

UNIVERSIDADE FEEVALE

EDUARDO WIEST

ESTUDO COMPARATIVO ENTRE ESP8266 E RASPBERRY PI  
NUMA APLICAÇÃO DE INTERNET DAS COISAS (*IOT*)

Novo Hamburgo  
2018

EDUARDO WIEST

ESTUDO COMPARATIVO ENTRE ESP8266 E RASPBERRY PI  
NUMA APLICAÇÃO DE INTERNET DAS COISAS (*IOT*)

Trabalho de Conclusão de Curso  
apresentado como requisito parcial  
à obtenção do grau de Bacharel em  
Ciência da Computação pela  
Universidade Feevale.

Orientador: Prof.º Ms. Vandersilvio Da Silva

Novo Hamburgo  
2018

## AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial: a minha esposa Aline, por sempre me apoiar e ter paciência nos momentos mais difíceis, a minha mãe Alice, por sempre me incentivar. Ao meu orientador Vandersílvia da Silva, por me auxiliar a desenvolver este projeto até mesmo em finais de semana e ser um exemplo de profissional. Aos meus colegas de curso pela amizade e ajuda nos momentos em que precisei e a todos que de uma forma ou outra contribuíram para a realização deste trabalho.

## RESUMO

A cada ano, são desenvolvidos diversos dispositivos integrados com inteligência para realizar a combinação das mais variadas tarefas, tais como sensoriamento, processamento e comunicação. Estes componentes, através da Internet das Coisas (*Internet of Things (IoT)*), podem ser conectados à Internet e serem acessados por usuários e sistemas. A *IoT* comporta uma quantidade significativa de possibilidades de interconexão, abrangendo residências, indústrias, cidades inteligentes, áreas da saúde, entre tantas outras. Hoje, no mercado, existem diversas plataformas para o desenvolvimento de aplicações na área de *IoT*, desde as simples até as mais complexas. Duas das soluções mais populares, são o ESP8266 e o Raspberry Pi, com significativa quantidade de projetos. Normalmente, estas ou qualquer outra plataforma obedecem a um fluxo de funcionamento simples: monitorar, processar e controlar/atuar. O esperado é que estes dispositivos sejam autônomos e com o maior ciclo de durabilidade entre substituições ou recarga de bateria quando instalados remotamente. Neste estudo, implementou-se e comparou-se duas soluções para acesso remoto de parâmetros de controle das condições básicas de sobrevivência de uma planta e disponibilizando os dados em sites de repositórios on-line, além medir o consumo de energia de cada uma das duas soluções, com ESP8266 e outra com Raspberry Pi.

Palavras Chave: Raspberry Pi. ESP8266. Internet das coisas *IoT*. Repositórios on-line.

## ABSTRACT

Each year, several integrated devices with intelligence are developed to perform the combination of the most varied tasks, such as sensing, processing and communication. These components, through the Internet of Things (*IoT*), can be connected to the Internet and accessed by users and systems. *IoT* contains a significant amount of interconnection possibilities, covering homes, industries, smart cities, health areas, among many others. Today, in the market, there are several platforms for the development of applications in the area of *IoT*, from the simple to the most complex. Two of the most popular solutions are ESP8266 and Raspberry Pi, with significant amount of projects. Usually, these or any other platform obeys a simple flow of operation: monitor, process and control / act. It is expected that these devices will be autonomous and with the longest cycle of durability between replacements or battery recharge when installed remotely. In this study, two solutions were implemented and compared for remote access control parameters of the basic conditions of survival of a plant and making the data available on online repositories sites, besides measuring the energy consumption of each of the two solutions, with ESP8266 and another with Raspberry Pi.

Keywords: Raspberry Pi. ESP8266. Internet of Things *IoT*. Online repositories.

## LISTA DE FIGURAS

Figura 1 - Evolução da Internet das Coisas .....	17
Figura 2 - Internet das Coisas .....	18
Figura 5 - GPIO Raspberry Pi 3 .....	23
Figura 6 - Versões disponíveis do módulo ESP8266 .....	24
Figura 7 - Pinagem ESP8266 .....	27
Figura 8 - Alcance vs Taxa de transferência de redes sem fio .....	29
Figura 9 - Diagrama de blocos de um controle de processos .....	33
Figura 10 - Funcionamento de um transdutor .....	34
Figura 11 - Sinal analógico.....	35
Figura 12 - Conversor A/D .....	35
Figura 13 - Sinal digital .....	36
Figura 14 - Sensor DHT11 .....	36
Figura 15 - Sensor de Luminosidade.....	37
Figura 16 - Sensor de Umidade do Solo.....	38
Figura 17 - Arquitetura do projeto.....	45
Figura 18 - Conexão dos sensores na placa ESP8266.....	47
Figura 19 - Conexão dos sensores na placa Raspberry Pi 3 .....	48
Figura 20 - Cron, agendador de tarefas do <i>Raspbian</i> .....	49
Figura 21 - Fluxograma seguido por cada aplicação.....	50
Figura 22 - Configuração do canal <i>ThingSpeak</i> .....	55
Figura 23 - Configuração do canal <i>ThingSpeak</i> .....	56
Figura 24 - Criando projeto dentro do ambiente <i>Google API</i> .....	57
Figura 25 - Informações da credencial de autenticação <i>Outh2</i> .....	58
Figura 26 - Inserção da chave de acesso.....	58

Figura 27 - Comparativo luminosidade dia x noite .....	60
Figura 28 - Posicionamento das plataformas.....	61
Figura 29 - Consumo do medidor de tensão e corrente.....	63
Figura 30 - Consumo da placa ESP8266 apenas com alimentação.....	64
Figura 31 - Consumo da placa ESP8266 em dois estados.....	64
Figura 32 - Consumo da placa Raspberry Pi 3 em dois estados sem ventilador .....	65
Figura 33 - Raspberry Pi 3 Comparação de temperatura x umidade do ar.....	67
Figura 34 - ESP8266 Comparação de temperatura x umidade do ar.....	68
Figura 35 - Comparação de umidade do solo .....	68
Figura 36 - Comparação de luminosidade.....	69
Figura 38 - Comparação de luminosidade com zona de corte.....	70
Figura 39 - Comparação de consumo entre placa ESP8266 e Raspberry Pi 3 .....	70

## LISTA DE TABELAS

Tabela 1 - Comparação entre as plataformas.....	28
Tabela 2 - ESP 8266 Tensão versus Corrente .....	65
Tabela 3 - Raspberry Pi 3 Tensão x Corrente sem ventilador com <i>WiFi</i> .....	66
Tabela 4 - Raspberry Pi 3 Tensão x Corrente com ventilador com <i>WiFi</i> .....	66
Tabela 5 - Envio de dados Raspberry Pi 3 e ESP8266.....	71



## LISTA DE QUADROS

Quadro 1 - Bibliotecas utilizadas para comunicação entre sensores e a placa.....	51
Quadro 2 - Área de inicialização de parâmetros <i>void setup()</i> .....	52
Quadro 3 - Área de desenvolvimento do código <i>void loop()</i> .....	52
Quadro 4 - Inserção de bibliotecas em <i>Python</i> .....	53
Quadro 5 - Desenvolvimento de aplicações em <i>Python</i> .....	54

## LISTA DE ABREVIATURAS E SIGLAS

AD	Analógico Digital
$\mu$ A	<i>Microampère</i>
6LoWPAN	<i>IPv6 over Low power Wireless Personal Area Networks</i>
API	<i>Application Program Interface</i>
ARM	<i>Advanced Risc Machine</i>
BLE	<i>Bluetooth Low Energy</i>
CPU	<i>Central Processing Unit</i>
GB	Gigabyte
GHz	Gigahertz
GPIO	<i>General Porpuse Input Output</i>
HDMI	<i>High-Definition Multimedia Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engeneering Task Force</i>
<i>IoT</i>	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
ISM	<i>Industrial Sientific and Medical</i>
kB	<i>Quilobyte</i>
mA	<i>Miliampère</i>
MB	Megabyte
MHz	Megahertz
MM	Milímetros
PWM	<i>Pulse Width Modulation</i>
QSPI	<i>Queued Serial Peripheral Interface</i>
RAM	<i>Random Access Memory</i>
RF	<i>Rádio Frequência</i>

RFID	<i>Radio Frequency Identification</i>
RISC	<i>Reduced Instruction Set Computer</i>
RJ45	Conector Modular Usado em Terminações de Telecomunicação
SD	<i>Secure Digital</i>
SoC	<i>System on a Chip</i>
SPI	<i>Serial Peripheral Interface</i>
TCP	<i>Transmission Control Protocol</i>
UART	<i>Universal asynchronous receiver transmitter</i>
UDP	<i>User Datagram Protocol</i>
USB	<i>Universal Serial Bus</i>
V	<i>Volts</i>
WEP	<i>Wired Equivalent Privacy</i>
Wi-Fi	Tecnologia de Comunicação que Não Faz Uso de Cabos
WLAN	<i>Wireless Local Area Network</i>
WPA2	<i>Wi-Fi Protected Access</i>
WPA2-PSK	<i>Temporal Key Integrity Protocol - Pre-Shared Key V2</i>
WPA-PSK	<i>Temporal Key Integrity Protocol - Pre-Shared Key</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>16</b>
2.1 INTERNET DAS COISAS .....	16
2.2 PLATAFORMAS .....	19
2.3 REDES SEM FIO .....	28
2.4 SENSORES E ATUADORES .....	32
<b>3 TRABALHOS CORRELATOS .....</b>	<b>40</b>
3.1 TRABALHO 1: INTERNET DAS COISAS: UM PROTÓTIPO USANDO A PLATAFORMA DE PROTOTIPAGEM ARDUINO E A PLACA E-HEALTH PARA A COLETA DE SINAIS VITAIS .....	40
3.2 TRABALHO 2: SISTEMA AUTOMATIZADO PARA CONTROLE DE UMIDADE E TEMPERATURA EM CULTURA DE MORANGOS APLICADOS AOS PEQUENOS PRODUTORES ....	41
3.3 TRABALHO 3: PROJETO DE UM SISTEMA REMOTO PARA AQUISIÇÃO DE DADOS METEOROLÓGICOS .....	41
<b>4 PROCEDIMENTO METODOLÓGICO .....</b>	<b>43</b>
<b>5 DESENVOLVIMENTO DO PROTÓTIPO E DA APLICAÇÃO .....</b>	<b>45</b>
5.1 DESCRIÇÃO DO PROTÓTIPO .....	45
5.2 CENÁRIOS .....	46
5.4 DESENVOLVIMENTO DAS APLICAÇÕES .....	50
5.5 VALIDAÇÃO DOS SENSORES .....	59
<b>6 EXPERIMENTO E ANÁLISE DOS RESULTADOS .....</b>	<b>61</b>
6.1 VALIDAÇÃO DO CONSUMO ELÉTRICO .....	62
6.2 UTILIZAÇÃO DE BATERIA .....	66
6.3 AVALIAÇÃO DOS RESULTADOS .....	67
<b>CONSIDERAÇÕES FINAIS .....</b>	<b>72</b>
<b>BIBLIOGRAFIA .....</b>	<b>74</b>

## 1 INTRODUÇÃO

A intercomunicação autônoma de objetos, ou dispositivos, em rede, sem a destinação final dos dados a humanos, tem crescido a ponto de surgir uma definição específica para este tema: Internet das Coisas (do inglês, *Internet of Things*) ou simplesmente *IoT*. A *IoT* vem sendo uma evolução tecnológica em diversas áreas revolucionando a própria internet com a conexão de outros equipamentos que não são operados por pessoas, mas sim por elas mesmas ou “coisas” (MANCINI, 2017).

Oliveira (2017, p. 17) descreve da seguinte forma a Internet das Coisas: “Não é somente ligar as “coisas” pela internet, mas também as tornar inteligentes e capazes de coletar e processar informações do ambiente ou das redes às quais estão conectadas.”. Muitos projetos e estudos estão sendo desenvolvidos no intuito de criar cidades inteligentes, carros autônomos, casas inteligentes, segurança pública, entre outros, em que tudo esteja conectado e se comunicando sem interferência externa.

Estudos têm apontado um crescimento explosivo dos dispositivos conectados na rede como apontado no trabalho apresentado por Evans (2011): em 2003, 500 milhões de dispositivos contra uma população de 6,3 bilhões, já para 2020 a projeção chega a 50 bilhões de dispositivos contra uma população de 7,6 bilhões.

Neste trabalho será apresentada uma aplicação *IoT* de monitoramento de uma planta, utilizando duas plataformas das mais populares, de baixo custo e de fácil implementação: ESP8266 e Raspberry Pi. Sendo estas, duas plataformas de prototipagem que possuem características semelhantes, proporcionando um estudo comparativo adequado para este projeto.

A plataforma ESP8266 possui tamanho reduzido e diversos recursos incorporados, como wireless por exemplo, e tem tido grande aceitação de mercado (OLIVEIRA, 2017). Para a criação de aplicações de forma simplificada com o ESP8266, pode-se utilizar a Arduino *Integrated Development Environment*, mais conhecida como *Arduino IDE*, desenvolvida pelos criadores da plataforma Arduino (ARDUINO IDE, 2017). A linguagem de programação utilizada é a linguagem C++. Embutidas na *Arduino IDE*, existem algumas bibliotecas para auxiliar nos passos iniciais (ARDUINO SOFTWARE IDE, 2017).

Com uma proposta um pouco diferente da anterior, não sendo micro controlado e sim micro processado, a plataforma de desenvolvimento Raspberry Pi possui funções semelhantes às do ESP8266, como portas de comunicação para realizar as conexões de entrada e saída,

porém com um custo superior. O projeto Raspberry Pi foi desenvolvido no Reino Unido pela fundação de mesmo nome. Ele é na verdade, um minicomputador integrado em uma única placa do tamanho de um cartão de crédito, com *hardware* aberto foi desenvolvido para auxiliar o aprendizado de jovens na manipulação de *hardware* e *software* (FIGUEIRÊDO, 2017).

Portando, seu poder computacional é superior quando comparado a plataforma de prototipagem ESP8266, tendo embarcados nele maior quantidade de periféricos, porém como desvantagem seu consumo é relativamente maior na comparação básica entre as duas placas.

Para realizar a programação de aplicações no Raspberry Pi, utiliza-se sua linguagem nativa de programação *Python*, uma linguagem de *script* orientada à produtividade de alto nível, vindo de encontro à proposta do criador da plataforma, além da vantagem de possuir como característica ser de código aberto.

Para que seja possível monitorar e entender o que ocorre com a planta, será necessário utilizar sensores que segundo a definição de Wendling (2010), são dispositivos sensíveis ao ambiente que realizam leitura de grandezas físicas como temperatura, luminosidade, humidade, pressão, etc. Basicamente, existem dois tipos de sensores: analógicos e digitais. Os sensores analógicos cujos valores de sinal variam ao longo do tempo dentro de uma faixa de operação, como por exemplo, a variação da temperatura. Já os sensores digitais compreendem apenas dois níveis lógicos em sua saída, sendo nível alto ou nível baixo ao tempo.

Depois de monitorar o meio em que a planta está, pode-se também controlar a falta de umidade no solo acionando atuadores e desta forma acionar uma saída atuando na umidificação do solo. Neste trabalho somente será desenvolvido o monitoramento, sem acionamento de atuadores.

De acordo com Oliveira (2017), é de extrema importância que os dispositivos a serem utilizados em projetos para Internet das Coisas sejam de baixo consumo, pois, do contrário, inviabilizam a implementação em cenários remotos ou mesmo a recarga e substituição das fontes de energia. Portanto, deve-se conhecer o consumo de cada plataforma de prototipagem quando conectadas na rede elétrica e também a autonomia quando alimentadas por baterias.

Dentro deste contexto, este trabalho tem por objetivo desenvolver um estudo comparativo entre dois cenários idênticos utilizando as plataformas Raspberry Pi e ESP8266. Em cada cenário serão coletadas as informações de temperatura e umidade ambiente, umidade do solo, luminosidade do ambiente e enviados para dois sites repositórios online *Thingspeak* e

*Google Apps Spreadsheet*, além do consumo de energia entre as duas plataformas, assim implementando o monitoramento das condições de uma planta.

O presente trabalho está estruturado em seis capítulos. O capítulo 1, destinado à introdução do trabalho, seus objetivos e à forma como será organizado. Já o capítulo 2 apresenta o referencial teórico do trabalho, abordando o tema Internet das Coisas, as plataformas Raspberry Pi e ESP8266. Ainda neste capítulo serão apresentados sensores e alguns exemplos de uso, e ao final apresenta-se o conceito de atuadores. No capítulo 3, serão apresentados trabalhos correlatos em três áreas onde a Internet das Coisas e as plataformas de prototipagem podem ser utilizadas. No capítulo 4, serão detalhados os objetivos do projeto incluindo a metodologia. Já o capítulo 5, irá apresentar o desenvolvimento do protótipo e os cenários para utilização das plataformas. Além disso, abrangerá o desenvolvimento da aplicação para cada plataforma, os sites repositórios online e a validação dos sensores. O capítulo 6, apresenta o experimento os dados do consumo elétrico de cada plataforma de prototipagem e um comparativo entre elas. Ainda neste capítulo, será apresentada a validação dos resultados obtidos para cada cenário. Por fim, as considerações finais farão o desfecho do trabalho com a conclusão do que foi exposto e apresentará algumas possibilidades de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Neste capítulo, serão detalhados os conceitos de Internet das Coisas, as plataformas Raspberry Pi e ESP8266, as redes sem fios utilizadas nos protótipos, e as definições de dispositivos sensores e atuadores.

### 2.1 INTERNET DAS COISAS

Ao longo dos últimos anos, a Internet das coisas ou *Internet of Things (IoT)* tornou-se um termo universal. A *IoT* é uma revolução tecnológica nas tecnologias da informação e comunicação (TIC) que expande os conceitos já comuns de “qualquer momento” e “qualquer lugar” para a conectividade para “qualquer coisa”. Esta tecnologia está crescendo rapidamente e se tornando uma realidade de acordo com vários analistas do setor e empresas de TIC (ZANELLA et al., 2014).

Na verdade, *IoT* já se tornou uma realidade, pelo menos com base em números. Em algum momento, entre 2008 e 2009, o número de dispositivos conectados à Internet superou a população mundial (Evans, 2011), que pode ser usado como indicador do surgimento do *IoT*. Prevê-se que, até 2020, haverá de 26 a 50 bilhões de coisas conectadas ao *IoT* (Evans, 2011), um número que é uma ordem de grandeza maior que a de todos os hosts atuais da Internet, incluindo conexões de telefones inteligentes, conforme Figura 1.

*IoT* não é um conceito facilmente compreendido e existem várias definições:

- Na pesquisa acadêmica, propõe-se a seguinte definição (OLIVEIRA, et al., 2016): “A *IoT* é um conceito inovador baseado na presença generalizada de uma variedade de coisas ou objetos - como tags RFID, sensores, atuadores, telefones celulares - que, através de esquemas de endereçamento únicos, são capazes de interagir uns com os outros e cooperar com seus vizinhos para alcançar metas comuns “.

- Mancini (2017, p. 1), define a *IoT* como “objetos inteligentes sendo possível detectar seu contexto, controlá-lo, viabilizar a troca de informações uns com os outros, acessar serviços de internet e interagir com as pessoas.”.

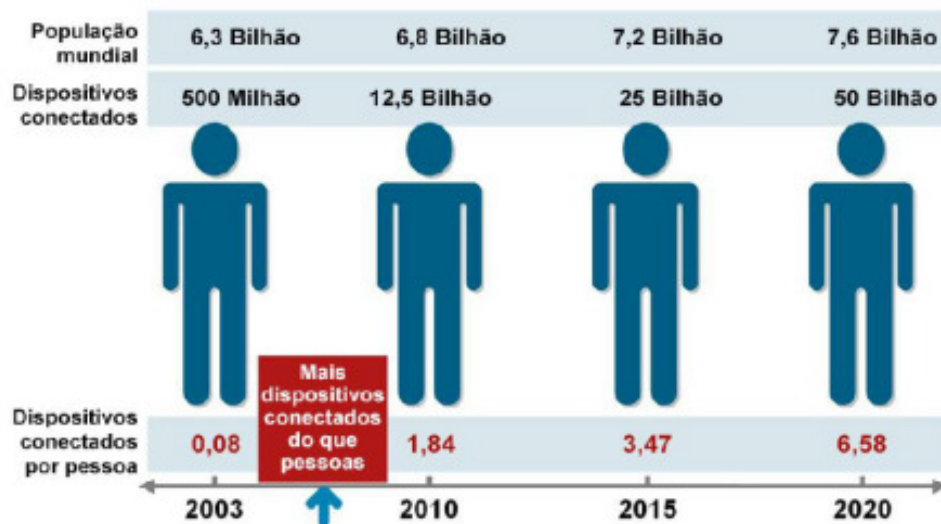
- Paulo et al.(2015, p. 2) descreve a *IoT*:

[..] é um paradigma que preconiza um mundo de objetos físicos embarcados com sensores e atuadores, conectados por redes sem fio e que se comunicam usando a Internet, moldando uma rede de objetos inteligentes capazes de realizar variados processamentos, capturar variáveis ambientais e reagir a estímulos externos.



Através da capacidade de identificação, captura de dados, processamento e comunicação, o *IoT* faz uso pleno de “coisas” para oferecer serviços a todos os tipos de aplicativos. As coisas são objetos capazes de serem identificados e integrados nas redes de comunicação. As coisas têm informações associadas, que podem ser estáticas ou dinâmicas.

Figura 1 - Evolução da Internet das Coisas



Fonte: EVANS (2011)

### 2.1.1 Contextualização de *IoT*

Quando ainda não se pensava no termo internet das coisas, o congresso *Interop '89 Conference* apresentou o que se pode chamar de primeiro dispositivo para *IoT*. Nada mais era que uma torradeira que podia ser ligada e desligada pela internet. Seu criador John Romkey realizou a conexão através de um computador com rede TCP / IP e fez muito sucesso.

Porém, o termo Internet das Coisas foi utilizado pela primeira vez por Kevin Ashton, em 1999, do *Massachusetts Institute Technology* ao realizar uma palestra na empresa *Procter & Gamble* e apresentou uma ideia de rastreamento de produtos por RFID. Para Ashton os produtos do mundo físico poderiam se conectar à internet. Sendo assim, considerado o marco para a Internet Das Coisas.

Atualmente, a *IoT* está em crescimento acentuadamente e existem diversas estimativas nesse mercado apontando que até 2025 o setor poderá faturar até US\$ 11 trilhões. De olho nessa oportunidade estão empresas como a IBM, Intel, Microsoft e outras empresas do setor de tecnologia fazendo parcerias e investimentos pesados nesta área.

Muitas empresas estão desenvolvendo soluções para conectar diferentes equipamentos a internet, assim estão gerando milhares de dados e guardando tudo em nuvem, gerando uma grande quantidade de *Big Data*, com isso as empresas esperam melhorar seus processos de manufatura. Dados indicam que 75% dos empresários estão procurando oportunidades nesta área.

Conforme estudos realizados utilizando como base o cruzamento de informações sobre o número de dispositivos conectados à internet, foi possível estabelecer, que a taxa de crescimento da internet dobra a cada 5,32 anos. Com cada vez mais dispositivos conectados, já existem problemas a serem enfrentados, como por exemplo a escassez de endereços IP, desta forma, necessitando acelerar o processo de troca do padrão de endereços IPV4 para IPV6.

### 2.1.2 Aplicações de *IoT*

As aplicações da *IoT* são muitas, em diversas áreas e estão realizando uma transformação tecnológica que tem como objetivo conectar os equipamentos usados no dia a dia à rede mundial de computadores. Mancini (2017) cita alguns setores, conforme Figura 2:

- Bens de consumo: Bens adquiridos pelos consumidores, tais como *smartphones*, *smart house*, *smart car* e *smart TV*;
- *eHealth*: *Fitness*, bioeletrônica e cuidados com a saúde, monitoramento de sinais vitais;
- Casas inteligentes: Medições remotas de consumo, economia de energia, segurança residencial, controle inteligente de equipamentos residenciais;
- Gestão da agricultura e dos recursos naturais: Segurança e rastreabilidade de produtos agrícolas, gerenciamento da qualidade, monitoramento ambiental;

Figura 2 - Internet das Coisas



Fonte: PubNub (2014)

## 2.2 PLATAFORMAS

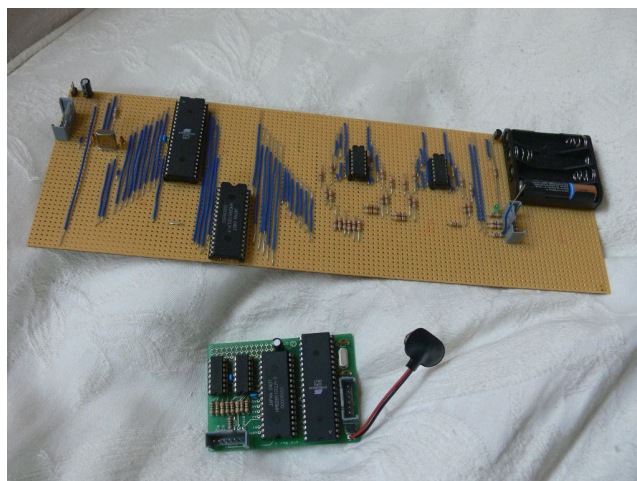
Com a popularização da *IoT* diversas empresas iniciaram projetos de desenvolvimento de placas voltadas para este mercado. Conforme mencionado anteriormente a tendência para os próximos anos é de crescimento acentuado neste setor. A seguir são apresentadas duas das plataformas mais populares: Raspberry Pi e ESP8266.

### 2.2.1 Raspberry Pi

O Raspberry Pi é uma plataforma de desenvolvimento de baixo custo, com dimensões muito reduzidas e o tamanho aproximado de um cartão de crédito. Mesmo sendo tão pequena possui recursos comparáveis a alguns computadores *desktops*, bastando apenas conectar um teclado, mouse e um monitor de computador ou uma televisão (RASPBERRY PI *FAQS*, 2017).

O desenvolvimento inicial da placa começou em 2006 com o micro controlador ATmega644 da Atmel disponível na época dos primeiros protótipos muito rudimentares comparados ao projeto atual (CIRIACO, s.d.). Conforme Upton (2013) as primeiras versões foram montadas na mesa da cozinha de sua casa com os maiores *protoboards* disponíveis, compradas em lojas de eletrônicos conforme a **Erro! A origem da referência não foi encontrada.**

**Figura 3 - Protótipo inicial com ATmega644**



Fonte: canaltech.com.br (2017)

Com a evolução dos dispositivos móveis, foram disponibilizados no mercado novos processadores, menores, mais completos e mais potentes que atendiam o desejo por uma plataforma de computação completa. O projeto seguiu com diversos protótipos até que em 2012 foi lançado o primeiro do Raspberry Pi conforme a **Erro! A origem da referência não foi encontrada.**<sup>4</sup>, no Reino Unido pela fundação Raspberry Pi.

**Figura 4 - Raspberry Pi**



**Fonte: Raspberrypi.org (2017)**

Sendo apresentado como uma solução de computador popular de baixo custo, US\$ 25,00 versão básica ou US\$ 35,00 na versão com rede, para a integração e aprendizado de programação a crianças nas escolas. Assim, cada criança poderia utilizar a sua placa já configurada e pronta para uso com algumas ferramentas e outros materiais disponíveis para que se pudesse praticar em casa. O objetivo foi despertar o interesse na manipulação de componentes internos de computadores pelos jovens, o que já não estava mais ocorrendo, visto que a maioria das arquiteturas atualmente disponíveis no mercado são fechadas e prioritárias (MORIMOTO, 2012).

Como dito anteriormente, o objetivo foi produzir uma placa para estudo nas escolas e possíveis *hobistas* no Reino Unido, mas o projeto cresceu tanto e atraiu a atenção da comunidade em geral que conforme descreve Upton (2013, p. 20): “Haviam 100 mil pessoas em nossa lista de correspondência querendo um Raspberry Pi...”. Assim, o que antes era um projeto criado em casa, com o propósito específico, passou a ser uma plataforma com o interesse dos mais variados setores.

Com a proposta de ter uma arquitetura de software aberta a escolha do sistema operacional utilizado foi pelo Linux, que além de ser mais leve também pode ser

customizado. A distribuição recomendada oficialmente pelo fabricante é a *Raspbian*, tendo como base a distribuição Debian. Esta, que foi modificada especialmente no intuito de aproveitar todos os recursos de hardware disponíveis na placa, tendo uma comunidade trabalhando constantemente em atualizações. Contudo, outras distribuições como *Arch Linux*, *Fedora*, *Ubuntu MATE* também podem ser utilizadas, pois a placa é vendida sem sistema operacional (RASPBerry PI FAQs, 2017).

Diferente dos computadores tradicionais o Raspberry Pi possui seu sistema instalado em um cartão SD, sendo utilizados estes cartões por serem baratos e fáceis de encontrar. Necessitando no mínimo 8Gb para o sistema e armazenamento, podem desta forma fornecer portabilidade para qualquer placa. Para facilitar o processo, existem imagens disponíveis na internet bastando realizar o download no site <http://www.raspberrypi.org/> para utilizar a imagem já compilada para a arquitetura ARM (*Advanced Risc Machine*). Este método permite de forma simples restaurar o sistema operacional caso o sistema seja corrompido, simplicidade é uma característica muito importante em um dispositivo educacional (RASPBerry PI FAQs, 2017).

De acordo com Oliveira (2017), Linux, como sistema operacional possui incorporado a ele diversos pacotes de softwares, entre eles podemos citar servidores de bancos de dados, servidores de protocolos de rede, interfaces gráficas e *API's* para integração aos mais variados dispositivos e inúmeras linguagens de programação como, por exemplo, C, C++, *Java*, *Scratch Ruby* e *Python*.

A escolha da linguagem de programação nativa para o desenvolvimento do projeto Raspberry Pi foi a *Python* devido à simplicidade de uso, sintaxe e acessibilidade. Desenvolvida como linguagem de alto nível e de código aberto Python é robusta e permite a portabilidade, pois os arquivos fontes gerados ou *scripts* não são compilados e sim interpretados, possuindo ainda uma grande documentação (BRITO, s.d.). Portanto, sendo perfeita ao propósito de aprendizagem para iniciantes e curiosos.

O hardware da plataforma segue o mesmo princípio do software, ou seja, aberto e livre. Para tanto, foi utilizada uma CPU BCM2835 com a tecnologia *system on a chip* (SoC) que integra diversos recursos em um mesmo *chip* da empresa Broadcom, mantendo assim os custos reduzidos. Com esta abordagem foi possível reduzir as dimensões da placa, tendo integrado os seguintes componentes (LIMA, 2014):

- Processador ARM de 700 MHz;
- GPU VideoCore IV;

- 256 MB de memória RAM (SoC);
- Entrada de cartão SD;
- Duas portas USB 2.0;
- Saída de vídeo HDMI e vídeo composto;
- Ethernet 10/100 (RJ45);
- Conector de 26 pinos das quais 17 são GPIO;
- Saída de áudio via conector jack de 3,5mm.

Atualmente, a versão mais recente da placa é a Raspberry Pi 3 que conta com o novo processador BCM2837, dotado de 4 processadores de 64 bits ARM Cortex-A53 trabalhando a 1.2 GHz. Foram mantidas, a memória com 1 GB e a GPU VideoCore IV, porém trabalhando com 400MHz, incrementando assim o desempenho em cerca de 50 % em relação à versão anterior (SOUZA, 2016).

Novos recursos voltados a *IoT* foram solicitados por Eben Upton a Broadcom, neste novo chip BCM43438 desenvolvido especialmente para o projeto. As novidades foram a inclusão dos padrões de comunicação *Wireless* e Bluetooth 4.1 eliminando assim componentes adicionais para redes sem fio. Outra grande novidade é o *Bluetooth Low Energy* ou BLE padrão relativamente novo voltado ao uso em componentes para *IoT*, que utilizam baixo consumo de energia a baixa potência e velocidade de transmissão (SOUZA, 2016).

Com mais componentes conectados na placa, a consequência foi o aumento no seu consumo exigindo uma fonte de alimentação com capacidade necessária para fornecer no mínimo 2,5 A para a versão mais nova, ante 1.8 A da versão anterior (RASPBERRY PI *FAQS*, 2017).

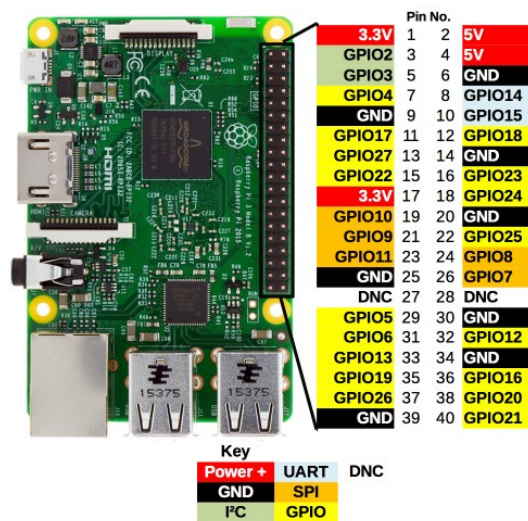
### 2.2.1.1 GPIO

Diferente de um computador normal, a placa possui integrada uma porta para comunicação. Nos casos em que se faz necessária a comunicação entre a Raspberry Pi e sistemas externos está disponível a porta de entrada e saída de uso geral ou GPIO (*General Purpose Input Output*), com 40 pinos, nos modelos mais completos conforme ilustrada na **Erro! A origem da referência não foi encontrada.5**.

A porta GPIO permite que sejam realizadas comunicações com outros equipamentos ou circuitos além de atuar como controlador em circuitos maiores. Outra importante função

desta porta é a leitura de sensores ou sinais ou o acionamento de atuadores para execução de alguma tarefa definida através do programa. Os estados das portas digitais podem ser configurados através da programação como entradas ou saídas. Além disso, as portas podem ser usadas para comunicar com outros dispositivos através dos protocolos SPI (*Serial Peripheral Interface* ou Interface Periférica Central), I2C (*Inter-Integrated Circuit* ou Circuito Inter-Integrado) e UART (*Universal Asynchronous Receiver / Transmitter* ou Receptor/Transmissor Universal) (Upton, 2013).

**Figura 5 - GPIO Raspberry Pi 3**



Fonte: <http://osoyoo.com> (2017)

Conforme Upton (2013), a tensão de trabalho nos terminais para a lógica é de 3,3 V, caso esse valor seja excedido o processador será seriamente danificado ou inutilizado. Por mais que exista um pino com alimentação de 5 V, sua finalidade é a de alimentar outras placas ou sensores, não podendo alimentar nenhum pino da GPIO necessitando de uma interface para o nível de tensão 3,3 V.

Deste modo, o consumo dos diferentes componentes do Raspberry Pi, sejam internos ou externos, precisam respeitar o limite de potência impostos a eles com risco de superaquecer o processador danificando a placa. Os valores máximos suportados podem variar dependendo do modelo de placa utilizada. Como o conjunto de pinos da porta GPIO suporta no máximo 50 mA, sendo 16 mA o máximo que se pode carregar por pino no Raspberry Pi (RASPBERRY PI FAQs, 2017).

### 2.2.2 ESP8266

Produzido pela empresa chinesa Espressif, o micro controlador ESP8266, surgiu como alternativa de baixo custo para complementar aplicações para *IoT*. Compartilhando a mesma construção SoC como em outras plataformas, com o seu tamanho reduzido tem tido grande aceitação no mercado em função de diversos recursos incorporados, como, wireless, protocolos de comunicação, portas de entrada e saída, além do baixo consumo (OLIVEIRA, 2017).

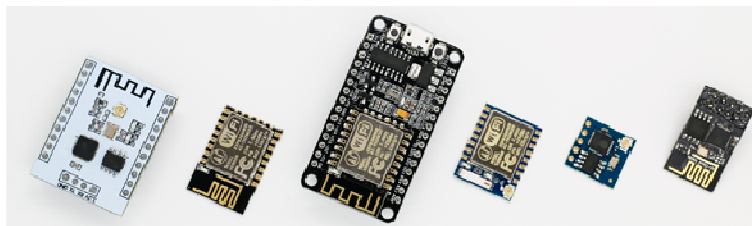
Logo após seu lançamento, em virtude de possuir pouca documentação em inglês, não estava sendo muito utilizado. Para Benchoff (2014), o módulo ESP8266 possuía grande potencial, mas necessitava de divulgação e incentivo para a criação de uma comunidade de adeptos. Na comparação com outras placas, o preço de US \$5 e a possibilidade de se conectar à internet com apenas uma placa diminuta, tornava-se uma boa solução para a criação de projetos nas áreas de automação residencial, rede de sensores, robótica, monitoramento remoto e outras (BENCHOFF, 2014).

O ESP8266, considerado uma alternativa disponível no mercado para auxiliar na tarefa de conexão entre os dispositivos, acabou sendo muito utilizado para comunicação digital, fazendo uso de um micro controlador capaz de coletar, tratar e realizar a transmissão dos dados em um hardware enxuto e barato. O módulo ESP8266 pode realizar sua comunicação de duas formas, via cabos através de interfaces seriais, ou sem cabos utilizando o padrão *Wi-Fi*.

Este controlador está disponível em diferentes versões, chamadas de módulos, conforme podem ser vistos na **Erro! A origem da referência não foi encontrada.**6, com características distintas, como, dimensões, variação do número de portas GPIO ou mesmo a inexistência delas. Muitas vezes, o seu uso se dá apenas como interface *wireless* para a conexão à internet de outros componentes, porém como base, sempre será utilizado o micro controlador ESP8266 (OLIVEIRA, 2017).

**Figura 6 - Versões disponíveis do módulo ESP8266**





Fonte: THOMSEN (2016)

Quanto aos recursos, este micro controlador é bastante completo e possui diversas características não encontradas, por exemplo, na plataforma Arduino. Seu processador, quantidade de memória e possibilidade de conexão via *Wi-Fi* o qualificam em alguns casos como substituto ao Arduino, quando na configuração NodeMCU. Abaixo estão listados os principais recursos e características do ESP8266 (OLIVEIRA, 2017):

- Processador: 32-bit RISC Tensilica Xtensa LX106 rodando a 80 / 160 MHz;
- Memória RAM: 64 kB instruções, 96 kB de dados;
- Memória Flash: QSPI Externo de 512 kB a 4 MB;
- Conexão às redes: 802.11 B/G/N;
- Alcance aproximado: 91 metros;
- Tensão de operação: 3.3 Vdc;
- Comunicação: SPI, I2C, I2S, UART, *Wi-Fi*;
- Modos de operação: Cliente, Access Point, Cliente + Access Point;
- Modos de segurança wireless: OPEN/WEP/WPA\_PSK/WPA2\_PSK/WPA\_WPA2\_PSK;
- Suporta comunicação TCP e UDP, com até 5 conexões simultâneas;
- Porta GPIO (dependendo do modelo);
- Consumo: 20  $\mu$ A modo *sleep*, 50 mA conectado *Wi-Fi* recepção e até 170 mA modo transmissão.

Por fim, para o desenvolvimento de aplicações, os dois ambientes de desenvolvimento mais utilizados são Lua e Arduino *Integrated Development Environment* (Arduino *IDE*). A linguagem Lua, utiliza uma linguagem interpretada e nativa para a placa, é muito rápida e útil para aplicações específicas. Outro ambiente que pode ser utilizado é o Arduino *IDE*, completamente compatível com a placa e conta com grande base de aplicações desenvolvidas para o Arduino que podem ser utilizadas tornando assim rápido e fácil o seu desenvolvimento (OLIVEIRA, 2017).

Neste trabalho o desenvolvimento de aplicações foi realizado no Arduino *IDE* (ARDUINO IDE, 2017). Esta IDE foi especificamente criada para o Arduino, mas é amplamente utilizado em várias outras plataformas. Nele estão integradas diversas bibliotecas para auxiliar com componentes que podem ser conectados a placa, opções para a programação de diferentes modelos de placas entre outras (ARDUINO SOFTWARE IDE, 2017).

A linguagem de programação utilizada é a linguagem C/C++, porém foi escrita em Java e utiliza os conceitos de *Processing* e *Wiring* voltadas a esquematização com uma visão que facilita a escrita dos códigos por pessoas leigas. A interface é muito simplificada e intuitiva com poucos botões, sendo um deles utilizado conferir se o código está escrito corretamente e o outro para a programação direta no hardware da plataforma sem a necessidade de maiores configurações. (ARDUINO SOFTWARE IDE, 2017).

Qualquer aplicação criada para a placa, chamado de *sketch*, terão no mínimo duas funções principais para programação: *setup()* local onde será feita a configuração do dispositivo e *loop()* onde será executado o código compilado de forma repetida enquanto placa estiver energizada. Ao final, a transferência do arquivo entre a *IDE* do computador e a placa é feita utilizando-se um cabo USB. Para isso a placa possui um conversor USB para serial padrão RS 232, que após compilar e gravar o programa na placa pode rodar de forma independente alimentado apenas por uma fonte (ARDUINO SOFTWARE IDE, 2017).

#### 2.2.2.1 Pinagem

Com a diversidade de projetos que podem ser criados utilizando a plataforma ESP8266, os desenvolvedores independentes criaram uma série de modelos explorando as características baseadas em seu *chip*. Com a utilização de um número maior de pinos o modelo mais popular chamado NodeMCU ESP8266-12E possui 16 GPIO divididos entre funções de uso geral e usos específico, conforme ilustrado na **Erro! A origem da referência não foi encontrada.** é possível verificar as funções de cada pino. Porém, dependendo da necessidade do projeto, é possível encontrar placas com número menor de pinos (MANITEL, 2016).

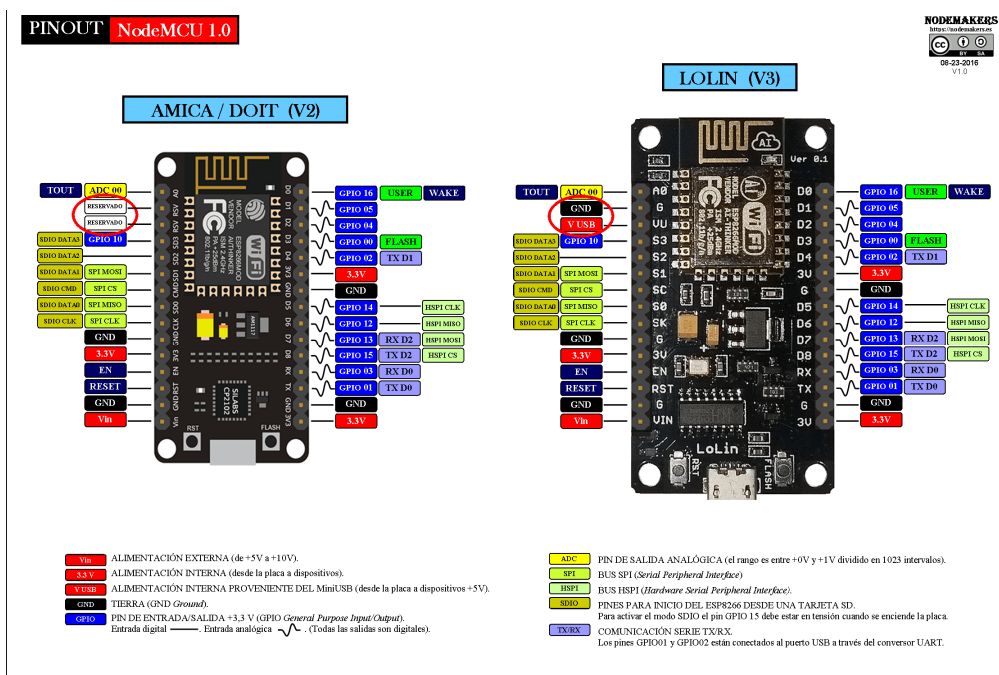
Manitel (2016), detalha o módulo NodeMCU que possui 16 pinos multiuso de entrada e saída, sendo 11 pinos digitais que podem ser utilizados para uso que operam com tensão de 3.3 V não podendo ultrapassar a corrente de 12 mA em cada pino. Esses pinos podem

proporcionar outras funções, como controle de motores, chamados PWM, comunicação serial, sendo os seguintes protocolos: UART, SPI e I2C, além de gerar interrupções externas.

O módulo disponibiliza para leitura de sinais analógicos 1 porta, que pode ser utilizada para medir a umidade do solo por exemplo. O nível de tensão suportado é de 1 V e a resolução de leitura é de 10 bits, podendo realizar leituras entre 0 até 1023 e os dados são processados através de um conversor AD (analógico/digital) utilizando o pino A0 como referência (OLIVEIRA, s.d.).

Para Manitel (2016), existem ainda existe a possibilidade de queima caso não seja respeitado o nível de tensão em 3.3 V. Por mais que exista literatura informando a possibilidade alimentação em 5 V, isto não é recomendado.

Figura 7 - Pinagem ESP8266



**NodeMCU ESP8266 (2018)**

Aqui foram apresentadas e detalhadas as propriedades das duas plataformas, Raspberry Pi e ESP8266. Conforme visto, as placas diferem internamente em seu funcionamento e quantidade de recursos, porém através das portas de comunicação podem até mesmo trabalhar em conjunto conforme apresentado por Thomsen (2015).

A Tabela 1 apresenta um resumo com as principais características das duas plataformas

Tabela 1 - Comparação entre as plataformas

<b>Características</b>	<b>Raspberry Pi</b>	<b>ESP8266</b>
Modelo	Raspberry Pi 3 Modelo B	ESP8266-12E
Processador	Broadcom BCM2837 1.2 GHz 64 bits Quad-Core ARM Cortex-A53	Tensilica Xtensa LX106 80 / 160 MHz 32 bits RISC
Memória	1GB SDRAM 400MHz	64 kB instruções 96 kB de dados QSPI 512 kB a 4 MB
Portas	4 Portas USB 2.0 Conector de Vídeo HDMI Conetor <i>Ethernet</i> Conector Áudio-Vídeo Slot para cartão micro SD 40 pinos GPIO Conector para câmera Conector para Monitor	Varia conforme o modelo
<i>Wi-Fi</i>	802.11n	802.11 B/G/N
<i>Bluetooth</i>	4.1 com BLE ( <i>Bluetooth Low Energy</i> )	Não possui
Dimensões	85,6 x 56 mm	25 x 14 mm
Consumo	750mW	Até 170 mW
Preço	US\$ 35,00	US\$ 5,00 ESP-01

Fonte: O autor (2017)

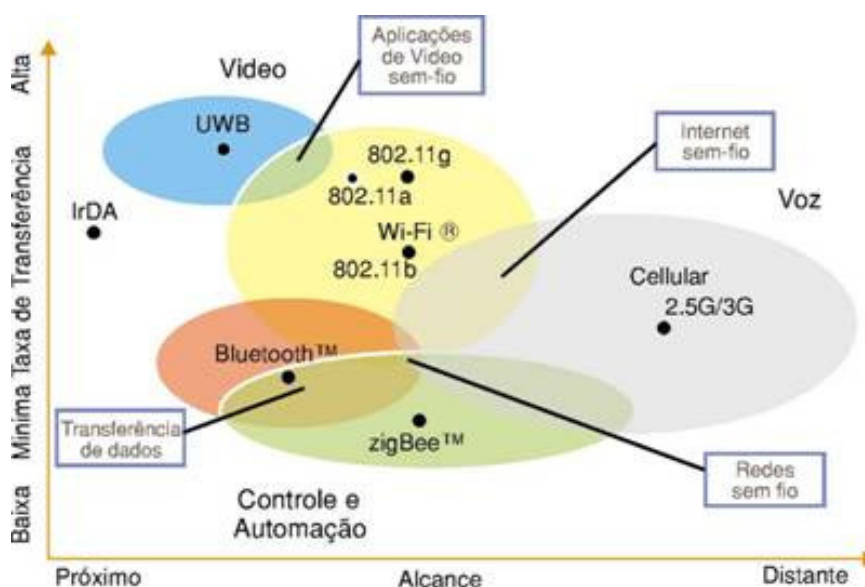
### 2.3 REDES SEM FIO

Com a popularização da informática em todo o globo, as redes para conexão com a internet necessitaram expandir-se para abranger todos os usuários, sejam corporativos ou domésticos. Neste contexto, houve a necessidade da evolução de redes para os novos dispositivos, agora com tecnologia para mobilidade como telefones celulares, *notebooks* e equipamentos portáteis (PINHEIRO, 2004).

Conforme Nério (2003), as redes cabeadas não atendiam mais esta demanda por mobilidade aos novos equipamentos, surgindo então, as redes sem fio ou *wireless* em inglês. As redes sem fio, passaram a ser utilizadas em diversas áreas como, telecomunicações, serviços de dados, acionamento e controle de equipamentos e a conectividade entre periféricos. Deste modo, tornando-se uma alternativa complementar as redes convencionais, tendo como diferença apenas o meio físico como são propagadas, ou seja, ar ou espaço livre.

Diante da demanda crescente por conexões sem fio, foram desenvolvidos diversos protocolos novos para redes específicas. Assim, houve a necessidade de padronização dos protocolos de acesso ao meio físico sem fio para permitir a interoperabilidade entre fabricantes. O órgão responsável pela padronização é o IEEE (*Institute of Eletrical and Electronic Engineers*). Os protocolos de redes wireless operam na faixa ISM (*Industrial Sientificand Medical*) (PINHEIRO,2004). Conforme **Erro! A origem da referência não foi encontrada.**<sup>8</sup> é possível verificar a taxa de transferência e o alcance de algumas redes.

**Figura 8 - Alcance vs Taxa de transferência de redes sem fio**



Fonte: SABERELETRONICA (2011)

### 2.3.1 Rede Wi-Fi

*Wi-Fi* é um conjunto de especificações para redes locais sem fio (WLAN - *Wireless Local Área Network*) baseada no padrão IEEE 802.11. Com a tecnologia *Wi-Fi*, é possível montar redes que conectem não apenas computadores, mas também outros aparelhos

compatíveis (*smartphones, tablets*, impressoras, consoles de videogame, etc) que estejam em um raio próximo.

A utilização desta representação oferece diversas vantagens aos usuários, como liberdade de locomoção dentro do limite de alcance e a facilidade para a adição de novos dispositivos na rede. Desta forma, a estrutura física onde a rede se encontra não necessita de modificações, porém merecendo atenção a localização do ponto de acesso que preferencialmente deve estar ao centro do raio de utilização.

A flexibilidade do *Wi-Fi* tornou viável a implementação de projetos para redes com esta tecnologia para diversos fins, como é o caso da *IoT*. Atualmente, dispositivos *Wi-Fi* possuem preços mais atrativos e maior disponibilidade sendo atualmente muito utilizados.

Quando se fala em conexões *wireless*, *Wi-Fi* pode ser considerado uma referência para computadores e outros dispositivos que estão constantemente conectados a uma fonte de energia ou através de baterias com capacidade elevada. Assim, funcionando bem para equipamentos maiores sem restrição de consumo.

Em contrapartida, equipamentos para *IoT* geralmente possuem dimensões reduzidas e autonomia de energia limitada, podendo-se utilizar a rede *Wi-Fi* levando-se em consideração no projeto seus fatores limitantes. Outro ponto a ser considerado, é a quantidade de dados e acessos baixos na comparação com o modo tradicional.

Portanto, é necessário um estudo criterioso considerando não apenas o custo, mas também o benefício da utilização do *Wi-Fi*.

### **2.3.2 Redes de baixo consumo para *IoT***

De acordo com Oliveira (2017), é de extrema importância que os dispositivos a serem utilizados em projetos para Internet das Coisas sejam de baixo consumo pois, do contrário, inviabilizam a implementação em cenários remotos ou mesmo a recarga e substituição das fontes de energia.

Ambientes remotos exigem a alimentação através de baterias ou fontes de energia alternativas como, energia solar, eólica, entre outras. No caso de baterias, o custo pode se tornar proibitivo dependendo do número de dispositivos que necessitam de alimentação.

Deste modo, conforme mencionado no site GTA<sup>1</sup>, hoje, no mercado existem diversas redes de comunicação para os dispositivos sem fio em *IoT*. Porém, cada rede possui suas particularidades e a incompatibilidade entre elas torna-se um grande problema, mesmo que diversos aperfeiçoamentos estão sendo feitos ao longo do tempo, ainda são necessárias soluções de integração entre dispositivos diferentes em uma mesma rede.

Alguns recursos estão sendo implementados para auxiliar a manter baixo o consumo energético, como o uso de técnicas que coloquem os dispositivos em modo de hibernação ou *sleep* e “despertando” após determinados ciclos de tempo, isto representa uma diferença enorme ao longo do tempo para as fontes de energia como bateria. Abaixo, estão listadas as redes mais difundidas atualmente para *IoT*.

6LoWPAN: de acordo com Olsson (2014), é uma rede de área pessoal de comunicação sem fios de baixa potência que se utiliza do novo padrão IPv6, pois o antigo padrão IPv4 está sendo substituído devido ao esgotamento de endereços IP.

6LoWPAN é um padrão de conexão aberto e certificado pelo *Internet Engineering Task Force* (IETF), entidade que faz a certificação de diversos padrões de internet, como UDP, TCP e HTTP. Para que seja possível essa comunicação, o padrão faz uso do protocolo IEEE 802.15.4, que utiliza os pacotes IPV6, criando uma nova camada para adaptar os pacotes IP ao novo formato, desta forma, reduzindo o consumo e o tráfego de dados, diferentemente do padrão sem fio tradicional.

Como característica positiva em relação a outras tecnologias, está o fato da camada de adaptação ser nativa ao padrão, não necessitando *gateways* em alguns casos complexos. Sendo assim, haverá redução na potência necessária, além de dispositivos, conhecidos como *hosts*, mais simples e, conseqüentemente, mais baratos.

Originalmente, a faixa de frequências usada é de 2,4Ghz, porém, em atualizações mais recentes, com o protocolo IEEE 802.15.4g, podem ser utilizadas frequências abaixo de 1 Ghz, reduzindo ainda mais o consumo para dispositivos utilizados em *IoT*, que podem operar com taxas de até 250kbps e ainda entrar em modo *sleep*, despertando quando necessário.

ZigBee: segundo Pinheiro (2004), o padrão também faz parte da definição 802.15.4, foi projetado para comunicação sem fio de forma confiável, com baixo consumo de energia e baixas taxas de transmissão para realizar a comunicação entre aplicações de monitoramento e controle.

---

<sup>1</sup> [https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16\\_1/iot/iot.html?](https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16_1/iot/iot.html?)

O ZigBee trabalha na faixa de frequências, não requerendo, assim, licença de funcionamento, suas taxas de transmissão estão entre 10kbps e 115kbps e seu raio de alcance situa-se entre 10 e 100 metros, dependendo do ambiente e obstáculos em seu caminho. Como prerrogativa do baixo consumo de energia, a vida útil de alimentação dos sistemas pode ser de até 6 meses, dependendo da aplicação que se está implementando, sendo usado em conjunto com pilhas por exemplo.

*Bluetooth Low Energy*: de acordo com o site Embarcados (2016), o *Bluetooth Low Energy* (BLE), ou também chamado *Bluetooth Smart*, por mais que possua o mesmo nome, foi concebido para um propósito diferente, tendo um consumo, taxas de transferência e custos menores, mas com um alcance semelhante a versão clássica.

Como seu foco em aplicações que necessitem de baixo consumo de energia e alta vida útil sem manutenções, o projeto prevê que o dispositivo fique o maior tempo em estado de espera ou *IDLE* (*sleep*). Desta forma, o consumo médio está em 1uA e em picos chegando a 15 mA, dependendo da aplicação.

O BLE também utiliza as faixas de rádio ISM liberadas, trabalhando na frequência de 2.4 Ghz. A máxima taxa de dados transmitidos, teoricamente, é de 1 Mbps, mas devido a diversos fatores como restrições em aplicações e rádio fazem com que a velocidade média fique em 10 kbps, sendo um valor relativamente alto para aplicações em *IoT* que utilizam poucos dados.

## 2.4 SENSORES E ATUADORES

Diversos são os eventos que estão ocorrendo ao nosso redor, muitos deles podem ser distinguidos em função dos nossos sentidos. O corpo humano é basicamente uma máquina sensorial que recebe diversos estímulos do ambiente para que possamos interagir com o que ocorre a nossa volta.

Essa interação, com o meio em que se está, muitas vezes não é percebida, porém é constantemente processada pelo corpo, assim gerando reações adversas dependendo do estímulo. Por exemplo, ao cobrir o rosto quando os raios solares atingem a face em um dia ensolarado ou, quando a janela é fechada ao entardecer para evitar o relento. Contudo, não apenas os seres vivos podem perceber o que está ocorrendo ao redor, mas também sistemas



automatizados podem receber sinais de entrada aplicar algum processamento e em seguida agir de alguma forma (KARVINEN; KARVINEN, 2016).

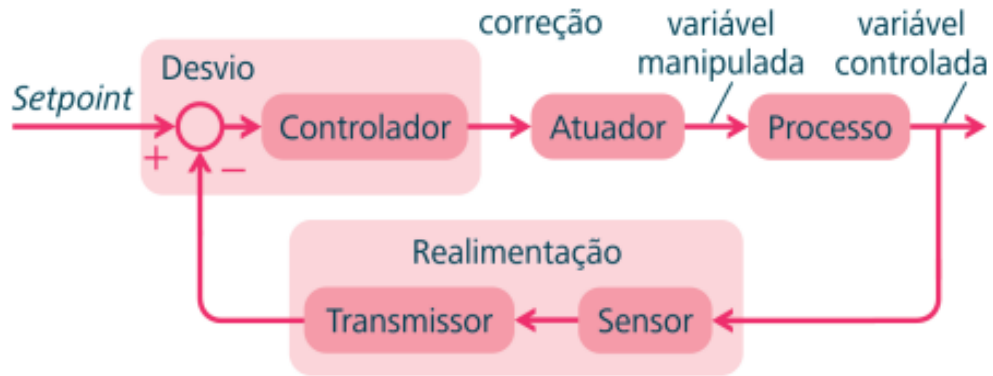
A automatização se desenvolveu a partir do século XIX quando houve a revolução industrial que demandou um rápido crescimento em diversas áreas para que a produção industrial pudesse ser satisfeita. No século seguinte, com a evolução da eletrônica e da computação, houve a integração entre as áreas industriais e a informática a partir da necessidade da leitura de sensores, processamento cada vez mais intenso e acionamento controles (SEIDEL, 2011).

#### 2.4.1 Definição dos Sensores

Para que seja possível monitorar e entender o que ocorre com o meio, utilizam-se sensores que, segundo a definição de Wendling (2010), são dispositivos sensíveis ao ambiente e realizam a leitura de grandezas físicas como temperatura, luminosidade, umidade, pressão, etc. De modo que possam ser entendidas, precisarão ser interpretadas, normalmente, utilizando-se de interfaces e/ou amplificadores e, desta forma, convertem os sinais lidos para que os circuitos eletrônicos os reconheçam.

Os sensores podem ser definidos quanto ao sistema de controle em dois tipos, malha fechada e malha aberta. Na malha fechada, conforme a **Erro! A origem da referência não foi encontrada.**<sup>9</sup> é possível verificar no diagrama de blocos que a saída é manipulada pelo controlador e dependente do sensor. Isso significa que, através da variável controlada, esta é monitorada pelo sensor, que por sua vez monitora a leitura das condições do sistema realizando a correção através da variável manipulada realimentando o sistema conforme a diferença entre a variável controlada e os parâmetros definidos. Como exemplo, é possível citar o controle de temperatura de um ar-condicionado (SEIDEL, 2011).

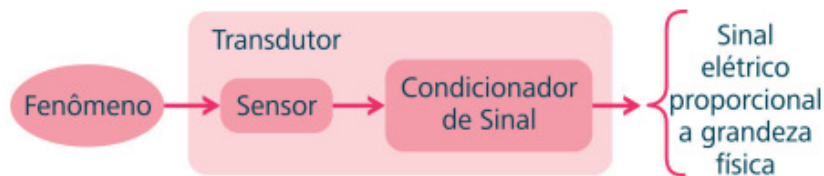
**Figura 9 - Diagrama de blocos de um controle de processos**



Fonte: Seidel (2011)

Já em um sistema de malha aberta, conforme a **Erro! A origem da referência não foi encontrada.**, a configuração de funcionamento difere da malha fechada, por tornar o sensor e o condicionador opcionais (SEIDEL, 2011). Neste caso podemos classificar o dispositivo como um transdutor, que pode ou não ter um sensor e um condicionador incorporados. A função de um transdutor é converter um tipo de energia em outro, um microfone, por exemplo, é um transdutor, já que transforma ondas sonoras em sinais elétricos que são amplificados através de uma interface para um alto-falante (PATSKO,2006).

Figura 10 - Funcionamento de um transdutor



Fonte: Seidel (2011)

## 2.4.2 Classificação de sensores

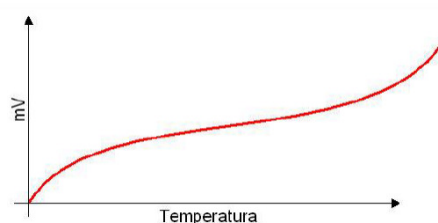
O meio em que vivemos está repleto de grandezas físicas que devemos monitorar. Para que isso seja possível, o desenvolvimento dos projetos deve considerar algumas características que diferenciam os sensores uns dos outros. Normalmente os sensores são classificados em duas categorias quanto à leitura dos estados do meio externo, analógicos e digitais.

### 2.4.2.1 Sensores analógicos

Os sensores analógicos detectam grandezas físicas contínuas, e produzem em sua saída um sinal elétrico proporcional que varia ao longo do tempo dentro de uma faixa de operação, podendo assumir qualquer valor, conforme a **Erro! A origem da referência não foi encontrada.**1 podemos ver o formato da leitura de um sensor de temperatura, onde a saída que é medida em milivolts, varia conforme a temperatura ao longo do tempo (WENDLING, 2010). As seguintes grandezas analógicas que podem ser medidas, temperatura, velocidade, deslocamento, pressão entre outras.

Normalmente os sinais tendem a ser de grandeza muito pequena, e os valores da diferença de potencial elétricos muito baixos, necessitando de circuitos eletrônicos para a leitura. Mesmo que a leitura do sensor seja analógica é possível realizar a conversão para dados digitais utilizando conversores analógicos digitais (A/D).

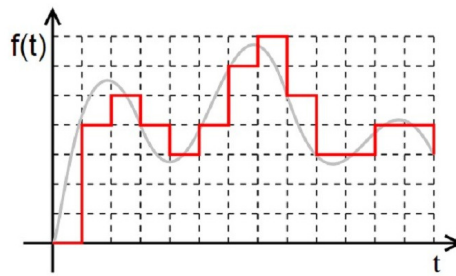
**Figura 11 - Sinal analógico**



**Fonte: Wendling (2010)**

Assim, as placas eletrônicas com entradas analógicas, já realizam a conversão A/D, como a ESP8266. Para que isso seja possível, a entrada recebe uma tensão variável proporcional, no ESP8266, a tensão pode variar de 0 a 3,3 V. Sua resolução para valores é de 10 bits, podendo receber valores entre 0 e 1023, ou seja, o valor da medição de tensão igual a 3,3 V é equivalente ao valor digital 1023 da saída do sensor (OLIVEIRA, 2017). Na **Erro! A origem da referência não foi encontrada.**2 é possível verificar o formato original do sinal analógico representado pela linha cinza e após a conversão A/D a representação digital pela linha vermelha. Nesta linha é possível verificar a amostragem para formato digital e quanto maior a resolução do equipamento de leitura da entrada, mais próxima ao sinal original será a representação.

**Figura 12 - Conversor A/D**

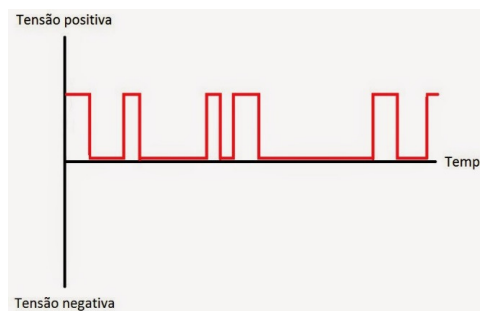


Fonte: Velloso (2014)

#### 2.4.2.2 Sensores digitais ou discretos

Os sensores digitais ou discretos, assim chamados por não serem contínuos, assumem em sua saída apenas dois valores ao longo do tempo (WENDLING, 2010). Tais valores são definidos como binários, ou seja, 1 para o valor de nível alto e 0 para o nível baixo, não possuindo valores intermediários, conforme a **Erro! A origem da referência não foi encontrada**.<sup>3</sup> Geralmente processam os dados após serem convertidos por circuitos eletrônicos. Podem ser aplicados em *encoders*, para a determinação de distância ou velocidade, detecção de passagem, sensores de presença, etc.

Figura 13 - Sinal digital



Fonte: Dantas (2014)

#### 2.4.3 Exemplificando os Sensores

Sensor de temperatura e umidade: O DHT11 é um sensor de temperatura e umidade que possui um sinal digital em sua saída. Possui um micro controlador de 8 bits e foi concebido com um elemento NTC do tipo resistivo. Seu tamanho é compacto e possui um consumo baixo, tendo uma ótima relação custo x benefício. A **Erro! A origem da referência não foi encontrada**.<sup>4</sup> ilustra o sensor DHT11.

Figura 14 - Sensor DHT11



**Fonte: Sensor DHT11 (2017)**

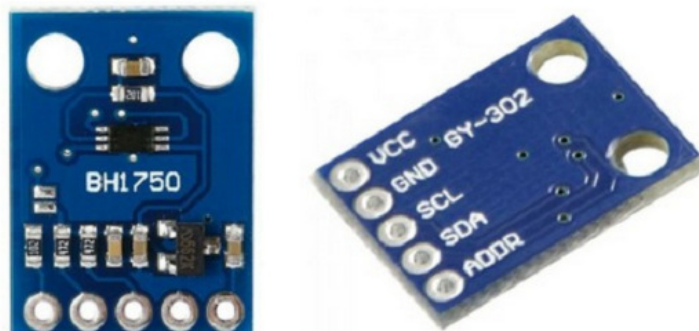
Este sensor possui as seguintes características:

- Tensão de alimentação: 3.3 a 5 Vcc;
- Sinal de Saída: Serial Digital;
- Temperatura: 0 a 50°C precisão de +/- 2C°;
- Umidade: 20 a 90% precisão de +/- 4%;
- Tempo de resposta: 2 segundos;

Sensor de luminosidade: possui um conversor analógico digital com resolução de 16 bits e sua comunicação é feita através do barramento de dados I2C presente em diversas placas de prototipagem. Este sensor comparado a um LDR comum possui maior precisão e possibilita ajuste do nível de luminosidade podendo ser utilizado em diversas aplicações que realizam leitura de luminosidade, **Erro! A origem da referência não foi encontrada.**

- Faixa de medição: 1 a 65535 lux;
- Resolução: 1 lux;
- Interface de comunicação: I2C;
- Resolução AD: 16bits;
- Tensão de alimentação: 3 a 5 Vcc.

**Figura 15 - Sensor de Luminosidade**



**Fonte: Sensor de luminosidade (2017)**

Sensor de umidade do solo: este sensor é utilizado para realizar a medição da umidade do solo, podendo ser inserido em diversos terrenos como na areia, terra ou até mesmo diretamente na água. Após realizar a calibração do sensor é possível identificar a condição do solo, se está úmido ou seco. Quando detectado que o solo está úmido, seu estado terá nível lógico alto e quando seco, seu estado lógico terá nível baixo, **Erro! A origem da referência não foi encontrada.6.**

**Figura 16 - Sensor de Umidade do Solo**



**Fonte: GBK Robotics (2017)**

#### **2.4.4 Atuadores**

Os atuadores são dispositivos que transformam energia elétrica, pneumática ou hidráulica em energia mecânica, ou seja, transformam um tipo de energia em trabalho mecânico. Esse trabalho mecânico pode ser usado para movimentação de polias, engrenagens ou correias.

Os atuadores podem ser classificados quanto a forma que realizam a conversão de energia conforme as características listadas abaixo:

- Atuadores hidráulicos: realizam a aplicação de trabalho através de um fluido, para a movimentação de um sistema qualquer utiliza pressão. Este tipo de atuador, normalmente, é empregado onde é necessário grande torque e/ou velocidade, porém não possuem grande precisão. Exemplo, freio de automóveis.
- Atuadores pneumáticos: realizam a aplicação de trabalho através de gases, porém possuem torque menor para a movimentação de um sistema. Contudo são sistemas mais baratos comparados aos hidráulicos, além de manutenção com menor

complexidade, entretanto também enfrenta problemas com baixa precisão. Exemplo, movimentador linear.

- Atuadores eletromagnéticos: motores elétricos (corrente alternada, corrente contínua, servo motores, motores de passo), realizam trabalhos mecânicos através do campo magnético gerado, apresentam um torque razoavelmente alto.

Entre os motores elétricos podemos dividi-los em dois tipos:

- Motores de corrente contínua (CC): possuem tamanho reduzido, os valores de torque permanecem constantes em variações de velocidade, porém necessitam de sensores de posição angular (*encoder*) ou de velocidade (tacômetro) para controle de posição ou velocidade em malha fechada (servocontrole). Como desvantagem, são motores mais caros e complexos. Exemplo, braço robótico.
- Motores de passo: podem funcionar em controle de malha aberta (posição e velocidade), são facilmente interligados com as plataformas anteriormente mencionadas, possuem baixo custo. Este motor apresenta algumas desvantagens, como, a queda do torque em altas rotações e a vibração em baixas rotações. Exemplo, movimentação de garras (FELIZARDO, 2005).

### 3 TRABALHOS CORRELATOS

Neste capítulo serão discutidos os trabalhos relacionados ao objeto desta dissertação. Atualmente, estão sendo desenvolvidos diversos trabalhos relacionados às possibilidades utilizando os conceitos da *IoT* com o auxílio de plataformas de prototipagens. Assim, na seção 3.1 será discutido o uso da plataforma Arduino na área da saúde em conjunto com a placa E-Health. Já na seção 3.2 será discutido o controle da irrigação de lavouras de morango automatizado com a utilização da plataforma de prototipagem Arduino. E por fim, na seção 3.3 será discutida a criação de uma estação meteorológica utilizando sensores e plataformas de prototipagem.

#### 3.1 TRABALHO 1: Internet Das Coisas: Um Protótipo Usando A Plataforma De Prototipagem Arduino E A Placa E-Health Para A Coleta De Sinais Vitais

A utilização de placas de prototipagem na área da saúde, vem se tornando frequente conforme ocorrem avanços tecnológicos. Aranda (2016), apresenta seu trabalho de *IoT* na área da saúde utilizando a plataforma E-Health para leitura e monitoramento de sinais vitais que pode ser utilizada em conjunto com plataformas de prototipagem.

Em seu estudo Aranda (2016), apresenta a importância da plataforma E-Health e os benefícios que ela pode trazer auxiliando médicos no acompanhamento dos pacientes a distância através da disponibilidade dos dados. Esses dados são enviados pela internet para um banco de dados, que com a segurança apropriada podem ser acessados pelos médicos de qualquer lugar.

O desenvolvimento de seu protótipo foi realizado utilizando a *shield E-Health* para o monitoramento dos sinais vitais de atletas da equipe de vôlei da Feevale. Esta placa era conectada com a plataforma Arduino que se encarregava de transmitir os dados para um repositório de dados na internet chamado *Thingspeak*. Esses dados eram acessados e disponibilizados através do desenvolvimento de um aplicativo para *Android* para que pudessem ser acessados por algum profissional da saúde.

Após a realização do estudo, foi possível avaliar positivamente o projeto que atendeu a proposta do protótipo para o monitoramento de sinais vitais utilizando a placa E-Health e disponibiliza-los para consulta. Também foi possível concluir a existência de muitas



possibilidades na utilização da placa para projetos em *IoT* gerando assim benéficos em diversas áreas.

### 3.2 TRABALHO 2: Sistema Automatizado Para Controle De Umidade E Temperatura Em Cultura De Morangos Aplicados Aos Pequenos Produtores

O trabalho de pesquisa realizado por Santos (2014), visou o a automação da irrigação de uma plantação morangueira com o objetivo de auxiliar pequenos produtores. Desta forma, reduzindo despesas com mão de obra e consumo de água, contribuindo também com o meio ambiente.

Santos (2014), verificou em seu estudo, que as propriedades rurais de pequeno e médio porte são desprovidas de aplicação de técnicas, tecnologias e conhecimento na produção agropecuária e agrícola consequentemente resultando em baixa produtividade. Outro ponto importante é a má utilização da água como recurso fundamental na produção agrícola.

Deste modo, a autora propôs um protótipo com um controle central que realiza a análise do ambiente, acionando as correções quando necessárias. Para que isso possa ocorrer, foi utilizada a placa Arduino em conjunto com sensores e atuadores que se comunicam através de circuitos de rádio frequência (RF).

Ao final, a autora conclui que o objetivo de implementar um sistema de irrigação automatizado a custos acessíveis para pequenos produtores foi atingido. Conforme a autora o sistema é de fácil manejo e todas as funções necessárias para o controle propostas auxiliam na melhor utilização dos recursos naturais evitando desperdícios e redução com mão de obra.

### 3.3 TRABALHO 3: Projeto De Um Sistema Remoto Para Aquisição De Dados Meteorológicos

Tenfen (2013), sugere um protótipo de uma estação meteorológica de forma a realizar a medição de fenômenos atmosféricos de forma simples e rápida. O objetivo do projeto é auxiliar áreas agrícolas, pecuárias e mesmo cidade pequenas que estejam longe de estações tradicionais.

Além disso, o projeto pretende ser uma alternativa para as atuais estações meteorológicas automáticas convencionais que possuem um custo elevado. Para tanto, Tenfen

(2013), propõem a utilização de plataformas de prototipagem em conjunto com uma série de sensores (temperatura, umidade do ar, pressão atmosférica, velocidade e direção do vento e precipitação) e a disponibilização dos dados em uma página web em tempo real, tudo isso a um custo muito inferior às convencionais.

Assim, Tenfen (2013), conclui seu estudo após a comparação a uma estação meteorológica convencional que o protótipo é economicamente viável, contudo ainda existem algumas barreiras a serem vencidas, como a aquisição de alguns sensores nacionais com valores elevados e melhorias que devem ser feitas tratando-se de um protótipo, reduzindo quando produzido em larga escala.

Por fim, neste capítulo foram apresentados alguns trabalhos da área de *IoT* onde é muito comum a utilização de placas de prototipagem e sensores com a capacidade de captura de diversos grandezas do meio em que estão. Em seguida, no capítulo 4 Procedimento Metodológico será apresentada a metodologia para a aplicação da pesquisa realizada.

## 4 PROCEDIMENTO METODOLÓGICO

Para chegar aos objetivos este projeto de pesquisa, pretende realizar um estudo que possibilitará verificar qual a melhor forma de implementar um dispositivo de acesso remoto aos parâmetros de controle de um conjunto de sensores para monitoramentos de uma planta. Para tanto, será realizado um estudo de pesquisa aplicada, no intuito de entender e implementar uma solução ao problema proposto (PRODANOV, FEITAS, 2013).

O método que melhor responderá ao problema de pesquisa será o método de natureza exploratória que nos permite um estudo aprofundado do assunto, em especial quanto a escolha da melhor plataforma de prototipagem ESP8266 ou Raspberry Pi que irá realizar leitura por meio de sensores, controle e eficiência energética, objetivando maior durabilidade de funcionamento do sistema sem a necessidade de recarga da bateria.

Com a finalidade de avaliar qual plataforma e sensores serão mais adequados ao projeto será utilizado o método de pesquisa bibliográfica, concomitantemente serão necessários utilizar procedimentos experimentais para a comprovação das pesquisas realizadas em artigos, material técnico, livros e sites de internet.

Após a realização do estudo teórico, foi realizada a implementação e análise do experimento que teve por objetivo a criação de dois cenários. O primeiro, utilizando a plataforma ESP8266 conectado aos sensores que realizarão o monitoramento de uma planta. O segundo, utilizará o mesmo esquema de ligação, porém será utilizada a plataforma Raspberry Pi.

Em cada cenário serão realizadas coletas das seguintes informações: tempo total de funcionamento do sistema, robustez, facilidade de implementação, tamanho, processamento e forma de comunicação sem fio.

Com base nas informações coletadas, estas serão apresentadas e avaliadas de forma quantitativa, a fim de comparar qual plataforma de prototipagem atinge a melhor solução a longo prazo com base nos cenários estudados.

Para tanto, foram seguidas as seguintes etapas:

- Etapa I: realização de uma pesquisa bibliográfica sobre as plataformas de prototipagem mais utilizadas e a forma como é possível realizar a conexão entre os sensores e atuadores disponíveis.
- Etapa II: De posse das informações da etapa anterior, define-se e projeta-se como será feita a coleta dos dados, através dos sensores, e parâmetros de

temperatura e umidade ambiente, umidade do solo, luminosidade e consumo de energia para posterior implementação.

- Etapa III: Com o levantamento da etapa anterior, implementa-se dois cenários com as plataformas Raspberry Pi e ESP8266 e os sensores que serão utilizados.
- Etapa IV: Finalmente, após finalizada a etapa anterior, será realizada a validação e análise dos dados apresentados nos dois cenários.

## 5 DESENVOLVIMENTO DO PROTÓTIPO E DA APLICAÇÃO

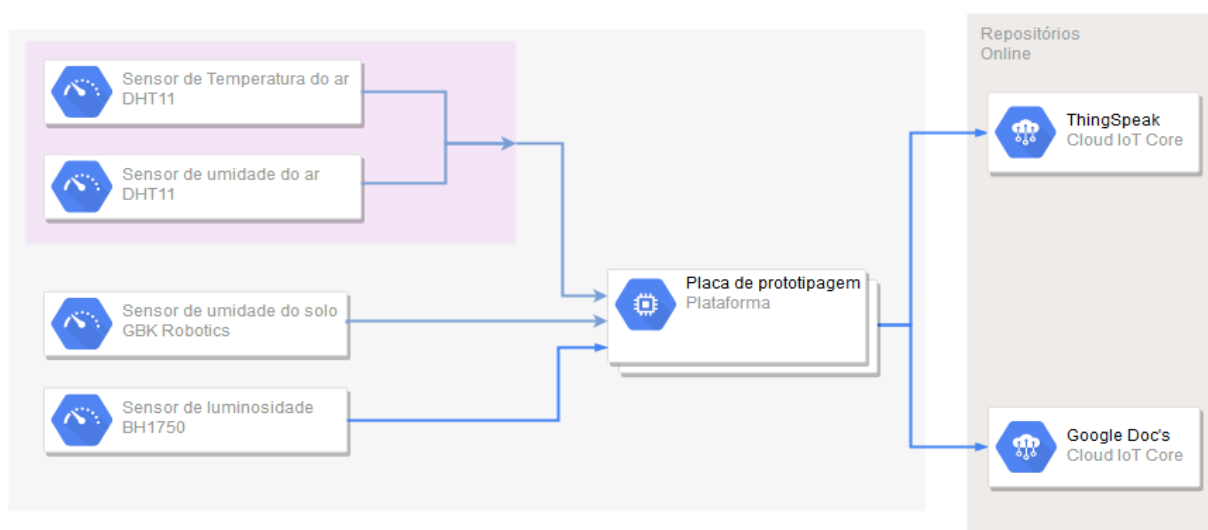
O presente capítulo irá apresentar o protótipo e todas as suas partes, descrevendo detalhadamente a montagem dos componentes, os parâmetros utilizados bem como a programação para a implementação. O estudo foi focado nas plataformas que possuem integrados componentes para a conexão com a internet como é o caso das placas ESP8266 e Raspberry Pi 3. Ao final serão apresentados os resultados do estudo comparativo entre as plataformas ESP8266 e Raspberry Pi 3.

### 5.1 DESCRIÇÃO DO PROTÓTIPO

Com base na bibliografia anteriormente apresentada, a proposta deste projeto será a de apresentar dois cenários similares. Dessa forma, utilizando sensores de temperatura, umidade do ar, umidade do solo e luminosidade ambiente para o monitoramento das condições ambientais e físicas de uma planta.

Estes dados serão coletados e processados pelas duas plataformas propostas, ESP8266 e Raspberry Pi 3, e após serão enviados para repositórios online na internet. Estes dados são armazenados no site e podem ser utilizados para visualização instantânea através de gráficos gerados pelo aplicativo *Matlab*, integrado ao site, como é o caso do *ThingSpeak*, ou para o manipulação e análises com o uso do site *Google Apps Spreadsheet* em que também é possível armazenar os dados e utilizar nas diversas ferramentas do site.

**Figura 17 - Arquitetura do protótipo**



Fonte: O autor (2018)

A Figura 17, apresenta a arquitetura do protótipo que realiza a leitura de três grandezas físicas, temperatura, umidade e luminosidade, utilizando dispositivos específicas para cada dado de entrada. Uma unidade de processamento das informações, ESP8266 ou Raspberry Pi, para posterior conexão e envio dos dados através da conexão *WiFi* aos repositórios na internet que neste projeto serão armazenados em dois locais diferentes, *ThingSpeak* e *Google Doc's*.

## 5.2 CENÁRIOS

Cada cenário será montado utilizando-se um sensor digital de temperatura e umidade do ar chamado DHT11. Este sensor é composto por quatro pinos, sendo o pino um a alimentação, que permite alimentar o sensor entre 3.3 Vcc a 5 Vcc; pino dois, saída digital, realiza a comunicação digital com as placas; para o pino 3 não existe conexão; e por fim o pino quatro, aterramento 0 Vcc.

O pino de dados necessita que seja utilizado um resistor de 4,7 k $\Omega$  em paralelo com a alimentação de 3.3 Vcc. Este resistor é chamado de *pull-up* e tem de a finalidade de evitar flutuação de tensão no pino de dados, deste modo eliminando a possibilidade de ruído.

Do mesmo modo, o sensor de luminosidade realiza a comunicação através das portas digitais, porém o protocolo utilizado é I2C. Sendo um barramento que comporta a conexão de grande variedade de dispositivos apenas com dois fios. Em paralelo ao barramento será necessária a conexão de dois resistores de 4,7 k $\Omega$ , um para cada fio, a fim de evitar interferências na comunicação. BH1750 é a referência do modelo utilizado.

Por fim, foi utilizado um sensor analógico padrão da empresa GBK Robotics com ótima relação custo x benefício que realiza a medição da umidade do solo. Ele se comunica com a placa de forma analógica, coletando todo espectro de valores, deste modo, não necessitando do uso de resistor. Sua conexão é extremamente simples já que se conecta diretamente a porta analógica do ESP8266 ou a um conversor analógico para digital, como na placa Raspberry Pi 3 que não possui entradas analógicas.

### 5.2.1 Montagem ESP8266

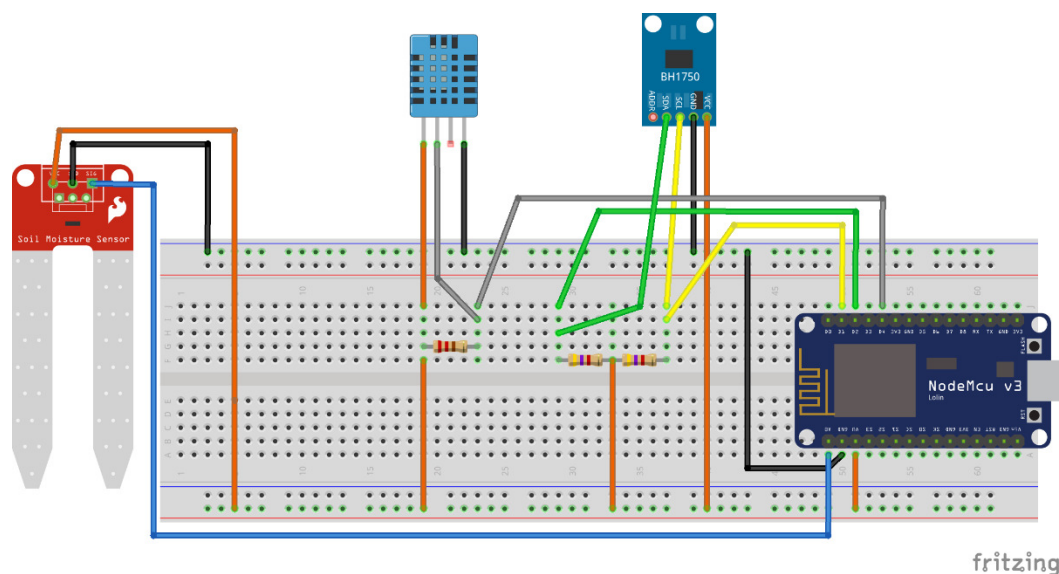
O módulo WiFi ESP8266 NodeMCU – ESP12 foi escolhido devido a sua relação custo x benefício tendo integrado em um único componente a comunicação sem fio, portas digitais, portas analógicas e protocolos de comunicação.

A alimentação de entrada é de 5 Vcc, podendo ser via USB, fonte chaveada ou 3,3 V. Já os terminais de alimentação interna é 3,3 Vcc sendo compatível com uma grande variedade de dispositivos.

A conexão entre o sensor DHT11 e a placa é feita através do pino de dados à porta digital conhecida como GPIO 12 (D6) apenas por um fio. Já o sensor de luminosidade realiza o envio de dados para o barramento I2C e foi conectado nas portas GPIO 5 (D1) e GPIO 4 (D2) que possuem suporte para este protocolo. As portas digitais podem ser identificadas como D1 e D2 quando simples entradas / saídas digitais ou SCL e SDA respectivamente nesta ordem. A utilização de resistores é obrigatória para manter a fidelidade dos sinais lidos, evitando interferência indesejadas.

Finalmente, o sensor de umidade do solo foi conectado na única porta analógica disponível na placa identificada como A0. Na Figura 18 é possível visualizar o esquema de ligação.

**Figura 18 - Conexão dos sensores na placa ESP8266**



Fonte: O autor (2018)

## 5.2.2 Montagem Raspberry Pi 3

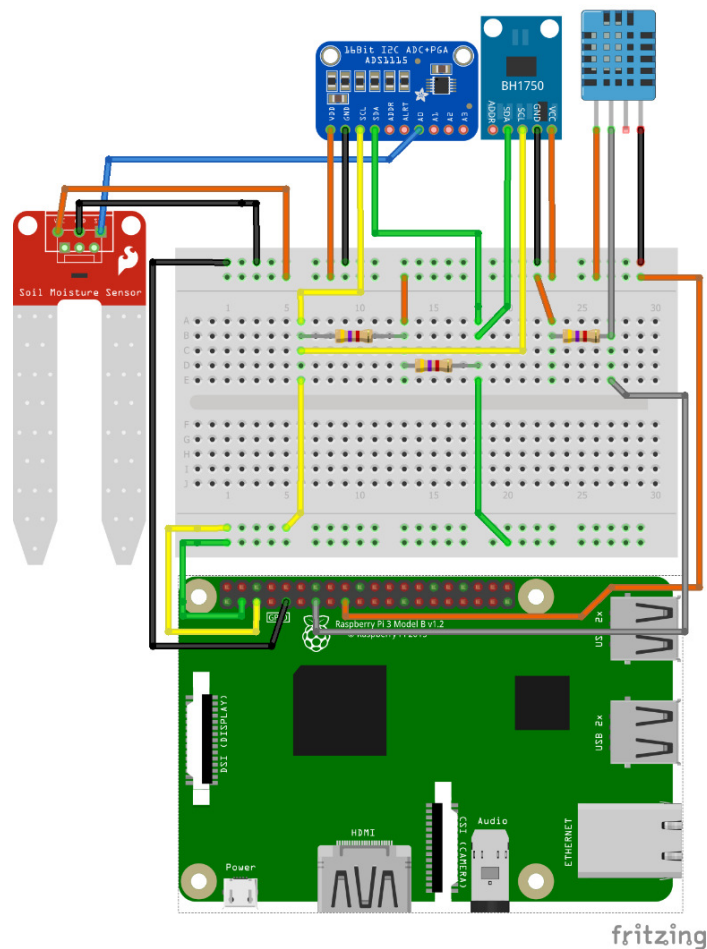
Com o propósito de evitar distorções nos resultados foram mantidos os mesmos sensores nas duas plataformas para realizar as leituras das condições da planta.

Deste modo, para a conexão do sensor de umidade do solo que produz em sua saída apenas sinal analógico houve a necessidade de adicionar um módulo conversor de sinais analógicos para digitais.

O modelo do módulo utilizado é o ADS1115 que possibilita a conexão de até 4 entradas analógicas. Possui resolução de 16 bits permitindo até 65.536 medidas. Sua comunicação com a placa é realizada através do protocolo I2C. Do mesmo modo que a placa ESP8266, será necessária a conexão de dois resistores de 4,7 k $\Omega$  conectados entre o positivo e cada um dos fios de entrada digital SDA (GPIO2) e SCL (GPIO3).

Para finalizar, o sensor DHT11 responsável por realizar a medição da temperatura e umidade do ambiente será conectado pelo seu pino de dados diretamente à porta digital GPIO 27. Na Figura 19, é possível visualizar a placa e os dispositivos conectados a ela.

**Figura 19 - Conexão dos sensores na placa Raspberry Pi 3**



Fonte: O autor (2018)

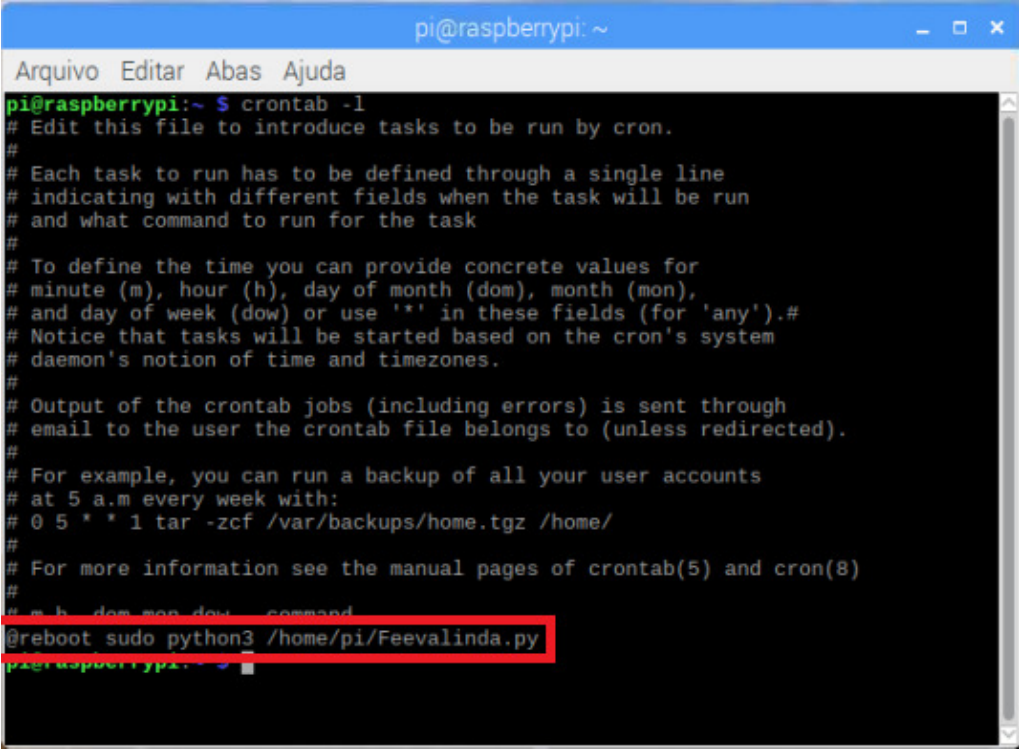


### 5.2.2.1 Agendador de tarefas cron

Para projetos autônomos é necessário que a aplicação dentro da plataforma seja iniciada após a finalização da inicialização da placa. Assim, ao contrário da placa ESP8266, que, ao receber uma aplicação em memória irá executar as instruções continuamente até ser desligado, já o Raspberry necessita que seja criada uma regra em seu serviço de agendamento de tarefas. Este agendamento pode entre outras tarefas executar um *script* em Python, linguagem de programação nativa para a plataforma.

Deste modo, a cada inicialização, o sistema operacional *Raspbian* irá executar o serviço de agendamento de tarefas. A ferramenta utilizada para realizar o agendamento de tarefas se chama *cron*. Então, para que se possa editar a lista de tarefas é necessário executar no terminal o comando *crontab -e* e inserir a tarefa que deseja agendar. A Figura 20 ilustra na área demarcada o comando que irá executar o *script* da aplicação ao carregar o sistema operacional.

Figura 20 - Cron, agendador de tarefas do *Raspbian*



```
pi@raspberrypi:~ $ crontab -e
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
@reboot sudo python3 /home/pi/Feevalinda.py
```

Fonte: O autor (2018)

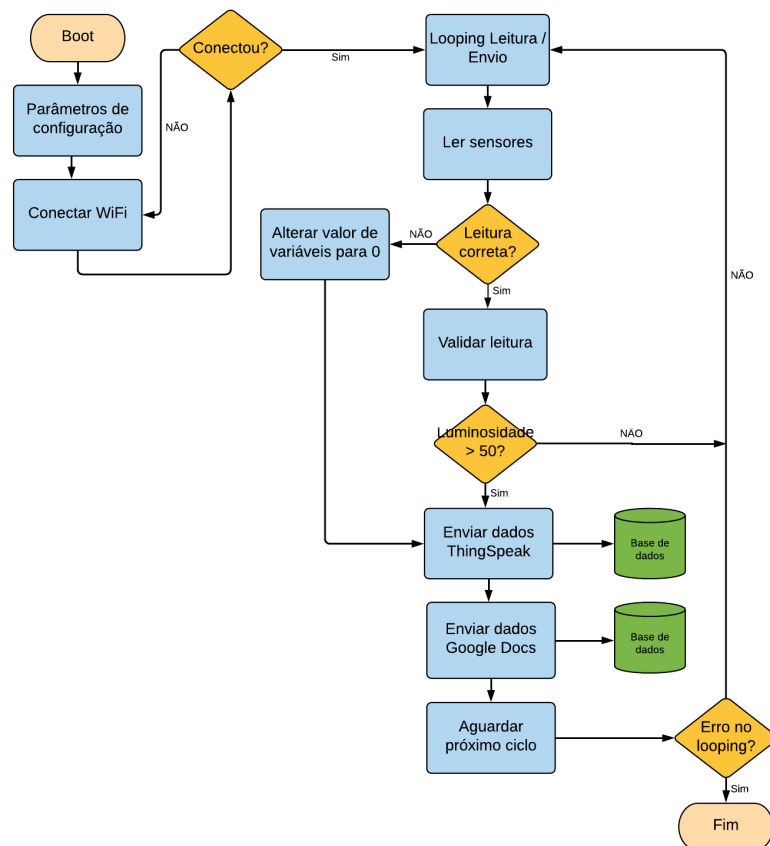
## 5.4 DESENVOLVIMENTO DAS APLICAÇÕES

Nesta sessão serão apresentadas as aplicações para cada plataforma assim como os seus detalhes, particularidades e funcionalidades. A plataforma ESP8266 utiliza a interface de programação Arduino *IDE* que foi desenvolvida baseada na linguagem de programação C++. De outro modo, para que se possa programar qualquer aplicação para o Raspberry Pi 3 utiliza-se a linguagem de programação Python como já mencionada sendo nativa para a plataforma.

Os dados capturados por cada plataforma através dos sensores recebem um processamento conforme as particularidades de cada linguagem. Após a validação as informações são enviadas para dois repositórios online *Thingspeak* e *Google Apps Spreadsheet*.

Podemos visualizar na Figura 21 o fluxograma básico de como foram desenvolvidas as aplicações para cada plataforma.

**Figura 21 - Fluxograma seguido por cada aplicação**



Fonte: O autor (2018)

### 5.4.1 Arduino *IDE*

A plataforma ESP8266 foi concebida para utilização com a linguagem de programação nativa Lua. Porém, sendo muito versátil, pode ser adaptada para utilização na interface de desenvolvimento Arduino *IDE*, assim, disseminando sua utilização pela facilidade de uso que passou a possuir.

O desenvolvimento de qualquer aplicação neste ambiente é dividido em três partes. Definição e inclusão de bibliotecas e variáveis a serem utilizadas, a segunda parte configuração e inicialização de bibliotecas e variáveis chamada *void setup()* e a terceira área onde a aplicação permanece rodando indefinidamente até o ponto de ser interrompida, nesta área onde o código está rodando, chamada de *void loop()*.

Assim, para que se possa interpretar as funcionalidades ou sinais dos sensores conectados na placa e transmitir essa informação para o programa são necessárias bibliotecas. A principal função delas é fornecer suporte através de software para acesso aos componentes e geralmente são fornecidos pelos fabricantes e encontradas em suas páginas na internet. Portanto, os componentes utilizados no projeto fazem uso desses códigos, a leitura da temperatura e umidade realizada pelo sensor DHT11 é um exemplo, sendo declarada da seguinte forma *#include <dht.h>*. Segue abaixo o Quadro 1 onde é possível verificar as bibliotecas utilizadas na aplicação.

**Quadro 1 - Bibliotecas utilizadas para comunicação entre sensores e a placa**

```
//Include da lib de Wifi do ESP8266
#include <ESP8266WiFi.h>
//Include da nova lib do sensor DHT11 e DHT22
#include <dht.h>
//Include da lib do sensor de iluminação
#include <BH1750.h>
//Include da lib para o barramento I2C
#include <Wire.h>
```

**Fonte: O autor (2018)**

A próxima etapa, consiste na inicialização ou configuração dos componentes e definição de funções. Isso ocorre na segunda área chamada *void setup()*. É neste ponto, por exemplo, que foram inseridos os parâmetros necessários para a conexão *WiFi*, informando o nome da rede e a senha através das variáveis *ssid* e *password* presentes na instrução *WiFi.begin(ssid, password)*. No Quadro 2 podemos verificar a inicialização dos dispositivos necessários para o projeto.

**Quadro 2 - Área de inicialização de parâmetros *void setup()***

```

void setup() {
pinMode(pino_sinal_analogico, INPUT);
pinMode(ESP8266_LED, OUTPUT);
//Configuracao da UART
Serial.begin(115200);
//Inicializa barramento I2C utiliza SCL(D1) e SDA(D2)
Wire.begin();
//Inicializa o BH1750
lightMeter.begin();
//Inicia o WiFi
WiFi.begin(ssid, password);

```

**Fonte: O autor (2018)**

Finalmente, na terceira área são desenvolvidos os códigos, que serão executados continuamente pela aplicação. Neste ponto os dados são coletados pelos sensores, em seguida processados conforme diretrizes previamente definidas e executam ações, como o envio de dados para a internet.

No Quadro 3 podemos verificar uma parte do código do projeto que realiza a leitura dos sensores e atribui a variáveis seus valores, um exemplo, é a instrução *umidade = DHT.humidity;*

**Quadro 3 - Área de desenvolvimento do código *void loop()***

```

void loop() {
//Espera 15 segundos para fazer a leitura
delay(15000);
digitalWrite(ESP8266_LED, HIGH);

//int chk =
DHT.read11(DHT11_PIN);
float umidade;
float temperatura;
float higr;
float higr_perc;
float lum;

//Leitura de umidade
umidade = DHT.humidity;

//Leitura de temperatura
temperatura = DHT.temperature;

//Le o valor do pino A0 do sensor
higr = analogRead(pino_sinal_analogico);
higr_perc = (higr) / 6.15;

```

**Fonte: O autor (2018)**

### 5.4.2 Python

Python é uma linguagem de programação interpretada e multi plataforma, ela foi concebida para ser simples e intuitiva. Pensada para facilitar o uso e aprendizado, rápida em aplicações simples e poderosa ao mesmo tempo quando tarefas mais elaboradas são necessárias. Deste modo, os desenvolvedores da plataforma Raspberry Pi adotaram essa linguagem como nativa no intuito de popularizar a plataforma.

Com uma flexibilidade um pouco maior comparada a Arduino *IDE*, Python não delimita áreas específicas para o desenvolvimento dos *scripts* como são chamadas as aplicações. Porém, segue uma linha lógica semelhante a outras linguagens de programação de forma mais intuitiva favorecendo a criação.

As bibliotecas são igualmente necessárias na plataforma e possuem uma grande base de dados colaborativa para que se possa conectar com os mais diversos dispositivos disponíveis para auxiliar na interpretação pela aplicação. A inclusão das funções adicionais através das bibliotecas se dá no início da aplicação conforme pode ser visto no Quadro 4.

**Quadro 4 - Inserção de bibliotecas em Python**

```
import sys
import urllib.request
from urllib.error import HTTPError
from time import sleep
import Adafruit_DHT as DHT
import Adafruit_ADS1x15 as ADS1115
import RPi.GPIO as GPIO
import requests
import smbus
import json
import time
import datetime
```

**Fonte: O autor (2018)**

Após a inserção das bibliotecas necessárias, o restante da área de desenvolvimento da ferramenta não possui mais áreas específicas podendo ser utilizada para a definição de funções, variáveis e inicialização de componentes. Logo após, é possível iniciar o desenvolvimento do código que será utilizado pela aplicação para processo leitura das entradas, processamento das informações conforme as condições determinadas e ao fim a execução de ações pré-definidas, conforme pode ser visto no Quadro 5.

**Quadro 5 - Desenvolvimento de aplicações em Python**

```

def leLuminosidade(addr=BH1750):
    data = bus.read_i2c_block_data
    (addr, ONE_TIME_HIGH_RES_MODE)
    return converteParaNumero (data)

def conexao_Abrir_Planilha(arquivo_Chave_Conf,
planilha_Google):
    try:
        scope = ['https://spreadsheets.google.com/feeds',
'https://www.googleapis.com/auth/drive']
        credenciais =
ServiceAccountCredentials.from_json_keyfile_name
(arquivo_Chave_Conf, scope)
        gc = gspread.authorize(credenciais)
        worksheet = gc.open(planilha_Google).sheet1
        return worksheet
    except Exception as ex:
        print('Falha ao conectar Google sheet com erro:',
ex)
        sys.exit(1)

worksheet = None

while (True):
    try:
        umid, temp = DHT11_data()
        higr = sensor_ADS1115.read_adc(0)
        if(higr < 0):
            higr = higr * (-1)
        lum = float(str(leLuminosidade()))

        if temp is not None and umid is not None and
higr is not None:

            temp = '%.2f' % temp
            umid = '%.2f' % umid
            lum = '%.2f' % lum
            higr_perc = (higr) / 152.54
            higr_perc = '%.2f' % higr_perc
            print("Temperatura: " + temp + " °C")
            print("Umidade: " + umid)
            print("Higrômetro: " + higr_perc + " %")
            print("Luminosidade: " + lum + " lux")
            sleep(5)

```

**Fonte: O autor (2018)**

### 5.4.3 ThingSpeak

O armazenamento dos dados foi um dos pré-requisitos após a coleta realizada pelos sensores sendo prevista na fase de projeto. A fim de que se pudesse armazenar e também visualizar os dados gerados pelo protótipo, foi criada uma conta no site *www.thingspeak.com* que disponibiliza um serviço gratuito para armazenamento e recuperação de informações baseado no protocolo *HTTP* permitindo ao usuário capturar e postar informações através dos comandos *GET* e *POST* para projetos da área de *IoT*.

Para que fosse possível utilizar o serviço, deve-se realizar o cadastro de uma conta no site. Completada esta etapa, foi possível criar um canal e realizar as configurações necessárias para o projeto como, nome, descrição, dados que serão monitorados através de até oito campos. Conforme pode ser visto abaixo, na Figura 22.

**Figura 22 - Configuração do canal ThingSpeak**

The screenshot shows the 'Channel Settings' page for a ThingSpeak channel named 'FEEVALINDA'. The channel ID is 565441, and the author is 'eduardowiest'. The channel is public and is used for 'MONITORAMENTO DE PLANTAS'. The settings include a name, description, and eight data fields, each with a checkbox to enable it. The 'Ajuda' (Help) section on the right provides instructions on how to configure the channel settings.

Campos	Nome do Campo	Ativado
Campo 1	ESP_Temperatura	<input checked="" type="checkbox"/>
Campo 2	ESP_Umidade	<input checked="" type="checkbox"/>
Campo 3	ESP_Higrômetro	<input checked="" type="checkbox"/>
Campo 4	ESP_Luminosidade	<input checked="" type="checkbox"/>
Campo 5	Pi_Temperatura	<input checked="" type="checkbox"/>
Campo 6	Pi_Umidade	<input checked="" type="checkbox"/>
Campo 7	Pi_Higrômetro	<input checked="" type="checkbox"/>
Campo 8	Pi_Luminosidade	<input checked="" type="checkbox"/>

**Ajuda**  
Channels store all the data eight fields that can hold a status data. Once you colle visualize it.

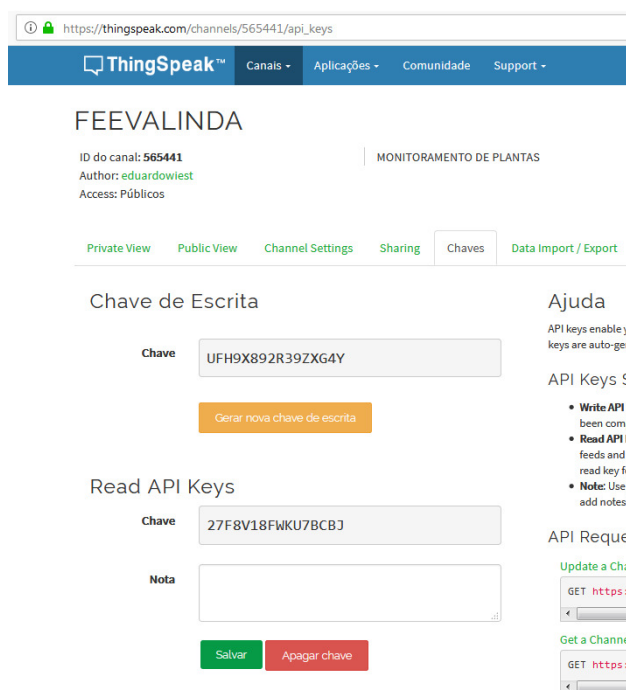
**Channel Setting**

- **Channel Name:** Ent
- **Description:** Enter a
- **Field#:** Check the bc channel can have up
- **Metadata:** Enter inf
- **Tags:** Enter keyword
- **Link to External Site:** ThingSpeak channe
- **Show Channel Loca**
  - **Latitude:** Spe latitude of th
  - **Longitude:** S longitude of t
  - **Elevation:** Sp the city of Lor
- **Video URL:** If you ha information, specify
- **Link to GitHub:** If yc repository URL.

Fonte: O autor (2018)

Finalizada a etapa de configuração, agora é possível iniciar a programação e inserir os valores do monitoramento na base de dados criada. Para que isto aconteça, é necessário gerar uma chave de escrita, forma como o site realiza a autenticação no canal. Na Figura 23 é possível visualizar as chaves de escrita no canal.

**Figura 23 - Configuração do canal *ThingSpeak***



**Fonte: O autor (2018)**

Com todo o ambiente configurado e com a posse da chave de acesso é criar a aplicação que irá coletar os dados, realizar a autenticação e inserir na base de dados. Para que isso ocorra basta executar o comando de *POST* passando os parâmetros de endereço do *site*, chave de acesso e campos com as respectivas informações. Assim, já será possível visualizar em tempo real um gráfico em função do tempo as variações numéricas.

#### **5.4.4 Google Apps Spreadsheet**

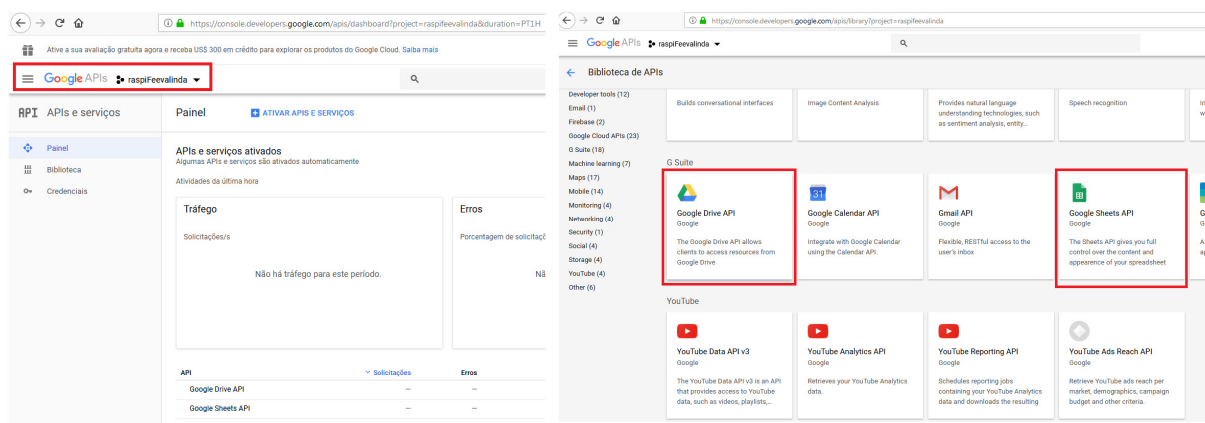
O *Google Apps Spreadsheet* é um serviço de planilhas online disponibilizado para empresa *Google*. Este serviço possui funcionalidades semelhantes as planilhas de cálculos utilizadas em *desktop*, permitindo a inserção, manipulação e até mesmo a criação de gráficos com os dados conforme a necessidade.



Utilizado no contexto do projeto como segundo local de armazenamento para os dados. O *Google API Spreadsheet* teve como objetivo a comparação entre as duas soluções de armazenamento para verificar a disponibilidade no recebimento dos dados enviados pela aplicação.

Para que seja possível utilizar o serviço, mesmo sendo oferecido gratuitamente é necessário que seja feita uma inscrição no *site* do *Google*. Após o cadastro deve-se acessar o endereço <https://console.developers.google.com/apis/>. Nesta janela foi criado o novo projeto chamado *raspiFeevalinda* para enviar os dados do Raspberry Pi 3, conforme pode ser visto no quadro vermelho da Figura 24 A. O passo seguinte foi a ativação das *API's* e serviços *Google Drive API* e *Google Sheets API*, este último responsável pelo a planilha onde os dados serão salvos conforme pode ser visto na Figura 24 B.

**Figura 24 - Criando projeto dentro do ambiente *Google API***



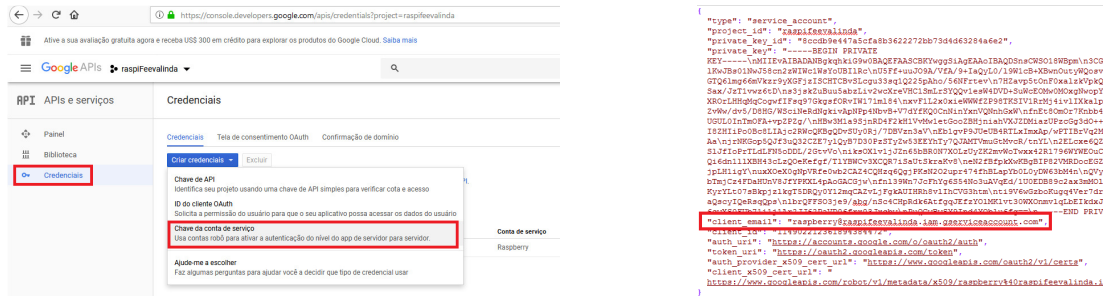
**A) Projeto *raspiFeevalinda* criado**

**B) Bibliotecas do *Google API* utilizadas**

**Fonte: O autor (2018)**

Em seguida foi necessária a criação da chave de conta de serviço, necessária para permitir a conexão com o *site*, em que é gerado um arquivo com a extensão *JSON* com as informações das credenciais de autenticação de segurança *OAuth2*, conforme Figura 25 A. Após o download do arquivo, deve-se abrir o arquivo e procurar pela linha marcada no quadro vermelho. Neste ponto, a informação entre aspas com o endereço de e-mail, conforme Figura 25 B, deve ser copiada para utilização posterior utilização como chave de acesso no próximo passo.

Figura 25 - Informações da credencial de autenticação Outh2



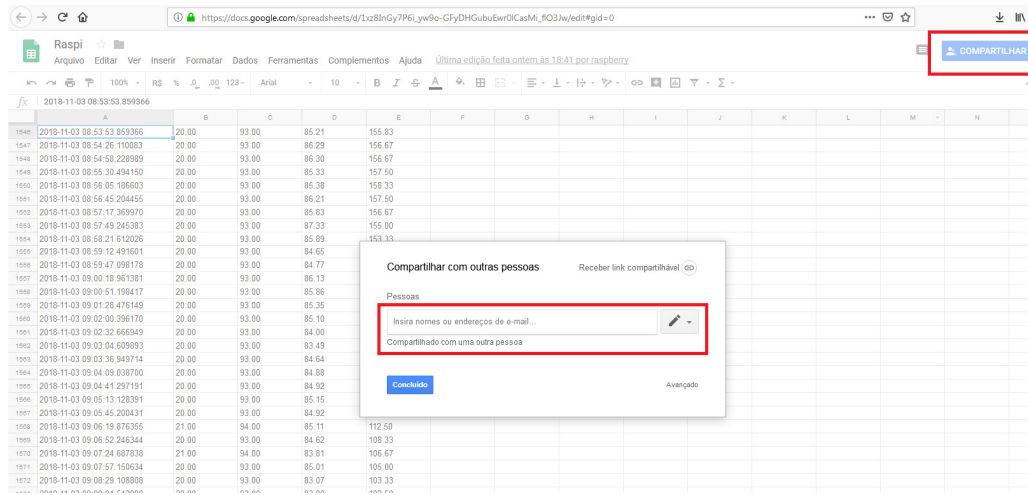
A) Criando credencial de autenticação

B) Chave de acesso

Fonte: O autor (2018)

E finalmente, após criar uma planilha eletrônica no ambiente em nuvem *Google Drive*, será utilizada a chave de acesso. Para tanto, é necessária a utilização de uma função chamada “Compartilhamento”, esta função está situada no conto superior direito da planilha. Neste ponto, é inserida a linha copiada anteriormente que servirá como chave de acesso. Conforme Figura 26.

Figura 26 - Inserção da chave de acesso



Fonte: O autor (2018)

Finalizada a etapa de configuração do ambiente, e iniciada a programação, foi necessário deixar uma cópia do arquivo que possui as credenciais na mesma pasta onde será rodado a aplicação na linguagem *Python*.

## 5.5 VALIDAÇÃO DOS SENSORES

Para realizar as medições de temperatura e umidade foi utilizado o sensor DHT11. Conforme as características técnicas apresentadas na sessão 2.4 Sensores e Atuadores, este sensor foi utilizado por atender perfeitamente as especificações do projeto, seja na leitura da faixa de medição de temperatura quanto de umidade.

Um ponto a ser observado no desenvolvimento do código da aplicação para este sensor, é o tempo de resposta entre cada leitura, sendo necessário respeitar um intervalo que não deve ser inferior a 2 segundos com possibilidade de erros de leitura, assim foi estabelecido que o tempo mínimo será de 10 segundos.

O envio das medições é realizado através de um micro controlador interno através de sinal serial utilizando apenas um fio, sendo interpretados por sua biblioteca. Para aferição foi realizado um procedimento simples de comparação com um medidor de temperatura tradicional.

Já o sensor de umidade do solo utilizado no projeto realiza leituras analógicas, desta forma, o sinal na sua saída está constantemente variando conforme a resistência do solo. O sensor utilizado é da marca GBK Robotics e não possui módulo digital, sendo alimentado com 3,3 V.

O funcionamento do sensor é semelhante a uma resistência variável que em contato com o solo irá permitir a passagem de tensão elétrica conforme a umidade do solo. Quanto maior a umidade do solo maior será a tensão e menor a resistência e vice-versa.

Para medir a umidade do solo e permitir um estudo comparativo, foi utilizada uma escala percentual, pois as duas placas diferem na resolução de leitura. A placa ESP8266 possui 10 bits de resolução, sendo possíveis 1024 leituras diferentes. Enquanto que a placa conversora de sinais analógicos para digital utilizada no Raspberry Pi 3, possui 16 bits de resolução, possibilitando até 65.536 medidas diferentes.

Para a calibração do sensor foi realizado o seguinte procedimento. Conectado a placa e coletando as diferentes leituras de valores, o sensor foi mergulhado em um reservatório contendo somente água até certa de 5/6 do seu comprimento, assim, simulando a mesma profundidade quando inserido no solo da planta. Em seguida, verificado no console de cada plataforma o maior valor medido nesta condição.

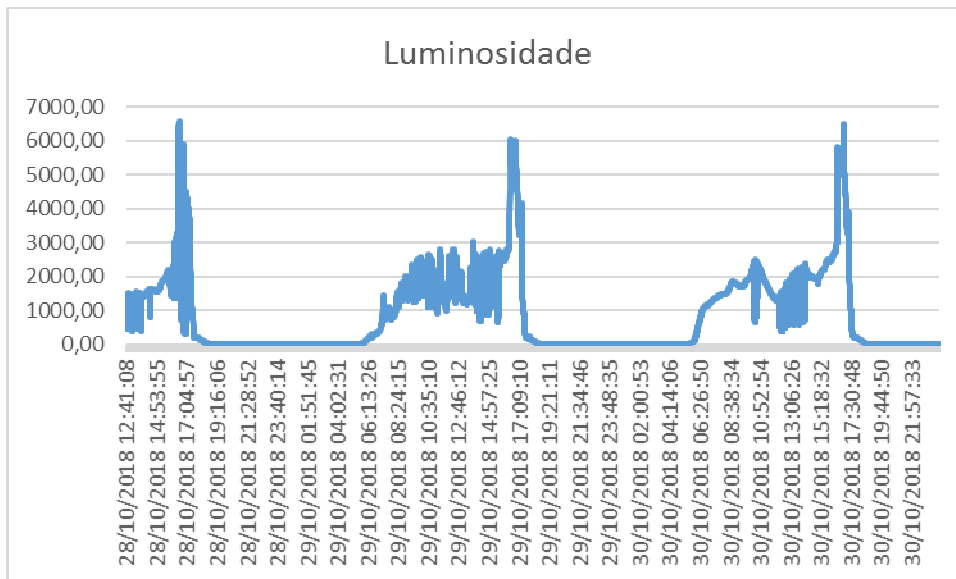
Com os dados de valor máximo coletados para cada cenário, é possível estabelecer o valor percentual para cada leitura com a aplicação da fórmula de equalização das medidas, como segue abaixo.

$$higr_{perc} = (higr) * \left( \frac{100}{higr_{m\acute{a}x.}} \right)$$

O BH1750 foi o modelo de sensor utilizado para a leitura dos valores de luminosidade ambiente. Este sensor com dimensões reduzidas possui uma resolução de 16 bits possibilitando a leitura da intensidade luminosa na faixa de 1 a 65.535 lux quando exposto diretamente a luz.

Com o objetivo de reduzir o consumo no período da noite em que não há contribuição significativa com dados para o estudo, este sensor foi utilizado para definir uma zona de corte. Conforme a Figura 27, é possível constatar que houve uma estabilização dos valores no menor patamar durante um período de aproximadamente 50% do tempo de leitura, portanto o valor de 50 lux foi estabelecido para a zona de corte.

**Figura 27 - Comparativo luminosidade dia x noite**



**Fonte: O autor (2018)**

Sendo assim, o número de conexões necessários para enviar os dados aos sites de repositórios será reduzido, representando uma economia significativa de conexões e consumo, conforme poderá ser visto no capítulo 6 Experimento e Análise dos Resultados.

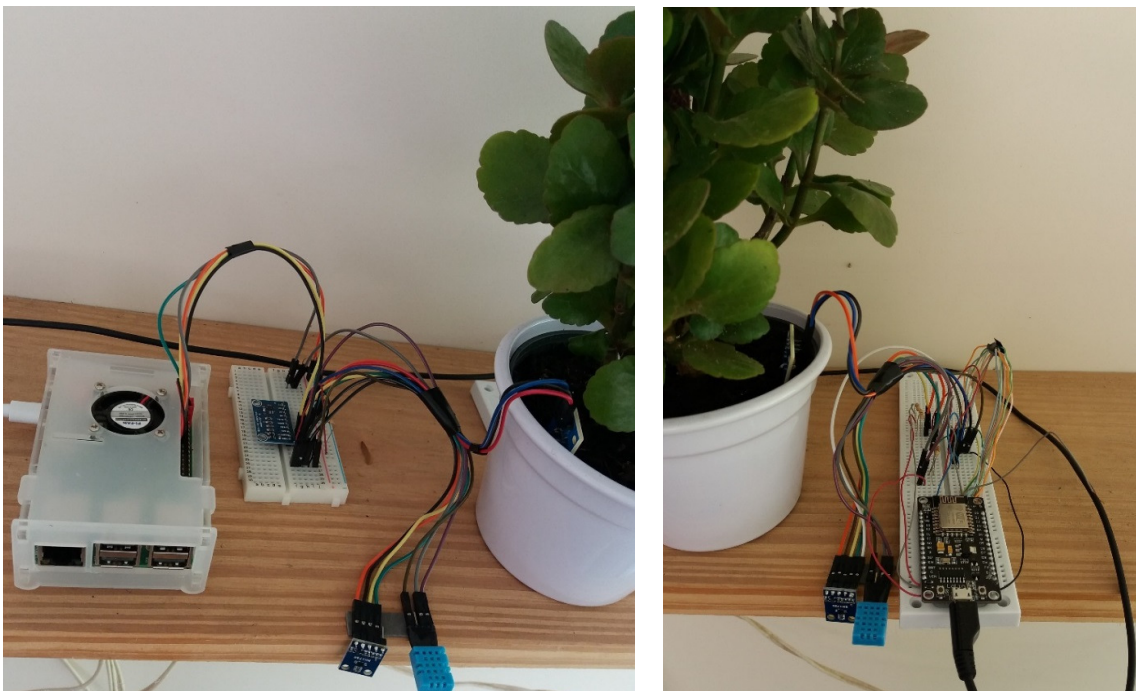
## 6 EXPERIMENTO E ANÁLISE DOS RESULTADOS

Neste capítulo será apresentado o experimento bem como os parâmetros utilizados para o balizamento do estudo comparativo entre as duas plataformas. Além disso, será feita uma análise com os dados coletados do consumo elétrico, leitura dos sensores, e o envio dos dados aos dois *sites* repositórios, *Thingspeak* e *Google Apps Spreadsheet*. Conforme já mencionado no subcapítulo 5.2 Cenários, foram utilizados os mesmos modelos de sensores para as duas plataformas.

Com o objetivo de realizar a coleta dos dados foram realizados por três dispositivos obtendo quatro medidas diferentes. As grandezas de temperatura ambiente medida em graus Celsius, umidade ambiente medida em valor percentual, a luminosidade ambiente medida em lumens e umidade do solo medida em unidade percentual.

Deste modo, para que não houvessem variações nas condições de leitura, cada plataforma foi posicionada em um dos lados da planta, conforme Figura 28. Desta forma, procurou-se eliminar ao máximo qualquer interferência referente a variação do ambiente no âmbito da luminosidade, visto que a sensibilidade do sensor permite causar distorções quando realizada a comparação em posições diferentes.

**Figura 28 - Posicionamento das plataformas**



**Fonte: O autor (2018)**

Ainda, referente a sua orientação solar, o protótipo estava em um local que recebia maior incidência da iluminação ao final do dia. Sendo assim, quando o sol estava a oeste, a iluminação era mais intensa.

Outro ponto a ser observado com relação ao sensor de luminosidade, está no fato de não haver captação de iluminação no período da noite, fato este, que foi mencionado anteriormente na validação do sensor. Baseado nisso, foi definido um ponto de corte com o valor de 50 lux, sendo que abaixo deste valor não serão feitas conexões com os servidores de dados, gerando uma redução de acessos desnecessários.

Com relação ao posicionamento do protótipo, é importante mencionar, que, para a coleta dos dados, não houvesse interferência externa ao projeto, este permaneceu em um ambiente protegido contra qualquer tipo de manipulação indesejada. Já que, uma simples obstrução do sensor de luminosidade ou mesmo a proximidade de uma fonte de calor no sensor de temperatura e umidade poderia alterar significativamente os dados do estudo comparativo.

Quanto ao sensor de umidade do solo, foi definido que a reposição de água se daria sempre no período da noite, período que possui menor índice de evaporação do solo em função da temperatura mais baixa. Sendo assim, auxiliando na melhor absorção do líquido pelo solo e mantendo a planta mais tempo hidratada.

Durante o período de coleta dos dados, que cada cenário obedeceria um ciclo de 30 segundos, sendo dividido da seguinte forma. Após o acesso à leitura dos sensores foi inserida uma pausa de 5 segundos, em seguida realizado o envio dos dados para o *site* repositório *Thingspeak*, novamente inserida uma pausa de 5 segundos e então os mesmos dados enviados para o segundo *site* repositório. Ao final foi inserida uma pausa de 20 segundos finalizando o ciclo.

## 6.1 VALIDAÇÃO DO CONSUMO ELÉTRICO

Um ponto muito importante quando se fala em dispositivos para *IoT* é o consumo elétrico. Pois, com um número de dispositivos cada vez maior no mercado conforme mencionado no primeiro capítulo é importante levar em conta no projeto quando se deseja implantar dispositivos em áreas remotas e utilizar fontes de energia com pouca autonomia.

Pensando nisso, foi realizado um estudo entre as duas plataformas ESP8266 e Raspberry Pi 3 a fim de verificar qual a melhor solução entre as duas propostas. Os dados

foram coletados levando em conta o consumo nos principais estados de funcionamento, conforme descritos abaixo:

- Boot: consumo máximo na inicialização de cada dispositivo;
- Aguardando: finalizada a etapa de inicialização do dispositivo e aguardando a execução da aplicação;
- Lendo sinais: após iniciada a execução da aplicação e realizando o acesso à leitura dos dados dos sensores;
- Transmitindo: com os dados processados inicia a etapa de envio dos dados para os repositórios online.

Com base nos parâmetros determinados anteriormente, foram realizadas as coletas dos dados para cada plataforma. O procedimento consistiu em conectar em série com a linha de alimentação um amperímetro, aparelho que mede a corrente (I) do circuito. Além disso, o aparelho utilizado possibilita a medição da tensão (U) elétrica no circuito. Com essas duas grandezas é possível ainda calcular a potência (P) instantânea que está sendo necessária para o funcionamento do circuito, conforme aplicação da equação, demonstrada abaixo:

$$P = U \times I$$

Deve se levar em conta que todo componente elétrico, mesmo que para a medição, terá um consumo, por mais baixo que seja. Para que não houvesse distorção nos valores de leitura de potência foi realizada a aferição do consumo do medidor. Para tal, foi aplicada a fórmula mencionada acima utilizando os valores conforme pode ser visualizado na Figura 29. O valor de tensão é de 5 Vcc e a corrente de 0,02 A, resultando na potência de 0,1 W que será descontada na medição realizada.

**Figura 29 - Consumo do medidor de tensão e corrente**

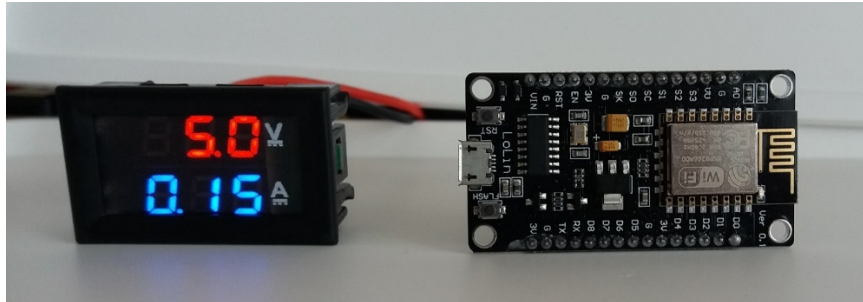


**Fonte: O autor (2018)**



Na Figura 30, é apresentada a plataforma ESP8266 conectada apenas a alimentação de 3.3 Vcc sem qualquer ligação elétrica com sensores drenando energia ou programa rodando em memória.

**Figura 30 - Consumo da placa ESP8266 apenas com alimentação**

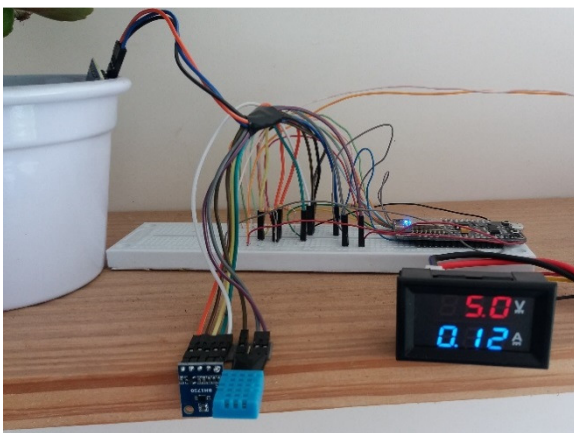


**Fonte: O autor (2018)**

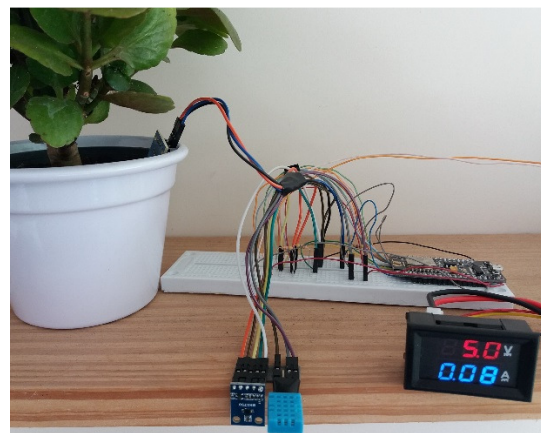
Já na Figura 31 A e B, são apresentadas as correntes da placa ESP8266 em dois momentos distintos, na Figura 31 A a placa está no estado de inicialização, ou em modo de *boot*. De outra forma, na Figura 31 B é apresentada a corrente da placa em modo de transmissão de dados, onde será necessária a conexão com a rede de dados sem fio para o envio dos dados.

Na comparação entre a Figura 30 e a Figura 31 A, houve uma diferença de 0,03 A. Essa diferença se deve a condição da placa no estado ligada e não estando conectada em nenhuma rede de dados sem fio, o *chip* responsável pela varredura de redes aumenta a potência no intuito de expandir o alcance, implicando significativamente no consumo.

**Figura 31 - Consumo da placa ESP8266 em dois estados**



**A) Corrente na fase de *boot***



**B) Corrente na fase de transmissão**

**Fonte: O autor (2018)**



Após realizadas as medições de tensão e corrente nos estados de funcionamento definidos para o ESP8266 foi gerada a Tabela 2. Ao mesmo tempo é possível visualizar a potência consumida por estes estados.

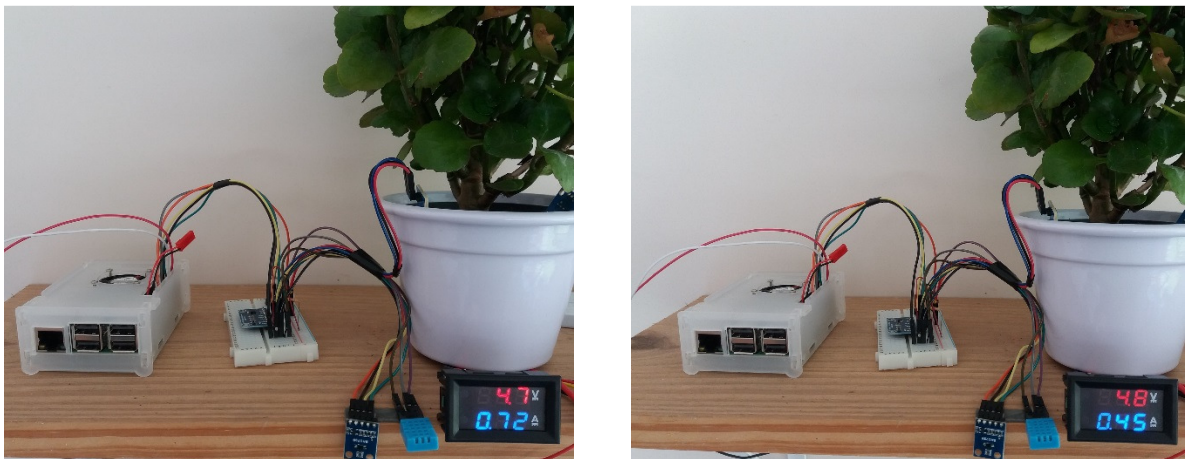
**Tabela 2 - ESP 8266 Tensão versus Corrente**

<b>Estado</b>	<b>Tensão (V)</b>	<b>Corrente (mA)</b>	<b>Potência (W)</b>	<b>Potência (W) sem medidor</b>
Boot	5	120	0,6	0,5
Aguardando	5	50	0,25	0,15
Lendo Sinais	5	60	0,3	0,2
Transmitindo	5	80	0,4	0,3

**Fonte: O autor (2018)**

Utilizando-se a mesma fonte de alimentação. É possível verificar nas medições apresentadas na Figura 32 A, que a tensão e corrente da placa Raspberry Pi 3, possuem um consumo extremamente alto, causando até mesmo uma queda de tensão de 5 Vcc para 4,7 Vcc no estado de *boot*. Já na Figura 32 B estando a placa no estado de transmitindo, o consumo de corrente é menor e a tensão retornou 0,1 Vcc.

**Figura 32 - Consumo da placa Raspberry Pi 3 em dois estados sem ventilador**



**A) Corrente na fase de *boot***

**B) Corrente na fase de transmissão**

**Fonte: O autor (2018)**

Diferentemente do ESP8266, que não possui ventilador para remover o excesso de calor, as medições do Raspberry Pi 3 foram agrupadas em dois conjuntos, com e sem

ventilador. Essa variação se deve em função da possibilidade de trabalhar sem a refrigeração na placa quando seu processador não é muito exigido, gerando assim menor consumo e consequentemente menor dissipação térmica.

Na Tabela 3 é possível visualizar os valores de tensão e corrente coletados sem o ventilador para refrigeração e ao final gerada a potência do produto das duas grandezas.

**Tabela 3 - Raspberry Pi 3 Tensão x Corrente sem ventilador com WiFi**

<b>Estado</b>	<b>Tensão (V)</b>	<b>Corrente (mA)</b>	<b>Potência (W)</b>	<b>Potência (W) sem medidor</b>
Boot	4,7	720	3,384	3,284
Aguardando	4,8	360	1,728	1,628
Lendo Sinais	4,9	370	1,813	1,713
Transmitindo	4,8	450	2,16	2,06
S.O. Encerrado	4,9	150	0,735	0,635

**Fonte: O autor (2018)**

Na Tabela 4 são apresentados os valores de tensão e corrente coletados utilizando o ventilador para refrigeração. Este ventilador ligado constantemente na placa consome 1 W aumentando a potência necessária para o funcionamento prolongado.

**Tabela 4 - Raspberry Pi 3 Tensão x Corrente com ventilador com WiFi**

<b>Estado</b>	<b>Tensão (V)</b>	<b>Corrente (mA)</b>	<b>Potência (W)</b>	<b>Potência (W) sem medidor</b>
Boot	4,7	950	4,465	4,365
Aguardando	4,8	560	2,688	2,588
Lendo Sinais	4,8	570	2,736	2,636
Transmitindo	4,8	650	3,12	3,02
S.O. Encerrado	4,9	300	1,47	1,37

**Fonte: O autor (2018)**

## 6.2 UTILIZAÇÃO DE BATERIA

A utilização de baterias é muito importante quando se deseja utilizar uma aplicação remota, ou seja, sem a alimentação através da rede de energia elétrica. Deste modo, sendo

autônoma, necessita possuir um consumo tão baixo quanto possível para não drenar em pouco tempo a carga disponível na fonte de alimentação.

No intuito de verificar a autonomia dos protótipos, as medições se deram utilizando baterias. Assim, para que o protótipo permaneça com dimensões reduzidas e baixo custo foram utilizadas baterias comuns de 9 V encontradas nas mais diversas aplicações.

A bateria utilizada possui 550 mAh, ou seja, ele pode fornecer em 1 hora 550 miliAmperes para uma tensão de 9 V. Porém, a alimentação das placas será feita com 5 V, assim, foi necessária a redução da tensão utilizando uma fonte. Não foram utilizadas técnicas de software para redução de consumo, a fim de diminuir o consumo das placas.

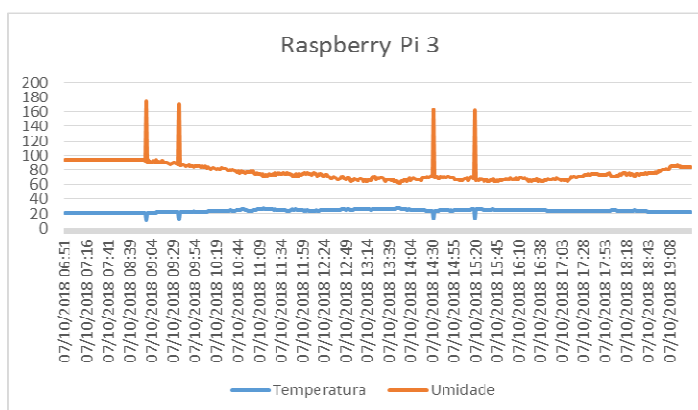
Para o ESP8266, o tempo de funcionamento foi de 36 horas. Com um consumo de 0,15 W e corrente de 50 mA consumiu a bateria neste tempo.

Já para o Raspberry Pi 3, o tempo de funcionamento foi de apenas 1 hora com a utilização do ventilador para refrigeração. Ou seja, o consumo de 2,688 W no estado aguardando, com uma corrente de 560 mA consumiu a bateria neste tempo.

### 6.3 AVALIAÇÃO DOS RESULTADOS

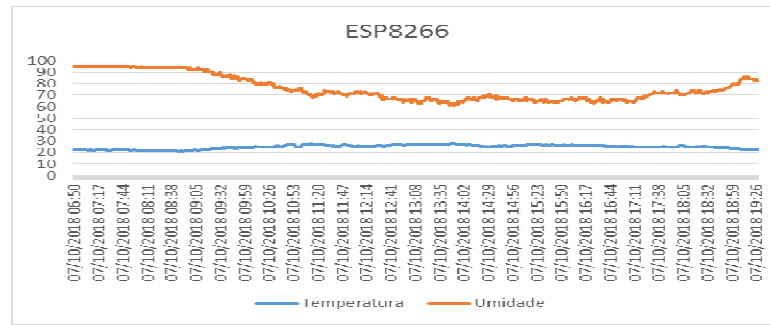
Para validar os dados do experimento apresentado, serão expostos os dados das coletas realizadas com os sensores DHT11 (sensor de temperatura e umidade), sensor de umidade do solo, sensor de luminosidade BH1750. Também será apresentado o resultado do consumo de energia entre as duas plataformas Raspberry Pi e ESP8266. Da mesma forma, apresentar a diferença causada pela zona de corte utilizando o sensor de luminosidade.

**Figura 33 - Raspberry Pi 3 Comparação de temperatura x umidade do ar**



**Fonte: O autor (2018)**

**Figura 34 - ESP8266 Comparação de temperatura x umidade do ar**



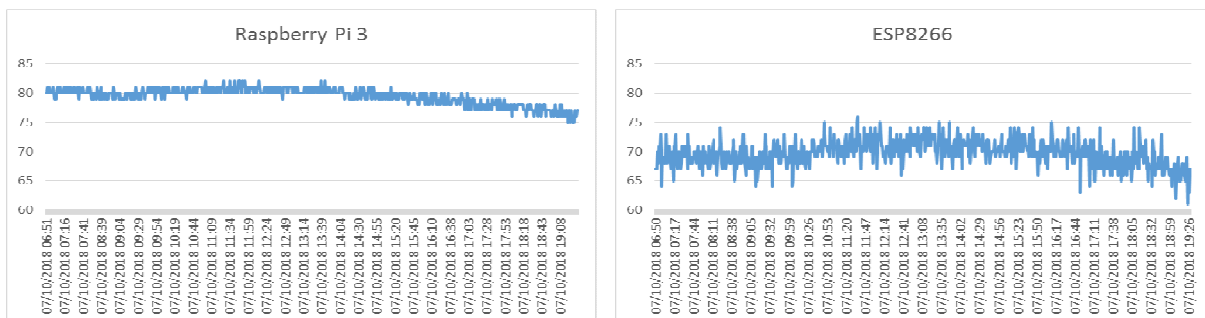
Fonte: O autor (2018)

Conforme pode ser visto nas Figuras 33 e 34, os dados coletados durante o período do dia estiveram dentro da mesma faixa por quase todo o período e apresentaram comportamento parecidos. Porém, a plataforma Raspberry Pi apresentou dados errôneos em determinados períodos extrapolando os valores esperados.

Com o objetivo de solucionar o problema da variação nos valores lidos pelo sensor de DHT11 foi realizada a sua substituição por outro idêntico. Porém sem sucesso, pois apresentou os mesmos valores errôneos em determinados momentos. O passo seguinte foi uma pesquisa em busca de uma nova biblioteca que pudesse ser capaz de eliminar a distorção, mas não foi encontrada nenhuma que solucionasse o problema.

Com o objetivo de verificar a representatividade das amostras com dados incorretos, o dia 07 de outubro foi definido para a coleta dos dados errôneos, tendo desta forma, coletadas 1078 amostras e 4 delas apresentaram erros, representando assim 0,46 % de erros, enquanto que a plataforma ESP8266, não apresentou problemas. Para que estes dados não comprometessem o estudo, na aplicação foi criado um filtro que descarta os valores de umidade e temperatura do ar quando o valor da umidade exceder o valor de 100 %.

**Figura 35 - Comparação de umidade do solo**

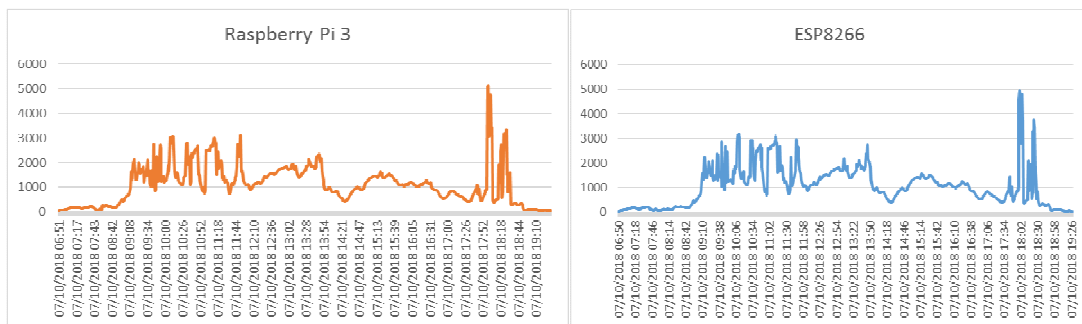


Fonte: O autor (2018)

Conforme apresentadas na Figura 35, as leituras do sensor de umidade de solo, permaneceram com a curva de leitura dos dados ao longo do tempo semelhantes para as duas plataformas. Porém, é possível notar que a faixa de leitura, apresenta diferença de cerca de 10% menor para o ESP8266, isso devido a fatores externos às placas de prototipagem, tendo como causa o posicionamento dos sensores em relação umidade do solo e a acomodação do solo ao redor do sensor, assim, não representando qualquer falha de leitura.

Outro ponto a ser observado nos valores obtidos deste sensor, está na variação de amplitude das medidas dos seus valores, este fato ocorreu em decorrência da depreciação da superfície que está em contato com a umidade do solo, fato este já conhecido em sensores de baixo custo que não possuem proteção apropriada na superfície.

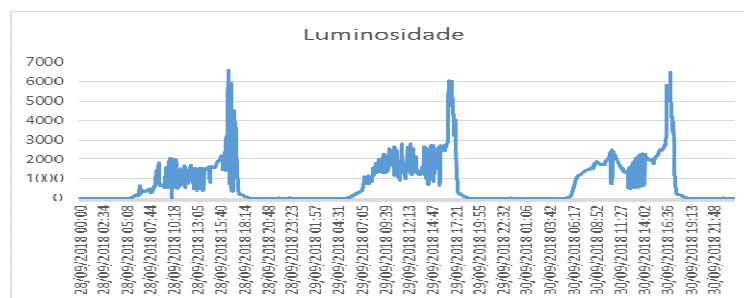
**Figura 36 - Comparação de luminosidade**



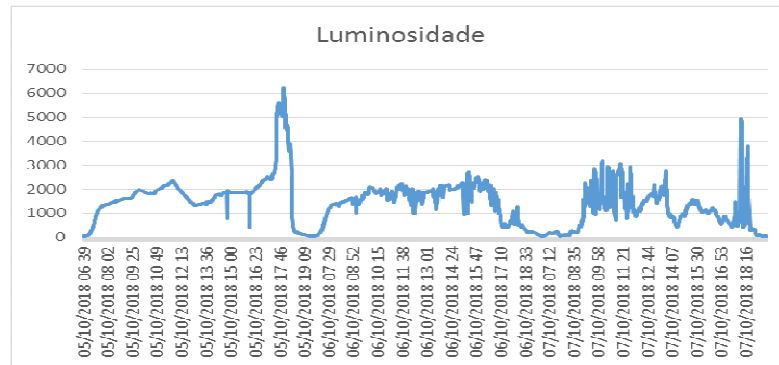
**Fonte: O autor (2018)**

Na Figura 36, são apresentadas as leituras do sensor de luminosidade, que permaneceram semelhantes durante todo o período de medição do dia. Conforme mencionado no início do capítulo 6 Experimento e Análise dos Resultados, é possível notar a amplitude dos valores ao final do dia, quando a maior incidência dos raios solares atinge o experimento e as duas plataformas enviaram os dados semelhantes.

**Figura 37 - Comparação de luminosidade sem zona de corte**



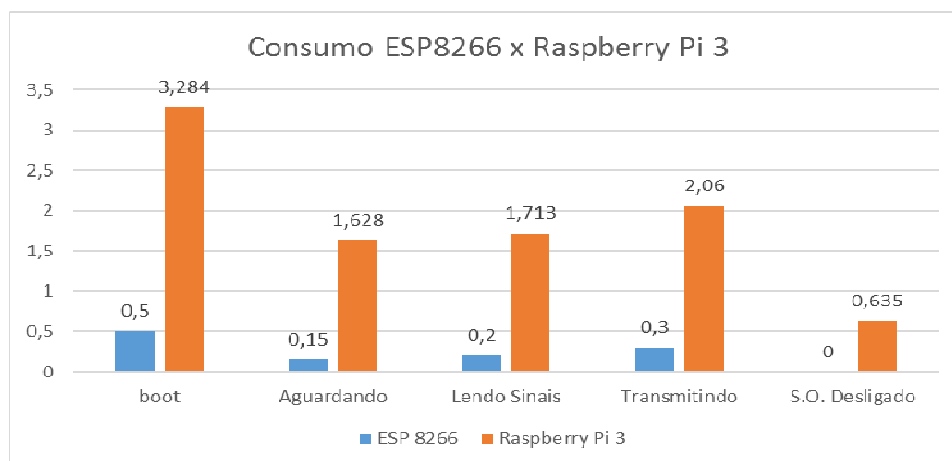
**Fonte: O autor (2018)**

**Figura 38 - Comparação de luminosidade com zona de corte**

**Fonte: O autor (2018)**

Nas Figuras 37e 38, são apresentadas as leituras de luminosidade para três dias cada. Na Figura 37, pode-se perceber que cerca de 50% do tempo de medição não houve variação de luminosidade no período da noite, desta forma, apresentando um consumo passível de reduzir. Após esta análise, foi implementada uma zona de corte para este período.

Os resultados da zona de corte são apresentados na Figura 38, onde é possível identificar uma utilização mais racional do protótipo. Desta forma, conforme o filtro implementado na aplicação, não haverá novas conexões com os sites repositórios enquanto o valor de 50 lux não for excedido, permitindo assim uma autonomia maior quando conectado em baterias.

**Figura 39 - Comparação de consumo entre placa ESP8266 e Raspberry Pi 3**

**Fonte: O autor (2018)**

Após a compilação dos dados apresentados pelas duas plataformas, foi gerado o gráfico apresentado na Figura 39. Nela estão relacionados os dados entre as duas placas nas condições mais favoráveis de consumo. Diante disso, com base nos dados, é possível afirmar

que o consumo nos estados aguardando e transmitindo, sendo os mais utilizados, a placa Raspberry Pi 3 está em cerca de 91% e 85% respectivamente superior.

**Tabela 5 - Envio de dados Raspberry Pi 3 e ESP8266**

	<i>Thingspeak</i>	<i>Google API Spreadsheet</i>
Raspberry Pi 3	1523	1523
ESP8266	1554	1554
	98%	98%

**Fonte: O autor (2018)**

Na Tabela 5, são apresentados os valores de informações inseridas nos *sites* repositórios *Thingspeak* e *Google API Spreadsheet*. Podemos notar que houve uma pequena perda de dados pela plataforma Raspberry Pi 3, de cerca de 2%, não igualando com a plataforma ESP8266 estando nas mesmas condições.

Este capítulo apresentou os resultados do consumo elétrico, o uso dos sensores utilizados e os dados disponibilizados após as leituras dos mesmos e o envio dos dados aos sites repositórios. Com base nos dados gerados pelo experimento nos dois cenários, considero que a plataforma que melhor atende aos objetivos do projeto é a plataforma ESP8266 devido ao menor consumo elétrico, suas dimensões reduzidas, relação custo benefício vantajosa.

## CONSIDERAÇÕES FINAIS

Conforme apresentado no trabalho, foi possível demonstrar a importância deste novo conceito, Internet das Coisas, que nos últimos anos está se disseminando para diversas áreas, seja na agricultura, médica ou meteorológica. Seu principal objetivo é atender áreas que anteriormente não eram atendidas por soluções comerciais.

As novas ferramentas disponíveis, plataformas de prototipagem, redes de conexão, sensores e atuadores estão em pleno desenvolvimento para atender as diversas áreas. Com sua versatilidade e baixo custo é possível desenvolver e implementar projetos customizáveis.

Baseado no que foi apresentado com o presente trabalho, os objetivos propostos, apresentando os conceitos, equipamentos e dispositivos. Além da definição de parâmetros, da implementação utilizando as duas plataformas, coleta de dados e processamento destes, para apresentação dos resultados foram plenamente atendidos.

A partir dos dados que são disponibilizados e compartilhados, nos *sites* repositórios, também é possível expor o conhecimento para pessoas que anteriormente possuíam acesso. Com a internet das coisas *IoT* foi possível a apresentação de uma diversidade de informações para a melhoria do entendimento das coisas que nos cercam.

O presente trabalho foi de grande valia no âmbito de aprendizado pessoal, pois através dele foi possível um amplo estudo nas plataformas e sensores estudados. Além disso, foi possível adquirir conhecimento de uma nova linguagem de programação desconhecida até o desenvolvimento do trabalho.

Houveram muitos desafios a serem enfrentados e que foram com grande esmero solucionados. Podendo citar o problema com a biblioteca do sensor DHT11 para a plataforma ESP8266, que passou a não realizar as leituras corretamente, necessitando um intenso estudo para detectar e substituir a biblioteca com problemas. Em outro momento foi necessária a modificação da autenticação com o *site* repositório *Google API Spreadsheet*, entre tantos outros que só vieram a agregar conhecimento nesta jornada.

Como trabalhos futuros, sugere-se o estudo das características avançadas do protocolo rede, para conexão sem fio WiFi, com o objetivo de trabalhar com as plataformas no modo *softAP + station*. Nesta configuração, é possível criar uma grande rede no formato *Mesh* com diversos protótipos comunicando-se e ampliando o alcance da rede. Desta forma, os dados podem ser retransmitidos para fora do alcance da rede principal.



Com o objetivo de manter a rede funcionando de forma autônoma, é possível considerar a utilização de placas solares permitindo assim funcionamento indeterminados dos diversos dispositivos da rede. Com o custo mais acessível torna-se uma ótima alternativa.

Pensando em autonomia e independência dos protótipos o uso de atuadores com reservatório de água pode ser considerado, pois através do monitoramento da umidade do solo é possível acionar uma saída e liberar uma quantidade de água para dar condições de sobrevivência a planta. Desta forma possuindo um projeto completo.

Conclui-se por fim, que ao ponto que as duas plataformas possuem características diversas, mas puderam executar e atender a todos os parâmetros definidos anteriormente, apresentam-se como grande solução para tarefas que ainda serão exploradas em novos projetos e espera-se que este trabalho possa ser útil tanto para alunos e professores, em âmbito acadêmico, como também para outros interessados nesta área de conhecimento.

## BIBLIOGRAFIA

ARANDA, J. A. S.. **Internet das Coisas: Um Protótipo Usando a Plataforma Arduino e a Placa E-Health Para a Coleta de Sinais Vitais.** 2016. Disponível em: <<http://biblioteca.feevale.br/monografia/MonografiaJorgeASAranda.pdf>>. Acesso em: 17 nov. 2017.

ARDUINO *SOFTWARE IDE.* s. d.. Disponível em: <<https://www.arduino.cc/en/Guide/Environment>>. Acesso em: 05 set. 2017.

ATUADORES e VÁLVULAS. Disponível em: <[http://www.clubedaeletronica.com.br/Automacao/Automa\\_telecurso/autoa05.pdf](http://www.clubedaeletronica.com.br/Automacao/Automa_telecurso/autoa05.pdf)>. Acesso em: 06 set. 2017.

BENCHOF, B.. **New Chip Alert: The ESP8266 Wi-Fi Module (It's \$5).** 2014. Disponível em: <<https://hackaday.com/2014/08/26/new-chip-alert-the-esp8266-wifi-module-its-5/>>. Acesso em: 09 nov. 2017.

BRITO, T. G.. **Porque aprender Python pode te levar mais longe na carreira.** s. d.. Disponível em: <<https://becode.com.br/porque-aprender-python/>>. Acesso em: 15 out. 2017.

CAMPOS, A.. **Do LED ao Arduino: aprendendo eletrônica no século 21.** 2016. Disponível em: <<https://br-arduino.org/2015/05/arduino-bootloader-atmega-standalone.html>>. Acesso em: 26 out. 2017.

CIRIACO, D.. **O que é Raspberry Pi?.** s. d.. Disponível em: <<https://canaltech.com.br/hardware/o-que-e-raspberry-pi/>>. Acesso em: 16 out. 2017.

DANTAS, R. **Eletrônica digital.** 2014. Disponível em: <<http://tecdantas.blogspot.com.br/>>. Acesso em: 06 nov. 2017.

Disponível em: <<http://www.pubnub.com/blog/worldwide-internet-of-things-market->  
EVANS, D. **A Internet das Coisas: Como a próxima evolução da Internet está mudando tudo.** White paper Cisco, Abril de 2011. Disponível em: <[http://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iiot\\_ibsg\\_0411final.pdf](http://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iiot_ibsg_0411final.pdf)>. Acesso em: 09 set. 2017.

<<http://www.pubnub.com/blog/worldwide-internet-of-things-market-exceed-7-trillion/>>. Acesso em: 16 nov. 2017.

FELIZARDO, I.. 2005 . **Processos de soldagem: manual, mecanizado, automatizados.** 2014. Disponível em: <<http://docplayer.com.br/9302533-Processos-de-soldagem-manual-mecanizados-e-automatizados.html>>. Acesso em: 06 nov. 2017.

FIGUEIRÊDO, R.. **Saiba o que é um Raspberry PI e para que ele serve**. 2017. Disponível em: <<http://tudosobreraspberry.info/2017/03/saiba-o-que-e-um-raspberry-pi-e-para-que-ele-serve>>. Acesso em: 05 set. 2017.

JENA, R. R. NodeMCU – ESP8266. 2018. Disponível em: <<http://www.rjprince.com/2018/07/interfacing-node-muc-esp-8266-with-lua.html>>. Acesso em: 04 nov. 2018

KARVINEN, K.; KARVINEN, T.. 2014. **Primeiros Passos com Sensores**. São Paulo, SP, 2014: Novatec, 2017.

LIMA, T.. **Raspberry Pi B+**. 2014. Disponível em: <<https://www.embarcados.com.br/raspberry-pi-modelo-b-plus/>>. Acesso em: 16 out. 2017.

MANCINI, M.. **Internet da Coisas: História, Conceitos, Aplicações e Desafios**. Project Management Institute SP, São Paulo - SP, 2017. Disponível em: <<https://pmisp.org.br/document-repository/acervo/241-internet-das-coisas-historia-conceitos-aplicacoes-e-desafios>>. Acesso em: 06 set. 2017.

MANITEL, P.. O guia básico de uso das GPIOs do ESP8266. 2016.. Disponível em: <<http://pedrominatel.com.br/pt/esp8266/o-guia-basico-de-uso-das-gpios-do-esp8266/>>. Acesso em: 04 nov. 2018.

MORIMOTO, C.. **A Revolução do Raspberry Pi**. 2012. Disponível em: <<http://www.hardware.com.br/artigos/raspberrypi/>>. Acesso em: 15 out. 2017.

NÉRIO, A.. **Teleprocessamento e Redes**. 2003. Disponível em: <<http://www.logicengenharia.com.br/mcamara/alunos/wireless.pdf>>. Acesso em: 15 nov. 2017.

OLIVEIRA, A. H.; NEVES, J. D. B.; REZENDE, T. T.; TEIXEIRA, P. A.. **Aplicações de Automação em IoT – Internet Of Things**. 2016. Disponível em: <[https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiaXRrczXAhUKGZAKHcjoAKwQFggmMAA&url=http%3A%2F%2Fperiodicos.faex.edu.br%2Findex.php%2Ffelocucaao%2Farticulo%2Fdownload%2F104%2Fpdf&usq=AOvVaw26tm sbF\\_KW2orQ2krE2CeE](https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiaXRrczXAhUKGZAKHcjoAKwQFggmMAA&url=http%3A%2F%2Fperiodicos.faex.edu.br%2Findex.php%2Ffelocucaao%2Farticulo%2Fdownload%2F104%2Fpdf&usq=AOvVaw26tm sbF_KW2orQ2krE2CeE)>. Acesso em: 15 nov. 2017.

OLIVEIRA, G.. **NodeMCU – Uma plataforma com características singulares para seu projeto IoT**. s.d.. Disponível em: <<http://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot/>>. Acesso em: 04 nov. 2018.

OLIVEIRA, S. de. **Internet das Coisas com ESP8266, Arduino, e Raspberry PI**. São Paulo, SP, 2017: Novatec, 2017.

PATSKO, L. F.. **Tutorial Aplicações, Funcionamento e Utilização de Sensores**. 2006. Disponível em: <[http://www.maxwellbohr.com.br/downloads/robotica/mec1000\\_kdr5000/tutorial\\_eletronica\\_-\\_aplicacoes\\_e\\_funcionamento\\_de\\_sensores.pdf](http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf)>. Acesso em: 08 set. 2017.

PINHEIRO, J. M. S.. **As Rede com ZigBee**. 2004. Disponível em: <[http://www.projetoderedes.com.br/artigos/artigo\\_zigbee.php](http://www.projetoderedes.com.br/artigos/artigo_zigbee.php)>. Acesso em: 15 nov. 2017.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho acadêmico**. 2. ed. Novo Hamburgo – RS: Universidade Feevale. 2013. Disponível em: <<http://www.feevale.br/Comum/midias/8807f05a-14d0-4d5b-b1ad-1538f3aef538/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>>. Acesso em: 02 set. 2017.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2 ed. Novo Hamburgo, RS: Feevale. 2013. 276 p. ISBN 9788577171583. Disponível em: <<http://www.feevale.br/cultura/editora-feevale/metodologia-do-trabalho-cientifico---2-edicao>>. Acesso em: 02 set. 2017.

PUBNUB. **Worldwide Internet of Things Market to Exceed \$7 Trillion by 2020**. 2014. RASPBERRY PI. s. d.. Disponível em: < <https://www.raspberrypi.org/help/faqs/>>. Acesso em: 14 out. 2017.

SABERELETRONICA. 2011. Disponível em: <[http://www.sabereletronica.com.br/files/image/SE448\\_CursoZ\\_F01.jpg](http://www.sabereletronica.com.br/files/image/SE448_CursoZ_F01.jpg)>. Acesso em: 15 nov. 2017.

SANTOS, L. B.. **Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores**. 2014. Disponível em: <<http://repositorio.uniceub.br/bitstream/235/5938/1/21016773.pdf>>. Acesso em: 15 nov. 2017.

SEIDEL, Á. R.. **Instrumentação Aplicada**. 2011. Disponível em: <[http://estudio01.proj.ufsm.br/cadernos\\_automacao/setima\\_etapa/instrumentacao\\_aplicada\\_2012.pdf](http://estudio01.proj.ufsm.br/cadernos_automacao/setima_etapa/instrumentacao_aplicada_2012.pdf)>. Acesso em: 04 nov. 2017.

SOUZA, F. **Raspberry Pi B+**. 2016. Disponível em: <<https://www.embarcados.com.br/hardware-da-raspberry-pi-3/>>. Acesso em: 16 out. 2017.

TENFEN, C. R. **Projeto de um Sistema Remoto para Aquisição de Dados Meteorológicos**. 2013. Disponível em: <<http://siaibib01.univali.br/pdf/CI%C3%A1udio%20Roberto%20Tenfen.pdf>>. Acesso em: 17 nov. 2017.

THOMSEN, A. **Qual módulo ESP8266 comprar?**. 2016. Disponível em: <<https://www.filipeflop.com/blog/qual-modulo-esp8266-comprar/>>. Acesso em: 09 nov. 2017.

VELLOSO, F.. **Sinal analógico ou digital?** Entenda as tecnologias e suas diferenças. 2014. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2014/12/sinal-analogico-ou-digital-entenda-tecnologias-e-suas-diferencas.html>>. Acesso em: 06 nov. 2017.

WENDLING, M.. **Sensores**. Universidade Estadual Paulista – Guaratinguetá, SP, 2010. Disponível em: <<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>. Acesso em: 06 set. 2017.

ZANELLA, A., BUI, N., CASTELLANI, A., VANGELISTA, L., ZORZI, M. 2014. **“Internet of Things for smart cities”**, IEEE Internet of Things Journal, vol. 1.