

# Construção de módulos baseados em Internet das Coisas para coleta de dados e gerenciamento de informações

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Roberta Spolon

# Sumário

---

## 1. Introdução

## 2. Fundamentação Teórica

- Internet das Coisas
- Message Queuing Telemetry Transport
- Raspberry Pi
- Arduino
- NodeMCU
- Sensores
- Ferramentas Utilizadas

# Sumário

---

## 3. Desenvolvimento

- Planejamento
- Configuração do Raspberry Pi
- Programação e Montagem dos Módulos
- Captação de Dados no Home Assistant
- Integração com Dashboard para Controle de Projetos

## 4. Conclusão

## 5. Trabalhos Futuros

## 6. Referências

# Introdução

---

# Introdução

---

Diante dos avanços tecnológicos a aplicação do conceito de Internet das Coisas permitiu que os mais diversos itens sejam capazes de se conectar a internet e possam coletar dados sem ajuda de humanos e enviá-los na rede para que sejam consumidos por outros dispositivos (ASHTON, 2009).

Com a queda do custo de processamento, foram desenvolvidos componentes eletrônicos de baixo custo e pequeno porte que possibilitaram a criação de diversos dispositivos IoT (*Internet of Things*). De acordo com Lueth (2018), mais de 17 bilhões de aparelhos se encontram conectados a internet e 7 bilhões deles são apenas dispositivos IoT.

# Introdução

---

Essa tecnologia permitiu um avanço nas mais diversas áreas, como por exemplo no monitoramento de ambientes de desenvolvimento, onde as condições climáticas e outros dados podem ser captados para aprimorar um ambiente de trabalho, assim diminuindo o número de reclamações e a ociosidade entre os trabalhadores (ROELOFSEN, 2002).

Dessa forma, com o desenvolvimento de módulos capazes de coletar esses dados e disponibiliza-los em um único servidor para que possam ser acessados por outras aplicações, pode-se monitorar um ambiente através de alguma aplicação que acesse esses dados e os mostre em uma interface gráfica como por exemplo um Dashboard.

Assim, esse trabalho propôs a construção de três módulos para captar dados e enviá-los a internet para serem utilizados por outras aplicações.

# Fundamentação Teórica

---

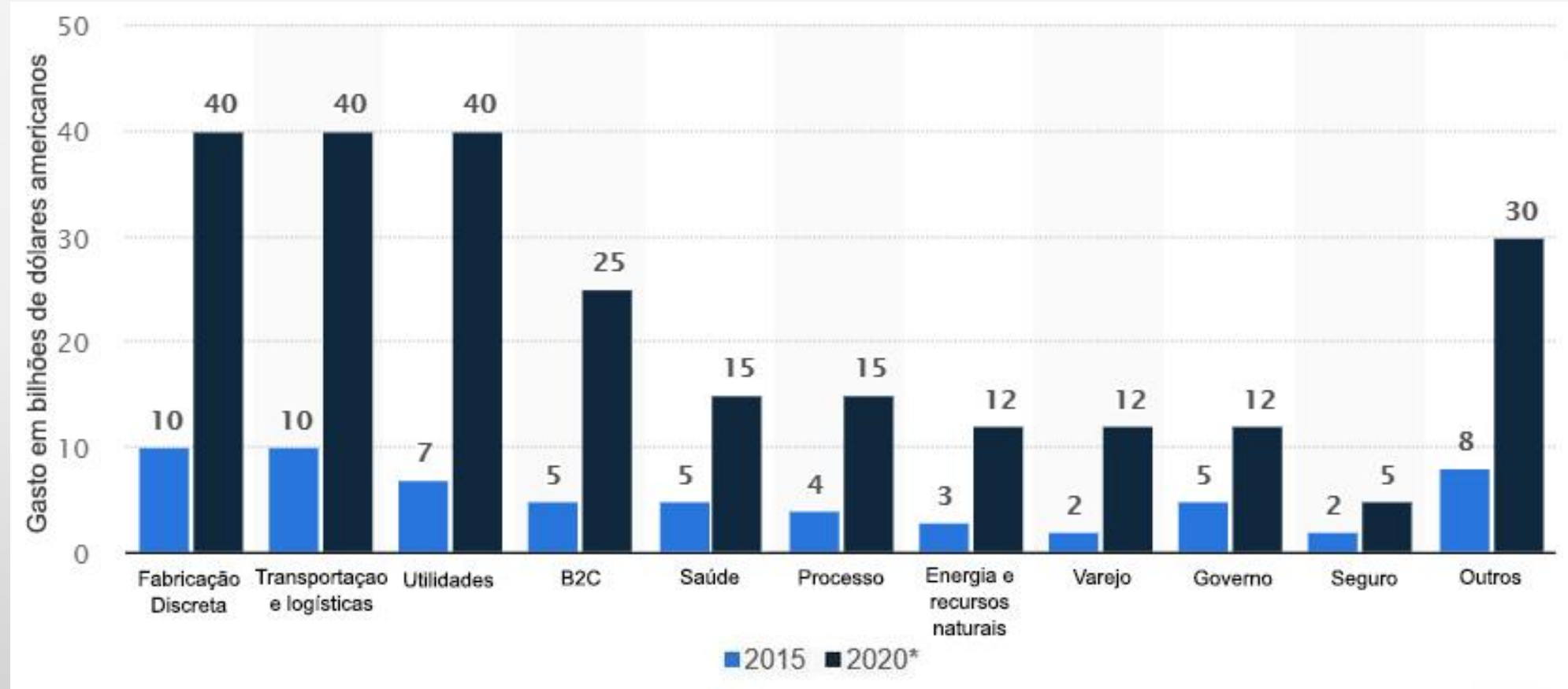
# Internet das Coisas

O crescimento na área de IoT tem alavancado um imenso investimento nesse mercado, o que tem despertado cada vez mais o interesse de desenvolvedores por essa área. Em 2015 um mapeamento da indústria de desenvolvimento de software estimou que 19% de um universo com 19 milhões de desenvolvedores estariam trabalhando em projetos relacionados a essa área (THIBODEAU, 2015).

Com um alto investimento em IoT no setor da saúde, foi criada uma ramificação da área conhecida como *Internet of Medical Things* (IoMT), que visa o estudo e desenvolvimento de técnicas e dispositivos que auxiliem no monitoramento da saúde da população (MARR, 2018). Incluindo o setor da saúde, os gastos com IoT em diversos setores irão superar o valor de 15 bilhões de dólares americanos até 2020 como pode ser visto na Figura 1 (STATISTA, 2018).



Figura 1 - Gastos com Internet das Coisas ao redor do mundo em 2015 e 2020



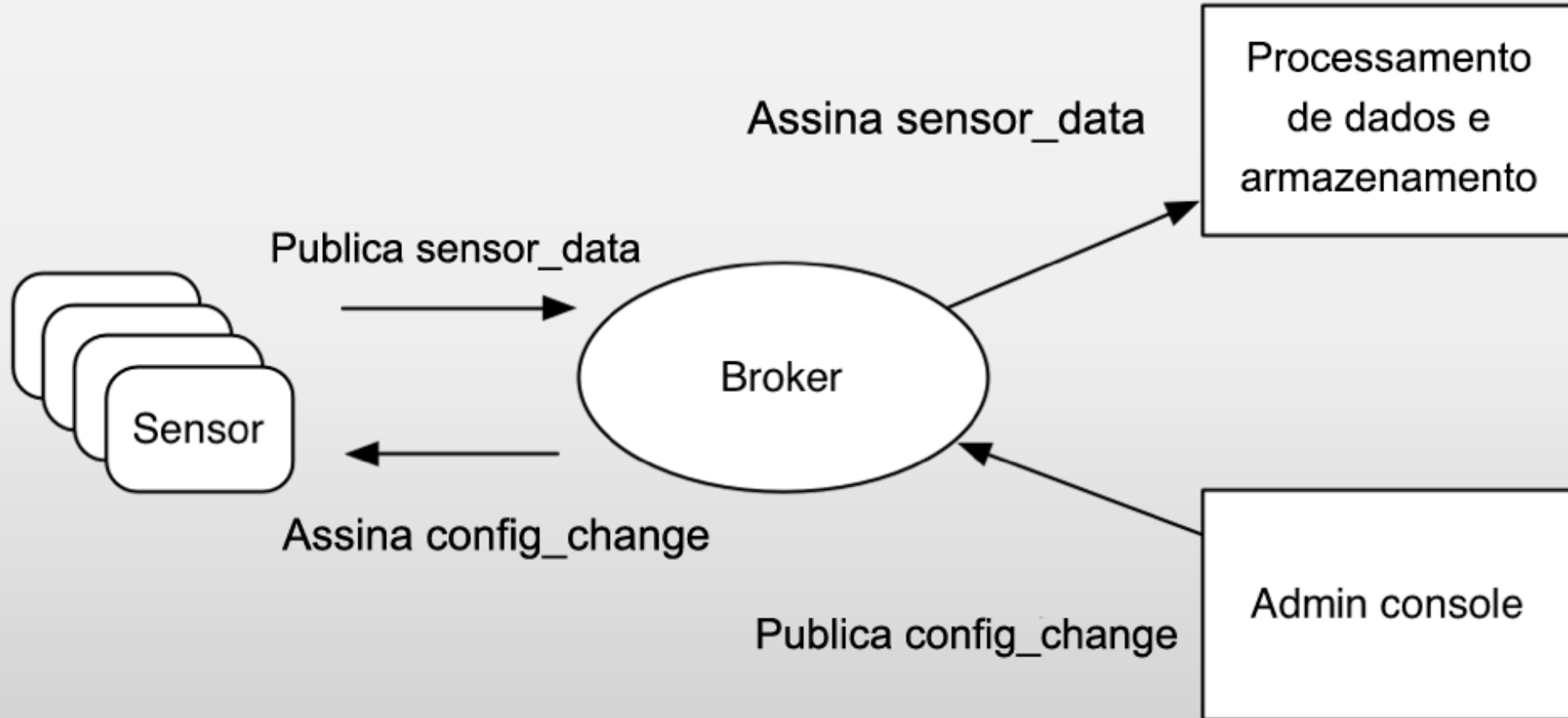
Fonte: (STATISTA, 2018, tradução nossa)

# *Message Queuing Telemetry Transport*

Criado pela *International Business Machines* (IBM), o protocolo *Message Queuing Telemetry Transport* (MQTT) utiliza um modelo de publicação e assinatura e tornou-se oficialmente um padrão aberto da *Organization for The Advancement of Structured Information Standards* (OASIS) com suporte nas mais diversas linguagens de programação, usando implementações de software livre (YUAN, 2017).

O modelo no qual é baseado funciona a partir de dois tipos de entidades, o cliente, que pode publicar mensagens ou assinar determinados tópicos para recebe-las, e o *broker*, que é o servidor responsável por receber todas as mensagens publicadas através dos clientes e de encaminha-las para todos aqueles que tiverem assinado o tópico na qual a mensagem foi publicada. Esse modelo é exemplificado através da Figura 2.

Figura 2 - Modelo MQTT de publicação e assinatura para sensores IoT



Fonte: (YUAN, 2017, tradução nossa)

# Raspberry Pi

---

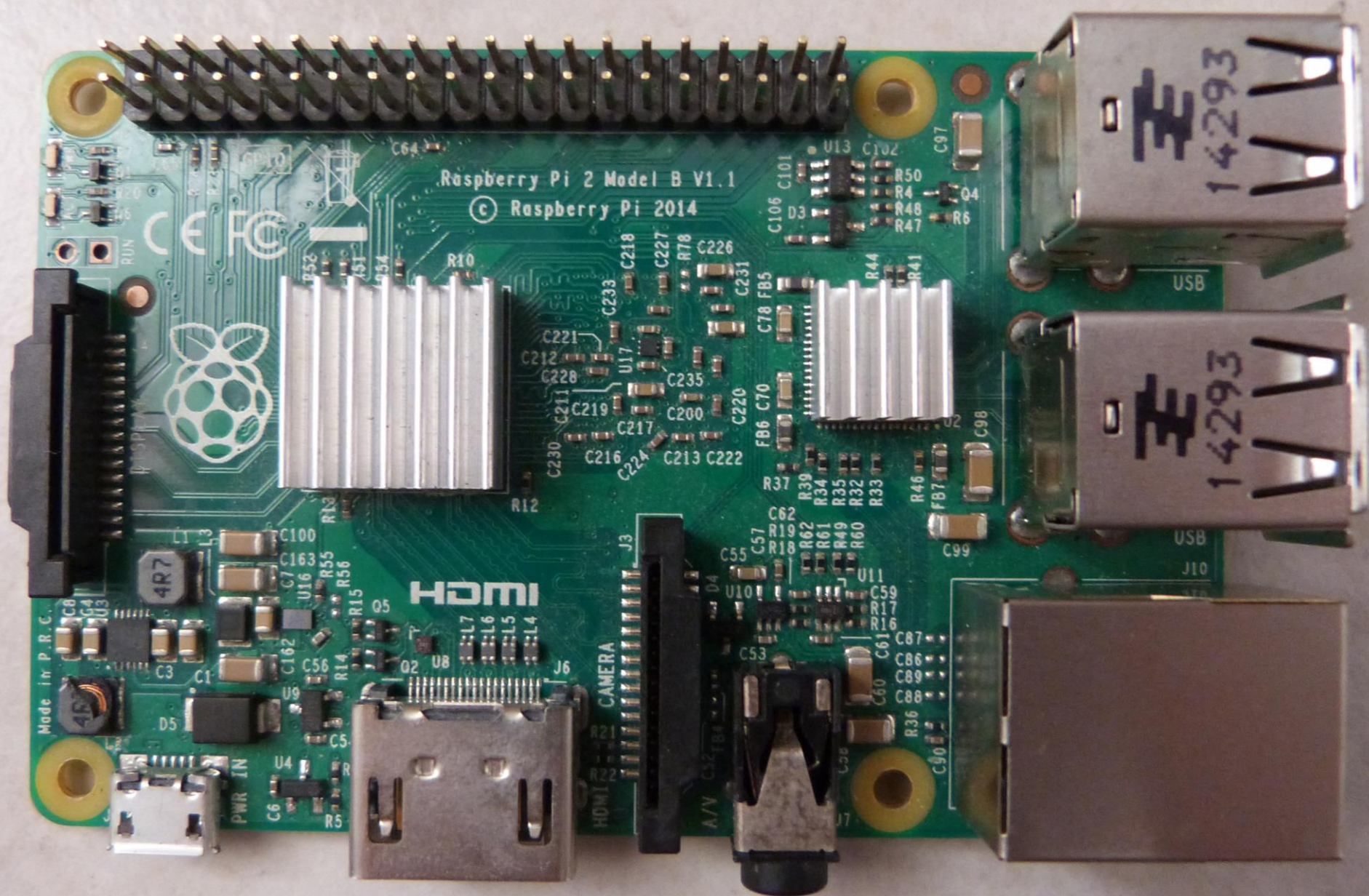
O Raspberry Pi é um computador de baixo custo que possui o tamanho de um cartão de crédito, apresentando entradas para periféricos como monitor, teclado, mouse e até mesmo fones de ouvido. Contém também pinos de entrada e saída digitais para que possam ser conectados sensores e outros dispositivos. Sua capacidade de funcionamento e quantidade de funcionalidades de hardware variam entre cada modelo (Figura 3). O sistema operacional recomendado para ser utilizado no dispositivo é o Raspbian, o qual é baseado na distribuição Linux chamada Debian (RASPBIAN, 2012). A Figura 4 mostra o Raspberry Pi 2 Modelo B.

Figura 3 - Comparativo entre os modelos de Raspberry Pi

Versão	2 Modelo B	3 Modelo B	3 Modelo B+
Memória RAM	1 GB	1 GB	1 GB
Processador	ARMv7 quad-core	ARMv8 quad-core	ARMv8 quad-core
Velocidade CPU	900 MHz quad-core single-core	1.2 GHz MHz quad-core	1.4 GHz quad-core
Portas USB	4	4	4
<i>Ethernet</i>	Sim	Sim	Sim
WLAN	Não	Sim	Sim

Fonte: elaborado pelo autor.

Figura 4 – Raspberry Pi 2 Modelo B



# Arduino

---

O Arduino é uma plataforma eletrônica *open-source* composta por uma placa de hardware, a qual pode ser facilmente programada através da linguagem de programação C++, e um software para desenvolvimento. As placas Arduino são de baixo custo e podem ser facilmente integradas com os mais diversos tipos de sensores e dispositivos eletrônicos capazes de fornecer dados, existem diversos modelos onde cada um deles possui uma grande variedade de aplicações, como pode ser visto na Figura 5.

Nenhum dos modelos possui acesso à internet, para que as placas possam se conectar a rede, é necessário que integrem componentes como o Wifi Shield CC3000 ou o ESP8266 ESP-01 à placa Arduino (THOMSEN, 2014; THOMSEN, 2015).

Figura 5 – Comparativo entre diferentes modelos de placas Arduino

Versão	Memória	Pinos Digitais	Pinos Analógicos	Clock
UNO	32Kb	14	6	16Mhz
MEGA 2560	256KB	54	16	16Mhz
NANO	16 ou 32Kb	14	8	12Mhz

Versão	Tensão de Operação
UNO	5V
MEGA 2560	5V
NANO	5V

Fonte: elaborado pelo autor



# NodeMCU

---

O NodeMCU é uma placa de desenvolvimento que possui um módulo ESP8266 12-E já integrado, o mesmo permite a comunicação em uma rede WLAN de internet. A placa possui todos os componentes necessários para a utilização do módulo wireless, dessa forma sua programação se torna simples e bastante similar aos diversos modelos de placa Arduino (MURTA, 2018).

A placa dispõe de 10 pinos *General Purpose Input/Output* (GPIO), uma velocidade de clock de até 160 megahertz e uma memória interna de até 4 megabytes.

# Sensores

---

Para o presente trabalho foram pesquisados sensores de temperatura, umidade, luminosidade e detecção de estado de abertura de porta. Dessa forma foi realizado o levantamento dos modelos e para os sensores que possuíam 3 modelos ou mais foram montadas tabelas comparativas para melhor visualizar suas diferenças. A Figura 6 mostra a comparação entre os sensores de temperatura e a Figura 7 é referente aos sensores de umidade.

Figura 6 - Comparativo entre sensores de temperatura

Modelo	Amplitude de Alcance	Precisão	Preço
LM35	-55°C à 150°C	0.5°C para mais ou menos	R\$9,90 <sup>1</sup> à R\$10,78 <sup>2</sup>
DS18B20	-55°C à 125°C	0.5°C para mais ou menos	R\$7,80 <sup>3</sup> à R\$39,90 <sup>4</sup>
DHT11	0°C à 50°C	2°C para mais ou menos	R\$6,80 <sup>5</sup> à R\$25,00 <sup>6</sup>

Fonte: elaborado pelo autor

Figura 7 - Comparativo entre sensores de umidade

Modelo	Amplitude de Alcance	Precisão	Preço
HS1101	1% à 99%	2% para mais ou menos	Indisponível
HTU21D	0% à 100%	2% para mais ou menos	R\$16,46 <sup>1</sup> à R\$39,63 <sup>2</sup>
DHT11	20% à 80%	5% para mais ou menos	R\$6,80 <sup>3</sup> à R\$25,00 <sup>4</sup>

Fonte: elaborado pelo autor

# Sensores

---

A captura do estado de porta aberta pode ser realizada através do sensor Intruder F105547, o qual envia seus dados através de rádio frequência de 433 mega-hertz, ou pode ser feita através de um modelo genérico conhecido como *reed switch* magnético de porta, que consiste na combinação de duas lâminas de material ferro magnético que se atraem ao receberem influência de um campo magnético (PICKERINGRELAY, 2018).

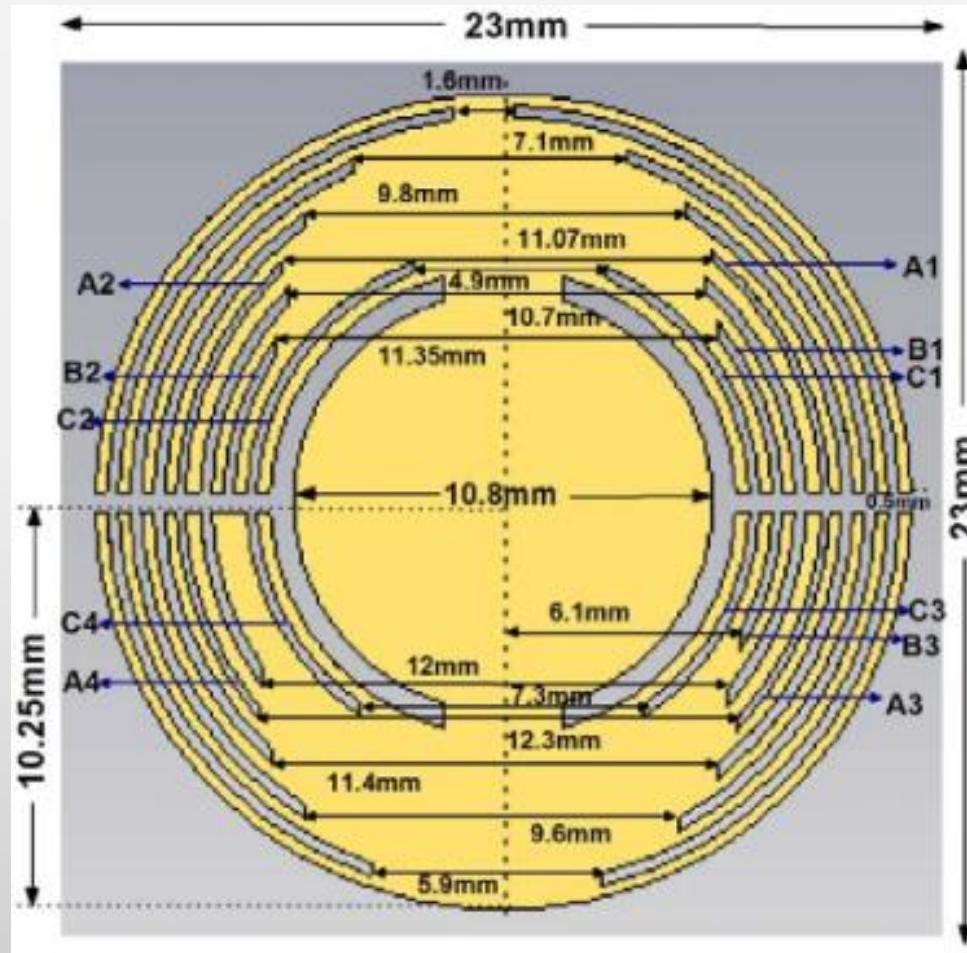
Já a captura de luminosidade pode ser feita com o sensor TSL256, que utiliza um fotodiodo capaz de obter a frequência emitida pela luz visível aos olhos humanos, retornando um sinal digital de saída facilitando seu uso, porém possui um preço bem mais elevado. Em contrapartida, o módulo sensor de luminosidade fotossensitivo, composto por um Resistor Dependente de Luz (LDR), tem o valor de sua resistência alterada conforme a quantidade incidente de luz (GIBBS, 2009), assim em sua saída digital pode-se obter se o ambiente está iluminado ou não (USINAINFO, s.d).

# Trabalhos Correlatos

---

O projeto apresentado por Satti et al. (2018) utiliza uma tecnologia recente conhecida como Chipless Radio Frequency Identification (RFID sem chip), a qual faz uso de um radiador de formato circular de cobre com pequenos sulcos que são preenchidos com sensores químicos capazes de captar a temperatura e a umidade para formar uma tag (Figura 8) apta a ser estimulada através da emissão de rádio frequência sendo suficiente para que a mesma possa emitir de volta os dados coletados. Assim, os dados podem ser coletados por um dispositivo receptor desse tipo de frequência e emitidos através da internet para monitorar a temperatura e umidade.

Figura 8 – Design datagRFID sem chip capaz de medir temperatura e umidade



Fonte: (SATTI et al., 2018)

# Ferramentas Utilizadas

---

- PlatformIO
- Docker
- Node.js e Yarn
- Home Assistant

# Desenvolvimento

---





# Planejamento

---

Inicialmente foi desenvolvido um diagrama que represente o sistema como um todo, que pode ser visto na Figura 9.

Para criação dos módulos IoT, foi elaborado um diagrama para representar a conexão entre cada componente eletrônico. Criou-se um primeiro esboço a mão para pensar na melhor distribuição dos pinos disponíveis na placa NodeMCU para que os sensores pudessem ser conectados, a maneira de disposição e conexão dos elementos permite que todos os sensores utilizem uma única placa ou sejam unificados em uma só (Figura 10).

Em seguida foram elaborados os diagramas finais digitais para os módulos de temperatura e umidade (Figura 11), luminosidade (Figura 12) e estado de abertura de porta (Figura 13).

Figura 9 – Arquitetura de funcionamento geral

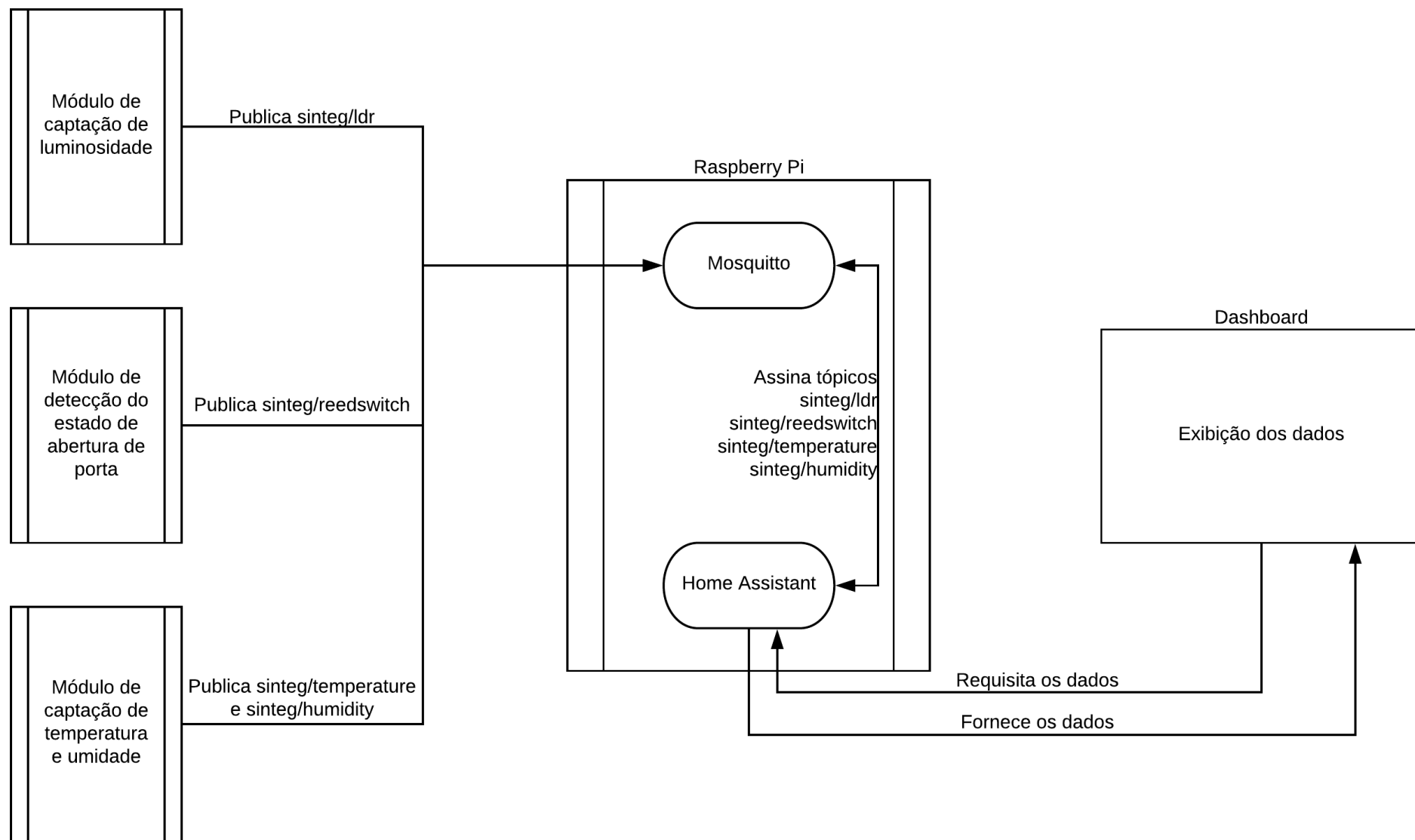
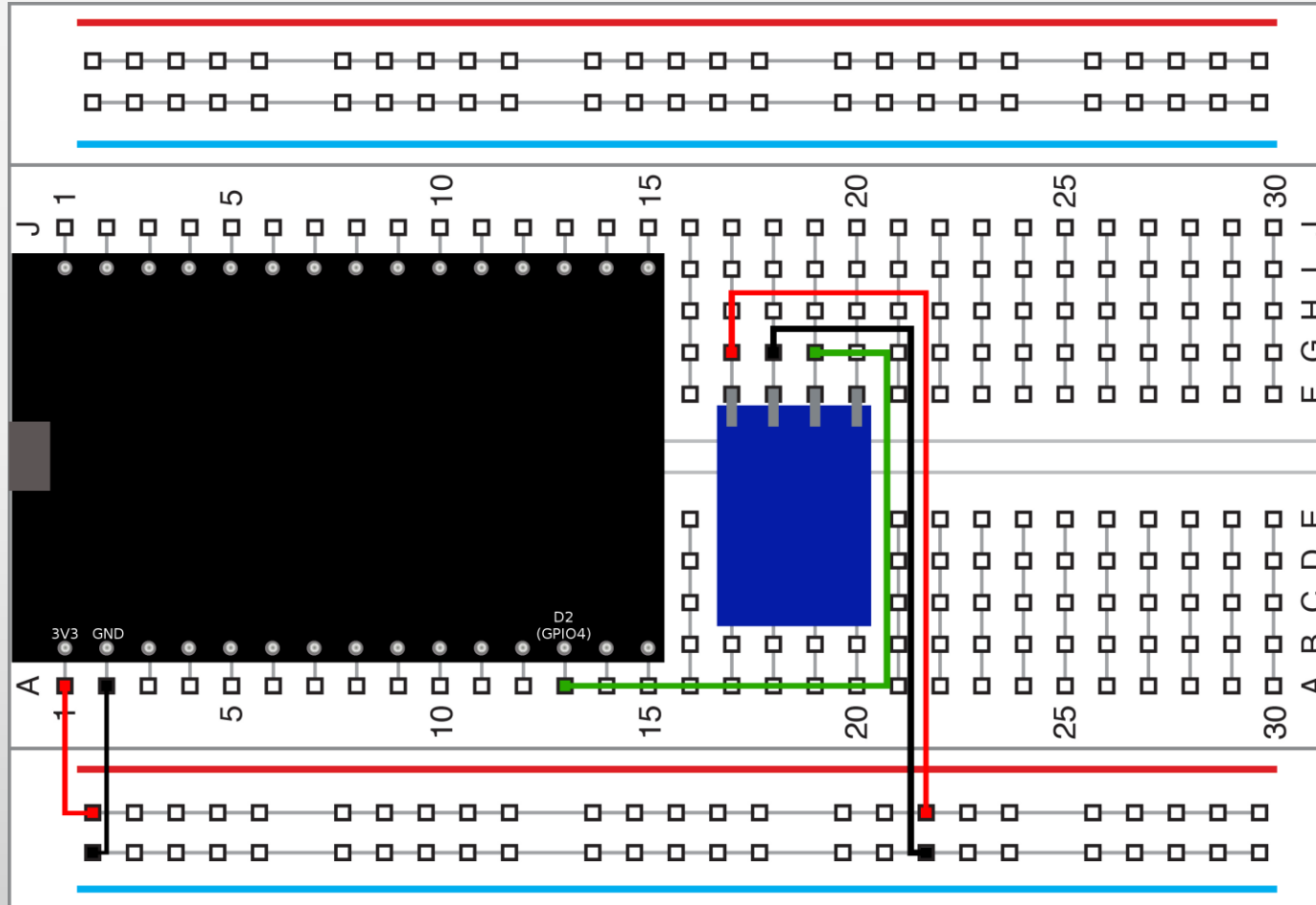




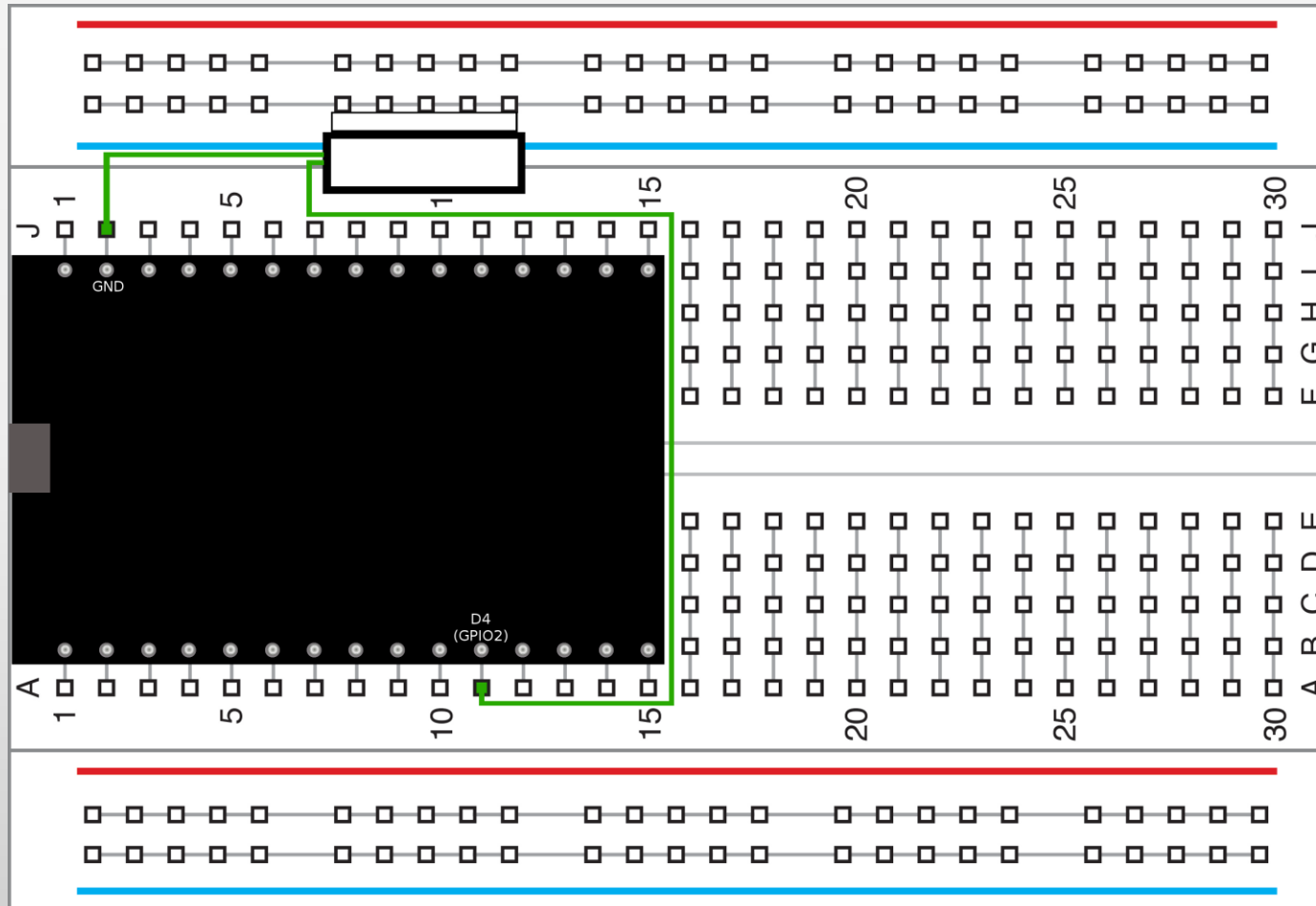


Figura 12 – Diagrama de conexão do sensor de luminosidade



Fonte: elaborado pelo autor

Figura 13 – Diagrama de conexão do sensor para captação do estado de abertura de porta



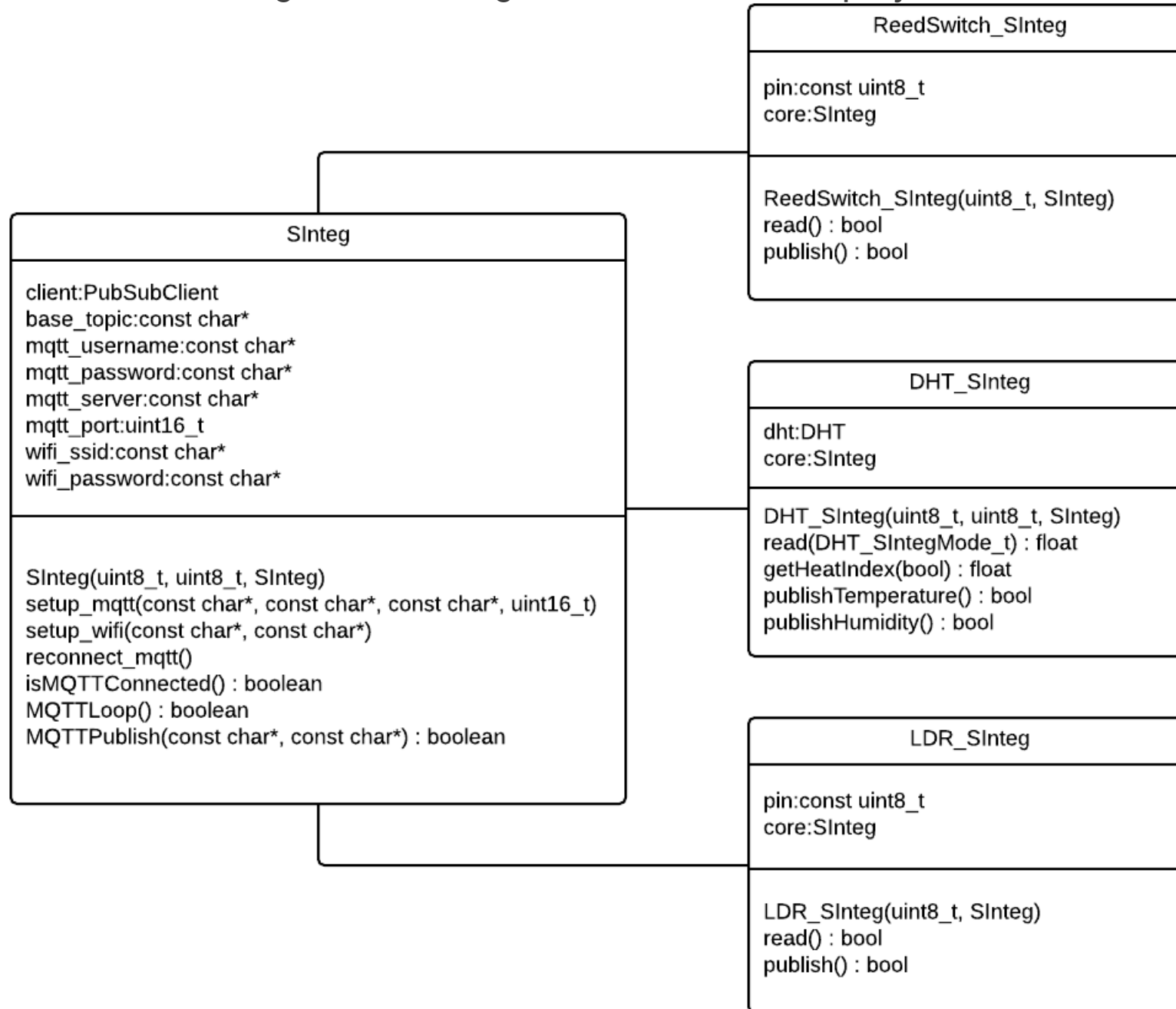
Fonte: elaborado pelo autor

# Planejamento

A programação da placa NodeMCU é realizada utilizando a linguagem C++. Por ser orientada a objetos, pode-se criar classes, as quais são consideradas formas para criação de objetos, onde cada uma contém funcionalidades específicas. Utilizando esse raciocínio foi desenvolvida uma classe para cada sensor, de forma que cada uma possuísse um algoritmo diferente para leitura dos dados, e uma classe que sirva como núcleo contendo as configurações e funções básicas do microcontrolador (Figura 14).

Para poder simular um ambiente similar ao disponibilizado pelo sistema operacional do Raspberry Pi, o desenvolvimento de todas as aplicações que seriam utilizadas nesse computador foram definidas em um único arquivo *docker-compose.yaml* para que cada serviço rode como um contêiner diferente, assim garantindo a não interferência de qualquer que seja a origem. A fim de melhor organização, foi definida uma hierarquia de pastas que contém todos os arquivos necessários para a execução das imagens Figura 15.

Figura 14 – Diagrama de classes do projeto



Fonte: elaborado pelo autor



Figura 15 – Hierarquia de Pastas



Fonte: elaborado pelo autor

# Configuração do Raspberry Pi

Para alcançar o objetivo de disponibilizar os dados captados para outras aplicações, foi escolhida a utilização da plataforma Home Assistant, que alcança esse objetivo por meio de uma REST API para que os dados sejam acessados. As informações também podem ser visualizadas através de uma interface gráfica disponibilizada pela plataforma.

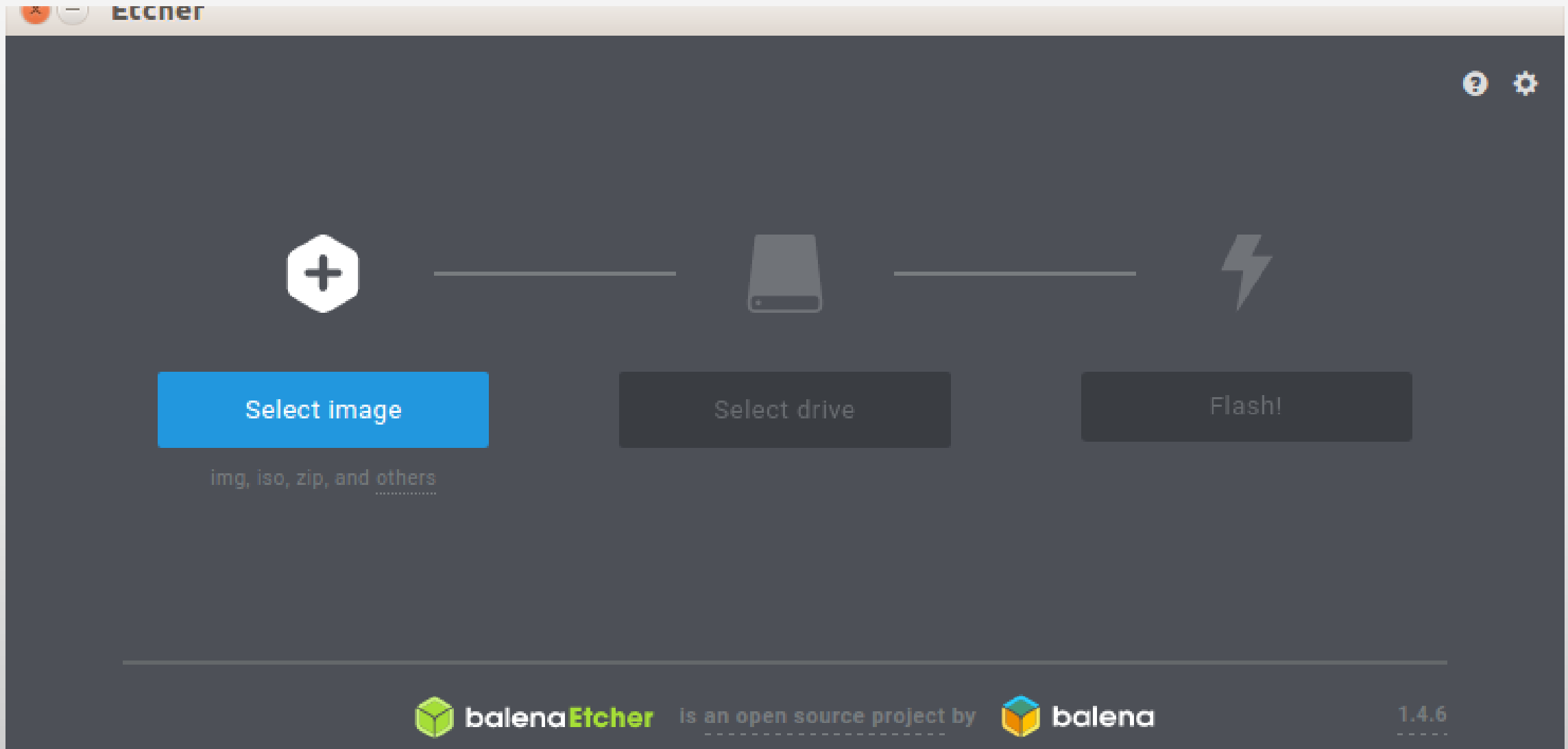
O modelo escolhido para atuar como servidor MQTT e para rodar a aplicação Home Assistant foi o Raspberry Pi 2 Modelo B

# Instalação do Sistema Operacional

O Raspberry Pi utiliza como memória um cartão do tipo microSD de 8 gigabytes, devido a isso a instalação do sistema operacional se dá através da inserção do cartão em um computador para gravar os arquivos necessários.

Com a utilização de um software específico, chamado Etcher (Figura 16), para gravação de imagens de SO em cartões de memória, foi possível instalar facilmente o sistema Raspbian no cartão.

Figura 16 – Tela principal do software Etcher



Fonte: elaborado pelo autor

# Desenvolvimento do Servidor MQTT

Para desenvolver o servidor MQTT foi utilizado o *broker* já disponibilizado pela fundação Eclipse, conhecido como Mosquitto. Sua configuração é dada através de um arquivo base de configuração chamado *mosquitto.conf*. O programa foi instalado em um computador pessoal para analisar seu funcionamento e em seguida foi configurado um container Docker no arquivo *docker-compose.yaml* (Figura 17) para rodar o servidor de forma isolada no Raspberry Pi.

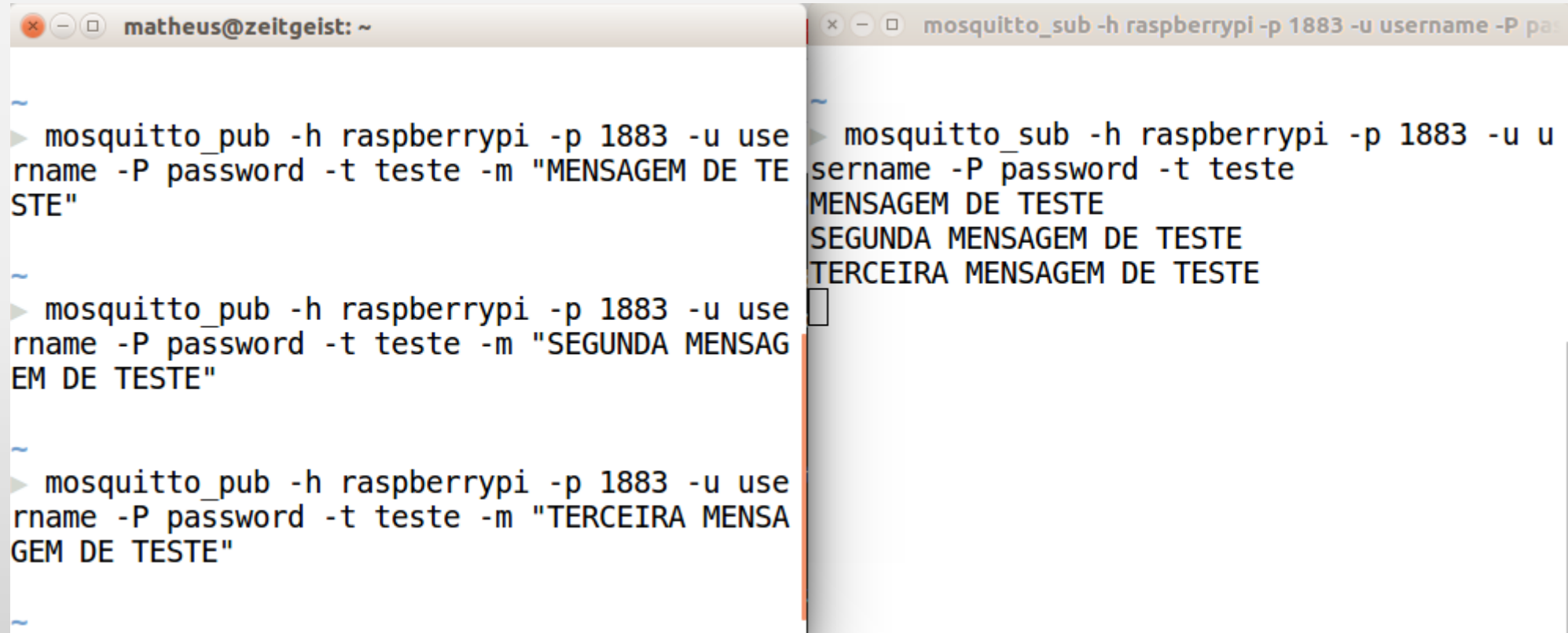
Em seguida foram realizados testes de envio e recebimento de mensagens através do Raspberry Pi (Figura 18).

Figura 17 – Configuração do container para o servidor MQTT  
**mosquitto:**

```
container_name: mosquitto
build:
  context: ./mosquitto
  dockerfile: Dockerfile
  args:
    MQTT_USERNAME: ${MQTT_USERNAME}
    MQTT_PASSWORD: ${MQTT_PASSWORD}
volumes:
  - ./mosquitto:/mosquitto
network_mode: host
```

Fonte: elaborado pelo autor

Figura 18 – Teste de funcionamento do servidor MQTT



The image shows two terminal windows side-by-side. The left window, titled 'matheus@zeitgeist: ~', shows three sequential commands for publishing MQTT messages. The right window, titled 'mosquitto\_sub -h raspberrypi -p 1883 -u username -P pa...', shows the corresponding subscription output for the first message.

```
matheus@zeitgeist: ~  
~  
> mosquitto_pub -h raspberrypi -p 1883 -u use  
rname -P password -t teste -m "MENSAGEM DE TE  
STE"  
~  
> mosquitto_pub -h raspberrypi -p 1883 -u use  
rname -P password -t teste -m "SEGUNDA MENSAG  
EM DE TESTE"  
~  
> mosquitto_pub -h raspberrypi -p 1883 -u use  
rname -P password -t teste -m "TERCEIRA MENSA  
GEM DE TESTE"  
~  
mosquitto_sub -h raspberrypi -p 1883 -u u  
sername -P password -t teste  
MENSAGEM DE TESTE  
SEGUNDA MENSAGEM DE TESTE  
TERCEIRA MENSAGEM DE TESTE
```

Fonte: elaborado pelo autor

# Instalação do Home Assistant

Para instalação do Home Assistant no Raspberry Pi, foi utilizado um container configurado no arquivo *docker-compose.yaml* de forma similar ao servidor MQTT. Seguindo o esquema de hierarquia de pastas demonstrado na seção de planejamento, foram criados arquivos para configuração dos sensores e dos componentes essenciais, sendo eles o mqtt e api.

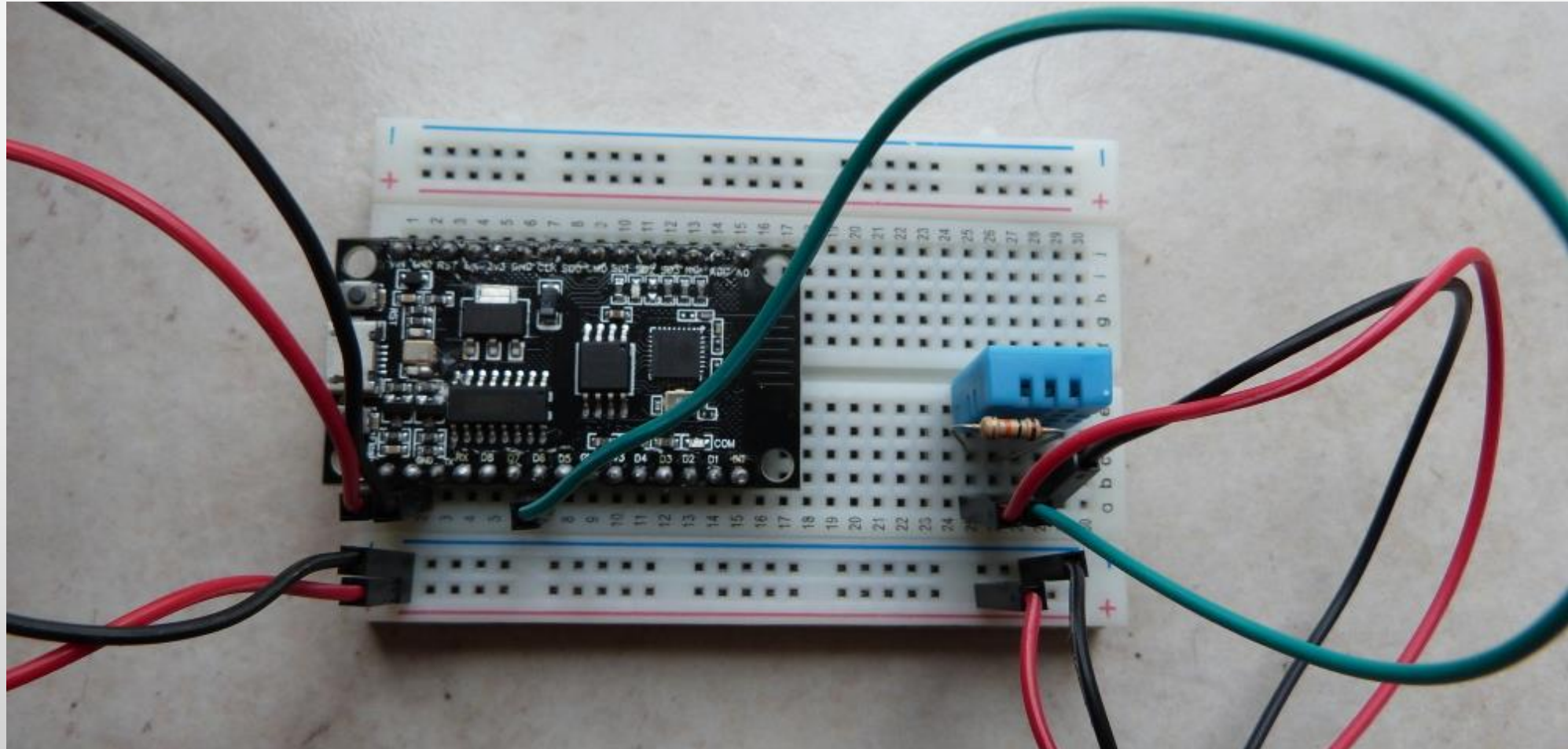


# Programação e Montagem dos Módulos

Nessa etapa do desenvolvimento foi utilizada a IDE PlatformIO integrada ao editor de texto Visual Studio Code, que permitiu o gerenciamento de bibliotecas auxiliares, a programação, a compilação e o envio do código ao NodeMCU. Para cada sensor foi desenvolvida uma classe responsável por lidar com a captura e o envio dos seus dados, como demonstrado na etapa de planejamento anteriormente.

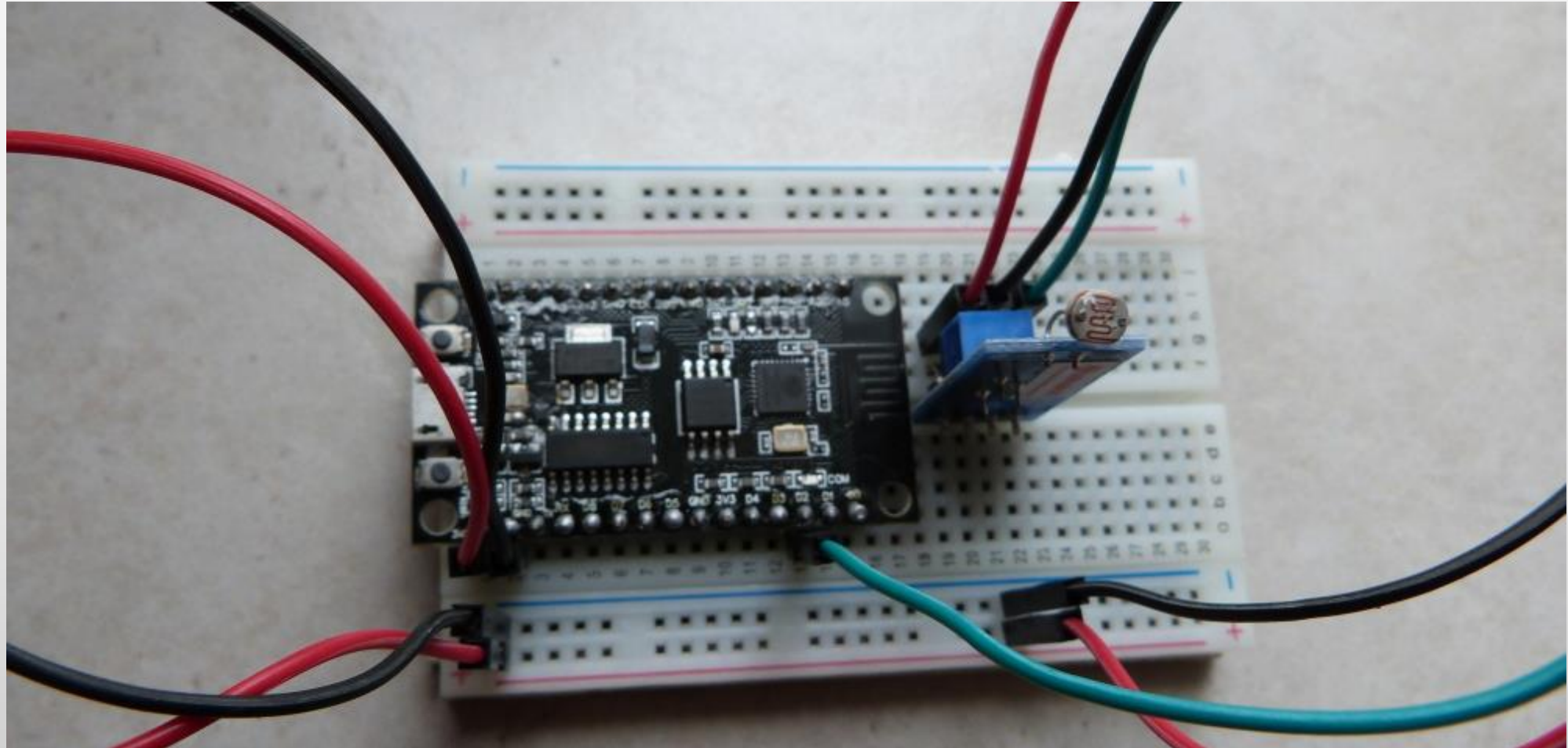
Para a montagem dos módulos foram utilizados um protoboard, a placa NodeMCU, o sensor específico para cada módulo, *jumpers* (fios de conexão) e para o módulo de temperatura e umidade também foi utilizado um resistor de  $10000\Omega$ . O resultado de cada módulo seguindo os diagramas do planejamento pode ser visto na Figura 19, Figura 20 e Figura 21.

Figura 19 – Montagem do módulo de temperatura e umidade



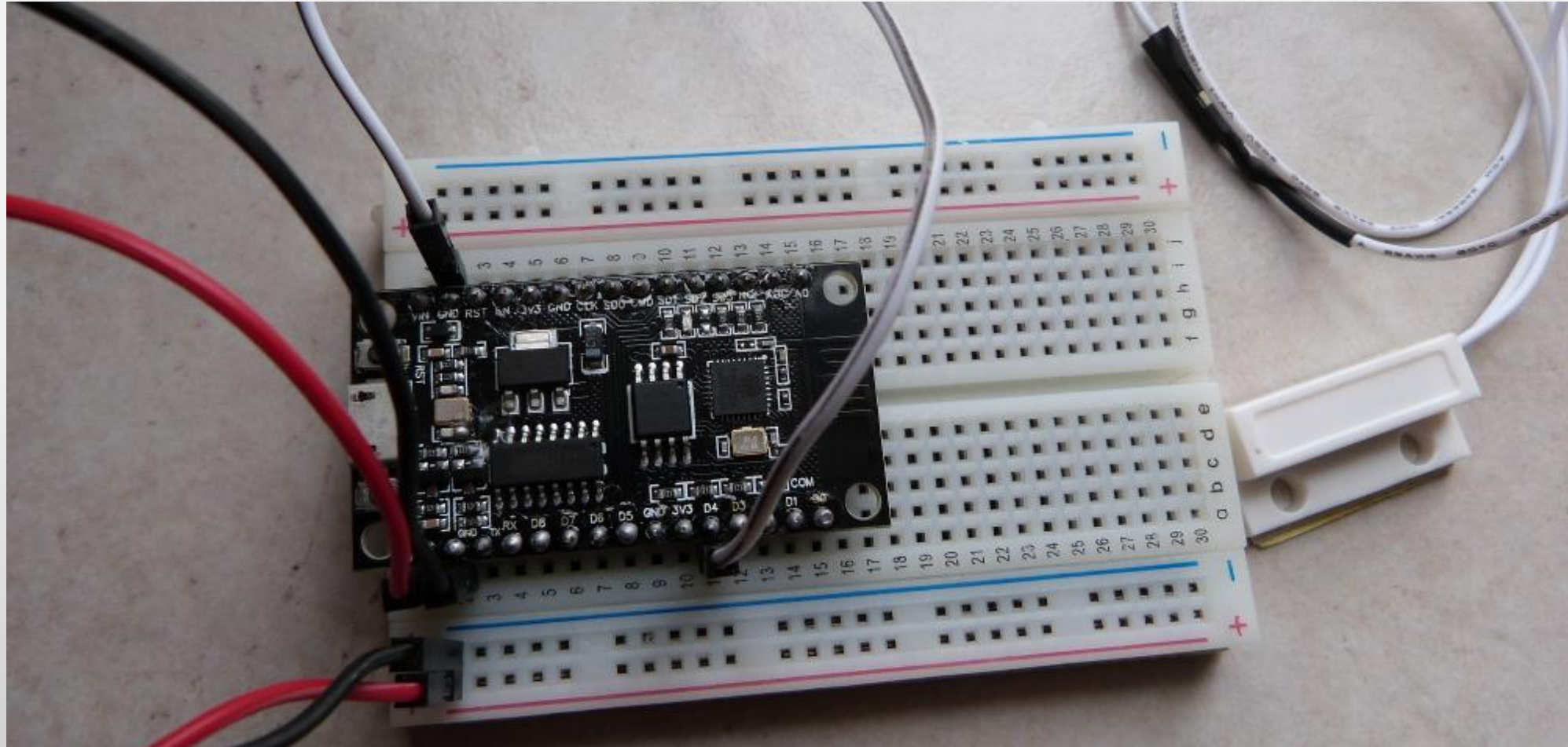
Fonte: elaborado pelo autor

Figura 20 – Montagem do módulo de luminosidade



Fonte: elaborado pelo autor

Figura 21 – Montagem do módulo de detecção do estado de porta



Fonte: elaborado pelo autor

# Captação dos Dados no Home Assistant

Após a etapa de configuração do Home Assistant no Raspberry Pi e da montagem e programação de cada módulo, pode-se efetuar de forma efetiva a captura dos dados. Para isso cada módulo envia seus dados em um tópico específico e a plataforma assina cada um dos tópicos, recebendo os dados e disponibilizando-os pela interface gráfica ou através da API. O resultado pode ser visto na Figura 22.

Figura 22 – Dados captados exibidos na interface do Home Assistant

## DHT



Temperature

30.66 °C



Humidity

71.30 %

## Light



LDR

Ligado

## Reed Switch



Reed Switch

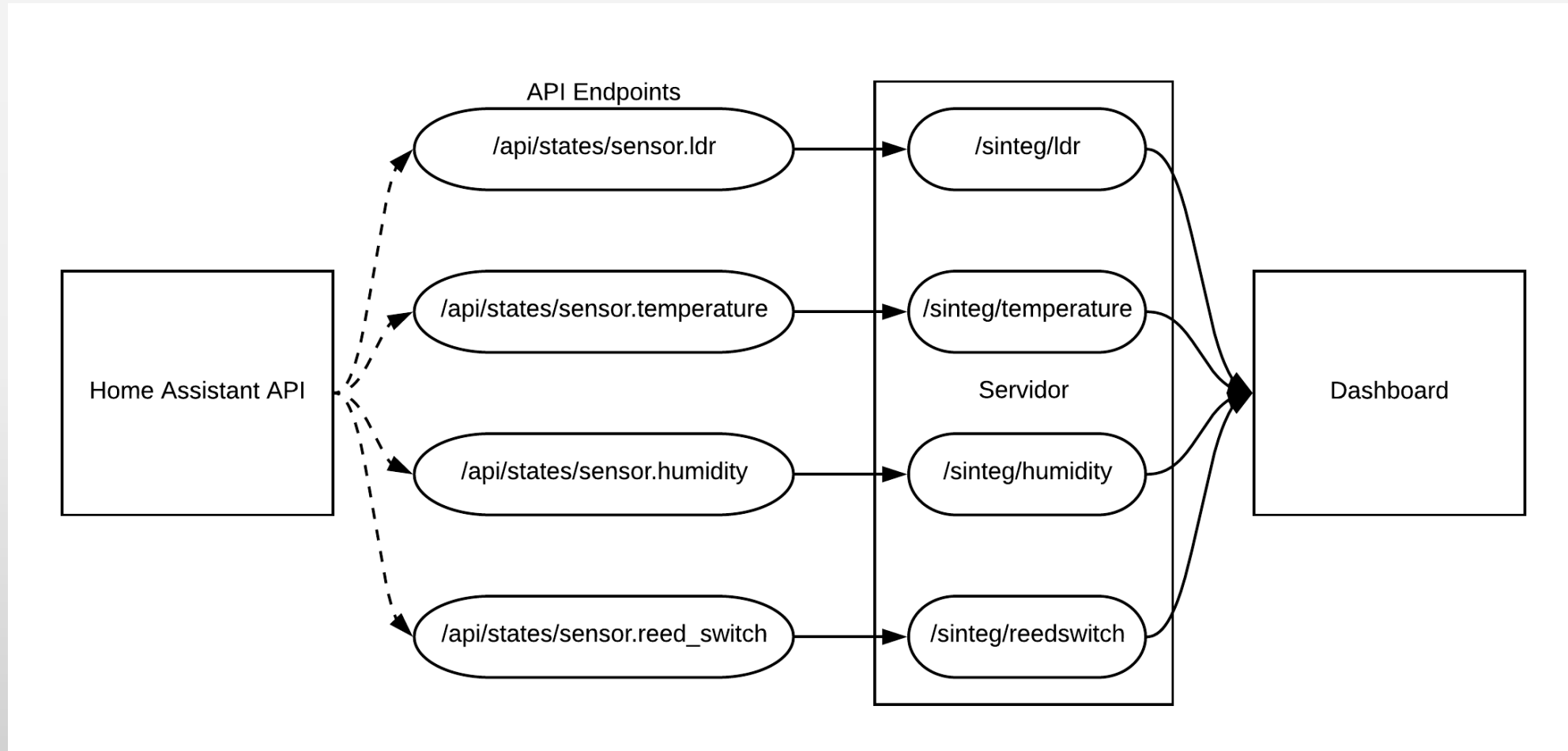
close

Fonte: elaborado pelo autor

# Integração com Dashboard para Controle de Projetos

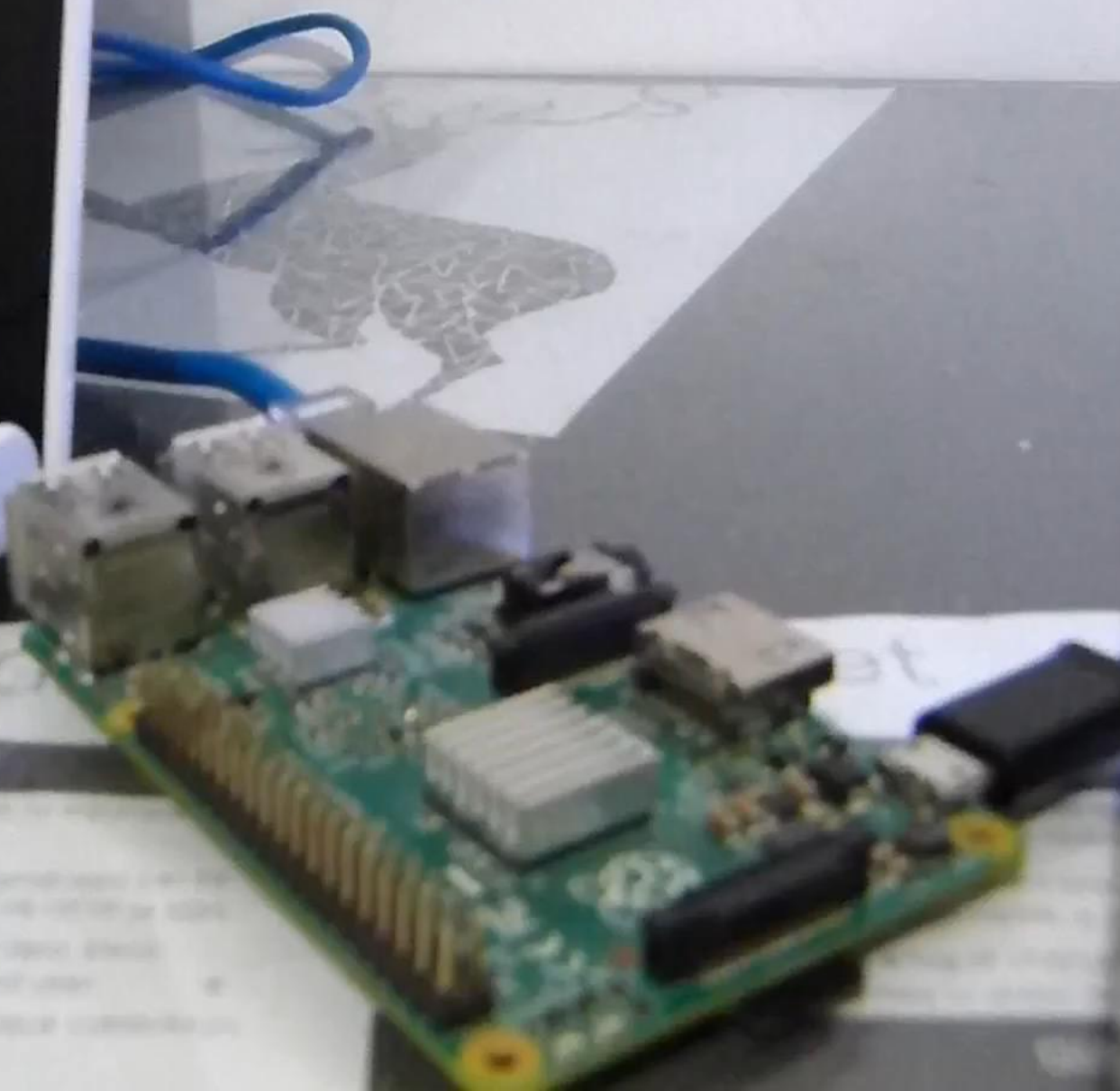
Para realizar a integração com um Dashboard para controle de projetos, foi elaborado um servidor utilizando Node.js para definir rotas simples para requisições HTTP que retornem uma resposta com o conteúdo do sensor solicitado. Com a utilização de um servidor intermediário novas rotas podem ser facilmente implementadas no servidor. Um diagrama do sistema como um todo é apresentado na Figura 23 e o resultado final pode ser visto no vídeo de funcionamento.

Figura 23 – Diagrama de integração com o Dashboard



Fonte: elaborado pelo autor





# Conclusão

---

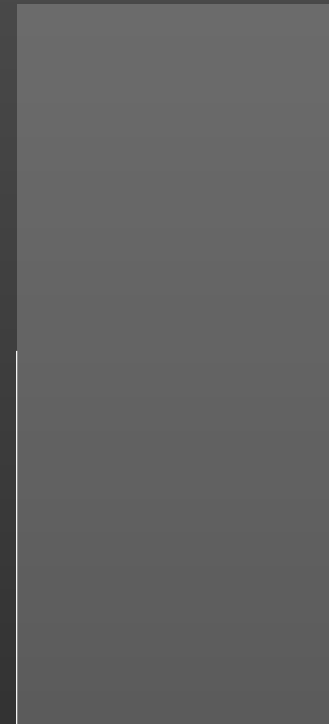
# Conclusão

---

- Objetivo de construção de três módulos IoT atingidos;
- Estudo e comparação de diversos modelos de sensores e placas de microcontroladores;
- Modularização de software e hardware permite implementações futuras sem complicações;
- Utilização de componentes de fácil acesso;
- Permite implementação fácil e barata.

# Trabalhos Futuros

---



# Trabalhos Futuros

---

- Automações integradas com outros componentes existentes no Home Assistant;
- Elaboração de um sistema de segurança para acesso as informações do servidor
- Implementação de uma interface para o NodeMCU que permita a conexão com outra rede sem fio sem ter que reprograma-lo;
- Elaboração de outros módulos utilizando o mesmo planejamento de software e hardware.

# Referências

- ☰ ASHTON, K. That 'Internet of Things' Thing. 2009. RFID Journal. Disponível em: <<https://www.rfidjournal.com/articles/view?4986>>. Acesso em: 10 out. 2018.
- ☰ GIBBS, K. Light dependent resistor (LDR) and thermistor. 2009. Schoolphysics. Disponível em: <[http://www.schoolphysics.co.uk/age14-16/glance/Electricity\%20and\%20magnetism/LDR\\\_and\\\_thermistor/index.html](http://www.schoolphysics.co.uk/age14-16/glance/Electricity\%20and\%20magnetism/LDR\_and\_thermistor/index.html)>. Acesso em: 12 out. 2018.
- ☰ LUETH, K. L. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. 2018. IoT Analytics. Disponível em: <<https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>>. Acesso em: 10 out. 2018.
- ☰ MURTA, G. Guia completo do NodeMCU – ESP12 – Introdução (1). 2018. Eletrogate Blog. Disponível em: <<http://blog.eletrogate.com/nodemcu-esp12-introducao-1/>>. Acesso em: 11 out. 2018.
- ☰ PICKERINGRELAY. Reed Relay Basics. 2018. Disponível em: <<https://www.pickeringrelay.com/introduction-reed-relay-basics-part-1/>>. Acesso em: 12 out. 2018.

# Referências

- ☰ ROELOFSEN, P. The impact of office environments on employee performance: The design of the workplace as a strategy for productivity enhancement. *Journal of Facilities Management, The Netherlands*, n. 3, p. 247–264, 2002.
- ☰ SATTI, J. A. et al. Miniaturized humidity and temperature sensing rfid enabled tags. *International Journal of Rf and Microwave Computer-Aided Engineering, Pakistan*, n. 1, p. n.p., 2018.
- ☰ STATISTA. Spending on Internet of Things worldwide by vertical in 2015 and 2020 (in billion U.S. dollars). 2018. Disponível em: <<https://www.statista.com/statistics/666864/iot-spending-by-vertical-worldwide/>>. Acesso em: 11 out. 2018.
- ☰ THIBODEAU, P. Um em cada cinco desenvolvedores já trabalha em projetos de IoT. 2015. *ComputerWorld*. Disponível em: <<https://computerworld.com.br/2015/05/18/um-em-cada-cinco-desenvolvedores-ja-trabalham-em-projetos-de-iot/>>. Acesso em: 11 out. 2018.
- ☰ THOMSEN, A. Como conectar o Arduino a uma rede Wi-Fi. 2014. *FilipeFlop*. Disponível em: <<https://www.filipeflop.com/blog/arduino-wifi-shield-cc3000/>>. Acesso em: 11 out. 2018.
- ☰ THOMSEN, A. Tutorial Módulo Wireless ESP8266 com Arduino. 2015. *FilipeFlop*. Disponível em: <<https://www.filipeflop.com/blog/esp8266-arduino-tutorial/>>. Acesso em: 11 out. 2018.

# Referências

---

- ☰ USINAINFO. Módulo Sensor de Luminosidade Fotossensitivo LDR. s.d. Disponível em: <<https://www.usinainfo.com.br/sensor-de-luminosidade-arduino/modulo-sensor-de-luminosidade-fotossensitivo-ldr-2539.html>>. Acesso em: 12 out. 2018.
- ☰ YUAN, M. Getting to know MQTT. 2017. IBM Developer. Disponível em: <<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 11 out. 2018.