



**ANTONIO MENEGHETTI FACULDADE - AMF
CURSO DE SISTEMAS DE INFORMAÇÃO**

BERNARDO CERVO ZAMBERLAN

**AUTOMAÇÃO RESIDENCIAL POR COMANDOS DE VOZ COM
ARDUINO**

RESTINGA SECA/RS

2016

BERNARDO CERVO ZAMBERLAN

**AUTOMAÇÃO RESIDENCIAL POR COMANDOS DE VOZ COM
ARDUINO**

Trabalho de Conclusão do Curso de Sistemas de Informação
Orientador: Prof^o. Ms. Fábio Sarturi Prass

RESTINGA SECA/RS

2016

AGRADECIMENTOS

A **Deus**, que em seu imenso poder e sabedoria, guiou-me para que este dia chegasse.

Aos meus pais **Bernardino Zamberlan** e **Maria Zamberlan** que me foram professores das coisas simples porem mais importantes da vida. Não há palavras que possam descrever a gratidão que sinto por vocês. Aos dois que abriram mãos de vários sonhos seus para que eu pudesse alcançar este objetivo.

A minha irmã **Kellen Zamberlan** pelos concelhos, apoio e força, sem dúvidas você foi um grande pilar para a realização deste sonho.

A Todos os meus familiares que de alguma forma contribuíram para que este dia chegasse, ao meu primo **Guilherme Cervo** pelas conversas e contribuições significativas para o desenvolvimento deste trabalho.

Ao meu orientador **Fábio Prass** que com sua sabedoria e experiência orientou-me para a realização deste trabalho. Obrigado por todo o aprendizado e dedicação.

Aos **professores** do curso de Sistemas de Informação que dedicaram seu tempo a transmitir os conhecimentos que possuem, nos instigando a ser mais e sempre buscar mais.

Aos professores das disciplinas **FOIL** que transmitiram conhecimentos que contribuem diariamente na vida profissional me tornando um profissional diferenciado.

Agradeço também ao patrono desta instituição **Acad. Prof. Antonio Meneghetti**, que embora não esteja referenciado neste trabalho, contribuiu de forma significativa na minha formação acadêmica.

A todos, meu muito obrigado!

“Se você mostrar às pessoas os problemas e as soluções,
elas estarão dispostas a agir”.

Bill Gates

RESUMO

A automação residencial é baseada em componentes eletrônicos e eletromecânicos, utilizando tecnologia da informação para controlar residências através de vários tipos de comunicação como: internet, comandos de voz, controles universais entre outros. Em face disso, tem-se o Arduino o qual é uma placa de código fonte aberto e de baixo custo, utilizada para criar sistemas que se comuniquem com o ambiente externo. Diante deste contexto, o objetivo deste trabalho é utilizar a placa Arduino Black board para controlar lâmpadas através de comandos de voz utilizando a Shield EasyVR. Para atingir os objetivos foi desenvolvido um protótipo de automação residencial em uma maquete de madeira simulando uma residência real. Todo o sistema foi conectado a energia elétrica igualmente é feito em uma residência.

Palavras-chave: Arduino; Automação residencial; Shield; Módulo relé serial.

ABSTRACT

The home automation is based on electronic components and electromechanical, using information technology to control homes through various types of communication such as: internet, voice commands, universal controls among others. Given this, it has the Arduino, which is a plaque of open source and low-cost, used to create systems that communicate with the external environment. Given this context, the objective of this work is to use the Arduino board Black to control lamps through voice commands using the Shield EasyVR. To achieve the objectives was developed a home automation prototype in a wooden mockup simulating a real residence. The whole system is connected to electricity is also done in a real home.

Keywords: Arduino; Home automation; Serial relay module.

LISTA DE ILUSTRAÇÕES

Figura 1. IDE de desenvolvimento do Arduino.....	15
Figura 2. Placa Arduino Uno. Fonte: VALIDO FI (2015)	16
Figura 3. Placa Arduino Mega.....	17
Figura 4. Arduino Shield Esasy VR 3.0.	18
Figura 5. Módulo relé serial.	19
Figura 6. Representação gráfica e real da maquete da casa.....	22
Figura 7. Protótipo da automação residencial.	23
Figura 8. EasyVR Commander.....	25

LISTA DE ABREVIATURAS

IDE - Integrated Development Environment

V - Volts

GND - GrouND

SMS - Short Message Service

RJ45 - Registered Jack Type 45

GSM - Groupe Special Mobile

GPRS - General Packet Radio Services

GPS - Global Positioning System

SUMÁRIO

RESUMO.....	5
ABSTRACT.....	6
LISTA DE ILUSTRAÇÕES	7
LISTA DE ABREVIATURAS	8
1 INTRODUÇÃO.....	10
1.1 PROBLEMA DE PESQUISA	11
1.2 OBJETIVOS	11
1.2.1 Objetivo geral.....	11
1.2.2 Objetivos específicos	11
1.3 JUSTIFICATIVA	12
2 ABORDAGEM TEÓRICA.....	13
2.1 AUTOMAÇÃO RESIDENCIAL OU DOMÓTICA	13
2.2 ARDUINO	14
2.2.1 Arduino Uno	16
2.2.2 Arduino Ethernet.....	17
2.2.3 Arduino Mega	17
2.3 SHIELD	18
2.4 MÓDULO RELÉ SERIAL	19
3 METODOLOGIA	20
4 ESTUDO DE CASO	21
4.1 DETALHAMENTO DO PROJETO.....	21
4.2 PROJETO	23
4.2.1 Comando de voz	23
4.2.2 Programação da automação residencial	25
5 CONSIDERAÇÕES FINAIS	31
6 REFERÊNCIAS.....	33

1 INTRODUÇÃO

A informática surgiu com o principal objetivo de ajudar o ser humano a automatizar algumas tarefas, realizando cálculos, armazenando informações, processando dados, entre outros. A tecnologia vem em uma crescente evolução a ponto de surgir ideias futuras de residências inteligentes que, segundo Teza (2002) já existem. A domótica¹ conhecida como automação de casas e prédios, vem crescendo cada vez mais. Ela une a eletricidade, eletrônica e tecnologias da informação, criando as residências inteligentes onde é possível controlar uma casa diretamente pela internet, usando comandos de voz, controles universais, entre outros.

O Arduino é uma plataforma de hardware *open source*² e é muito utilizada para a criação de ferramentas que façam interação com o meio externo e não somente interno (SOUZA *et. al*, 2011). Tem base em uma placa de entrada e saída (*input/output*) da mesma maneira que no ambiente de desenvolvimento e utiliza a linguagem *Processing*³ que, por sua vez, é baseada na linguagem C. Sua principal característica é a facilidade de utilização e criação de sistemas inteligentes, mantendo sua premissa de ajudar pessoas a trabalhar com tecnologia (BEGHINI, 2013).

A automação residencial é baseada na integração de dispositivos eletrônicos e eletromecânicos (BOLZANI, 2007). O Arduino é um componente que pode ser utilizado para realizar a automação e, juntamente com outros componentes, pode criar sistemas capazes de

¹ A palavra domótica originou-se do latim *domus* que significa casa. Ela é a automatização e o controle aplicado à residência, utilizando sistemas que podem ser controlados por dispositivos móveis utilizando a internet ou comandos de voz. (MURATORI & HENRIQUE, 2015).

² *Hardware Open Source* é um projeto de código aberto que pode ser utilizado de qualquer forma sem precisar pagar direitos a empresa que produz o componente. (BRAGA, 2015).

³ *Processing* é uma linguagem de programação de código aberto que foi inicialmente desenvolvida para ensinar fundamentos da programação computacional num contexto visual (MAIA, 2012).

controlar residências (BEGHINI, 2013). Desta forma, pode haver mais segurança e conforto aos usuários além de economizar em alguns recursos como água e energia elétrica.

Visando entender esses conceitos foi desenvolvido um projeto de uma residência autônoma, utilizando comandos de voz, tornando possível controlar algumas funções como: acender e apagar lâmpadas, desligar tomadas, abrir e fechar portas ou portões.

1.1 PROBLEMA DE PESQUISA'

A automação residencial tem sido um nicho de mercado na área de eletrônica. Para operar essa automação necessita-se de sistemas eletrônicos eficientes. Diante disso, surge o seguinte questionamento: como controlar de forma automatizada uma residência utilizando Arduino?

1.2 OBJETIVOS

A fim de resolver o problema apresentado anteriormente, este trabalho apresenta os seguintes objetivos.

1.2.1 Objetivo geral

Identificar e compreender as contribuições do Arduino;

Desenvolver um protótipo para o controle de lâmpadas para a automação residencial utilizando o reconhecimento de voz.

1.2.2 Objetivos específicos

- a) Estudar o sistema de automação residencial e os componentes eletrônicos envolvidos.
- b) Estudar e entender os componentes da placa Arduino e sua linguagem de programação.

c) Utilizar uma placa Arduino e uma Shield com reconhecimento de voz para realizar a automação residencial em um projeto, em uma casa em forma de maquete.

1.3 JUSTIFICATIVA

O interesse pelo tema se dá pela curiosidade do autor em aprender algo que não esteja vinculado ao seu dia-a-dia, mas, que tenha alguma utilidade para o ambiente em que o mesmo está inserido. Além disso, é a segurança e a comodidade das residências autônomas proporcionada pelo sistema, o que facilitaria o cotidiano das pessoas, que poderão ter o controle da sua residência sem sair do sofá ou do quarto, sem a interação física com o ambiente. Com o uso da automação residencial e a criação de residências inteligentes o imóvel fica mais valorizado no mercado, sendo um diferencial na hora da compra ou venda. (NOGUEIRA *et al.*, 2013).

A automação residencial tem também o objetivo de diminuir o custo de energia e de água para os moradores, sendo que é possível planejar um horário para que as luzes acendam e apaguem, programar para ligar a irrigação do jardim todos os dias em determinado horário, dentre outros. O fato da automação residencial estar evoluindo rapidamente também instiga a se estudar esta temática, além de desenvolver pesquisas sobre o reconhecimento de voz, o que facilitará o dia-a-dia das pessoas.

2 ABORDAGEM TEÓRICA

Este capítulo apresenta o referencial teórico que dará base para este trabalho. Na seção 2.1 será abordado o tema da automação residencial apresentando a definição e a história. Na seção 2.2 será descrito o Arduino, expondo o surgimento da placa e os principais modelos. Na seção 2.3 será abordado a Shield para Arduino. Na seção 2.4 será vislumbrado o modulo Relé Serial.

2.1 AUTOMAÇÃO RESIDENCIAL OU DOMÓTICA

A automação residencial é um conjunto de serviços tecnológicos integrados com o objetivo de satisfazer as necessidades de segurança, comunicação, gerenciamento de energia elétrica e água e o conforto de uma habitação (MURATORI e HENRIQUE, 2015). Para que uma residência esteja totalmente automatizada, é necessário a integração do sistema de automação com todos os sistemas tecnológicos da residência.

Segundo Teza (2002) a automação teve seu surgimento ainda nos primórdios da humanidade. Considera-se automatização todo ou qualquer processo que ajude o ser humano no cotidiano, seja no âmbito industrial, comercial ou residencial. A automação utiliza sistemas inteligentes com a finalidade de agilizar um processo em questão como, por exemplo, criar um sistema que faça envios de e-mails diariamente para determinadas pessoas, poupando o trabalho manual desta tarefa.

Uma residência automatizada pode ajudar as pessoas nas tarefas diárias evitando preocupações como o esquecimento de lâmpadas ligadas ou janelas abertas. Segundo Bolzani (2004) a automação permite controlar a residência remotamente sem interação física com o ambiente, poupando tempo com tarefas, economizando energia e água.

A automação residencial abrange os seguintes setores do ambiente interno e externo:

- Instalação elétrica: iluminação, cortinas, gestão de energia entre outros.

- Sistemas de segurança: alarmes técnicos contra incêndios ou vazamentos e alarmes contra intrusão.
- Sistemas de multimídia: áudio e vídeo, jogos eletrônicos, som ambiente, entre outros.
- Sistemas de comunicação: telefonia, redes, TV por assinatura.
- Utilidades em geral: controle de temperatura, aquecimento da água, irrigação, controle de portas ou portões, entre outros.

Um ambiente automatizado pode ser controlado de diferentes maneiras, sendo as mais reconhecidas são pela internet utilizando um aplicativo para dispositivos móveis, por comandos de voz e por controles universais. A automação também pode ser controlada por envio de mensagens SMS ou por redes sociais. Uma opção para realizar a automatização de residências e edifícios é o Arduino.

2.2 ARDUINO

O Arduino teve início no *Interaction Design Institute* na cidade de Ivrea, na Itália, em 2005. O professor Massino Banzi procurava um meio barato de tornar mais fácil os estudantes de *design* trabalhar com tecnologia. O professor Massimo dividiu o seu problema o pesquisador David Guartielles da Universidade de Malmö, na Suécia. Como no mercado só existiam produtos caros e difíceis de usar, então criaram um microcontrolador⁴ que seria utilizado pelos estudantes de *Design* nos seus projetos (EVANS, NOBLE & HOCHENBAUM, 2013).

A principal exigência era que a placa fosse de custo baixo e a ideia era que o estudante não poderia gastar com o Arduino mais do que se o mesmo saísse para “comer uma pizza”. A placa foi chamada de Arduino em referência a um bar local que é frequentado pelos membros docentes e discentes do instituto (EVANS, NOBLE & HOCHENBAUM, 2013). As primeiras placas foram vendidas em formas de kit para que os alunos por conta própria criassem seus projetos. Pelo baixo custo da placa e sua fácil utilização, o Arduino foi ganhando fama, pois também introduz para a programação de microcontroladores.

⁴ Segundo o autor Martins microcontroladores são pastilhas inteligentes, que têm um processador, pinos de entradas/saídas e memória

A plataforma original do Arduino tinha algumas restrições de memória interna. Entretanto, ela possuía grandes possibilidades de realizar diversos projetos. Possuía seis entradas analógicas⁵, porém, atualmente, possuem placas com 16 entradas analógicas (SOUZA et al., 2011). Existem outras plataformas para microcontroladores, mas o Arduino continua se destacando pela sua facilidade, baixo custo e sua versatilidade. Segundo Maia (2012) uma das vantagens do Arduino é de ser um ambiente multiplataforma, ou seja, pode ser executado no Linux, Mac OS e Windows. Outra vantagem é que o código fonte pode ser enviado para o Arduino com um cabo USB, sem a necessidade de utilizar uma porta serial, uma vez que já não estão mais presentes nos computadores atuais.

A IDE para programar o Arduino pode ser encontrada gratuitamente no site www.arduino.cc. Todavia, existem *plug-ins* para programar o Arduino, que podem ser instalados em outras IDEs de desenvolvimento como o Eclipse e o Visual Studio. Seguindo o propósito do Arduino, a IDE de desenvolvimento é simples e de fácil instalação e utilização. A mostra a IDE de programação do Arduino com as principais funções utilizadas.

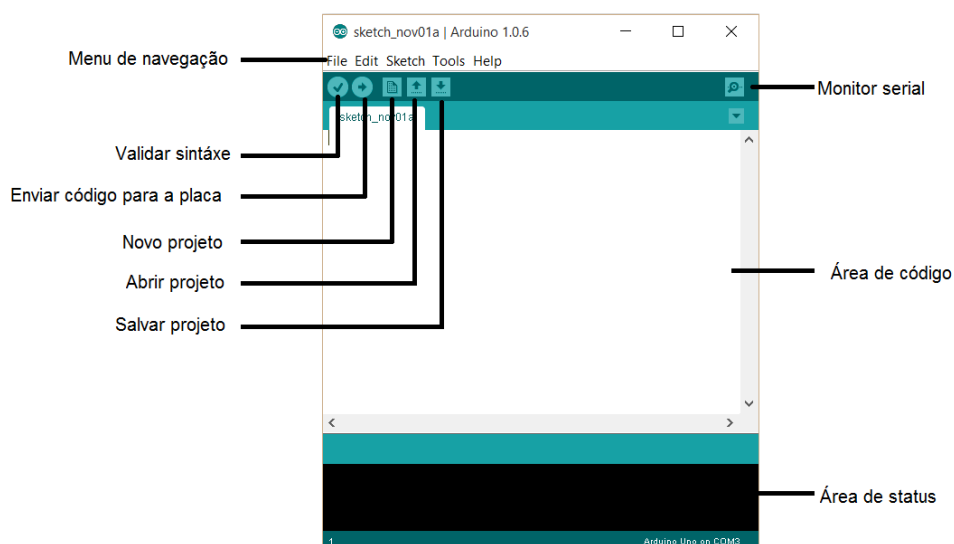


Figura 1. IDE de desenvolvimento do Arduino.

⁵ As entradas analógicas são módulos conversores analógico/digital, que convertem um sinal de entrada em um valor digital, normalmente de 12 bits (4096 combinações) (MATUCHAKI, 2011).

2.2.1 Arduino Uno

Em 25 de setembro de 2010 foi anunciado o Arduino Uno, que possui compatibilidade de pinos com as versões anteriores deste microcontrolador. A principal diferença das placas anteriores é a inclusão de um microcontrolador *ATmega8U2* como um conversor USB para serial, assim substituindo o *chipset FTDI* das placas anteriores. Como o Arduino é de código fonte aberto, é possível reprogramar o *ATmega8U2* fazendo com que a placa vire outro dispositivo como, por exemplo, um mouse (EVANS, NOBLE, HOCHENBAUM, 2013). A Figura 2. Placa Arduino Uno mostra o Arduino Uno e a descrição dos seus componentes.

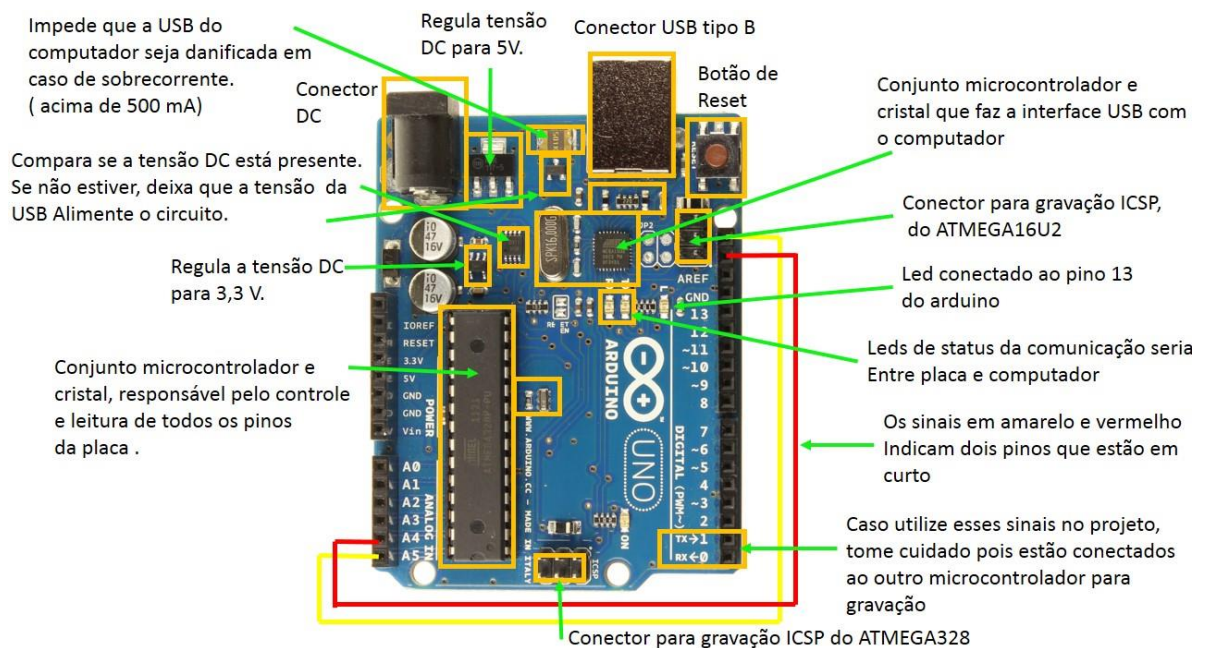


Figura 2. Placa Arduino Uno.

Fonte: VALIDO FI (2015)

O Arduino Uno é uma opção interessante para iniciar os estudos, uma vez que é de baixo custo. Como o microcontrolador também é de código fonte aberto, existem outras placas que foram desenvolvidas baseadas no Arduino Uno, como a BlackBoard desenvolvida pela empresa brasileira Robocore.

2.2.2 Arduino Ethernet

Esta versão do Arduino foi lançada na mesma época que o Arduino Uno, tem baixa potência e as principais diferenças para a versão Uno é que a Arduino Ethernet possui conector RJ45 para conexão com Ethernet e também para leitor de cartão micro SD. Com essas características o Arduino Ethernet é ideal para sistemas de monitoramento remoto e estações de registros de dados (EVANS, NOBLE & HOCHENBAUM, 2013).

2.2.3 Arduino Mega

A nova versão do Arduino possui memória flash de 256 *kilobytes*, fornece um aumento significativo na funcionalidade de entrada e saída, ou seja, com o aumento da memória ele é ideal para grandes projetos que controlam grandes quantidade de entradas e saídas. A placa possui 4 portas serial de hardware e 54 pinos digitais de entrada e saída, como mostra a , um número bem maior que o Arduino Uno, que só possui 14 pinos digitais.

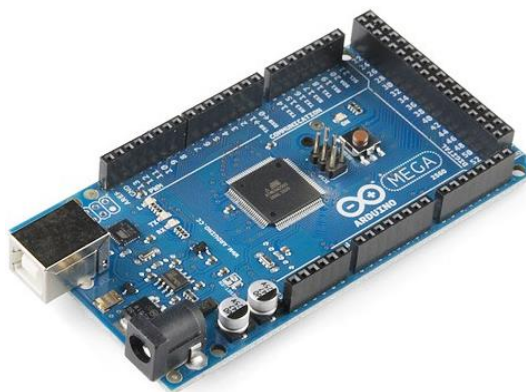


Figura 3. Placa Arduino Mega.
Fonte: Robocore.net

2.3 SHIELD

Placas especializadas chamadas Shields podem expandir a funcionalidade básica do Arduino. Elas podem ser empilhadas umas sobre as outras para adicionar ainda mais funcionalidade (EVANS; NOBLE; HOCHENBAUM, 2013). As Shields são muito utilizadas na automação residencial sendo que é possível conecta-las na internet facilitando o controle das residências por meio de sistemas web que podem ser acessados pelo celular.

Existem diversos modelos de Shields para Arduino cada uma com uma função diferente a lista abaixo mostra algumas Shields e sua principal função.

- Arduino Ethernet Shield: sua principal função é conectar o Arduino a internet, ela possui uma porta RJ45 para facilitando a comunicação do Arduino com o ambiente externo.
- Arduino Shield Celular e GPS: é caracterizada por possuir as tecnologias GSM, GPRS e GPS. Com esta Shield é possível enviar e receber SMS, fazer ligações, acessar a internet, apenas com um chip de celular, sendo muito utilizada para rastrear carros e bens pessoais.
- Arduino Shield Joystick: possui botes de analógicos de *joystick*, que podem servir para fazer controles remotos.
- Arduino Shield Easy: esta placa possui reconhecimento de comandos de voz, deixando o Arduino ainda mais dinâmico e versátil. Ela possui um microfone para captura do áudio e saída para alto-falante onde é possível retornar áudios para maior interação com o usuário. A Figura 4. Arduino Shield Esasy VR 3.0. mostra a Shield de reconhecimento de voz que será utilizada neste trabalho.

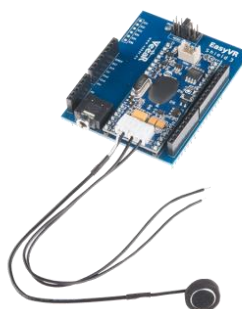


Figura 4. Arduino Shield Esasy VR 3.0.

Fonte: robocore.net

2.4 MÓDULO RELÉ SERIAL

Esse módulo constitui um item muito importante para a automação residencial. A relé é um dispositivo eletrônico capaz de controlar grandes tensões e correntes entre 100V (volts) a 240 V a partir de pequenas tensões de outro circuito (MAIA, 2012).

A mostra o componente que será utilizado para controlar a energia elétrica na automação residencial deste projeto. A diferença entre o módulo relé e o módulo relé serial é a quantidade de cargas que cada um pode controlar, ambos utilizam dois pinos digitais do Arduino. O módulo relé serial pode controlar até quatro cargas por unidade, sendo possível conectar outras unidades acionando no máximo quarenta relés, ou seja, com apenas dois pinos digitais do Arduino é possível controlar até quarenta cargas elétricas. Para utilizar o módulo é necessária obrigatoriamente que o circuito seja alimentado com uma fonte de 12 V.



Figura 5. Módulo relé serial.
Fonte: robocore.net

3 METODOLOGIA

O presente trabalho é caracterizado por um estudo exploratório e fará o uso da pesquisa do tipo qualitativa que, segundo Godoy (1995), envolve a obtenção de dados descritivos sobre pessoas, lugares e processos interativos pelo contato direto do pesquisador com a situação estudada, procurando compreender os fenômenos dos participantes da situação em estudo. O contexto do projeto de automação residencial se dará em uma maquete de madeira, simulando uma residência.

Afim de alcançar os objetivos aqui propostos e solucionar o problema de pesquisa foi desenvolvida uma maquete de uma residência autônoma de modo que as lâmpadas sejam acionadas por comando de voz. Para isso, será utilizada uma placa Arduino Black Board. Para iniciar o estudo dos componentes da placa e sua linguagem de programação foi adquirido uma Shield para Arduino modelo EasyVR 3.0 com reconhecimento de voz e três módulos relés que ficarão responsáveis por controlar a energia em 220V.

A ideia do projeto da residência automatizada é que o Arduino compreenda e execute alguns comandos pronunciados pelo usuário, tendo por exemplo: “sala”, “acender”. O Arduino deve entender esse comando e ligar a lâmpada da sala. O mesmo deve ocorrer quando o usuário pedir para desligar a lâmpada da sala. Além disso o usuário poderá ligar ou desligar todas as lâmpadas da casa com apenas dois comandos de voz.

A ferramenta foi desenvolvida e testada em laboratório, a Shield EasyVR 3.0 está acoplada encima do Arduino e o módulo relé conectado na Shield através de jumpers⁶ macho e fêmea. O fio da energia elétrica está conectado ao módulo relé o qual faz o processo de ligar e desligar as lâmpadas.

⁶ Jumpers são fios que servem para conectar o Arduino ou Shields a outros componentes (CURY, HIRSCHMANN,2014)

4 ESTUDO DE CASO

A fim de alcançar os objetivos propostos neste trabalho e solucionar o problema de pesquisa foi criado um projeto de uma maquete em madeira simulando uma residência. Foi implementado o acionamento de lâmpadas ligadas diretamente na energia elétrica e também a simulação do acionamento de um portão eletrônico, ambos por comandos de voz. Todos os itens elétricos, a placa Arduino e a Shield EasyVR utilizados para realizar esta automação são os mesmos que podem ser aplicados diariamente nas residências criando conforto, comodidade e redução de custos para as pessoas.

4.1 DETALHAMENTO DO PROJETO

Para realizar a automação residencial por comandos de voz foram necessários os seguintes itens:

- Back Board: a placa Arduino desenvolvida no Brasil junta as melhores partes das placas Arduino originais da Itália de baixo custo. Ela será o cérebro de todo o sistema da automação.
- Shield Easy VR 3.0: é umas das peças mais importantes deste projeto. Ela é responsável por reconhecer os comandos de voz e se comunicar com o Arduino.
- Modulo relé serial: componente elétrico que fará o controle das cargas elétricas em 220v que irão acionar as lâmpadas da casa.
- Motor de passo: componente elétrico que é responsável por abrir e fechar o portão da garagem.
- Jumpers: são fios que servem para conectar o Arduino ou Shields a outros componentes (CURY & HIRSCHMANN, 2014). Eles foram utilizados para conectar o modulo relé serial e o motor de passo ao Shield.

Para a demonstração visual da automação foi criada uma maquete em madeira, primeiramente foi criado em 3D utilizando a ferramenta SketchUp. Pelo projeto em 3D foi construída a maquete dividida em quarto, sala e garagem. Os detalhes do projeto podem ser observados na o projeto em 3D e a mesma projetada em madeira.



Figura 6. Representação gráfica e real da maquete da casa.

O desenvolvimento do projeto é baseado na plataforma Arduino. Desta forma foi criado o protótipo da automação utilizando a ferramenta Fritzing que permite criar circuitos elétricos com Arduino. A **Erro! Fonte de referência não encontrada.** foi criado o esquema da automação residencial com as mesmas peças utilizadas na maquete em madeira. A Shield EasyVR foi acoplada em cima do Arduino, os pinos digitais e analógicos da placa Arduino ficam na mesma posição, ou seja, os componentes foram conectados diretamente á Shield sendo que a mesma está conectada ao Arduino para fazer a comunicação conforme o item 1 da Figura 7.

O modulo relé serial, item 2 foi ligado aos pinos digitais do Arduino quatro e cinco, também foi conectado ao pino GND e ao pino VIM que transmite a mesma corrente elétrica que o Arduino recebe. Como o Arduino está sendo alimentado por uma fonte de 12V o pino VIM passa uma

corrente de 12V, que é o necessário para ativar o relé. O motor de passo que pode ser visto no item 3 foi conectado ao seu drive e o drive ao Arduino utilizando quatro pinos digitais, o pino GND e o pino 5V demonstrado no item 4 da Figura 7.

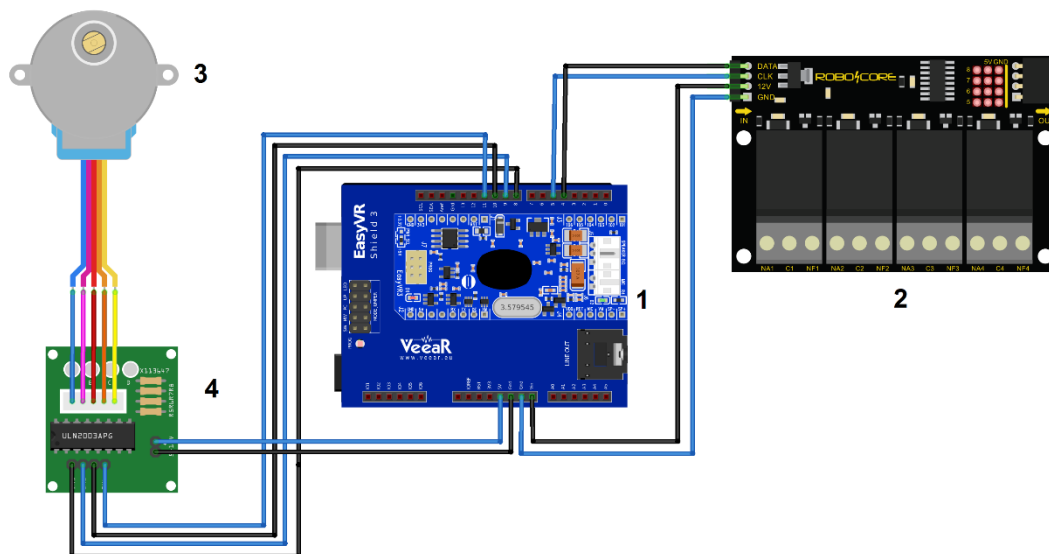


Figura 7. Protótipo da automação residencial.

4.2 PROJETO

O desenvolvimento do projeto da automação residencial foi dividido em duas partes: Primeiramente, foi ensinado a Shield EasyVR os comandos de voz que ele deveria compreender. Posteriormente, foi desenvolvido o código para que o Arduino execute as tarefas da automação, após a Shield compreender o comando mencionado.

4.2.1 Comando de voz

Os comandos de voz são ensinados a Shield EasyVR por meio do software EasyVR Commander que é disponibilizado no site da fabricante da placa. Ao conectar o Arduino no

computador estando com a Shield acoplada o EasyVR Commander detecta todos os comandos de voz que a placa possui. Os comandos de voz são divididos nos seguintes grupos:

- *Trigger* índice zero: é o grupo onde são ensinados os comandos de gatilho para a Shield, ou seja, ao falar este comando o Shield entende que está sendo chamado e espera um próximo comando para executar uma ação.
- *Group* índice 1 ao 15: é o grupo de comandos que podem ser executados após o grupo da *Trigger*. Neles podem ser gravados diversos comandos como por exemplo: ligar, apagar, abrir e fechar.
- *Password*: é o grupo onde pode ser ensinado uma senha ao Shield, ou seja, após compreender o comando da *trigger* a Shield vai esperar que o usuário fale a senha de acesso. Isso funciona como a autenticação do sistema assim a placa não irá responder a qualquer pessoa que tentar chamá-la.
- *WordSet*: é o grupo onde são ensinadas as palavras ao Arduino. Qualquer pessoa pode falar esses comandos a placa.
- *SoundTable*: esta Shield tem saída para alto-falante. Isso faz que ela seja ainda mais versátil entendendo um comando que ela pode reproduzir um áudio, mostrando mais interação com o usuário.

Para criar a automação residencial proposta neste trabalho, foi ensinado a Shield os comandos para, ligar e desligar a luz, abrir e fechar o portão. **A Erro! Fonte de referência não encontrada.** mostra os comandos de voz que foram ensinados a Shield para executar a automação. Os comandos de voz foram divididos em três grupos: *Trigger*, *Group 1* e *Group 2*. No grupo da *Trigger* foi ensinado o comando computador utilizado como nome do sistema. Toda vez que este nome for mencionado, a Shield irá reconhecer e esperar um comando do *Group 1*. Neste grupo são esperados alguns comandos referentes aos ambientes da casa que são: sala, quarto, garagem e casa.

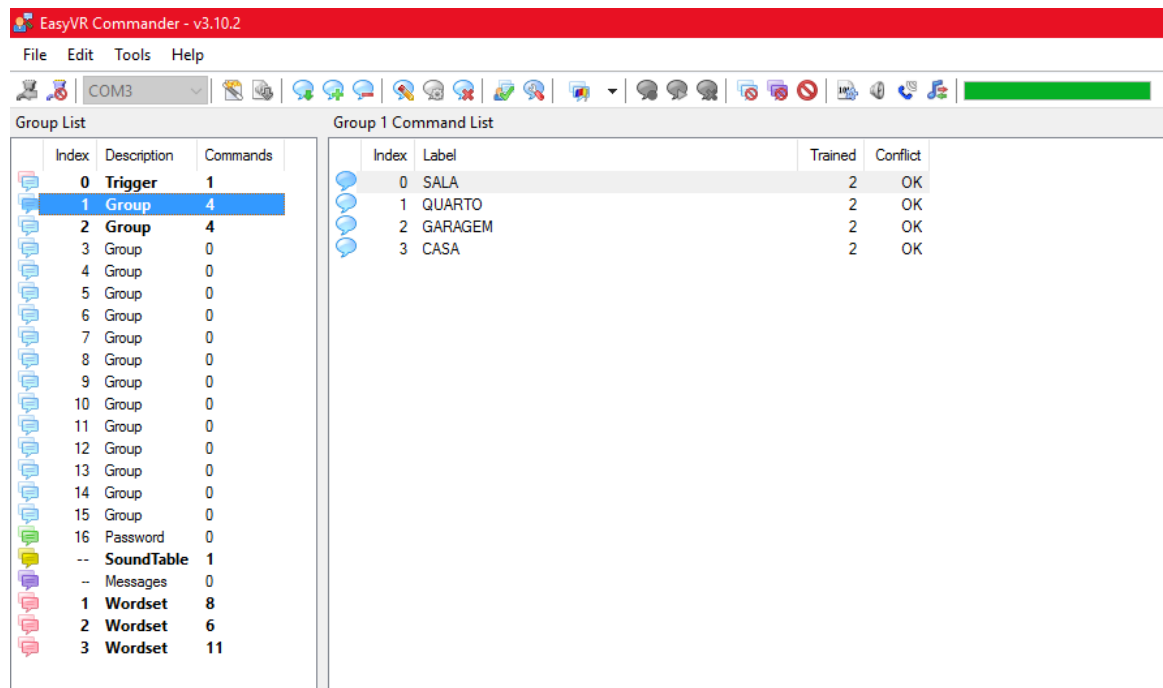


Figura 8. EasyVR Commander.

Ao falar algum destes comandos a Shield espera um comando do *Group 2* que são: acender, apagar, abrir e fechar. Então no *Group 1* é selecionado o ambiente da casa, e no *Grup 2* é informado qual a ação que o Arduino deve realizar sobre este ambiente.

4.2.2 Programação da automação residencial

O EasyVR Commander facilita ao desenvolvimento do código da automação residencial, após ensinar os comandos para a Shield ele tem uma opção chamada *Generate Code*. Esta função gera todo o código que será passado para o Arduino. O código gerado faz toda a comunicação com a Shield para reconhecer os comandos de voz utilizando uma biblioteca disponibilizada pelo fabricante porem é necessário customizar para que este código atenda aos requisitos desse projeto. A Listagem 1 mostra as bibliotecas que foram incluídas para executar a automação.

Listagem 1. Bibliotecas utilizadas.

1. `#include "EasyVR.h"`
2. `#include <Stepper.h>`
3. `#include <SerialRelay.h>`
4. (...)

Na linha 1 é incluído a biblioteca da ShieldEasy VR, que é a classe que faz o reconhecimento dos comandos que foram ensinados a placa. Na linha 2, observa-se a biblioteca do motor de passo, que é responsável por controlar o motor de passo que abre e fecha o portão da garagem. E, por último, a biblioteca do modulo relé serial que faz o controle das cargas elétricas. Ambas as bibliotecas são desenvolvidas pelos fabricantes dos componentes e podem ser baixadas pelo site das mesmas.

Listagem 2. Modulo Rele

1. (...)
2. `const` byte NumeroModulos = 1;
3. `SerialRelay` relays(4, 5, NumeroModulos);
4. (...)

Na Listagem 2 é exibido o código que diz ao Arduino em quais pinos digitais o módulo relé foi conectado e quantos módulos relés estão acoplados entre si.

Para cada grupo de comandos o EasyVR Commander, criou-se um enumerador que são tipos de campos que consistem em um conjunto fixo de constantes, sendo uma lista de valores pré-definidos (VINÍCIUS, 2012). Na Listagem 3, relaciona-se o código dos enumeradores criados automaticamente pelo EasyVR Commander, cada enumerador representa um comando de voz que foi ensinado ao Shield. Eles possuem os mesmos nomes que foram dados aos comandos apresentados na Figura 8.

Listagem 3. Enumeradores.

1. (...)
2. `enum` Groups
3. {
4. GROUP_0 = 0,
5. GROUP_1 = 1,
6. GROUP_2 = 2,
7. };
8. `enum` Group0
9. {
10. G0_COMPUTADOR = 0,
11. };
12. `enum` Group1
13. {
14. G1_SALA = 0,
15. G1_QUARTO = 1,
16. G1_GARAGEM = 2,
17. G1_CASA = 3,
18. };
19. `enum` Group2

```

20. {
21.     G2_ACENDER = 0,
22.     G2_APAGAR = 1,
23.     G2_ABRIR = 2,
24.     G2_FECHAR = 3,
25. };
26. enum LocalAmbiente
27. {
28.     SALA = 0,
29.     QUARTO = 1,
30.     GARAGEM = 2,
31.     CASA = 3,
32. }Ambiente;
33. (...)

```

Nas linhas 2 a 7 foi criado o enumerador contendo todos os grupos, entre a linha 8 e 10 destaca-se o código do enumerador do grupo da trigger com apenas um valor que é o nome dado ao sistema. O próximo enumerador criado representa o grupo dos ambientes da residência, nas linhas 12 a 18 o enumerador do grupo 1 com os valores dos respectivos ambientes da residência que foram ensinados a Shield. O último enumerador criado automaticamente pelo EasyVR Commander representa o grupo das ações. Os valores que estão no enumerador entre as linhas 19 e 25 são os mesmos que foram ensinados à placa. Como a residência possui ambientes que tem a mesma ação, foi criado um enumerador para controlar em qual ambiente deve ser executada determinada função. Por exemplo, na sala ou no quarto contém a mesma tarefa, que é acender ou apagar a lâmpada. Então o enumerador tem o valor do ambiente para executar o acionamento ou não da lâmpada no local correto.

Após a Shield reconhecer um comando de voz é esperado que uma ação seja executada, ou seja, o código gerado automaticamente pelo EasyVR Commander desenvolve um método chamado *action* que é a ação propriamente dita. A Listagem 4 demonstra o código do método gerado com as alterações necessárias para que ocorra a automação residencial nos ambientes da casa. O método basicamente verifica qual comando foi reconhecido utilizando a estrutura de controle *switch*, primeiramente é verificado para qual grupo foi disparado o comando, após isso o sistema verifica quando foi o comando reconhecido para o grupo.

Listagem 4. Automação residencial.

```

1. void action() {
2.     switch (group) {
3.         case GROUP_0:
4.             switch (idx) {
5.                 case G0_COMPUTADOR:

```

```

6.         group = GROUP_1;
7.         break;
8.     }
9.     break;
10.    case GROUP_1:
11.        switch (idx) {
12.            case G1_SALA:
13.                Ambiente = SALA;
14.                group = GROUP_2;
15.                break;
16.            case G1_QUARTO:
17.                Ambiente = QUARTO;
18.                group = GROUP_2;
19.                break;
20.            case G1_GARAGEM:
21.                Ambiente = GARAGEM;
22.                group = GROUP_2;
23.                break;
24.            case G1_CASA:
25.                Ambiente = CASA;
26.                group = GROUP_2;
27.                break;
28.        }
29.        break;
30.    case GROUP_2:
31.        switch (idx) {
32.            case G2_ACENDER:
33.                switch (Ambiente) {
34.                    case SALA:
35.                        relays.SetRelay(1, SERIAL_RELAY_ON, 1);
36.                        break;
37.                    case QUARTO:
38.                        relays.SetRelay(4, SERIAL_RELAY_ON, 1);
39.                        break;
40.                    case CASA:
41.                        relays.SetRelay(1, SERIAL_RELAY_ON, 1);
42.                        delay(50);
43.                        relays.SetRelay(4, SERIAL_RELAY_ON, 1);
44.                        break;
45.                }
46.                group = GROUP_0;
47.                break;
48.            case G2_APAGAR:
49.                switch (Ambiente) {
50.                    case SALA:
51.                        relays.SetRelay(1, SERIAL_RELAY_OFF, 1);
52.                        break;
53.                    case QUARTO:
54.                        relays.SetRelay(4, SERIAL_RELAY_OFF, 1);
55.                        break;
56.                    case CASA:
57.                        relays.SetRelay(1, SERIAL_RELAY_OFF, 1);
58.                        delay(50);
59.                        relays.SetRelay(4, SERIAL_RELAY_OFF, 1);
60.                        break;
61.                }
62.                group = GROUP_0;

```

```

63.         break;
64.     case G2_ABRIR:
65.         for (int i = 0; i <= 1; i++) {
66.             myStepper.step(-2048);
67.         }
68.         myStepper.step(-512);
69.         group = GROUP_0;
70.         break;
71.     case G2_FECHAR:
72.         for (int i = 0; i <= 1; i++) {
73.             myStepper.step(2048);
74.         }
75.         myStepper.step(512);
76.         group = GROUP_0;
77.         break;
78.     }
79.     break;
80. }
81. }

```

A função principal do método da Listagem 4 é executar a ação após reconhecer um comando de voz. Primeiramente, ele verifica para qual grupo foi dito o comando, ou seja, se o comando mencionado pelo usuário foi para grupo zero que é o grupo da *trigger* e o mesmo foi entendido pela Shield, o sistema passa a esperar por um comando de outro grupo neste caso o grupo um. Ao reconhecer o comando do primeiro grupo, o método valida qual comando foi reconhecido e executa a ação definida no método.

Nas linhas 13 a 26 está separado os códigos em que o sistema seleciona o ambiente da casa, conforme o comando de voz e, em seguida passa a esperar um comando do grupo dois. Este grupo é responsável por executar a ação no ambiente escolhido.

Nas linhas 32 a 45 o sistema verifica o ambiente em que deve ser executado, há o acionamento da carga do relé e, então a lâmpada é ligada. Os métodos que fazem a comunicação com o relé são disponibilizados na biblioteca fornecida pelo fabricante do componente. Para este método deve ser necessário passar alguns parâmetros como: o número do relé que vai ser executado, ativar ou desativar a carga no relé e o número do modulo em que está o relé sendo possível ter mais de um modulo conectado.

Nas linhas 48 a 61 o processo é o mesmo, porém as cargas são desativadas e a lâmpada é desligada. Com isso, a automação do acionamento das lâmpadas já está pronta e parte da casa já pode ser controlada por voz.

Entre as linhas 64 e 77 é exibido o código que faz a comunicação com o motor de passo que está sendo utilizado para simular um motor de um portão. Na linha 66 e 73 é chamado o método que é disponibilizado na biblioteca do componente fazendo com que o motor gire para abrir e fechar o portão. O valor que é passado para o método foi obtido através de vários testes, verificando quantas voltas eram necessárias para que o portão fosse completamente aberto utilizando os recursos disponíveis para simular o mesmo na vida real.

O principal objetivo deste item é simular a abertura de um portão por comandos de voz independente do mecanismo utilizado para abrir do portão o mesmo deve ser ativado pelo Arduino quando um comando de voz for reconhecido.

5 CONSIDERAÇÕES FINAIS

Conforme apresentado no presente trabalho, a automação residencial vem crescendo em grande escala, se tornando uma opção ideal para deixar qualquer imóvel totalmente inteligente. O diferencial é que a automação varia conforme a necessidade das pessoas que irão utilizar estes serviços, sendo que, com ela é possível tornar ambientes mais confortáveis, seguros, práticos e eficientes. O controle de residências através de sistemas embarcados, utilizando o Arduino como um computador central mostrou que pode ser uma boa opção e de fácil implementação. Os sistemas de automação podem ser inseridos em vários contextos atendendo a população de idade avançada, além de pessoas com necessidades especiais, tornando mais dinâmica e acessível a vida de quem a utiliza.

O trabalho desenvolvido propôs a criação de um sistema que fosse capaz de controlar algumas funções de uma casa utilizando comandos de voz para interagir com a ambiente externo. O sistema proposto partiu da ideia de se controlar de forma automatizada a energia elétrica de uma residência, utilizando uma placa Arduino BlackBoard, uma Shield EasyVR de reconhecimento de voz e um Relé Serial.

O estudo de caso deste projeto tem foco em alcançar os objetivos propostos no presente trabalho, utilizando os componentes estudados na fundamentação teórica. Após o desenvolvimento do projeto notou-se que a automação residencial foi aplicada com sucesso a uma maquete simulando uma residência utilizando comandos de voz, solucionando assim o problema de pesquisa abordado neste trabalho.

O tempo de resposta entre os comandos de voz são curtos, embora a Shield EasyVr precise evoluir no reconhecimento de comandos maiores, como por exemplo, frases, ela é uma boa opção a ser utilizada nos sistemas inteligentes que podem ser aplicados nas residências dando mais versatilidade e uma valorização maior para o imóvel onde o sistema é inserido.

Por fim conclui-se que a automação residencial pode ser aplicada por meio de várias comunicações conforme mencionado anteriormente. Por comandos de voz, pela internet ou controles universais entre outros, a maneira como ela é desenvolvida deve ser adequada as

necessidades do usuário. Por fim observou-se que a automação residencial está em constante crescimento e os sistemas inteligentes podem ser aplicados a outros setores como o da saúde, educação indústria entre outros agilizando processos e contribuindo para o desenvolvimento do setor.

6 REFERÊNCIAS

BEGHINI, L. B. **Automação Residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android**, nov. 2013. Disponível em: <<http://goo.gl/K39oLz>>. Acessado em: 16 out. 2015.

BOLZANI, C.A.M. **Desmistificando a Domótica**. Sinergia, São Paulo, v. 8, n. 1, p. 17-20, jan./jun. 2007 Disponível em: <<http://goo.gl/PEHUUo>>. Acessado em: 16 out. 2015.

CURY, T. E.; HIRSCHMANN D. R. **Ensino de matemática através do Arduino**. Disponível em: <<http://goo.gl/wLOubO>>. Acessado em 22 nov. 2015.

DIAS, C. L. A.; PIZZOLATO, N. D. **DOMÓTICA**. Aplicabilidade e Sistemas de Automação Residencial. VÉRTICES, v. 6, n. 3, set./dez. 2004. Disponível em: < <http://goo.gl/tCIqQw>>. Acessado em 24 abr. 2016.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. Editora Novatec, São Paulo, p. 25-26, ago. 2013.

GODOY, A. S. **Introdução à pesquisa qualitativa e suas possibilidades**. Revista de Administração de Empresas São Paulo, v. 35, n. 2, p. 57-63, mar./abr. 1995. Disponível em: < <http://goo.gl/uJs6Hu>>. Acessado em 15 nov. 2015.

MAIA, G. M. F. **Acionamento remoto de portões elétricos via celular através de microcontrolador**. Centro Universitário de Brasília -UniCEUB, jun. 2012. Disponível em: <<http://goo.gl/OhapMa>>. Acessado em 22 out. 2015.

MATUCHAKI, G. **Projeto de um sistema de climatização de aviário**, 2011. Disponível em: < <http://goo.gl/IR5WJP> >. Acessado em 30 out 2015.

MURATORI, J. R.; HENRIQUE, P. H. D. **Automação residencial: histórico, definições e conceitos**. Disponível em < <http://goo.gl/dMBh4o>>. Acessado em 22 out 2015.

NOGUEIRA, C. L. M.; ALARCÃO, L. M.; BRAZ, A. R. B.; MOREIRA, J. R. **AutoControl: uma proposta para acessibilidade e segurança residencial com o apoio da plataforma Arduino**, jun. 2013. Disponível em: < <http://goo.gl/ULNii2>>. Acessado em 26 nov. 2015.

SOUZA, A. R.; PAIXÃO, A. C.; UZÊDA, M. A. D.; DUARTE, S.; AMORIM, H. S. **A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC**. Instituto de

Física, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, mar. 2011. Disponível em: <<http://goo.gl/3AeDq3>>. Acessado em: 25 set. 2015.

TEZA, V.R. **Alguns aspectos sobre a automação residencial – domótica**, mai. de 2002 Disponível em: <<https://goo.gl/dzX8KU>>. Acessado em 30 out. 2015.

VALIDO FI M. M. P. **Composição de cores** - ensinando conceitos de física com Arduino, fev. 2015. Disponível em: <<http://goo.gl/SRRk3W>>. Acessado em 21 abr. 2016.

VINÍCIUS, T. **Tipos Enum no Java**, set. 2012. Disponível em: <<http://goo.gl/IN8c9M>>. Acessado em 21 abr. 2016.