

Bárbara Conceição Cardoso Martins

Fundamentos da Internet das Coisas

São Cristóvão-SE, Brasil

Maio de 2017

Bárbara Conceição Cardoso Martins

Fundamentos da Internet das Coisas

Trabalho de Conclusão de Curso II de Graduação apresentado ao Departamento de Engenharia Elétrica (DEL) da Universidade Federal de Sergipe, como parte dos requisitos necessários à obtenção do título de Graduada em Engenharia Eletrônica.

Universidade Federal de Sergipe – UFS
Departamento de Engenharia Elétrica – DEL

Orientador: Antonio Ramirez Hidalgo

São Cristóvão-SE, Brasil

Maio de 2017

Fundamentos da Internet das Coisas

Bárbara Conceição Cardoso Martins

Trabalho de Conclusão de Curso II de Graduação apresentado ao Departamento de Engenharia Elétrica (DEL) da Universidade Federal de Sergipe, como parte dos requisitos necessários à obtenção do título de Graduada em Engenharia Eletrônica. Examinado por:

Prof. Antonio Ramirez Hidalgo

Prof. Jânio Coutinho Canuto

Prof. Ícaro de Vasconcelos Brito

São Cristóvão-SE, Brasil

Maio de 2017

Agradecimentos

É difícil agradecer todas as pessoas que de algum modo, nos momentos serenos e/ou apreensivos, fizeram ou fazem parte da minha vida, por isso primeiramente agradeço a Deus por me rodear de pessoas maravilhosas durante toda essa jornada.

Agradeço aos meus pais por toda dedicação e paciência, por compartilhar das minhas conquistas, me dar conforto nos momentos de fraqueza, compreender as minhas ausências, e principalmente, por acreditar e incentivar os meus sonhos. Aos meus irmãos, que por mais difíceis que sejam as circunstâncias sempre estão comigo. Às minhas avós, primo(a)s e tio(a)s por todo apoio. Em especial a tia Ana pelo carinho, mas principalmente por tornar tudo isso possível, serei eternamente grata.

Agradeço também a Vinícius, por toda paciência, palavras de incentivo, por me fazer sorrir mesmo quando tudo estava dando errado, por estar sempre disposto a ajudar, mas principalmente, por acreditar que seria possível quando eu já não acreditava.

Sou grata a Brenda, por apesar da distância (e minha ausência) ter mantido a nossa amizade como se estivéssemos sempre por perto.

Aos amigos que fiz na graduação, jamais esquecerei pelas risadas, brincadeiras, confidências...de como vocês tornaram essa caminhada tão mais simples. Agradeço em especial a Vitão, por me ajudar sempre que precisei, pelas nossas conversas, simplesmente sou grata por nossa amizade incondicional.

Não poderia deixar de agradecer às pessoas que contribuíram para desenvolvimento deste trabalho. Agradeço a Thiago por ter compartilhado comigo seu conhecimento, e a Emission por toda dedicação em me ajudar; por tomar pra si as minhas dificuldades, meu muito Obrigada!

Por fim, agradeço a todos os professores que fizeram parte da minha formação, que contribuíram para a profissional que serei, especialmente ao meu orientador Ramirez Hidalgo por todos os ensinamentos.

Resumo

A Internet das Coisas, ou Internet of Things (IoT), é apresentada como a evolução da Internet, de tal forma que a interação homem-máquina deixa de ser o foco principal. Neste novo conceito, a internet passa a ser observada através da perspectiva máquina-máquina, tendo por objetivo a conexão entre objetos "inteligentes", capazes de se comunicar entre si e com o ambiente à sua volta. Uma vez conectados, os objetos formam sistemas autônomos capazes de proporcionar uma melhor qualidade de vida. Porém, desde que foi inicialmente idealizada, a Internet das Coisas apresenta problemas em sua implementação que ainda não permitiram a total disseminação desse novo conceito de Internet. Sendo assim, neste trabalho é feito um estudo sobre a arquitetura da Internet das Coisas, seus desafios e expectativas para o futuro. Além disso é desenvolvido um sistema inteligente baseado nos conceitos da IoT, que será capaz de controlar, a partir de dados sensoriais, a temperatura e iluminação de um determinado ambiente via Wi-Fi.

Palavras-Chave: Internet das Coisas. Arquitetura IoT. Plataformas IoT. Controle de Temperatura e Iluminação. Arduino.

Lista de ilustrações

Figura 1.1 – Ilustração da Internet das Coisas. (DOUKAS, 2012)	2
Figura 2.1 – Interação entre o ambiente físico, virtual e de rede de comunicação para os sistemas da IoT (WAKA, 2015).	7
Figura 2.2 – Divisão da arquitetura IoT em ordem crescente de abstração (WAKA, 2015).	7
Figura 2.3 – Módulo RFID e sua antena (DOUKAS, 2012).	10
Figura 2.4 – Módulo de Comunicação <i>Bluetooth</i> (DOUKAS, 2012).	11
Figura 2.5 – Módulo de Comunicação <i>Zigbee</i> (DOUKAS, 2012).	12
Figura 2.6 – Módulo de Comunicação Wi-Fi(DOUKAS, 2012).	12
Figura 2.7 – Funcionamento Básico do Protocolo MQTT (BARROS, 2015).	14
Figura 3.1 – Diagrama de blocos do sistema desenvolvido (Próprio Autor).	19
Figura 3.2 – Diagrama de blocos do <i>Hardware</i> do Sistema (Próprio Autor).	22
Figura 3.3 – Gráfico da Resposta do LDR (PATSKO, 2014).	23
Figura 3.4 – Divisor de Tensão (PATSKO, 2014).	24
Figura 3.5 – Circuito para aquisição dos dados do sensor LDR (Próprio Autor). . .	25
Figura 3.6 – Identificação dos pinos do sensor DHT11 (AOSONG, 2016).	26
Figura 3.7 – Circuito para a aquisição dos dados do sensor DHT11 (Próprio Autor). .	26
Figura 3.8 – Identificação das entradas/saídas e dos dispositivos presentes na placa Arduino Mega (SOUZA, 2014).	28
Figura 3.9 – Identificação dos pinos do receptor infravermelho TSOP1737 (VISHAY, 2016).	30
Figura 3.10–Circuito para decodificar os sinais de controle do ar condicionado (Próprio Autor).	30
Figura 3.11–Controle remoto usado na decodificação dos sinais de comando do ar condicionado (Próprio Autor).	31
Figura 3.12–Vetor correspondente a sinal do comando desligar do ar condicionado Hitachi (Próprio Autor).	32
Figura 3.13–Esquema do circuito para emissão dos dados através de um LED emissor de infravermelho (Próprio Autor).	32
Figura 3.14–Protótipo da cortina motorizada (Próprio Autor).	33
Figura 3.15–Configuração de uma ponte H de quatro chaves (PATSKO, 2006). . .	34
Figura 3.16–Módulo ponte H com o circuito integrado L9110 (RAMÍREZ, 2015). .	35
Figura 3.17–Esquemático do circuito para o controle de rotação do motor CC da cortina (Próprio Autor).	36
Figura 3.18–Representação dos pinos do ESP8266 (DOUKAS, 2012).	37

Figura 3.19–Esquemático do circuito para transmissão dos dados entre a plataforma e o microcontrolador (Próprio Autor).	38
Figura 3.20–Diagrama de blocos do <i>Software</i> (Próprio Autor).	39
Figura 3.21–Janelas para as duas opções de visualização dos dados dos sensores (Próprio Autor).	40
Figura 3.22–Interface de comunicação da plataforma que é disponibilizada ao usuário (Próprio Autor).	42
Figura 3.23–Diagrama de blocos da lógica de controle da iluminação (Próprio Autor).	42
Figura 3.24–Circuito para realizar a atualização do <i>firmware</i> do ESP8266 (Próprio Autor).	44
Figura A.1–Esquema elétrico dos circuitos do sistema integrados.	53

Lista de tabelas

Tabela 2.1 – Plataformas para Aplicação IoT (SANTOS, 2014).	15
Tabela 3.1 – Principais características do LDR (SUNROM, 2008).	23
Tabela 3.2 – Comparação entre as características do Arduino UNO e do Arduino Mega (ARDUINO, 2016a).	29
Tabela 3.3 – Consumo de corrente do ESP8266, na transmissão de dados, para di- ferentes padrões de comunicação (ESPRESSIF, 2017).	38
Tabela 3.4 – Faixas de valores adotados para a representação da luminosidade (Pró- prio Autor).	40
Tabela 3.5 – Configuração da conexão entre o ESP8266 e o Arduino para realizar a atualização do <i>firmware</i> (Próprio Autor).	45
Tabela 3.6 – Lista de comandos AT para configurar o ESP8266 (ESPRESSIF, 2016).	45

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	3
1.2	Organização do Trabalho	3
1.3	Metodologia	4
2	INTERNET DAS COISAS	5
2.1	Cronograma de Evolução	5
2.2	Arquitetura da Internet das Coisas	6
2.2.1	Camada de Borda	7
2.2.2	Camada de Gateway de Acesso	8
2.2.3	Camada de Internet	8
2.2.4	Camada de Middleware	8
2.2.5	Camada de Aplicação	8
2.3	Tecnologias de Hardware da IoT	9
2.3.1	Sensores	9
2.3.2	Unidade de Processamento	9
2.3.3	Unidade de Comunicação	10
2.3.4	Fontes de Energia	13
2.4	Protocolos da Camada de Aplicação	13
2.5	Nuvem	14
2.6	Plataformas para IoT	15
2.7	Aplicações: A Internet das Coisas e a Indústria	16
2.8	Os Desafios da Internet das Coisas	17
2.9	Impactos e Expectativas da Internet das Coisas	17
3	ESTUDO DE CASO: SISTEMA PARA CONTROLE DE ILUMINAÇÃO E TEMPERATURA VIA INTERNET	19
3.1	Descrição do Sistema	19
3.2	Definição da Plataforma do Sistema	20
3.3	Metodologia do <i>Hardware</i>	22
3.3.1	Implementação dos circuitos de entrada do sistema.	22
3.3.1.1	LDR	23
3.3.1.2	DHT11	25
3.3.2	Unidade de processamento	27
3.3.3	Implementação dos circuitos de saída do sistema.	29
3.3.3.1	Ar Condicionado	29

3.3.3.2	Cortina	33
3.3.4	Unidade de Comunicação	36
3.3.5	Integração do <i>Hardware</i>	38
3.4	Metodologia do <i>Software</i>	38
3.4.1	Aquisição dos dados dos sensores	39
3.4.2	Aquisição dos comandos da plataforma	40
3.4.3	Lógica de controle da iluminação	42
3.4.4	Lógica de controle do ar condicionado	43
3.4.5	Transmissão dos dados entre o sistema e a plataforma	44
	4 CONCLUSÕES E CONSIDERAÇÕES FINAIS	46
4.1	Conclusões	46
4.2	Propostas para Projetos Futuros	47
	REFERÊNCIAS	49
	A DIAGRAMA DO CIRCUITO	53
	B CÓDIGO DO <i>SOFTWARE</i>	54

Capítulo 1

Introdução

Os avanços tecnológicos permitiram o acesso à informação de forma cada vez mais rápida e simplificada. Sendo assim, estar informado sobre tudo ao seu redor passou a ser uma necessidade do homem moderno. Saber a previsão do tempo antes de sair de casa, ou como estará o trânsito na rota para o trabalho, ou até mesmo a temperatura de um determinado ambiente, são alguns exemplos de informações que passaram a fazer parte do cotidiano das pessoas. Para que isso fosse possível, o avanço da internet foi, e tem sido, fundamental.

Todavia, com o surgimento da Internet das Coisas (IoT) essa realidade passa a ser um pouco diferente. Enquanto hoje o foco principal são as pessoas se conectando à rede mundial de computadores por meio de dispositivos tais como celulares, *tablets*, notebooks, etc; a IoT objetiva conectar os próprios aparelhos do cotidiano à essa rede, para que eles interajam uns com os outros e assim possam obter informações relevantes ao dia a dia do usuário. Mas a IoT não se limita à aquisição e análise de dados. Sistemas baseados nessa tecnologia também são capazes de atuar automaticamente sobre o meio no qual estão inseridos baseados nas configurações estabelecidas e nos dados armazenados ao longo da sua atuação.

De acordo com (DOUKAS, 2012), pode-se afirmar que a IoT é uma revolução tecnológica onde o foco muda da interação homem-máquina e passa a ser observada da perspectiva máquina-máquina. Por exemplo, os carros modernos possuem sensores que determinam quão bem o carro está atuando, tanto para o consumo de combustível quanto para a pressão dos pneus, e essas informações são enviadas ao usuário para que este tenha controle sobre a situação. Já os carros com sistema IoT embarcado poderiam atuar sobre possíveis falhas sem a intervenção do usuário. Neste caso, o automóvel possui um computador que não só capta dados, mas também é capaz de interagir com serviços externos de forma inteligente e autônoma.

Para que seja possível essa interação, é necessário que dispositivos sensoriais e

atuadores sejam embarcados nos objetos físicos, e programados para agirem de acordo com a finalidade da aplicação de cada um deles. Nesse ponto, entende-se como “objeto” tudo que está à sua volta e que possa, mediante sensores, atuadores e microcontroladores, captar, processar, transmitir e armazenar dados, além de se adaptar e interagir com o meio à sua volta. Na Figura 1.1 é observado uma representação de como a Internet das Coisas se organiza.

Dispositivos eletrônicos são aplicados ao ambiente no qual se deseja conectar à internet. Alguns desses dispositivos serão responsáveis pelo envio de informações do ambiente à internet, onde esses dados são analisados e, através de interfaces de comunicação, disponibilizados aos usuários para que eles também possam monitorar e controlar os ambientes. Neste caso, o controle e monitoramento podem ser realizados a qualquer momento, desde que tenha conexão com a rede na qual o sistema encontra-se conectado.

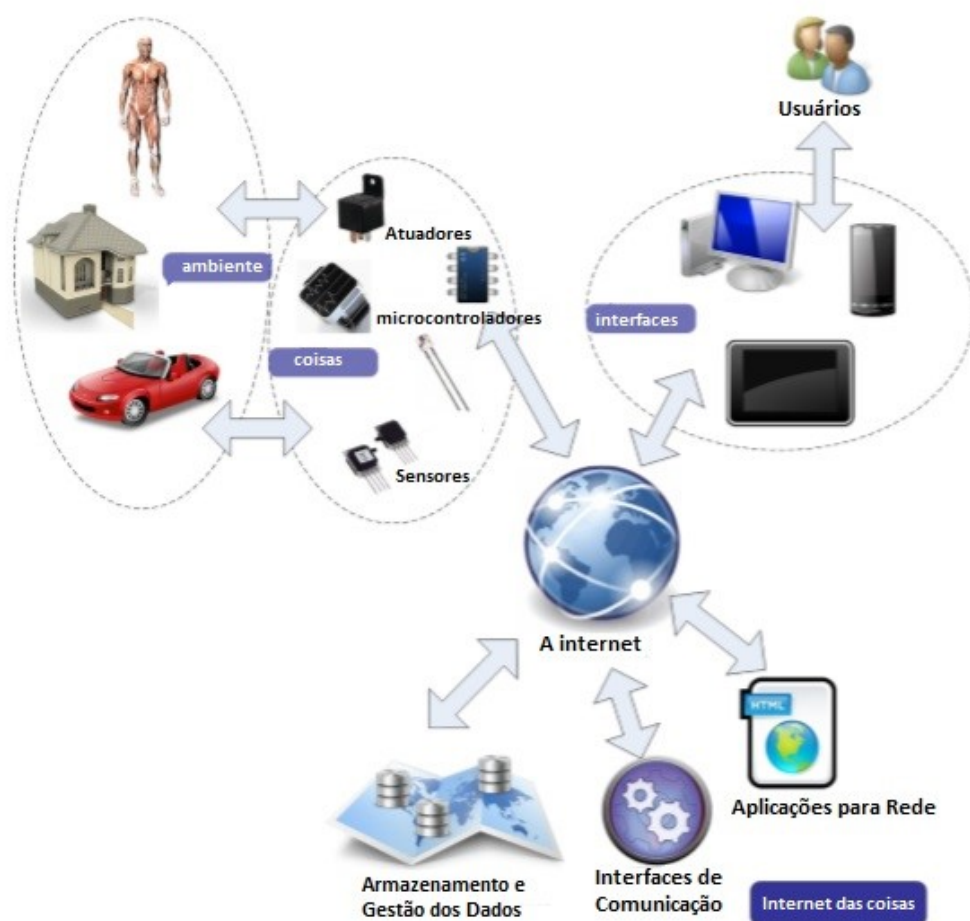


Figura 1.1 – Ilustração da Internet das Coisas. (DOUKAS, 2012)

Apesar da Internet das Coisas ainda ser compreendida como uma tecnologia voltada para aplicações de cunho pessoal, estando limitada às rotinas diárias, à exemplo da automação residencial, esta tecnologia não se restringe a esse meio. As possibilidades de

aplicações utilizando a IoT são inúmeras, podendo ser implementadas nas mais diferentes áreas possíveis, tais como, a indústria, agricultura, medicina, entre outras. Inclusive, essa tecnologia tem sido amplamente aplicada na área da medicina, facilitando diagnósticos e tratamentos.

Entretanto, segundo (DOUKAS, 2012), a IoT ainda apresenta uma série de problemas e desafios a serem enfrentados para total disseminação desta tecnologia. A grande diversidade de *hardware* e *software*, por exemplo, dificulta o desenvolvimento de aplicações capazes de interagir entre si. Devido a esta dificuldade tem surgido diversas plataformas de comunicação com o objetivo de facilitar a troca de informações entre os diferentes tipos de dispositivos de um ambiente.

1.1 Objetivos

O objetivo geral deste trabalho é compreender os elementos que constituem a Internet das Coisas, explorando sua estrutura e os dispositivos que compõem essa tecnologia, e apresentando os desafios e expectativas acerca do tema. Como complementariedade para a compreensão dos conceitos explorados, este TCC tem por objetivo específico a realização de um estudo de caso, que se trata da implementação de um sistema para controle de iluminação e temperatura, para exemplificar como os conceitos da IoT podem ser aplicados na prática.

1.2 Organização do Trabalho

No Capítulo 1 é apresentada uma introdução sobre a Internet das Coisas, fazendo uma breve explanação sobre como essa tecnologia se organiza, suas aplicações e possíveis desafios. Temas que serão melhor explorados nos outros capítulos deste TCC.

No Capítulo 2 é feita uma revisão sobre todo o conceito da IoT, explorando em ordem cronológica os fatos mais relevantes acerca do assunto até os dias atuais. Neste capítulo também é observado como se estrutura a IoT, analisando cada uma das camadas que formam sua arquitetura. Além disso, também são relatados os principais componentes que fazem parte dessa nova tecnologia, bem como as suas plataformas de comunicação. Os desafios, impactos econômicos, expectativas e possibilidade de aplicações para o futuro também são temas abordados neste capítulo.

O Capítulo 3 é dedicado ao estudo de caso. Nele é feita a descrição do sistema para o controle de iluminação e temperatura via Wi-Fi. Além disso, também é apresentada a plataforma de internet utilizada, bem como os materiais e métodos empregados no desenvolvimento do sistema.

No Capítulo 4 são descritas as conclusões acerca da Internet das Coisas e do estudo de caso realizado. Neste capítulo também são apresentadas propostas para trabalhos futuros.

1.3 Metodologia

Para desenvolvimento deste TCC foram realizados estudos baseados em livros e pesquisas na Internet que serviram de base para a melhor compreensão do tema. Além disso, os conhecimentos adquiridos ao longo da graduação também foram de fundamental importância para a composição desta atividade.

No que diz respeito ao estudo de caso realizado, inicialmente foram feitas pesquisas sobre projetos já desenvolvidos que envolviam os conceitos a serem empregados. Livros práticos tais como (SCHWARTZ, 2014) e (NORRIS, 2015), também foram usados para se obter familiaridade com os dispositivos empregados no projeto.

Em seguida foram determinados os dispositivos que seriam usados. Estes componentes foram escolhidos com base em seus custos e acessibilidade. Sendo assim, escolheu-se para o processamento dos dados a plataforma de *hardware* Arduino e o para a comunicação via Wi-Fi o módulo ESP8266, que vem sendo bastante aplicado nos projetos de IoT.

Após a definição desses dois *hardwares* que são cruciais para o sistema, foi definida, com base na comparação de três importantes plataformas para Internet das Coisas, qual seria usada no estudo de caso.

Com todos os componentes definidos, deu-se início aos estudos e montagem dos circuitos que compõem o projeto para controle de iluminação e temperatura via Wi-Fi.

Capítulo 2

Internet das Coisas

Neste capítulo é feita uma revisão bibliográfica acerca do tema. Nele é descrito a evolução da Internet das Coisas, os desafios a serem enfrentados para que se consolide no meio tecnológico, as expectativas quanto ao futuro e algumas possíveis aplicações. Além disso, são apresentadas algumas das tecnologias, protocolos e plataformas que a compõem.

2.1 Cronograma de Evolução

O termo Internet das Coisas foi usado pela primeira vez em 1999 pelo pesquisador Kevin Ashton durante uma apresentação para difundir a então inovadora tecnologia de identificação por rádio frequência, popularmente conhecida como *RFID* (*Radio Frequency Identification*). Desde então, o termo vem sendo amplamente usado para descrever sistemas nos quais objetos são conectados à internet.

Apesar do termo ter surgido apenas em 1999, o conceito por traz dele já vinha sendo explorado há muito tempo. Em 1926, de acordo com (PRESS, 2014), o físico Nikola Tesla fez a seguinte declaração em uma entrevista à revista *Colliers Magazine*:

“Quando a comunicação sem fio for aplicada perfeitamente toda a terra será convertida em um enorme cérebro, que na verdade é, todas as coisas sendo partículas de um todo real e rítmico... e os instrumentos através dos quais seremos capazes de fazer isso serão incrivelmente simples em comparação com o nosso telefone atual. Um homem será capaz de transportar um no bolso do colete”.

Neste momento, Tesla visava o mundo completamente conectado, se tornando um só conjunto onde tudo pudesse ser visto como partícula deste, basicamente ele previa o que hoje nomeia-se de Internet das Coisas.

Segundo (RIJMENAM, 2014), alguns anos após a esta declaração, objetos comuns passaram a receber sensores capazes de aprimorar seu funcionamento, um exemplo disto

foram os óculos desenvolvidos por Hubert Upton, em 1967, capaz de auxiliar na leitura labial através de um *display* de montagem.

Porém, foi apenas em 1974, com o surgimento da internet como conhecemos hoje, que a possibilidade de interconectividade entre os objetos, já prevista por Tesla, tornou-se viável. Sendo assim, a busca pela implantação de recursos adicionais aos objetos físicos aumentou.

Com o passar dos anos, os avanços nas pesquisas relacionadas ao conceito da Internet das Coisas e a evolução tecnológica, principalmente na área da eletrônica, que proporcionou sensores cada vez mais eficientes e menores, permitiram o desenvolvimento de sistemas baseados nos conceitos da Internet das Coisas, mostrando ao mundo que é possível um ambiente conectado. Ainda de acordo com (RIJMENAM, 2014), “A Casa do Futuro” projetada por Chriet Titulaer e construída em 1989, foi um desses sistemas. Uma casa inteligente que tem por foco a interação homem máquina através do reconhecimento de voz.

Em 1999, após o surgimento do termo Internet das Coisas, um grande interesse foi despertado sobre o assunto, o que fez com que empresas investissem em pesquisas e desenvolvimento de dispositivos interconectados à internet. Apesar disso, nem todos esses dispositivos podem ser realmente classificados dentro do conceito de Internet das Coisas, isso porque nem todos são capazes de interagir com o meio a sua volta de forma autônoma.

Atualmente, mesmo com todos esses empecilhos, a Internet das Coisas ainda é vista como a internet do futuro. Embora inúmeras aplicações tenham surgido desde 1999, ainda há uma limitação no que diz respeito à quantidade de dispositivos que podem ser conectados à Internet. Para contornar esse problema, um investimento em um protocolo de internet com um número de endereços possíveis maior, o IPV6, tem sido o foco das pesquisas atuais. Apesar dessa atual limitação, de acordo com (IEEE, 2016), em 2020 haverá bilhões de objetos interagindo entre si através da internet.

2.2 Arquitetura da Internet das Coisas

A Internet das Coisas é uma rede que objetiva conectar vários objetos físicos com funcionalidades diferentes num mesmo ambiente virtual, e ainda fazer com que estes objetos interajam entre si. Entretanto, este último é seu grande desafio, uma vez que cada objeto possui seu próprio padrão de conexão, principalmente no meio industrial. Isso porque esta área apresenta uma diversidade de máquinas com funcionalidades diferentes que tornaria inviável a aplicação da IoT.

Na Figura 2.1 é mostrada a interação existente na implementação da Internet das

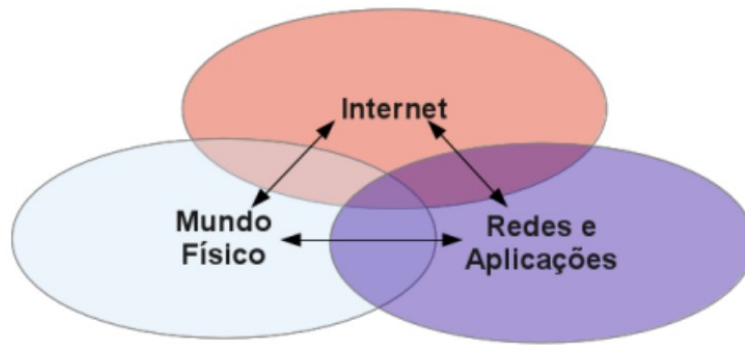


Figura 2.1 – Interação entre o ambiente físico, virtual e de rede de comunicação para os sistemas da IoT (WAKA, 2015).

Coisas entre o mundo físico, virtual e a rede de comunicação.

Diante desse meio heterogêneo, para que a implementação da IoT seja viável, é necessário que ela possua uma arquitetura muito bem estruturada, que vise toda essa variedade de protocolos e padrões. Diante disso, em (WAKA, 2015) foi apresentada uma divisão da arquitetura da IoT em 5 camadas: camada de aplicação, camada de *middleware*, camada de internet, camada de *gateway* de acesso e a camada de borda (Figura 2.2). Cada uma dessas camadas será melhor detalhada nos tópicos seguintes.

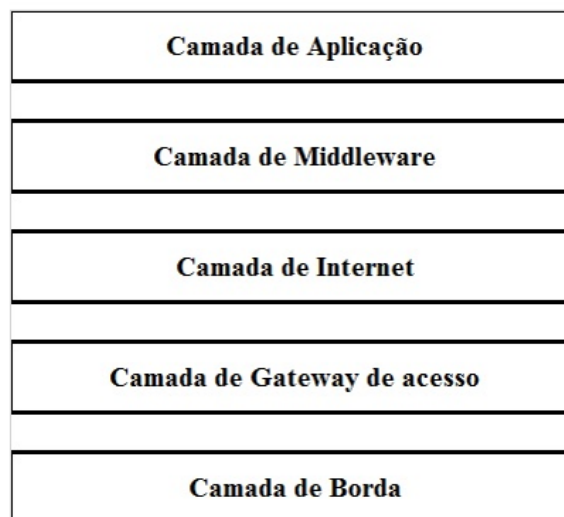


Figura 2.2 – Divisão da arquitetura IoT em ordem crescente de abstração (WAKA, 2015).

2.2.1 Camada de Borda

A camada de borda é a camada de nível mais baixo desta arquitetura, ou seja, a mais concreta. Ela está em contato direto com a aquisição de dados que determinado objeto está designado a processar. Além disso, a camada de borda também é capaz de

processar informação e dar suporte à comunicação. Sistemas embarcados, etiquetas RFID e sensores no geral são alguns dos *hardwares* que compõem esta camada. (ZARGHAMI, 2013)

2.2.2 Camada de Gateway de Acesso

A camada de *gateway* pode ser vista como aquela que possibilita na prática o uso de objetos com tecnologias de comunicação diferentes. Para tal, ela faz uma tradução da tecnologia que está sendo usada para os níveis de compreensão da Internet, e assim envia os dados obtidos para a camada imediatamente superior (ver Figura 2.2). Pensando do ponto de vista de redes de computadores, esta camada é considerada uma prestadora de serviço.

2.2.3 Camada de Internet

A camada de internet é a responsável por conectar os objetos à rede mundial, permitindo que estes interajam entre si e/ou com o meio ao qual estão inseridos por meio de plataformas virtuais (SANTOS, 2014).

2.2.4 Camada de Middleware

A camada de *middleware* é como se fosse um filtro final antes que as informações adquiridas pelo hardware sejam repassadas aos usuários por meio da camada de aplicação. Sua função é controlar e monitorar o estado dos dispositivos com a finalidade de se obter a melhor interoperabilidade, escalabilidade e mobilidade possível das partes envolvidas (FERREIRA, 2014).

2.2.5 Camada de Aplicação

A camada de aplicação é a última antes da execução dos serviços que são providos pelos objetos físicos, ou seja, a camada mais próxima do usuário final. Nesta camada os dados provenientes da camada de *middleware* são processados e em seguida são prestados os serviços aos quais determinado objeto foi predisposto a realizar. É nesta camada que observa-se a Internet das Coisas tomando formas reais.

2.3 Tecnologias de Hardware da IoT

Para que a Internet das Coisas seja implementada é necessário a presença de certos dispositivos eletrônicos que tornem possível a execução dos seus conceitos.

Dentre os inúmeros dispositivos que podem ser agregados aos objetos, e que são capazes de juntos proporcionar a estes funcionalidades em meio virtual, os mais básicos são: sensores, unidades de processamento, atuadores, unidades de comunicação e fontes de energia.

2.3.1 Sensores

A difusão dos sensores em meio à Internet das Coisas se deve ao fato de que estes dispositivos são facilmente manipulados e, por estarem cada vez menores, podem ser acoplados nos mais diferentes objetos sem maiores problemas. Além disso, os sensores normalmente consomem pouca energia.

Os dispositivos sensoriais são os componentes mais básicos da Internet das Coisas. Normalmente, eles apresentam uma entrada analógica, ou seja eles são capazes de mensurar variáveis físicas, tais como pressão, temperatura e umidade, e em alguns casos converter essas informações em dados digitais (DOUKAS, 2012).

2.3.2 Unidade de Processamento

Uma unidade de processamento é um “computador”, com todas as partes que o compõem, tal como o processador, memória e periféricos de saída e entrada programáveis. No entanto, para a Internet das Coisas, além dessas características essa unidade deve ser capaz proporcionar mobilidade, visando que esses dispositivos serão acoplados a objetos físicos já existentes. Pensando assim, os computadores usados precisam ser pequenos, sendo necessário o uso de microcontroladores, que não passam de pequenos computadores produzidos em circuitos integrados (DOUKAS, 2012).

A unidade de processamento tem por função receber os dados coletados pelos sensores, processá-los, tomar decisões localmente e enviar os dados processados para uma nuvem de dados online. Diante disso, sua função é de extrema importância para aplicação de Internet das Coisas porque caso ela não existisse os sensores enviariam os dados coletados direto para a nuvem, sem haver uma filtragem prévia.

2.3.3 Unidade de Comunicação

Para que a comunicação aconteça, ou seja, as informações mensuradas sejam repassadas, é preciso que haja uma unidade de comunicação. Isto é, um dispositivo capaz de traduzir dados reais em linguagem computacional para que esses dados sejam enviados. Essa comunicação pode ser feita de duas formas diferentes: uma é usando um *gateway*, uma espécie de “ponte” capaz de fazer a comunicação entre dois componentes que não são diretamente compatíveis; e a outra é por meio do próprio objeto usado na implementação da IoT, desde que este possua um caminho de comunicação direto (pode ser via rádio) com a internet (DOUKAS, 2012).

A capacidade de comunicação é um ponto crucial para a Internet das Coisas, porque é a partir dela que será definido a forma como os objetos irão se conectar e o tipo de tecnologia a ser usada para realizar a conexão. Essa escolha irá impactar diretamente no consumo de energia dos objetos.

Como vimos anteriormente, os dispositivos usados nos sistemas inteligente normalmente são muito pequenos, não dispendo assim de grandes cargas de energia. Diante dessa problemática, torna-se fundamental uma análise mais criteriosa no momento da definição de qual tecnologia de comunicação será aplicada, de acordo com a sua finalidade. Sendo assim, será feita a seguir uma breve análise sobre as principais tecnologias usadas, levando-se em consideração suas características mais relevantes.

- **RFID** (Radio Frequency Identification) é a tecnologia de comunicação mais antiga, ela surgiu antes mesmo do termo Internet das Coisas ser amplamente usado. Foi desenvolvido para identificar e rastrear objetos através de pequenos chips eletrônicos chamados de etiquetas (Figura 2.3).

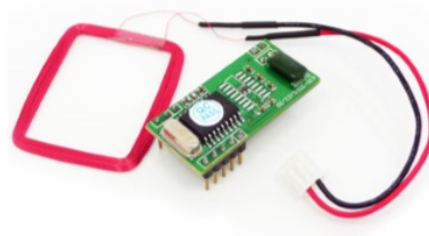


Figura 2.3 – Módulo RFID e sua antena (DOUKAS, 2012).

De acordo com (DOUKAS, 2012), as etiquetas RFID se dividem em três categorias que se diferenciam de acordo com a sua autonomia energética, sendo elas:

- i) passivas, não possuem baterias. Portanto só podem realizar transmissão de dados quando ligadas ao leitor, este proverá a energia necessária;

- ii) ativas, possuem sua própria bateria, o que permite dados serem enviados constantemente;
- iii) ativas passivas, essas possuem bateria própria, mas só realizam a transmissão de dados quando ligada ao leitor.

Essas etiquetas são relativamente simples, formadas por um módulo e uma antena. Devido a essa simplicidade e ao baixo custo, inicialmente tornou-se a tecnologia mais usada na comunicação entre os objetos.

- **Bluetooth** é uma tecnologia voltada para transmissão de dados em pequenas distâncias, e é usada tanto em objetos fixos quanto nos móveis, a exemplo dos *smartphones*. O *Bluetooth*, (Figura 2.4), é normalmente aplicado na conexão de pequenos dispositivos devido à sua capacidade de automaticamente criar redes de troca de informações, e também por causa do seu baixo consumo de energia. Apesar das restrições no que diz respeito à conexão com a internet, o *Bluetooth* vem sendo aprimorado quanto ao alcance e velocidade de dados, surgindo assim o *Bluetooth 5*. Uma versão que, de acordo com (BLUETOOTH, 2017), oferece flexibilidade para os projetos da IoT, podendo fazer ajustes em seu alcance, velocidade e segurança, de forma a atender a finalidade da aplicação. Além disso, foi desenvolvido visando-se a redução da interferência entre outras tecnologias sem fio, com o intuito de garantir sua coexistência no ambiente IoT.



Figura 2.4 – Módulo de Comunicação *Bluetooth* (DOUKAS, 2012).

- **Zigbee** é uma tecnologia com baixo custo, com baixa vazão de dados e o mais importante com um baixo consumo de energia (Figura 2.5). Um dos grandes motivos para esse baixo consumo é porque o *Zigbee* permite que os dispositivos aos quais estão integrados entrem em modo de economia de energia por longos períodos, assim a bateria acaba sendo poupada. Esse padrão de comunicação opera em frequências pré-definidas, podendo estas serem de 2.4GHz, 868 MHz ou 900 MHz.

Se comparado ao *Bluetooth*, o *Zigbee* apresenta algumas vantagens, tais como, uma melhor eficiência no consumo da energia e um raio de alcance maior, ponto muito



Figura 2.5 – Módulo de Comunicação *Zigbee* (DOUKAS, 2012).

importante a ser considerado para a Internet das Coisas. Porém, assim como o *Bluetooth*, o *Zigbee* não se conecta diretamente à internet.

- **Wi-Fi** foi desenvolvido como alternativa para o padrão cabeado *Ethernet*, possui um elevado alcance de conexão e é a tecnologia de comunicação mais usada nos últimos tempos. Apesar dessa popularização, esta tecnologia ainda não é amplamente aplicada na implementação de objetos para a IoT, isso porque durante seu desenvolvimento não houve uma preocupação quanto ao consumo limitado de energia por parte dos dispositivos que a usariam, fator que é crucial para a Internet das Coisas. Desta forma, é uma ótima opção para as aplicações fixas, as quais podem ser ligadas diretamente em fonte de energia. Na figura 2.6 é mostrado um dos módulos existentes para aplicação desse tipo de comunicação.

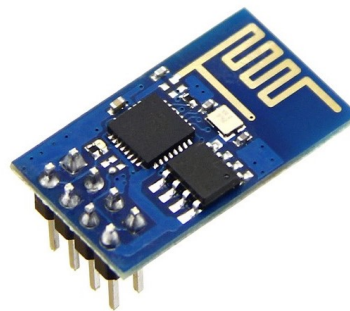


Figura 2.6 – Módulo de Comunicação Wi-Fi(DOUKAS, 2012).

- **Rede de Celular (3G/4G)** também é uma alternativa para a comunicação entre objetos, porém dentre as que já foram apresentadas esta é a menos eficiente para a Internet das Coisas. Apesar de seu alto alcance de conexão e de se conectar diretamente à internet, o consumo de energia é elevadíssimo. Mas, é uma boa alternativa, em questões financeiras, para ambientes muito afastados e com baixa mobilidade (SANTOS, 2014).

Analisando as tecnologias apresentadas, percebe-se que todas apresentam alguma dificuldade para serem implementadas nas aplicações da Internet das Coisas, seja devido ao curto alcance ou ao consumo de energia elevado. Contudo, a tecnologia Wi-Fi apresenta-se como a mais propícia a ser implementada, uma vez que dispõem de um longo alcance para comunicação se comparado com os outros dispositivos, além de já estar popularizada. Mas para que seja realmente utilizada, serão necessários alguns estudos e avanços tecnológicos que propiciem um consumo de energia mais baixo. Entretanto, o *Bluetooth* 5 também mostra-se vantajoso para as aplicações IoT, apesar de não se conectar a internet, tem apresentado outras características relevantes para o ambiente da Internet das Coisas, tais como flexibilidade nas conexões, segurança, maior alcance e baixo consumo de energia.

2.3.4 Fontes de Energia

Dispositivos eletrônicos precisam de fontes de energia para mantê-los operando por um determinado tempo. Sendo os componentes que formam a base para a implementação dos conceitos de internet das coisas dispositivos eletrônicos pequenos, devido ao meio no qual serão inseridos, a fonte de energia por eles usadas também deve ser pequena. Devido a isso, são usadas baterias para energizá-los.

O problema é que, por serem tão pequenas, essas fontes de energia normalmente apresentam baixa eficiência. Por esse motivo, os microcontroladores podem ser programados para emitir sinais de alerta aos usuários quanto ao desempenho da bateria, assim é possível se programá-los para que não haja perdas na aquisição de dados. Uma outra opção seria poupar a energia das baterias aplicadas a sistemas de longa duração através da utilização da configuração *sleep* que alguns dispositivos oferecem, ou seja, programar os aparelhos para que em determinado período, no qual o sistema não esteja requisitando mais energia, operem com o mínimo de energia possível.

2.4 Protocolos da Camada de Aplicação

Existem diversos protocolos de aplicação que podem ser aplicados na IoT, dentre os mais usados estão: HTTP, CoAP e MQTT.

O HTTP (*Hypertext Transfer Protocol*) foi desenvolvido para rede de computadores, e sua comunicação baseia-se no princípio requisição/resposta. Por ter sido projetado para comunicação entre computadores, esse protocolo apresenta algumas restrições no que se diz respeito à Internet das Coisas, uma vez que os dispositivos usados na IoT apresentam baixo poder computacional (SANTOS, 2014).

O CoAP (*Constrained Application Protocol*) foi desenvolvido para aplicações ponto a ponto, destinado a dispositivos que possuem restrições de memória e energia. Este protocolo divide-se em duas camadas: em uma é realizado o mecanismo de requisição/resposta, enquanto na outra é detectada a duplicação das mensagens. Alguns dos seus benefícios são a confiabilidade e a fácil interação com o protocolo de rede HTTP (WAKA, 2015).

O MQTT (*Message Queue Telemetry Transport*) é um protocolo simples e leve, sendo destinado a dispositivos com alto grau de limitação. Seu funcionamento baseia-se no princípio *publish/subscribe* (Figura 2.7), ou seja, quando um dispositivo deseja obter alguma informação ele se inscreve a um *broker* (servidor virtual) fazendo uma requisição, então o *broker* “avisará” a este dispositivo quando a mensagem de interesse for publicada por outro dispositivo (SANTOS, 2014).

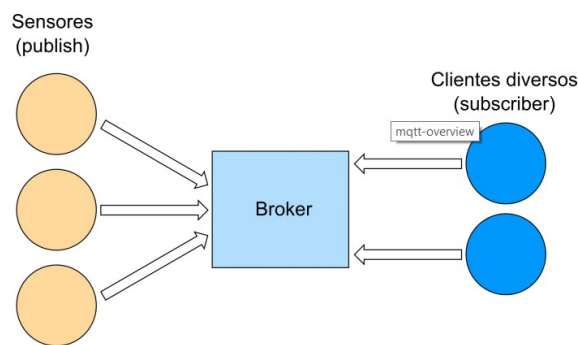


Figura 2.7 – Funcionamento Básico do Protocolo MQTT (BARROS, 2015).

2.5 Nuvem

De acordo com (TAURION, 2009), computação em nuvem é um termo para se descrever uma imensa rede de servidores, sejam eles físicos ou virtuais. Uma definição simples seria: um conjunto de recursos com capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na internet.

A computação em nuvem possibilita ao usuário acessar determinadas informações e realizar tarefas online sem a necessidade de instalar um aplicativo em seu computador. As principais características dos ambientes em nuvem são escalabilidade, elasticidade, tolerância a falhas e auto gerenciamento.

De forma geral, os ambientes disponibilizados em nuvem terão que apresentar uma capacidade de suportar um grande número de atualizações e análise de dados. Para a Internet das Coisas o objetivo da nuvem é agregar os dados recebidos dos mais diferentes sensores que possam haver em um determinado sistema. Em seguida, os dados devem ser armazenados por um longo período de tempo (RADOVICI, 2016).

2.6 Plataformas para IoT

Devido ao alto grau de heterogeneidade tanto de dispositivos quanto de protocolos que compõem os ambientes voltados para a IoT, como explicado na seção 2.2 deste trabalho, surgem no mercado inúmeras plataformas destinadas às aplicações do tema que visam promover a interoperabilidades entre os dispositivos usados.

Segundo (SANTOS, 2014) *“Tais plataformas são inseridas entre as aplicações e a infraestrutura (de comunicação, processamento e sensoriamento) subjacente, provendo um meio padronizado para o acesso aos dados e serviços fornecidos pelos objetos através de uma interface de alto nível”*. Porém até agora não há um padrão estabelecido para o desenvolvimento dessas plataformas, isso tem feito com que as mesmas apresentem diferentes modelos de programação que nem sempre são compatíveis.

Na Tabela 2.1, são apresentadas algumas das plataformas que mais se destacam atualmente e suas características básicas. Em sua maioria, a funcionalidade destas plataformas é baseada no modelo dos dados a serem coletados, uma vez que estes estão adequados ao sistema eles são armazenados.

Outros serviços significativamente importantes também são oferecidos por estas plataformas além do armazenamento, sendo eles: marcação de tempo dos dados recebidos, algum tipo de processamento, definição de regras para que uma atividade seja executada, entre outros (WAKA, 2015).

Tabela 2.1 – Plataformas para Aplicação IoT (SANTOS, 2014).

Plataforma	Endereço	Descrição	Tipo de Conta
Carriots	carriots.com	Plataforma com foco empresarial. Provê serviços de gerenciamento e comunicação.	Conta gratuita, mas com limitações.
Microsoft Azure IoT	microsoft.com/iot	Plataforma voltada para IoT com foco empresarial.	Conta gratuita por um mês.
ThingSpeak	thingspeak.com	Plataforma robusta, com várias funcionalidades, como sensores públicos e busca por histórico.	Conta gratuita.
Cayenne	cayenne-mydevices.com	Plataforma voltada ao controle de dispositivos e integração via celular.	Conta gratuita.
Mosquitto	mosquitto.org	Plataforma em nuvem de fácil implementação.	Gratuita com biblioteca para C++.

2.7 Aplicações: A Internet das Coisas e a Indústria

Depois das três revoluções industriais que marcaram o desenvolvimento da sociedade humana, alguns especialistas acreditam estarmos diante da mais nova revolução industrial, também chamada de Revolução 4.0.

Essa revolução nada mais é que a aplicação da Internet das Coisas nos processos industriais. Nessa nova fase industrial que abrange áreas como manufatura, automotivo, transporte, produção de energia, entre outros, o ambiente industrial passa a ter dispositivos inteligentes capazes de se comunicar entre eles e com as pessoas para melhor desenvolvimento do setor.

Ainda que a ideia central sobre as aplicações da IoT que norteia a maioria da população sejam de cunho pessoal, limitado à rotinas diárias, esta tecnologia definitivamente vai além desse limite. As possibilidades de aplicações, apesar de limitadas quanto ao tamanho e alcance dos dispositivos a serem usados, são incontáveis (GHAREGOZLOU, 2016).

Assim como numa casa, que possui sistema de controle do ambiente, é possível verificar temperatura, umidade e iluminação por meio de sensores, em um automóvel também é possível usar a IoT para coletar dados relevantes, e atuar sobre o objeto. Atualmente, por exemplo, já existem carros que são capazes de circular sem um humano operando, e é capaz de tomar ações corretivas baseadas nos dados obtidos do meio ao qual está inserido, tal como parar em um sinal, trocar de faixa, identificar obstáculos, tudo isso baseado em sensores conectados à rede de computadores.

Na medicina, a prevenção de doenças deve ser o foco principal nos cuidados com a saúde, e a Internet das Coisas possibilita um melhor acompanhamento de pacientes para que estes não cheguem a sofrer grandes traumas. Como exposto em (PRESSER, 2012), com a Internet das Coisas é possível acompanhar o desempenho do paciente diariamente através de micro dispositivos acoplados ao seu corpo, e os dados coletados serem enviados em tempo real para uma central onde o médico possa avaliar. O próprio sistema pode ser programado para identificar no organismo do paciente padrões fora da normalidade, e assim acionar o socorro caso seja necessário.

Na agricultura, conceitos da Internet das Coisas já são usados para monitorar as plantações 24 horas por dia e obter um melhor controle das mudanças climáticas. Já é possível em laboratório, por exemplo, máquinas realizarem o plantio e acompanhamento de plantações de maneira autônoma. Para tal, elas são conectadas à internet e, mediante cruzamento de dados, são capazes de tomar decisões.

Ainda de acordo com (PRESSER, 2012), na área da energia o desenvolvimento da Internet das Coisas também vem sendo bastante explorado. Podendo oferecer soluções

para alguns dos problemas mais recorrentes, tais como a busca pela redução do consumo de energia. A iluminação pública é um exemplo disso, a Internet das Coisas pode ser uma alternativa para a redução do consumo de energia neste setor, pois além de ligar e desligar as lâmpadas, é possível através dessa tecnologia identificar a necessidade ou não de determinado objeto estar ligado.

Um outro exemplo da aplicação da Internet das Coisas no setor de energia é com relação aos painéis fotovoltaicos. Atualmente, o desempenho das placas solares instaladas pode ser acompanhado através de dispositivos, tal como o celular, em tempo real. Este monitoramento é possível devido à inserção de dispositivos da IoT em sua estrutura.

2.8 Os Desafios da Internet das Coisas

À medida que o número de aplicações para a Internet das Coisas aumenta, os desafios a serem enfrentados por ela também crescem. Sendo esta nova tecnologia voltada para a conexão e comunicação dos mais diversos objetos, nas mais diferentes áreas e implementados de formas diferentes, para finalidades diferentes, ela possui um ambiente altamente heterogêneo, difícil de ser interligado. Desta forma, a Internet das coisas apresenta uma série de desafios a serem enfrentados antes de conseguir se difundir completamente.

A definição de um padrão para o desenvolvimento de ambiente IoT é um dos obstáculos a serem superados. Apesar do alto interesse nessa nova era da internet, muitas empresas de tecnologia têm se fechado e buscado montar seu próprio ecossistema IoT, cada uma com seus próprios protocolos, tecnologias e padrão. Esse desenvolvimento particularizado inviabiliza o avanço da Internet das Coisas, sendo fundamental que as empresas de tecnologia entrem em acordo, e consequentemente sigam um padrão no qual todo e qualquer objeto seja capaz de interagir com o meio, como o IoT propõe.

Um outro problema a ser enfrentado é com a segurança em torno da Internet das Coisas, tanto em relação às pessoas, quanto às informações por ela transmitida. Afinal, a partir do momento em que tudo está conectado, existe a dúvida quanto à propriedade dos dados. Afinal, no momento em que tudo esteja ligado à internet, a grande questão é se ainda haveria privacidade (SANTOS, 2014).

2.9 Impactos e Expectativas da Internet das Coisas

Grandes empresas têm mostrado interesse no desenvolvimento dessa nova tecnologia que é a Internet das Coisas. A Huawei, uma empresa de telecomunicações chinesa, por exemplo, tem colocado a Internet das Coisas como prioridade na empresa, e tem atingido grandes avanços em pesquisas. Desde o desenvolvimento de *gateways* mais eficientes à

plataforma de gestão de conexão, a Huawei tem sido pioneira. Outras grandes empresas, como Google e Apple, também estão investindo na área.

Diante de tamanho interesse e popularização da IoT, questiona-se como que as empresas de modo geral irão se beneficiar dessa evolução tecnológica. O foco é transformar dados físicos em digitais, e esse provavelmente terá um impacto direto no setor econômico das empresas. Com dados digitais, a empresa poderá ter um controle maior com relação às perdas desde a produção quanto à manutenção de seus produtos. Por exemplo, uma concessionária de energia é capaz de gerenciar todo o seu processo de geração e distribuição de eletricidade, mas com a Internet das Coisas ela será capaz de prever com maior precisão problemas apresentados pelas redes, e medidas corretivas pouparão investimentos frequentes e desnecessários (MANZANO, 2016).

Contudo, não é muito difícil concluir que com o desenvolvimento da IoT oportunidades de novas carreiras também surgirão. Ainda não é possível afirmar quais tipos de profissões podem surgir, devido à vasta possibilidade de aplicação, mas é certo que em algum momento a IoT proporcionará o surgimento de novos profissionais.

O que observa-se é que a Internet das Coisas é uma revolução tecnológica capaz de causar transformação em todos os setores que cercam a humanidade. É uma tecnologia, que assim como aconteceu com a própria internet já difundida, irá impactar a sociedade moderna como um todo (IDG, 2016).

A expectativa é que a Internet das Coisas cresça consideravelmente nos próximos anos. Espera-se que a IoT se torne uma rede com bilhões e até trilhões de objetos, pessoas e/ou processos conectados nos próximos anos. Uma pesquisa realizada pela Cisco, por exemplo, uma das maiores empresas de tecnologia do mundo, afirma que até 2020 serão 30 bilhões de objetos conectados através da Internet das Coisas, e serão 1,7 trilhões de dólares sendo movimentados direta ou indiretamente por essa tecnologia. Neste mesmo ano, espera-se que a Internet das Coisas represente 6% da economia global (IEEE, 2016).

O número de investimentos e pesquisas aumentarão ao passo que novos benefícios e novas descobertas acerca da IoT sejam expostos ao mundo. Com isso, a expectativa é que em algumas décadas a maioria dos objetos e sistemas que hoje usamos sejam compatíveis com as propostas da Internet das Coisas. Isso permitirá que a Internet das Coisas se integre ao dia a dia das pessoas de maneira imperceptível.

Capítulo 3

Estudo de caso: Sistema para Controle de Iluminação e Temperatura via Internet

Neste capítulo é apresentado o estudo de caso proposto como aplicação dos conceitos de Internet das Coisas que foram até então explorados. Inicialmente é descrito todo o funcionamento do sistema e seus propósitos, e em seguida é feita uma descrição da metodologia utilizada para a implementação dos componentes de *hardwares* e os *softwares* necessários ao desenvolvimento do mesmo.

3.1 Descrição do Sistema

O objetivo do projeto é implementar um sistema para controlar a iluminação e temperatura de um ambiente via Internet. Na Figura 3.1 é apresentado o diagrama em blocos do sistema.

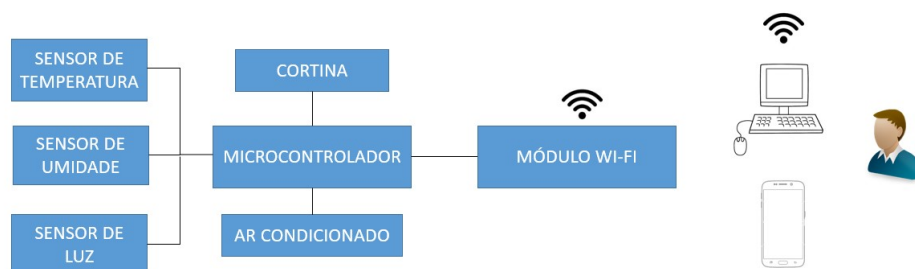


Figura 3.1 – Diagrama de blocos do sistema desenvolvido (Próprio Autor).

Como pode ser observado no diagrama, o projeto atua sobre dois objetos, sendo

eles uma cortina motorizada e um ar condicionado. Esses objetos são acoplados a um microcontrolador que estará recebendo e processando as informações coletadas pelos sensores.

A função do microcontrolador é garantir que a temperatura e iluminação do ambiente permaneçam dentro dos parâmetros estabelecidos. Inicialmente, esses parâmetros são configurados com base nos valores de temperatura e luminosidade que forem considerados ideais para o ambiente. Sendo assim, caso os dados coletados pelos sensores estejam fora desse padrão, o microcontrolador atuará sobre a cortina e/ou o ar condicionado até que os mesmos se ajustem aos valores esperados.

Por outro lado, também é disponibilizado aos usuários a possibilidade de alterar tais parâmetros de acordo com suas necessidades. Para isso, as informações dos sensores que são transmitidas ao microcontrolador também são enviadas para uma plataforma online, onde estarão disponíveis para o usuário monitorar e controlar remotamente o ambiente em questão. O envio dos dados para a plataforma é feito através da unidade de comunicação por meio de um protocolo Wi-Fi.

Utilizando os comandos presentes na plataforma, tais como aumentar/diminuir temperatura, ligar/desligar ar condicionado e ajustar a iluminação, o usuário será capaz de atuar sobre o sistema independentemente dos parâmetros iniciais. Caso não haja nenhuma modificação por parte do usuário, o sistema irá seguir os *setpoints* previamente estabelecidos.

Contudo, o objetivo do projeto é coletar dados dos sensores (temperatura, umidade e luminosidade) para que o sistema seja capaz de ajustar a temperatura do ar condicionado e/ou a iluminação do ambiente, de acordo com os níveis de iluminância e temperatura configurados no sistema. Além de possibilitar ao usuário acompanhar e atuar sobre o ambiente monitorado via Internet.

3.2 Definição da Plataforma do Sistema

Na Seção 2.6 foram apresentadas cinco diferentes plataformas que têm se destacado nas aplicações da Internet das Coisas. Dentre as cinco citadas, apenas três foram objeto de estudo deste TCC, com o objetivo de se determinar qual delas seria a mais adequada para o desenvolvimento do projeto. As plataformas em questão são: Mosquitto, ThingSpeak e Cayenne.

A plataforma Mosquitto é basicamente um *broker* que utiliza o protocolo MQTT para realizar comunicação com os dispositivos que estejam nele conectados. Esse protocolo foi baseado na comunicação TCP/IP, o que facilita a interação entre os dois protocolos em questão. Essa característica é importante para os casos em que pretende-se conectar

vários sistemas, com protocolos diferentes, em uma única rede (MOSQUITTO, 2016).

Alguns aspectos importantes deste *broker* é a utilização da linguagem C++ em sua programação e a capacidade de até 3MB de memória RAM. Por outro lado, a desvantagem da Mosquitto é que através dela não é possível atuar sobre o sistema desejado. Por se basear no método de subscrever e publicar, sua função é apenas monitorar os dados dos dispositivos nela conectados.

Em contrapartida, a Thingspeak é uma plataforma voltada para as aplicações da IoT, que disponibiliza dados coletados pelos sensores através de canais de comunicação. Tais dados são armazenados e disponibilizados na forma de gráficos. A configuração desses canais depende da origem dos dados que deseja-se monitorar, e cada canal pode conter no máximo 8 sensores. A plataforma também apresenta aplicativos que permitem realizar a análise e visualização dos dados através do *software* Matlab (THINGSPEAK, 2016).

Alguns dos serviços disponibilizados aos usuário por esta plataforma são: coleta e análise dos dados que foram passados pelos sensores, envio de alertas ao usuário e a possibilidade de agendar um evento para ser executado num futuro pré definido.

Por outro lado, de acordo com (HACKSTER, 2016), a plataforma Cayenne também é um desenvolvedor de projetos IoT que permite os usuários criarem sistemas nesse ambiente computacional, e gerenciá-los em tempo real. Essa plataforma disponibiliza uma série de dispositivos eletrônicos virtuais, pré configurados, que permitem os usuários controlar remotamente componentes eletrônicos reais, tais como sensores, atuadores, microcontroladores, entre outros.

Os dados gerados por estes componentes são disponibilizados na Cayenne através de "janelas", e cada uma dessas deve ser associada ao dispositivo cuja irá prestar serviço. Essas janelas apresentam uma pré configuração, que pode ser facilmente formatada pelos usuários. É possível por exemplo disponibilizar os dados tanto em forma de gráficos quanto por meio do valor atual apresentado no sistema.

Assim como na Thingspeak, a plataforma Cayenne também disponibiliza serviços para o agendamento de ações que deseja-se executar automaticamente em um determinado espaço de tempo. Essas ações podem ser tanto o envio de um sinal de alerta ao usuário, ou seja, enviar por email uma notificação de que algo está errado, baseado nos parâmetros configurados, quanto a definição do acionamento de um dispositivo que está sendo controlado pela plataforma, podendo este acionamento ocorrer periodicamente ou no dia e hora configurados (CAYENNE, 2016).

Contudo, como um dos objetivos do projeto é permitir que o sistema não só seja monitorado, mas também controlado pelo usuário via Internet, a plataforma de comunicação Cayenne mostrou-se mais apropriada para o desenvolvimento do sistema, sendo esta a plataforma usada neste projeto.

3.3 Metodologia do *Hardware*

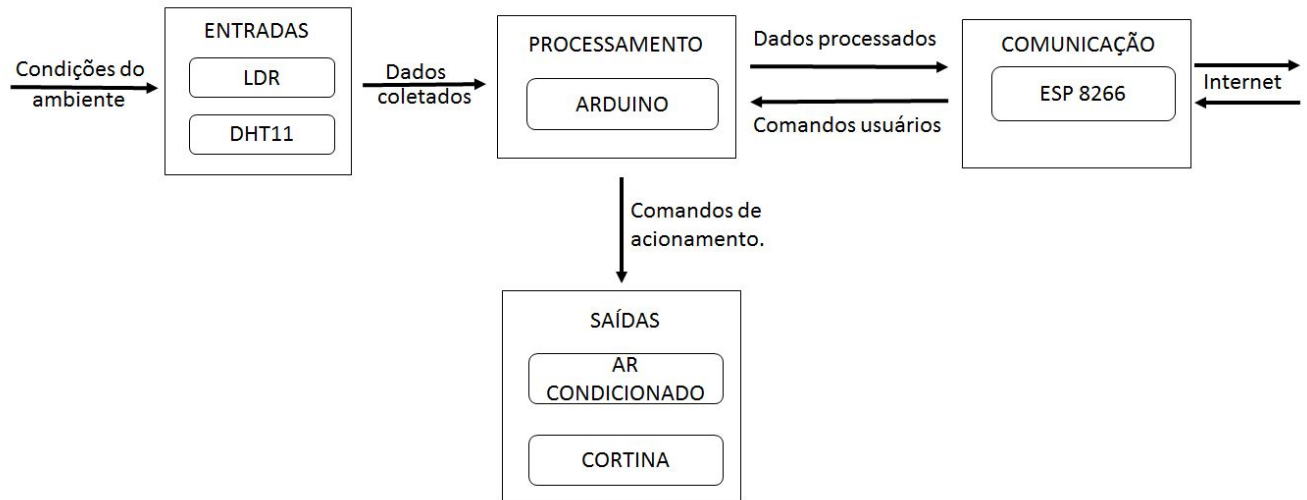


Figura 3.2 – Diagrama de blocos do *Hardware* do Sistema (Próprio Autor).

No diagrama da Figura 3.2, o sistema é dividido em função das entradas, do processamento, das saídas e da unidade de comunicação. Os sensores estão inseridos no bloco de entrada do sistema, e é neste bloco que as condições de temperatura, iluminação e umidade do ambiente são adquiridas e enviadas para o Arduino. No bloco do processamento, essas variáveis referentes ao ambiente são comparadas com os parâmetros do sistema, a fim de corrigi-los caso haja distorção.

O microcontrolador também é responsável pelo acionamento dos objetos, ou seja, sempre que as características do ambiente não estão de acordo com o esperado, comandos são enviados do Arduino para os objetos a fim de que haja a correção necessária para se adequar aos *setpoints*. Já o módulo Wi-Fi é a unidade de comunicação responsável pelo envio e recebimento de informações entre o microcontrolador e a plataforma da Internet utilizada.

Nos tópicos a seguir será detalhada a implementação de cada um dos blocos do diagrama da Figura 3.2.

3.3.1 Implementação dos circuitos de entrada do sistema.

Na entrada do sistema foram utilizados dois sensores, sendo eles: o LDR, usado para monitorar a luminosidade, e o DHT11, responsável pela coleta dos dados de temperatura e umidade.

3.3.1.1 LDR

O LDR (*Light Dependent Resistor*) é um sensor analógico cuja resistência varia a partir da luminosidade que é incidida sobre ele. De acordo com (PATSKO, 2014), esses sensores são compostos por sulfeto de cádmio, um semicondutor capaz de diminuir a sua resistência à passagem de corrente mediante a uma maior incidência luminosa. Ou seja, quanto mais intensa a luz sobre o sensor, menor será a sua resistência.

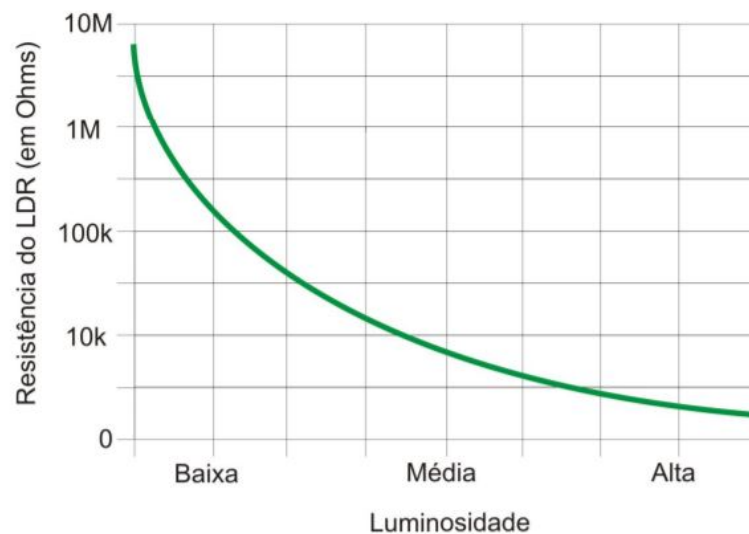


Figura 3.3 – Gráfico da Resposta do LDR (PATSKO, 2014).

Na Figura 3.3 é apresentada a curva de resposta do sensor. Nesta curva é possível verificar o comportamento quase exponencial da relação entre a resistência do sensor e a luminosidade incidida sobre ele.

Algumas das principais características elétricas do LDR estão na Tabela 3.1. Com base nessas informações, pode-se observar que na ausência total de luz o LDR apresentará uma resistência de 1M Ohms, e em contrapartida, ao aumentar-se a luminosidade a níveis consideravelmente elevados, da ordem de 1000 lux por exemplo, essa resistência diminuirá para aproximadamente 400 Ohms.

Tabela 3.1 – Principais características do LDR (SUNROM, 2008).

Parâmetro	Condições	Min	Typ.	Max	Unidades
Resistência	1000 lux	—	400	—	Ohms
	10 lux	—	9	—	Kilohms
Resistência no escuro	—	1.0	—	—	Megaohms
Capacitância no escuro	—	—	3.5	—	pF

Baseado nas características apresentadas, uma forma simples de identificar a variação da luminosidade de um ambiente por meio do sensor LDR é através de um circuito

divisor de tensão (Figura 3.4), ou seja, conectar dois resistores em série a uma fonte. Desta forma, é possível utilizar os valores das resistências em conjunto da Equação 3.1 para determinar a tensão de saída entre esses resistores.

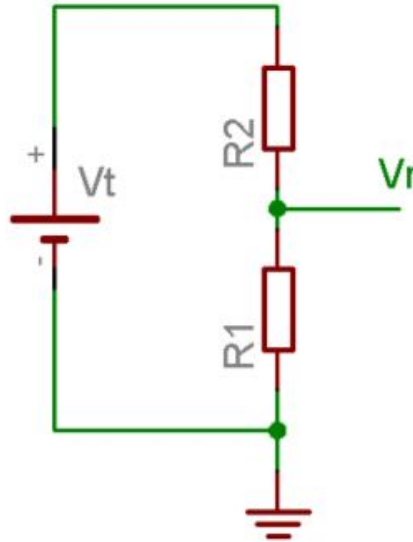


Figura 3.4 – Divisor de Tensão (PATSKO, 2014).

$$V_r = \frac{V_t R_1}{R_1 + R_2} \quad (3.1)$$

Como o LDR também é uma resistência, basta substituir um dos resistores deste divisor de tensão pelo sensor, e assim a saída do circuito será dependente da incidência de luz sobre o LDR. A fim de se obter uma relação direta entre a intensidade da luz sobre o sensor e a tensão de saída (V_r) do sistema, a substituição da resistência deve ser realizada em R_2 . Sendo assim, ao aumentar-se a intensidade da luz, a resistência do sensor irá diminuir e a tensão de saída aumentará. Logo, quanto maior a intensidade luminosa maior será a saída do circuito, e vice-versa.

Para o desenvolvimento deste projeto, a tensão aplicada na entrada do divisor foi de 5 volts. Além disso, o resistor R_1 foi substituído por uma resistência com valor entre o máximo e mínimo apresentados pelo LDR. Isso porque o valor deste resistor irá influenciar na interpretação da tensão captada pelo microcontrolador. Sendo assim, para facilitar a tradução da informação passada do circuito para o Arduino, substitui-se R_1 por uma resistência no valor de 10KOhms (Ver Figura 3.4).

Sendo assim, enquanto a iluminação sobre o sensor for intensa, o Arduino identificará aproximadamente a tensão de 5V da fonte em seu pino de entrada. À medida que a intensidade da luz diminuir, o valor observado pelo Arduino também reduzirá, chegando a ser aproximadamente metade do valor da tensão de entrada, uma vez que o sensor LDR

e o resistor R1 terão aproximadamente os mesmos valores. Logo, é possível a partir da tensão de entrada no pino do Arduino identificar a variação da luminosidade no ambiente.

Na Figura 3.5 está a representação do circuito que foi montado para a leitura do LDR.

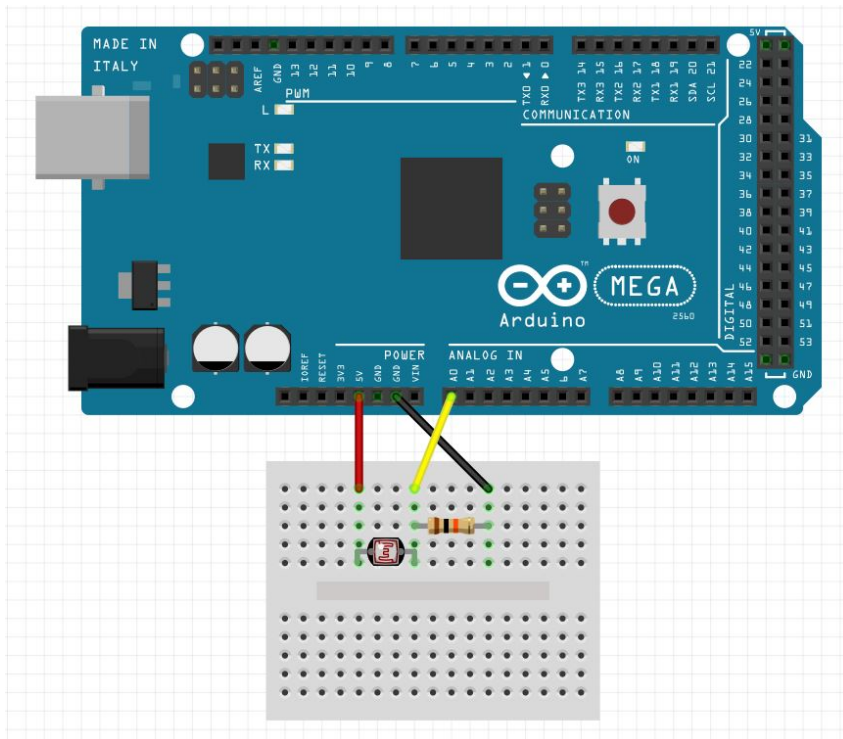


Figura 3.5 – Circuito para aquisição dos dados do sensor LDR (Próprio Autor).

3.3.1.2 DHT11

O sensor de temperatura do DHT11 é formado por um termistor NTC, dispositivo elétrico que possui uma resistência variável cujos valores são alterados de acordo com a temperatura sob o qual é submetido (AOSONG, 2016).

Este sensor é capaz de realizar medições de temperatura entre 0 a 50 graus Celsius, e de umidade relativa entre 20% a 90%, simultaneamente. A temperatura tem uma faixa de precisão de 2 graus, enquanto a precisão da umidade é de 5%. Este sensor pode ser considerado relativamente lento, visto que apresenta um tempo de resposta entre as medições de aproximadamente 2s (AOSONG, 2016).

Para o sistema implementado neste projeto, o DHT11 é totalmente compatível, uma vez que para o controlar o ar condicionado não se exige altas taxas de velocidade nas mudanças dos parâmetros. Além disso, o baixo custo e a facilidade de acesso foram outros fatores considerados na escolha deste componente.

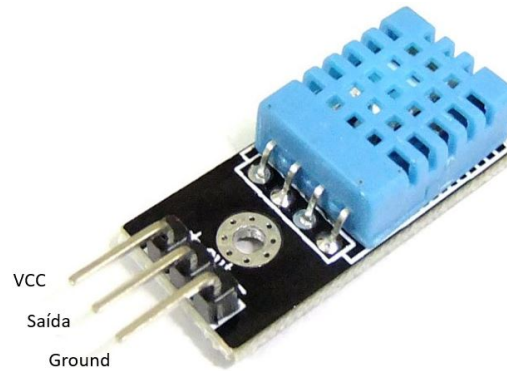


Figura 3.6 – Identificação dos pinos do sensor DHT11 (AOSONG, 2016).

Existem diferentes sensores DHT11, tomando como exemplo o modelo apresentado na Figura 3.6, tem-se que VCC é o pino de alimentação, *Ground* é a referência e Saída é o pino de leitura do sensor.

Algumas das principais características elétricas deste sensor são:

- Alimentação: 3,0 a 5,0 VCC (5,5 VCC máximo);
- Corrente: 200uA a 500mA, em *stand by* de 100uA a 150 uA;
- Tempo de amostragem maior que 2s.

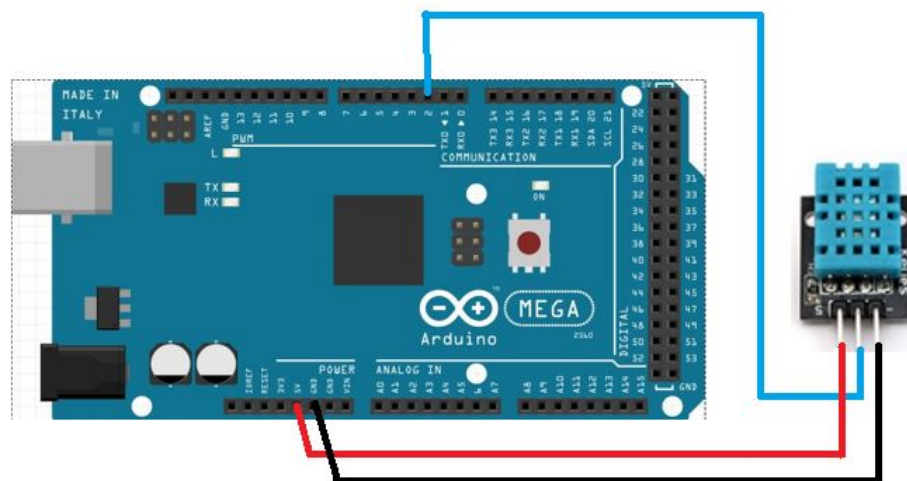


Figura 3.7 – Circuito para a aquisição dos dados do sensor DHT11 (Próprio Autor).

A montagem do DHT11 com o arduino é bastante simples, não sendo necessário o uso de nenhum outro componente no circuito, como pode ser verificado na Figura 3.7. O VCC é alimentado com 5V, o Ground é conectado à referência do sistema e a saída é ligada a algum dos pinos digitais do Arduino.

3.3.2 Unidade de processamento

A unidade de processamento é a parte central do sistema. Através desta, os dados dos sensores serão coletados, processados, armazenados e repassados para o módulo de comunicação e, quando necessário, para as saídas do sistema como explicado na seção 2.3.2 deste trabalho.

Para o desenvolvimento deste projeto buscou-se utilizar uma plataforma de processamento de fácil compreensão, versátil e amplamente acessível, com o objetivo de facilitar a aplicação dos conceitos da Internet das Coisas em possíveis propostas futuras. Logo, o *hardware* escolhido para o processamento foi o Arduino.

O Arduino é uma placa microcontroladora capaz de se conectar a um computador através de uma porta serial, possibilitando usuários desenvolverem e testarem seus projetos sem a exigência de grandes investimentos. Existem diferentes versões disponíveis, e que diferem entre si quanto ao número de entradas analógicas e digitais, processador, memória SRAM, memória *Flash*, entre outras características.

Esta placa pode ser energizada tanto pela porta serial, quanto por fontes de alimentação externas, desde que limitadas a uma faixa de tensão entre 6 e 20 volts. Além disso, seus pinos de saída também disponibilizam uma tensão de 5 volts fixas aos circuitos neles conectados.

A priori, a versão da placa escolhida para o desenvolvimento deste projeto foi a UNO. Este dispositivo foi o primeiro de uma série de placas USB desenvolvidas para a plataforma Arduino. Ela disponibiliza 14 entradas/saídas digitais e 6 analógicas, taxa de processamento de 16MHz e uma porta de comunicação serial. Seu microcontrolador é o ATmega328P, que possui 32 KB de memória *Flash* e 2KB de memória RAM (ARDUINO, 2016b).

Durante o desenvolvimento do projeto, foi constatada a necessidade de uma memória RAM maior, devido aos sinais infravermelhos responsáveis pelos comandos do ar condicionado usado neste projeto. Estes sinais correspondem a vetores de aproximadamente 700 *bytes* cada, e como precisa-se de um sinal para cada função executada, a memória RAM do Arduino UNO (2K *bytes*) foi excedida. Logo, buscou-se substituir por outra placa que atendesse aos requisitos de memória necessário, tal como o Arduino MEGA.

Como pode ser observado na Figura 3.8, essa placa possui 54 entradas/saídas digitais e 16 entradas analógicas, permitindo que pequenos componentes eletro/eletrônicos externos sejam conectados à mesma.

O microcontrolador presente no Arduino Mega, e responsável por todo o processamento dos dados, é o ATmega2560. Este microcontrolador apresenta uma memória

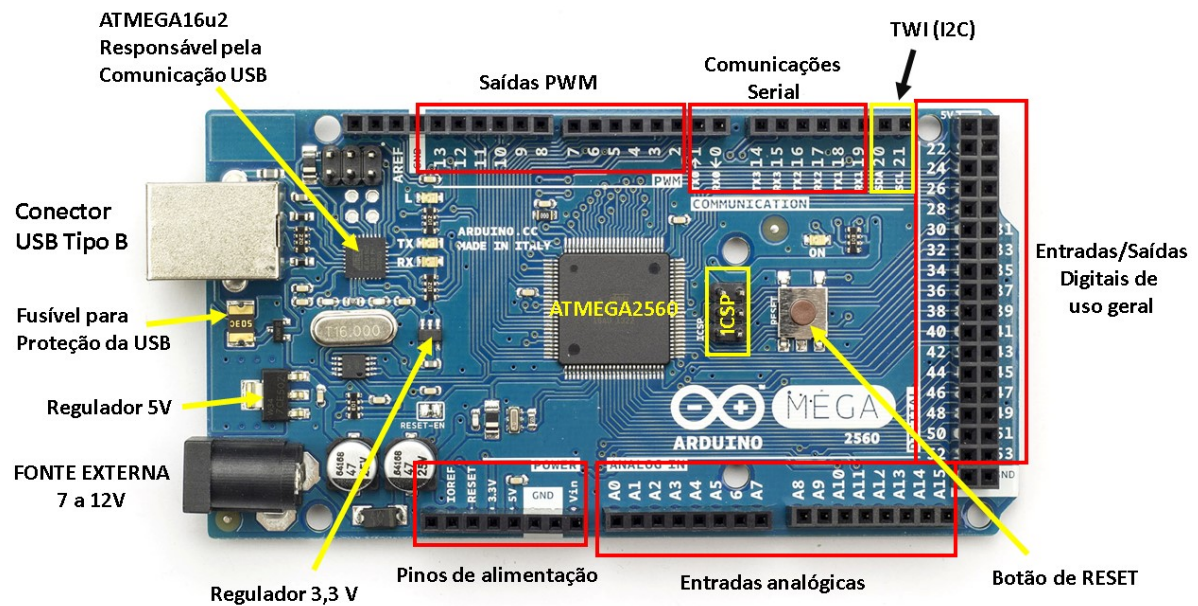


Figura 3.8 – Identificação das entradas/saídas e dos dispositivos presentes na placa Arduino Mega (SOUZA, 2014).

SRAM de 8KB, que foi determinante para a escolha desta placa no projeto, além de apresentar 256 KB de memória *Flash*, 4 portas para comunicação serial e uma taxa de processamento de 16 MHz.

Algumas das suas principais características são:

- Voltagem de operação: 5V;
- Alimentação de entrada recomendada: 7-12V;
- Alimentação de entrada limite: 6-20V;
- Corrente DC dos pines I/O: 20mA;
- Corrente DC para o pino de 3.3V: 50mA;
- Memória *flash*: 256KB;
- Memória SRAM: 8KB;
- EEPROM: 4 KB;

- Velocidade do *clock*: 16 MHz.

Na Tabela 3.2 é exposta uma comparação entre as duas placas microcontroladora em questão, evidenciando a superioridade do Arduino Mega.

Tabela 3.2 – Comparação entre as características do Arduino UNO e do Arduino Mega (ARDUINO, 2016a).

Nome	Processador	Velocidade de processamento	EEPROM (KB)	SRAM (KB)	Flash (KB)	UART
Mega 2560	ATmega2560	16MHz	4	8	256	4
Uno	ATmega328	16MHz	1	2	32	1

3.3.3 Implementação dos circuitos de saída do sistema.

Como visto no diagrama da Figura 3.2, o sistema apresenta duas saídas, sendo elas um ar condicionado, que é ativado através de um emissor infravermelho, e uma cortina, que é acionada por meio de um motor CC (Corrente Contínua) acoplado à sua estrutura. O acionamento de cada uma das saídas dependerá da configuração dos parâmetros que são enviados ao microcontrolador.

3.3.3.1 Ar Condicionado

O controle do ar condicionado é realizado remotamente através do envio de sinais por meio de um emissor infravermelho. Para operar desta forma, foi necessário inicialmente decodificar os sinais emitidos pelo controle remoto da máquina a ser usada. Após a decodificação, foi montado um circuito para realizar o envio destes sinais até o receptor do aparelho.

Para realizar a primeira parte do processo, ou seja, para decodificar os sinais, utilizou-se um receptor infravermelho chamado TSOP 1737, cuja função é identificar sinais infravermelhos. Na Figura 3.9 está a representação deste componente, juntamente com a identificação de seus pinos.

De acordo com (VISHAY, 2016), o TSOP 1737 possui as seguintes características:

- Receptor infravermelho e pré-amplificador;
- Filtro interno para frequência PCM;

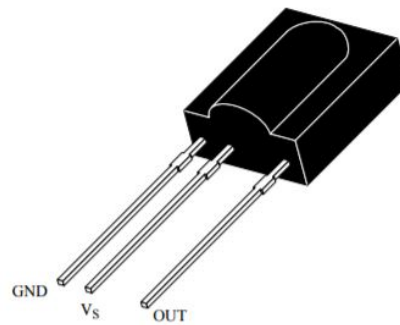


Figura 3.9 – Identificação dos pinos do receptor infravermelho TSOP1737 (VISHAY, 2016).

- Melhor proteção contra distúrbios de campo elétrico;
- Saída ativa baixa;
- Baixo consumo de energia;
- Possível transmissão contínua de dados(1200bit/s);

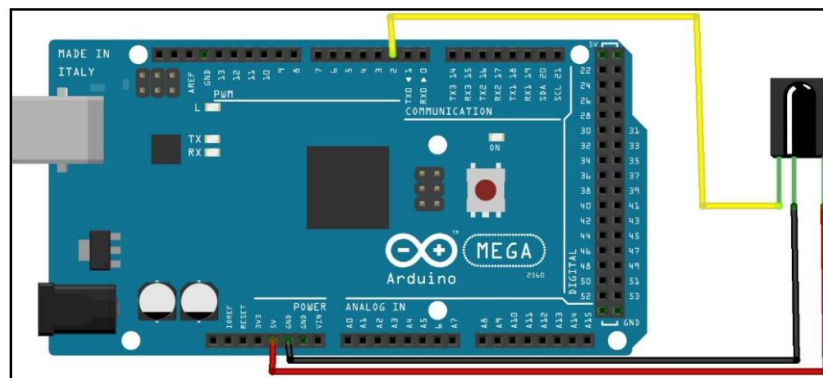


Figura 3.10 – Circuito para decodificar os sinais de controle do ar condicionado (Próprio Autor).

Neste projeto, os comandos usados para o controle do ar condicionado são: ligar, desligar, aumentar e diminuir temperatura. Para obter essas informações, um circuito como o da Figura 3.10 foi utilizado, e juntamente com o controle remoto do ar condicionado que se deseja controlar e o compilador do Arduino, foi possível identificar o protocolo de cada uma das funções desejadas. Para realizar os comandos de aumentar e diminuir a temperatura, foi necessário decodificar os sinais correspondentes ao valor de cada temperatura considerada no sistema.

Ou seja, apertando-se os botões do controle original do ar condicionado usado, neste caso o controle remoto Utopia da Hitachi (Figura 3.11), em direção o circuito, foi

possível realizar a leitura de cada um dos sinais, e baseada na programação aplicada, identifica-los e armazena-los.



Figura 3.11 – Controle remoto usado na decodificação dos sinais de comando do ar condicionado (Próprio Autor).

Durante este processo, verificou-se que os vetores correspondentes aos sinais gerados pelo controle remoto do ar condicionado Hitachi, marca do ar condicionado usado para executar este projeto, eram da ordem de aproximadamente 700 *bytes* cada, observar a Figura 3.12. Esse fato inviabilizava o envio dos sinais utilizando a placa microcontroladora Arduino UNO, sendo necessário o uso de uma placa com memória RAM maior, conforme explicado na seção 3.3.2.

De posse dos sinais de comando, o passo seguinte foi enviar os sinais ao receptor do próprio ar condicionado, permitindo que o mesmo fosse controlado através do Arduino. Para esta parte do projeto, buscou-se representar da forma mais fiel possível o circuito referente ao controle da Figura 3.11. Desta forma, foram retirados alguns componentes de um controle em desuso, e após observar o esquemático elétrico deste, o circuito mostrado na Figura 3.13 foi implementado.

Neste *layout*, tem-se um transistor NPN, um LED emissor de infravermelho e dois resistores, sendo um de 10 Ohms e outro de 220 Ohms.

Uma tensão de 5 volts é diretamente aplicada a um resistor de 10 Ohms que está

Desligar:

{3392, 1668, 480, 1224, 508, 384, 508, 360, 508, 356, 508, 384, 480, 384, 480, 388, 500, 388, 480, 384, 480, 388, 504, 388, 532, 332, 480, 1224, 480, 412, 508, 356, 480, 388, 504, 388, 532, 332, 480, 384, 476, 416, 480, 384, 508, 360, 476, 416, 504, 360, 532, 332, 508, 384, 480, 384, 508, 360, 532, 360, 532, 332, 504, 1200, 500, 392, 476, 1228, 532, 1172, 504, 1224, 480, 1224, 508, 1196, 508, 1224, 508, 356, 480, 1224, 508, 1224, 480, 1224, 508, 1196, 508, 1224, 504, 1200, 480, 1224, 480, 1248, 508, 1200, 504, 380, 488, 384, 476, 388, 508, 356, 508, 384, 508, 360, 504, 360, 504, 408, 460, 384, 480, 1224, 508, 1224, 504, 360, 508, 356, 508, 1224, 476, 1228, 508, 1200, 504, 1224, 504, 360, 508, 356, 536, 1196, 480, 1224, 508, 356, 508, 384, 480, 388, 504, 1200, 476, 416, 504, 360, 504, 1200, 536, 356, 504, 360, 480, 384, 504, 1228, 508, 356, 480, 1224, 508, 1224, 476, 388, 480, 1224, 508, 1224, 480, 1224, 508, 356, 480, 1252, 476, 388, 508, 356, 480, 412, 508, 360, 476, 388, 476, 416, 536, 328, 480, 384, 536, 1196, 508, 1196, 504, 1200, 508, 1224, 480, 1224, 480, 1224, 504, 1228, 476, 1224, 484, 1220, 508, 1224, 508, 1196, 536, 1168, 508, 1224, 508, 356, 508, 360, 504, 384, 480, 388, 480, 384, 508, 408, 484, 356, 508, 360, 480, 1248, 508, 1196, 508, 1196, 508, 1224, 508, 380, 484, 1196, 536, 1196, 508, 380, 484, 380, 512, 380, 488, 380, 484, 380, 484, 1224, 508, 380, 484, 360, 508, 1220, 536, 1168, 536, 1168, 508, 384, 508, 1196, 508, 1196, 536, 356, 536, 328, 508, 360, 508, 1220, 536, 332, 532, 1168, 536, 360, 504, 360, 536, 1168, 536, 1192, 536, 1168, 512, 356, 532, 1196, 508, 360, 532, 332, 536, 356, 536, 328, 508, 360, 532, 356, 508, 360, 536, 328, 508, 1220, 512, 1192, 540, 1164, 540, 1192, 536, 1168, 536, 1168, 512, 1220, 508, 1196, 536, 328, 512, 384, 532, 332, 532, 332, 564, 328, 508, 1196, 508, 356, 536, 1192, 540, 1168, 508, 1196, 508, 1220, 536, 1168, 536, 1168, 536, 360, 508, 1192, 536, 332, 536}

Figura 3.12 – Vetor correspondente a sinal do comando desligar do ar condicionado Hitachi (Próprio Autor).

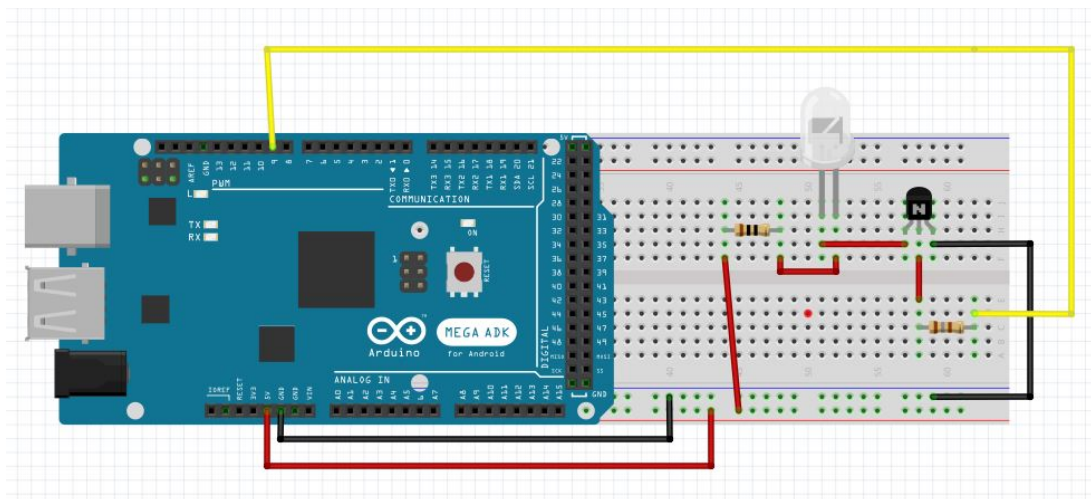


Figura 3.13 – Esquema do circuito para emissão dos dados através de um LED emissor de infravermelho (Próprio Autor).

ligado a um dos pinos do LED emissor. O outro pino do LED é conectado ao coletor do transistor, que tem seu pino emissor ligado à referência do circuito e sua base é conectada ao resistor de 220 Ohms que esta diretamente ligado a uma das entradas digitais do Arduino.

Após a montagem do circuito, foram realizados testes para verificar a distância a

qual o LED era capaz de acionar o ar condicionado. Durante esses testes, percebeu-se que quanto menor o valor do resistor conectado ao LED, maior é o alcance do sinal, devido ao valor da corrente que é fornecida. Apesar de no circuito original do controle, o valor dessa resistência usado ser de 1,8 Ohms, usou-se no circuito um resistor de 10 Ohms, que foi suficiente para acionar o aparelho à distância.

3.3.3.2 Cortina

Para a simulação da cortina, a qual é responsável pela variação da incidência luminosa sobre o ambiente, foi construído um protótipo, conforme mostrado na Figura 3.14.



Figura 3.14 – Protótipo da cortina motorizada (Próprio Autor).

Um parafuso de eixo sem fim foi usado para sustentar o peso da cortina, sendo fixado na estrutura de madeira que compõe o protótipo. Para realizar o movimento de abrir e fechar, um fio de *nylon* foi acoplado a um motor de corrente contínua e duas bobinas. Sendo assim, ao ligar o motor numa direção, uma das bobinas irá enrolar o fio enquanto a outra desenrola.

Devido a alta velocidade do motor e o curto espaço para deslocamento da cortina, polias foram usadas para reduzir a velocidade com a qual a cortina se desloca, com isso também aumentou-se o torque.

Para o controle da direção de rotação do motor, uma ponte H foi conectada ao sistema com a finalidade de receber os dados de comando do microcontrolador, e assim girar o motor no sentido horário ou anti-horário, a depender dos parâmetros estabelecidos na programação.

Uma ponte H consiste em diversos transistores polarizados como chaves eletrônicas (modos saturação e corte), dispostos de tal forma que podem conduzir a corrente elétrica através do motor CC em ambos os sentidos (do pólo positivo para o negativo, ou vice-versa) (RAMÍREZ, 2015). Uma configuração básica de uma ponte H é composta por 4 chaves, como mostra a Figura 3.15.

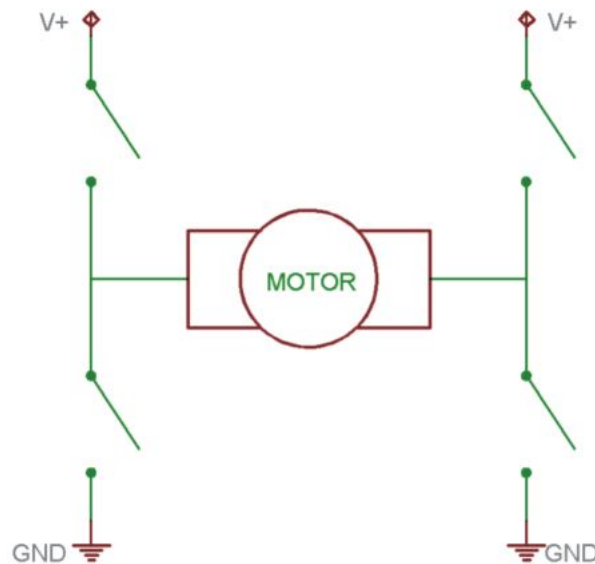


Figura 3.15 – Configuração de uma ponte H de quatro chaves (PATSKO, 2006).

O uso da ponte H é importante justamente para se ter um controle sobre a direção de atuação do motor, tendo em vista que ao ligar um motor de corrente contínua diretamente a uma fonte CC este irá girar com velocidade constante e em uma única direção. Por outro lado, utilizando uma ponte H, basta fechar um par de chave diagonalmente opostas que a corrente irá fluir pelo motor em um determinado sentido, e ao alterar o par de chaves a condução ocorrerá no sentido contrário.

Para a cortina deste projeto, o *driver* de ponte H usado foi o da Figura 3.16, que utiliza o *chip* L9110 e é capaz de atuar sobre dois motores CC simultaneamente.

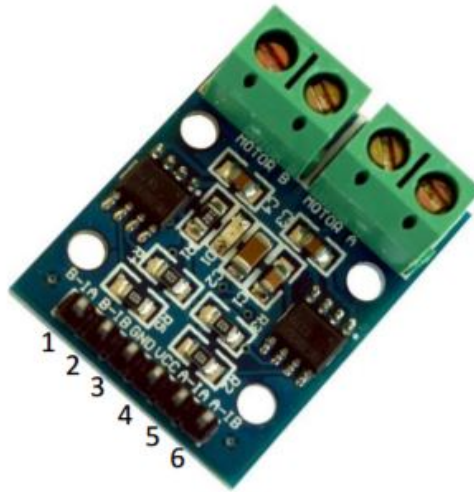


Figura 3.16 – Módulo ponte H com o circuito integrado L9110 (RAMÍREZ, 2015).

Identificação dos pinos:

- 1: entrada A do motor B;
- 2: entrada B do motor B;
- 3: *ground*;
- 4: tensão de entrada (2.5V-12V CC);
- 5: entrada A do motor A;
- 6: entrada B do motor A.

Além disso, também foram usadas chaves de fim de curso para limitar o máximo que a cortina irá abrir ou fechar. Desta forma, o sistema terá controle sobre a posição em que a cortina se encontra, identificando se já está completamente aberta ou fechada.

O circuito montado para realizar o controle do motor, e consequentemente da cortina, está expresso na Figura 3.17. Conforme é possível verificar no esquema do circuito em questão, a ponte H foi energizada através de uma fonte externa de 12V, isso se deve ao fato da tensão e corrente do motor usado serem superiores aos valores fornecidos pelo Arduino.

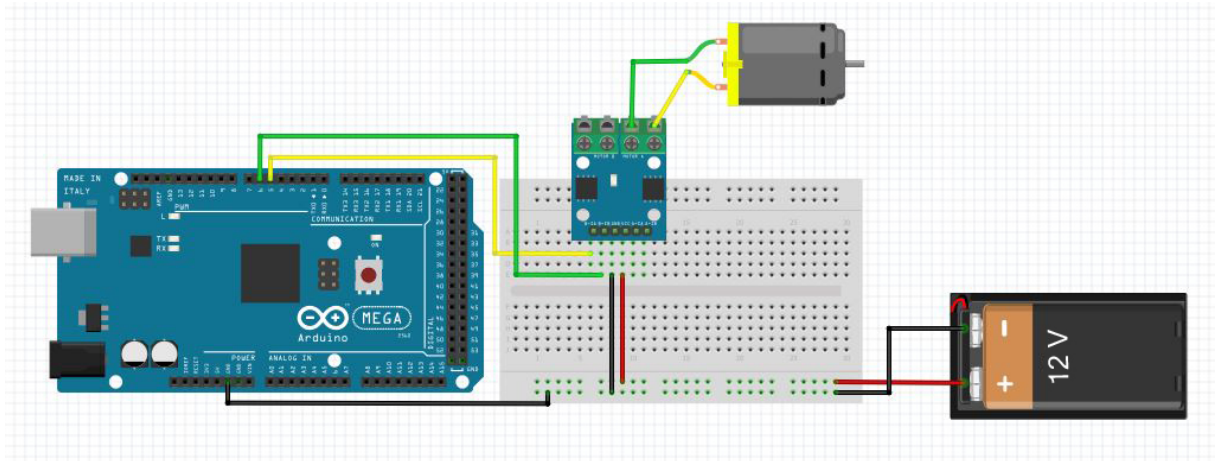


Figura 3.17 – Esquemático do circuito para o controle de rotação do motor CC da cortina (Próprio Autor).

3.3.4 Unidade de Comunicação

A unidade de comunicação é responsável pelo envio dos dados à internet e pela recepção dos comandos solicitados através da plataforma. Essa troca de informação é feita via *Wi-Fi*, e para realizar esta comunicação o componente escolhido foi o ESP8266.

Este dispositivo é um módulo de comunicação Wi-Fi, que apresenta o protocolo TCP/IP integrado à sua placa, permitindo que microcontroladores sejam capazes de se comunicar com outros dispositivos e/ou com a internet sem o uso de cabeamento (ESPRESSIF, 2017).

Este módulo pode ser configurado de duas maneiras distintas, sendo estas o modo *Access Point* (AP) e o modo *Station*.

Na configuração AP, o ESP8266 se comporta como um mini roteador, fazendo a extensão do sinal da rede a qual está conectado e permitindo que outros dispositivos se conectem através dele ao sistema. Já na configuração *Station*, o próprio ESP se conecta a um *Access Point*, tal como um roteador, permitindo que o sistema seja acessado através da Internet. Além disso, também é possível programar o ESP para que atue nos dois modos alternadamente, de acordo com a exigência do usuário.

Na Figura 3.18 é apresentado um dos modelos do ESP8266, o ESP-01. Este modelo possui 1MB de memória *flash* e utiliza o padrão de rede 802.11 b/g/n (ESPRESSIF, 2017). Por possuir apenas dois pinos de entrada e saída, a aplicação do dispositivo nos projetos da Internet das Coisas é limitada. Entretanto, para a presente proposta, este módulo atende todas as expectativas.

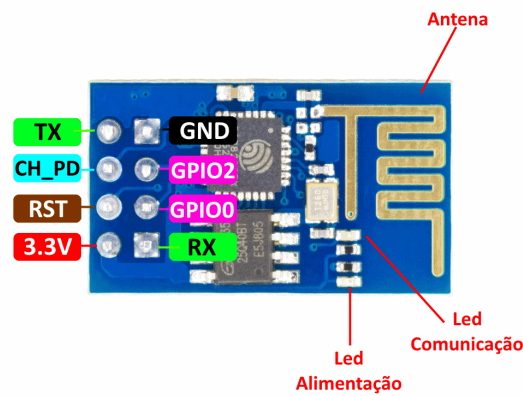


Figura 3.18 – Representação dos pinos do ESP8266 (DOUKAS, 2012).

Identificação dos pinos do ESP8266:

- TX: transmissor do módulo;
- RX: receptor do módulo;
- GND: *Ground*;
- VCC: terminal de alimentação (Recomendado 3.3V);
- CHPD: sinal para habilitar o chip;
- RST: sinal para reiniciar o módulo;
- GPIO0 e GPIO2: entradas e saídas do módulo;

Na Figura 3.19 é apresentado o circuito que foi implementado para realizar a comunicação entre o microcontrolador e a plataforma da Internet.

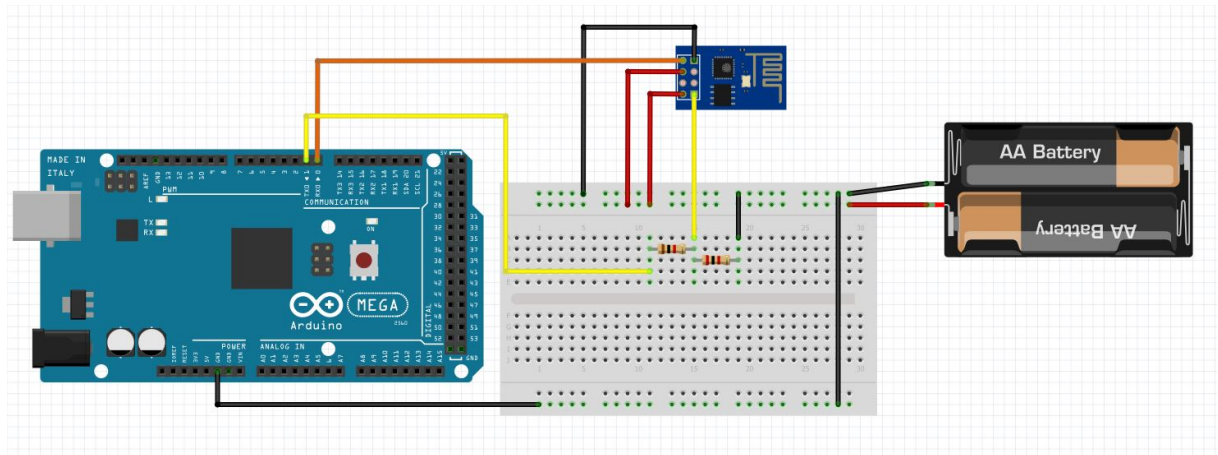


Figura 3.19 – Esquemático do circuito para transmissão dos dados entre a plataforma e o microcontrolador (Próprio Autor).

Apesar do Arduino fornecer uma tensão de 3.3 volts através de um dos seus pinos, a corrente exigida pelo ESP8266 pode chegar a 170 mA dependendo da operação realizada, conforme pode ser observado na Tabela 3.3, que é muito superior aos 50 mA fornecidos por este pino. Sendo assim, o ESP foi energizado por uma fonte externa de 3.3 volts.

Tabela 3.3 – Consumo de corrente do ESP8266, na transmissão de dados, para diferentes padrões de comunicação (ESPRESSIF, 2017).

Parâmetros	Típico	Unidade
Tx 802.11b,CCK 11Mbps, P OUT=+17dBm	170	mA
Tx 802.11b,OFDM 54Mbps, P OUT=+15dBm	140	mA
Tx 802.11n, MCS7, P OUT=+13dBm	120	mA
Rx 802.11b, 1024 bytes, -80dBm	50	mA
Rx 802.11g, 1024 bytes, -70dBm	56	mA
Rx 802.11n, 1024 bytes, -65dBm	56	mA

3.3.5 Integração do *Hardware*

Após a montagem de cada uma das partes do sistema separadamente e a verificação do seu funcionamento, os circuitos foram agrupados conforme mostra o diagrama do Apêndice A.

3.4 Metodologia do *Software*

O ambiente para desenvolvimento do código de controle do sistema foi o *Arduino Software (IDE)*, versão 1.6.13, utilizando a linguagem de programação C++. Este *soft-*

ware apresenta algumas bibliotecas que por proporcionarem funcionalidades extras, facilitam o controle de dispositivos periféricos e a manipulação dos dados.

Assim como no *hardware*, a programação de cada parte do software do sistema também foi subdividida em blocos conforme apresentado na Figura 3.20. Os códigos usados para a programação do sistema estão no Apêndice B.

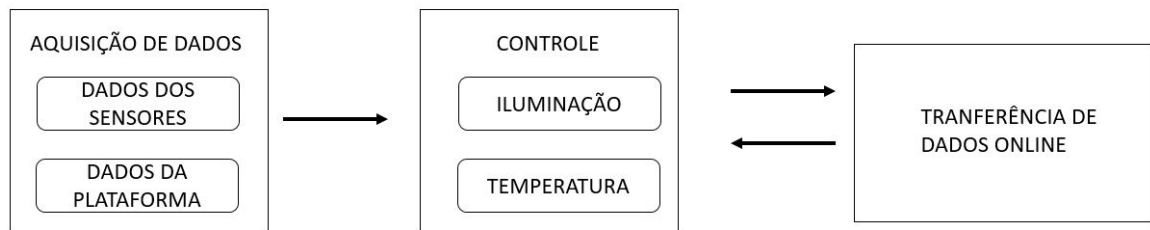


Figura 3.20 – Diagrama de blocos do *Software* (Próprio Autor).

3.4.1 Aquisição dos dados dos sensores

O microcontrolador foi programado para realizar leitura contínua dos dados de cada um dos sensores simultaneamente. Para o sensor LDR foi necessário realizar a interpretação das informações passadas ao microcontrolador, uma vez que os valores obtidos pelo Arduino não se referem diretamente ao valor da luminosidade incidida sobre o LDR, mas sim à tensão de saída do circuito ao qual o sensor esta conectado (Figura 3.5).

Como a tensão obtida na leitura do pino é digitalizada por um conversor analógico-digital de 10 bits, os valores passados ao software seguem uma faixa de 0 a 1023. Sendo 0 o valor correspondente ao nível mais baixo de luminosidade sobre o sensor, enquanto 1023 refere-se ao máximo da luminosidade interpretada pelo LDR no circuito.

De tal forma que os extremos desta faixa são os valores correspondentes ao mínimo e máximo da tensão de saída do circuito ao se variar a luminosidade que incide sobre o LDR.

Portanto, os valores recebidos pelo Arduino foram relacionados com a variação da luminosidade. O procedimento adotado, foi a realização de testes onde variou-se a incidência de luz sobre o sensor LDR, e a cada variação eram observados os valores obtidos no microcontrolador. Com base nesses testes, foram determinados, qualitativamente, as condições de iluminação do ambiente dentre o intervalo de 0 a 1023. Tais condições e suas respectivas faixas de valores estão expressos na Tabela 3.4.

Para a configuração da aquisição dos dados da temperatura foi utilizada uma biblioteca pública do sensor DHT11, específica para o Arduino, a DHT.h. Esta biblioteca faz automaticamente a conversão dos dados obtidos pelo sensor para valores de temperatura dentro da escala de graus Celsius.

Tabela 3.4 – Faixas de valores adotados para a representação da luminosidade (Próprio Autor).

Padrão de luminosidade		
Baixa	Ideal	Alta
0 - 449	450 - 750	751 - 1023

3.4.2 Aquisição dos comandos da plataforma

A priori, a plataforma Cayenne foi configurada para atuar sobre a placa do Arduino Mega, e neste processo um código de acesso foi disponibilizado para se estabelecer a conexão. Este código é enviado à plataforma através do ESP8266 sempre que se deseja ativar o sistema.

Como dito na Seção 3.2, a plataforma Cayenne utiliza janelas para exibir os dados dos sensores e realizar alguma ação sobre o sistema. Logo, para exibir ao usuário as informações de temperatura, iluminação e umidade em tempo real, desconsiderando-se o atraso da comunicação entre os competentes, três diferentes "janelas" foram configuradas para receber as informações dos sensores constantemente.

Essas janelas possuem duas formas de visualização, tanto por meio dos valores específicos no momento que o ambiente esta sendo monitorado, quanto através de um gráfico do comportamento de cada uma das três variáveis ao longo do funcionamento do sistema. Na Figura 3.21 está a representação de cada uma das formas.



Figura 3.21 – Janelas para as duas opções de visualização dos dados dos sensores (Próprio Autor).

Assim como as janelas para visualização dos dados, atuadores também foram configurados para possibilitar ao usuário alterar a temperatura e iluminação do ambiente em tempo real, além de ligar e desligar o ar condicionado.

Para o acionamento do ar condicionado através da plataforma foi utilizado um botão digital denominado "Power Ar". Esse botão envia sinais de 0 ou 1 ao microcontrolador, que após uma análise das condições estabelecidas irá desligar ou ligar o ar condicionado, respectivamente. Já para a variação da temperatura, foi configurado um atuador, classificado como "Variar Temperatura", que envia ao sistema sinais que representam os novos *setpoints* de temperatura desejados pelo usuário. Os valores destes sinais variam de 18° a 26° Celsius.

Entretanto, devido à conversão analógica-digital realizada na transmissão destes dados, o valor recebido pelo microcontrolador corresponde ao valor da temperatura configurada na plataforma multiplicado por 1023. Logo, é preciso realizar a divisão deste valor, a fim de se obter o valor exato da temperatura para a qual deseja-se alterar.

Cada vez que o usuário altera o valor deste atuador, o sistema irá enviar ao ar condicionado o sinal que corresponde à temperatura deste valor. Mas para realizar o envio, alguns parâmetros são avaliados como por exemplo, verificar se o ar condicionado realmente está ativado.

Quanto à iluminação, também utilizou-se um atuador analógico da plataforma para que o usuário possa alterar o nível de iluminância desejado, caso o mesmo queira um valor diferente do que foi determinado como ideal.

Sendo assim, é disponibilizado um botão para variação da iluminação. Este botão apresenta três diferentes níveis, sendo eles (100, 450 e 800). Cada um desses valores representa uma faixa de iluminação predefinida, que correspondem a luminosidade baixa, ideal e alta, respectivamente. Isso significa que ao definir o *setpoint* como 450 por exemplo, o sistema buscará se ajustar à faixa de valores de iluminação considerados ideais, que foram apresentados na Tabela 3.4. O mesmo ocorre para os outros níveis.

Caso não haja nenhuma alteração neste atuador, o sistema irá sempre buscar se ajustar aos valores ideais de luminosidade.

Com a implementação de cada uma das janelas descritas acima na plataforma, a interface gráfica disponibilizada para o usuário é mostrada na Figura 3.22.

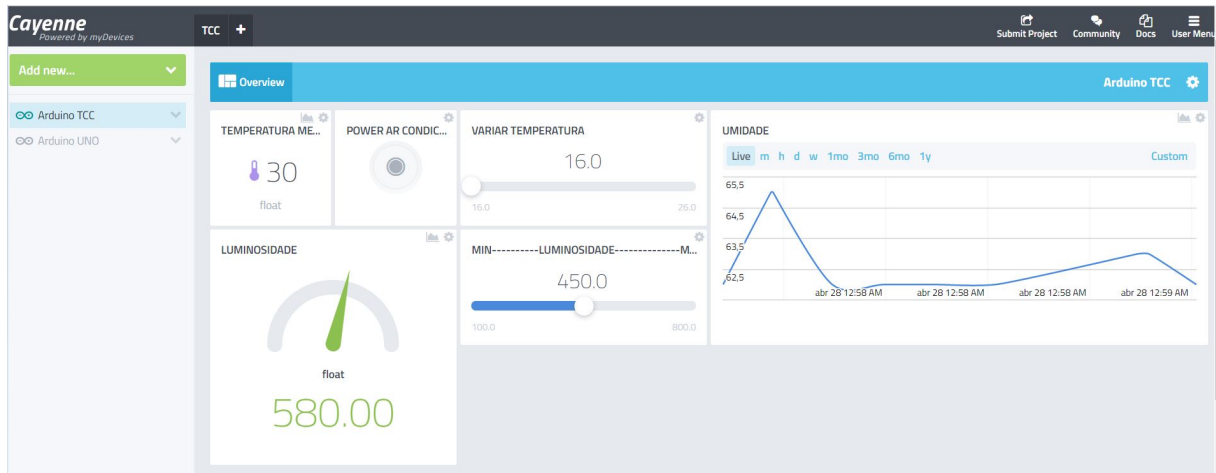


Figura 3.22 – Interface de comunicação da plataforma que é disponibilizada ao usuário (Próprio Autor).

3.4.3 Lógica de controle da iluminação

O controle da iluminação é realizado por uma malha fechada com realimentação negativa, de tal forma que a realimentação corresponde ao valor da luminosidade medido pelo sensor LDR presente no sistema. O diagrama de blocos deste controle está representado na Figura 3.23:

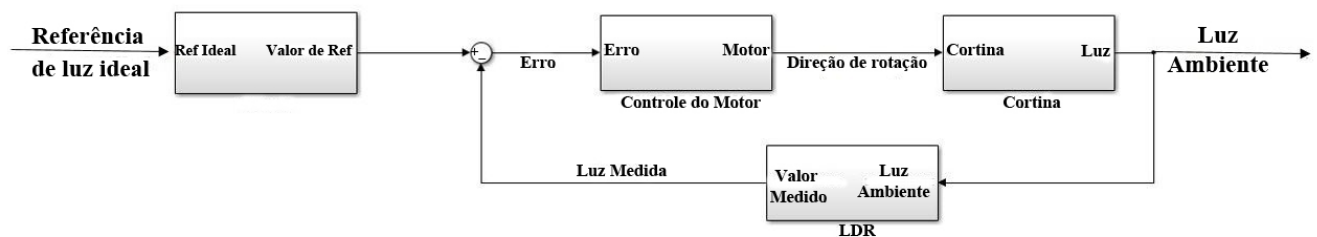


Figura 3.23 – Diagrama de blocos da lógica de controle da iluminação (Próprio Autor).

A entrada do sistema corresponde à referência de luz que é predefinida por uma faixa de valores entre 400 e 750, que são correspondentes ao estado de iluminação ideal conforme mostrado na Tabela 3.4, ou ao valor da referência alterada pelo usuário através da plataforma Cayenne.

O valor da entrada é comparado com o valor medido pelo sensor LDR e seu resultado é passado para o controle do motor CC, que verificará em qual sentido o motor deverá rotacionar.

As decisões do controlador são baseadas no erro, de tal forma que se o erro for negativo isso significa que a quantidade de luz no recinto é maior que a referência escolhida. Neste caso, a cortina irá fechar até a luminosidade se ajustar aos níveis esperados.

3.4.4 Lógica de controle do ar condicionado

O ar condicionado é controlado baseado no setpoint de referência dado ao sistema. A priori, o valor da temperatura ideal considerada é de 22° Celsius. Assim, ao iniciar o sistema, o ar condicionado é automaticamente configurado para atuar nesta temperatura. Concomitante a isso, os dados obtidos pelo sensor DHT11 são enviados à plataforma online.

O ar condicionado pode ser inicializado ou desligado pelo usuário a qualquer momento através do botão "Power Ar" disponibilizado na plataforma. Sempre que este botão for ativado, o microcontrolador recebe um sinal lógico 1, e verifica o estado atual do aparelho, ou seja, se este já encontra-se ligado ou não. Caso determinado que o aparelho estava desligado, um sinal infravermelho é enviado a seu receptor para ativá-lo. A ação contrária é realizada ao desativar o botão.

Além disso, também é possível variar a temperatura do ar condicionado sempre que o usuário desejar. Neste caso, ao variar o botão da temperatura disponibilizado na plataforma, o sistema recebe e identifica o novo *setpoint*, em seguida um sinal infravermelho correspondente a este valor de temperatura é enviado ao ar condicionado, e a temperatura de referência é então alterada. Esta ação só ocorre caso seja confirmado pelo microcontrolador que o ar condicionado encontra-se ligado.

Uma vez estabelecido o *setpoint*, seja ele o valor configurado como ideal ou estabelecido pelo usuário, o sistema passa a verificar a temperatura em que o ambiente se encontra após um determinado intervalo de tempo, a fim de determinar se ela condiz com o valor da referência estabelecida. Isso é feito porque apesar do ar condicionado estar configurado para a temperatura desejada, é possível que haja variações nesse valor, ao longo funcionamento do sistema, por condições ambientais ou devido a aglomerações de pessoas e/ou aparelhos eletrônicos no ambiente, por exemplo.

Desta forma, caso seja observado que o ambiente está com uma temperatura superior ao da referência, o sistema irá enviar o sinal correspondente ao valor da temperatura imediatamente abaixo do que está na referência. Por outro lado, se observado que o ambiente está com uma temperatura inferior ao da referência, o sistema irá enviar o sinal correspondente ao valor da temperatura imediatamente acima desta. Este processo se repetirá continuamente, a fim de se atingir o *setpoint* configurado.

Tabela 3.5 – Configuração da conexão entre o ESP8266 e o Arduino para realizar a atualização do *firmware* (Próprio Autor).

Conexões	
ESP 8266	Arduino
VCC	3.3V
Ground	Ground
CHPD	3.3V
RST	3.3V
GPIO0	3.3V

Após a montagem do circuito, o programa ESP FLASH TOOL foi configurado com as informações da *firmware*, e então a atualização foi realizada. Em seguida, testou-se a comunicação entre o Arduino e o ESP8266 através do monitor serial, a fim de verificar se o *firmware* estava funcionando corretamente. Toda a comunicação entre os dois componentes é realizada por meio de comandos AT, exemplificados na Tabela 3.6.

Contudo, a comunicação realizada entre a plataforma Cayenne e o ESP utiliza os comandos AT automaticamente. Sempre que o programa é compilado, o microcontrolador envia os comandos necessários para realizar a conexão com a rede. Uma vez estabelecida a comunicação com a rede, é solicitado o acesso à plataforma através do protocolo TCP/IP, e então o código de acesso é enviado.

Uma vez conectado com a plataforma, o ESP está habilitado para realizar a transmissão de dados entre o Arduino e a Cayenne constantemente.

Tabela 3.6 – Lista de comandos AT para configurar o ESP8266 (ESPRESSIF, 2016).

Comandos	Descrição
AT	Testa a inicialização dos comandos.
AT-RST	Reinicia o módulo.
AT-GMR	Verifica a versão dos comandos AT da Firmware utilizadas.
AT+CWMODE?	Verificar o modo de operação que o ESP esta funcionando, retornando: 1 - para modo Station; 2 - para modo Access Point; 3 - para modo Station + AccesPoint.
AT+CWMODE=<mode>	Configura o modo de operação, sendo: <1> - para modo Station; <2> - para modo Access Point; <3> - para modo Station + AccesPoint.
AT+CWLAP	Verifica as redes disponíveis.
AT+CWLAP	Conecta o dispositivo à rede predefinida.
AT+CWLIF	Verifica o endereço IP da rede que o dispositivo esta conectado.

Capítulo 4

Conclusões e Considerações Finais

Neste capítulo serão discutidas as conclusões do trabalho proposto e os desafios enfrentados durante o desenvolvimento do projeto, bem como, ideias para aprimorar o sistema.

4.1 Conclusões

Diante dos conceitos explorados no decorrer deste trabalho, nota-se a importância de aprofundar os conhecimentos sobre a Internet das Coisas, pelo fato dessa tecnologia trazer mudanças significativas no que diz respeito às relações de comunicação e interação entre uma rede de computadores e seus usuários. Isso porque com a introdução da IoT, os objetos também passaram a ser usuário da rede, através de dispositivos, *softwares* e protocolos adaptados para a Internet das Coisas.

Além disso, acredita-se que em alguns anos a IoT estará cada vez mais presente no cotidiano das pessoas, inserida em vários dispositivos, de forma que os equipamentos serão capazes de monitorar ambientes ou pessoas remotamente, podendo proporcionar conforto, economia, qualidade de vida, entre outras vantagens.

Outro ponto importante a se considerar é que a Internet das Coisas também tem influenciado a economia. Empresas têm considerado essa tecnologia uma oportunidade para negócios e investimentos, e assim têm surgido diversas aplicações inseridas nessa área. Muitas empresas tais como Samsung, Google e Sony têm investido em relógios e óculos que, por meio de dispositivos inseridos, a eles são capazes de se conectar a determinadas redes e realizar troca de informações.

Entretanto, como visto neste trabalho, ainda são muitos os desafios a serem enfrentados para a dispersão unificada dessa tecnologia. Será necessário padronizar os protocolos que vêm sendo aplicados nos sistemas IoT e unificar arquiteturas de desenvolvimento para que seja facilitada a comunicação entre dispositivos de sistemas diferentes. Devido a este

problema, passaram a surgir plataformas de comunicação que proporcionam a interação entre diferentes dispositivos.

Essas plataformas, além de promoverem a interoperabilidade, também têm facilitado o desenvolvimento de projetos voltados para a Internet das Coisas por não exigir níveis altos de programação. Esse avanço tem permitido que cada vez mais pessoas explorem os conceitos dessa nova tecnologia.

Sendo assim, foi com base nesses conceitos da Internet das Coisas, e com o uso de uma das plataformas disponíveis para IoT que o objetivo específico deste trabalho foi alcançado. O sistema de controle da temperatura e luminosidade de um ambiente funcionou conforme o esperado, ajustando a seus parâmetros sempre que as condições do ambiente se alteravam ou quando o usuário solicitava alguma alteração via Internet.

Aplicar na prática os conceitos da IoT foi importante por possibilitar o contato com alguns dos problemas de compatibilidade de comunicação existentes entre dispositivos, sendo necessário a busca por configurações que possibilitem os objetos operarem sob o mesmo protocolo. Um exemplo disso foi a necessidade de atualização de *firmware* do módulo ESP8266, que permitisse a comunicação entre ele, o microcontrolador e plataforma usada.

Contudo, a Internet da Coisas apresenta-se como uma ferramenta de elevado potencial econômico e tecnológico, que devido a seus diversos benefícios e inúmeras aplicações possíveis, pode ser considerado como mais uma revolução no que diz respeito à comunicação virtual.

4.2 Propostas para Projetos Futuros

Com base no que foi apresentado neste trabalho, sugere-se para trabalhos futuros:

- Realizar o estudo de outras arquiteturas disponíveis para as aplicações da Internet das Coisas;
- Explorar outras plataformas que são aplicadas nos projetos de IoT;
- Realizar estudos de impactos econômicos com relação à aplicação de projetos de Internet das Coisas;
- Acrescentar ao projeto desenvolvido neste trabalho o acionamento de uma lâmpada para ser acionada quando o nível de iluminação natural estiver inferior ao mínimo estabelecido no sistema, ou seja, para a situação em que a cortina esta completamente aberta mas ainda assim a luminosidade do ambiente encontra-se abaixo do configurado;

- Desenvolver um aplicativo para celular que permita acessar o sistema com mais praticidade.

Referências

- AOSONG. *Temperature and Humidity module, DHT11 Product Manual*. 2016. Disponível em: <http://img.filipeflop.com/files/download/Datasheet_DHT11.pdf?_ga=1.122709323.1767147774.1492289418>. Acesso em: 15 Fev 2017.
- ARDUINO. *Compare board specs*. 2016. Disponível em: <<https://www.arduino.cc/en/Products/Compare>>. Acesso em: 05 Abr 2017.
- ARDUINO. *Getting Started with Arduino*. 2016. Disponível em: <<https://www.arduino.cc/en/Guide/HomePage>>. Acesso em: 18 Out 2016.
- BARROS, M. *MQTT - Protocolos para IoT*. 2015. Disponível em: <<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>>. Acesso em: 17 Out 2016.
- BLUETOOTH. *Bluetooth is the Future*. 2017. Disponível em: <<https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth/5/>>. Acesso em: 17 Mar 2017.
- CAYENNE. *Cayenne*. 2016. Disponível em: <<http://www.cayenne-mydevices.com/>>. Acesso em: 10 Out 2016.
- DOUKAS, C. *Building Internet of Things with the Arduino*. [S.l.: s.n.], 2012.
- ESPRESSIF. *ESP8266 AT Instruction Set*. 2016. Disponível em: <https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf>. Acesso em: 20 Jan 2017.
- ESPRESSIF. *ESP8266EX Datasheet*. 2017. Disponível em: <http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acesso em: 20 Jan 2017.
- FERREIRA, H. G. C. *Arquitetura de Middleware para Internet das Coisas*. Dissertação de Mestrado, 2014.
- GHAREGOZLOU, M. *A Abordagem Industrial para a Internet das Coisas*. 2016. Disponível em: <<http://ecommercenews.com.br/artigos/tendencias-artigos/a-abordagem-aberta-para-o-desenvolvimento-de-aplicacoes-para-a-internet-industrial-das-coisas-iiot>>. Acesso em: 24 ago 2016.
- HACKSTER. *Cayenne*. 2016. Disponível em: <<https://www.hackster.io/Cayenne/products/cayenne>>. Acesso em: 15 Dez 2016.

- IDG. *Carreira: Habilidade Necessárias para Vencer no Mundo da IoT*. 2016. Disponível em: <<http://computerworld.com.br/carreira-habilidades-necessarias-para-vencer-no-mundo-da-iot>>. Acesso em: 24 Ago 2016.
- IEEE. *Authoritative Dictionary of IEEE Standards Terms*. 2000. Disponível em: <<http://ieeexplore.ieee.org/document/4116787/>>. Acesso em: 21 Abr 2017.
- IEEE. *The Future of IoT*. 2016. Disponível em: <<https://transmitter.ieee.org/the-future-of-iot/>>. Acesso em: 24 Ago 2016.
- MANZANO, H. *Como a Internet das Coisas Vai Revolucionar o Setor Financeiro*. 2016. Disponível em: <<http://computerworld.com.br/carreira-habilidades-necessarias-para-vencer-no-mundo-da-iot>>. Acesso em: 24 Ago 2016.
- MOSQUITTO. *Mosquitto*. 2016. Disponível em: <<https://mosquitto.org/>>. Acesso em: 11 12 2016.
- NORRIS, D. *The Internet of Things: Do-Itself Projects with Arduino, Raspberry Pi and BeagleBone Black*. [S.l.: s.n.], 2015.
- PATSKO, L. F. *Montagem da ponte H*. 2006. Disponível em: <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_montagem_de_uma_ponte_h.pdf>. Acesso em: 20 Mar 2017.
- PATSKO, L. F. *Aplicações, funcionamento e utilização de sensores*. 2014. Disponível em: <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>. Acesso em: 17 Fev 2017.
- PRESS, G. *A very short history of Internet of Things*. 2014. Disponível em: <<http://www.forbes.com/sites/gilpress/2014/06/18/a-very-short-history-of-the-internet-of-things/2/>>. Acesso em: 17 Ago 2016.
- PRESSER, M. *Inspirando a Internet das Coisas*. 2012.
- RADOVICI, A. *Lecture 4: Introduction to IoT*. 2016. Disponível em: <<http://ocw.cs.pub.ro/courses/iot/courses/04>>. Acesso em: 30 Set 2016.
- RAMÍREZ, E. F. F. *Acionamento de motor com ponte H*. 2015. Disponível em: <http://www.uel.br/pessoal/ernesto/arduino/11_Acionamento_de_motor_com_ponte_H_via_teclado.pdf>. Acesso em: 20 Mar 2017.
- RIJMENAM, M. V. *Where does the Internet of Things come from?* 2014. Disponível em: <<https://datafloq.com/read/where-does-the-internet-of-things-come-from/524>>. Acesso em: 17 Ago 2016.
- SANTOS, B. P. *Internet das Coisas: da Teorias à Prática*. 2014.
- SCHWARTZ, M. *Arduino Networking*. [S.l.: s.n.], 2014.
- SOUZA, F. *Arduino Mega 2560*. 2014. Disponível em: <<https://www.embarcados.com.br/arduino-mega-2560/>>. Acesso em: 10 Mar 2017.

SUNROM. *Light Dependent Resistor - LDR*. 2008. Disponível em: <<http://kennarar.vma.is/thor/v2011/vgr402/ldr.pdf>>. Acesso em: 17 Fev 2017.

TAURION, C. *Cloud Computing - Computação em Nuvem*. [S.l.: s.n.], 2009.

THINGSPEAK. *The open data platform for Internet of Things*. 2016. Disponível em: <<https://thingspeak.com/>>. Acesso em: 12 Out 2016.

VISHAY. *IR Receiver Modules for Remote Control Systems*. 2016. Disponível em: <<http://www.vishay.com/docs/82459/tsop48.pdf>>. Acesso em: 15 Mar 2017.

WAKA, G. Controle remoto de tomadas elétricas baseado no conceito da internet das coisas. 2015.

ZARGHAMI, S. *Middleware for Internet of Things*. Dissertação de Mestrado, 2013.

Apêndice A

Diagrama do circuito

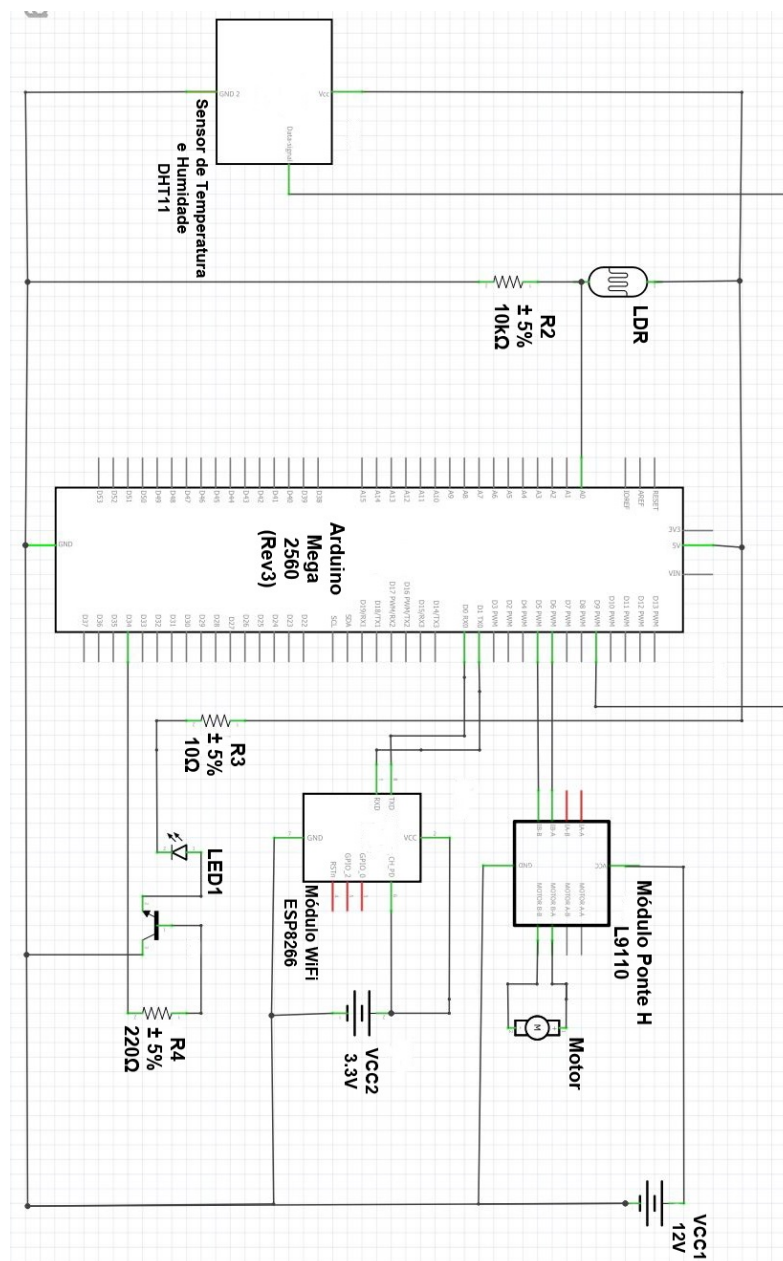


Figura A.1 – Esquema elétrico dos circuitos do sistema integrados.

Apêndice B

Código do *Software*

```
//BIBLIOTECAS
#include <CayenneESP8266Shield.h> //PLATAFORMA
#include "DHT.h" // SENSOR DHT11
#include <IRremote.h> //EMISSOR INFRAVERMELHO

//CÓDIGO DE ACESSO À PLATAFORMA
char token[] = "-----";

// INFORMAÇÕES DA REDE WIFI QUE ESTÁ SE CONECTANDO

//char ssid[] = "-----";
//char password[] = "-----";

// CONFIGURANDO O ESP

#define EspSerial Serial
ESP8266 wifi(EspSerial);

// CONFIGURAÇÃO DO OBJETO PARA ENVIO DO SINAL INFRAVERMELHO
IRsend irsend;

// DHT11 CONFIGURATION

#define DHTPIN 2 // pino que estamos conectado
#define DHTTYPE DHT11 // DHT 11
```



```
DHT dht(DHTPIN, DHTTYPE);

//LDR CONFIGURATION

int sensorPin = A0; // select the input pin for the LDR sensor
int sensorValue=0; // variable to store the value coming from the sensor


// PINOS VIRTUAIS PARA TRANSFERÊNCIA DE DADOS COM A PLATAFORMA
#define VIRTUAL_PIN1 V1 //PINO SENSOR DE TEMPERATURA
#define VIRTUAL_PIN2 V2 //PINO SENSOR DE UMIDADE
#define VIRTUAL_PIN0 V3 //PINO SENSOR DE LUMINOSIDADE
#define VIRTUAL_PIN12 V12
#define VIRTUAL_PIN13 V4
#define VIRTUAL_PIN14 V13


// DEFINIÇÃO DAS VARIÁVEIS


//AR CONDICIONADO
int setpointAtual=0;
int temperatura = 0;
int khz = 38;
int ligado = 0;
int setpointRef = 22;
int enviou = 0;
int cont1 = 0;
int cont2 = 0;


// CORTINA
int HORARIO = 5; //PINO MOTOR
int ANTI_HORARIO = 6; //PINO MOTOR
int pinopwm = 7;
int pwm = 180;
float ref=0; //SETPOINT ALTERADO PELO USUÁRIO
float ref_inicial=450; //SETPOINT DE REFERÊNCIA
long previousMillis = 0;
long interval = 120000;
```

```
void setup()
{
  EspSerial.begin(115200);
  dht.begin();

  pinMode(HORARIO,OUTPUT);
  pinMode(ANTI_HORARIO,OUTPUT);

  digitalWrite(HORARIO,LOW);
  digitalWrite(ANTI_HORARIO,LOW);

  Cayenne.begin(token, wifi, ssid, password);

}

void loop()
{

  Cayenne.run();

  //FUNÇÃO DE CONTROLE DA CORTINA
  float a = ref;
  CORTINA(a);

  //CÓDIGO PARA CONTROLE DA TEMPERATURA
  unsigned long currentMillis = millis();
  if((currentMillis - previousMillis > interval) && (ligado==1)){

    if(temperatura > setpointRef){
      DiminuirTemp(setpointRef);
      previousMillis = currentMillis;
      setpointRef = setpointRef - 1;

    }else if(temperatura < setpointRef){
      AumentarTemp(setpointRef);
      setpointRef = setpointRef - 1;
      previousMillis = currentMillis;
    }
  }
```

```
}  
}  
  
///// LEITURA DOS SENSORES  
  
//SENSOR DHT11 - TEMPERATURA  
  
CAYENNE_OUT(VIRTUAL_PIN1)  
{  
  temperatura = dht.readTemperature();  
  Cayenne.virtualWrite(V1, temperatura);  
}  
  
//SENSOR DHT11 - UMIDADE  
  
CAYENNE_OUT(VIRTUAL_PIN2)  
{  
  float umidade = dht.readHumidity();  
  Cayenne.virtualWrite(V2, umidade);  
  
}  
  
//SENSOR LDR  
  
CAYENNE_OUT(VIRTUAL_PIN0)  
{  
  sensorValue = analogRead(sensorPin);  
  Cayenne.virtualWrite(V3, sensorValue);  
}  
  
/////SETPOINT DOS USUÁRIOS  
  
// Ligar/Desligar AR - O LED ESTÁ CONECTADO AO PINO 9 (ARDUINO MEGA)  
CAYENNE_IN(V8){  
  int power = getValue.asInt();  
  if(power == 1 && ligado==0){  
    ligarAr();  
    ligado=1;  
  }else if((power == 0 && ligado==1)){
```

```
desligarAr();
ligado=0;
}
}

//VARIACÃO DA TEMPERATURA
CAYENNE_IN(V5){
int vPin1 = getValue.asInt();
setpointAtual = vPin1/1023;
if ((ligado ==1) && (setpointAtual != setpointRef)){
trocarRef (setpointAtual);
setpointRef = setpointAtual;
}
}

//VARIACÃO DA LUMINOSIDADE
CAYENNE_IN(V6)
{ //AJUSTE FEITO PELO SISTEMA, TEM COMO REFERENCIA O VETOR [451...750];
double a = getValue.asDouble();
if(a==0){
ref=ref_inicial;
}else{
ref = a/1023;
}
}

//FUNÇÃO DE CONTROLE DA CORTINA - RESTRITO AOS SETPOINTS
void CORTINA(double ref)
{
int sensorPin = A0; // select the input pin for the LDR sensor
double sensorValue = analogRead(sensorPin);
int aberta = digitalRead(22); //laranja
int fechada = digitalRead(24); //verde

if(ref==100 && sensorValue>ref && fechada==1 ){
//Gira o Motor A no sentido horario
digitalWrite(HORARIO, HIGH);
digitalWrite(ANTI_HORARIO, LOW);
analogWrite(pinopwm,pwm);
}
```

```
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
}else if(ref==800 && sensorValue<ref && aberta==1){
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, HIGH);
analogWrite(pinopwm,pwm);
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
}else if(ref==450){
AUTOMATICO();
}else{
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
}
}

//FUNÇÃO DE CONTROLE DA CORTINA - AJUSTANDO-SE AUTOMATICAMENTE
void AUTOMATICO()
{
int sensorPin = A0; // select the input pin for the LDR sensor

int aberta = digitalRead(22);
int fechada = digitalRead(24);
int sensorValue = analogRead(sensorPin);

if(sensorValue<400 && aberta==0 && fechada==1){
//Gira o Motor A no sentido anti-horario
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
analogWrite(pinopwm,pwm);
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);

}

if(sensorValue>750 && aberta==0 && fechada==1){
```

```
digitalWrite(HORARIO, HIGH);
digitalWrite(ANTI_HORARIO, LOW);
analogWrite(pinopwm,pwm);
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
}

if(sensorValue<400 && fechada==0 && aberta==1){
//Gira o Motor A no sentido anti-horario
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, HIGH);
analogWrite(pinopwm,pwm);
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
}

if(sensorValue>750 && fechada==0 && aberta==1){

digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);
delay(50);
}

if(sensorValue<400 && fechada==1 && aberta==1){
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, HIGH);
analogWrite(pinopwm,pwm);
delay(50);
digitalWrite(HORARIO, LOW);
digitalWrite(ANTI_HORARIO, LOW);

}

if(sensorValue>750 && aberta==1 && fechada==1){
digitalWrite(HORARIO, HIGH);
digitalWrite(ANTI_HORARIO, LOW);
```

```
analogWrite(pinopwm,pwm);  
delay(50);  
digitalWrite(HORARIO, LOW);  
digitalWrite(ANTI_HORARIO, LOW);  
}  
}
```

```
//FUNÇÕES DE CONTROLE DO AR CONDICIONADO
```

```
//FUNÇÃO PARA TROCAR A REFERÊNCIA DO SISTEMA
```

```
void trocarRef (int setpointAtual){  
  switch(setpointAtual){  
    case 18:  
    {  
      Temp18();  
    }  
    break;  
  
    case 19:  
    {  
      Temp19();  
    }  
    break;  
  
    case 20:  
    {  
      Temp20();  
    }  
    break;  
  
    case 21:  
    {  
      Temp21();  
    }  
    break;  
  
    case 23:  
    {  
      Temp23();  
    }  
  }  
}
```

```
}  
break;  
  
case 24:  
{  
Temp24();  
}  
break;  
  
case 25:  
{  
Temp25();  
}  
break;  
  
case 26:  
{  
Temp26();  
}  
break;  
  
default:  
  
break;  
  
}  
}  
  
//FUNÇÃO PARA DIMINUIR A TEMPERATURA  
void DiminuirTemp(int setpointRef){  
switch(setpointRef){  
case 18:  
  
break;  
  
case 19:  
Temp18();  
break;
```



```
case 20:
Temp19();
break;
```

```
case 21:
Temp20();
break;
```

```
case 23:
Temp23();
break;
```

```
case 24:
Temp23();
break;
```

```
case 25:
Temp24();
break;
```

```
case 26:
Temp25();
break;
}
}
```

```
//FUNÇÃO PARA AUMENTAR A TEMPERATURA
```

```
void AumentarTemp(int setpointRef){
switch(setpointRef){
case 18:
Temp19();
break;
```

```
case 19:
Temp20();
break;
```

```
case 20:
Temp21();
break;
```

```
case 21:
Temp21();
break;
```

```
case 23:
Temp24();
break;
```

```
case 24:
Temp23();
break;
```

```
case 25:
Temp26();
break;
```

```
case 26:
break;
}
```

```
}
```

```
//SINAIS DO AR CONDICIONADO DO DEL - HITACHI
```

```
void ligarAr(){
unsigned const irSignal[] = {3472, 1588, 532, 1196, 532, 332, 536, 328, 564, 328, 5
irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]), khz);
}
```

```
void desligarAr(){
unsigned const irSignal[] = {3392, 1668, 480, 1224, 508, 384, 508, 360, 508, 356, 5
irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]), khz); //Note the a
}
//
void Temp18(){
```

