

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COINT - TECNOLOGIA EM SISTEMAS PARA INTERNET
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

FELIPE KOSOUSKI

SMART LIGHTING: SOLUÇÃO DE LÂMPADAS PARA IOT

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2017

FELIPE KOSOUSKI

SMART LIGHTING: SOLUÇÃO DE LÂMPADAS PARA IOT

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Hermano Pereira
Universidade Tecnológica Federal do Paraná

GUARAPUAVA
2017

RESUMO

KOSOUSKI, Felipe. Smart Lighting: Solução de lâmpadas para IOT. 2017. 30 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2017.

Vivemos em uma época onde as conexões e a rede fazem cada vez mais parte do nosso cotidiano que chegamos a estar a todo momento conectados com pessoas e dispositivos. O avanço da tecnologia tem se expandido para as mais diversas áreas, sendo uma delas, as casas inteligentes, parte de uma área maior ainda chamada Internet das Coisas. Com uma grande variedade de sensores e dispositivos conectados, um usuário ou até mesmo um sensor pode emitir um sinal para controlar diversas coisas. Uma das áreas de estudo que provém das casas inteligentes é a chamada *Smart Lighting*, ou seja, o controle de luzes e lâmpadas pela rede. Dentre os equipamentos hoje utilizados, a maioria possui um custo elevado devido a mão de obra necessária para instalação e são inviáveis do ponto de vista físico. Este projeto visa desenvolver uma solução para controle de lâmpadas via rede utilizando um microcontrolador Arduino e um Web Service responsável por trabalhar com os dados enviados e recebidos, sendo essencialmente, um sistema de fácil instalação e configuração.

Palavras-chave: Microcontroladores. Internet das Coisas. Arduino. CoAP. Smart Lighting.

ABSTRACT

KOSOUSKI, Felipe. Smart Lighting: Lamp solution for IOT. 2017. 30 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2017.

We are living in a time where the network and connections has increasingly became part of our daily life that we get to be connected all the time, both to people and devices. The advance in technology has expanded in such a manner to the most diverse areas, being one of them, the smart houses, part of a even larger area called Internet of Things. With a wide variety of interconnected sensors and devices, a user, or even a sensor itself, can send a signal to control various "things". One of the study areas that comes from the intelligent house idea is called Smart Lighting, that is, the control lights and lamps through the network. Among the equipment used nowadays, most of them have a high cost due to labor required for installation and are physically unviable. This project aims to develop a solution to control lamps via network using an Arduino microcontroller and a Web Service responsible for working with the sent and received data, being essentially a system of easy installation and configuration.

Keywords: Microcontrollers. Internet of Things. Arduino. CoAP. Smart Lighting.

LISTA DE FIGURAS

Figura 1 – Internet das coisas	4
Figura 2 – Smart Home	6
Figura 3 – Arduino MEGA 2560	8
Figura 4 – Shield Ethernet W5100	9
Figura 5 – Módulo relé	10
Figura 6 – Características do protocolo CoAP	12
Figura 7 – Esquema de funcionamento de uma página web em PHP	14
Figura 8 – Web Server acendendo lâmpadas	16
Figura 9 – Inserindo um dispositivo no Lelylan	17
Figura 10 – Kit Smart Lighting Ikea	18
Figura 11 – Circuito de componentes	21
Figura 12 – Montagem do Shield no Arduino	22
Figura 13 – Protótipo de funcionamento	23
Figura 14 – Execução inicial do Riot OS no Arduino	25
Figura 15 – Circuito físico finalizado	26

LISTA DE TABELAS

Tabela 1 – Comparativo entre alguns dos sistemas operacionais mais conhecidos para IoT.	11
---	----

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
API	Application Programming Interface
CBOR	Concise Binary Object Representation
CSS3	Cascading Style Sheets 3
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
HTML	HyperText Markup Language
IDE	Integrated Development Environment
IPv6	Internet Protocol Version 6
IoT	Internet of Things
M2M	Machine To Machine
MQTT	Message Queue Telemetry Transport
PHP	Hypertext Preprocessor
REST	Representational State Transfer
RFID	Radio Frequency Identification
RPL	IPv6 Routing Protocol for Low Power and Lossy Network
UDP	User Datagram Protocol
UHD	Ultra High Definition
URI	Uniform Resource Identifier
USB	Universal Serial Bus
W3C	World Wide Web Consortium

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 ORGANIZAÇÃO DO TRABALHO	2
1.2 OBJETIVOS	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivos Específicos	2
2 – REVISÃO DA LITERATURA	4
2.1 INTERNET DAS COISAS	4
2.2 SMART HOME	5
2.2.1 GESTÃO E EFICIÊNCIA ENERGÉTICA	6
2.2.2 SAÚDE	6
2.2.3 ENTRETENIMENTO	7
2.2.4 SEGURANÇA	7
2.3 SMART LIGHTING	7
2.4 FUNDAMENTAÇÃO TEÓRICA	8
2.4.1 ARDUINO MEGA 2560	8
2.4.2 SHIELD ETHERNET W5100	8
2.4.3 MÓDULO RELÉ	9
2.4.4 RIOT OS	10
2.4.5 COAP	11
2.4.6 HTML	12
2.4.7 WEB SERVICE REST	13
2.4.8 PHP	13
2.4.9 CSS	14
2.5 TRABALHOS RELACIONADOS	15
2.5.1 FilipeFlop - Acendendo Lâmpadas pela Internet	15
2.5.2 Lelylan - IOT Cloud Platform	16
2.5.3 IKEA Trådfri - Smart Lamps	17
2.5.4 DIFERENCIAL TECNOLÓGICO	18
3 – METODOLOGIA	19
3.1 DEFINIÇÃO DAS TECNOLOGIAS	19
3.2 ESTUDO DAS TECNOLOGIAS	19
3.3 ELABORAÇÃO E MONTAGEM DO CIRCUITO FÍSICO	19
3.4 DESENVOLVIMENTO DO SERVIDOR E APLICAÇÃO	20
3.5 DESENVOLVIMENTO DO WEB SERVICE	20

3.6	REALIZAÇÃO DE TESTES	20
3.7	DESENVOLVIMENTO	20
3.7.1	HARDWARE - PARTE FÍSICA	21
3.7.1.1	ARDUINO	21
3.7.1.2	ETHERNET SHIELD	22
3.7.1.3	RELÉ	22
3.7.1.4	LÂMPADA	23
3.7.2	SOFTWARE - PARTE LÓGICA	23
3.7.2.1	RIOT OS	24
3.7.2.2	CoAP	24
3.7.2.3	APLICAÇÃO RIOT OS E ARDUINO	24
3.7.2.4	WEB SERVICE	24
3.8	DESENVOLVIMENTO PRELIMINAR	25
4	– CONCLUSÃO	27
4.1	CONSIDERAÇÕES FINAIS	27
	Referências	28

1 INTRODUÇÃO

A sociedade atual está passando por uma grande mudança tecnológica, estamos entrando aos poucos na era da *Internet of Things*, em tradução direta, Internet das Coisas. Essa mudança será uma revolução no que diz respeito não somente a como nós, pessoas, interagiremos com *gadgets* e objetos que tenham capacidade de se conectar a Internet, mas também, como os próprios objetos interagirão entre si. A IoT, fará com que tais objetos, aliados a diversos sensores, possam ser controlados e acessados remotamente, fazendo com que se tornem inteligentes, com capacidade de comunicação entre si e processamento de várias informações (MANCINI, 2017).

Com a ideia de objetos inteligentes, podemos levar a IoT para diversos ambientes, e uma das áreas de grande pesquisa diz respeito às casas inteligentes, ou *Smart Homes*. Vários sensores independentes porém todos interconectados pela rede pessoal. Controle de temperatura, luminosidade, umidade, detecção de movimento, eletrodomésticos inteligentes, todos compartilhando informações, processando dados e acessíveis ao controle dos usuários. Resultando de uma convergência de tecnologias em diversas áreas como entretenimento, segurança, gerenciamento de energia, e cuidados com a saúde, as *Smart Homes* podem ajudar a resolver vários problemas e desafios atuais, como por exemplo, o consumo eficiente e econômico de energia elétrica (NASCIMENTO, 2016), já que, os sensores além de operarem sob baixo custo de eletricidade, poderão fazer com que a conta de luz fique mais baixa, fazendo com que luzes se apaguem automaticamente e ambientes tenham sua temperatura ajustada automaticamente às necessidades dos usuários, e, juntamente com todos esses equipamentos, dispositivos também farão a medição dos gastos para verificar o que está consumindo mais, e ordenando os mesmos a usar menos energia (ROBLES; KIM, 2010).

No decorrer do desenvolvimento do presente projeto, será oferecida uma solução que possibilite ao usuário ter controle sobre as informações de estado de uma lâmpada, bem como realizar alterações entre esses estados, ou seja, ligar ou desligar uma lâmpada, através de um *Web Service*.

1.1 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em quatro capítulos principais, sendo eles, o capítulo 1, a introdução, onde além do texto de introdução do projeto, encontram-se descritos os objetivos do projeto, sendo separados em objetivo geral e objetivos específicos.

No capítulo 2 será feita a revisão da literatura e fundamentação teórica, contendo conceitos e informações importantes sobre a área de estudo em que o projeto se encontra. O capítulo 2 está estruturado contendo seções menores. Na seção 2.1 será apresentado o conceito de Internet das Coisas, na seção 2.2 será abordado sobre Smart Homes, na 2.3 introduziremos o assunto de Smart Lighting, e na 2.4 serão apresentadas as tecnologias estudadas e empregadas no desenvolvimento deste projeto. Finalizando a o capítulo 2, serão apresentados 3 trabalhos relacionados, um web server com Arduino para acender lâmpadas pela Internet, uma plataforma e ferramenta para controle de dispositivos IoT, e por fim, um kit de iluminação inteligente desenvolvido por uma empresa do ramo de móveis que já se encontra disponível no mercado.

No capítulo 3, será apresentada a metodologia envolvida no trabalho, passando por diversas seções descrevendo cada etapa que será realizada no desenvolvimento do projeto, desde a definição e estudo das tecnologias, montagem do circuito, desenvolvimento da aplicação e testes. Na seção 3.7, será descrito de maneira mais aprofundada e com esquemas o que cada componente do circuito será responsável por fazer, e também qual será a função de cada tecnologia empregada. Finalizando na seção 3.8 com o desenvolvimento preliminar, mostrando o que já foi feito até o presente momento. Por fim, no capítulo 4, serão feitas as conclusões e as considerações finais acerca do tema e do projeto.

1.2 OBJETIVOS

Nesta seção, são apresentados os objetivos do presente trabalho, sendo estes, divididos em objetivo geral, e objetivos específicos.

1.2.1 Objetivo Geral

O objetivo do trabalho é desenvolver e apresentar uma solução para *Smart Lighting* através de um circuito físico e um *Web Service* que se comuniquem entre si por meio da rede e façam com que seja possível interferir no estado de uma lâmpada, ou seja, ligar ou desligar, bem como obter informações sobre ela.

1.2.2 Objetivos Específicos

- Implementar o controle da lâmpada fazendo uso do Arduino Mega.
- Desenvolver o Web Service para comunicação com o microcontrolador em linguagem PHP.

- Realizar a comunicação do microcontrolador com o *Web Service* através do protocolo de rede CoAP.
- Certificar-se de que é possível acender, apagar e obter o status sobre a lâmpada através da rede.
- Gerenciar a lâmpada remotamente.

2 REVISÃO DA LITERATURA

Nesse capítulo, são abordados a revisão da literatura, delimitando o escopo, área de estudo e aplicação onde o projeto está alocado, bem como a fundamentação teórica, detalhando as tecnologias que serão utilizadas ao decorrer do projeto. Também será apresentado o estado da arte, uma descrição de trabalhos relacionados e soluções semelhantes ao tema. E por fim, o diferencial tecnológico, uma breve análise comparativa entre o estado da arte e este projeto.

2.1 INTERNET DAS COISAS

O termo Internet das Coisas foi utilizado pela primeira vez em 1999 por Kevin Ashton para descrever um sistema onde objetos no mundo físico poderiam se conectar com a Internet por meio de sensores. Nesse sistema, Ashton ilustrava o poder de conectar dispositivos de identificação por radio frequência (RFID) na rede para contar e rastrear objetos sem intervenção humana. Atualmente, a Internet das Coisas têm se tornado popular na descrição de cenários onde a conectividade e o poder computacional se estendem para uma larga variedade de objetos, dispositivos, sensores e itens comuns do dia-a-dia. A figura 1 exemplifica a capacidade da IoT em estar disponível a partir de qualquer dispositivo, qualquer contexto, para qualquer pessoa, qualquer serviço, em qualquer rede e em qualquer lugar (MULANI; PINGLE, 2016).

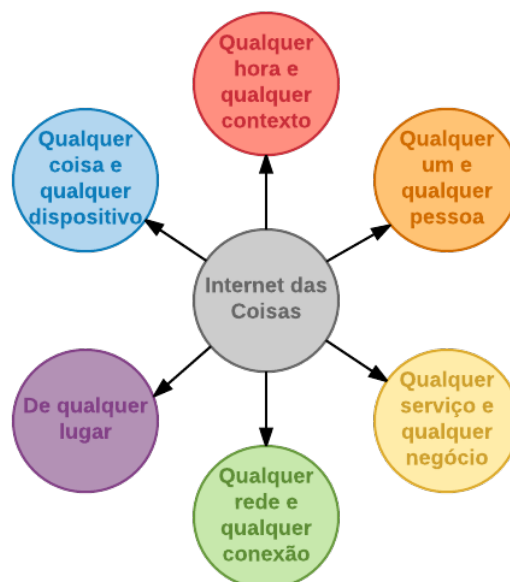


Figura 1 – Internet das coisas

Fonte: Adaptado de Mulani e Pingle (2016)

A Internet das Coisas é fundamentada no conceito de Web Ubíqua, na qual a conecti-

vidade e interatividade entre pessoas, informações, processos e objetos por meio de tecnologias que garantem o acesso à rede torna-se cada vez mais presente no cotidiano que quase não haverá percepção. Haverão diversos dispositivos que incluem sensores inteligentes, equipamentos eletrônicos, eletrodomésticos, automóveis, que, a partir de aplicações, se adaptarão automaticamente e dinamicamente às necessidades dos seus usuários (LACERDA; LIMA-MARQUES, 2015).

Existem diversas questões que são levantadas em relação a IoT, dentre elas estão incluídas questões de, segurança, privacidade, interoperabilidade, legalidade e questões econômicas. Não seria nem um pouco interessante que pessoas desconhecidas tivessem acesso aos nossos dispositivos e sensores pessoais, pois isso deixaria portas abertas para ataques ao mesmo tempo que também seria violação de privacidade, tais questões exigirão criação de novas leis ou adaptações de leis existentes. Outro problema está relacionado a interoperabilidade e aos padrões de conexões. Como os objetos precisam se comunicar, é necessário que todos façam uso dos mesmos protocolos e linguagens, garantindo assim a praticidade na hora de inserir novos dispositivos na rede (MULANI; PINGLE, 2016). Do ponto de vista econômico, segundo a Gartner, Inc., uma das principais empresas de pesquisa do mundo, haverão cerca de 26 bilhões de dispositivos IoT em 2020, excluindo computadores pessoais, smartphones e tablets, representando um aumento de aproximadamente 30 vezes o número de 0,9 bilhões de 2009. A IoT em produtos e prestação de serviços gerará uma receita adicional superior a 300 bilhões de dólares, o que resultará em 1,9 trilhões na economia global em valor agregado (GARTNER, 2013).

2.2 SMART HOME

Smart Home, que, em tradução direta significa Casa Inteligente, é um conceito que envolve várias áreas de conhecimento da ciência e engenharia. O termo *Smart Home* é comumente utilizado para definir uma residência que integra tecnologia e serviços através de uma rede com o objetivo de aprimorar a eficiência energética e melhorar a qualidade de vida (MOWAMAD; FATHY; HAFEZ, 2014).

Seguindo a ideia de IoT, onde há vários objetos conectados que trocam informações, a figura 2 exemplifica uma *Smart Home*, onde uma residência pode possuir sistemas avançados para controlar iluminação, temperatura, interruptores, equipamentos de entretenimento e multimídia, monitoramento e segurança. O ambiente deve moldar-se aos moradores da casa, podendo ser controlado por comandos de voz, acesso remoto ou computador pessoal, transformando assim, uma simples casa em uma casa inteligente e proporcionando aos seus moradores: conforto, eficiência, independência e segurança (TULY, 2016).

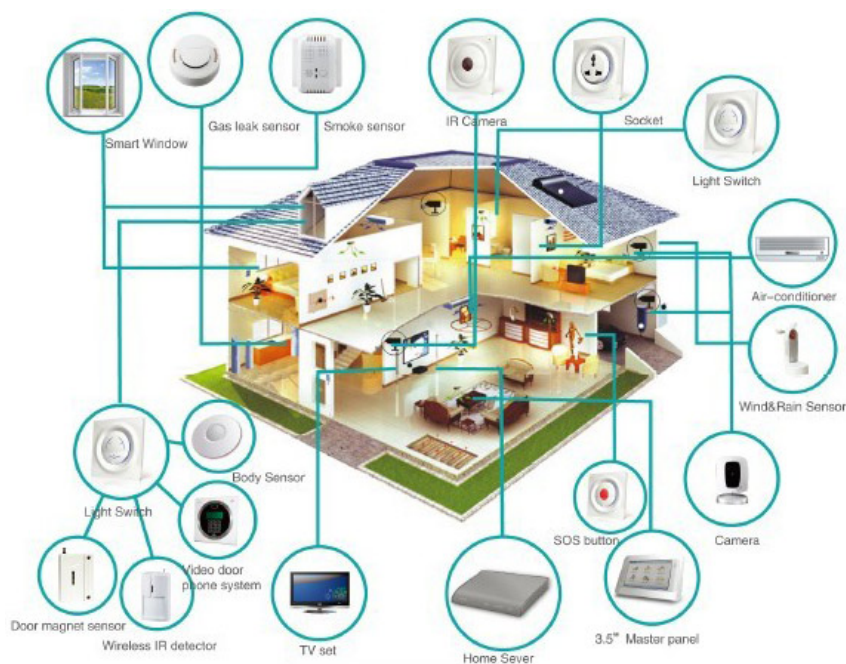


Figura 2 – Smart Home

Fonte: [Energy \(2016\)](#)

Ainda segundo [Tuly \(2016\)](#), as *Smart Homes* são divididas em quatro áreas nas quais os dispositivos precisam ser capazes de oferecer bons resultados, sendo elas:

- Gestão e Eficiência de Energia
- Saúde
- Entretenimento
- Segurança

2.2.1 GESTÃO E EFICIÊNCIA ENERGÉTICA

A maior parte do consumo mundial de energia atual é proveniente do uso residencial. E com a população crescendo rapidamente, a demanda por energia elétrica cresce junto. Com a entrada da IoT e das *Smart Homes* na sociedade, a quantidade de energia consumida por residências aumentará ainda mais. Tendo isso em mente, um dos objetivos do conceito de *Smart Home* é a busca por maneiras de se obter um sistema de gerenciamento residencial de energia sustentável e eficiente ([TULY, 2016](#)).

2.2.2 SAÚDE

Essa área de aplicação da *Smart Home* é de grande interesse por parte de pesquisadores, visto que existe um significativo aumento da população idosa. Acredita-se que o estudo da área será útil para resolução de várias questões relacionadas pessoas com idade avançada, como por exemplo, solidão, incapacidade, limitações cognitivas e a saúde em geral, sendo

que será facilitada a tarefa de monitoramento (TULY, 2016). Vários tipos de dispositivos e tecnologias relacionadas a saúde inteligente podem conter sensores que verificam o corpo, coletando informações importantes e as enviando para o médico ou para os responsáveis pelo paciente.

2.2.3 ENTRETENIMENTO

Segundo Mendes et al. (2015), o consumo de mídias digitais, sejam elas músicas, vídeos, imagens, jogos eletrônicos, tem crescido cada vez mais dentro de casas com o passar dos anos, e novas formas de entretenimento estão se tornando bastante populares, mudando permanentemente a maneira como agimos e interagimos com elas. E essa área têm mostrado enorme potencial. Formatos como *Ultra High Definition* (UHD) necessitam de conexões com latências extremamente baixas e alta capacidade de banda, além disso, a grande quantidade de dados irá requerir conexão em tempo real de vários, possivelmente distribuídos, recursos de processamento de mídia.

2.2.4 SEGURANÇA

As tecnologias de *Smart Home* contribuem muito no quesito de segurança para seus habitantes. Segundo Tuly (2016), temos dois tipos de segurança em que os dispositivos nos ajudarão. Primeiro com relação aos eventos anormais dentro da casa, onde sensores térmicos, líquidos, de movimentos e de proximidade avisam sobre incêndios, alagamentos, e possíveis acidentes que possam vir a ocorrer. Já o segundo, é um grande avanço no que diz respeito a detecção de invasões, roubos e assaltos. É importante que os equipamentos e a rede sejam de boa qualidade, pois esse tipo de solução requer tráfego de informações em tempo real e uma capacidade autosuficiente de informar se detecções e alterações no status dos sensores são reais ou apenas falsos positivos (MENDES et al., 2015).

2.3 SMART LIGHTING

Como visto anteriormente, a demanda por energia elétrica têm crescido drasticamente com a chegada de novos aparatos eletrônicos no mercado e isso afeta diretamente no preço que pagamos pelo fornecimento desse recurso. A IoT busca, como um de seus princípios, o uso eficiente da energia, e o conceito de *Smart Lighting* está ligado diretamente a isso, sendo a área de estudo da IoT voltada a soluções para iluminação inteligente de ambientes.

Segundo Martirano (2011), o uso de energia elétrica para a iluminação deve adotar maneira ativas e passivas para reduzir o consumo de energia pelo sistema sem afetar sua performance e ainda aumentando o conforto e segurança dos usuários. De acordo com o autor, existem duas estratégias para esse fim, eficiência e eficácia. A eficiência dos sistemas pode ser melhorada com o uso de equipamento de alta performance e pelo posicionamento e design da iluminação no ambiente, colocando luzes em pontos onde sua capacidade será melhor

aproveitada. Já a eficácia, pode ser aperfeiçoada adotando controles automáticos de iluminação baseados nas necessidades tanto dos usuários quanto do ambiente em si, apagando luzes quando preciso, ajustando-se de acordo com o nível de luz natural e controlando a intensidade.

2.4 FUNDAMENTAÇÃO TEÓRICA

Nessa seção, aborda-se sucintamente sobre as tecnologias e componentes que serão utilizados durante todo o desenvolvimento desse projeto.

2.4.1 ARDUINO MEGA 2560

O Arduino, criado em 2005, é uma placa microcontroladora de baixo custo, funcional, e acessível para pessoas que não possuem um grande conhecimento de programação, podendo assim, ser utilizada para estudos e até projetos mais complexos. Com a proposta de *Open Hardware* (Hardware Livre), o Arduino se destaca no mercado já que qualquer pessoa interessada na tecnologia pode montar, modificar e personalizar a placa partindo do mesmo hardware básico (THOMSEN, 2014).

O Arduino Mega 2560 (Figura 3) é uma placa da plataforma Arduino baseada no microcontrolador ATmega2560. Ela possui 54 portas de entrada/saída digitais, 16 portas analógicas, clock de 16Mhz, conexão USB e conector para alimentação de energia externa. Com relação a memória, ele possui 256kb de memória Flash, dos quais 8kb são utilizados para o bootloader, 8kb de memória SRAM e 4kb de EEPROM (ARDUINO, 2017). A principal razão da escolha do Arduino para esse projeto, dá-se por conta de sua característica *Open Hardware*, pois o mercado disponibiliza fácil acesso aos módulos e sensores.

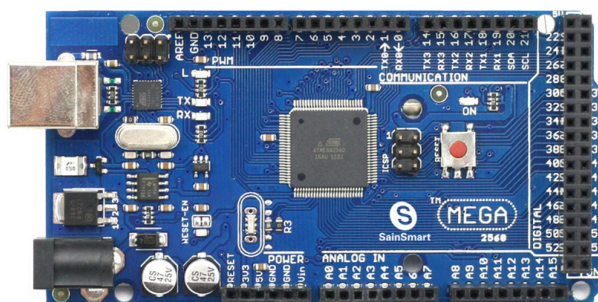


Figura 3 – Arduino MEGA 2560

2.4.2 SHIELD ETHERNET W5100

Por ser *open hardware*, é possível estender as funcionalidades de uma placa fazendo uso de Shields, placas de *hardware* que são encaixadas no Arduino e podem exercer diversas funções como por exemplo, conexão com a Internet, motores, Wi-Fi, e muitos outros, podendo até mesmo empilhar Shields uns sobre outros, já que a maioria disponível no mercado possui seus próprios conectores (REIS, 2015).

O Shield Ethernet W5100 (Figura 4) permite que o Arduino seja conectado a uma rede local ou à Internet, fazendo com que seja possível ao microcontrolador responder a comandos remotos, monitorar a placa ou sensores conectados a ela. O Shield será utilizado no projeto para configurar o endereço a qual a aplicação estará executando.



Figura 4 – Shield Ethernet W5100

2.4.3 MÓDULO RELÉ

Módulos são dispositivos que são conectados nas portas analógicas e/ou digitais do Arduino e permitem ao desenvolvedor estender a capacidade da placa. Porém, diferentemente dos Shields, que mesmo conectados na placa, ainda disponibilizam a maioria das portas para utilização no circuito, os módulos precisam fazer uso das portas, tornando as exclusivas para eles.

Existem vários módulos disponíveis no mercado, como por exemplo, módulo Bluetooth, Ethernet, relé, entre outros. Para este projeto, será utilizado o módulo relé. Um relé é uma chave mecânica que é acionada por um meio eletrônico, sendo assim, um dispositivo eletromecânico. Ele funciona como um interruptor, sendo utilizado para ligar ou desligar dispositivos, permitindo assim o funcionamento de outros aparelhos conectados ao circuito por conta de uma alteração no equipamento causada pela passagem de corrente elétrica (CARVALHO, 2015).

O módulo relé (Figura 5) é um módulo que pode ser utilizado no arduino, possui três pinos para conexão com a placa, e três para saída (LIMA, 2015).



Figura 5 – Módulo relé

2.4.4 RIOT OS

Um sistema operacional é responsável por gerenciar os recursos disponíveis do hardware e fazer o intermédio entre o *hardware* e o *software*. Com o advento da IoT, várias questões relacionadas ao tema e a nível de software e hardware foram levantadas. A quantidade de hardwares com especificações diferentes no mercado é bastante grande, e na maioria das vezes, não são compatíveis uns com os outros, requerem linguagens de programação específicas, trabalham de maneiras diferentes com energia elétrica, possuem limitação de memória. Tendo isso em vista, o cenário ideal seria onde todos os dispositivos conseguissem se comunicar através dos mesmos protocolos e padrões, sendo assim, a ideia de um sistema operacional exclusivo para IoT é bastante atraente.

O Riot OS é um sistema operacional *open source* comunitário para IoT que vem sendo desenvolvido desde 2008 e é capaz de rodar em uma grande variedade de *hardwares*. Com a premissa de ser amigável ao desenvolvedor, executar códigos nas linguagens C ou C++, atingir o máximo de eficiência de energia, ter capacidades de processamento em tempo real e ter suporte a diversos protocolos de rede como por exemplo 6LoWPAN, IPv6, RPL, UDP, CoaP e CBOR, e ainda exigir pouquíssima memória, como podemos ver na tabela abaixo, onde o sistema é comparado com outros bastante conhecidos como por exemplo, o Contiki, Tiny OS e o próprio Linux, o Riot OS é uma ótima opção para a IoT, conseguindo oferecer várias funcionalidades de um sistema operacional completo, em um sistema leve, prático, funcional e com muitos recursos (BACCELLI et al., 2013).

OS	Min RAM	Min ROM	Suporte C	Suporte C++	Multi-Threading	Real-Time
Contiki	<2kB	<30kB	○	∅	○	○
Tiny OS	<1kB	<4kB	∅	∅	○	∅
Linux	~1MB	~1MB	✓	✓	✓	○
RIOT OS	~1.5kB	~5kB	✓	✓	✓	✓

Tabela 1 – Comparativo entre alguns dos sistemas operacionais mais conhecidos para IoT. (○) representa suporte parcial, (∅) representa nenhum suporte e (✓) representa suporte total.

Fonte: [Baccelli et al. \(2013\)](#)

Ao analisarmos a tabela, chegamos a conclusão que o Riot OS é, entre os sistemas comparados, o que melhor oferece condições de cumprir com os objetivos deste projeto. Tendo em vista esse fator, a capacidade de trabalhar com as linguagens C e C++, e também a capacidade de oferecer suporte nativo ao Arduino, microcontrolador escolhido anteriormente, o Riot OS se torna o sistema ideal para o projeto.

2.4.5 COAP

Quando se fala sobre comunicação entre dois ou mais dispositivos que estão conectados a uma rede, surge a necessidade de um conjunto de regras sobre como essa comunicação será realizada, e esse conjunto é chamado de protocolo. Tais regras irão ditar a maneira com que será feita a troca de mensagens e/ou dados que compõe a rede, levando em consideração as limitações impostas pelo ambiente.

O *Constrained Application Protocol* (CoA P) foi criado em 2010 por um grupo chamado Constrained RESTful Environments (CoRE), com o objetivo de ser um framework para aplicações que tratam recursos simples em dispositivos conectados em redes limitadas, como por exemplo, aplicações que monitoram sensores, medidores, controlam atuadores e gerenciam dispositivos que compõe a rede. De acordo com a documentação do protocolo, o CoaP é voltado para nós com pequena quantidade de memória RAM e redes restritas onde a taxa de perda de pacotes é alta. Foi também projetado para aplicações M2M, como por exemplo, projetos de *Smart Energy* e *Smart Home*, interagindo diretamente com a Web da mesma maneira que o HTTP, podendo ser considerado um protocolo RESTful ([MARTINS; ZEM, 2014](#)).

Do ponto de vista técnico, o CoAP foi desenvolvido para trabalhar na camada de aplicação, cuja qual, normalmente é utilizado o protocolo HTTP para prover serviços web. No entanto, o HTTP tem uma complexidade computacional muito alta, alto consumo de energia e baixa taxa de transferência de dados, fazendo com que sua aplicação seja inviável em dispositivos com recursos limitados. Sendo assim, a intenção do CoAP, é alcançar a mesma eficiência que o HTTP tem, porém, voltado exclusivamente para a IoT ([CHEN, 2014](#)). Na figura 6, podemos ver as principais características do protocolo CoAP, que são:

- Protocolo web que cumpre com o requisitos M2M para hardwares limitados.

- Segurança atrelada ao DTLS (Datagram Transport Layer Security).
- Troca de mensagens assíncronas.
- Baixa complexidade de cabeçalho.
- Suporte a URI e Content-type.
- Capacidade de trabalhar de maneira simples com proxy e cache.
- Atrelado ao UDP e com suporte a *unicast* e *multicast*.
- Mapeamento HTTP sem estados, permitindo a construção de proxies.

Protocolo Web que cumpre com os requisitos M2M para hardwares limitados
Segurança atrelada ao DTLS (Datagram Transport Layer Security)
Troca de mensagens assíncronas
Baixa complexidade de cabeçalho
Suporte a URI e Content-Type
Capacidade de trabalhar de maneira simples com proxy e cache
Atrelado ao UDP e com suporte a unicast e multicast.
Mapeamento HTTP sem estados, permitindo a construção de proxies que provêm acesso à recursos CoAP através de HTTP de uma maneira uniforme, ou para que simples interfaces HTTP sejam reconhecidas através de CoAP

Figura 6 – Características do protocolo CoAP

Fonte: Adaptado de [Chen \(2014\)](#)

Como neste projeto utilizaremos um dispositivo com recursos limitados, e o foco do trabalho está no estudo e utilização das novas tecnologias voltadas para IoT, utilizaremos a implementação do CoAP para fazer a comunicação e envio de mensagens do web da aplicação para o servidor.

2.4.6 HTML

Hypertext Markup Language, ou simplesmente HTML, é uma linguagem de marcação utilizada para desenvolvimento de *websites* e aplicações web. Criada em 1991 por Tim Berners-lee, foi a primeira linguagem de marcação a ser utilizada em escala global ([PACIEVITCH, 2016](#)). E atualmente, a maioria das aplicações web fazem uso do HTML, mesclado com outras linguagens de programação como por exemplo PHP, Ruby, Python, Java, entre outras, viabilizando assim, a construção de aplicações mais completas, dinâmicas e complexas.

A escrita dos códigos HTML é baseada em marcações, mais conhecidas como *tags*. Existem uma grande variedade de *tags* que podem ser utilizadas para várias finalidades, como

por exemplo, as *tags* `<body>` e `</body>`, que indicam respectivamente a abertura e o fechamento do corpo, ou conteúdo de uma página web. E os arquivos que contêm código fonte na linguagem HTML podem ser criados e editados com simples editores de texto, sem a necessidade de uma ferramenta específica.

A cada nova versão lançada da linguagem, ela se torna mais intuitiva e fácil de ser interpretada e utilizada. Para esse projeto, será utilizada a versão 5, que além de ser a versão mais atual, conta com todos os componentes que serão necessários para o desenvolvimento do projeto.

2.4.7 WEB SERVICE REST

Web Services são, de acordo com a W3C [Booth et al. \(2004\)](#), um sistema de software responsável por realizar a comunicação M2M através da rede. E para que um *Web Service* seja construído de acordo com os padrões e funcione perfeitamente, ele precisa seguir alguns princípios.

REST (*Representational State Transfer*) é caracterizado como um paradigma arquitetural, ou seja, uma série de princípios arquiteturais para que se possa construir *Web Services*. Nota-se que, uma API RESTful, é uma API que segue a arquitetura REST. Várias linguagens de programação aplicam os padrões REST, e uma delas é o PHP, pois trabalha nativamente com as requisições HTTP (GET, POST, DELETE, PUT), as quais os *Web Services* implementados fazendo a utilização do REST também utilizam ([PASQUA, 2015](#)).

2.4.8 PHP

Inicialmente chamado de *Personal Home Page* e criado em 1994 por Rasmus Lerdorf, o PHP, hoje chamado de *Hypertext Preprocessor*, é uma linguagem de script *open source* (código livre) utilizada para o desenvolvimento de aplicações web, que, devido a capacidade de se mesclar ao HTML, faz com que seja capaz de gerar conteúdo dinamicamente, tornando páginas e *websites* mais interativos e ricos em conteúdo ([ALVES; GONÇALVES, 2012](#)).

O PHP é uma linguagem *server-side*, ou seja, quando o cliente acessa uma página que contém códigos na linguagem, uma requisição é enviada ao servidor pedindo o acesso a tal página, o servidor interpreta o código PHP e retorna para o cliente a página solicitada, sendo que para o cliente, será exibida apenas a marcação HTML. Esse ciclo pode ser visto na figura [7](#).

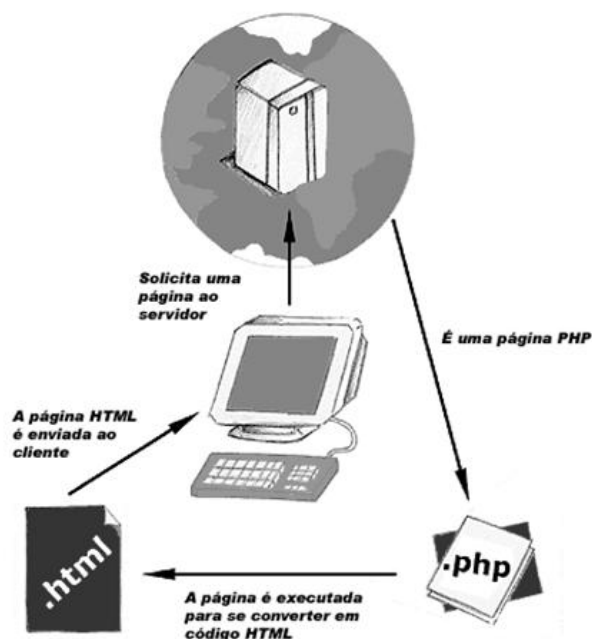


Figura 7 – Esquema de funcionamento de uma página web em PHP

Fonte: [Alves e Gonçalves \(2012\)](#)

Segundo [Mello \(2016\)](#), a linguagem pode ser utilizada para vários tipos de aplicações, dentre elas os comércios eletrônicos, fóruns, sistemas de gerenciamento de conteúdo, aplicativos para o Facebook, entre outros. Suas principais vantagens são a facilidade de aprendizado, a praticidade para acesso aos SGBDs (Sistema Gerenciador de Banco de Dados), e o fato de ser uma linguagem multiplataforma, permitindo que as aplicações escritas utilizando PHP sejam acessíveis a partir qualquer computador que tenha um navegador de Internet, independentemente do sistema operacional. Vale lembrar também que, ainda segundo o autor, existem algumas desvantagens do PHP, como por exemplo a compatibilidade entre as versões não ser totalmente confiável, e o fator segurança, que se mostra inferior se comparada a um Servlet do Java EE.

Apesar das desvantagens, o presente projeto utilizará o PHP na versão 7.0 ou superior para o desenvolvimento do *Web Service*. Os motivos da escolha se dão pela ampla comunidade de desenvolvedores e projetos abertos disponíveis na Internet, o suporte multiplataforma e a facilidade de interação com a linguagem de marcação HTML.

2.4.9 CSS

Abreviação para *Cascading Style Sheets*, o CSS é uma linguagem bastante utilizada em *websites*, páginas web e aplicações em geral, para descrever estilos em documentos HTML ou XML (*Extensible Markup Language*), definindo a forma como cada elemento será apresentado em termos de cores, bordas, fontes, posicionamento, entre outros ([MELLO, 2016](#)).

O funcionamento do CSS dá-se por meio do conceito de seletores, ou seja, são maneiras que a linguagem tem para encontrar um determinado conteúdo que está definido na arquitetura

da página, escrita em HTML, podendo assim aplicar estilos e efeitos diferentes para cada elemento da aplicação. O CSS pode ser aplicado de três maneiras diferentes:

- No próprio elemento HTML.
- Dentro de uma *tag* específica para estilo no arquivo HTML.
- Em um arquivo externo com a extensão .css.

O conceito de cascata presente no nome é devido a maneira em que é estabelecida a prioridade de uma regra sobre um elemento, sendo que definir o estilo no próprio elemento terá a maior prioridade, e definir em um arquivo externo, a menor (MELLO, 2016). O CSS fará parte desse projeto com o objetivo de apresentar ao usuário final, uma interface intuitiva e com boa aparência.

2.5 TRABALHOS RELACIONADOS

Nessa seção, serão apresentados projetos e trabalhos que se relacionam ao tema, possuem algumas características, ou se assemelham ao projeto proposto, bem como uma pequena discussão analisando e comparando os trabalhos com o projeto proposto.

2.5.1 FilipeFlop - Acendendo Lâmpadas pela Internet

Nesse trabalho, o autor faz o uso do Arduino e alguns componentes, como o Shield Ethernet, o módulo relé e lâmpadas para criar um sistema que controla luzes pela Internet. Foi desenvolvido um *Web Server* utilizando as bibliotecas nativas do Arduino, e o mesmo consegue obter informações e alterar o status das lâmpadas através do acionamento dos relés. Tal acionamento é feito por um script na linguagem JavaScript, que, ao detectar o evento de clique no botão definido no HTML, envia uma informação com o estado da lâmpada para uma *div* (elemento HTML) que está oculta da visão do usuário. Essa informação é então capturada pelo Arduino, verificada e por fim enviada ao módulo relé, permitindo assim, a passagem ou não da corrente elétrica para acender ou desligar a lâmpada, como mostrado na figura 8 (THOMSEN, 2015).



Figura 8 – Web Server acendendo lâmpadas

Fonte: [Thomsen \(2015\)](#)

2.5.2 Lelylan - IOT Cloud Platform

O projeto Lelylan é uma plataforma em nuvem de microserviços arquiteturais para a IoT voltada ao público desenvolvedor. Em outras palavras, a ferramenta fornece um painel de controle e uma API para monitorar e controlar dispositivos físicos, com o objetivo de dividir a complexidade da Internet das Coisas em vários pequenos e independentes microserviços, podendo ser instalada em qualquer servidor de nuvem público e com suporte, no momento para as linguagens Curl, Ruby, Node.js e AngularJS. A API também dá suporte para vários *hardwares* encontrados no mercado, como por exemplo o Arduino Yun, Raspberry PI, Eletric Imp, Spark Core, Netduino, Texas Instruments CC3000/CC3200, ou quaisquer outros que ofereçam suporte ao protocolo MQTT ([LELYLAN, 2017](#)).

Por meio do painel de gerenciamento, o usuário pode facilmente ter acesso a opções de adicionar, editar, remover e monitorar quaisquer dispositivos, como por exemplo, sensores de umidade, movimento, alarmes, câmeras, fechaduras e inclusive lâmpadas inteligentes que sejam suportados pela API, tornando assim menos trabalhosa a conexão de sensores e atuadores na rede. Na figura 9 é mostrado o cadastro de um dispositivo, após esse cadastro, será fornecido um código identificador e uma senha para realizar a conexão com a rede por meio do protocolo MQTT ([LELYLAN, 2017](#)).

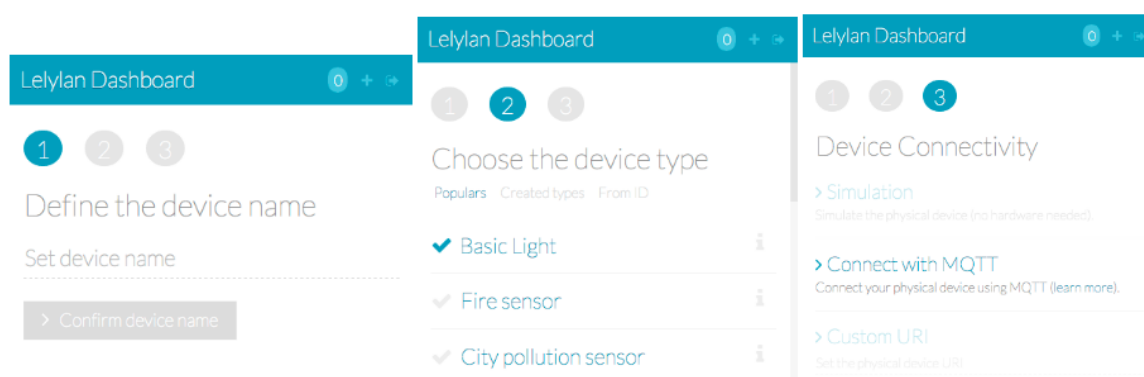


Figura 9 – Inserindo um dispositivo no Lelylan

Fonte: [Lelylan \(2017\)](#)

Após criado o dispositivo no sistema, o usuário apenas conecta o dispositivo físico por meio dos dados fornecidos, e assim, ele ficará disponível para controle a partir do painel de gerenciamento.

2.5.3 IKEA Trådfri - Smart Lamps

No mês de março de 2017, a empresa sueca Ikea, maior empresa do mundo no setor de fabricação e venda de móveis, e mundialmente conhecida por seus preços baixos (segundo [Jappe \(2014\)](#)), lançou uma nova linha de produtos de automação residencial voltada para iluminação inteligente, chamada Trådfri. A linha consiste de lâmpadas, sensores de presença, ponto de conexão (*gateway*) para várias lâmpadas e ajuste de iluminação por controle remoto, podendo ser encontrada por valores variando entre \$11.99 para uma única lâmpada até \$79.99 para o kit (figura 10) com duas lâmpadas, controle remoto e *gateway* ([IKEA, 2017](#)). ([LELYLAN, 2017](#)).

Do ponto de vista técnico, o produto é bastante atraente dentro dos conceitos da IoT, pois funciona sem necessitar de muita configuração. O kit com lâmpada e controle remoto já vem pré-configurado, e a aquisição do *gateway* é somente necessária se o usuário quiser utilizar o controle pela rede através de um computador ou do aplicativo para *smartphone* e definir configurações inteligentes. A comunicação com o *gateway* é feita através do protocolo CoAP, e entre os dispositivos é através do ZigBee, um protocolo de comunicação com baixo consumo energético projetado para redes locais e de curta distância baseado no padrão IEEE802.15.4 ([DENG et al., 2016](#)), isso significa que não é necessário que as lâmpadas na rede sejam somente da mesma marca, ou seja, o usuário pode comprar e conectar facilmente lâmpadas de outras marcas, como por exemplo a Philips Hue e outros ([SCHOUTSEN, 2017](#)).

Uma das grandes vantagens do produto é também a compatibilidade com os sistemas de gerenciamento inteligentes como por exemplo o *Apple Homekit*, *Google Home*, e também a assistente virtual *Amazon Alexa*. E ainda segundo a seção de FAQ (perguntas mais comuns) do



Figura 10 – Kit Smart Lighting Ikea

Fonte: [IKEA \(2017\)](#)

site da empresa, eles estão trabalhando em uma API livre para desenvolvedores que queiram projetar seus próprios ambientes de iluminação inteligente baseado na linha Trådfri. Mas apesar de todas as vantagens do produto, uma grande desvantagem, devido ao fato da limitação da rede provida pelo ZigBee, é que o sistema funcionará apenas na rede local de onde for instalado, inviabilizando o controle remoto das lâmpadas ([SCHOUTSEN, 2017](#)).

2.5.4 DIFERENCIAL TECNOLÓGICO

O primeiro trabalho utiliza as bibliotecas nativas do Arduino a fim de cumprir o objetivo, que é acender as lâmpadas via rede utilizando IP. A construção do projeto é simples e servirá como base para a montagem do circuito físico do presente projeto. Uma rápida busca na internet e nos deparamos com vários projetos semelhantes utilizando o Ethernet Shield. Já o segundo trabalho, consegue alcançar o mesmo objetivo mas de uma maneira mais complexa e fazendo o uso de um protocolo de rede para IoT, o MQTT, oferecendo um painel de controle onde é possível interagir com os dispositivos. Um diferencial do segundo projeto é que a API fornecida oferece suporte a diversos *hardwares* e linguagens de programação de alto nível. E por fim, o terceiro projeto é um produto que já está no mercado desde o primeiro semestre do ano de 2017, sendo em suma, bastante parecido com este projeto, fazendo inclusive uso do mesmo protocolo de rede para a comunicação das lâmpadas e *gateway*, o CoAP, porém, com o principal diferencial de utilizar ZigBee para funcionamento apenas na rede local.

Apesar de o objetivo final de todos os projetos ser o mesmo, acender uma lâmpada via rede, o principal diferencial dos três trabalhos apresentados, para com o trabalho deste projeto, está na utilização do sistema operacional RIOT OS, sistema voltado para IoT que executa na plataforma Arduino e em diversas outras, alterando a maneira com que trabalhamos com a placa.

3 METODOLOGIA

Neste capítulo, serão apresentadas e descritas as etapas que serão realizadas a fim de cumprir com os objetivos inicialmente propostos. Sendo assim, o presente capítulo está organizado da seguinte maneira:

- Definição das tecnologias;
- Estudo de cada uma das tecnologias selecionadas;
- Elaboração e montagem do circuito físico;
- Desenvolvimento do servidor e aplicação;
- Desenvolvimento do Web Service;
- Realização de testes gerais;

Após isso, na seção 3.7 serão apresentados os protótipos, e juntamente com eles, descrições e esquemas de funcionamento do projeto, explicando e detalhando como as tecnologias interagirão entre si.

3.1 DEFINIÇÃO DAS TECNOLOGIAS

Em um primeiro momento, serão definidas as tecnologias a serem utilizadas. É importante ressaltar que o projeto será desenvolvido levando em consideração as limitações impostas pelas tecnologias definidas. Levando isso em consideração, será importante que a escolha seja feita com base nas quais atendam as necessidades do projeto. Para a escolha, critérios como a comunidade relacionada, facilidade de acesso, eficácia e possível importância dentro do contexto definido serão avaliados.

3.2 ESTUDO DAS TECNOLOGIAS

Esta etapa é constituída por diversas pesquisas na Internet, em livros, documentações, manuais, entre outros, referentes a todas as tecnologias que serão utilizadas no projeto, até que seja formada uma base de conhecimento necessária para dar início ao desenvolvimento propriamente dito. Porém, vale lembrar que o estudo da literatura disponível se estende durante todo o desenvolvimento.

3.3 ELABORAÇÃO E MONTAGEM DO CIRCUITO FÍSICO

Após a definição e estudos iniciais das tecnologias, o próximo passo será montar o circuito físico que será utilizado no projeto para dar prosseguimento à fase de codificação. O circuito será composto por uma placa microcontroladora Arduino Mega 2560, um shield

Ethernet, um módulo relé, cabos de conexão e uma lâmpada. Serão tomadas notas sobre as conexões e especificadas quais portas do hardware serão utilizadas.

3.4 DESENVOLVIMENTO DO SERVIDOR E APLICAÇÃO

Após a montagem do circuito e todas as conexões físicas necessárias, utilizando uma IDE ou um editor de texto voltado para a programação, será desenvolvido o código necessário para o funcionamento do servidor e protocolo de conexão. Nesta etapa também, será construído o algoritmo responsável por trabalhar com as requisições, dados e solicitações e que o Web Service realizará futuramente.

3.5 DESENVOLVIMENTO DO WEB SERVICE

Nesta etapa, também utilizando uma IDE ou editor de texto, será construída a aplicação web utilizando as linguagens PHP, HTML e CSS. Tal aplicação será responsável por enviar solicitações e requisições para o servidor anteriormente construído e após isso, apresentará as informações através de uma interface para o usuário de forma organizada e interativa.

3.6 REALIZAÇÃO DE TESTES

Juntamente com o desenvolvimento e codificação do projeto, serão feitos testes nas funcionalidades com o objetivo de identificar quaisquer erros que possam prejudicar o sistema, garantindo o funcionamento de acordo com os objetivos anteriormente propostos, verificando a estabilidade do serviço e consistência dos dados enviados e recebidos através do servidor, efetuando correções sempre que necessário.

3.7 DESENVOLVIMENTO

Nessa seção, serão descritas mais aprofundadamente as etapas do desenvolvimento, o que cada parte do hardware será responsável por fazer, as características de cada etapa e como funcionará a conexão entre as diversas partes do sistema como um todo.

Tendo as tecnologias e objetivos bem definidos, dá-se início à prototipagem do sistema em geral, procurando entender todos os pontos de comunicação e quais serão os principais desafios. Podemos, de maneira simplória, separar o sistema em duas partes maiores, sendo elas o hardware, e o software, as quais, para melhor entendimento, a partir desse ponto, chamaremos respectivamente de parte física, e parte lógica. Ambas serão organizadas como subções do desenvolvimento. E abrindo o escopo, separaremos também o desenvolvimento da parte física e lógica em sub-subseções mais específicas dentro de cada uma.

3.7.1 HARDWARE - PARTE FÍSICA

A parte física do sistema será construída a partir dos componentes selecionados anteriormente, que são os seguintes:

- Arduino Mega 2560
- Shield Ethernet
- Módulo Relé
- Lâmpada comum

Para termos um melhor entendimento de como os componentes se comportarão e como serão conectados entre si, antes de prosseguirmos com a montagem do circuito de maneira física, foi desenvolvido através da ferramenta Fritzing¹, um protótipo do circuito, representado a seguir na figura 11, que nos ajudará a entender melhor a parte física do projeto.

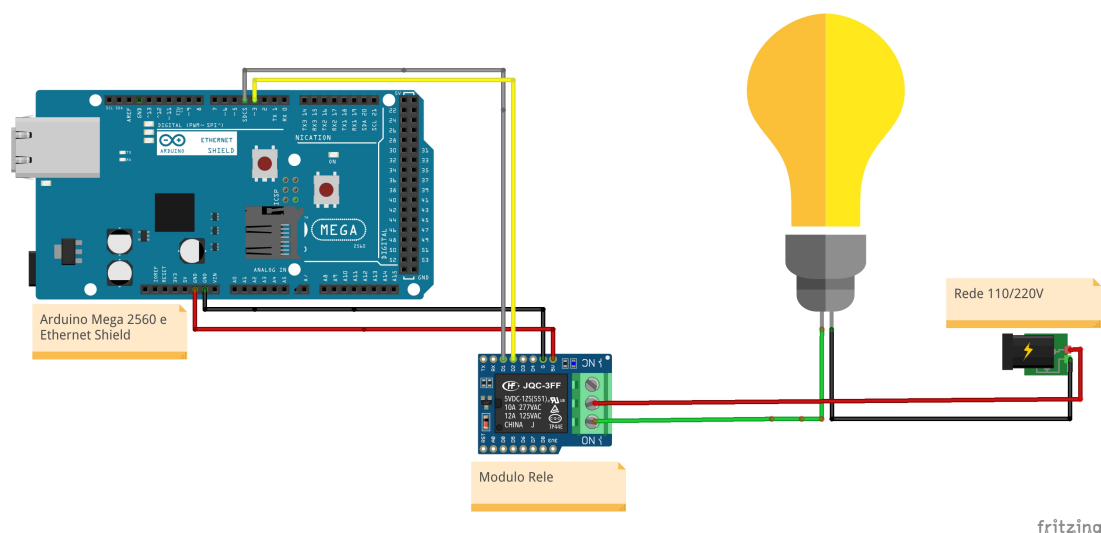


Figura 11 – Circuito de componentes

Fonte: Autoria própria

Analisando a figura, dividiremos as funções de cada um dos componentes.

3.7.1.1 ARDUINO

A placa microcontroladora Arduino será responsável gerenciar toda a aplicação, integrando todos os componentes, enviando informações para o servidor e disparando sinais para o módulo relé com o objetivo de acender ou apagar a lâmpada. Em outras palavras, a placa é o cérebro da aplicação, já que toda a lógica e codificação será executada por ela.

¹Ferramenta grátis e de código aberto com o intuito de tornar a área da eletrônica acessível para qualquer pessoa que tenha interesse, oferecendo um software e um fórum para a comunidade ativa, permitindo aos usuários documentar e compartilhar seus projetos, bem como criar e produzir circuitos e placas profissionalmente (FRITZING, 2017).

3.7.1.2 ETHERNET SHIELD

Como o Arduino Mega 2560 não possui nativamente nenhum tipo componente que permita a comunicação com a Internet, utilizaremos o shield para estender a funcionalidade da placa e permitir com que ele se conecte à rede. No que diz respeito a conexão das portas, o shield foi construído com o intuito de ser facilmente integrado à placa, sendo que é apenas necessário posicioná-lo de forma com que os pinos centrais se encaixem corretamente e a porta de conexão do RJ45 fique exatamente em cima da porta USB do Arduino, como podemos ver na figura 12.

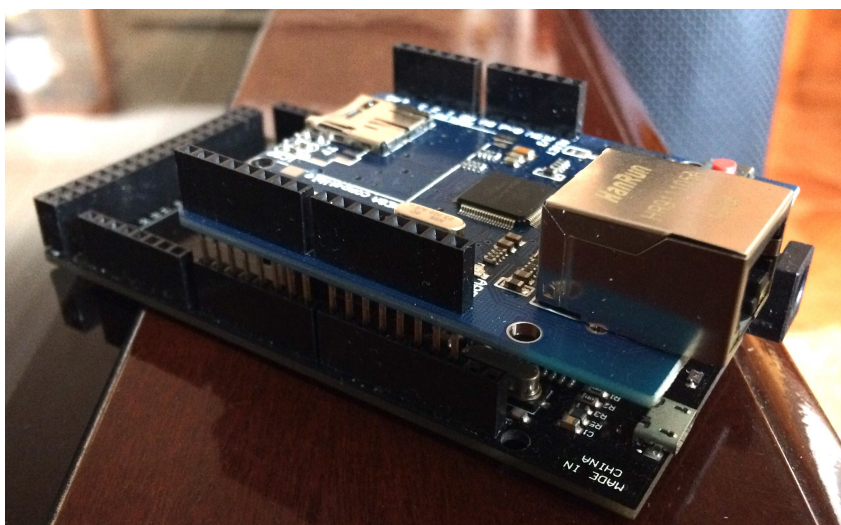


Figura 12 – Montagem do Shield no Arduino

Fonte: Autoria própria

Pode-se dizer que o shield é um dos componentes mais importantes do projeto, pois sem ele não conseguiríamos fazer com que o servidor, futuramente codificado funcione da maneira esperada, e sem o servidor, a aplicação web não ficaria disponível para acesso.

3.7.1.3 RELÉ

O módulo relé será conectado como no esquema mostrado na figura 11, um cabo sairá da porta 5v do Arduino para a porta 5v do relé, permitindo que ele tenha energia suficiente para ser acionado. Outro cabo sairá da porta GND (ground) do Arduino e irá até a porta GND do relé, o objetivo dessa conexão é evitar que se tenha interferências na corrente elétrica do projeto. E a última porta, sendo a porta lógica, saíra de alguma das portas disponíveis para a única que sobrou do relé.

Na saída do relé, cabos isolados são conectados na lâmpada e na rede elétrica. O relé atuará como o interruptor no circuito, pois é ele quem receberá o sinal para abrir ou fechar, deixando com que a corrente elétrica passe ou não para acender a lâmpada.

3.7.1.4 LÂMPADA

Por último e no fim do circuito, temos uma pequena lâmpada comum, conectada à saída do relé por meio de cabos isolados para evitar choques, e também à rede elétrica, para poder ter energia suficiente para acender completamente. A lâmpada irá ficar acesa ou apagada conforme a ação do relé.

3.7.2 SOFTWARE - PARTE LÓGICA

A parte lógica do sistema é onde se encontram os códigos e instruções que realizam a comunicação com o hardware e fazem o sistema funcionar, sendo responsável por trabalhar com todas as informações enviadas e recebidas. Tal parte será construída utilizando as tecnologias definidas anteriormente, que são as seguintes:

- RIOT OS
- Protocolo CoAP
- Linguagens C, PHP, HTML e CSS

Novamente para entendermos melhor onde as tecnologias serão aplicadas, como se comportarão e interagirão entre si, foi construído utilizando a ferramenta GIMP², um esquema simplificado, representado a seguir na figura 13, que mostra o funcionamento da parte lógica do sistema.

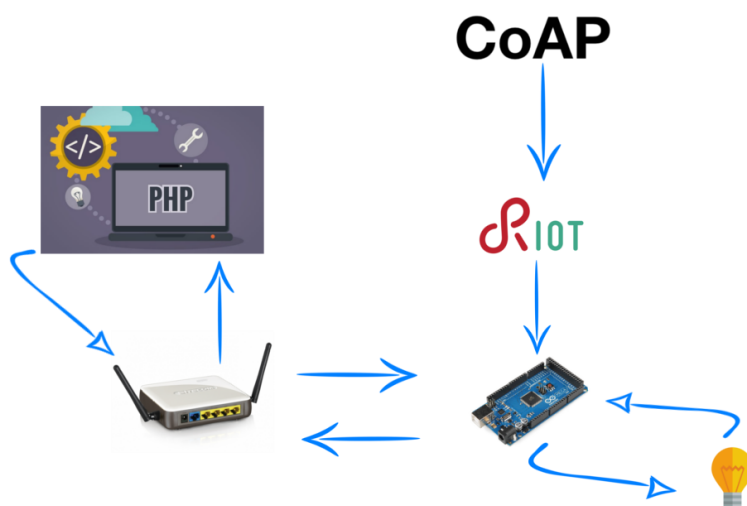


Figura 13 – Protótipo de funcionamento

Fonte: Autoria própria

²Software gratuito e open source multiplataforma para edição e manipulação de imagens, oferecendo ferramentas sofisticadas para tratamento de fotografias, ilustrações e design.

Analisando a figura, faremos a separação das tecnologias e explicaremos suas funções dentro do sistema.

3.7.2.1 RIOT OS

O sistema operacional Riot OS é a base da aplicação proposta. Ele irá substituir o firmware do Arduino, tornando possível a utilização das bibliotecas disponibilizadas pelo sistema para que seja alcançado o objetivo proposto. O sistema será responsável por executar a aplicação que será construída em linguagem C, gerenciando a memória, os processos e fazendo todo o circuito funcionar. O sistema também oferece um *shell*, ou seja, um terminal de comando, fazendo com que seja possível monitorar a aplicação e executar comandos para interagir com o sistema.

3.7.2.2 CoAP

O sistema fará a utilização da implementação do CoAP fornecida pelo Riot OS. Tal implementação interagirá diretamente com o shield ethernet utilizado no circuito físico, fazendo-o se conectar a rede e fornecendo o endereço cujo qual o usuário irá acessar o Web Service. O servidor será responsável por receber e enviar os dados através de requisições GET e POST, sendo a requisição GET para recuperar os dados sobre a lâmpada com o fim de exibir se ela está acesa ou não, e a requisição POST para enviar dados com o objetivo de acender ou desligar a lâmpada.

3.7.2.3 APLICAÇÃO RIOT OS E ARDUINO

A aplicação que será executada pelo sistema operacional no Arduino será construída utilizando a linguagem C, linguagem de programação estruturada e de uso geral, que pode ser escrita a partir de qualquer editor de texto, porém, é preferível que o editor seja voltado à programação, garantindo assim que o código seja bem escrito. O código da aplicação primeiramente deverá mapear os componentes do circuito físico à suas respectivas portas para em seguida prosseguir com as instruções que farão cada componente funcionar do modo esperado.

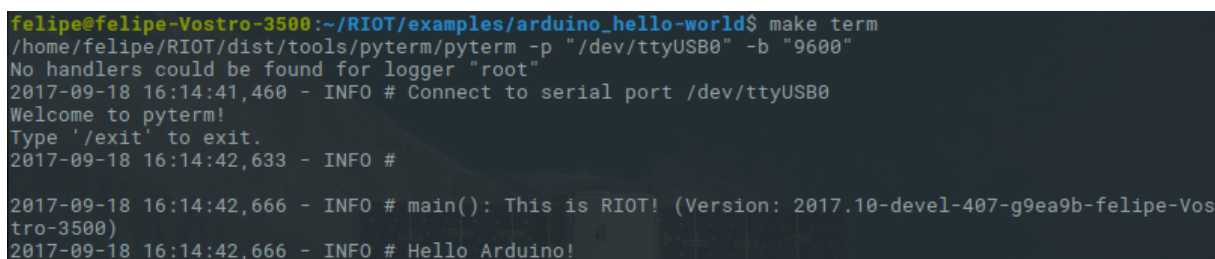
3.7.2.4 WEB SERVICE

O Web Service será constituído de uma página web que fornecerá a interface para o usuário interagir com a aplicação e visualizar as informações na tela. Será construído utilizando as linguagens HTML, CSS e PHP, sendo que, cada uma é responsável por uma parte da aplicação. O HTML será utilizado para construir a estrutura da página, o CSS será utilizado a partir do Materialize CSS, um framework CSS para construção de interfaces de usuário e

estilização de páginas web baseado no Material Design³ da Google. E por último, o PHP, que será utilizado para construir os scripts responsáveis por capturar a informação e enviar para o servidor e também recuperar os dados do servidor e apresentá-los de maneira correta ao usuário.

3.8 DESENVOLVIMENTO PRELIMINAR

Dando início ao processo de desenvolvimento, após todas as tecnologias terem sido definidas e o que cada componente será responsável por fazer, foram iniciados os testes com a execução do sistema operacional Riot OS no Arduino por meio de uma aplicação básica de "Hello World". Foi feito o *upload* do sistema e aplicação para a placa e foi verificado que o sistema funciona perfeitamente com o modelo do Arduino selecionado. Podemos ver na figura 14, o terminal do Riot informando o sucesso da execução. Porém, para antigirmos essa saída, houveram algumas dificuldades no processo com relação a algumas bibliotecas faltantes para o compilador, e também pelo fato de não haverem muitos exemplos na Internet, já que o Riot OS é uma tecnologia consideravelmente nova.



```
felipe@felipe-Vostro-3500:~/RIOT/examples/arduino_hello-world$ make term
/home/felipe/RIOT/dist/tools/pyterm/pyterm -p "/dev/ttyUSB0" -b "9600"
No handlers could be found for logger "root"
2017-09-18 16:14:41,460 - INFO # Connect to serial port /dev/ttyUSB0
Welcome to pyterm!
Type '/exit' to exit.
2017-09-18 16:14:42,633 - INFO #
2017-09-18 16:14:42,666 - INFO # main(): This is RIOT! (Version: 2017.10-devel-407-g9ea9b-felipe-Vostro-3500)
2017-09-18 16:14:42,666 - INFO # Hello Arduino!
```

Figura 14 – Execução inicial do Riot OS no Arduino

Fonte: Autoria própria

Prosseguindo com o desenvolvimento do projeto, assim que todos os componentes físicos foram adquiridos, foi montado, com base no esquema da figura 11, o circuito físico do sistema, que podemos ver na figura 15, atendo-se aos detalhes e portas descritos no esquema anterior.

³Conceito de design criado em 2014 pela Google, pensado para ser fluído, natural, intuitivo e de simples compreensão, sintetizando os conceitos clássicos de um bom design com a inovação e possibilidades trazidas com a tecnologia e a ciência (RALLO, 2017).

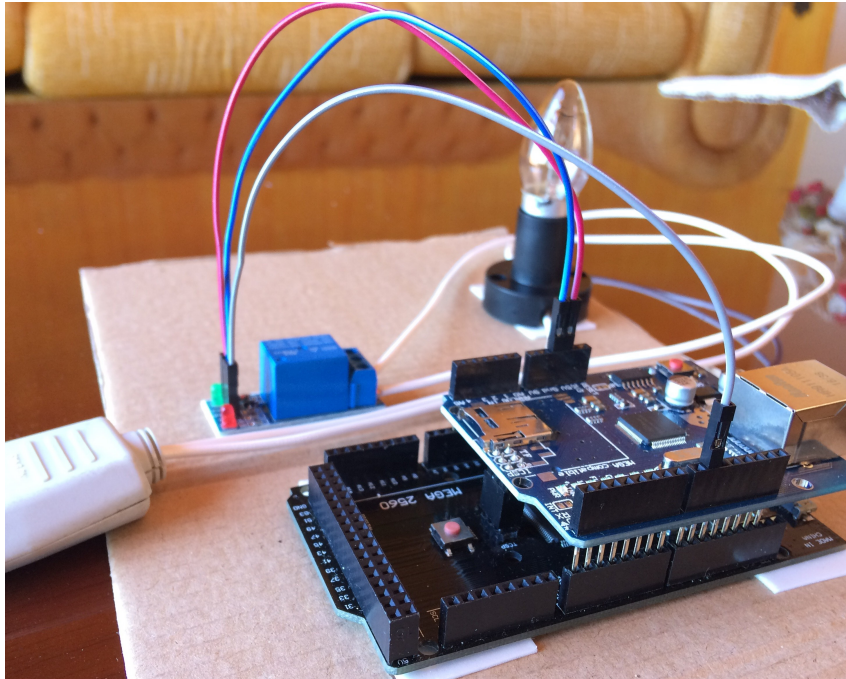


Figura 15 – Circuito físico finalizado

Fonte: Autoria própria

O próximo passo do trabalho é fazer o Riot OS reconhecer o shield ethernet presente no circuito, e assim, fazendo com que seja possível colocar o servidor CoAP em funcionamento.

4 CONCLUSÃO

Nesse projeto, foram apresentados os conceitos e tecnologias que serão utilizados no decorrer do desenvolvimento, bem como descrições mais detalhadas das etapas e procedimentos que serão realizados para alcançar os objetivos propostos. Espera-se, ao término do trabalho, obter o sistema proposto de maneira completamente funcional, cumprindo com todos objetivos e metas estipuladas anteriormente. Como dito anteriormente, houveram algumas dificuldades e problemas nos primeiros testes com o sistema, mas que foram resolvidas e sanadas de maneira satisfatória após várias pesquisas, leituras complementares e novos testes.

A escassez de informações sobre o sistema operacional utilizado torna o processo de desenvolvimento bastante trabalhoso, visto que não existem muitos guias ou tutoriais a serem seguidos e nem mesmo muito material acadêmico ou científico sobre, apenas a documentação e alguns artigos mais gerais sobre a tecnologia. Porém, por se tratar de tecnologias novas, existe uma grande possibilidade de crescimento e maior informação técnica disponível, visto que várias pessoas também estão fazendo testes e contribuindo com documentações e melhorias no sistema. O principal apelo do projeto é a exploração e trabalho com tecnologias recentes que, no futuro podem vir a ser úteis para a área de IoT como um geral, e até mesmo revolucionar o mercado da tecnologia voltada para a Internet das Coisas.

Por fim, são esperadas dificuldades no desenvolvimento do *Web Service*, na utilização do protocolo CoAP, e principalmente na construção do algoritmo em linguagem C, porém, acredita-se que com trabalho e pesquisa será possível chegar ao objetivo definido.

4.1 CONSIDERAÇÕES FINAIS

A IoT é uma área bastante nova de estudos e pesquisas. Vários campos ainda precisam ser estudados para que se alcance o objetivo desejado pela IoT, como por exemplo, a padronização dos protocolos de comunicação, melhorias na segurança das redes, redução do consumo de energia elétrica, redução de custos de equipamentos. Acredita-se que a IoT irá revolucionar vários setores e trará um aumento considerável de oportunidades de trabalho para profissionais da área de tecnologia, pois, para projetos de grande porte, irão ser necessárias atividades de vários campos de conhecimento dentro da computação. Sendo assim, o estudo das novas tecnologias empregadas neste trabalho é um importante primeiro passo para a evolução da área.

Referências

- ALVES, P.; GONÇALVES, R. Sistema de agendamento de gravações para o projecto educast. ipb. 2012. Disponível em: <<http://projinf.estig.ipb.pt/~a22550a21416/relatorio.pdf>>. Acesso em: 6 de novembro de 2017. Citado 2 vezes nas páginas 13 e 14.
- ARDUINO. **Arduino MEGA 2560**. 2017. Disponível em: <<https://www.arduino.cc/en/Main/arduinoBoardMega2560/#>>. Acesso em: 11 de setembro de 2017. Citado na página 8.
- BACCELLI, E. et al. Riot os: Towards an os for the internet of things. **32nd IEEE International Conference on Computer Communications (INFOCOM)**, 2013. Disponível em: <<https://riot-os.org/docs/riot-infocom2013-abstract.pdf>>. Acesso em: 12 de setembro de 2017. Citado 2 vezes nas páginas 10 e 11.
- BOOTH, D. et al. Web services architecture. 2004. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso em: 12 de setembro de 2017. Citado na página 13.
- CARVALHO, R. **O que são relés - Tipos e usos**. 2015. Disponível em: <<http://www.centraldalapa.com/pagina/o-que-sao-reles-tipo-e-usos/>>. Acesso em: 13 de setembro de 2017. Citado na página 9.
- CHEN, X. Constrained application protocol for internet of things. 2014. Disponível em: <<https://www.cse.wustl.edu/~jain/cse574-14/ftp/coap/index.html>>. Acesso em: 10 de novembro de 2017. Citado 2 vezes nas páginas 11 e 12.
- DENG, H. et al. Design of led street lamp intelligent control system based on zigbee technology. **2016 3rd International Conference on Engineering Technology and Application**, v. 5, 2016. Citado na página 17.
- ENERGY, S. H. **What is a Smart Home?** 2016. Disponível em: <<http://smarthomeenergy.co.uk/what-smart-home>>. Acesso em: 14 de setembro de 2017. Citado na página 6.
- FRITZING. **Fritzing - Electronics Made Easy**. 2017. Disponível em: <<http://fritzing.org/home/>>. Acesso em: 8 de setembro de 2017. Citado na página 21.
- GARTNER. **Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020**. 2013. Disponível em: <<http://www.gartner.com/newsroom/id/2636073>>. Acesso em: 14 de setembro de 2017. Citado na página 5.
- IKEA. **Smart Lighting - IKEA**. 2017. Disponível em: <<http://www.ikea.com/us/en/catalog/categories/departments/lighting/36812/>>. Acesso em: 11 de novembro de 2017. Citado 2 vezes nas páginas 17 e 18.
- JAPPE, R. **IKEA: Diferencial estratégico no mercado de móveis**. 2014. Disponível em: <<http://designculture.com.br/ikea-diferencial-estrategico-no-mercado-de-moveis>>. Acesso em: 11 de novembro de 2017. Citado na página 17.
- LACERDA, F.; LIMA-MARQUES, M. Da necessidade de princípios de arquitetura da informação para a internet das coisas. **Perspectivas em Ciência da Informação**, v. 20, 2015. Citado na página 5.

- LELYLAN. **Lelylan - Open Source Internet of Things**. 2017. Disponível em: <<http://www.lelylan.com/>>. Acesso em: 16 de setembro de 2017. Citado 2 vezes nas páginas 16 e 17.
- LIMA, I. **Módulo relé**. 2015. Disponível em: <<http://autocorerobotica.blog.br/modulo-rele/>>. Acesso em: 13 de setembro de 2017. Citado na página 9.
- MANCINI, M. Internet das coisas: História, conceitos, aplicações e desafios. 2017. Disponível em: <<https://pmisp.org.br/documents/acervo-arquivos/241-internet-das-coisas-historia-conceitos-aplicacoes-e-desafios/file>>. Acesso em: 11 de setembro de 2017. Citado na página 1.
- MARTINS, I. R.; ZEM, J. L. Estudo dos protocolos de comunicação mqtt e coap para aplicações machine-to-machine e internet das coisas. 2014. Citado na página 11.
- MARTIRANO, L. A smart lighting control to save energy. **The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications**, 2011. Citado na página 7.
- MELLO, D. A. P. de. Solução para monitoramento ambiente utilizando arduino. 2016. Acesso em: 6 de novembro de 2017. Citado 2 vezes nas páginas 14 e 15.
- MENDES, T. D. et al. Smart home communication technologies and applications: Wireless protocol assessment for home area network resources. **Energies** **2015**, 2015. Citado na página 7.
- MOWAMAD, M. A. E.-L.; FATHY, A.; HAFEZ, A. Smart home automated control system using android application and microcontroller. **International Journal of Scientific & Engineering Research**, v. 5, 2014. Citado na página 5.
- MULANI, T. T.; PINGLE, S. V. Internet of things. **International Research Journal of Multidisciplinary Studies**, v. 2, 2016. Citado 2 vezes nas páginas 4 e 5.
- NASCIMENTO, M. T. V. Análise comparativa de protocolos em smart home: Considerações em conectividade. v. 1, 2016. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/171417/TCC_MARCO_FIM.pdf>. Acesso em: 11 de setembro de 2017. Citado na página 1.
- PACIEVITCH, Y. **HTML**. 2016. Disponível em: <<https://www.infoescola.com/informatica/html/>>. Acesso em: 31 de outubro de 2017. Citado na página 12.
- PASQUA, D. V. Webservices rest e php, introdução. 2015. Disponível em: <<http://www.douglaspasqua.com/2015/11/20/webservices-rest-e-php-introducao/>>. Acesso em: 12 de setembro de 2017. Citado na página 13.
- RALLO, R. **Material Design: aprenda tudo sobre o design do Google!** 2017. Disponível em: <<https://marketingdeconteudo.com/material-design/>>. Acesso em: 11 de novembro de 2017. Citado na página 25.
- REIS, F. dos. **Arduino - Conhecendo os Shields**. 2015. Disponível em: <<http://www.bosontreinamentos.com.br/eletronica/arduino/arduino-conhecendo-os-shields/>>. Acesso em: 12 de setembro de 2017. Citado na página 8.

ROBLES, R. J.; KIM, T. hoon. Applications, systems and methods in smart home technology: A review. **International Journal of Advanced Science and Technology**, v. 15, 2010. Disponível em: <https://www.researchgate.net/profile/Rosslin_Robles/publication/242630611_Applications_Systems_and_Methods_in_Smart_Home_Technology_A_Review/links/582c24e908ae138f1bf4cc79/Applications-Systems-and-Methods-in-Smart-Home-Technology-A-Review.pdf>. Acesso em: 11 de setembro de 2017. Citado na página 1.

SCHOUTSEN, P. Ikea trådfri: Internet of things done right. 2017. Disponível em: <<https://home-assistant.io/blog/2017/04/17/ikea-tradfri-internet-of-things-done-right/>>. Acesso em: 11 de novembro de 2017. Citado 2 vezes nas páginas 17 e 18.

THOMSEN, A. **O que é Arduino?** 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 11 de setembro de 2017. Citado na página 8.

THOMSEN, A. **Automação Residencial com Arduino: acenda lâmpadas pela internet.** 2015. Disponível em: <<https://www.filipeflop.com/blog/automacao-residencial-com-arduino-acenda-lampadas-pela-internet/>>. Acesso em: 16 de setembro de 2017. Citado 2 vezes nas páginas 15 e 16.

TULY, K. F. A survey on novel services in smart home (optimized for smart electricity grid). 2016. Citado 3 vezes nas páginas 5, 6 e 7.