

**FACULDADE DOCTUM
JEAN DIAS ROCHA**

**ANÁLISE DE COMPONENTES PARA VIABILIZAR UM SISTEMA RESIDENCIAL
AUTOMATIZADO COM SUPORTE IOT**

Juiz de Fora
2019

JEAN DIAS ROCHA

**ANÁLISE DE COMPONENTES PARA VIABILIZAR UM SISTEMA RESIDENCIAL
AUTOMATIZADO COM SUPORTE IOT**

Monografia de Conclusão de Curso,
apresentada ao curso de Engenharia
Elétrica, Faculdade Doctum de Juiz de
Fora, como requisito parcial à obtenção
do título de Bacharel em Engenharia
Elétrica.

Orientação: Prof. José Carlos Miranda
Grizendi

Juiz de Fora
2019

Ficha Catalográfica elaborada pela Biblioteca Faculdade Doctum/JF

Dias Rocha, Jean.

Análise de componentes para viabilizar um sistema residencial automatizado com suporte para IoT / Jean Dias Rocha - 2019.

56f.

Monografia (Curso de Engenharia Elétrica) –
Faculdade Doctum Juiz de Fora.

1. Automação. 2. IoT

I. Análise de componentes para viabilizar um sistema residencial automatizado com suporte para IoT. II. Faculdade Doctum Juiz de Fora

FOLHA DE APROVAÇÃO

O Trabalho de Conclusão de Curso intitulado **ANÁLISE DE COMPONENTES PARA VIABILIZAR UM SISTEMA RESIDENCIAL AUTOMATIZADO COM SUPORTE IOT**, elaborado pelo aluno **JEAN DIAS ROCHA** foi aprovado por todos os membros da Banca Examinadora e aceita pelo curso de ENGENHARIA ELÉTRICA, como requisito parcial da obtenção do título de Bacharel em ENGENHARIA ELÉTRICA.

Juiz de Fora, 12 de dezembro de 2019.



Professor Orientador MSc. José Carlos M Grizendi



Professor MSc. Luis Gustavo Schröder e Braga



Professora DSc. Daniele de Alcantara Barbosa

AGRADECIMENTOS

Agradecimentos são sentimentos de retribuição pela gratidão que sentimos pelas pessoas que nos ajudam, nos zelam, nos cuidam. E é impossível não lembrar daqueles que estiveram e estão próximos de mim, me motivando e me fazendo não desistir do que acredito que seja o melhor.

Primeiramente, agradeço a Deus por ter me dado saúde e força para superar as dificuldades, além de sempre me conduzir pelo caminho certo.

Aos meus pais, Cleres e Herika, que me deram a vida e me ensinaram a vivê-la com dignidade, além de sempre me apoiarem, incentivando a continuar em frente. Agradeço aos meus irmãos, Túlio por ser exemplo de superação e amor, e Lucas por me incentivar, além de sempre me divertir. Agradeço também a todos da minha família por tornarem essa conquista possível e sempre estarem ao meu lado.

Ao meu orientador, o professor Mestre José Carlos Miranda Grizendi, pela confiança e paciência, além das várias lições ensinadas dentro e fora de sala ao longo do curso, me transformando em um eterno aprendiz.

Ao coordenador do curso, professor Luís Gustavo Schroder e Braga, pela atenção e dedicação prestada na elaboração desse trabalho, contribuindo na organização das ideias, bem como na revisão do trabalho.

A todos os professores que me ensinaram ao longo desses cinco anos o que é ser um engenheiro, sem essas pessoas maravilhosas não seria possível aprender nada disso. E a toda equipe do Campos de Engenharia Doctum, que sempre me prestaram um serviço de excelência atendendo sempre que possível as demandas com muita eficiência.

A todos os colegas de sala e amigos adquiridos nessa jornada, pela ajuda e compreensão em momentos difíceis. Aos grandes amigos, Harison Lucas Lira de Carvalho, Matheus Mendes Fonseca e Raphael Alves dos Reis, que além de me ajudarem muito, me emprestaram alguns dos dispositivos que foram usados nesse projeto.

Enfim, agradeço a todas as pessoas que, diretamente ou indiretamente, fizeram parte dessa etapa decisiva em minha vida.

RESUMO

ROCHA, Jean. **Análise de componentes para viabilizar um sistema residencial automatizado com suporte para IoT**. 56f. Monografia de Conclusão de Curso (Graduação em Engenharia Elétrica). Faculdade Doctum, Juiz de Fora, 2019.

Com o passar dos anos, as pessoas tem percebido que assim como na indústria o avanço da tecnologia também pode beneficiar suas vidas e facilitar até mesmo as atividades mais simplórias do seu dia a dia. E com o desenvolvimento da tecnologia isso está se tornado cada vez mais conhecido para todos. Para se adequar a isso surgiu o conceito de Internet das Coisas (*Internet of Things* - IoT). Com esse intuito surgiu a ideia de desenvolver um sistema de automação residencial aliado ao conceito de Internet das Coisas com uma interfase acessível a todos, avaliando custo integral do trabalho. Para facilitar o desenvolvimento do protótipo será proposto uma divisão de etapas do trabalho. Essas etapas são: Pesquisa bibliográfica, Programação, Montagem do circuito e Integração e teste. Na Pesquisa Bibliográfica será feito um estudo de projetos similares, dos anos de 2012 à 2019, com o intuito de recolher informações sobre protótipos de automação residencial. A Programação será desenvolvida no sistema IDE (1.8.10) do Arduino, por ser um *software* livre. A Montagem do circuito será feita no Arduino UNO com o auxílio de sensores, por ser um *hardware* de fácil obtenção e prático de se trabalhar. Por fim será feita o *upload* da programação para o *hardware*, para que o mesmo possa executar as funções antes pré-programadas. Ao final do projeto será desenvolvido um protótipo de um sistema de automação residencial que consiga efetuar todas as funções propostas nesse trabalho.

Palavras-chave: Automação. IoT. Interfase. Custo.

ABSTRACT

Over the years, people have come to realize that just as in industry, the advancement of technology can also benefit their lives and facilitate even the simplest activities of their daily lives. With the development of technology this is becoming more and more known to everyone. To fit in with this came the concept of Internet of Things (IoT). For this purpose came the idea of developing a home automation system combined with the concept of Internet of Things with an interface accessible to all, evaluating the full. To simplify the development of the prototype will be proposed a division of steps of work. These steps are: Bibliographic Search, Programming, Circuit Assembly, and Integration and Testing. In the Bibliographic Research will be made a study of similar projects, from 2012 to 2019, in order to collect information on prototypes of home automation. Programming will be developed on Arduino IDE (1.8.10), because it is free software. The Circuit Assembly will be made in Arduino UNO with the aid of sensors, as it is an easily obtainable hardware and practical to work with. Finally, the programming will be uploaded to the hardware, so that it can perform the functions previously preprogrammed. At the end of the project will be developed a prototype of a home automation system that can perform all the functions proposed in this work.

Keywords: Automation. IoT. Interphase. Cost.

LISTA DE FIGURAS

Figura 1: Fluxograma das etapas do trabalho	16
Figura 2: Imagem representativa de AR.....	18
Figura 3: Pilares da AR	19
Figura 4: Arduino UNO R3	20
Figura 5: IDE	21
Figura 6: Sensor DYP-ME003.....	22
Figura 7: Sensor HC-SR04.....	22
Figura 8: Esquema HC-SR04.....	23
Figura 9: Sensor KY-038.....	23
Figura 10: Figura ilustrativa IoT.....	24
Figura 11: Shield Ethernet W5100	25
Figura 12: Shield Ethernet acoplado ao Arduino Mega	26
Figura 13: Blynk	27
Figura 14: Esquema de ligação clapper.....	28
Figura 15: Circuito Clapper.....	29
Figura 16: Esquemático sensor PIR.....	30
Figura 17: Circuito sensor PIR	31
Figura 18: “Pulsos” sensor PIR	32
Figura 19: Esquema de ligações HC-SR04.....	33
Figura 20: Circuito Montado HC-SR04.....	34
Figura 21: Monitor Serial HC-SR04.....	34
Figura 22: Esquemático circuito Ethernet.....	35
Figura 23: Circuito montado Ethernet.....	36
Figura 24: Programação interfase HTML	37
Figura 25: Exemplos de Possíveis Interfases.....	37
Figura 26: Circuito montado Blynk + Ethernet.....	38
Figura 27: Código Blynk	39
Figura 28: Conexão com o Blynk	39
Figura 29: Blynk Funções.....	40
Figura 30: Sonoff BASICR2 Wi-Fi DIY Smart Switch	41

LISTA DE QUADROS

Quadro 1: Quadro de preços do protótipo	17
Quadro 2: Comparação de preços	41
Quadro 3: Quadro geral de preços	42

LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things</i> – “Internet das Coisas”
AR	Automação Residencial
IDE	<i>Integrated Development Environment</i> – “Ambiente de Desenvolvimento Integrado”
PIR	<i>Passive Infrared</i> – “Infravermelho Passivo”
HTTP	<i>Hypertext Transfer Protocol</i> - "Protocolo de Transferência de Hipertexto"
HTML	<i>HyperText Markup Language</i> – “Linguagem de Marcação de Hipertexto”
BLE	<i>Bluetooth Low Energy</i>
Wi-Fi	<i>Wireless Fidelity</i> – “Fidelidade Sem Fio”
LED	Light-Emitting Diode – “Diodo Emissor de Luz”
USB	Universal Serial Bus - “Porta Universal”
IP	Internet Protocol – “Protocolo da Internet”

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	14
1.1.1 Objetivo Geral	14
1.1.2 Objetivos Específicos	14
1.1.3 Justificativa	14
2 METODOLOGIA.....	16
3 REFERENCIAL TEÓRICO	18
3.1 Automação Residencial.....	188
3.2 Arduino	19
3.3 Sensores	21
3.3.1 Sensor de presença/movimento PIR	21
3.3.2 Sensor Ultrassônico HC-SR04	22
3.3.3 Sensor de som	23
3.4 <i>Internet of Things</i> – (IoT).....	24
3.5 Shield Ethernet.....	25
3.6 Blynk.....	26
4 RESULTADO	28
4.1 Sensor de Som ("Clapper")	29
4.2 Sensor de Presença.....	30
4.3 Sensor de Distancia	32
4.4 Ethernet.....	35
4.5 Ethernet e Blynk	38
4.6 Comparações	40
4.7 Custo de Projeto.....	41

5 CONCLUSÃO.....	43
REFERÊNCIAS.....	44
ANEXOS	47

1 INTRODUÇÃO

Com o avanço da tecnologia os equipamentos que facilitam nossas vidas tornam-se cada vez mais viáveis e necessários. Hoje em dia, com celular, já se é possível controlar diversos eletrodomésticos, desde ares-condicionados, televisores até máquinas de lavar e micro-ondas.

Com o passar dos anos as pessoas têm percebido que assim como na indústria o avanço da tecnologia também pode beneficiar suas vidas e facilitar até mesmo as atividades mais simplórias do seu dia a dia (MORAIS,2017). E com o desenvolvimento da tecnologia isso está se tornando cada vez mais conhecido para todos. Para se adequar a isso surgiu o conceito de Internet das Coisas (*Internet of Things - IoT*).

Segundo Wanzeler, Fülber e Merlin (2016) o IoT tem o objetivo de interligar equipamentos eletrônicos por meio da internet, e com o uso de sensores gerar dados, analisando essas informações com o intuito de trazer algum benefício ao usuário. Nos últimos 10 anos a Automação Residencial (AR) está passando por um momento de grande ápice, devido ao surgimento de equipamentos melhores e ao avanço da internet, tanto em velocidade quanto em área de alcance da mesma (NICHELE,2010).

Com o objetivo de trazer mais conforto e facilitar a vida das pessoas foi criado a domótica. Esse termo é usado nos países da Europa para se referir à automação residencial. Ele vem da junção das palavras *domus*, que significa casa, e robótica. Além da facilidade e comodidade que a AR nos traz ela também permite que possamos avaliar a eficiência energética de nossa casa assim reduzindo nossos gastos, além da própria economia com equipamentos que se tornam cada vez mais baratos com os avanços da tecnologia nessa área (MORAIS,2017).

Mesmo com a melhoria em equipamentos e na internet, a AR ainda é uma área que necessita de muita atenção e estudo para que ela se torne cada vez mais viável e presente na vida das pessoas (BOLZANI,2004). Mesmo que cada vez mais as pessoas tenham mais acesso a informação, alavancado pelo avanço constante dos computadores pessoais e celulares, é necessário que a AR seja feita de modo que se torne acessível a todos. Por isso deve-se cada vez mais conhecer e pesquisar essa área para desenvolver um sistema automatizado disponível a todos tanto na facilidade de acessar essa interface quanto no preço.

Este trabalho tem intuito de elaborar um sistema de automação residencial que tenha uma interconectividade de dispositivos através do IoT mas sempre pensando na acessibilidade para que possa ser fácil e prático para qualquer pessoa usar, levando em conta o custo do projeto.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema de automação residencial aliado ao conceito de Internet das Coisas com uma interfase acessível a todos, avaliando o custo integral do trabalho.

1.1.2 Objetivos Específicos

- Desenvolver um protótipo que seja fácil de reproduzir e usar;
- Propor um sistema interconectado por IoT.

1.1.3 Justificativa

O avanço da tecnologia é inegável e cada vez mais a informação se torna acessível com um simples toque. Com o auxílio da internet que se torna cada vez mais presente e essencial no dia-a-dia das pessoas. Por isso faz se necessário cada vez mais pesquisar e desenvolver formas de tornar essas informações mais acessíveis a todos. Com isso em mente, surgiu a motivação de propor um sistema de automação residencial que fosse viável para todas as pessoas, que tanto tenha uma interfase que possa ser usada por pessoas com alta experiência na área quanto as que não tenham nenhuma. Além do fato que esse sistema tem o intuito de facilitar as tarefas do dia-a-dia das pessoas, como por exemplo, de acender uma lâmpada a fechar um portão da garagem, isso sempre levando em conta também o preço da implementação do mesmo.

Esse projeto também será muito engrandecedor, pois querendo ou não estamos caminhando para um futuro onde tudo será automatizando, e entender sobre essa área é algo que será um diferencial profissionalmente para quem o

realizar. Além do fato que saber sobre domótica pode ajudar muito caso tenha que reformar uma casa para um parente mais velho ou alguém que necessite de alguma acessibilidade, ou para simples conhecimentos do dia-a-dia.

2 METODOLOGIA

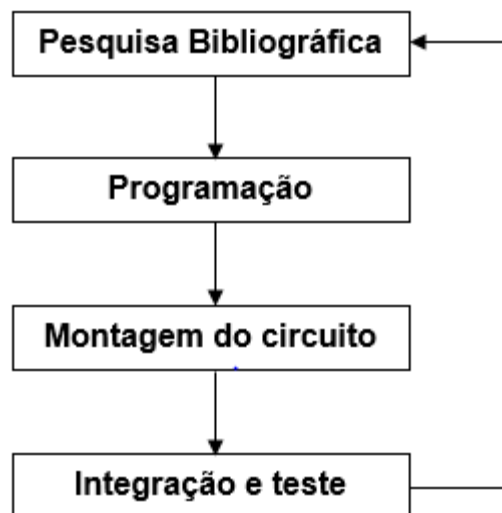
Gil (2002) afirma que: “[...] a pesquisa desenvolve-se ao longo de um processo que envolve inúmeras fases, desde a adequada formulação do problema até a satisfatória apresentação do problema.” Para que isso seja feito é necessário que seja feito filtrar de maneira cuidadosa os dados disponíveis.

Para que o estudo seja realizado é necessário um levantamento bibliográfico no intuito de conhecer, aprender, comparar e decidir as ferramentas que serão utilizadas na realização do projeto.

Começar uma pesquisa para a construção de um protótipo pode se tornar uma tarefa complicada dependendo do intuito do projeto. Então para facilitar a execução do mesmo faz-se necessário uma divisão de etapas para se ter um maior controle do progresso do mesmo.

Então para facilitar o desenvolvimento desse foi proposto essas etapas para trabalho. Essas etapas são: Pesquisa bibliográfica, Programação, Montagem do circuito e Integração e teste. Abaixo segue um figura 1 da divisão de tarefas.

Figura 1: Fluxograma das etapas do trabalho



Fonte: O Autor (2019)

Na Pesquisa Bibliográfica será feito um estudo de projetos similares, dos anos de 2015 à 2019, com o intuito de recolher informações sobre protótipos de automação residencial. A Programação será desenvolvida no sistema IDE (1.8.10) do Arduino, por ser um *software* livre. A Montagem do circuito será feita no Arduino UNO com o auxílio de sensores, por ser um *hardware* de fácil obtenção e prático de se trabalhar. Por fim será feita o *upload* da programação para o *hardware*, para que o mesmo possa executar as funções antes pré-programadas. Após esse ponto o projeto irá consistir em fazer testes e voltar nos passos anteriores caso necessário.

Para a realização desse projeto é necessário a compra de alguns equipamentos. O Quadro 1 abaixo mostra os equipamentos comprados e seus preços, sendo que os itens marcados com “*” são emprestados ou já eram possuídos pelo autor.

Quadro 1: Quadro de preços do protótipo

Equipamento	Quantidade	Preço Total
Arduino UNO R3 + Cabo USB para Arduino*	1	R\$ 54,90
Fonte de Alimentação 5V	1	R\$ 9,90
Shield Ethernet W5100*	1	R\$ 56,90
Sensor de Presença PIR	1	R\$ 11,50
Sensor de Som KY - 038	1	R\$ 12,00
Sensor de Distancia HCSR04*	1	R\$ 11,50
Módulo Relé 1 canal	1	R\$ 12,90
Módulo Relé 2 canais*	1	R\$ 11,25
Protoboard 400 pontos*	1	R\$ 8,90
Jumpers Macho/Macho	20	R\$ 10,00
Jumpers Macho/Fêmea	20	R\$ 10,00
LED's	10	R\$ 2,30
Roteador TP-LINK 3 Antenas 300mbps*	1	R\$ 122,71
Total Geral:		R\$ 334,76
Total descontando os emprestados:		R\$ 66,30

Fonte: O Autor (2019)

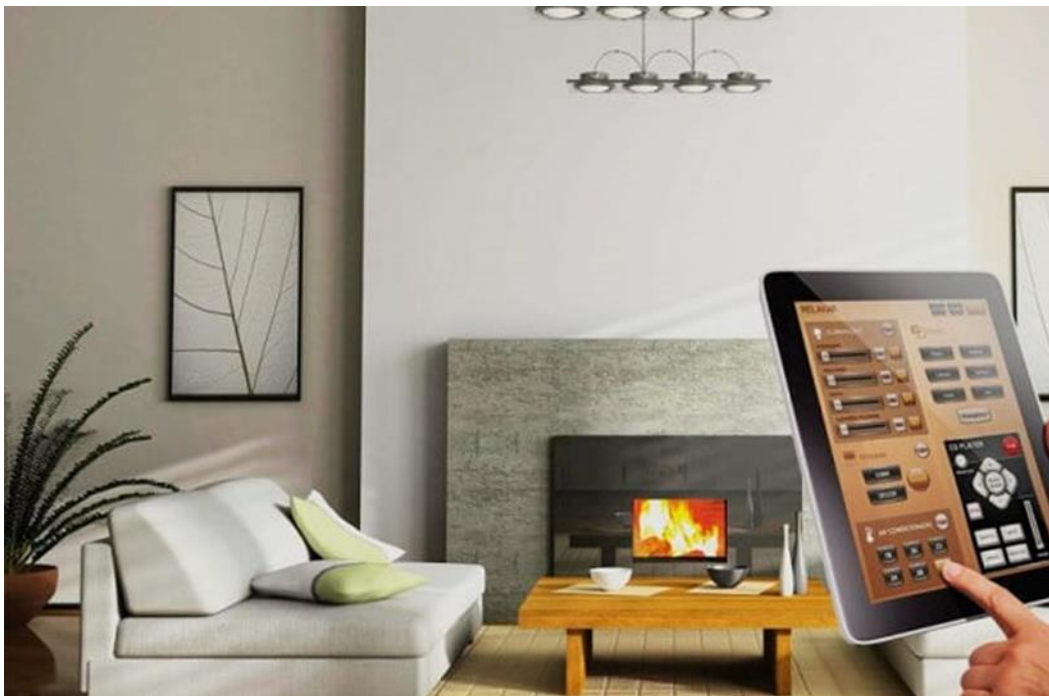
Os preços do Quadro 1 são referentes aos meses de agosto, setembro e novembro de 2019. Os mesmos não são referentes a um site ou loja específico, e os valores de frete não foram inclusos.

3 REFERENCIAL TEÓRICO

3.1 Automação Residencial

Automação Residencial (AR), domótica, casa inteligente e casa conectada são expressões que se referem a mesma coisa e tem o mesmo conceito (COSTA, 2019). A automação residencial é basicamente o uso da tecnologia para facilitar tarefas diárias e habituais de uma casa convencional, pegando essas tarefas, e com o auxílio de sensores, e as automatizando (AUTOMATICHOUSE, 2019). Abaixo a figura 2 uma representação da Automação residencial.

Figura 2: Imagem representativa de AR



Fonte: Viva Decora Pro (2018)

Já que hoje em dia as pessoas não se preocupam em apenas formas de deixarem suas casas mais seguras mas também formas de aperfeiçoar suas tarefas, de modo há demandar menos tempo e proporcionar uma sensação maior de conforto, segurança e bem-estar. Assim a domótica pode proporcionar um nível maior de conforto, comodidade e segurança.

A AR tem como objetivos o conforto, segurança, entretenimento, economia e acessibilidade aos usuários, como mostra a figura 3.

Figura 3: Pilares da AR



Fonte: Fiteck (2019)

3.2 Arduino

Criado na Itália em 2005, por 5 pesquisadores (Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis), o Arduino é uma plataforma de prototipagem eletrônica, *open-source* baseada em *hardware* e *software*. Foi concebido com o objetivo de ser um ambiente de desenvolvimento que pode ser utilizado por todos devido a sua facilidade de uso e seu preço (GOMES, 2014).

Existe uma lista infinita de possibilidades de coisas que se pode fazer com o Arduino, desde automatizar a sua casa até criar um novo brinquedo. Com isso em mente também existem diversos modelos de Arduino para se escolher, de pequenos à grandes ou até para serem usados em *wearables*. A placa que será usada vai depender muito do projeto e do número de portas que vão ser necessárias no mesmo (THOMSEN, 2014).

O termo “*Arduino*” refere-se tanto a placa física do Arduino (onde o mais comum é o UNO) quanto o sistema Arduino em geral. Esse sistema inclui também o *software* que deve ser instalado e executado no seu computador, para poder programar a placa, e os Shields periféricos que podem ser encaixados na placa (MONK, 2014). Segundo Passos (2016), o Arduino UNO já foi ultrapassado em termos tecnológicos por outras placas lançadas mais recentemente, mesmo assim, ainda hoje, é possível encontrar diversos sistemas com essa placa. Sendo assim o

Arduino UNO pode ser considerado o mais popular e versátil dos “Arduinos”. Segue abaixo uma figura 4 do Arduino UNO:

Figura 4: Arduino UNO R3

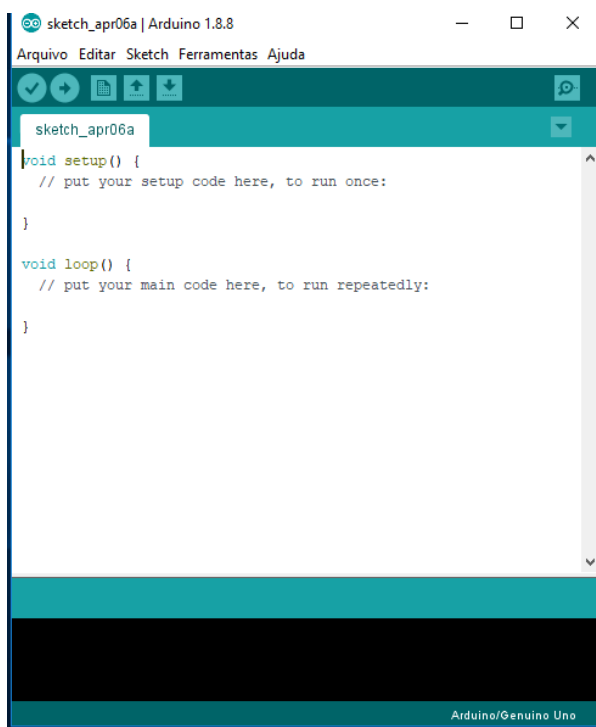


Fonte: Thompen (2014)

Segundo Souza (2013), o Arduino é constituído principalmente por um microcontrolador Atmel ATMEGA328, um dispositivo 8 bits da família AVR com arquitetura RISC (acrônimo de *Reduced Instruction Set Computer*, em português, "Computador com um conjunto reduzido de instruções") avançada e com encapsulamento DIP28. Ele possui uma memória Flash de 32 kB, sendo que 512 bytes são utilizados para o *bootloader*, 2 kB de RAM e 1 kB de EEPROM, além de pinos digitais e analógicos tanto de saída como de entrada.

Para usar o Arduino é necessário conectar o mesmo em um computador via porta USB, e programar ele em seu próprio ambiente de programação, uma IDE (*Integrated Development Environment* – “Ambiente de desenvolvimento integrado”), um *software* gratuito para baixar no próprio site oficial do Arduino. Essa IDE usa um compilador gcc (C e C++) que usa uma interface feita em Java que envia a programação escrita na mesma para o Arduino (TOGGWEILER; MARQUES, 2017). A seguir uma figura 5 da tela inicial do IDE:

Figura 5 – IDE.



Fonte: O Autor (2019)

3.3 Sensores

3.3.1 Sensor de presença/movimento PIR (*Passive Infrared* – “Infravermelho passivo”)

Os sensores de movimento são comumente usados em garagens, corredores e escadarias para acender lâmpadas sem necessidade de interruptores. Normalmente esses dispositivos são sensores infravermelhos que detectam a presença das pessoas pela fonte de calor emitida dos corpos (MORAIS, 2017). O modelo utilizado nesse projeto foi o DYP-ME003. Esse sensor consegue detectar movimentos até 7 metros de distância. Segue uma imagem (figura 6) do sensor:

Figura 6 - Sensor DYP-ME003.

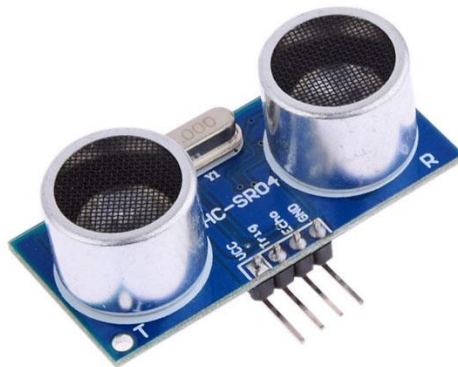


Fonte: Eletrogate (2017)

3.3.2 Sensor Ultrassônico HC-SR04

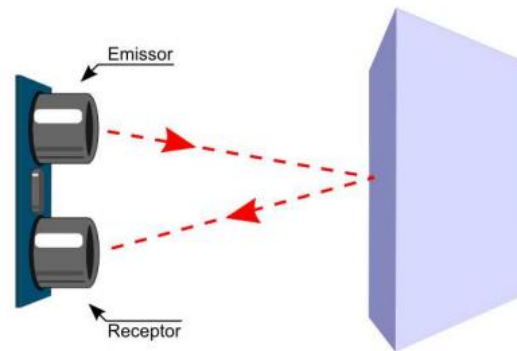
O sensor ultrassônico pode indicar a presença de um objeto além de poder dizer a distância entre ele e o objeto. Esse sensor basicamente envia uma onda sonora, inaudível para humanos, que reflete no objeto em questão e volta para o sensor. Com o sinal que volta para o sensor é possível medir a distância usando o tempo da emissão até o retorno (GOMES, 2014). Segue abaixo a figura 7 do sensor HC-SR04 e uma outra figura 8 do esquema de funcionamento desse mesmo sensor.

Figura 7 – Sensor HC-SR04.



Fonte: Filipe Flop (2019)

Figura 8 - Esquema HC-SR04.

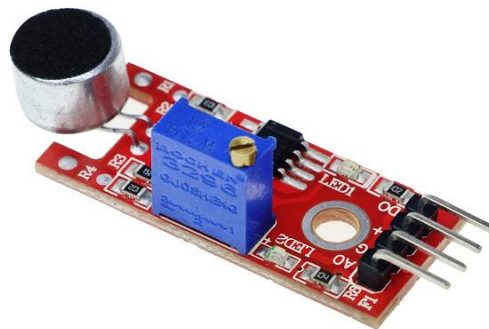


Fonte: Gomes (2014)

3.3.3 Sensor de som

O sensor detector de som KY-038 é um módulo eletrônico que tem a função de captar variações sonoras no ambiente e executar uma ação a partir dessa variação (OLIVEIRA, 2018). Esse sensor é muito usado para acender lâmpadas com palmas por conta disso é popularmente conhecido como *clapper*. Segue abaixo a figura 9 desse sensor.

Figura 9 - Sensor KY-038.

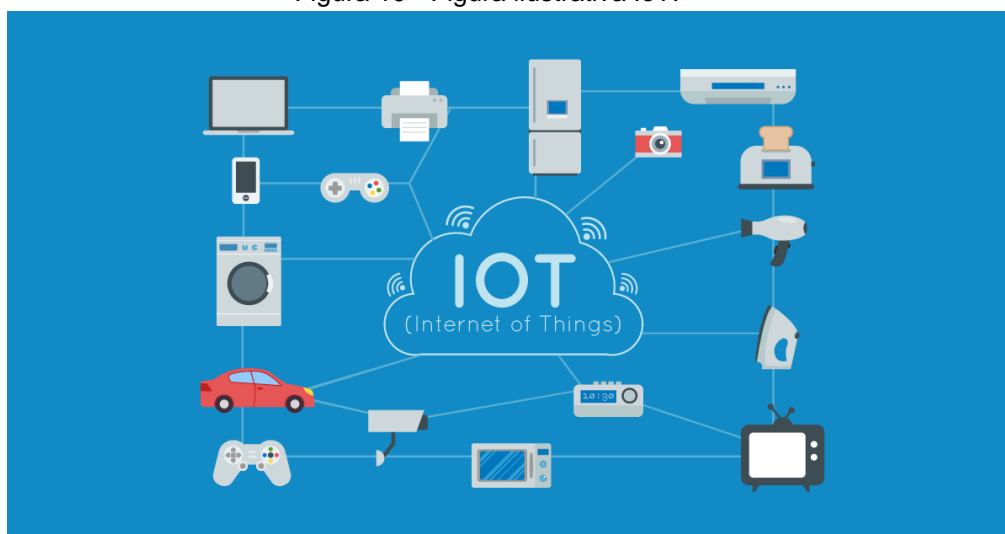


Fonte: Oliveira (2019)

3.4 Internet of Things – (IoT)

O conceito da Internet das Coisas foi proposto, por Kevin Ashton, em 1999. A ideia do IoT consiste em conectar diversos aparelhos eletrônicos como: sensores, eletroeletrônicos, celulares, etc, entre si. Com essa conexão é possível gerar um fluxo de dados entre esses equipamentos assim fazendo uma Internet das Coisas (WANZELER; FÜLBER; MERLIN, 2016). Como mostra a Figura 10.

Figura 10 - Figura ilustrativa IoT.



Fonte: Portal Lubes (2018)

O IoT representa um grande avanço para a inovação tecnológica pois com ele será possível conectar diversos equipamentos à sensores. Assim, por exemplo, podendo programar o seu ar-condicionado para ligar, à menos 5° C da temperatura do dia, quando você estiver a 3 km de casa. Mas não é só em coisas do dia-a-dia que o IoT trabalha ele também pode auxiliar em hospitais e clínicas, medindo seus batimentos e pressão, e enviando em tempo real para o sistema. Ou até mesmo na agropecuária com sensores espalhados pela plantação medindo as características do solo e clima, e calculando o melhor momento para a colheita (ALECRIM, 2017).

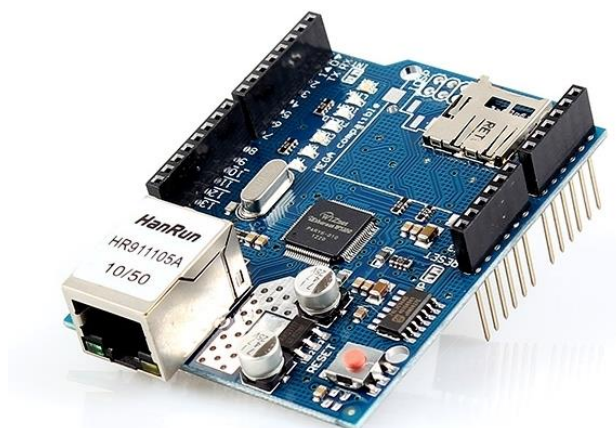
Nos últimos 10 anos o IoT não consiste mais em só em conectar equipamentos entre si mas também em conectar esses equipamentos também à internet ou web, criando assim o termo *Web of Things*. Por muitos dos aparelhos que usamos, atualmente, já terem conexão com a internet, e também devido a

interatividade desses mesmos aparelhos o conceito da Internet das Coisas já se torna algo palpável e concreto (SILVEIRA, 2015).

3.5 Shield Ethernet

Shields compatíveis com o Arduino UNO, são placas de circuito que se encaixam sobre (em cima) o mesmo de em que seus pinos ficam perfeitamente conectados com o intuito de expandir as possíveis aplicações desse microcontrolador. O *Shield Ethernet W5100* por sua vez possui dois módulos. O primeiro é um módulo para cartão de memória (micro SD) onde é possível armazenar arquivos que podem ser enviados pela rede local/internet. O segundo, e principal, é o módulo *Ethernet* que permite que o Arduino se conecte a um roteador via cabo de rede RJ45. O mesmo possui o chip Wiznet W5100 e suporta até quatro conexões de socket simultaneamente (OLIVEIRA, 2018). Abaixo segue a figura 11 do *shield ethernet* e a na figura 12 esse *shield* conectado no Arduino.

Figura 11: *Shield Ethernet W5100*



Fonte: Thomsen (2014)

Figura 12: Shield Ethernet acoplado ao Arduino Mega



Fonte: Lima, Nobre e Alencar (2015)

Com esse shield é possível conectar o Arduino na rede local ou na internet com isso é a biblioteca nativa do mesmo é possível transformar ele em um servidor. Um servidor consiste em um *software* dentro de um computador central. E por meio de uma rede particular ou da internet o mesmo pode receber e/ou transmitir para “clientes”. A relação servidor/cliente funciona basicamente com o servidor recebendo, processando e respondendo dados enviados pelo cliente através de algum protocolo de comunicação. O principal exemplo dessa relação servidor/cliente é os computadores que hospedam sites atuando como servidor e as pessoas navegando nesses sites como clientes. Nessa relação se utiliza principalmente o protocolo HTTP (*Hypertext Transfer Protocol* - "Protocolo de Transferência de Hipertexto") que é responsável pelo tratamento dos pedidos e respostas entre cliente e servidor. Esses tipos de servidores são conhecidos como Servidores *Web*. Para a criação dessas páginas na internet é utilizada a linguagem HTML (*HyperText Markup Language* - “Linguagem de Marcação de Hipertexto”) (MADEIRA, 2017). Para criação de servidores web com o *shield* ethernet e o arduino é necessário também a linguagem HTML além do C++.

3.6 Blynk

Segundo o site oficial do Blynk, ele é uma plataforma IoT independente de hardware com aplicativos móveis de marca branca, nuvens privadas, gerenciamento

de dispositivos, análise de dados e aprendizado de máquina. Ele pode se conectar com diversos tipos de placas como por exemplo. Arduino, ESP8266, ESP32, NodeMCU, Particle Photon, Raspberry Pi e etc. Usando o smartphone é possível se conectar via Internet, Bluetooth e BLE (*Bluetooth Low Energy*) (PLAY STORE, 2019).

Com o Blynk os problemas relacionados a memória, comunicação entre as placas e a necessidade de usar o módulo ESP8266 como *webserver* tendo que continuamente manter a conexão com a placa Arduino, todas essas questões são contornadas utilizando esse aplicativo (ROBOCORE, 2018).

Nesse aplicativo é possível inserir botões e displays de forma simples e fácil pelo modo *drag-and-drop* (“arraste e solte”) que consiste em literalmente arrastar o *widget* e colocar ele no lugar desejado. Ele também simplifica bastante o código/programação. Na figura 13 é possível ver a simplicidade do aplicativo.

Figura 13: Blynk – Da esquerda para a direita – “FACIL CONFIGURAÇÃO” – “FUNCIONA COM TUDO” – “FERRAMENTAS PARA TODAS NECESIDADES e ainda mais logo!”



Fonte: Play Store (2019)

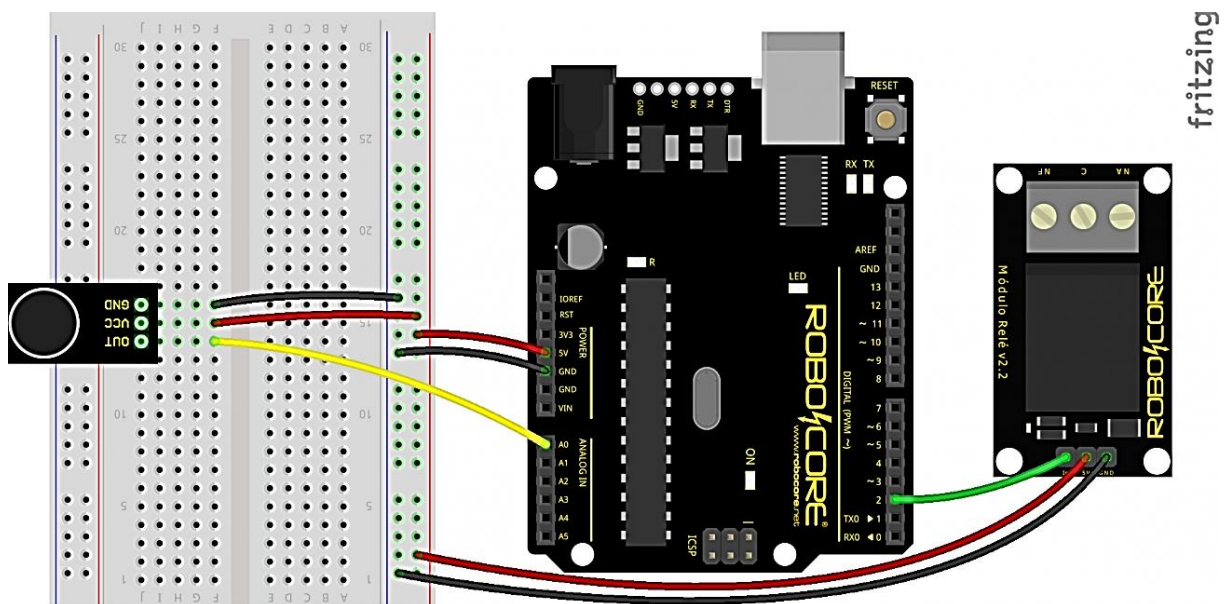
4 RESULTADO

Nesse capítulo será apresentado os resultados dos protótipos, eles serão divididos em tópicos cada um com o respectivo teste do sensor. Em cada tópico será abordado as vantagens e desvantagens do mesmo. Todos os testes foram feitos utilizando relés como um representante da carga, seja ela podendo ser uma lâmpada, ventilador, ar-condicionado e etc. Todas as programações usadas como base estarão nos apêndices.

4.1 Sensor de Som (“Clapper”)

O teste do sensor de som, foi pensando na ideia de acender lâmpadas com o bater de palmas. A programação que foi usada como base foi tirada do site da Robocore. O computador foi usado como fonte mas poderia ter sido usado também uma fonte 5V. A figura 14 abaixo mostra o esquemático de ligações.

Figura 14: Esquema de ligação *clapper*



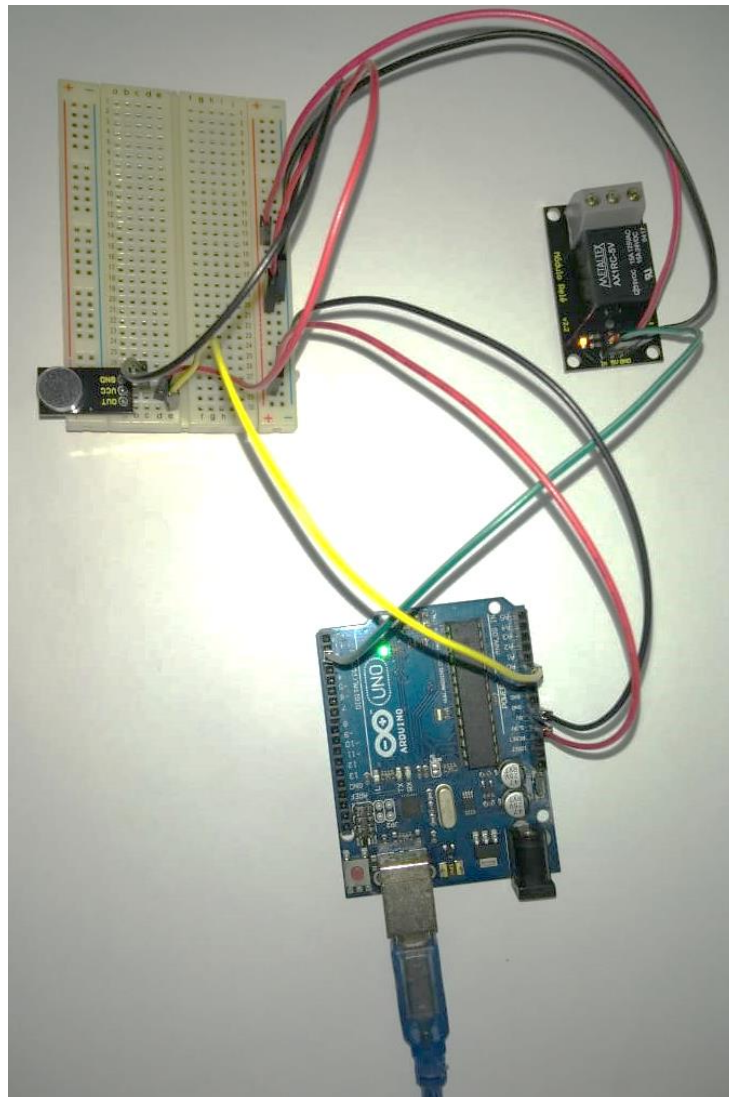
Fonte: Robocore (2018)

O sensor sonoro é um sensor bem interessante, ele pode ser usado deste modo para acender lâmpadas com palmas mas também pode ser usado para ler qualquer tipo de som, e executar uma ação baseado nisso. Mas ele deve sempre ser programado em relação ao tipo (amplitude, frequência e tempo) de som detectado,

assim ele se tornaria inútil na questão de detectar uma invasão na casa, pois não teria como prever o som que essa invasão teria.

Com relação a essa programação, esse sensor seria vantajoso em relação a um programação com o módulo WiFi ou ethernet, pois não teria a necessidade da internet ou de um smartphone para acender a lâmpada. Mas seria totalmente inútil comparado com um sensor de presença/movimento, pois o mesmo acenderia a lâmpada sem a necessidade de palmas, apenas com a aproximação da pessoa. Abaixo segue a figura 15 referente a montagem do circuito.

Figura 15: Circuito *Clapper*



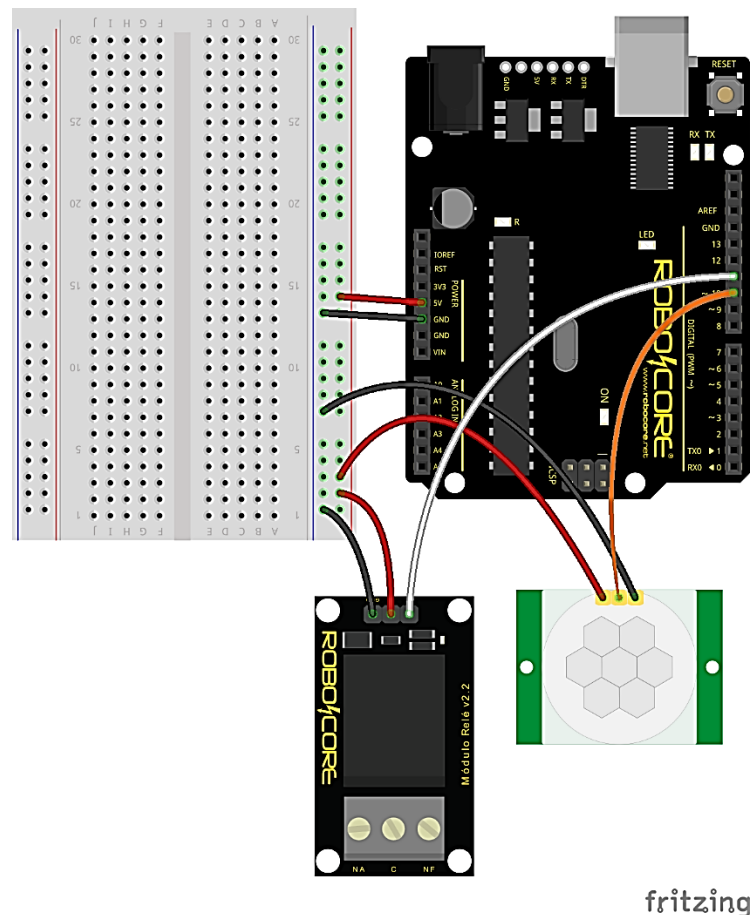
Fonte: O Autor (2019)

Os equipamentos usados nesse protótipo foram: Arduino UNO R3, Fonte 5V, Relé, Sensor de som, *protoboard* e *jumpers*. O preço só desses equipamentos, levando em conta os preços apresentados na metodologia, ficaria em: R\$ 108,60

4.2 Sensor de Presença

O sensor PIR foi usado com o objetivo de acender uma lâmpada caso haja movimento e caso não haja acender outra. Novamente a programação base foi tirada do site da Robocore e o computador foi usado como fonte. A figura 16 a seguir mostra o esquemático de ligações desse teste.

Figura 16: Esquemático sensor PIR



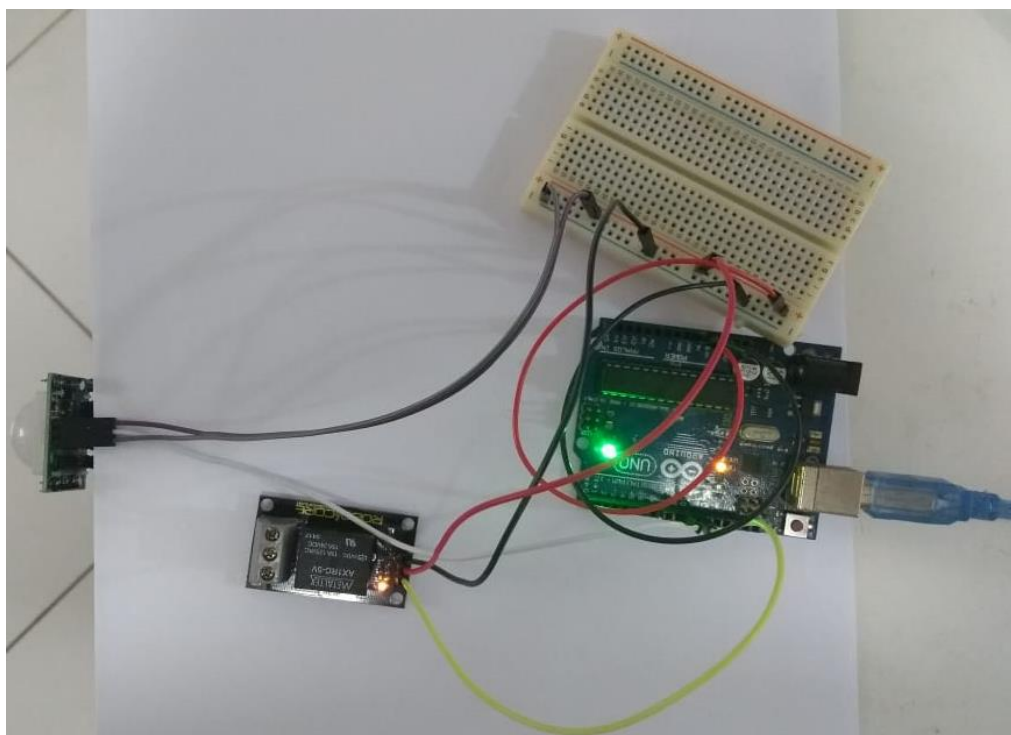
Fonte: Robocore 2018

O sensor de movimento é um dos mais úteis, por exemplo, no caso de um corredor muito longo ao invés de usar um interruptor, seria possível colocar alguns sensores de presença no caminho e com a proximidade das pessoas as lâmpadas seriam acessas. Esse sensor poderia ser usado também como sistema de alarme,

no qual a pessoa iria inserir um interruptor/botão, para ligar a noite em um local no qual a mesma não fosse transitar, e caso houvesse um movimento um alarme seria acionado. Também poderia ser programando para que em uma certa hora esse sensor ligasse automaticamente.

Comparando esse sensor com outros é possível observar que ele é muito útil e não teria a necessidade de estar conectado à rede para funcionar. Considerando que nos dias de hoje seja basicamente impossível encontrar alguém sem celular e sem internet, sensor perderia um pouco a utilidade já que com apenas um apertar na *touchscreen* seria possível acender a lâmpada da garagem e apaga-la dentro de casa. A figura 17 mostra a montagem do circuito anterior.

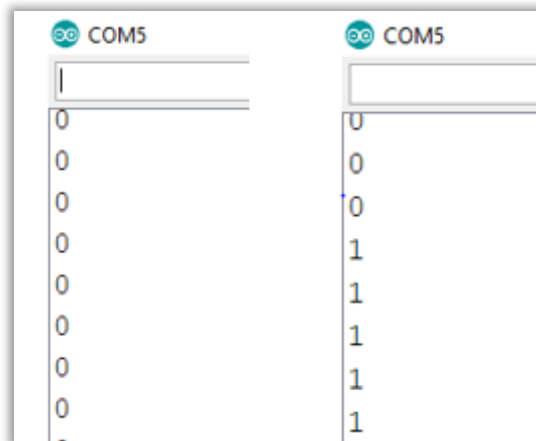
Figura 17: Circuito sensor PIR



Fonte: O Autor (2019)

O sensor PIR funciona com “pulsos digitais” ou seja 0 ou 1, onde 0 seria sem movimento e 1 quando houvesse movimento. A figura 18 a seguir exemplifica a atuação desse fato.

Figura 18: “Pulsos” sensor PIR



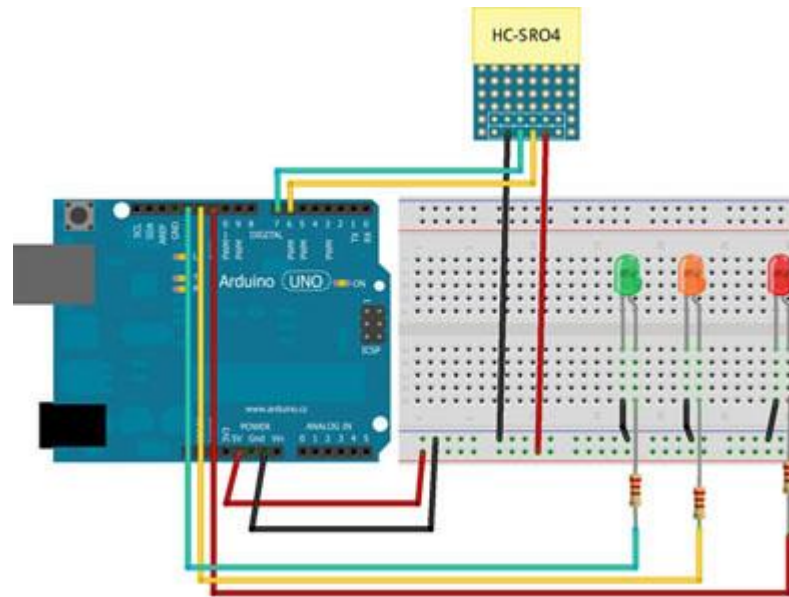
Fonte: O Autor (2019)

Os equipamentos usados nesse projeto foram: Arduino UNO R3, Fonte 5V, Relé, Sensor PIR, *protoboard* e *jumpers*. O preço só desses equipamentos ficaria em: R\$ 108,10. Levando em consideração os preços apresentados na metodologia.

4.3 Sensor de Distância

Esse sensor foi usado de modo a representar um sensor de estacionamento de três distâncias: perto (vermelho), média (amarelo) e longe (verde). As distâncias usadas para determinar o que seria “perto e longe” foram meramente ilustrativas, perto seria $<10\text{cm}$, media seria $\geq 10\text{cm}$ e $\leq 20\text{cm}$ e longe seria $>20\text{cm}$. A programação base para esse teste foi tirada do site Como Fazer as Coisas. Na figura 19 é possível ver o esquema de ligações desse projeto.

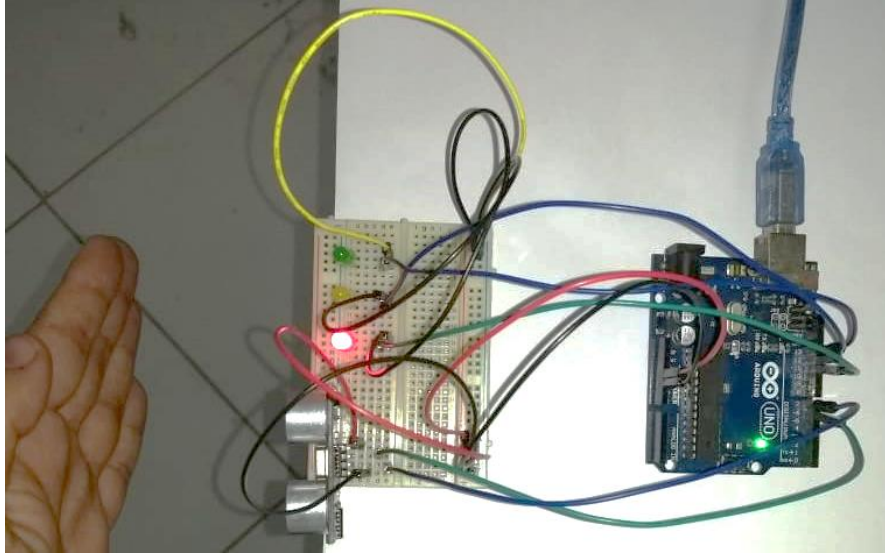
Figura 19: Esquema de ligações HC-SR04



Fonte: Como Fazer as Coisas (2017)

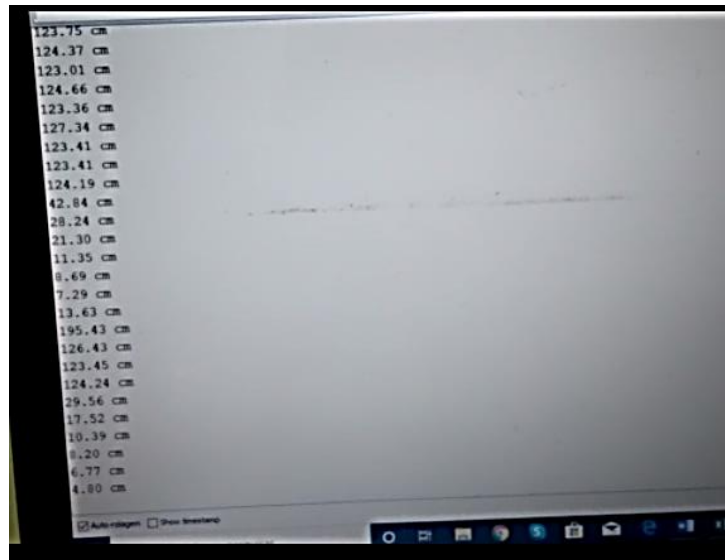
Esse sensor seria um bom substituto para o sensor de presença já que poderia ser programado para quando alguém estivesse a uma certa distância dele a lâmpada acender. Mas esse sensor não captaria apenas a movimentação das pessoas, qualquer coisa que passasse em sua frente acionaria esse sensor. E também devido a ele ser um sensor ultrassônico com uma entrada e uma saída ele costuma não ser tão preciso em suas medições. Em compensação, mesmo com seus defeitos, ele é um dos sensores com maior abrangência do que se pode fazer com o mesmo. Pode ser feito desde um sensor de estacionamento, a uma régua digital e até mesmo um robô aspirador de pó. A seguir segue a figura 20 com o circuito montado do protótipo proposto e a figura 21 com as medições no monitor serial do Arduino.

Figura 20: Circuito Montado HC-SR04



Fonte: O Autor (2019)

Figura 21: Monitor Serial HC-SR04



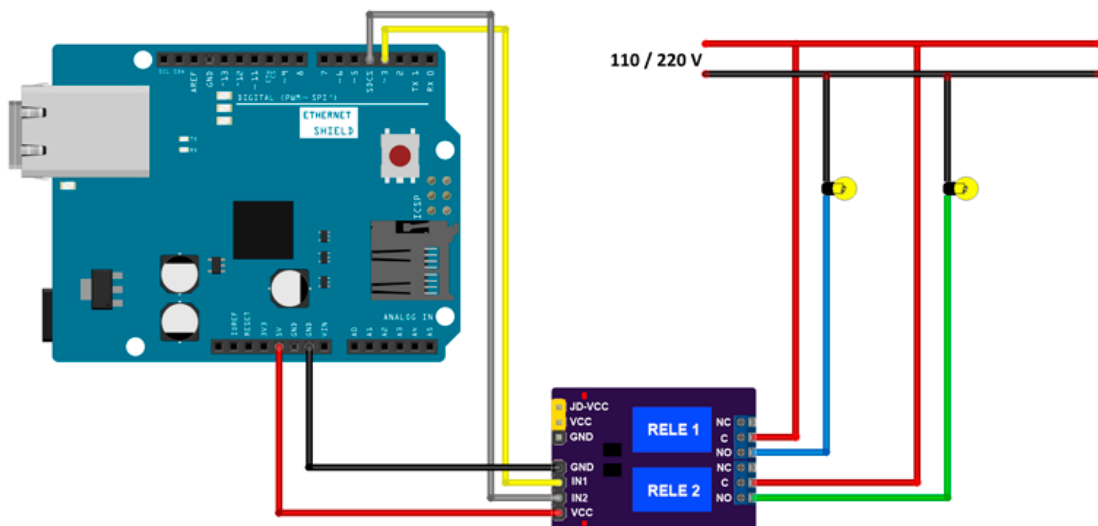
Fonte: O Autor (2019)

Nesse projeto foi utilizado: Arduino UNO R3, Fonte 5V, LED's (*light-emitting diode* – “diodo emissor de luz”), Sensor de distância HC-SR04, *proto-board* e *jumpers*. O preço só desses equipamentos ficaria em: R\$ 97,50. Levando em consideração os preços apresentados na metodologia.

4.4 Ethernet

No teste do Ethernet a ideia foi usar o mesmo para criar uma página na internet na qual se poderia controlar a interruptores. Nessa etapa foi usado um Arduino Leonardo ao invés de um Arduino UNO devido a problemas na mesma. As duas placas não apresentam muitas diferenças em relação a preço e operação, sua maior diferença seria em relação a sua conexão com o computador que no caso do UNO seria via USB (*Universal Serial Bus* - “Porta Universal”) tipo B e o Leonardo usa o cabo micro USB. A programação usada como base foi tirada do livro: Programação com Arduino: Começando com Sketches do Simon Monk. Abaixo segue o esquemático das ligações na figura 22.

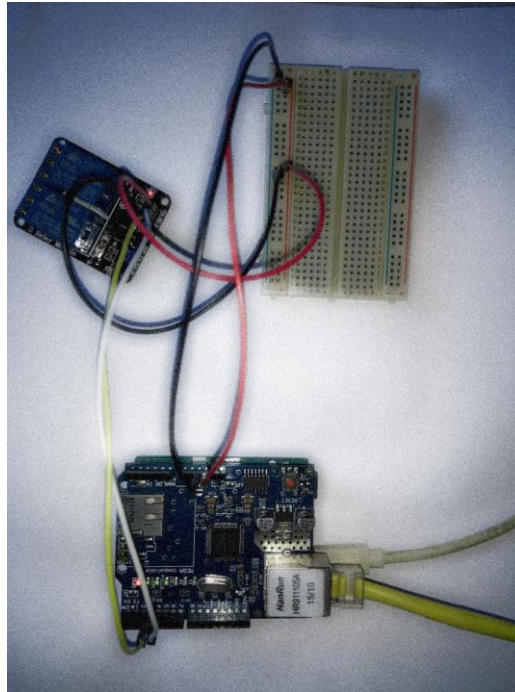
Figura 22: Esquemático circuito Ethernet



Fonte: Thomsen (2015)

O circuito, por ser demonstrativo so foi montado até a parte do rele ou seja a parte da lâmpadas não foi inserida. A seguir a figura 23 com a montagem do circuito.

Figura 23: Circuito montado Ethernet



Fonte: O Autor (2019)

O *shield* Ethernet facilita muito as conexões com o Arduino, já que ele simplesmente se encaixa sobre o mesmo. A vantagem de se usar o Ethernet como conexão é que o mesmo não necessita de internet, apenas se conectando no roteador que ele está ligado e usando o IP (*Internet Protocol* – “Protocolo da Internet”) selecionado e possível acessar a página que foi criada pelo mesmo. E sua maior desvantagem seria ter que ficar conectado diretamente ao roteador.

Se investido um certo tempo e conhecimento na parte HTML da criação da página e possível fazer diversas interfaces possíveis. A seguir segue um *print screen* da programação na figura 24 e alguns exemplos de interfaces na figura 25.

Figura 24: Programação interfase HTML

```

sketch_10_02 | Arduino 1.8.10
Arquivo Editar Sketch Ferramentas Ajuda

sketch_10_02

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();

// send the body
client.println("<html><body>");
client.println("<h1>Output Pins</h1>");
client.println("<form method='GET'>");
setValuesFromParams();
setPinStates();
for (int i = 0; i < numPins; i++)
{
    writeHTMLforPin(client, i);
}
client.println("<input type='submit' value='Update'>");
client.println("</form>");
client.println("</body></html>");

client.stop();

```

Fonte: O Autor (2019)

Figura 25: Exemplos de Possíveis Interfases



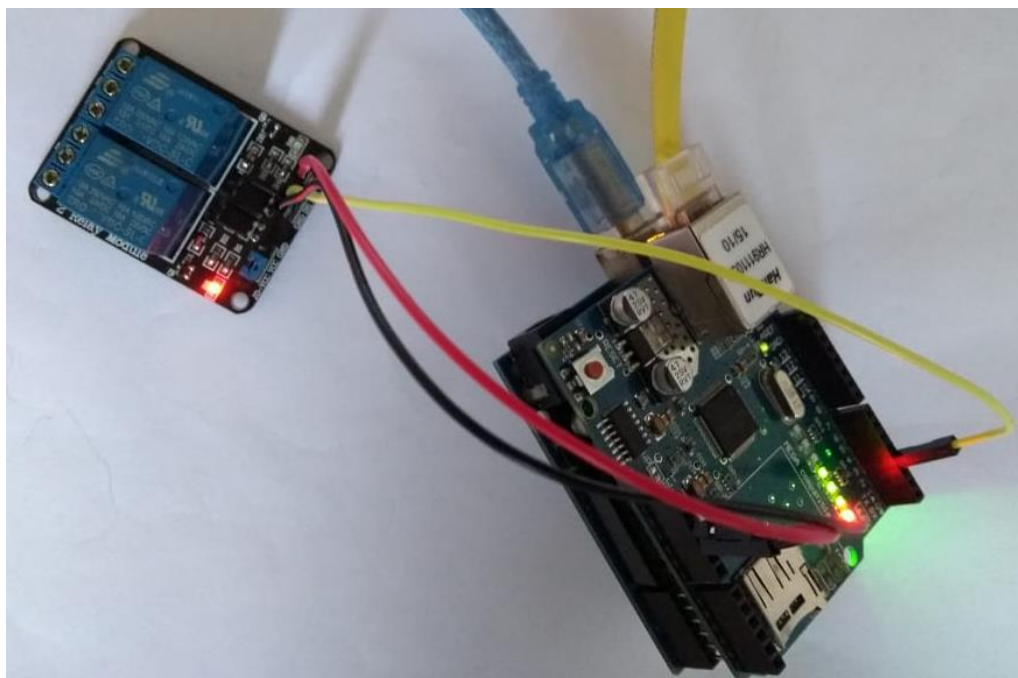
Fonte: Da esquerda pra direita: Monk (2013); Robocore (2016); Thomsen (2015).

Nesse projeto foi utilizado: Arduino Leonardo, Fonte 5V, Relé, Shield Ethernet, *proto-board* e *jumpers*. O preço só desses equipamentos ficaria em: R\$ 151,85 sem o roteador e com o valor do roteador ficaria: R\$ 274,56. Levando em consideração os preços apresentados na metodologia

4.5 Ethernet e Blynk

Após o teste de vários sensores/programações foi descoberto uma plataforma que diminuiria consideravelmente a quantidade de linhas de código e permitiria que tudo fosse controlado a distância por meio de um simples aplicativo: Blynk. Então nessa etapa foi feito exatamente os passos do item (4.4) anterior só que usando o aplicativo Blynk. Isso faz com que a programação fique bem mais básica e permite que tudo seja controlado pelo aplicativo Blynk. O esquema do circuito usado foi o mesmo do item anterior ou seja a figura 24. Nesse projeto foi usado novamente o Arduino UNO. Mas abaixo segue a figura 26 do circuito montado.

Figura 26: Circuito montado Blynk + Ethernet

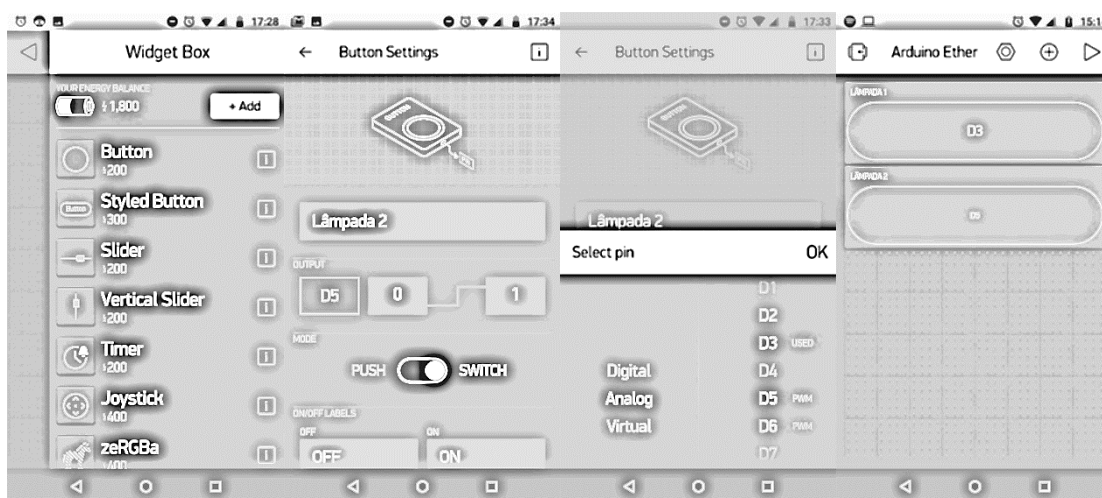


Fonte: O Autor (2019)

A Programação base usada nesse projeto foi a que vem nos exemplos quando a biblioteca do Blynk é instalada. Ela é um código muito simples onde

O Blynk permite adicionar vários “*widgets*” que pode ser programados com alguns simples toques. Ele permite que a pessoa escolha a porta que será usada no programa e avisa se a mesma estiver ocupada. Ele também permite que seja alterada a função do botão seja “PUSH” (só ficará ligado enquanto o botão estiver sendo pressionado) e “SWITCH” (varia entre ligado e desligado com o pressionar do botão). Todos os *widgets* adicionados custam uma certa “energia”, em todos projetos (2000), para serem adicionados mas caso sejam retirados a energia retorna, caso necessário também é possível comprar essa “energia”. A figura 29 abaixo mostra algumas dessas funções apresentadas. As cores foram um pouco alteradas para melhor visualização.

Figura 29: Blynk Funções



Fonte: O Autor (2019)

Utilizando o Blynk também é possível adicionar displays para medições, notificações. Nesse projeto foram usados os mesmos equipamento do item 4.4, então os preços foram os mesmo: R\$ 151,85 sem o roteador e com o valor do roteador ficaria: R\$ 274,56.

4.6 Comparações

Após todos os experimentos feitos nesse trabalho foi realizado uma comparação dos preços. No quadro com os preços também foi adicionado o Sonoff WiFi *Smart Switch* BASICR2. Segundo a página oficial do Sonoff, ele é um

dispositivo que possui a habilidade de transformar “ferramentas” do dia-a-dia em “dispositivos inteligentes”. Fazendo com que esses possam ser controlados por um aplicativo (eWeLink) de celular de qualquer lugar. A seguir segue o quadro 2 com a comparação de preços.

Quadro 2: Comparação de preços

Protótipo	Preço
Sensor de Som	R\$ 108,60
Sensor PIR	R\$ 108,10
Sensor de Distância	R\$ 97,50
Ethernet	R\$ 274,56
Ethernet s/ roteador	R\$ 151,85
Ethernet + Blynk	R\$ 274,56
Ethernet + Blynk s/ roteador	R\$ 151,85
Sonoff	R\$ 36,90

Fonte: O Autor (2019)

Como é possível observar nesse quadro o Sonoff tem um preço bem mais acessível. Embora o mesmo não tenha todas as funcionalidades de todos os sensores. A montagem se daria conectando as saídas do cabo do equipamento e continuando essas saídas depois até a tomada. O funcionamento seria basicamente o do “Ethernet + Blynk” mas sem a necessidade de uma conexão direta com o roteador. Isso tudo com um preço menor do que o do Arduino UNO sozinho. A seguir segue a figura 30 com uma imagem do Sonoff.

Figura 30: Sonoff BASICR2 Wi-Fi DIY Smart Switch



Fonte: Site oficial Sonoff (2019)

4.7 Custo de Projeto

O protótipo final (Ethernet + Blynk) custaria cerca de R\$ 151,85 para ser reproduzido caso usado exatamente as mesmas peças usadas no mesmo, considerando os preços apresentados na metodologia. Embora o mesmo tenha custado, para o autor, apenas R\$ 10,00 por já possuir alguns dos equipamentos ou ter conseguido emprestado. Abaixo segue o quadro 3 com todos os valores gastos no trabalho inteiro.

Quadro 3: Quadro geral de preços

Equipamento	Preço Total	Valor Gasto
Arduino UNO R3 + Cabo USB para Arduino	R\$ 54,90	R\$ -
Fonte de Alimentação 5V	R\$ 9,90	R\$ 9,90
Shield Ethernet W5100	R\$ 56,90	R\$ -
Sensor de Presença PIR	R\$ 11,50	R\$ 11,50
Sensor de Som KY - 038	R\$ 12,00	R\$ 12,00
Sensor de Distancia HCSR04	R\$ 11,50	R\$ -
Módulo Relé 1 canal	R\$ 12,90	R\$ 12,90
Módulo Relé 2 canais*	R\$ 11,25	R\$ -
Protoboard 400 pontos*	R\$ 8,90	R\$ -
Jumpers Macho/Macho	R\$ 10,00	R\$ 10,00
Jumpers Macho/Fêmea	R\$ 10,00	R\$ 10,00
LED's	R\$ 2,30	R\$ 2,30
Roteador TP-LINK 3 Antenas 300mbps*	R\$ 122,71	R\$ -
Protótipo	Preço Total	Valor Gasto
Sensor de Som	R\$ 108,60	R\$ 44,80
Sensor PIR	R\$ 108,10	R\$ 44,30
Sensor de Distância	R\$ 97,50	R\$ 12,30
Ethernet	R\$ 274,56	R\$ 10,00
Ethernet + Blynk	R\$ 274,56	R\$ 10,00

Fonte: O Autor (2019)

5 CONCLUSÃO

Com o aumento do estudo de novas tecnologia e a baixa eminente da mesma, além do maior acesso a *smartphones*, cada vez mais a domótica vem se tornando viável. Com isso o seu estudo e teste se faz mais possível e necessário.

Com isso em mente esse trabalho buscou apresentar o estudo de vários sensores, além de algumas de suas vantagens e desvantagens, e desenvolver um protótipo que fosse prático, simples e atendesse as necessidades propostas no mesmo. Durante o estudo foi encontrado o aplicativo Blynk. Essa plataforma permitiria diminuir a programação consideravelmente, além de facilitar o controle da mesma, já que o mesmo usa um método de *drag-and-drop*. Esse aplicativo também permite a criação, e distribuição, do seu próprio aplicativo. O mesmo necessita do pagamento de uma mensalidade, mas transforma o projeto criado no Blynk em um aplicativo próprio de forma simples.

Mesmo com o resultado “negativo” em relação ao preço, sendo que o dispositivo já existente (Sonoff) executa algumas das funções do protótipo por um valor mais acessível. Embora o Sonoff seja um produto já construído, ou seja o mesmo não permite que você o “edite”. E utilizando o Arduino, e o Blynk, é possível criar o seu próprio dispositivo com as funções que desejar, mesmo não tendo um conhecimento avançado em programação já que muitos projetos prontos podem ser encontrados na internet.

Então ao final do projeto conclui-se que esse estudo foi satisfatório, a cerca de entender os novos auxílios que a domótica pode nos dar, como o próprio programa Blynk que auxilia na criação de um sistema IoT.

Como sugestão de trabalhos futuros, propõe-se o uso do ESP8266 ESP-01 no local do Arduino e do Ethernet, já que o mesmo pode funcionar sem a necessidade de um microcontrolador. Ele executaria as mesmas funções do ethernet sem a necessidade de um cabo conectado diretamente ao roteador, já que ele é um dispositivo Wi-Fi. Também fica a ideia de testar a plataforma Twilio, que permite, por meio da linguagem de programação python, o controle do Arduino pelo Whatsapp. E, por fim, fica também a ideia de criar um banco de dados que com o auxílio de uma I.A. (Inteligência Artificial), aprenderia os horários em que o “cliente” ligaria os equipamentos.

REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, E. **O que é Internet das Coisas (Internet of Things)?**. Disponível em: <https://www.infowester.com/iot.php> Acessado em 06/04/2019.

AUTOMATIC HOUSE. **Automação Residencial como auxílio às necessidades do lar**. Disponível em: <https://www.automatichouse.com.br/automacao-residencial/o-que-e-automacao-residencial> Acessado em 03/11/2019.

BLYNK. **Blynk library**. 2019. Disponível em: <https://github.com/blynkkk/blynk-library/releases/tag/v0.6.1> Acessado em: 08/11/2019.

BOLZANI, C. A. M.; NETTO, M. L. **Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes: uma introdução aos sistemas domóticos**. 2004.

COMO FAZER AS COISAS. **Arduino com sensor de obstáculos ultrassônico HC-SRO4**. Disponível em: <http://www.comofazerascoisas.com.br/arduino-com-sensor-de-obstaculos-ultrasonico-HC-SRO4.html> Acessado em: 03/11/2019.

COSTA, A. **Automação residencial: tudo o que você precisa saber a respeito**. Disponível em: <https://www.tuacasa.com.br/automacao-residencial/> Acessado em 03/11/2019.

FILIFE FLOP. **Sensor de Distância Ultrassônico HC-SR04**. Disponível em: <https://www.filipeflop.com/produto/sensor-de-distancia-ultrasonico-hc-sr04/> Acessado em 06/04/2019.

FITECK. **Automação Residencial**. Disponível em: <http://www.fiteck.com.br/automacao-residencial/> Acessado em 19/06/2019

GIL A. C. **Como elaborar projetos de pesquisa**. 4.ed.São Paulo: Atlas,2004

GOMES, M. **Conceito, referência e programação básicas com arduino**. Disponível em: https://issuu.com/marciogomes60/docs/conceitos__refer__ncias_e_programa__PortoAlegre – RS: Editora SMED 2014. Acessado em 06/04/2019.

LIMA, E. M. S.; NOBRE, A. Y. M.; ALENCAR, R. A. E. **Automação Residencial de baixo custo com Arduino Mega e Ethernet Shield**. 2015.

MADEIRA, D. **Shield Ethernet W5100 – Criando um Servidor Web com Arduino**. Disponível em: <https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web/> Acessado em 06/10/2019.

MARQUES, L. D. S.; TOGGWEILER, J. G. **Automação residencial para conservação e eficiência energética por meio de técnicas de inteligência artificial**. 2017.

MONK, S. **Programação com Arduino**: Começando com *Sketches*. Tradução de Anatólio Laschuk. 1ª ed. Porto Alegre: Bookman, 2013.

MONK, S. **Programação com Arduino II**: Passos Avançados com *Sketches*. Tradução de Anatólio Laschuk. 1ª ed. Porto Alegre: Bookman, 2015.

MORAIS, H. S. **Automação residencial: sistema integrado de uma residência inteligente utilizando controlador lógico programável**. 2017.

NICHELE, D. B. **Automação residencial: um grande auxílio para idosos e deficientes**. Trabalho de conclusão de Curso em engenharia elétrica. Universidade São Francisco. Itatiba/São Paulo, 2010.

OLIVEIRA, E. **Como usar o Arduino – Sensor (Detector) de som – KY-038**. Disponível em: <http://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-detector-de-som-ky-038/> Acessado em 06/04/2019.

OLIVEIRA, E. **Como usar com Arduino – Ethernet Shield W5100 (Web server)**. Disponível em: <https://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-ethernet-shield-w5100-via-web-server/> Acessado em 04/11/2019.

PASSOS, M. A. F. F. **Controle domótico via smartphone e arduino**. 2016.

PLAY STORE. **Blynk - IoT for Arduino, ESP8266/32, Raspberry Pi**. Disponível em: https://play.google.com/store/apps/details?id=cc.blynk&hl=pt_BR Acessado em 05/11/2019.

PORTAL LUBES. **Internet das Coisas – Rivais ARM e Intel firmam acordo**. Disponível em: <http://portallubes.com.br/2018/10/internet-das-coisas/> Acessado em 06/04/2019.

ROBOCORE. **Acionando uma Lâmpada pela rede Ethernet – Robocore**. Disponível em: <https://www.robocore.net/tutorials/acionando-uma-lampada-pela-rede-ethernet> Acessado em 27/10/2019.

ROBOCORE. **Kit Avançado p/ Arduino - 9. Projeto Clapper**. Disponível em: <https://www.robocore.net/tutorials/kit-avancado-para-arduino-projeto-clapper> Acessado em 06/10/2019.

ROBOCORE. **Kit Avançado p/ Arduino - 10. Sensor PIR**. Disponível em: <https://www.robocore.net/tutorials/kit-avancado-para-arduino-sensor-pir> Acessado em 06/10/2019.

ROBOCORE. **Kit Avançado p/ Arduino - 19. Hello Blynk®**. Disponível em: <https://www.robocore.net/tutorials/kit-avancado-para-arduino-hello-blynk> Acessado em 06/10/2019.

SILVERA, C. E. R. **Proposta de automatização de unidades de informação a partir da interconectividade da Internet das Coisas**. 2015.

SOUZA, F. **Arduino UNO**. Disponível em: <https://www.embarcados.com.br/arduino-uno/> Acessado em 06/04/2019.

THOMPEN, A. **O que é Arduino?**. Disponível em: <https://www.filipeflop.com/blog/o-que-e-arduino/>. Acessado em 06/04/2019.

THOMPEN, A. **Como comunicar com o Arduino Ethernet Shield W5100**. Disponível em: <https://www.filipeflop.com/blog/tutorial-ethernet-shield-w5100/> Acessado em: 04/11/2019.

THOMPEN, A. **Automação Residencial com Arduino: acenda lâmpadas pela internet**. Disponível em: <https://www.filipeflop.com/blog/automacao-residencial-com-arduino-acenda-lampadas-pela-internet/> Acessado em: 28/10/2019.

VIVA DECORA PRO. **Casa automatizada? Realize o sonho do seu cliente com a automação residencial**. Disponível em: <https://www.vivadecora.com.br/pro/tecnologia/automacao-residencial/> Acessado em 19/06/2019.

ULTRASONIC. **Ultrasonic HC-SR04 library**. 2015. Disponível em: <https://github.com/filipeflop/Ultrasonic> Acessado em: 08/11/2019.

VITAL, V. **Automação residencial: Sensor de presença com Arduino**. Disponível em: <http://blog.eletragate.com/automacao-residencial-sensor-de-presenca-com-arduino/> Acessado em 06/04/2019.

WANZELER, T.; FULBER, H.; MERLIN, B. **Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT)**. XXXIV Simpósio Brasileiro de Telecomunicações. Santarém, PA, p. 40-44, 2016.

ANEXOS

Anexos 1 – Programação base usada no projeto do sensor de som.

Créditos da programação: Robocore. **Kit Avançado p/ Arduino - 9. Projeto Clapper**. Disponível em: <https://www.robocore.net/tutorials/kit-avancado-para-arduino-projeto-clapper>

```

const int pino_microfone = A0; // pino onde o potenciometro está conectado
int leitura = 0; // variável para armazenar o valor lido pelo ADC

const int pino_rele = 2; // pino onde o rele está conectado
int estado_rele = 0; // variável para armazenar o estado do rele

float amplitude = 0.0; // armazenará o valor de pico a pico da onda
unsigned int valor_max = 0; // armazenará o valor máximo lido pelo sensor
unsigned int valor_min = 1024; // armazenará o valor mínimo lido pelo sensor
float valor_limite = 4.5; // valor mínimo para considerar uma palma (0.0 - 5.0)
unsigned long inicio_amostragem; // armazena o instante que começou a amostragem
const unsigned long tempo_amostragem = 250; // tempo de amostragem em ms

int palmas = 0; // contador de palmas
unsigned long tempo_palmas; // armazena o instante que começou a amostragem
const unsigned long tempo_palmas_max = 500; // período de amostragem de palmas

void setup() {
  // Inicia e configura a Serial
  Serial.begin(9600); // 9600bps

  pinMode(pino_microfone, INPUT); // configura o pino com potenciometro como entrada
  pinMode(pino_rele, OUTPUT); // configura o pino com o rele como saída
}

void loop() {

  // Coleta dados durante o tempo de amostragem
  if (millis() - inicio_amostragem < tempo_amostragem) {

    // le o valor de tensão no pino do sensor
    leitura = analogRead(pino_microfone);

    // se valor lido for maior que valor máximo
    if (leitura > valor_max) {
      valor_max = leitura; // atualiza valor máximo
    }
    // se valor lido for menor que valor mínimo
    else if (leitura < valor_min) {

```

```

    valor_min = leitura; // atualiza valor mínimo
  }

}
else { // processa os dados
  inicio_amostragem = millis(); // reseta o tempo inicial da amostragem

  // calcula, converte e imprime o valor de pico-a-pico em tensão elétrica
  amplitude = valor_max - valor_min; // max - min = pico-a-pico, amplitude
  amplitude = (amplitude * 5.0) / 1023.0; // converte para tensão(V)

  Serial.println(amplitude); // imprime a amplitude no Monitor Serial

  // Reseta os valores máximos e mínimos
  valor_max = 0;
  valor_min = 1024;

  // Se o valor de amplitude superar o valor limite, inverte o estado da carga
  if (amplitude > valor_limite) {
    if (palmas == 0){ // se for a primeira palma daquele período
      tempo_palmas = millis(); // reinicia a amostragem de palmas
    }
    palmas++; // incrementa o número de palmas
  }
}

// se o tempo de amostragem de palmas é atingido
if (millis() - tempo_palmas > tempo_palmas_max) {

  tempo_palmas = millis(); // reseta tempo inicial da amostragem de palmas

  // Se o número de palmas captadas naquele periodo for igual a 2
  if (palmas == 2) {
    estado_rele = ! estado_rele; // inverte o estado do rele e
    digitalWrite(pino_rele, estado_rele); // atualiza o estado do rele
  }

  palmas = 0; // reinicia o contador de palmas
}
}

```


Anexos 2 – Programação base usada no projeto do sensor PIR.

Créditos da programação: Robocore. **Kit Avançado p/ Arduino - 10. Sensor PIR.**
Disponível em: <https://www.robocore.net/tutorials/kit-avancado-para-arduino-sensor-pir>

```
boolean pir_status;

void setup() {
  pinMode(pir_signal_pin,INPUT);
  pinMode(relay,OUTPUT);
  pinMode(desligado,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  pir_status = digitalRead(pir_signal_pin);
  Serial.println(pir_status);
  if(pir_status == HIGH){
    digitalWrite(relay,HIGH);
    digitalWrite(desligado,LOW);
    delay(1000);
  }
  else{
    digitalWrite(relay,LOW);
    digitalWrite(desligado,HIGH);
  }
}
```

Anexos 3 – Programação base usada no projeto do sensor de distância.

Créditos pela programação: Como Fazer as Coisas. **Arduino com sensor de obstáculos ultrassônico HC-SRO4**. Disponível em:

<http://www.comofazerascosas.com.br/arduino-com-sensor-de-obstaculos-ultrasonico-HC-SRO4.html>

```

/*
Projeto Arduino
Arduino com sensor de proximidade ultrasonico HHC-SRO4
Por Jota
-----
--=<| www.ComoFazerAsCoisas.com.br |>==
-----
*/
//Incluindo biblioteca Ultrasonic.h
#include "Ultrasonic.h"

//criando objeto ultrasonic e definindo as portas digitais
//do Trigger - 6 - e Echo - 7
Ultrasonic ultrasonic(6,7);

//Declaração das constantes referentes aos pinos digitais.
const int ledVerde = 13;
const int ledAmarelo = 12;
const int ledVermelho = 11;

long microsec = 0;
float distanciaCM = 0;

void setup() {
  Serial.begin(9600); //Iniciando o serial monitor

  //Definindo pinos digitais
  pinMode(ledVerde,OUTPUT); //13 como de saída.
  pinMode(ledAmarelo,OUTPUT); //12 como de saída.
  pinMode(ledVermelho,OUTPUT); //11 como de saída.
}

void loop() {
  //Lendo o sensor
  microsec = ultrasonic.timing();

  //Convertendo a distância em CM
  distanciaCM = ultrasonic.convert(microsec, Ultrasonic::CM);

  ledDistancia();

  Serial.print(distanciaCM);

```

```
Serial.println(" cm");  
delay(1000);  
}
```

```
//Método que centraliza o controle de acendimento dos leds.
```

```
void ledDistancia() {
```

```
    //Apagando todos os leds
```

```
    digitalWrite(ledVerde,LOW);
```

```
    digitalWrite(ledAmarelo,LOW);
```

```
    digitalWrite(ledVermelho,LOW);
```

```
    //Acendendo o led adequado para a distância lida no sensor
```

```
    if (distanciaCM > 20) {
```

```
        digitalWrite(ledVerde,HIGH);
```

```
    }
```

```
    if (distanciaCM <=20 and distanciaCM >= 10) {
```

```
        digitalWrite(ledAmarelo,HIGH);
```

```
    }
```

```
    if (distanciaCM < 10) {
```

```
        digitalWrite(ledVermelho,HIGH);
```

```
    }
```

```
}
```

Anexos 4 – Programação base usada no projeto do sensor do Ethernet.

Créditos pela programação: Simon Monk. **Programação com Arduino: Começando com Sketches**. Tradução de Anatólio Laschuk. 1ª ed. Porto Alegre: Bookman, 2013.

```
// sketch 10-02 Internet Pins

#include <SPI.h>
#include <Ethernet.h>

// MAC address just has to be unique. This should work
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// The IP address will be dependent on your local network:
byte ip[] = { 192, 168, 1, 30 };
EthernetServer server(80);

int numPins = 5;
int pins[] = {3, 4, 5, 6, 7};
int pinState[] = {0, 0, 0, 0, 0};
char line1[100];

void setup()
{
  for (int i = 0; i < numPins; i++)
  {
    pinMode(pins[i], OUTPUT);
  }
  Serial.begin(9600);
  Ethernet.begin(mac, ip);
  server.begin();
}

void loop()
{
  EthernetClient client = server.available();
  if (client)
  {
    while (client.connected())
    {
      readHeader(client);
      if (! pageNamels("/"))
      {
        client.stop();
        return;
      }
      client.println("HTTP/1.1 200 OK");
      client.println("Content-Type: text/html");
      client.println();

      // send the body
    }
  }
}
```

```

client.println("<html><body>");
client.println("<h1>Output Pins</h1>");
client.println("<form method='GET'>");
setValuesFromParams();
setPinStates();
for (int i = 0; i < numPins; i++)
{
    writeHTMLforPin(client, i);
}
client.println("<input type='submit' value='Update'/>");
client.println("</form>");
client.println("</body></html>");

client.stop();
}
}
}

```

```

void writeHTMLforPin(EthernetClient client, int i)

```

```

{
    client.print("<p>Pin ");
    client.print(pins[i]);
    client.print("<select name=''");
    client.print(i);
    client.println(">");
    client.print("<option value='0'");
    if (pinState[i] == 0)
    {
        client.print(" selected");
    }
    client.println(">Off</option>");
    client.print("<option value='1'");
    if (pinState[i] == 1)
    {
        client.print(" selected");
    }
    client.println(">On</option>");
    client.println("</select></p>");
}

```

```

void setPinStates()

```

```

{
    for (int i = 0; i < numPins; i++)
    {
        digitalWrite(pins[i], pinState[i]);
    }
}

```

```

void setValuesFromParams()

```

```

{
    for (int i = 0; i < numPins; i++)

```

```

    {
        pinState[i] = valueOfParam(i + '0');
    }
}
void readHeader(EthernetClient client)
{
    // read first line of header
    char ch;
    int i = 0;
    while (ch != '\n')
    {
        if (client.available())
        {
            ch = client.read();
            line1[i] = ch;
            i++;
        }
    }
    line1[i] = '\0';
    Serial.println(line1);
}

boolean pageNameIs(char* name)
{
    // page name starts at char pos 4
    // ends with space
    int i = 4;
    char ch = line1[i];
    while (ch != ' ' && ch != '\n' && ch != '?')
    {
        if (name[i-4] != line1[i])
        {
            return false;
        }
        i++;
        ch = line1[i];
    }
    return true;
}

int valueOfParam(char param)
{
    for (int i = 0; i < strlen(line1); i++)
    {
        if (line1[i] == param && line1[i+1] == '=')
        {
            return (line1[i+2] - '0');
        }
    }
    return 0;
}

```

Anexo 5 – Programação base usada no projeto do sensor do Ethernet + Blynk.

Créditos pela programação: Blynk. **Exemplos de biblioteca Blynk.** Disponível em: <https://blynk.io/en/getting-started>

/*****

Download latest Blynk library here:

<https://github.com/blynkkk/blynk-library/releases/latest>

Blynk is a platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. You can easily build graphic interfaces for all your projects by simply dragging and dropping widgets.

Downloads, docs, tutorials: <http://www.blynk.cc>

Sketch generator: <http://examples.blynk.cc>

Blynk community: <http://community.blynk.cc>

Social networks: <http://www.fb.com/blynkapp>
http://twitter.com/blynk_app

Blynk library is licensed under MIT license

This example code is in public domain.

This example shows how to use Arduino Ethernet shield (W5100) to connect your project to Blynk.

NOTE: Pins 10, 11, 12 and 13 are reserved for Ethernet module. DON'T use them in your sketch directly!

WARNING: If you have an SD card, you may need to disable it by setting pin 4 to HIGH. Read more here: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

Feel free to apply it to any other example. It's simple!

*****/

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial
```

```
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
```

```
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";
```

```
#define W5100_CS 10
```

```
#define SDCARD_CS 4

void setup()
{
  // Debug console
  Serial.begin(9600);

  pinMode(SDCARD_CS, OUTPUT);
  digitalWrite(SDCARD_CS, HIGH); // Deselect the SD card

  Blynk.begin(auth);
  // You can also specify server:
  //Blynk.begin(auth, "blynk-cloud.com", 80);
  //Blynk.begin(auth, IPAddress(192,168,1,100), 8080);
  // For more options, see Boards_Ethernet/Arduino_Ethernet_Manual example
}

void loop()
{
  Blynk.run();
}
```