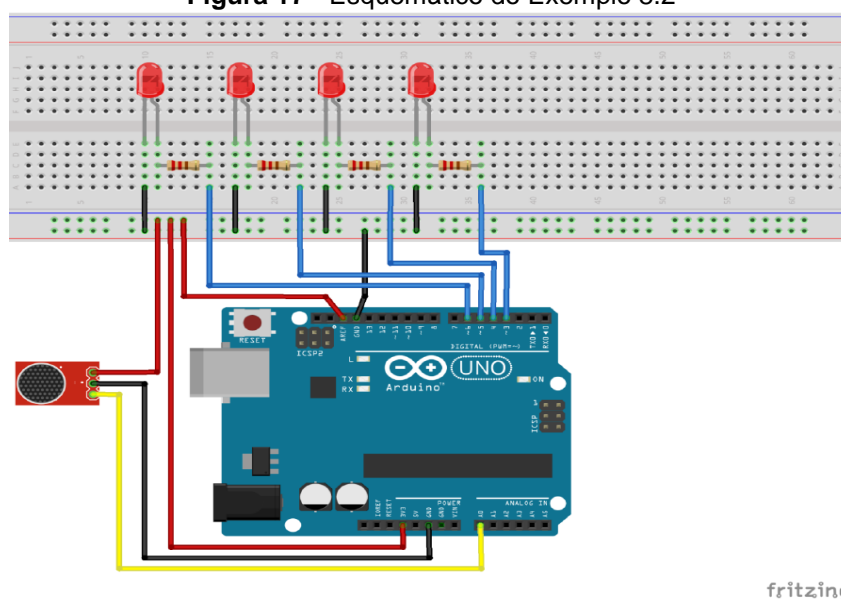
Figura 16 - Circuito do Exemplo 8.1 montado e funcionando



Fonte: O autor

No Exemplo 8.2 (Reconhecimento de fala), além da obtenção do estado do servidor, obtidos no exemplo anterior, foi implementado o reconhecimento de fala, usando um microfone conectado ao Arduino por uma entrada analógica. A montagem do circuito foi feita seguindo a Figura 17 e o código usado foi obtido diretamente na IDE Arduino, abrindo o sketch SpeechRecognition (Arquivo > Exemplos > BVSMic > Examples > SpeechRecognition) e foi carregado o código na placa.

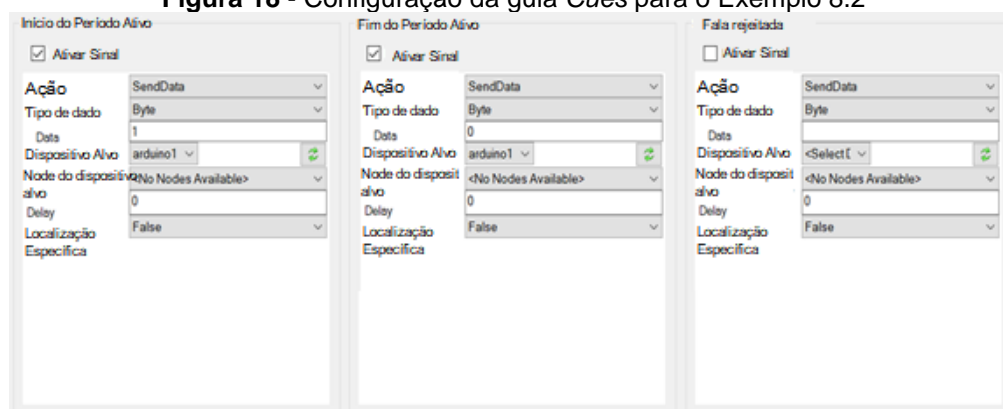
Figura 17 - Esquemático do Exemplo 8.2



Fonte: BitSophia 2015

O BitVoicer Server, manteve a configuração de localização e de dispositivo, alterando apenas o campo Type, selecionando a opção Mixed, na guia General. A guia Cues foi configurada segundo a Figura 18.

Figura 18 - Configuração da guia Cues para o Exemplo 8.2



Fonte: O autor

Após isso foi criado um esquema de voz, em que a palavra de ativação escolhida foi “Arduino”. Na guia *Input Devices*, do esquema de voz, o dispositivo criado foi selecionado e na guia *Sentences* o texto “pisca todos LEDs” adicionado à sentença. E na guia *Anagrams/Commands* o seguinte comando foi adicionado ao anagrama (BitSophia, 2015), como mostrado na Figura 19:

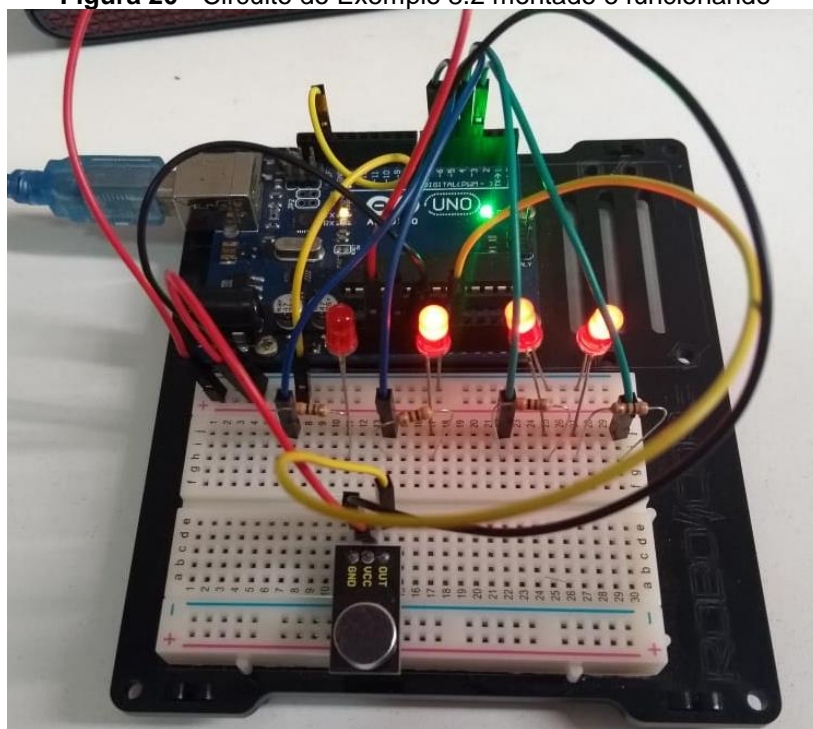
Figura 19 - Comando adicionado ao anagrama no Exemplo 8.2

Propriedades de Comando	
Ação	SendData
Tipo de dado	String
Dado	blink
Dispositivo Alvo	arduino1
Node do Dispositivo Alvo	<No Nodes Available>
Delay	3890
Localização Específica	False

Fonte: O autor

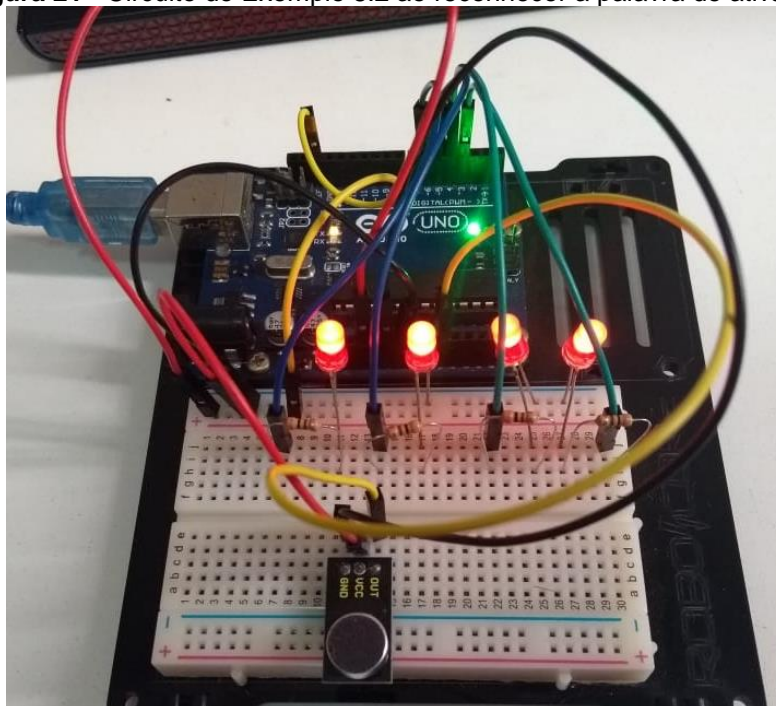
O circuito funcionou como esperado, conforme exibido na Figura 20 e na Figura 21. Os LEDs conectados as entradas 3, 4 e 5 permaneceram acesos e conectado ao 6, apagado até a palavra de ativação (Arduino) ser dita. Assim que todos os LEDs estavam acessos, ao dizer “pisca todos LEDs”, o comando foi executado e todos LEDs piscaram.

Figura 20 - Circuito do Exemplo 8.2 montado e funcionando



Fonte: O autor

Figura 21 - Circuito do Exemplo 8.2 ao reconhecer a palavra de ativação



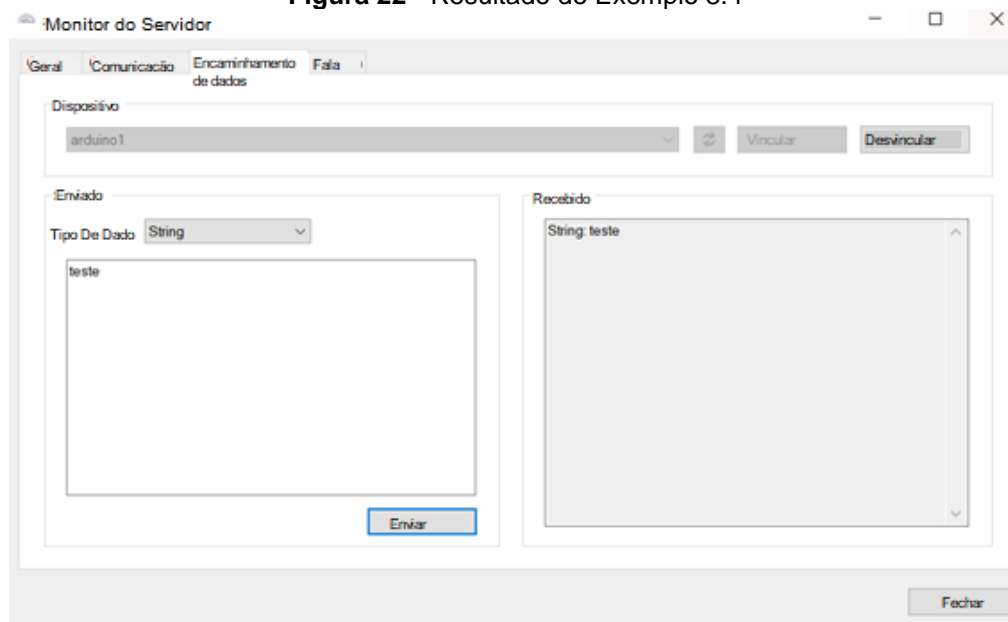
Fonte: O autor

Não foi possível a realização do Exemplo 8.3 (Reconhecimento de fala e síntese de voz) pois requeria o uso do Arduino DUE (que contém o microcontrolador ATMEGA SAM3X8E ARM Cortex-M3) e outros componentes indisponíveis.

O exemplo 8.4 (Encaminhamento de Dados) não necessitou de nenhum circuito a ser montado. Apenas conectar o Arduino UNO ao computador através do cabo serial e carregar o código de exemplo DataForwarding (Arquivo > Exemplos > BVSP > Examples > DataForwarding). A configuração do BitVoicer de localização e dispositivo foram as mesmas usadas no exemplo 8.1.

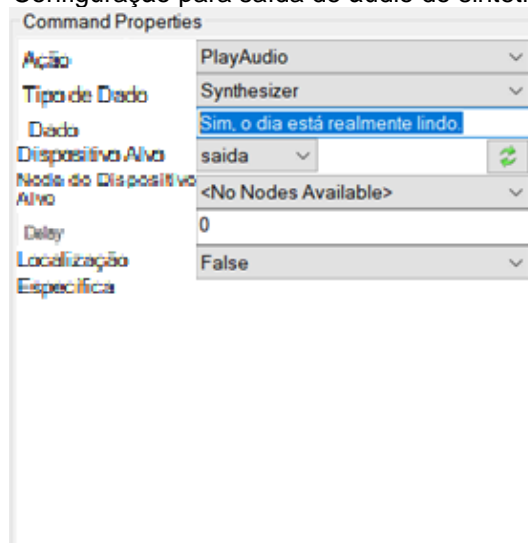
Após isso, foi aberta a ferramenta Monitor do Servidor e escolhida a aba Data Forwarding. O dispositivo criado foi escolhido e selecionado o botão Bind. Selecionado o tipo de dado *string* a ser enviado, foi digitado o texto “teste” na parte esquerda e recebido o mesmo dado na parte direita, tendo o resultado mostrado na Figura 22. Quando o botão *Send* foi acionado, foi possível perceber os LEDs RX e TX do Arduino piscando rapidamente (BitSophia, 2015).

Figura 22 - Resultado do Exemplo 8.4



Fonte: O autor

O Exemplo 8.5 (Usando o adaptador de áudio do servidor) propôs usar o adaptador de áudio do servidor, porém, como isso não seria usado neste trabalho, já que o microfone utilizado seria o conectado ao Arduino UNO, foi aproveitado todo o esquema do exemplo 8.2 para a entrada, configurado o alto-falante do computador como saída e adicionada algumas sentenças e comandos, para simular uma conversa simples com o sintetizador, que ao reconhecer uma frase específica, daria uma resposta específica. A sentença adicionada foi “Hoje é um dia lindo”, condizente com a Figura 23 e a resposta foi “Sim, o dia está realmente lindo.”

Figura 23 - Configuração para saída de áudio do sintetizador de voz

Fonte: O autor

Após este exemplo, foi notado que além do BitVoicer Server Manager estar aberto, também é necessário que o BitVoicer Server UI Link também esteja, para que tudo funcione corretamente.

4.2. DA REALIZAÇÃO DA DOMÓTICA

O primeiro passo para construção do projeto de um sistema automatizado, comandado por voz, foi a configuração do BitVoicer Server. Executando o BitVoicer Server no computador, na interface do software foi criada uma nova localização ao clicar no ícone “*New Location*” e nomeando essa localização como “Geral”. Em seguida, foi criado um dispositivo novo ao clicar em “*New Device*”. O dispositivo criado foi configurado na aba “*General*” nomeando para “arduino1”, selecionando o tipo “*Mixed*” (já que o Arduino irá se comportar como entrada e como saída), inserindo o número serial da placa (obtido através da IDE do Arduino) e selecionando o local “Geral”. As outras opções não foram alteradas, sendo utilizadas as opções padrões do programa. A Figura 24 indica como foi feita a configuração.

Figura 24 - Configuração do dispositivo

ID: 1
 Nome: arduino1
 Tipo: Mixed
 Número: 55437333637351602242
 Nível do Microfone: 90
 Nível do Auto Falante: 50
 Volume do Sintetizador: 100
 Nível Audio Mínimo: 20
 Localização: Geral
 Desativar: ☐

Fonte: O autor

Na aba “*Communication*” reproduzida na Figura 25, foi escolhida a interface serial e colocado o nome da porta onde o Arduino está conectado: “COM3” e o restante, mantido nas configurações padrões.

Figura 25 - Configuração de comunicação

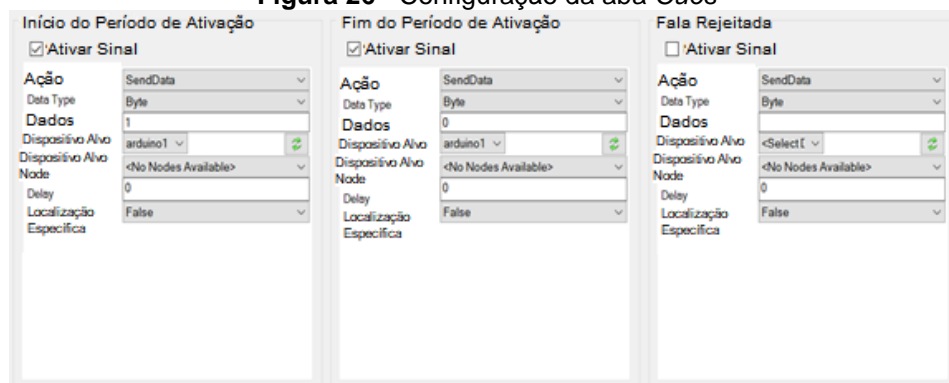
Interface
☒ Serial ☐ TCP/IP
 Nome da Porta: COM3
 Taxa: 115200 (bps)
 Data bits: 8
 Paridade: None
 Stop bits: 1
 Controle de fluxo: None
 RTS: ☐
 DTR: ☐
 Não travar a porta: ☐
 IP address: 0 . 0 . 0 . 0
 Inactivity timeout: 10 (seconds)
Saída
 Taxa de Dados: 14400 (bytes per second)
 MSSS: 64 (bytes)
 Protocolo: BitVoicer Server
Buffers
 Tamanho Buffer entrada: 16384 (bytes)
 Tamanho Buffer saída: 2048 (bytes)

Fonte: O autor

Na aba “*Cues*” o dispositivo foi configurado de maneira mostrada na Figura 26. Para que ao início do Período de Ativação, em que há reconhecimento de voz, seja

mandado um bit 1 ao Arduino, para que seja aceso um LED e ao fim do período, um bit 0 seja enviado para apagar o LED.

Figura 26 - Configuração da aba Cues



Fonte: O autor

Então foi criado um esquema de voz, ao clicar em “*New Voice Schema*”, que na aba “*General*” foi selecionada a palavra “Arduino” como palavra de ativação e os outros parâmetros foram mantidos como padrão. Já na aba “*Input Devices*” o dispositivo *arduino1* foi selecionado como dispositivo de entrada. As abas “*Sentences*” e “*Anagrams/Commands*” foram incrementadas conforme a implementação de novos elementos no circuito.

Após a configuração inicial no BitVoicer Server Manager, antes de cada implementação de um novo componente ao circuito, os componentes foram testados separadamente e depois foram sendo integrados ao todo. Isso foi importante para saber como cada parte seria implementada.

Na IDE do Arduino foi necessário adicionar as bibliotecas do BitVoicer *BVSP.h* e *BVSMic.h* para que as funções usadas sejam reconhecidas. Também foi preciso definir os pinos em que os LEDs e o módulo do microfone seriam conectados, para serem ligados de maneira correta, com cada pino tendo seu componente específico; definir, criar os *buffers* de *string* e áudio; inicializar constantes globais das classes referentes às bibliotecas; iniciar a comunicação serial em uma taxa definida de bits por segundo; definir o tipo de conexão dos pinos do Arduino (entrada ou saída) e criação de algumas funções auxiliares. A seguir, o código utilizado para tal configuração, comentários sobre o funcionamento:

```
//Incluir bibliotecas próprias do BitVoicer
#include <BVSP.h>
#include <BVSMic.h>

// Definição dos pinos que serão usados pelo Arduino
#define statusLED 5
```

```

#define periodoat 6
#define entaudio 0

//Define as constantes que serão enviadas como parâmetro para a função
bvsp.begin
const unsigned long STATUS_REQUEST_INTERVAL = 2000;
const unsigned long STATUS_REQUEST_TIMEOUT = 1000;

//Inicializar uma nova instância global da classe BVSP
BVSP bvsp = BVSP();

//Inicializar uma instância global da classe BVSMIC
BVSMic bvsm = BVSMic();

//Cria um buffer que será usado para ler amostras gravadas do BVSMIC,
com um buffer de audio do tamanho de 64
byte audioBuffer[64];

//Cria um buffer de string para receber strings do BitVoicerServer,
sendo que o seu tamanho é 64
char stringBuffer[64];

void setup()
{
    //Inicia a comunicação serial em 115200 bps
    Serial.begin(115200);

    // Define os modos de pinagem do arduino
    pinMode(statusLED, OUTPUT);
    pinMode(periodoat, OUTPUT);

    // Define a porta serial do arduino que será usada na comunicação,
    quanto tempo levará até que uma solicitação de status expire
    //e com que frequência as solicitações de status devem ser enviadas
    ao BitVoicer Server

    bvsp.begin(Serial, STATUS_REQUEST_TIMEOUT,
        STATUS_REQUEST_INTERVAL);

    // Define a função frameReceived
    bvsp.frameReceived = BVSP_frameReceived;

    // Prepara o time da classe do BVSMic
    bvsm.begin();
}

void loop()
{
    // Se o intervalo de solicitação do status expirar, envia nova
    solicitação de status pro BitVoicer Server
    bvsp.keepAlive();

    // Verifica se há dados disponíveis na porta serial

```

```

    bvsp.receive();

    // Obtém os status do servidor, verifica se há SRE disponível e
    // verifica se há encaminhamento de dados
    if (bvsp.isSREAvailable() && bvsp.isBVSRRunning() &&
        bvsp.isDataFwdRunning() )
    {
        // Liga o LED de status
        digitalWrite(statusLED, HIGH);

        // Se o microfone não estiver ativo, ativa e começa a gravar
        if (!bvsm.isRecording)
        {
            bvsm.setAudioInput(entaudio, EXTERNAL);
            bvsm.startRecording();
        }

        // Verifica se tem amostras disponíveis
        if (bvsm.available)
        {
            // Garante que o modo de entrada seja STREAM_MODE

            if (bvsp.inboundMode == FRAMED_MODE)
                bvsp.setInboundMode(STREAM_MODE);

            // Lê as amostras de audio do microfone
            int bytesRead = bvsm.read(audioBuffer, 64);

            // Envia o audio para o BitVoicer Server
            bvsp.sendStream(audioBuffer, bytesRead);
        }
    }
    else
    {
        // Se não tem SRE disponível, desliga o LED de status
        digitalWrite(periodoat, LOW);

        // Se o microfone está gravando, para
        if (bvsm.isRecording)
            bvsm.stopRecording();
    }
}

void BVSP_frameReceived(byte dataType, int payloadSize)
{
    // Executa a ação apropriada dependendo do tipo de dado enviado pelo
    // servidor
    switch (dataType)
    {
        case DATA_TYPE_BYTE:
            // Liga ou desliga o LED de periodo ativo dependendo do valor
            // recebido
            digitalWrite(periodoat, bvsp.getReceivedByte());
            break;
    }
}

```

```

    case DATA_TYPE_STRING:
        // Executa ações adicionais somente se os caracteres forem
        // reconhecidos

        if (bvsp.getReceivedString(stringBuffer, 64) != 0);
        {
            // Caso seja recebido o string "pisca",
            // as ações necessárias para os LEDs piscar serão ativadas if
            (strcmp(stringBuffer, "pisca") == 0)
            {
                desligaLED();
                delay(100);
                ligaLED();
                delay(100);
                desligaLED();
                delay(100);
                ligaLED();
                delay(100);
                desligaLED();
                delay(100);
                ligaLED();
            }

        }

        break;
    }
}

// Liga todos os LEDs de status
void ligaLED()
{
    digitalWrite(statusLED, HIGH);
    digitalWrite(periodoat, HIGH);
}

// Desliga todos os LEDs de status
void desligaLED()
{
    digitalWrite(statusLED, LOW);
    digitalWrite(periodoat, LOW);
}

```

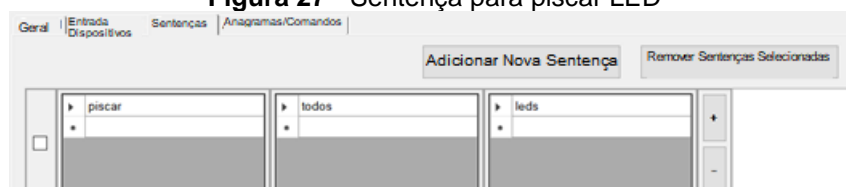
No BitVoicer Server, foi adicionada uma sentença (Figura 27) que será o modelo a se comparar com o áudio recebido do microfone e qual comando será atribuído caso essa comparação resulte em resultado positivo. Caso o a frase dita seja “Piscar todos LEDs”, o BitVoicer Server retornará uma string “pisca” (comando mostrado na Figura 28), que ativará o código que fará os dois LEDs conectados ao Arduino piscarem e na enviará a frase sintetizada “Sim mestre” para a saída de áudio do computador.

O módulo de microfone foi conectado à protoboard e os pinos VCC e GND conectados ao 5V e GND do Arduino, respectivamente. O pino OUT, foi ligado a porta A0.

Os dois LEDs foram conectados da mesma maneira: com o catodo conectado ao GND e o anodo a um resistor de 330Ω , que foi então conectado a respectiva porta do Arduino. O LED que define o status do servidor foi ligado à porta 5 e o que define se está em o reconhecimento de voz está em período ativo ou não, à porta 6.

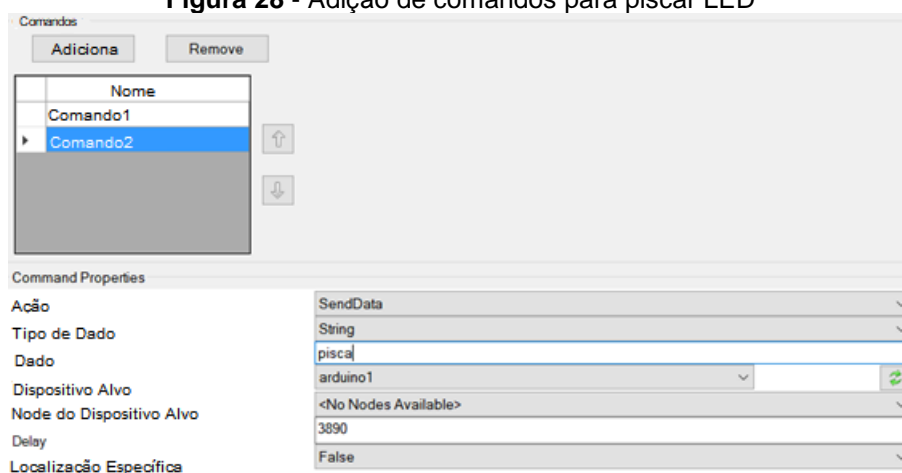
A Figura 29 é um esquemático da montagem que acabara de ser descrita.

Figura 27 - Sentença para piscar LED



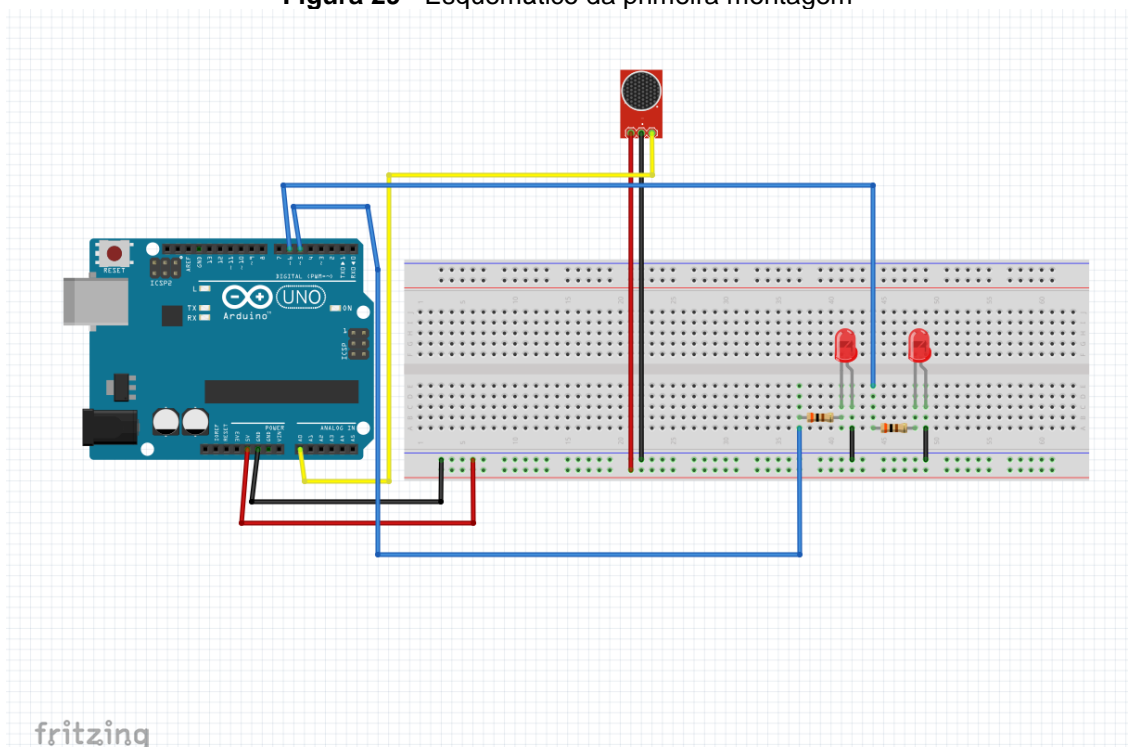
Fonte: O autor

Figura 28 - Adição de comandos para piscar LED



Fonte: O autor

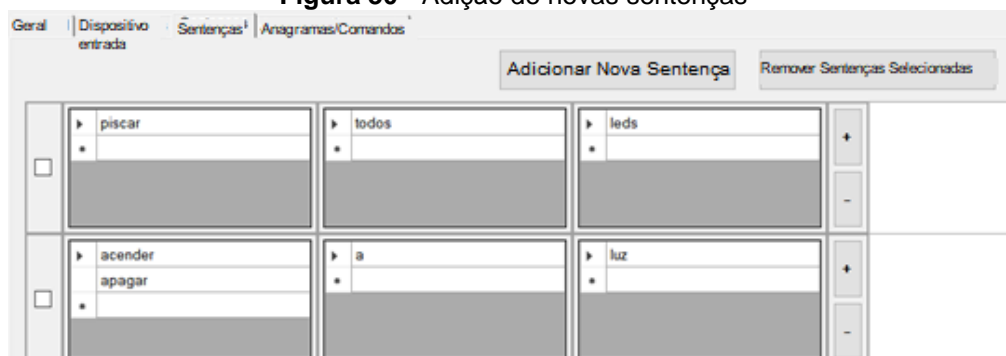
Figura 29 - Esquemático da primeira montagem



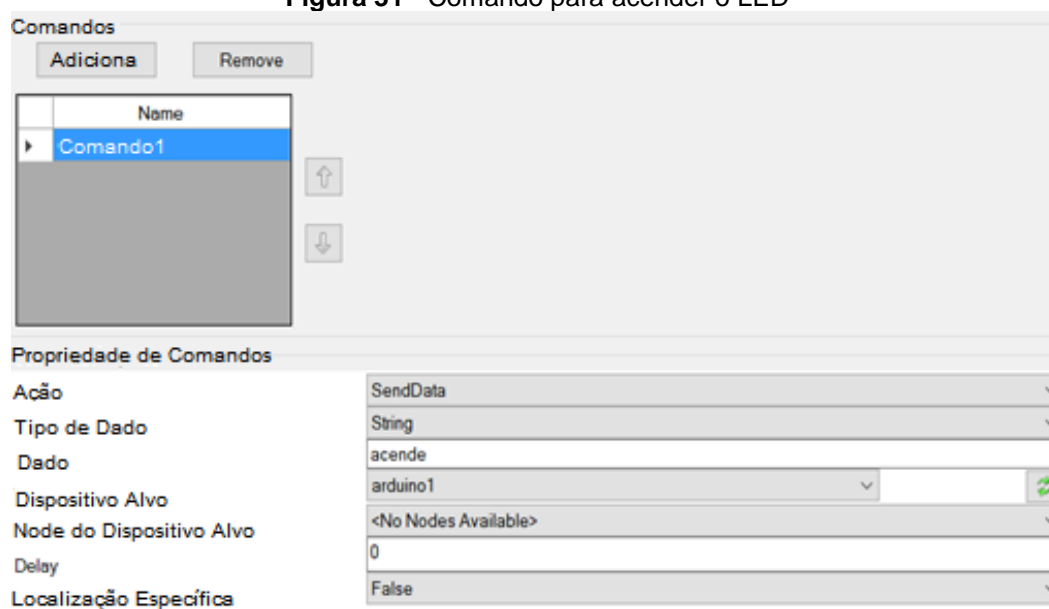
Fonte: O autor

A seguir, foi adicionado um LED amarelo representando a iluminação de um cômodo, por exemplo. Ele foi conectado da mesma maneira que os outros LEDs, ao GND e ao resistor de 330Ω e este, conectado ao pino 4 do Arduino, retratado na Figura 33. Mais uma sentença foi adicionada ao esquema de voz do BitVoicer Server, agora com duas possibilidades: acender ou apagar a luz, como pode ser visto na Figura 30. Um comando foi definido para cada anagrama como as Figuras 31 e 32 indicam: caso seja reconhecida a frase “Acender a luz”, será enviado um *string* “acende”, que será reconhecido e mandará o comando para acender o LED, e caso seja reconhecida a frase “Apagar a luz”, será enviado o *string* “apaga” que mandará o comando para apagar o LED. Para isso será necessário adicionar um código no condicional que executa as ações apenas se os caracteres forem reconhecidos: “if (bvsp.getReceivedString(stringBuffer, 64) != 0);”

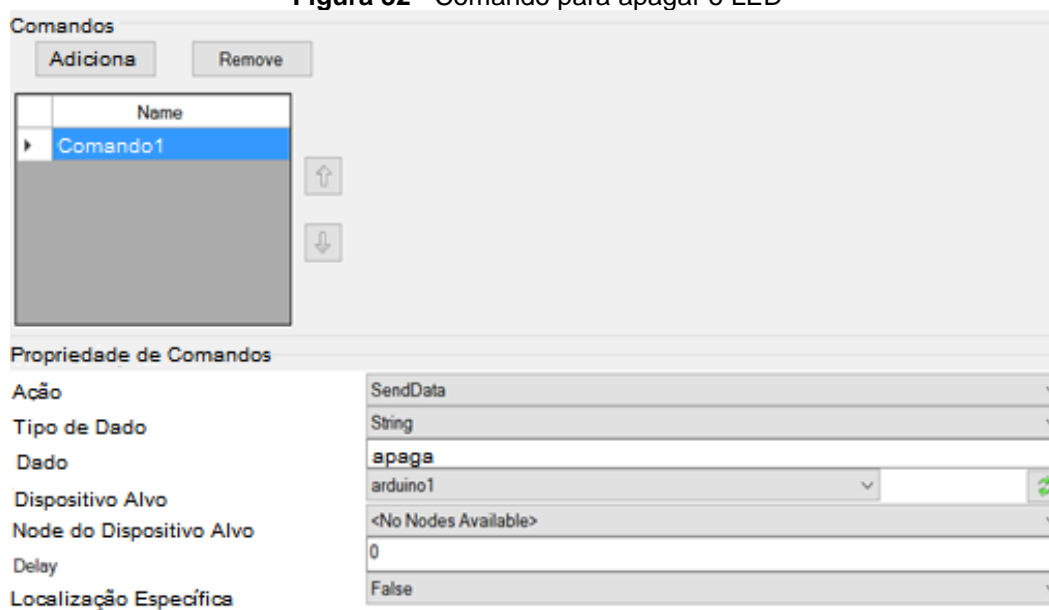
```
if (strcmp(stringBuffer, "acende") == 0)
{ digitalWrite(lampadal, HIGH);
}
if (strcmp(stringBuffer, "apaga") == 0)
{ digitalWrite(lampadal, LOW);
}
```

Figura 30 - Adição de novas sentenças

Fonte: O autor

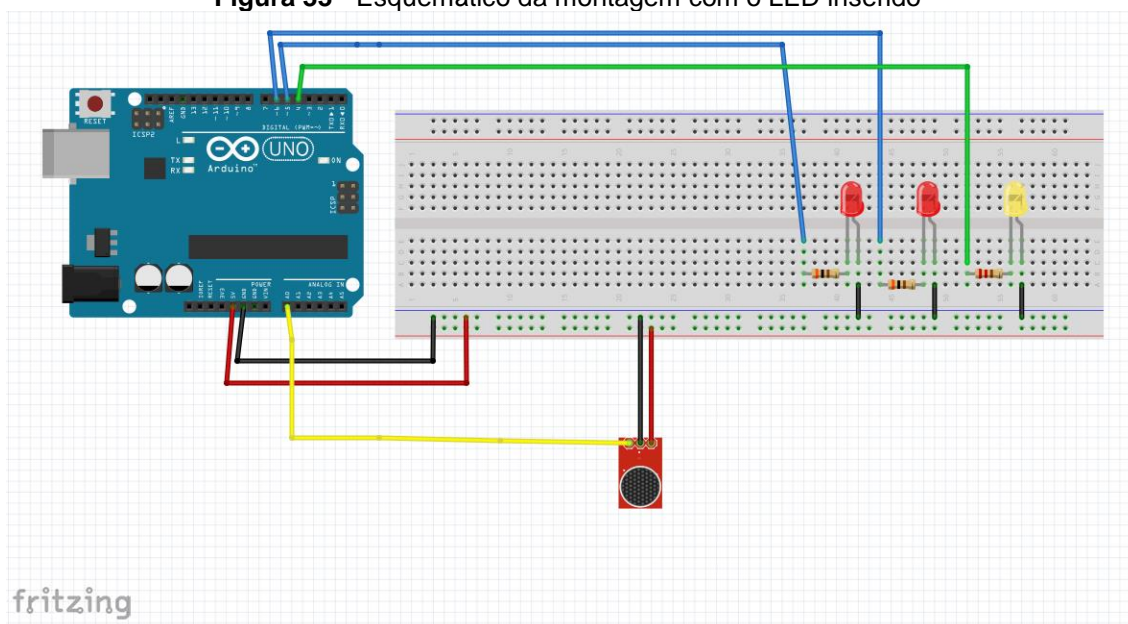
Figura 31 - Comando para acender o LED

Fonte: O autor

Figura 32 - Comando para apagar o LED

Fonte: O autor

Figura 33 - Esquemático da montagem com o LED inserido



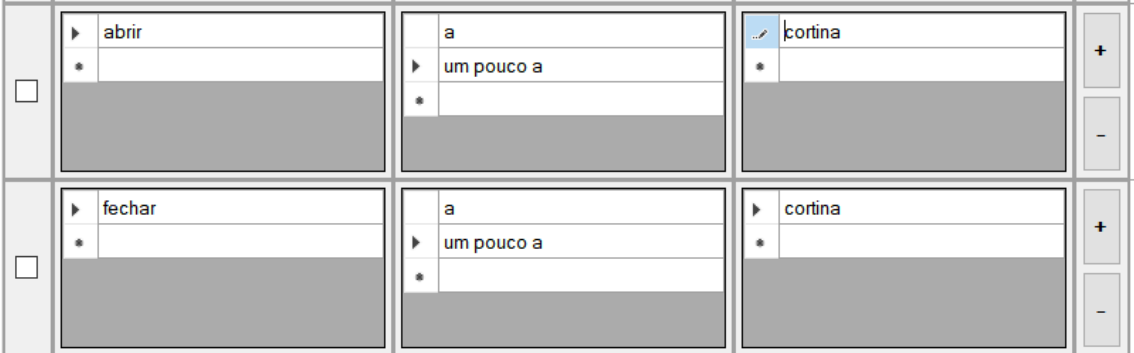
Fonte: O autor

Um servo motor foi adicionado ao trabalho para simular uma cortina com abrimento controlado. O servo motor utilizado possui a liberdade de 180° e foi definido, como exemplo, a cortina fechada com o servo ajustado para 20°, um pouco aberta ou fechada para 80° e um fechada para 150°. Ele foi conectado ao GND, ao 5V e ao pino 9 do Arduino, de maneira que pode ser observada no esquemático da Figura 36. É possível observar na Figura 34 que foram criadas sentenças dentro do esquema de voz do BitVoicer Server para o controle da cortina, sendo possível solicitar para “Abrir a cortina”, “Fechar a cortina”, “Abrir um pouco a cortina” ou “Fechar um pouco a cortina”. Cada anagrama, como mostrado na Figura 35, recebeu um comando específico para enviar um *string* ao Arduino e fazer com que o servo motor vá para a posição desejada. No código, foi necessário adicionar a biblioteca Servo.h (`#include <Servo.h>`) e inserir as seguintes linhas dentro da condição de caracteres reconhecidos:

```
if (strcmp(stringBuffer, "fecha") == 0)
{
    servomotor.write(20); //Move o servo pra posição 20 graus
    delay(15);
}
if (strcmp(stringBuffer, "pouco") == 0)
{
    servomotor.write(80); //Move o servo para posição 80 graus
    delay(15);
}
```

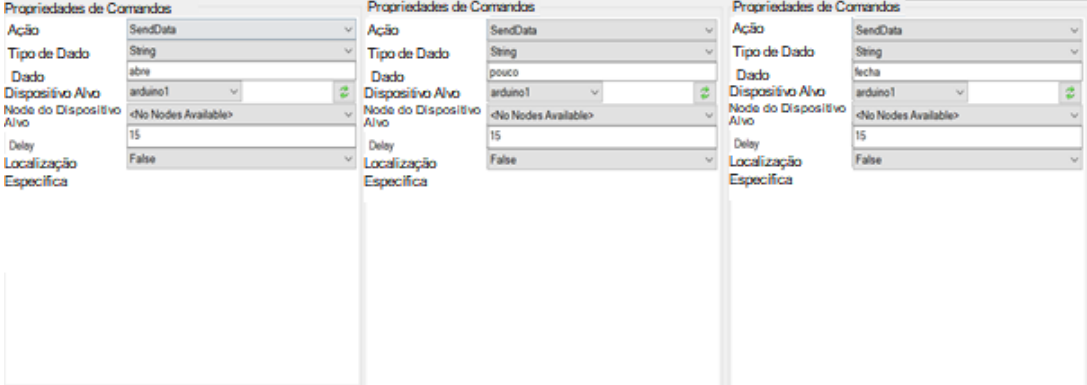
```
    }  
    if (strcmp(stringBuffer, "abre") == 0)  
    {  
        servomotor.write(150); //move o servo para a posição 150  
        delay(15);  
    }  
}
```

Figura 34 - Sentenças para o Servo Motor

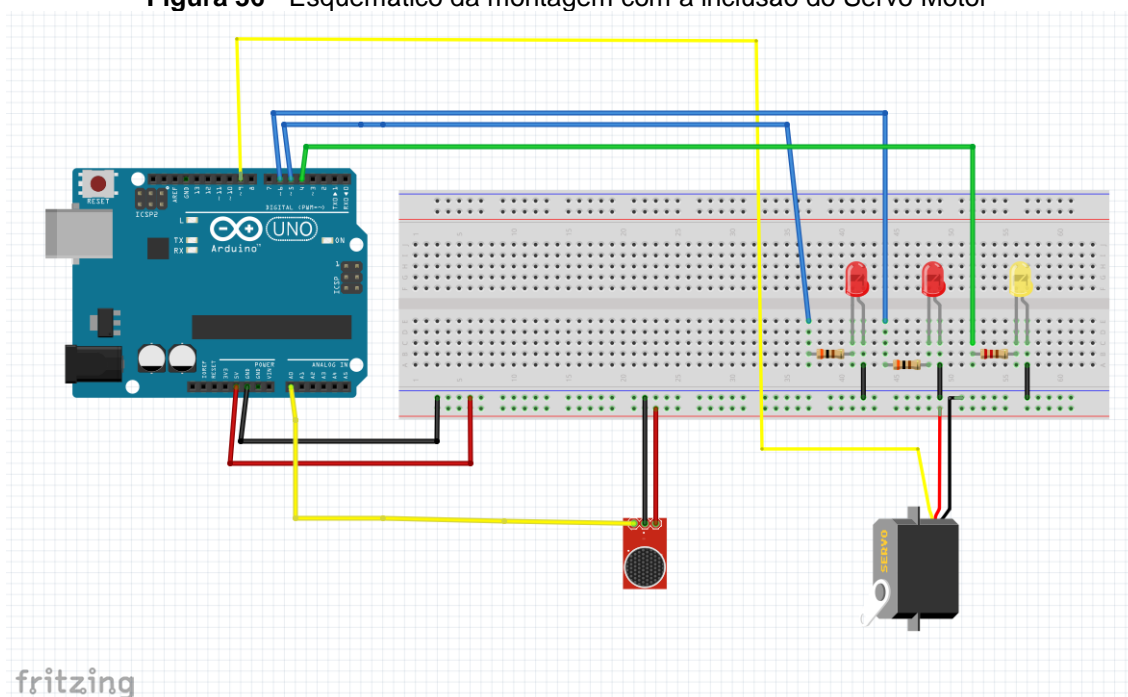


Fonte: O autor

Figura 35 - Comandos para o Servo Motor



Fonte: O autor

Figura 36 - Esquemático da montagem com a inclusão do Servo Motor

Fonte: O autor

Com a finalidade de tornar o trabalho mais próximo da realidade de uma casa automatizada foi adicionada uma lâmpada LED de 9W que será conectada à rede e seu acendimento controlado através de um módulo relé. O terminal comum do relé (C) foi ligado diretamente ao soquete da lâmpada e o terminal normalmente aberto em um terminal de plug para tomada e o outro terminal do plug da tomada foi ligado ao soquete da lâmpada. Os pinos In, 5V e GND do módulo relé foram conectados aos pinos 7, 5V e GND da placa Arduino, respectivamente, de modo que pode ser visto na Figura 38.

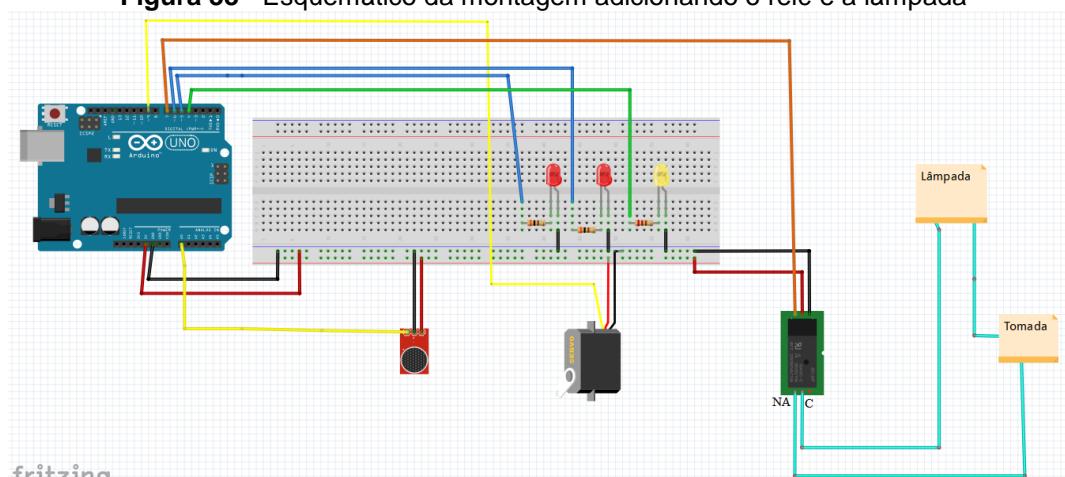
Aproveitando a sentença usada para acendimento do LED, apenas foi adicionada a palavra “lâmpada”, da forma feita na Figura 37. Foram usados os comandos para enviar as strings “acendel” e “apagal” (não é possível usar os mesmos comandos, pois acenderia o outro LED). No código, foi definido um pino para o relé (usando `#define relelamp 7`) e o pino foi definido como saída (`pinMode(relelamp, OUTPUT);`). Novamente, no condicional de caracteres conhecidos, foi adicionado o código:

```
if (strcmp(stringBuffer, "acendel") == 0)
{ digitalWrite(relelamp, HIGH);
}
if (strcmp(stringBuffer, "apagal") == 0)
{ digitalWrite(relelamp, LOW);
```

}

Figura 37 - Sentenças para o acendimento da lâmpada

Fonte: O autor

Figura 38 - Esquemático da montagem adicionando o relé e a lâmpada

fritzing

Fonte: O autor

5. CONCLUSÃO

A automação residencial é uma tecnologia que está sendo cada vez mais difundida, estudada e desenvolvida. Os benefícios de uma casa automatizada são vários como conforto, segurança personalização e economia de energia. O controle por voz permite que se tenha ainda mais benefícios, eliminando a necessidade de interação com uma interface física. Com isso, a acessibilidade é outro ponto que ganha força com a implementação de um trabalho como esse. A vida de pessoas com deficiência visual pode ser afetada de maneira extremamente positiva caso ela viva em um ambiente automatizado.

Já existem no mercado algumas soluções para a automação com controle por voz, como o Google Home, que foi citado nesse trabalho, entretanto o custo ainda é elevado. O Arduino e o BitVoicer Server, apresentados na seção 2.3 e na subseção 2.4.2, respectivamente, surgem como opções mais em conta, que podem ser mais personalizáveis, sendo possível escolher os comandos, as repostas ao invés de se utilizar das que já vêm determinadas na programação de outros dispositivos durante o trabalho, com relativa simplicidade e eficácia.

Durante o desenvolvimento do trabalho, foi construído um circuito para representar uma casa automatizada, incluindo elementos como microfone, LEDs, servo motor e relé. A maior dificuldade durante a realização do trabalho foi conseguir integrar o BitVoicer Server com o Arduino. Houve dificuldades na programação e configuração das duas partes. Mas que foram solucionados ao realizar os exemplos propostos pelo manual do BitVoicer Server fornecido pela desenvolvedora BitSophia entendendo e aprendendo os processos.

A conexão via Wi-Fi utilizando o módulo ESP8266 seria uma parte importante deste trabalho, no entanto, devido a sucessivas falhas durante sua tentativa de elaboração e à falta de recursos e tempo disponíveis, não foi possível a conclusão desta etapa. A utilização de Arduino DUE em conjunto com o módulo Wi-Fi RN171VX (como sugerido no manual do BitVoicer Server) ou um Arduino MEGA com Wi-Fi já integrado podem ser, em um momento futuro, a solução para os problemas apresentados.

Os elementos implementados (com exceção do microfone) são atuadores, ou seja, apenas realizam uma ação a partir de um comando. Para a automação residencial, também é importante o uso de sensores. Contudo, não foi possível a integração de sensores no trabalho, pois, segundo o suporte da BitSophia, o

BitVoicer Server ainda não possui a função de receber os dados dos sensores e conseguir sintetizar e expressar os resultados. Exemplo: Ao fazer a pergunta “Qual a temperatura agora?”, o BitVoicer Server deveria ser capaz de enviar um comando ao Arduino, para que seja feita a leitura da temperatura através de um sensor de Temperatura e Umidade. Essa leitura seria enviada de volta ao BitVoicer Server que enviaria a saída de áudio a resposta sintetizada: “Está 25 graus Celsius agora.”. Também segundo o suporte, a empresa planeja lançar em 2020 uma atualização do software, tornando possível este tipo de aplicação. Assim, este seria mais um passo para um projeto futuro.

A integração do Arduino e do BitVoicer Server se mostrou uma boa alternativa para a automação residencial, devido seu menor custo e uma boa comunicação mostrada entre eles. As falas dificilmente não eram reconhecidas e os circuitos do Arduino funcionaram sem problemas ao receber comandos enviados pelo BitVoicer Server. Outra vantagem em usar estes itens é a facilidade de aprendizado, já que é possível encontrar facilmente modelos de programação disponíveis na internet para o Arduino e além dos exemplos disponibilizados pela empresa BitSophia, ela oferece um bom suporte online ao ser contatada.

6. REFERÊNCIAS

- ARDUINO **Arduino**. Disponível em: <<https://www.arduino.cc/>> Acesso em: 08/11/2018.
- BALBINOT, A.; BRUSAMARELLO, V. J. **Instrumentação e Fundamentos de Medidas** Vol.1, LTC, 2006.
- BITSHOPHIA TECNOLOGIA LTDA. **BitVoicer Server 1.0 - Manual do Usuário** - 1.ed. [S.l.], 2015.
- CAMPOS, A. L. P. d. S. **Aplicação De Wavelets Para Remoção De Ruídos Em Sinais Unidimensionais** - Scientific Figure on ResearchGate. Disponível em: <https://www.researchgate.net/figure/Figura-04-Sinal-de-voz-gravado_fig3_287514255> Acesso em: 14 nov. 2019.
- CNET Home **What works with Google Assistant?**, Disponível em: <<https://www.cnet.com/products/google-assistant/compatibility/>> Acesso em: 10 ago. 2019.
- COOL BLUE B.V. **Google Home**, c1999 – 2019, Disponível em: <<https://www.coolblue.nl/en/product/803578/google-home.html>> Acesso em: 12 nov. 2019.
- DOMÍNGUEZ, H. M.; VACAS, F. S. **Domótica: Un enfoque sociotécnico**. 1.ed. Madrid. Junho de 2006.
- DORF, R. C; BISHOP, R. H. **Sistemas de controle modernos** - 8.ed. Rio de Janeiro, RJ: Livros Técnicos e Científicos, 2001.
- FERRASI, F. A.; GALVANI, A., ANDRADE, C. R. d.; MORGADO, E. M. Internet das Coisas: Uma Possibilidade de Aplicação das Tecnologias Móveis Na Educação. **Redin - Revista Educacional Interdisciplinar**, v. 1, n. 5, p.357-368, 1 nov. 2016
- FERREIRA, V. Z. G. **A Domótica Como Instrumento Para a Melhoria da Qualidade de Vida dos Portadores de Deficiência**, Instituto Federal de Educação E Tecnologia Da Paraíba, João Pessoa, 2010.
- GARBIN, S. M. et al. **Estudo da evolução das interfaces homem-computador**. 2010. Tese de Doutorado. UNIVERSIDADE DE SÃO PAULO.
- GAONKAR, R. S. **Microprocessor Architecture, Programming and Applications with the 8085**. Upper Saddle River, New Jersey 5.ed. 2002.
- GATES, B. **A Estrada do Futuro**. Companhia das Letras, São Paulo, 1995.
- GOOGLE. **Talk to the Google Assistant in multiple languages**, Disponível em: <https://support.google.com/googlenest/answer/7550584?hl=en&ref_topic=7549809> Acesso em: 10/08/2019.

- GRUBER, T. R. et al. **Intelligent Automated Assistant**, 2010. Disponível em: <<https://patents.google.com/patent/US20120016678A1/en>> Acesso em: 10 ago. 2019.
- GUILHOTO, P. J. d. S.; ROSA, S. P. C. d. S. **Reconhecimento de Voz**. Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Coimbra, 2001.
- GUIMARÃES FILHO, Heli Teodoro; SOUZA, Rafael Vilela; JÚNIOR, Walteno Martins Parreira. INFLUÊNCIA DA DOMÓTICA NO MERCADO IMOBILIÁRIO: UM ESTUDO DE CASO. **Intercursos Revista Científica**, 2017.
- HAVARD, S. **Google Home Teardown**, 2016. Disponível em: <<https://pt.ifixit.com/Teardown/Google+Home+Teardown/72684>> Acesso em: 10 ago. 2019.
- IBERDROLA. **Domótica**, c2019. Disponível em: <<https://www.iberdrola.com/inovacao/domotica>> Acesso em: 12 nov. 2019.
- LEMOS, A. **A Comunicação das Coisas: teoria ator-rede e cibercultura**. São Paulo: Annablume, 310 p., 2013.
- LEMOS, A. City and mobility. Cell phones, post-mass functions and informational territories. **MATRIZES**, v. 1, n. 1, p. 121-137, 15 out. 2007.
- LI, B. et al. Acoustic Modeling for Google Home. **Interspeech**. 2017. p. 399-403.
- LUCENA, G. G. de. **Automação Residencial por Comando de Voz Utilizando Microcontrolador**. Monografia apresentada à banca examinadora para conclusão do curso e obtenção do título de bacharel em Engenharia da Computação do UniCEUB, Brasília, 2006.
- MEDRADO, V. **Assessoria de Imprensa Serifa Comunicação**. 2008. Disponível em: <http://www.farolcomunitario.com.br/uberlandia_100_0046.htm> Acesso em: 07 nov. 2018.
- MURATORI, J. R.; DAL BÓ, P. H. **Automação residencial: histórico, definições e conceitos**. 2011. Disponível em: <http://www.instalacoeseltricas.com/download/Automacao_residencial1.pdf>. Acesso em: 07 nov. 2018.
- PAREDE, I. M.; GOMES, L. E. L. **Eletrônica – Automação Industrial**, Fundação Padre Anchieta, São Paulo, 2011.
- PEREIRA, A. P. **Como Funciona o Reconhecimento de Voz?** 2009. Disponível em: <<http://www.tecmundo.com.br/curiosidade/3144-como-funciona-o-reconhecimento-devoz-.htm>> Acesso em: 08 nov. 2018.

- RIBEIRO, C. E. **DOMÓTICA: viabilidade da Automação Residencial**. Trabalho de Conclusão de Curso, apresentado ao curso de Engenharia Elétrica do Centro Universitário do Sul de Minas, Varginha, 2018.
- RODRIGUES, G. R. **Desenvolvimento e Estudo de Um Sistema de Automação Residencial**, Projeto de Trabalho de Graduação do Curso de Engenharia Elétrica apresentado à Faculdade de Engenharia de Bauru/UNESP, Bauru, 2018.
- SANTOS, M. N. D. **Medidas de Qualidade de Voz Em Redes IP**, Dissertação apresentada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná – UFPR, Curitiba, 2006.
- SILVEIRA, L.; LIMA Q. W. **Um breve histórico conceitual da automação industrial e redes para automação industrial**. Redes para Automação Industrial. Universidade Federal do Rio Grande do Norte, 2003.
- SOUZA, Marcelo Varela de. **Domótica de baixo custo usando princípios de IoT**. Dissertação de Mestrado em Engenharia de Software – Universidade Federal do Rio Grande do Norte, Natal, 2016.
- SOUZA, S. A. d. **Utilização da Transformada de Fourier Janelada para caracterização de distúrbios na qualidade da energia elétrica**. Dissertação de Mestrado em Engenharia Elétrica - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.
- SVETLIK, J. **Alexa, Cortana, Google Assistant: What are voice assistants and how do they work?**, 2019. Disponível em: <<https://home.bt.com/tech-gadgets/internet/broadband/alexa-cortana-google-assistant-what-are-voice-assistants-and-how-do-they-work-11364211957737>> Acesso em: 10 ago. 2019.
- THOMSEN, A. **O que é Arduino**, 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 08 nov. 2018.