



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS - FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

LUCAS VINÍCIUS DA SILVA ROCHA

**AUTOMAÇÃO RESIDENCIAL:
MICROCONTROLADOR ARDUINO ATRAVÉS DE
UMA APLICAÇÃO WEB**

Orientador: Prof. Francisco Javier de Obaldia Diaz

BRASÍLIA - DF
Dezembro de 2019

LUCAS VINÍCIUS DA SILVA ROCHA

**AUTOMAÇÃO RESIDENCIAL:
MICROCONTROLADOR ARDUINO ATRAVÉS DE UMA APLICAÇÃO WEB**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de certificado de conclusão de curso de Engenharia da Computação.

Orientador: Prof. Francisco Javier de Obaldia Díaz

Brasília

Dezembro de 2019

LUCAS VINÍCIUS DA SILVA ROCHA

**AUTOMAÇÃO RESIDENCIAL: MICROCONTROLADOR ARDUINO ATRAVÉS DE
UMA APLICAÇÃO WEB**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de certificado de conclusão de curso de Engenharia da Computação.

Este trabalho foi julgado adequado para a obtenção do título de Engenheiro da Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas - FATECS.

Banca Examinadora:

Prof. Francisco Javier de Obaldia Diz - Mestre
Orientador

Prof. Ivandro da Silva Ribeiro - Mestre
Membro da banca

Prof. Oscar Fernando Gaidos Rosero - Doutor
Membro da banca

AGRADECIMENTOS

Agradeço, primeiramente, a Deus pela saúde, por renovar minhas energias todos os dias e por me proporcionar a oportunidade de concluir mais uma etapa em minha vida.

Agradeço aos meus pais pela criação que me deram e por todo apoio durante minha jornada dentro e fora da faculdade, sempre incentivando o meu crescimento pessoal e profissional.

Agradeço ao meu professor orientador, Francisco Javier, pela sua paciência e apoio, pois através de suas orientações, proporcionou a realização desse trabalho.

A todos vocês, meu muito obrigado!

Lucas Vinícius da Silva Rocha

“Bem-aventurada é a nação cujo Deus é o Senhor, e o povo ao qual escolheu para sua herança.”

Salmos 33:12

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO.....	13
1.1. Motivação	13
1.2. Apresentação do problema.....	13
1.3. Objetivos.....	13
1.4. Escopo.....	14
1.5. Resultados esperados	14
1.6. Estrutura do trabalho.....	15
CAPÍTULO 2: REFERENCIAL TEÓRICO.....	16
2.1. Domótica: Histórico.....	16
2.2. Domótica: Atualmente	20
2.3. Componentes utilizados	20
2.3.1. Arduino Uno	20
2.3.2. IDE e linguagem de programação	22
2.3.3. Ethernet Shield e servidor <i>web</i>	23
2.3.4. Trava elétrica	24
2.3.5. Módulo <i>driver</i> ponte H L298N.....	25
2.3.6. Relé.....	28
CAPÍTULO 3: DESENVOLVIMENTO.....	29
3.1. Apresentação geral do projeto proposto	29
3.2. Metodologia.....	30
3.3. Descrição das etapas	30
3.3.1. Servidor <i>web</i>	31
3.3.2. Central de controle e comandos, a aplicação <i>web</i>	31
3.3.3. Acionamento do portão automático.....	33
3.3.4. Destramento da porta	35
3.3.5. Controle de iluminação.....	37
CAPÍTULO 4: TESTES E RESULTADOS.....	40
4.1. Condições para os testes	40
4.1.1. Resultado 1 - Controle de iluminação	40
4.1.2. Resultado 2 – Destramento da porta.....	41
4.1.3. Resultado 3 – Acionamento do portão automático.....	42

4.2. Comparação dos resultados obtidos.....	44
4.3. Custos do projeto	44
CAPÍTULO 5: CONCLUSÃO.....	46
5.1. Conclusão.....	46
5.2. Sugestões para trabalhos futuros.....	46
REFERÊNCIAS	48
APÊNDICE A - APLICAÇÃO WEB	51
APÊNDICE B - <i>SKETCH</i> ARDUINO	55

LISTA DE FIGURAS

Figura 1 - Componentes do Arduino Uno	21
Figura 2 - Código em Arduino que faz piscar um LED	23
Figura 3 - Módulo Ethernet Shield	24
Figura 4 - Trava elétrica	25
Figura 5 - Circuito ponte H.....	26
Figura 6 - Módulo <i>driver</i> ponte H L28N.....	27
Figura 7 - Módulo relé de 4 canais	28
Figura 8 - Sistemas integrados com o Arduino	29
Figura 9 - Interface da aplicação <i>web</i>	32
Figura 10 - Fluxograma do funcionamento da aplicação <i>web</i>	33
Figura 11 - Motor acoplado à maquete.....	34
Figura 12 - Maquete da porta com a trava elétrica	36
Figura 13 - Conexão de interruptores <i>three way</i>	37
Figura 14 - Lâmpada acoplada à maquete	38
Figura 15 - Protótipo concluído com os componentes utilizados.....	39
Figura 16 - Página <i>web</i> do módulo de iluminação	41
Figura 17 - Página <i>web</i> do módulo de destravamento da porta.....	42
Figura 18 - Página <i>web</i> do módulo de acionamento do motor	43

LISTA DE TABELAS

Tabela 1 - Configuração dos pinos do módulo do motor	35
Tabela 2 - Descrição dos componentes e materiais utilizados e seus respectivos custos.....	45

LISTA DE ABREVIATURAS E SIGLAS

ASME - American Society of Mechanical Engineers.

AURESIDE - Associação Brasileira de Automação Residencial e Predial

CSS - Cascading Style Sheets

CV - Cavalo-vapor

HTML - Hypertext Markup Language

IDE - Integrated Development Environment

LED - Light Emitting Diode

MDF - Medium Density Fiberboard

PWM - Pulse-Width Modulation

RX - Recepção

TCP - Transmission Control Protocol

TX - Transmissão

UDP - User Datagram Protocol

USB - Universal Serial Bus

V - Volt

VCC - Tensão de Corrente Contínua

XML - Extensible Markup Language

RESUMO

Este trabalho apresenta as especificações e detalha a implementação de um protótipo de automação residencial. Com o sistema, será possível gerenciar o controle de alguns elementos encontrados em residências, como por exemplo, a abertura e o fechamento de um portão eletrônico de garagem, o destravamento da porta de entrada da casa e acender e apagar a lâmpada de um quarto. Esses módulos poderão ser controlados através de uma aplicação que rodará em um servidor *web* e que poderá ser acessada de qualquer lugar do mundo, através de dispositivos móveis e de computadores, bastando apenas ter acesso à *internet* e uma credencial de acesso.

Palavras chave: Automação residencial, Arduino, controle, microcontrolador, domótica, aplicação *web*.

ABSTRACT

This paper presents the specifications and details the implementation of a home automation prototype. With the system, it will be possible to manage the control of some elements found in homes, such as opening and closing an electronic garage door, unlocking the entrance door of the house and turning on and off the lamp in a room. These modules can be controlled through an application that will be running on a web server and can be accessed from anywhere in the world, via mobile devices and computers, simply by having access to the internet and an access credential.

Keywords: Home automation, Arduino, control, microcontroller, web application.

CAPÍTULO 1: INTRODUÇÃO

1.1. Motivação

O avanço da tecnologia está gerando diversas alterações que atingem os vários segmentos da vida. É fácil observar a contribuição das aplicações dessa tecnologia para a transformação do mundo. A partir dessa transformação surgiu a automação residencial.

Automatizar tarefas diárias e domésticas, como por exemplo, irrigação automática do jardim, entrar em casa de forma segura sem a necessidade de um molho de chaves, acender e apagar lâmpadas através do celular, traz ganho de tempo, comodidade e conforto em um mundo que demanda cada vez mais agilidade e respostas rápidas.

1.2. Apresentação do problema

É comum acontecer a situação de esquecer o controle do portão em algum lugar ou de não ter ele exatamente por perto quando seu uso se faz necessário ou, então, é comum acontecer de esquecer a chave da porta de casa e ficar trancado por fora. Além disso, a maneira mais habitual de controlar lâmpadas dos cômodos de uma casa, é através do uso de interruptores, porém esse procedimento exige locomoção e a necessidade de estar no mesmo cômodo. Com a crescente popularização da internet e os dispositivos móveis mais acessíveis do que nunca, agora é possível dar um novo significado para a maneira em que fazemos cada uma dessas coisas. Agora é possível fazer uso do smartphone, que sempre está em mãos, para ter controle sobre esses elementos presentes no cotidiano e tornar tudo mais ágil, prático e eficiente.

1.3. Objetivos

O objetivo geral deste trabalho é desenvolver um protótipo que contenha alguns elementos presentes em um ambiente residencial automatizado, fornecendo uma forma de comunicação entre sistemas, a fim de trazer benefícios e comodidade ao usuário do sistema, gerando controle através de um computador e de dispositivos móveis.

Nos objetivos específicos, destaca-se o desenvolvimento de uma arquitetura para implementação física do projeto. Implementar no Arduino as lâmpadas que representam as

luzes da casa, ligando nas saídas digitais do microcontrolador, com o intuito de fornecer a energia necessária quando o comando for enviado pela aplicação web.

Outro destaque fica por conta da programação do Arduino. Desenvolver o código de programação na linguagem própria do microcontrolador, para que este receba os comandos da aplicação web construída em HTML, CSS e JavaScript e os interprete corretamente.

Além disso, outro destaque é a montagem da maquete residencial. Ou seja, o desenvolvimento de uma maquete de uma residência e o protótipo de uma porta e de um portão, para a representação dos ambientes e demonstração do funcionamento da fechadura elétrica.

1.4. Escopo

Esse projeto baseia-se na implementação física de um protótipo de automação residencial. O protótipo será apresentado usando uma maquete contendo todos os módulos que serão automatizados, tais como o controle de abertura e fechamento do portão eletrônico, o controle de acesso da porta de entrada e o controle de iluminação de um dos cômodos da residência.

Além disso, o projeto contempla uma aplicação *web* que permitirá o controle desses itens da casa, podendo ser acessada através de dispositivos móveis e computadores. Essa aplicação estará baseada em um sistema de autenticação, que permite que somente pessoas autorizadas tenham acesso ao controle automatizado da residência.

O projeto não prevê um banco de bateria, caso haja suspensão do fornecimento de energia elétrica para o devido funcionamento dos componentes e acionamento dos dispositivos.

1.5. Resultados esperados

Ao final do projeto, esperava-se o desenvolvimento, a construção e a montagem do *hardware* para o correto acionamento de equipamentos eletroeletrônicos pertencentes à um ambiente residencial, e de uma aplicação para internet para o controle desses módulos. Além disso, espera-se ter, para demonstração, a maquete de uma casa contendo as partes que serão automatizadas, tais como, o portão da garagem, a porta de entrada da casa e a lâmpada de um dos cômodos.

1.6. Estrutura do trabalho

Esse trabalho contém a seguinte estrutura:

O capítulo 1 apresenta o problema, os objetivos, o escopo do projeto e informa os resultados esperados ao final do projeto.

O capítulo 2 possui o referencial teórico sobre automação residencial, sobre o microcontrolador Arduino e sobre os módulos utilizados.

No capítulo 3 será apresentado como aconteceu o desenvolvimento do projeto, expondo as funcionalidades e especificando cada recurso utilizado para cumprir o objetivo proposto.

O capítulo 4 apresentará os testes realizados, os resultados obtidos e os custos da realização do projeto.

No capítulo 5 são apresentadas as conclusões, as sugestões para trabalhos futuros e as considerações finais do projeto.

CAPÍTULO 2: REFERENCIAL TEÓRICO

2.1. Domótica: Histórico

Domótica, ou automação residencial, é um conceito que cuida do controle de recursos habitacionais. O termo tem origem através da palavra “domus”, que significa casa, e da palavra “robótica”, que vem do sentido de automatizar, ou seja, realizar tarefas de forma automática. (LENZ, 2017).

Buscando proporcionar conforto, praticidade e comodidade para quem a utiliza, essa tecnologia visa, antes de qualquer coisa, integrar componentes, dispositivos, aparelhos e sistemas, alguns desses são novidades em ambientes domésticos, mas muitos já fazem parte da realidade de muitas residências, mas antes não estavam interligados e não funcionavam de maneira autônoma. Por esse e por outros motivos, a domótica é considerada um conceito recente e que apresenta muitos pontos a serem desenvolvidos.

A automação residencial teve seu surgimento ainda nos primórdios da Humanidade. Qualquer processo que ajude o ser humano diariamente em suas atividades é considerado automatização, podem ser tarefas domésticas, industriais, comerciais ou tarefas destinadas ao campo. Um exemplo que pode ser citado é a roda d'água que auxiliava o processo de moagem, ferrarias, trituração de grãos e na serraria. Impulsionada no século XVIII, a Revolução Industrial possibilitou mais ainda a automação pelo mundo. (TEZA, 2002).

O francês Júlio Verne foi um dos primeiros a escrever sobre esse assunto em seus livros de literatura, e desde então, a automação residencial passou a ser um tema característico de histórias envolvendo a ficção científica, mas a automação como conhecemos hoje só passou a ser algo mais palpável a partir da introdução generalizada de energia elétrica nas residências, no começo do século XX, visivelmente com o fim da primeira guerra mundial, e conseqüentemente com o gradual avanço dos sistemas de controle automático e da tecnologia da informação. (LENZ, 2017).

Olhando para a história, é possível analisar que de maneira geral e até o presente momento, a automação residencial vem seguindo os princípios da automação predial, apresentando-se como uma extensão desta, em especial com enfoque em pretender controlar a iluminação, a segurança, a climatização e a ligação entre esses três elementos. Entretanto, vem ganhando cada vez mais força e impulsos comerciais próprios que vão além do enfoque original.

Por causa das pesquisas de Michael Faraday e de Ányos Jedlik, em 1832, William Sturgeon estabeleceu um princípio para converter energia elétrica em energia mecânica. (LENZ, 2017). Posteriormente esse princípio gerou várias inovações que foram capazes de trazer à tona o motor elétrico de corrente contínua que se mostrou possível de ser utilizado nas indústrias.

O tempo foi passando e enquanto isso a automação em prédios já estava caminhando. Em 1885, após a criação do termostato elétrico de ambiente, a empresa fundada por Warren Johnson passou a utilizar essa tecnologia para controle da temperatura de ambientes internos de prédios. Um dos primeiros projetos feitos por essa empresa foi o de regulação de temperatura nos prédios da biblioteca pública e da prefeitura de Milwaukee, nos Estados Unidos.

Em 1890 começaram surgir os primeiros dispositivos para controle remoto ou automático, principalmente para usos industriais. A empresa *Johnson Electric Service Company* realizou a patente e melhorou uma série de dispositivos atuadores e sensores para a tecnologia de controle térmico pneumático em 1895, sendo o primeiro sistema para controle de temperatura para prédios. Hoje em dia, Warren Johnson é conhecido como o avô de todos os sistemas de controle pela ASME - *American Society of Mechanical Engineers*.

Por mais que a eletricidade tenha começado a se tornar disponível comercialmente para as residências dos Estados Unidos na década de 1880, alguns obstáculos interferiram na implementação da tecnologia que estava em desenvolvimento até próximo de 1920. (LENZ, 2017).

Isso aconteceu porque as empresas de serviços públicos não estavam prontas para investir de forma adequada na infraestrutura necessária para contemplar a demanda de energia elétrica para as indústrias e para a iluminação doméstica simultaneamente.

As consequências políticas e econômicas do comprometimento dos Estados Unidos na Primeira Guerra Mundial aceleraram a inserção de aparelhos elétricos nas casas. Inclusive, o Conselho Industrial de Guerra reconheceu o significado da energia elétrica e forneceu uma alta prioridade na alocação de força de trabalho e materiais para a indústria de aparelhos eletrodomésticos. Uma vez que, durante a guerra, a imigração foi intensamente reduzida e o recurso primordial de mão de obra barata era de imigrantes europeus e esse recurso foi cortado.

Além disso, a demanda dos consumidores por bens duráveis estava limitada pelas alocações de esforço de guerra e a liberação desta demanda pelo Conselho Industrial de Guerra beneficiou bastante as indústrias de eletrodomésticos. A tecnologia do telefone, da

luz elétrica, do bonde elétrico e do telégrafo tinha desenvolvido um grande conjunto de trabalhadores qualificados em elétrica nos EUA, por causa dos esforços da Primeira Guerra Mundial.

Dessa forma, os fatores necessários para a eletrificação das residências americanas estavam de acordo com o fim da guerra e, simultaneamente, a maioria dos americanos de classe média estavam diante de eletrodomésticos que estavam ao seu alcance por causa da evolução da tecnologia. O avanço da tecnologia teve quatro fatores de suma importância.

O primeiro deles foi a invenção de uma resistência elétrica para lâmpadas que durava bem mais do que o comum e que era de baixo custo. O filamento de tungstênio criado por Sándor Just e por Franjo Hanaman, e que depois foi aperfeiçoado pela empresa *General Electric*, superou todos os outros tipos que existiam até então. (LENZ, 2017).

Outro fator aconteceu em 1906 quando Albert Leroy Marsh realizou a patente do resistor de fio nicromo, tornando-se a descoberta base de todas as outras seguintes no segmento de aparelhos elétricos de aquecimento. A liga de níquel cromo que Marsh desenvolveu tinha uma alta resistência elétrica, alto ponto de fusão, não custava caro para ser fabricada e pouco passível de sofrer oxidação. (LENZ, 2017).

Além disso, outro fator que veio de encontro com as necessidades das aplicações domésticas, foi o começo da fabricação de motores de indução de corrente alternada monofásicos e de pequeno porte. A empresa *WestingHouse* fabricou o primeiro motor de corrente alternada de porte menor do que 1 CV (cavalo-vapor) na década de 1890, a partir dos projetos de Nikola Tesla, porém a maior parte dessas unidades teve como fim as indústrias, não tinham uma velocidade boa e atuavam em três fases. (LENZ, 2017).

C. H. Hamilton realizou o projeto e começou a produzir em 1909 um motor de potência fragmentada confiável que produzia de 8.000 a 10.000 rotações por minuto, o que representou um importante avanço para a época. Outra vez a *General Electric* aparece e amplia o campo com um motor desenvolvido especialmente para ser usado em máquinas de lavar residenciais em 1913. Alfred Mellowes desenvolveu em 1915 uma unidade de refrigeração elétrica diferente daquelas que estavam sendo produzidas naquela época, uma vez que o compressor e o motor estavam contidos no fundo do compartimento. Em 1916 a empresa *Frigidaire*, que antes era conhecida como *Guardian Refrigerator Company*, começou a fabricar e a vender a versão dessa geladeira. Mesmo sendo um produto de alta qualidade, ela fabricou apenas 40 unidades nos primeiros dois anos. (LENZ, 2017).

Após realizadas melhorias constantes em como o refrigerador era produzido, foi possível chegar em um produto de melhor qualidade e que tivesse um custo menor na linha de

produção. Já em 1929, aconteceu um importante avanço na produção comparado com os anos anteriores, e foram produzidas 1 milhão de geladeiras.

Da mesma forma, o desenvolvimento da indústria de automação de prédios seguia seu próprio ritmo. No início do século XX era construído em Nova Iorque o primeiro arranha-céu do mundo, o *Singer Tower*, ficando pronto em 1908. A empresa *Johnson Service Company* desenvolveu e construiu o sistema de controle térmico do prédio, com 1200 salas sendo monitoradas e controlando 1800 válvulas. (LENZ, 2017).

Em 1911, após a morte de Warren Johnson, a empresa sofre uma grande alteração em sua estrutura. Rapidamente as construções de edifícios se multiplica e o foco dos negócios é mais centrado em automação e controle de térmico dos ambientes.

Os negócios acabaram caindo e os contratos com o governo foram dificultados durante a Primeira Guerra Mundial e o que ajudou a empresa a se manter no mercado foi a necessidade de reformar os sistemas de controle de temperatura de prédios mais antigos. Com o fim da guerra, a quantidade de salas de cinema aumentou consideravelmente e através disso surgiu a oportunidade de projetos que fizessem o controle para o devido condicionamento de ar. Através do seu sistema pneumático, a empresa apostava na segurança e na eficiência energética, caindo em cima daqueles que ainda usavam combustíveis.

A companhia adquire em 1925 a patente de um termostato pressostato. Para a época, tratava-se de um dispositivo revolucionário que tinha a capacidade de detectar pela variação de pressão do ar, se tinham ou não pessoas na sala, e dessa forma, poupava energia elétrica de forma automática quando as salas estavam desocupadas. Durante o período da grande depressão econômica, a empresa conseguiu se manter viva no mercado devido a ajuda de controles focados em eficiência energética que foram derivados da tecnologia do termostato pressostato e de inovações realizadas nos sistemas de refrigeração e controle de umidade. (LENZ, 2017).

Após a automação ser consolidada, o comércio foi em seguida contemplando com sua automação que vem até hoje evoluindo, especialmente com o avanço acelerado da informática e dos softwares de supervisão e gerenciamento. Hospitais, supermercados, lojas de departamentos, hotéis e outros têm a operação de seus serviços totalmente integrados, desde o setor de logística, passando pelo setor de vendas, finanças e etc. Tal automação atinge até os pequenos comércios e prestadores de serviços.

2.2. Domótica: Atualmente

Inicialmente, automação residencial pode ser referenciada como uma novidade que vez e outra causa perplexidade por causa do seu grau de tecnologia envolvida e pela alusão ao futurismo e ao mesmo tempo é considerada como sendo um símbolo de modernidade e de alto valor aquisitivo. Porém, ao realizar uma observação mais realista, é possível ver que a automação residencial proporciona o conforto e a praticidade que o ser humano almeja.

José Cândido Forti, da Associação Brasileira de Automação Residencial e Predial, a AURESIDE, declarou em um congresso em 2001 que “transformar casas em confortáveis refúgios capazes de oferecer segurança e economia de custos é uma das vantagens da automação residencial. O que antes parecia ser um privilégio apenas da família *Jetson*, começa a se difundir nos empreendimentos residenciais de alto nível, transformando o conceito de casa do futuro em casa do presente.”

Os sistemas de automação podem ser do tipo que cuida da segurança, da comunicação, da iluminação, do controle de temperatura de ambientes, entre outros, sempre com o objetivo de promover comodidade, conforto, segurança e eficiência energética. Ou seja, utiliza vários elementos e através de uma forma sistêmica, utiliza as vantagens dos meios de informática e dos meios mecatrônicos, aproveitando e gerenciando de forma integrada os diversos equipamentos de uma residência. (TEZA, 2002).

2.3. Componentes utilizados

Para montagem do protótipo, conforme os requisitos que foram apresentados e com a finalidade de alcançar os resultados esperados, foi indispensável o uso dos seguintes componentes apresentados a seguir:

2.3.1. Arduino Uno

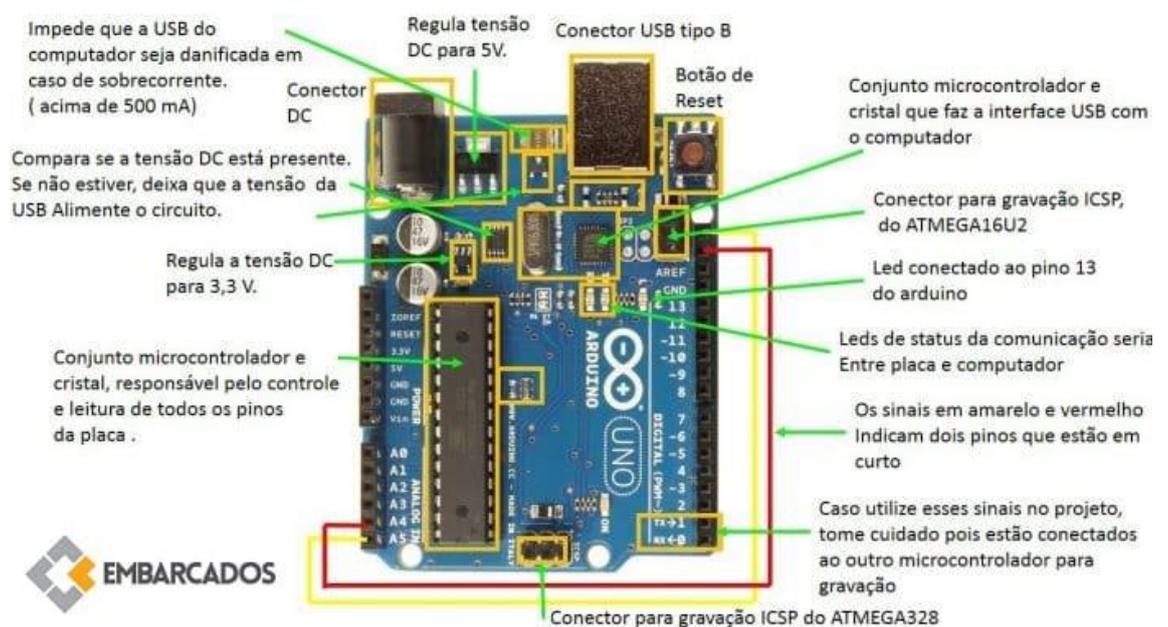
Para implementar um circuito interativo antigamente, era obrigatório lidar com projetos do zero para ter em mãos aplicações específicas. Além disso, para fazer alterações mínimas nas funcionalidades do circuito era necessária uma análise técnica criteriosa, detalhada e muito trabalho envolvido. Esse era o cenário encontrado até o surgimento dos microcontroladores.

Com o advento dos microcontroladores, passou a ser possível lidar com *softwares* para resolver problemas que antes eram tratados somente com *hardwares*. Assim, programas de computador podem ser escritos para gerar funções totalmente diferentes em um mesmo circuito, alterando e reprogramando alguns parâmetros do código que havia sido implementado. (PORTUGAL, 2017)

O Arduino foi criado em 2005 justamente com esse propósito. Um grupo de pesquisadores italianos teve a ideia de elaborar um dispositivo que fosse ao mesmo tempo barato, fácil de programar, que fosse funcional, além disso, acessível para estudantes e projetistas. O conceito de *hardware* livre acompanha o Arduino, dessa forma, qualquer um pode melhorar, modificar, personalizar e montar o Arduino, tendo como princípio o mesmo *hardware* básico. (TELES, 2016)

Apesar de não ter a opção de fazer *debug*, trata-se de uma ótima ferramenta para quem está estudando o assunto ou para quem deseja fazer simulação de projetos antes de partir para uma placa finalizada, com os componentes soldados. É simples e possui características de *hardware* mínimas, mas que atendem perfeitamente vários tipos de projetos. A conectividade USB e a facilidade em programar são atrativos diferenciados e que possuem grande importância para quem busca fazer projetos destinados ao Arduino. A figura 1 mostra as principais partes que compõem o Arduino.

Figura 1 - Componentes do Arduino Uno



A partir de todos esses conceitos e objetivos, foi criada uma pequena e potente placa composta por um microcontrolador Atmel, circuitos de entrada e de saída e que pode ser conectada de forma muito simples à um computador e programada através de uma IDE, *Integrated Development Environment*, utilizando uma linguagem baseada em C/C++, precisando apenas de um cabo USB e dispensando a necessidade de equipamentos adicionais.

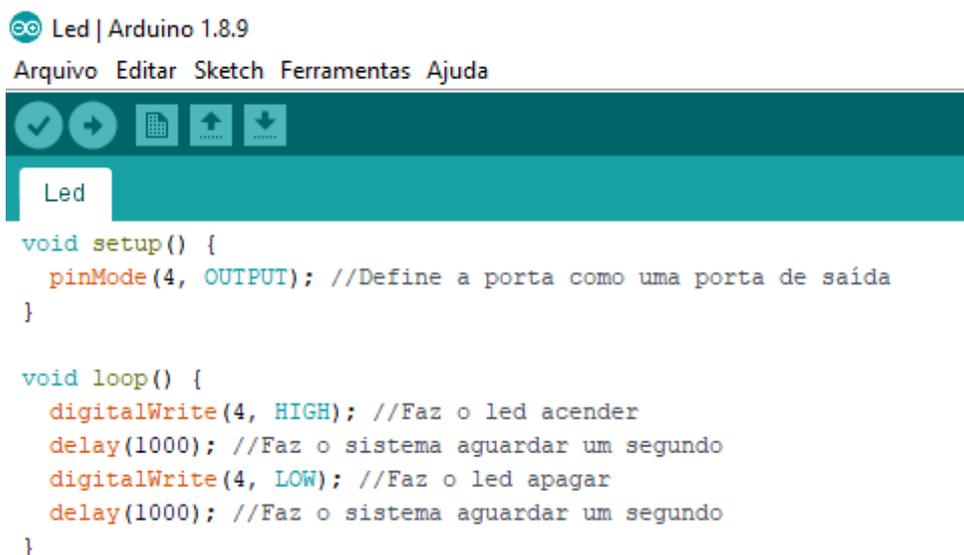
O Arduino possui várias versões e tipos, por exemplo, o Uno, Mega, Leonardo, Due, Mega ADK, Nano, Pro Mini, e o Esplora, tendo como principais diferenças os microcontroladores utilizados, número de portas, conexões possíveis e memória. Para esse projeto, o modelo Arduino Uno irá atender as necessidades previamente especificadas.

2.3.2. IDE e linguagem de programação

Para implementar programas para serem utilizados em um Arduino é necessário utilizar uma IDE, onde todo o código é escrito, são feitos testes para encontrar possíveis erros e para transferir o programa para a placa. Esse ambiente de programação pode ser baixado de forma gratuita no site oficial da plataforma: <https://www.arduino.cc/en/Main/Software/>.

O código escrito para o Arduino é composto por duas partes básicas. A primeira delas é chamada de *setup()* e é nesse bloco que as opções iniciais do programa são definidas e escritas. Nessa parte, por exemplo, é possível definir se uma porta será usada como entrada ou saída. No segundo bloco, chamado de *loop()*, é escrita a estrutura de comandos que será executada de forma constante, ou até que um comando de “pare” seja enviado ao Arduino. A figura 2 mostra um código de exemplo, escrito na própria IDE do Arduino, em que fica visível os conceitos por trás de cada bloco.

Figura 2 - Código em Arduino que faz piscar um LED



```
Led | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda

void setup() {
  pinMode(4, OUTPUT); //Define a porta como uma porta de saída
}

void loop() {
  digitalWrite(4, HIGH); //Faz o led acender
  delay(1000); //Faz o sistema aguardar um segundo
  digitalWrite(4, LOW); //Faz o led apagar
  delay(1000); //Faz o sistema aguardar um segundo
}
```

Fonte: Print de tela gerado pelo autor

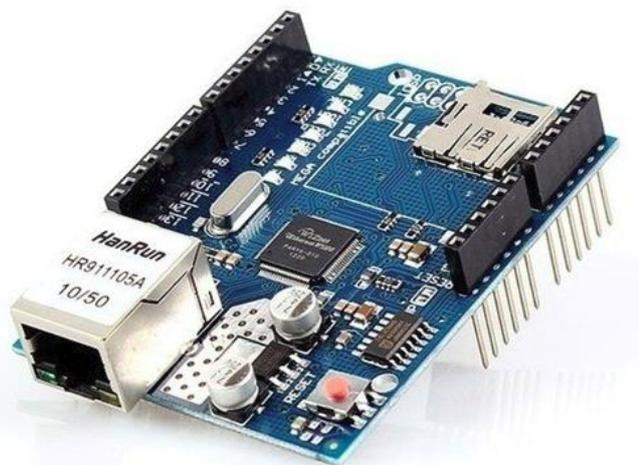
2.3.3. Ethernet Shield e servidor *web*

Com o uso cada vez mais popular do Arduino e com a popularidade sempre maior da internet, surgiu a necessidade de conectar a plataforma à rede de computadores, tendo como objetivo o acionamento e controle de forma mais fácil e inovadora, sendo possível ser controlado por computadores ou dispositivos móveis em qualquer lugar do mundo.

É baseado no chip *WIZnet Ethernet W5100* e duas bibliotecas nativas são usadas para poder ser utilizado, tais como, *Ethernet Library* e a *SD Library*. Para ser utilizado, basta plugar um cabo de rede do tipo RJ45 e conectá-lo ao modem/roteador e ter as ligações do projeto feitas diretamente no *shield*. (VIDAL, 2019).

O módulo Ethernet Shield foi criado para suprir essa necessidade. Com seu uso, é possível conectar o Arduino na internet usando os protocolos TCP ou UDP, tornando ainda mais eficiente sua utilização e controle. Outro diferencial é o seu formato de *shield* que oferece rapidez e praticidade no momento da instalação do Arduino. A figura 3 traz a representação de um módulo Ethernet Shield.

Figura 3 - Módulo Ethernet Shield



Fonte: Site Eletrogate (<https://blog.eletrogate.com/ethernet-shield-w5100-com-arduino/>), 2019, Acesso em set. 2019)

Trata-se de um *shield* compatível com as versões Uno e Mega, por exemplo, e possui um slot para cartão micro-SD, que pode ser usado para armazenar arquivos que podem ser enviados pela rede local ou internet. Além disso, o Ethernet Shield suporta até quatro conexões de socket simultâneas. (OLIVEIRA, 2018).

O Ethernet Shield cria um servidor web que permite a leitura de dados através de sensores conectados ao Arduino ou o controle desses sensores através de uma página que pode ser acessada facilmente em um navegador de celular ou de computador.

2.3.4. Trava elétrica

Para realizar o controle de movimentação da porta de entrada do projeto, será necessário fazer uso de uma trava elétrica. A trava elétrica utilizada nesse projeto é muito semelhante com as travas que existem em portas comuns (Ver figura 4). Os dois tipos são compostos por uma estrutura que possui uma parte que se movimenta e pode impedir ou liberar a abertura da porta de acordo com sua posição atual. Enquanto na trava tradicional a posição dessa parte é controlada por uma chave, as travas elétricas mudam seu estado de acordo com a aplicação de uma certa tensão em seus terminais.

Quando 12V são aplicados nos terminais da trava elétrica, a parte que controla a abertura da porta é recolhida e fica condicionada dentro da estrutura e fica dessa

forma até que a trava esteja recebendo um sinal de tensão. Da mesma forma, se a trava não estiver recebendo tensão na entrada de seus terminais, essa mesma parte fica para fora da estrutura e impede a abertura da porta. (MADEIRA, 2018).

Figura 4 - Trava elétrica



Fonte: Site Vida de Silício (<https://portal.vidadesilicio.com.br/trava-eletrica-solenoid/>, 2018, Acesso em set. 2019)

Essas travas necessitam de 12V para funcionar, porém as placas Arduino não conseguem fornecer tanta tensão. Para resolver isso, é necessário fazer uso de algum tipo de artifício. Nesse caso, um módulo relé será utilizado para que sua bobina possa ser energizada com 5V, de uma maneira que, quando receber tensão, alguns de seus contatos serão fechados e fornecerão caminho para a corrente proveniente de uma fonte de externa de 12V e assim será possível operar a trava elétrica através do Arduino.

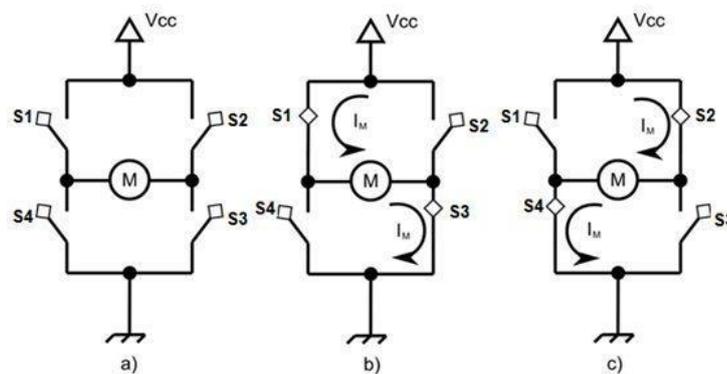
2.3.5. Módulo *driver* ponte H L298N

A utilização de motores DC é amplamente necessária em vários projetos de robótica, tais como locomoção de robôs, movimentação de braços mecânicos, etc. Motores desse tipo são cargas indutivas que geralmente precisam de uma quantidade de corrente superior à que as portas do Arduino conseguem fornecer. Dessa forma, não é recomendável ligar estes motores diretamente nas portas do Arduino, pois se o motor demandar uma corrente acima de 40mA nas portas digitais, que é a corrente máxima fornecida pelo Arduino, é possível que queime a porta e danifique a placa.

Para resolver esse problema da corrente, ter a função de inverter o giro do motor e de controlar sua velocidade, basta utilizar um módulo *driver* ponte H L298N. Seu uso é muito simples e traz grande utilidade para o projeto.

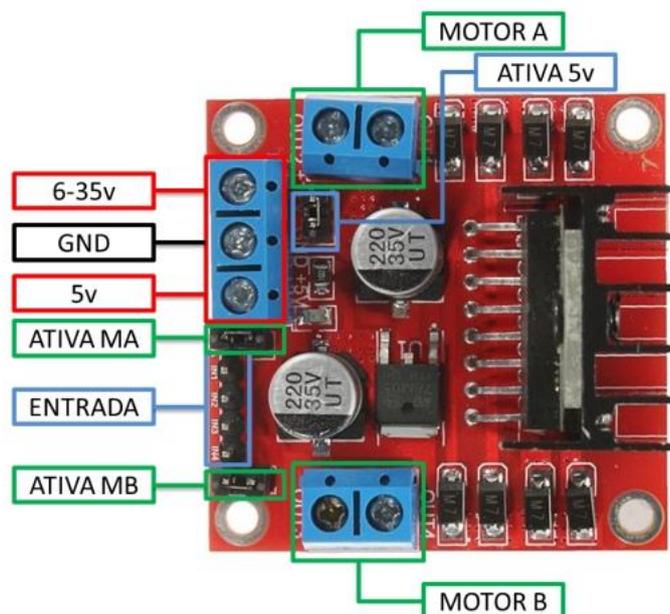
A ponte H possui esse nome porque o formato em que seu circuito é montado lembra a letra H, como pode ser visto na figura 5. O circuito utiliza quatro chaves (S1, S2, S3 e S4) que são acionadas de forma alternada, ou seja, (S1 - S3) ou (S2 - S4). Dependendo da configuração entre as chaves, tem-se corrente percorrendo o motor em um momento através de um e sentido e em outro momento, em outro sentido. Quando nenhum par de chaves está acionado, o motor está desligado (a). Quando o par S1 - S3 é acionado, a corrente percorre esse fluxo fazendo com que o motor gire em um sentido (b). Já quando o par S2 - S4 é acionado, a corrente percorre outro caminho fazendo com que o motor gire no sentido contrário (c). (CARDOSO, 2017).

Figura 5 - Circuito ponte H



Fonte: Site Vida de Silício (<https://portal.vidadesilicio.com.br/driver-motor-com-ponte-h-l298n/>, 2017, Acesso em out. 2019)

Outra vantagem do L298N é a utilização de sinais PWM. Assim, é possível controlar a tensão de saída e dessa forma regular a velocidade dos motores. O PWM, *Pulse Width Modulation* (Modulação por Largura de Banda), consiste basicamente em aplicar uma onda quadrada de amplitude VCC e frequência alta no lugar da tensão contínua VCC. A figura 6 apresenta os componentes que fazem parte do módulo L28N.

Figura 6 - Módulo *driver* ponte H L28N

Fonte: Site FILIPEFLOP (<https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>, 2013, Acesso em out. 2019)

- **(Motor A) e (Motor B):** Se referem aos conectores para ligação de dois motores DC ou um motor de passo.
- **(Ativa MA) e (Ativa MB):** São os pinos responsáveis pelo controle PWM dos motores A e B. Se estiver com *jumper*, não haverá controle de velocidade, pois os pinos estarão ligados aos 5V. Esses pinos podem ser utilizados em conjunto com os pinos PWM do Arduino.
- **(Ativa 5V) e (5V):** Este *driver* possui um regulador de tensão integrado. Quando o *driver* está operando entre 6 – 35V, este regulador disponibiliza uma saída regulada de +5V no pino (5V) para um uso externo (com *jumper*), podendo alimentar, por exemplo, outro componente eletrônico.
- **(6 – 35V) e (GND):** Aqui será conectada a fonte de alimentação externa quando o *driver* estiver controlando um motor que opere entre 6 – 35V. Por exemplo, se estiver usando um motor DC 12V, basta conectar a fonte externa de 12V neste pino e no (GND).
- **(Entrada):** Este barramento é composto por IN1, IN2, IN3 e IN4. Sendo estes os pinos responsáveis pela rotação do Motor A (IN1 e IN2) e Motor B (IN3 e IN4).

2.3.6. Relé

Os módulos relés são componentes eletromecânicos capazes de controlar circuitos externos de grandes correntes a partir de pequenas tensões ou correntes, em outras palavras, é possível utilizar um módulo relé, como o mostrado na figura 7, acionado por uma pilha para controlar um motor que esteja ligado em 110 ou 220 volts, por exemplo.

Figura 7 - Módulo relé de 4 canais



Fonte: Site Eletrogate (<https://blog.eletrogate.com/modulo-rele-para-automacao-residencial-com-arduino/>, 2018, acesso em set. 2019)

A partir do Arduino, é possível controlar cargas de corrente alternada de forma prática e simples. Seu uso é muito comum em projetos de automação, e nesse projeto, será utilizado para controle da lâmpada e da trava elétrica da porta de entrada. Um módulo relé de 4 canais será utilizado para fazer o controle de todos os elementos especificados.

O funcionamento dos relés se dá de uma maneira muito simples, basicamente precisando de corrente elétrica para funcionar. Quando passa corrente pela bobina, um pequeno campo magnético é gerado e um material condutor é atraído, fazendo com que o circuito fique no modo aberto ou fechado. O campo magnético gerado só termina quando a corrente elétrica é interrompida na bobina, e isso faz com que o relé volte para sua posição original. (MOTA, 2017).

Os fundamentos e conceitos técnicos que foram descritos neste capítulo são a base para o desenvolvimento do protótipo contemplado no projeto, de acordo com o que poderá ser visto nos próximos capítulos.

CAPÍTULO 3: DESENVOLVIMENTO

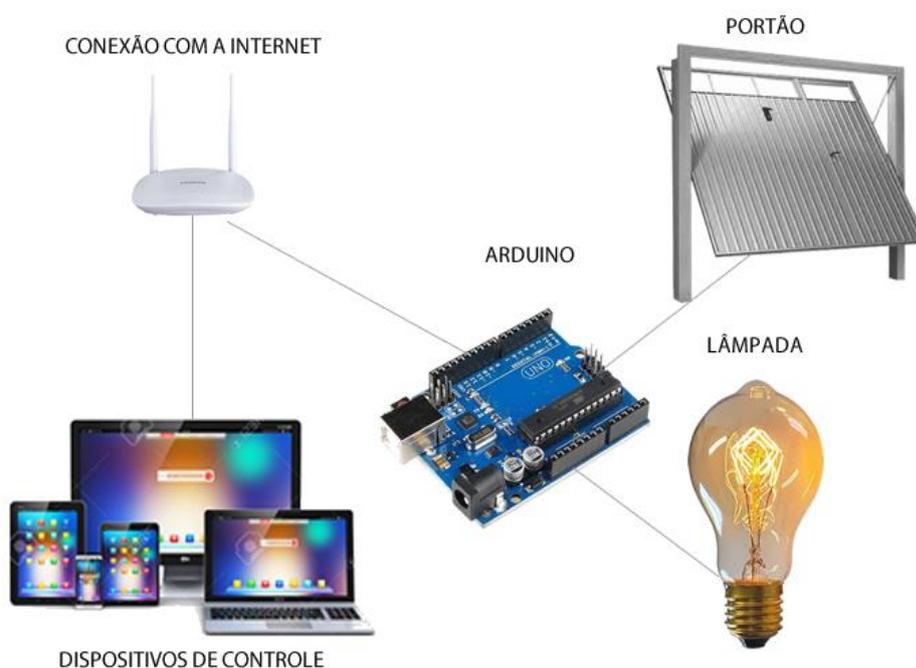
3.1. Apresentação geral do projeto proposto

Trata-se do desenvolvimento de um protótipo que contenha alguns elementos presentes em um ambiente residencial automatizado, fornecendo uma forma de comunicação entre sistemas, a fim de trazer benefícios e comodidade ao usuário do sistema, gerando controle através de um computador e de dispositivos móveis. A automação dos dispositivos do sistema integrado com a aplicação web são os relacionados a seguir:

- Controle de acionamento do portão automático da residência;
- Controle de abertura da porta de entrada da residência;
- Controle de iluminação de um dos cômodos da residência.

Para atender as especificações da aplicação *web*, uma interface foi construída com o propósito de possibilitar a comunicação com o Arduino e o controle de cada dispositivo automatizado. A figura 8 é um esquemático que mostra de forma simplificada como a interação do usuário com os dispositivos automatizados ocorre.

Figura 8 - Sistemas integrados com o Arduino



Fonte: Montagem de imagens gerada pelo autor

Através de um *smartphone*, *tablet* ou computador conectado à internet, será possível realizar requisições ao servidor *web* implementado no próprio Arduino com o uso do módulo Ethernet Shield, que automaticamente se comunica com a placa controladora passando os comandos desejados pelo usuário para que seja realizado o controle da lâmpada, a abertura da porta e o acionamento do portão eletrônico.

3.2. Metodologia

A metodologia do projeto compreende quatro partes principais: os componentes físicos, a comunicação, os softwares e as maquetes para demonstração. Os componentes físicos podem ser compreendidos em relés, placa Arduino, módulo Ethernet Shield, *jumpers*, trava elétrica, motor, *notebook*, celular, e etc. A parte de comunicação acontecerá através de Wi-Fi, *ethernet* e USB, usando os protocolos TCP ou UDP. Já a parte de software envolve ambientes de desenvolvimento, sendo eles o *Sublime Text* para codificação de toda a aplicação *web* e o *software* Arduino para criação e compilação do *sketch* utilizado. As maquetes compreendem a utilização de placas de madeira do tipo MDF, que serão usadas para compor as paredes, o teto e os muros da casa, e na maquete da porta para demonstração do funcionamento da trava elétrica.

3.3. Descrição das etapas

Uma das maiores preocupações de projetistas e instaladores de sistemas desse tipo é justamente a integração entre os dispositivos que compõem esses sistemas. Por causa disso, uma das soluções encontradas foi a de centralizar esses dispositivos em um único sistema.

Com a intenção de integrar o controle desses dispositivos, passou a fazer parte desse projeto a implementação de um sistema interativo onde o usuário pudesse controlar com facilidade cada componente. Outro fator que motivou a escolha de uma aplicação *web* para controle dos dispositivos automatizados foi justamente o fato de que atualmente a tecnologia está ao alcance das pessoas em quase todas as classes sociais, dessa forma, o uso de dispositivos móveis e computadores que permitem o acesso à internet pode ser utilizado para facilitar fortemente o controle e gerenciamento de sistemas em casas automatizadas, sem falar que são objetos que estão sempre ao alcance das mãos e por isso evita deslocamento ou o uso de objetos adicionais, gerando conforto e comodidade.

3.3.1. Servidor *web*

A função do servidor *web* é a de hospedar a aplicação construída para proporcionar um ambiente de fácil interação entre o usuário e os dispositivos que ele deseja controlar. Como é uma aplicação que não demanda tanto poder de processamento, foi utilizado o módulo Ethernet Shield que conecta o Arduino à internet e cria um servidor local que pode ser usado para comunicação com a rede. A aplicação pode ser escrita no próprio código *sketch* do Arduino ou pode ser escrita em arquivos separados e esses podem ser armazenados em um cartão de memória micro SD em um slot que o próprio módulo Ethernet Shield disponibiliza. Nesse projeto, os arquivos estão armazenados no cartão de memória micro SD, o que permite maior personalização e escalabilidade da aplicação *web*.

A conexão do Arduino com a internet é feita conectando todos os pinos do Ethernet Shield ao Arduino e posteriormente ligando um cabo *ethernet* em um roteador. A partir disso, qualquer dispositivo conectado na mesma rede local já consegue se comunicar com o Arduino, basta digitar na barra de endereço do navegador de internet o endereço IP que foi definido no código *sketch* compilado no Arduino. Esse endereço IP precisa ser um endereço válido e disponível na rede local, pois a partir desse momento de configuração, esse IP ficará atrelado ao microcontrolador.

Algumas configurações na página de personalização do roteador são necessárias para permitir a conexão em redes externas. É necessário que todas as requisições HTTP, ou seja, a liberação da porta 80, feitas por qualquer dispositivo, sejam encaminhadas ao servidor *web*. Cada roteador possui uma tela de configuração diferente e varia de acordo com o fabricante.

3.3.2. Central de controle e comandos, a aplicação *web*

A central de controles e comandos, também chamada de aplicação *web*, conta com um item em um menu horizontal para cada dispositivo automatizado. Através disso, o usuário pode selecionar o dispositivo que deseja controlar e acionar, de forma intuitiva, todos os comandos desejados.

Para garantir a segurança da aplicação e evitar que acessos indesejados aconteçam, foi adicionado um produto *Google* chamado *Firebase*. Trata-se de uma

ferramenta que disponibiliza diversos recursos em várias linguagens com o propósito de ajudar no desenvolvimento de sistemas web e mobile com aplicações de alta qualidade e performance. *Firebase* é uma plataforma de desenvolvimento web mobile adquirida pela *Google* em 2014. Com foco em ser um *back-end* completo e de fácil usabilidade, essa ferramenta disponibiliza diversos serviços diferentes que auxiliam no desenvolvimento e gerenciamento de aplicativos.

Para utilizar o *Firebase*, um console web foi construído pela *Google* para facilitar a implementação. Neste, o desenvolver adiciona um projeto e inclui os serviços que desejar, cada um com uma explicação de como proceder. Nem todos os serviços são gratuitos, porém é possível criar um plano conforme as necessidades do desenvolvedor, caso ele precise de algo a mais do que já é oferecido gratuitamente.

A aplicação *web* do projeto foi construída de forma a ter integração com essa ferramenta. Dessa forma, só podem acessar a aplicação os usuários que estiverem previamente cadastrados na base de dados dessa ferramenta. Esse recurso torna a aplicação mais segura, confiável e transparente.

Além disso, foi utilizado HTML, CSS para estilizar os componentes utilizados na página e JavaScript para criação de algumas funções, tratativa de condições de elementos dentro da página e para a comunicação do Arduino com a aplicação *web*. Todo o código foi construído e implementado com o uso da ferramenta *Sublime Text*.

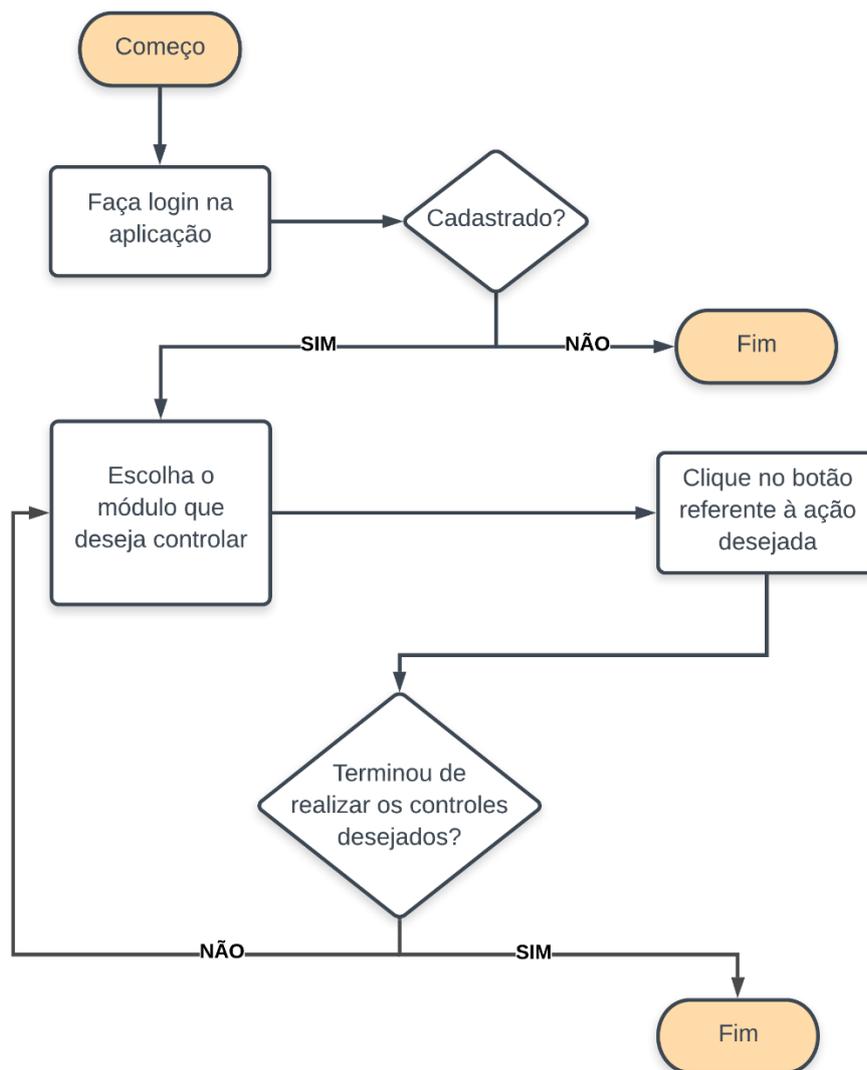
Figura 9 - Interface da aplicação *web*



Fonte: Print de tela gerado pelo autor

Um arquivo XML que é gerado no servidor Arduino no momento em que o microcontrolador é ligado tem o controle dinâmico do estado em que a lâmpada e a porta se encontram, ligado ou desligado, e faz com que a página carregue e seja atualizada conforme cada instante diferente. A figura 10 mostra o fluxograma de funcionamento da aplicação web construída:

Figura 10 - Fluxograma do funcionamento da aplicação web



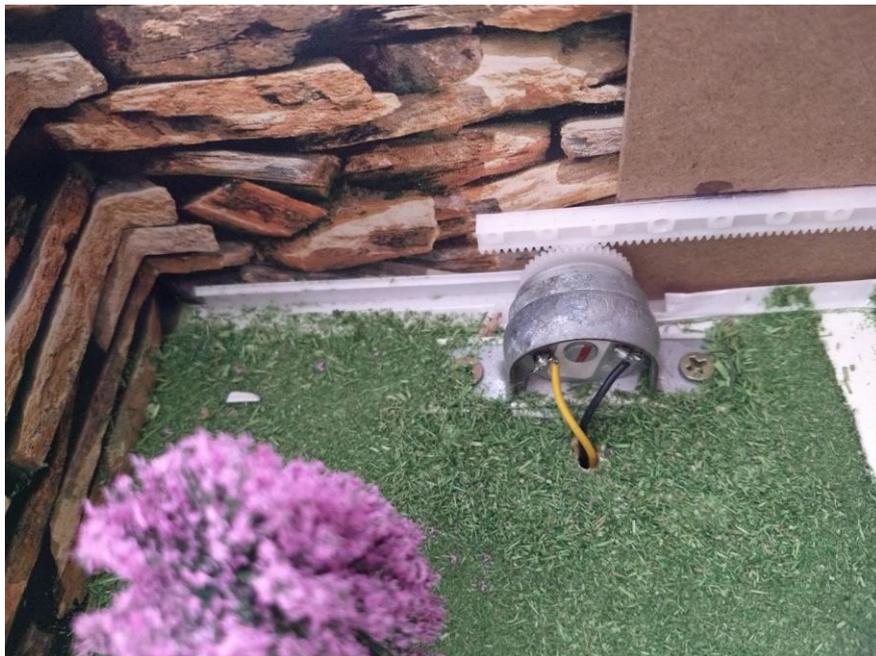
Fonte: Autor

3.3.3. Acionamento do portão automático

Para fazer o acionamento do portão automático, foi utilizado um motor retirado de um leitor e gravador velho de CD e DVD. Além disso, nessa parte do projeto foi

utilizado um módulo *driver* para motor com dupla ponte H L298N. Esse módulo é responsável por controlar o sentido de rotação do motor e faz o controle da velocidade de rotação através de um pino PWM do Arduino, que faz o controle de frequência do sinal com o objetivo de variar e controlar a informação disponível. A figura 11 mostra o motor conectado à estrutura da maquete.

Figura 11 - Motor acoplado à maquete



Fonte: Autor

Quando o comando de abertura do portão for enviado, o módulo com dupla ponte H L298N irá fazer com que o motor gire para um determinado lado, e o comando de fechamento irá girar o motor para o lado contrário. Ambos os movimentos acontecerão durante um tempo determinado para que o portão seja deslocado sobre a engrenagem do motor até que complete sua trajetória de um lado ao outro.

O controle de velocidade é feito através da função do Arduino *analogWrite()*. Essa função permite dois parâmetros. O primeiro é para informar qual pino PWM do Arduino está conectado ao pino do módulo do motor referente ao motor escolhido, pois esse módulo permite a conexão de dois motores ao mesmo tempo. No segundo parâmetro, é informado um número entre 0 e 255, pois usa uma informação de 8 bits para variar a saída do pino PWM. Com 0 a saída permanece em nível baixo e com

255 a saída permanece sempre em nível alto, enquanto com 128 de parâmetro, o nível fica em 50%.

O controle de rotação fica por conta da função *digitalWrite()*. No primeiro parâmetro é informado o número do pino do Arduino em que a entrada do módulo do motor está conectada e o segundo parâmetro recebe o nível lógico de acordo com a rotação que deseja executar. A tabela a seguir mostra como fica essa configuração:

Tabela 1 - Configuração dos pinos do módulo do motor

MOTOR	IN1	IN2
HORÁRIO	5V	GND
ANTI-HORÁRIO	GND	5V
PONTO MORTO	GND	GND
FREIO	5V	5V

Fonte: Site FILIPEFLOP (<https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-1298n/>, 2013, Acesso em out. 2019)

Os botões de acionamento do motor na interface central de controle e comandos são responsáveis por fazer requisições específicas na aplicação, fazendo com que ao clicar no primeiro botão, o portão seja aberto, e ao clicar no segundo botão, o portão seja fechado.

A alimentação do módulo do motor está vindo de uma fonte externa, uma fonte de 5V conectada na tomada. Dessa forma, é possível garantir que nenhum componente será sobrecarregado, o que causaria prováveis danos, além disso, evitar instabilidades no circuito.

3.3.4. Destravamento da porta

Para esse módulo, uma maquete foi construída separadamente. Trata-se de uma maquete que contém a porta, acompanhada do seu portal, além da trava elétrica para fazer o devido controle da abertura.

Para realizar o destravamento da porta é necessário usar a interface da central de controle e comandos, permitindo a entrada pela porta principal. Para quem se encontra do lado de dentro e deseja sair, é necessário acionar um *push button* localizado no lado interno da casa que irá acionar a trava elétrica e fará a liberação da porta. Entretanto, pode-se fazer uso também do botão disponível na interface central de controle e comandos para fazer a liberação da porta e permitir a saída.

A trava elétrica que mantém a porta fechada só é destravada quando uma tensão é aplicada nos seus terminais. Tanto o *push button* como o botão da central de controle e comandos mandam essa tensão necessária no momento em que são acionados. A figura 12 mostra como a trava elétrica está implementada na maquete da porta.

Figura 12 - Maquete da porta com a trava elétrica



Fonte: Autor

Para fazer o devido acionamento da trava elétrica é necessário utilizar uma fonte de alimentação externa, não sendo possível utilizar a tensão e a corrente provenientes do Arduino, pois ela demanda uma tensão de 12V para funcionar, sendo superior ao que o microcontrolador pode oferecer. Para isso, foi utilizada uma fonte de alimentação DC.

O módulo relé fica inicialmente em seu estado desligado. Quando o comando é enviado através do botão da interface ou através do *push button*, o canal do módulo

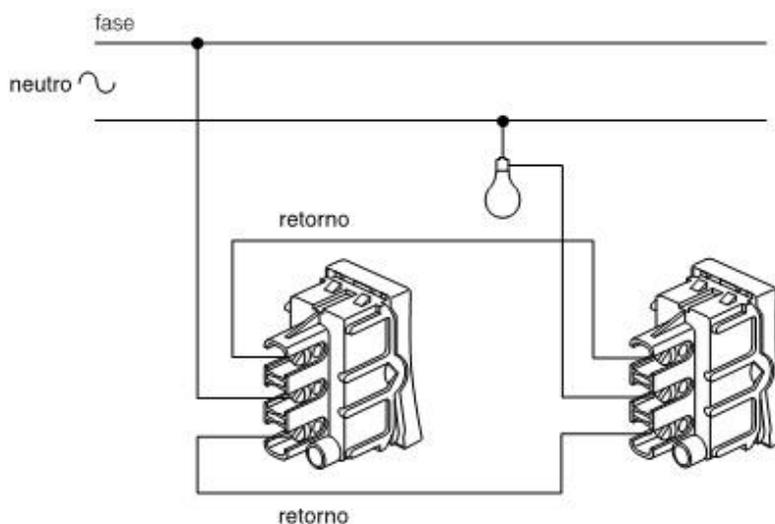
relé referente à trava elétrica tem seu estado alterado para ligado durante alguns segundos. Nesse momento, tensão e corrente são liberadas no circuito, a trava é acionada e a porta pode ser liberada. Após isso, o canal do relé volta para o estado desligado e todo o procedimento pode ser feito novamente.

3.3.5. Controle de iluminação

O controle de iluminação pode ser adicionado em qualquer cômodo da casa. Para realizar essa parte do protótipo, a lâmpada foi conectada em um dois canais do relé e no circuito de energia elétrica da residência para fornecer a alimentação necessária para o acionamento da lâmpada. Nessa parte, é possível adicionar um interruptor *three way* ligado em paralelo com o módulo relé para proporcionar tanto um acionamento local como um acionamento remoto.

Na figura 13 é possível ver como fica um circuito conectado com interruptores *three way*. Esse interruptor além de proporcionar um maior conforto para o usuário, aumenta os aspectos quanto a segurança, devido ao comando da iluminação estar em mais de um ponto.

Figura 13 - Conexão de interruptores *three way*



Fonte: Aldo Fernandes (<http://aldoalemao.blogspot.com/2012/10/three-way-interruptor-paralelo-o-que-e.html/>, 2012, Acesso em out. 2019)

Para acender ou apagar a lâmpada, o Arduino faz uso do mecanismo de chaves, o módulo relé, que está conectado aos pinos digitais número 3 e 9. De acordo com a requisição enviada pelo navegador de internet, o microcontrolador irá atuar e

enviará sinal alto ou baixo para as portas específicas, ativando ou não o relé que está controlando a lâmpada. A figura 14 mostra como a lâmpada está conectada à maquete.

Figura 14 - Lâmpada acoplada à maquete

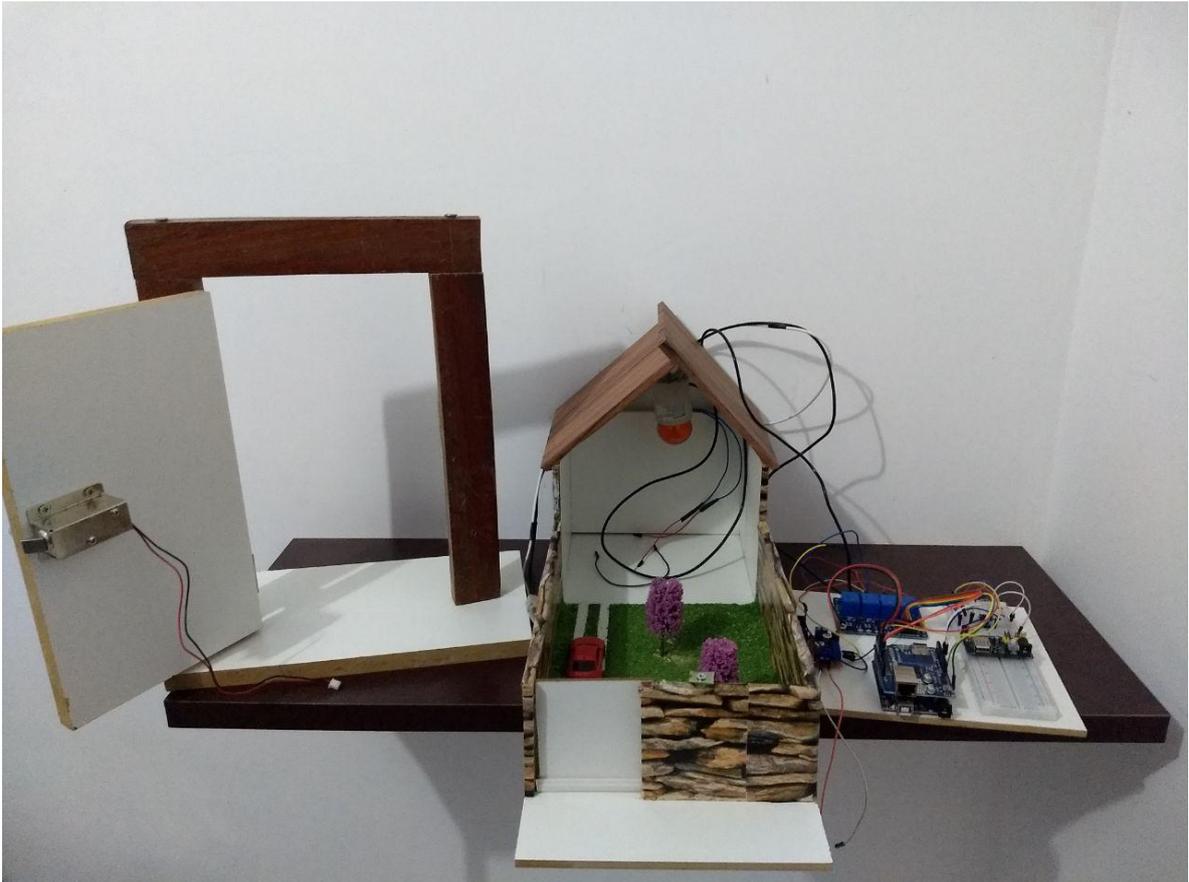


Fonte: Autor

A importância de ter uma segunda opção no momento de fazer o acionamento de lâmpadas, e não ter somente a opção de acender e apagar através do interruptor, é a mesma que as novas tecnologias de automação vem trazendo aos seus usuários: a ubiquidade. Analisando, virou hábito inquestionável o uso e a permanência constantemente próxima de seus smartphones, dessa maneira, surgiu a possibilidade de acender e apagar lâmpadas de uma maneira um pouco mais ágil e prática do que ter que ir até o interruptor e fazer o acionamento.

Com os procedimentos descritos anteriormente, a montagem do protótipo foi concluída, conforme pode ser observado na figura 15.

Figura 15 - Protótipo concluído com os componentes utilizados



Fonte: Autor

No próximo capítulo serão apresentados os resultados das simulações dos acionamentos e a demonstração de cada funcionamento.

CAPÍTULO 4: TESTES E RESULTADOS

4.1. Condições para os testes

Como se trata de uma prototipagem para fins acadêmicos, foram realizados apenas testes de implementação em escala reduzida. Porém, através de um estudo e avaliação é possível constatar sua viabilidade comercial e sua aplicação em ambientes heterogêneos, não apenas para automação de portões eletrônicos, portas e da iluminação de cômodos, mas de uma série de dispositivos e em diferentes tipos de ambientes, sejam eles residenciais, industriais e prediais.

Para o desenvolvimento desse projeto, foi considerada a decomposição do sistema em camadas, cada uma com suas respectivas responsabilidades e funções. A comunicação e a integração de todas essas camadas foram cruciais para o devido funcionamento do sistema, dependendo assim, de informações de todas as camadas para o sucesso do projeto.

Alguns testes foram realizados para verificar o funcionamento de todos os componentes do protótipo, tais como seguem a seguir:

4.1.1. Resultado 1 - Controle de iluminação

O primeiro resultado diz respeito ao sistema de controle de iluminação, onde é possível acender e apagar a lâmpada através de dispositivos móveis ou de computadores.

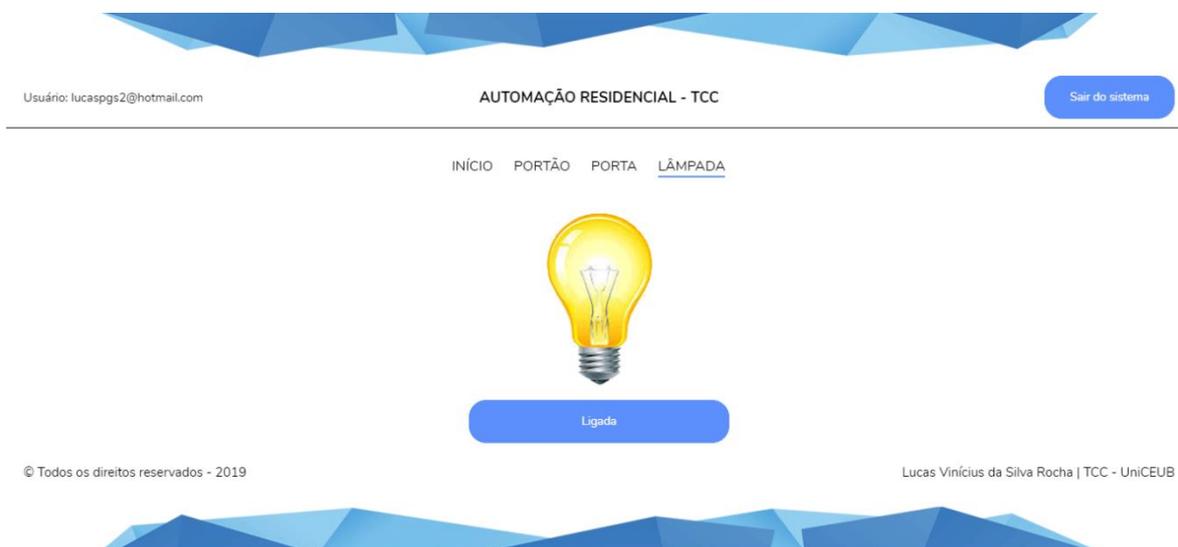
Inicialmente, para facilitar os testes, os acionamentos foram feitos utilizando-se um LED conectado em uma protoboard. Após verificar que seu funcionamento estava de acordo com as especificações, um bocal e uma lâmpada de tamanho e intensidade reduzidos foram utilizados nessa parte do projeto. A escolha da lâmpada pequena é para combinar melhor com a escala da maquete que foi construída.

Com a finalidade de ter uma segunda opção para o controle do estado da lâmpada, um interruptor *three way* pode ser ligado em paralelo com o respectivo canal do módulo relé. Apesar de tal possibilidade, esse recurso não foi adicionado ao projeto, pois como se trata de um protótipo, sua utilização não se fazia necessária. Porém, seu uso em uma situação real é completamente recomendável, além de que pode ser uma sugestão para trabalhos futuros, o que será feito com mais detalhes no próximo capítulo, no tópico de sugestões para trabalhos futuros.

Nos testes, foi possível identificar que o controle da lâmpada funcionava de acordo com os comandos enviados e que a página web era atualizada de acordo com o estado atual da lâmpada.

A figura 16 mostra a aplicação *web* com essas atualizações de estado da lâmpada:

Figura 16 - Página *web* do módulo de iluminação



Fonte: Print de tela gerado pelo autor

Não foram encontradas maiores dificuldades para a execução desse módulo do projeto e ele encontra-se funcionando de acordo com as especificações iniciais.

4. 1.2. Resultado 2 – Destravamento da porta

O segundo resultado obtido é referente ao acionamento de abertura da porta de entrada da casa. Os acionamentos iniciais dessa parte do projeto também contaram com o auxílio de um LED para facilitar os testes. Após verificação satisfatória dos resultados encontrados, a trava elétrica de 12V foi adicionada ao circuito e à sua maquete específica para demonstração.

Quando o comando é enviado, o módulo relé fica com sinal baixo, ou seja, ligado, durante poucos segundos, até que seu estado volte para alto, fique desligado, e o mecanismo da trava possa ser recolhido para a porta ser trancada novamente.

O módulo referente à porta na interface *web* também conta com uma atualização automática do estado de abertura da porta. Através da interface, é possível obter uma resposta em tempo real da situação da trava, sabendo quando o botão foi acionado e

se o mecanismo foi de fato recolhido realizando a abertura da porta. Da mesma forma, é possível saber também quando o mecanismo fica pronto para travar a porta novamente.

A figura 17 mostra como a aplicação *web* controla o acionamento de abertura da porta.

Figura 17 - Página *web* do módulo de destravamento da porta



Fonte: Print de tela gerado pelo autor

Não foram encontradas maiores dificuldades para a execução desse módulo do projeto e ele encontra-se funcionando de acordo com as especificações iniciais, porém durante os testes, o módulo de destravamento da porta apresentou alguns problemas de atraso de resposta após alguns acionamentos consecutivos. Inicialmente, quando era ligado, funcionava perfeitamente, porém após cerca de 5 acionamentos consecutivos, o sistema apresentou atraso nas respostas somente nesse módulo do projeto.

4. 1.3. Resultado 3 – Acionamento do portão automático

O acionamento do portão automático foi o módulo que deu mais trabalho para implementar. Alguns problemas foram encontrados no decorrer dessa parte do projeto. O primeiro deles diz respeito à dificuldade de encontrar uma engrenagem e uma cremalheira que fossem compatíveis com o tamanho do motor utilizado.

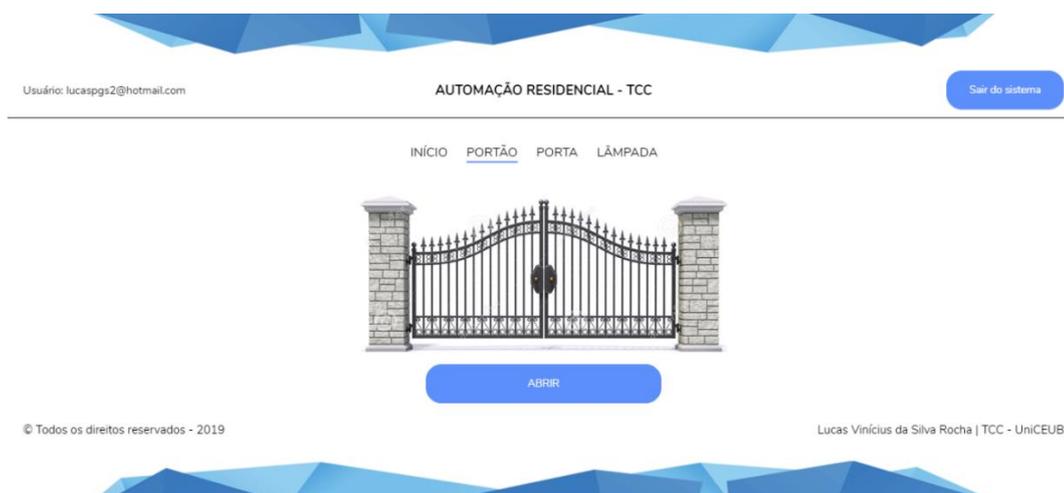
Tais componentes são necessários para fazer o motor conseguir movimentar o portão da maquete. Todas as lojas buscadas só vendiam um *kit* fechado de alto valor com vários componentes que não seriam utilizados, e por isso a compra não fazia sentido em um primeiro momento. Esse problema teria sido evitado se tais lojas vendessem a unidade dessas peças ou se fosse uma situação de aplicação real do projeto, onde todas essas peças são encontradas muito facilmente.

O segundo problema aconteceu no momento de acionar o módulo *driver* ponte H L298N após a conexão com o Ethernet Shield. Nenhum teste de acionamento funcionava nesse momento e foram realizadas inúmeras tentativas diferentes até que a causa do problema fosse percebida.

O problema é que os pinos de ativação do módulo L298N estavam sempre recebendo um sinal lógico alto, quando não deveriam. O código do Arduino tinha sido desenvolvido para iniciar o programa com os pinos em estado lógico baixo e deveriam alternar seus estados de acordo com o momento de acionamento do motor.

Estavam sendo utilizadas as portas 0 e 1 do Arduino para conexão com o módulo, pois já era conhecido que as portas 4, 10, 11, 12 e 13 já ficam automaticamente ocupadas quando se utiliza um Ethernet Shield e também porque as portas 2, 3, 8 e 9 estavam sendo utilizadas para a conexão com os outros módulos do sistema. Porém, depois de fazer testes com as portas 0 e 1, foi possível perceber que ambas realmente sempre mandam um sinal lógico alto por estarem ocupadas da mesma forma. Uma pesquisa sobre a utilização das portas do Ethernet Shield comprovou o fato observado, tais portas são utilizadas para o TX e RX.

Figura 18 - Página *web* do módulo de acionamento do motor



Fonte: Print de tela gerado pelo autor

O que parecia ser um problema de programação ou de configuração do circuito, era apenas uma necessidade de trocar as portas que estavam sendo utilizadas por portas que estavam disponíveis, para que o acionamento do motor através do módulo fosse possível. Bastou conectar tais pinos nas portas 6 e 7 do Arduino para que o problema fosse resolvido. Dessa forma, o sentido e a velocidade de rotação do motor puderam ser controlados da maneira em que era esperado.

4.2. Comparação dos resultados obtidos

Como resultado da finalização do protótipo, infere-se a possibilidade de acionar atuadores para abrir e fechar portões eletrônicos, fazer o controle de entrada em portas, apagar ou acender lâmpadas, entre outras coisas. Tal resultado pode ser obtido fazendo uso de um conjunto de *hardwares* (sensores, atuadores, microcontroladores) e *softwares*, através de um dispositivo móvel ou um computador, possibilitando uma interação entre o homem e a máquina e gerando uma comunicação com o Arduino, interpretando mensagens e acionando o relé correspondente de cada módulo do projeto.

4.3. Custos do projeto

Uma série de componentes foi utilizada neste projeto. Como sugestão para redução de custos, é possível citar o fato da plataforma Arduino ser *Open-Hardware*, ou seja, a montagem encontra-se disponível no próprio site da fabricante. Da mesma maneira, para reduzir custos, a compra de alguns itens pode ser feita através da importação de outros países, dispensando a compra em mercado nacional, o que pode baratear o custo do projeto.

Antes de realizar a compra dos materiais, uma pesquisa de preços foi realizada em diferentes sites e lojas de produtos do gênero, com o propósito de encontrar o menor preço e oferecer um custo menor para o desenvolvimento e construção do projeto.

Na tabela 2 são apresentados os materiais utilizados para a montagem do protótipo, além de seus custos de compra no comércio local.

Tabela 2 - Descrição dos componentes e materiais utilizados e seus respectivos custos

PRODUTO	PREÇO
Arduino UNO com ATmega328 (SMD)	R\$ 44,29
Shield Ethernet/Rede - WIZnet W5100	R\$ 88,58
<i>Protoboard</i> 830 Pontos	R\$ 16,54
Módulo de Relés 5V - 4 Canais	R\$22,09
<i>Jumpers</i> Macho-Macho	R\$ 11,90
<i>Jumpers</i> Macho-Fêmea	R\$ 9,88
Módulo <i>Driver</i> Motor com Dupla Ponte H - L298N	R\$ 18,32
Fonte de Alimentação 5V para <i>Protoboard</i>	R\$ 7,66
Fonte de Alimentação 12V DC 110/220V 1,5A com <i>Plug</i> P4	R\$ 18,76
LED Difuso Vermelho e Verde (5mm)	R\$ 0,27
<i>Kit</i> de Engrenagens	R\$ 43,23
Árvores em Miniatura	R\$ 22,50
Pó de Serragem	R\$ 7,80
Ferragens para Montagem da Maquete	R\$ 7,50
Mini Trava Elétrica – 12V	R\$ 40,90
Fonte DC – 9V	R\$ 12,00
Adaptador para Fonte	R\$ 5,00
Adesivo de Decoração	R\$ 13,35
Mini Lâmpada	R\$ 5,00
Bocal de Cerâmica	R\$ 5,00
CUSTO TOTAL	R\$ 400,57

CAPÍTULO 5: CONCLUSÃO

5.1. Conclusão

As oportunidades tecnológicas oferecidas atualmente, discutidas e detalhadas ao longo desse trabalho, fizeram com que o mercado de automação residencial fosse alavancado. O cenário abordado ao longo da pesquisa traz como evidência a necessidade e os benefícios oferecidos dos sistemas de automação residencial. Tal contexto nos mostra que a automação trata de um conjunto de recursos e ferramentas que promovem a praticidade, o bem-estar, conforto e economia para quem a utiliza, maximizando as interações entre os moradores e as suas residências.

É possível perceber que a plataforma Arduino pode ser uma boa solução para desenvolver sistemas de automação residencial. O projeto desenvolvido durante esse trabalho consegue atender perfeitamente os requisitos básicos de um sistema de controle de residências. Todos os objetivos foram atingidos dentro do que foi proposto, sendo validados através de uma arquitetura adequada e da integração entre os códigos desenvolvidos.

A escalabilidade é uma característica importante desse projeto, pois poder acrescentar novos ambientes e módulos de controle é um recurso que traz grandes vantagens, como pode ser analisado no código *sketch* do Arduino e no código da aplicação *web*.

A principal dificuldade do projeto foi a de realizar efetivamente a devida comunicação entre a aplicação *web* e o microcontrolador Arduino, além disso, manter a página da aplicação atualizada com o estado de cada módulo automatizado, demandou uma programação que fosse capaz de garantir o envio correto das informações de um lado para o outro. Tal dificuldade foi superada depois de realizar várias pesquisas, recorrendo às bibliografias e tutoriais específicos.

5.2. Sugestões para trabalhos futuros

A primeira sugestão é a ideia de criar um aplicativo para Android e iOS para o controle do sistema de automação. Gerando a possibilidade de controle através de um aplicativo que rode nos dois principais sistemas operacionais do mercado de celulares.

Como já mencionado, o interruptor *three way* pode ser adicionado em paralelo com o canal do módulo relé responsável pelo controle da lâmpada, tanto para demonstração no protótipo quanto em um sistema aplicado em uma situação real.

Outra sugestão é a de passar o protótipo para uma placa de circuito impresso, eliminando o mal contato da protoboard e gerando um circuito mais organizado e sem o risco de desconexão dos *jumpers*, por exemplo.

A última proposta é a de amplificar a automação nas outras partes da casa e gerar mais funções no sistema já implementado, adicionando novos equipamentos conectados ao Arduino, como por exemplo:

- Sistema de detecção de presença;
- Sistema de detecção e de alarme para casos de vazamento de gás inflamável, como o gás butano.
- Sistema de verificação de umidade e temperatura do ambiente.

REFERÊNCIAS

OLIVEIRA NETO, Reinaldo. Automação de iluminação residencial utilizando microcontrolador Arduino e tablet/iPad via WI-FI. 2011. Trabalho de Conclusão de Curso (Graduação - Engenharia da Computação) - UniCEUB - Centro Universitário de Brasília, Brasília - DF, 2011. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/123456789/3197/2/20615176.pdf/>. Acesso em: 17 abr. 2019.

CAMPOS, Roberto Augusto Freitas. Automação residencial utilizando Arduino e aplicação web. 2014. Trabalho de Conclusão de Curso (Graduação - Engenharia da Computação) - UniCEUB - Centro Universitário de Brasília, Brasília - DF, 2014. Disponível em: https://repositorio.uniceub.br/jspui/bitstream/235/5461/1/Monografia_Roberto.pdf/. Acesso em: 17 abr. 2019.

CRUZ, José Carlos da Silva Santa. Automação residencial utilizando smartphone Android, bluetooth e conversor RS-485. 2013. Trabalho de Conclusão de Curso (Graduação – Engenharia da Computação) - UniCEUB - Centro Universitário de Brasília, Brasília - DF, 2013. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/4909/1/20663730.pdf/>. Acesso em: 17 abr. 2019.

STRAUB, Matheus Gebert. Automação residencial Arduino: ideias para deixar sua casa igual à do Homem de Ferro. [S. l.], 5 abr. 2017. Disponível em: <https://blog.usinainfo.com.br/automacao-residencial-arduino-ideias-para-deixar-sua-casagual-homem-de-ferro/>. Acesso em: 17 abr. 2019.

MADEIRA, Daniel. Shield Ethernet W5100 - Criando um servidor web com Arduino. [S. l.], 13 nov. 2017. Disponível em: <https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web/>. Acesso em: 22 abr. 2019.

RODRIGUES, Marcelo. Tutorial: Controlando Relés via Wi-Fi e pela Internet. [S. l.], 26 fev. 2016. Disponível em: <http://labdegaragem.com/profiles/blogs/tutorial-controlando-reles-viawi-fi-e-pela-internet/>. Acesso em: 22 abr. 2019.

LENZ, André Luis. A Pré-História da Domótica. [S. l.], 27 dez. 2012. Disponível em: <http://info-domotica.blogspot.com/2012/12/a-pre-historia-da-domotica.html/>. Acesso em: 19 ago. 2019.

FORTI, José Cândido, AURESIDE, Principais Protocolos e Padrões Usados em Automação Residencial, Congresso BICSI 2001, São Paulo, 2001.

PORTUGAL, Arduino. O que é o Arduino UNO?. [S. l.], 14 abr. 2017. Disponível em: <https://www.arduinoportugal.pt/o-que-e-o-arduino/>. Acesso em: 31 ago. 2019.

TELES, Elaine. Arduino: O que é? Para que serve? Quais as possibilidades?. [S. l.], 22 set. 2016. Disponível em: <https://medium.com/nossa-coletividade/arduino-o-que-%C3%A9-para-que-serve-quais-as-possibilidades-efbd59d33491/>. Acesso em: 31/ ago. 2019.

OLIVEIRA, Euler. Como usar com Arduino – Ethernet Shield W5100 (Web server). [S. l.], 01 out. 2018. Disponível em: <http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-ethernet-shield-w5100-via-web-server/>. Acesso em: 01 set. 2019.

MOTA, Allan. Módulo relé – Acionando cargas com Arduino. [S. l.], 21 nov. 2017. Disponível em: <https://portal.vidadesilicio.com.br/modulo-rele-com-arduino/>. Acesso em: 01 set. 2019.

BAUERMEISTER, Giovanni. Automação de lâmpadas com Módulo Relé Arduino. [S. l.], 08 out. 2018. Disponível em: <https://blog.fazedores.com/automacao-de-lampadas-modulo-rele-arduino/>. Acesso em: 01 set. 2019.

MADEIRA, Daniel. Trava elétrica solenoide com Arduino. [S. l.], 19 jun. 2018. Disponível em: <https://portal.vidadesilicio.com.br/trava-eletrica-solenoide/>. Acesso em: 03 set. 2019.

TEZA, Vanderlei Rabelo. Alguns Aspectos Sobre a Automação Residencial - Domótica. 2002. Trabalho de Conclusão de Curso (Mestrado - Ciência da Computação) - UFSC - Universidade de Santa Catarina, Florianópolis - SC, 2002. Disponível em:

<https://repositorio.ufsc.br/bitstream/handle/123456789/83015/212312.pdf/>. Acesso em: 06 set. 2019.

VIDAL, Vitor. Ethernet Shield W5100 com Arduino – Parte 1. [S. l.], 04 jun. 2019. Disponível em: <https://blog.eletrogate.com/ethernet-shield-w5100-com-arduino/>. Acesso em: 06 set. 2019.

CARDOSO, Daniel. Driver motor com Ponte H L298N – Controlando Motor DC com Arduino. [S. l.], 13 mai. 2017. Disponível em: <https://portal.vidadesilicio.com.br/driver-motor-com-ponte-h-l298n/>. Acesso em: 06. out. 2019.

THOMSEN, Adilson. Motor DC com Driver Ponte H L298N. [S. l.], 14 mar. 2013. Disponível em: <https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>. Acesso em: 06. out. 2019.

DE CASTRO, Giovanni. Controlando Motores com o Módulo L298N. [S. l.] 21 abr. 2015. Disponível em: <https://www.robocore.net/tutoriais/243/>. Acesso em: 06. out. 2019.

SOUZA, Fábio. Arduino UNO. [S. l.] 29 nov. 2013. Disponível em: <https://www.embarcados.com.br/arduino-uno/>. Acesso em 21. out. 2019.

VIDAL, Vitor. Módulo Relé para automação residencial com Arduino. [S. l.] 21 de abr. 2018. Disponível em: <https://blog.eletrogate.com/modulo-rele-para-automacao-residencial-com-arduino/>. Acesso em 21. out. 2019

FERNANDES, Aldo. Three-Way (Interruptor Paralelo): O que é? Como funciona?. [S. l.] 16 out. 2012. Disponível em: <http://aldoalemao.blogspot.com/2012/10/three-way-interruptor-paralelo-o-que-e.html/>. Acesso em 26. out. 2019.

APÊNDICE A - APLICAÇÃO WEB

```

<html lang="pt-br">
<meta charset="utf-8"/>
<head>
  <title>TCC - UniCEUB</title>
  <link rel="shortcut icon" type="image/x-icon" href="http://arduino-
tcc.atwebpages.com/imagens/favicon.ico">
  <link href="https://fonts.googleapis.com/css?family=Nunito:400,600,700"
rel="stylesheet">
  <link rel="stylesheet" href="http://arduino-tcc.atwebpages.com/css/estilo.css"/>
  <link rel="stylesheet" href="http://arduino-tcc.atwebpages.com/css/menu.css"/>
  <link rel="stylesheet" href="http://arduino-tcc.atwebpages.com/css/responsividade.css"/>
  <script src="http://arduino-tcc.atwebpages.com/javascript/funcoes_interface.js"
charset="UTF-8"></script>
  <script src="http://arduino-tcc.atwebpages.com/javascript/funcoes_arduino.js"
charset="UTF-8"></script>

  <script type="text/javascript">
    function submitEnter(myfield,e) {
      var keycode;
      if (window.event) keycode = window.event.keyCode;
      else if (e) keycode = e.which;
      else return true;
      if (keycode == 13) {
        login();
        return false;
      }
      else {
        return true;
      }
    }
  </script>

</head>
<body onload="GetArduinoIO()">

  <div class="imagem-topo" id="imagem-topo"></div>

  <div id="login_div" class="div-principal">
    <h1>AUTOMAÇÃO RESIDENCIAL</h1>
    <h2>Trabalho de Conclusão de Curso</h2>
    <h2>Lucas Vinícius da Silva Rocha</h2>
  </div>

```

```

<h2>Engenharia da Computação</h2>
<input type="email" onKeyPress="return submitEnter(this,event)" placeholder="E-mail"
id="campo_email" />
<input type="password" onKeyPress="return submitEnter(this,event)"
placeholder="Senha" id="campo_senha" />
<button onclick="login()">Fazer login</button>
</div>

<div id="alerta" class="alerta">
</div>

<div id="user_div" class="div-login">
<h1>AUTOMAÇÃO RESIDENCIAL - TCC</h1>
<div class="usuario">
<p id="usuario"></p>
</div>
<button onclick="logout()">Sair do sistema</button>
</div>

<div id="conteudo" class="conteudo">
<nav id="menu">
<ul>
<li><a onclick="inicio_menu()" id="inicio_menu">INÍCIO</a></li>
<li><a onclick="portao_menu()" id="portao_menu">PORTÃO</a></li>
<li><a onclick="porta_menu()" id="porta_menu">PORTA</a></li>
<li><a onclick="lampada_menu()" id="lampada_menu">LÂMPADA</a></li>
<li><a onclick="temperatura_menu()" id="temperatura_menu"></a></li>
</ul>
</nav>
<div class="inicio" id="inicio">

<h3>Escolha um dispositivo no menu para controlar</h3>
</div>
<div class="portao" id="portao">
<p class="estado_portao" id="estado_portao">0</p>
<h3></h3>


<h3></h3>
<button id="abre_portao" onclick="abre_portao()">ABRIR</button>
<button id="fecha_portao" onclick="fecha_portao()">FECHAR</button>
<button id="para_portao" onclick="para_portao()">PARAR</button>
<div id="tempo_portao" class="tempo_portao">

```

```

    <h3 id="segundos_portao"></h3>
  </div>
</div>
<div class="porta" id="porta">
  
  
  <h3></h3>
  <div id="tempo_porta" class="tempo_porta">
    <h3>PORTA DESTRAVADA</h3>
    <h3 id="segundos_porta"></h3>
  </div>
  <button
    type="button"
    id="porta_botao"
    onclick="controla_porta()">DESTRAVAR</button>
</div>
<div class="lampada" id="lampada">
  
  
  <h3></h3>
  <button
    type="button"
    id="botao_lampada"
    onclick="controla_lampada()">Desligada</button>
</div>
</div>
<div class="rodape" id="rodape">
  <div id="texto">
    <div id="direitos">
      <script>
        var mydate = new Date()
        var year = mydate.getFullYear()
        if (year < 1000)
          year += 1900
        document.write("© Todos os direitos reservados - " + year)
      </script>
    </div>
    <div id="titulo">
      Lucas Vinícius da Silva Rocha | TCC - UniCEUB
    </div>
  </div>
</div>
<script src="https://www.gstatic.com/firebasejs/5.10.1/firebase.js"></script>
<script>
  var config = {

```

```
apiKey: "AIzaSyAeuD9JFjnAEeYY6QmQMbrHUOcSEZFeltI",
authDomain: "myapp-ba085.firebaseio.com",
databaseURL: "https://myapp-ba085.firebaseio.com",
projectId: "myapp-ba085",
storageBucket: "myapp-ba085.appspot.com",
messagingSenderId: "1071431276112"
};
firebase.initializeApp(config);
</script>
<script src="http://arduino-tcc.atwebpages.com/javascript/autenticacao_firebase.js"
charset="UTF-8"></script>
<script>controla_lampada()</script>
<script>estado_portao()</script>
</body>
</html>
```

APÊNDICE B - *SKETCH* ARDUINO

```

#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#define REQ_BUF_SZ 60
char BOTA_POR = 2;
char BOTA_LAM = 3;
char PINO_ENA = 5;
char PINO_IN1 = 6;
char PINO_IN2 = 7;
char PINO_POR = 8;
char PINO_LAM = 9;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip (192, 168, 1, 100);
EthernetServer server (80);
File webFile;
char HTTP_req[REQ_BUF_SZ] = {0};
char req_index = 0;
boolean estado[4] = {1};

void setup() {
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);
  Serial.begin(9600);
  Serial.println(F("Iniciando cartão SD..."));
  if (!SD.begin(4)) {
    Serial.println(F("ERRO - A inicialização do cartão SD falhou."));
    return;
  }
  Serial.println(F("SUCESSO - Cartão SD inicializado."));
  if (!SD.exists(F("index.htm"))){
    Serial.println(F("ERRO - Não foi possível encontrar o arquivo index.htm."));
  }
}

```

```

    return;
}
Serial.println(F("SUCESSO - Arquivo index.htm encontrado."));

pinMode(BOTA_POR, INPUT);
pinMode(BOTA_LAM, INPUT);
pinMode(PINO_POR, OUTPUT);
pinMode(PINO_LAM, OUTPUT);
pinMode(PINO_ENA, OUTPUT);
pinMode(PINO_IN1, OUTPUT);
pinMode(PINO_IN2, OUTPUT);

digitalWrite(PINO_IN1, LOW);
digitalWrite(PINO_IN2, LOW);
digitalWrite(PINO_ENA, LOW);

Ethernet.begin(mac, ip);
server.begin();
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (req_index < (REQ_BUF_SZ - 1)) {
          HTTP_req[req_index] = c;
          req_index++;
        }
        if (c == '\n' && currentLineIsBlank) {
          client.println("HTTP/1.1 200 OK");
          if (StrContains(HTTP_req, "ajax_inputs")) {

```

```
client.println("Content-Type: text/xml");
client.println("Connection: keep-alive");
client.println();
setDispositivos();
XML_response(client);
}
else {
client.println("Content-Type: text/html");
client.println("Connection: keep-alive");
client.println();
webFile = SD.open("index.htm");
if (webFile) {
while (webFile.available()) {
client.write(webFile.read());
}
webFile.close();
}
}
Serial.print(HTTP_req);
req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
if (c == '\n') {
currentLineIsBlank = true;
}
else if (c != '\r') {
currentLineIsBlank = false;
}
}
}
delay(1);
client.stop();
}
```

```

    ButtonDebounce();
}

void ButtonDebounce(void) {
    static byte buttonState[2] = {LOW, LOW};
    static byte lastButtonState[2] = {LOW, LOW};
    static char lastDebounceTime[2] = {0};
    char debounceDelay = 50;
    byte reading[2];
    reading[0] = digitalRead(BOTA_POR);
    reading[1] = digitalRead(BOTA_LAM);
    for (char i = 0; i < 2; i++) {
        if (reading[i] != lastButtonState[i]) {
            lastDebounceTime[i] = millis();
        }
        if ((millis() - lastDebounceTime[i]) > debounceDelay) {
            if (reading[i] != buttonState[i]) {
                buttonState[i] = reading[i];
                if (buttonState[i] == HIGH) {
                    estado[i] = !estado[i];
                }
            }
        }
    }
    digitalWrite(PINO_POR, estado[0]);
    digitalWrite(PINO_LAM, estado[1]);
    lastButtonState[0] = reading[0];
    lastButtonState[1] = reading[1];
}

void setDispositivos(void) {
    //Porta
    if (StrContains(HTTP_req, "Porta=1")) {
        estado[0] = 1; //Salva o estado da porta
    }
}

```

```
digitalWrite(PINO_POR, HIGH);
}
else if (StrContains(HTTP_req, "Porta=0")) {
    estado[0] = 0; //Salva o estado da porta
    digitalWrite(PINO_POR, LOW);
}
//Lâmpada
if (StrContains(HTTP_req, "Lampada=1")) {
    estado[1] = 1; //Salva o estado da lâmpada
    digitalWrite(PINO_LAM, HIGH);
}
else if (StrContains(HTTP_req, "Lampada=0")) {
    estado[1] = 0; //Salva o estado da lâmpada
    digitalWrite(PINO_LAM, LOW);
}
//Motor
if (StrContains(HTTP_req, "Motor=1")) {
    digitalWrite(PINO_IN1, HIGH);
    digitalWrite(PINO_IN2, LOW);
    analogWrite(PINO_ENA, 245);
    delay(7000);
    digitalWrite(PINO_IN1, LOW);
    digitalWrite(PINO_IN2, HIGH);
    digitalWrite(PINO_ENA, LOW);
}
else if (StrContains(HTTP_req, "Motor=0")) {
    digitalWrite(PINO_IN1, HIGH);
    digitalWrite(PINO_IN2, LOW);
    analogWrite(PINO_ENA, 245);
    delay(7000);
    digitalWrite(PINO_IN1, LOW);
    digitalWrite(PINO_IN2, LOW);
    digitalWrite(PINO_ENA, LOW);
}
```

```

}

void XML_response(EthernetClient cl) {
    char analog_val;
    char count;
    char sw_arr[] = {BOTA_POR, BOTA_LAM};
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    cl.print("<DISPOSITIVO>");
    if (estado[0]) {
        cl.print("ligado");
    }
    else {
        cl.print("desligado");
    }
    cl.println("</DISPOSITIVO>");
    cl.print("<DISPOSITIVO>");
    if (estado[1]) {
        cl.print("ligado");
    }
    else {
        cl.print("desligado");
    }
    cl.println("</DISPOSITIVO>");
    cl.print("</inputs>");
}

```

```

void StrClear(char *str, char length) {
    for (char i = 0; i < length; i++) {
        str[i] = 0;
    }
}

```

```

char StrContains(char *str, char *sfind) {

```

```
char found = 0;
char index = 0;
char len;
len = strlen(str);
if (strlen(sfind) > len) {
    return 0;
}
while (index < len) {
    if (str[index] == sfind[found]) {
        found++;
        if (strlen(sfind) == found) {
            return 1;
        }
    }
    else {
        found = 0;
    }
    index++;
}
return 0;
}
```