

# INTERNET DAS COISAS (IOT) APLICADA AO MONITORAMENTO DO NÍVEL DE ÁGUA EM RESERVATÓRIOS DOMÉSTICOS

*INTERNET OF THINGS (IOT) APPLIED TO WATER LEVEL MONITORING IN DOMESTIC TANKS*

*INTERNET DE LAS COSAS (IOT) APLICADO AL MONITOREO DEL NIVEL DE AGUA DE TANQUES DOMÉSTICOS*

Jaime Ferreira do Nascimento<sup>1</sup>  
Ederson Cichaczewski<sup>2</sup>

## Resumo

Este artigo apresenta um projeto que aferiu o nível de água em um reservatório doméstico, por intermédio do microcontrolador ESP8266. Tal dispositivo possui um módulo Wi-Fi, que permite a conexão à internet, e recursos que reduzem o consumo de energia. Para monitorar o nível do reservatório, utilizou-se um sensor de distância ultrassônico HC-SR04, que possibilita, segundo o fabricante, leituras entre 2 cm e 4 metros, com precisão de até 3 mm. O protocolo MQTT foi escolhido para comunicação entre os dispositivos e o servidor; ademais, é uma alternativa superior ao HTTP, pois possui um cabeçalho que gera menos tráfego de dados na rede. A investigação incluiu, ainda, o desenvolvimento de um aplicativo para *smartphone* que opera como um *dashboard*, o que permite ao usuário efetuar as configurações do *hardware* e observar as informações coletadas do sensor; visualizam-se, assim, o nível de água do reservatório e um gráfico com o histórico de consumo. Os resultados obtidos indicaram ser possível efetuar o monitoramento por meio do sensor de distância.

**Palavras-chave:** IoT; MQTT; microcontrolador; monitoramento.

## Abstract

This article presents a project that measured the water level in a domestic tank, through the ESP8266 microcontroller. This device has a Wi-Fi module, which allows the microcontroller to connect to the internet, and resources that reduce energy consumption. To monitor the level of the tank, an ultrasonic distance sensor HC-SR04 was used, which, according to the manufacturer, allows readings between 2 cm and 4 meters, with an accuracy of up to 3mm. The MQTT protocol was chosen for communication between devices and the server; in addition, it is a superior alternative to HTTP, as it has a header that generates less data traffic on the network. The investigation also included a smartphone application development that operates as a dashboard, which allows the user to configure the hardware and observe the information collected from the sensor; thus, the water level of the tank and a graph with the consumption history are visualized. The results obtained indicated that it is possible to carry out the monitoring through the distance sensor.

**Keywords:** IoT; MQTT; microcontroller; monitoring.

## Resumen

Este artículo presenta un proyecto que mide el nivel de agua en un tanque doméstico, por medio del microcontrolador ESP8266. Tal dispositivo cuenta con un módulo wifi, que permite la conexión con internet y recursos que reducen el consumo de energía. Para el monitoreo del nivel del tanque, se utilizó un sensor de distancia ultrasónico HC-SR04, que permite, según el fabricante, lectura entre 2 cm a 4 m, con precisión de hasta 3 mm. Se optó por el protocolo MQTT para la comunicación entre los dispositivos y el servidor; además, es una alternativa superior al HTTP, pues cuenta con una cabecera que genera menos tráfico de datos en la red. La investigación incluye, también, el desarrollo de un aplicativo para *smartphone* que opera como un *dashboard*, que permite al usuario realizar las configuraciones del *hardware* y observar las informaciones recolectadas por el

---

<sup>1</sup> Graduado em Engenharia da Computação pela UNINTER. E-mail: jaime.desenvolvimento@hotmail.com.

<sup>2</sup> Orientador, Mestre em Engenharia Biomédica, Professor da UNINTER. E-mail: ederson.c@uninter.com.

sensor; se visualizan, de esa manera, el nivel de agua del tanque y un gráfico con el histórico de consumo. Los resultados obtenidos indican que es posible hacer el monitoreo por medio de un sensor de distancia.

**Palabras-clave:** IoT; MQTT; microcontrolador; monitoreo.

## 1 Introdução

A internet das coisas ou IoT (*Internet of Things*) é considerada uma das áreas mais promissoras e com maior potencial de transformação da vida moderna.

A comunicação de dados entre humanos para máquinas ou máquinas para máquinas é chamada de Internet das Coisas (IoT) e está ficando popular na maior parte do mercado industrial. A IoT é uma das dez tendências tecnológicas estratégicas de 2016 de acordo com o Gartner (HWANG, 2016).

Por meio do uso de IoT, pode-se obter maior controle sobre os equipamentos e ambientes. O monitoramento do nível de água em reservatório doméstico é um exemplo claro da utilidade dessa tecnologia, pois permitirá acompanhar o nível desta; gera-se, então, maior domínio e economia desse recurso fundamental, além de auxiliar no combate ao desperdício — um dos maiores problemas no uso da água potável nas residências. Isto posto, este trabalho visa investigar se é possível monitorar remotamente o nível de água em reservatório doméstico, por intermédio do microcontrolador ESP8266 WiFi e protocolo MQTT.

Um dos maiores problemas que a sociedade enfrenta no século 21 é a escassez de água potável. Segundo as Nações Unidas (2014), mais de 2 bilhões de pessoas podem ter seu acesso à água potável reduzido; a previsão é de que em 2050 uma em cada quatro pessoas, ao menos, viverá em um país afetado por escassez crônica ou recorrente desse recurso.

## 2 Fundamentação Teórica

*Internet of Things (IOT)* significa Internet das Coisas. Segundo Witkowski (2017), o conceito de IoT começou a se desenvolver em 1999, impulsionado pelas ideias do pesquisador britânico Kevin Ashton. Ashton vislumbrava a possibilidade de fazer com que todos os dispositivos do mundo material se comunicassem com computadores, utilizando a internet como canal de comunicação. Neste contexto, a IoT passou a englobar tudo que pode ser uma variável: objetos, equipamentos, pessoas, animais, fenômenos atmosféricos, entre outros.

Shankar (2018) afirma que enquanto na internet convencional os agentes emissor e receptor da comunicação são seres humanos, na IoT emissor e/ou receptor são as coisas; isto é, objetos que utilizam a internet como um canal de comunicação. Ademais, a Internet das Coisas

(IoT) pode ser pensada como um sistema de conexão de objetos ou dispositivos físicos, através de uma rede interconectada, como a internet.

Ainda, conforme Shankar (2018), as ‘coisas’ na IoT podem ser sensores embutidos, sistemas embarcados, entre outros, que contenham um endereço de IP atribuído, de modo a se comunicarem, coletarem e transferirem dados através da rede; assim, a tecnologia contida nestes dispositivos ajuda o sistema operar baseado nos dados, obtidos sem auxílio do usuário.

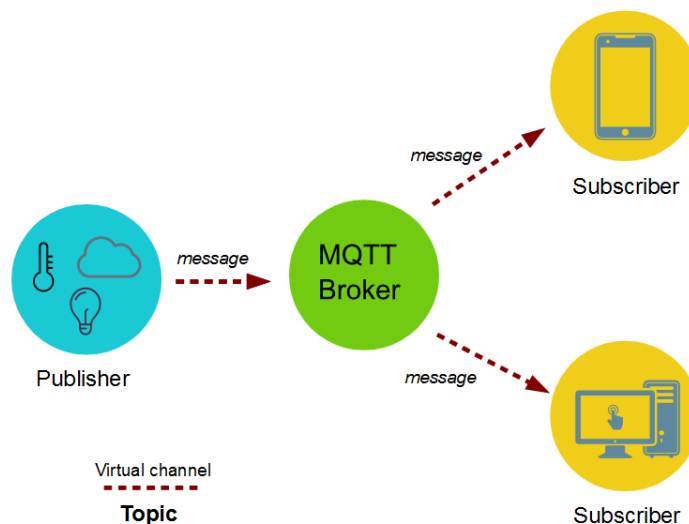
Perumal (2015) assevera que uma das rápidas influências da IoT está no campo do monitoramento ambiental, especialmente no gerenciamento de desastres, nos sistemas de alerta antecipado e na análise de dados ambientais. Destarte, uma das questões importantes em um monitoramento ambiental é o monitoramento da água e a sustentabilidade.

## 2.1 *Message Queuing Telemetry Transport* (MQTT)

MQTT é um protocolo desenvolvido pela IBM e Eurotech. Este protocolo baseia-se no princípio de publicar mensagens e assinar tópicos, o que é tipicamente referido como um modelo de publicação/assinatura. O MQTT assegura a entrega de mensagens, mantendo-as o mais leve possível. O usuário pode se inscrever em tópicos que pertencem a ele e, assim, receber quaisquer mensagens publicadas neles.

A Figura 1 exibe sua arquitetura.

**Figura 1:** Arquitetura do protocolo MQTT

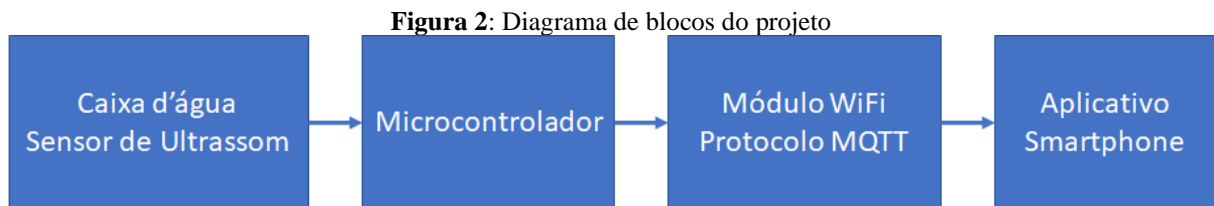


Fonte: SWA (2016).

### 3 Metodologia

Quanto à metodologia do presente estudo, utilizou-se a pesquisa empírica, e esta seção aborda os materiais e métodos utilizados para o projeto e implementação do protótipo proposto.

A Figura 2 apresenta um diagrama de blocos com as partes principais do projeto, que serão descritas nesta seção.



**Fonte:** o autor (2021).

#### 3.1 Sensor Ultrassônico HC-SR04

A Figura 3 exibe a pinagem do sensor de distância HC-SR04, utilizado para aferir a distância entre 2cm até 4 metros, com precisão de 3mm — de acordo com o *datasheet* (folha de dados) deste componente.

O funcionamento do sensor baseia-se no envio de pulsos de som em alta frequência, que ao colidir com algum obstáculo, retorna até o sensor; logo, a distância pode ser calculada com base na diferença do tempo de envio e retorno do sinal sonoro.

**Figura 3:** Sensor Ultrassônico HC-SR04



**Fonte:** <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.

A distância entre o sensor e um objeto pode ser calculada por meio da Equação 1:

$$Distancia[m] = \frac{T[s] * 340 \frac{[m]}{[s]}}{2} \quad \text{(Equação 1)}$$

Onde: T[s] é o tempo em que o pino *Echo* do sensor fica em nível lógico alto, que corresponde à distância medida pelo sensor.

Devido à impossibilidade de contato com a água, escolheu-se o sensor ultrassônico de distância para medir o nível de água, pois seu princípio de funcionamento permite que ele seja aferido ao disparar um ultrassom que, ao colidir com o espelho de água, retorna até o sensor — possibilitando o cálculo do nível de água do de.

### 3.2 Reservatório

O reservatório monitorado tem capacidade de armazenar até 10.000 litros de água. A Figura 4 exibe o formato e as dimensões do reservatório, e o volume é calculado por meio da Equação 2.

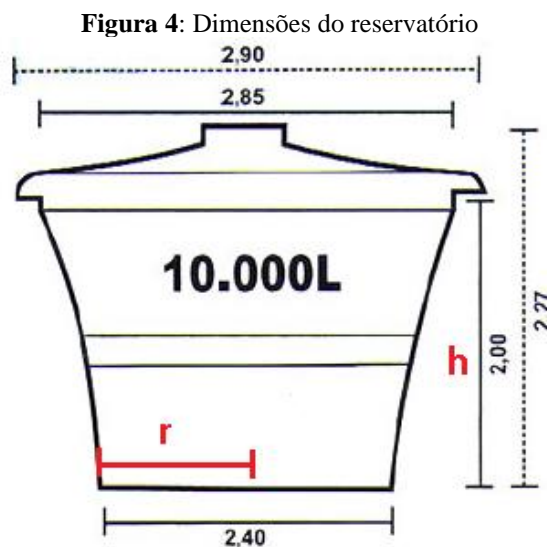
$$V = \left[ \pi * \int_0^h (0.1125 * r + 1.2)^2 dr \right] * 1000 \quad (\text{Equação 2})$$

Sendo:

V = volume em litros.

r = raio em metros.

h = altura entre o sensor e o espelho de água em metros.



Fonte: o autor (2021).

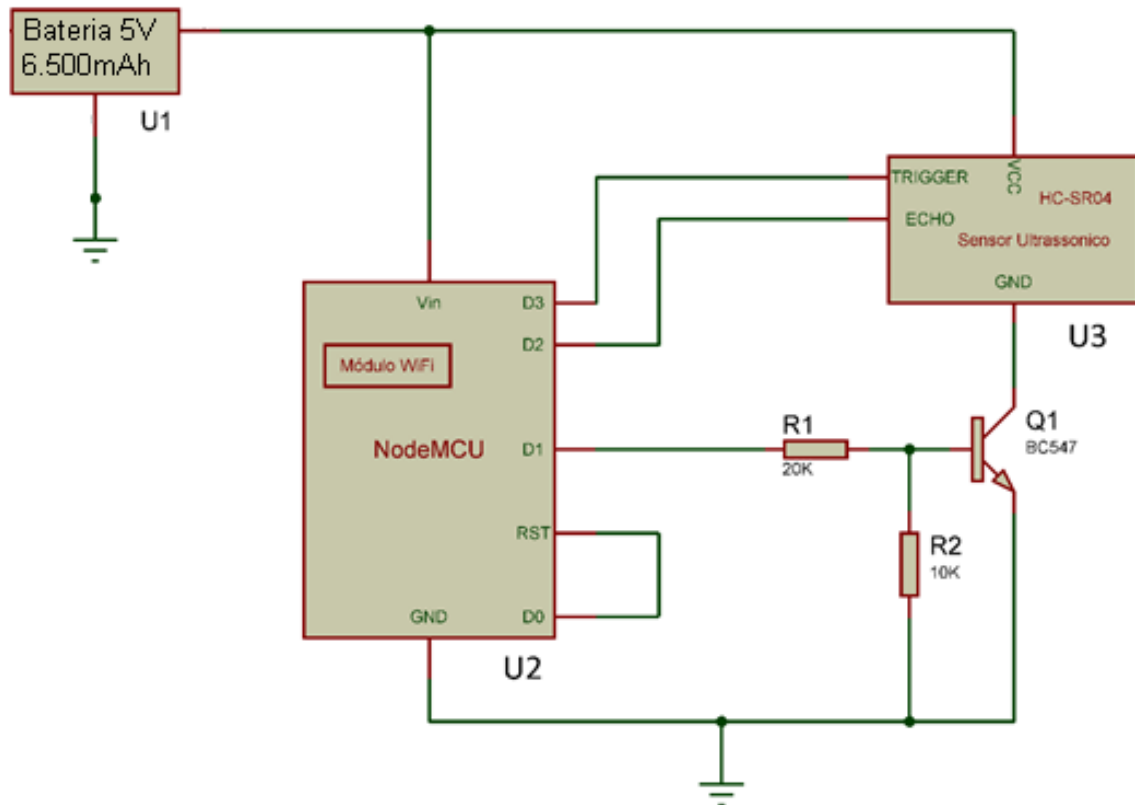
### 3.3 Microcontrolador

O diagrama esquemático do circuito do microcontrolador é apresentado na Figura 5. O módulo NodeMCU compreende o microcontrolador ESP8266 com módulo Wi-Fi e antena integrada. O transistor Q1 é utilizado como chave para desativar o sensor ultrassônico durante

o período que o sistema está em modo de espera, reduzindo o consumo de corrente da bateria; ademais, uma bateria alimenta o NodeMCU e o sensor de distância com 5V.

O NodeMCU tem regulador interno para 3,3V e possui portas de I/O tolerantes a 5V; já o sensor de distância HC-SR04 trabalha diretamente com 5V de tensão.

**Figura 5:** Diagrama esquemático do hardware



**Fonte:** o autor (2021).

O ESP8266 conta com 3 tipos de *Sleep Mode* que permite reduzir o consumo de acordo com a necessidade: *Modem-Sleep*, *Light-Sleep* e *Deep-Sleep*.

Neste trabalho, utilizou-se o modo *Deep-Sleep*, para reduzir o consumo de energia ao máximo.

### 3.4 Fonte de alimentação

O sistema foi alimentado por uma bateria portátil da marca ELG modelo PB65; este *power bank* tem capacidade para 6.500 mAh de corrente com saída de 5v de tensão. A Figura 6 exibe a bateria ELG utilizada.

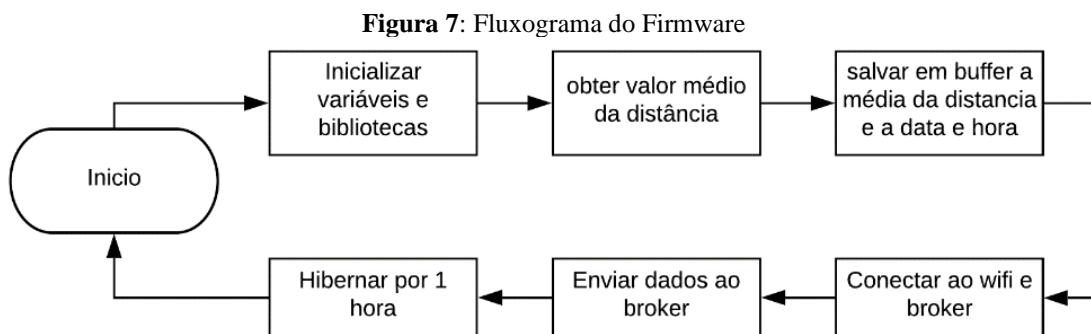
**Figura 6** - Power bank utilizado como fonte de alimentação, modelo PB65



**Fonte:** <https://www.elg.com.br/PT/produtos/acessorios-para-smartphones/power-banks/power-bank-mosaic--cabo-micro-usb--6-500mah--pb65>.

### 3.5 Firmware

A Figura 7 apresenta um fluxograma simplificado do *firmware* desenvolvido. Durante a inicialização, o programa carrega as bibliotecas e inicializa as variáveis; em seguida, efetuam-se 10 leituras do sensor para obter uma média da distância, e este valor é armazenado em um *buffer* com a data e hora. Após armazenar o valor, o sistema tenta se conectar ao *Wi-Fi* e também ao *Broker*. Caso a conexão ocorra sem problemas, os dados do *buffer* são enviados em formato JSON; então, o sistema retorna ao seu estado de hibernação, em que aguarda 1 hora e todo o processo é novamente reiniciado.

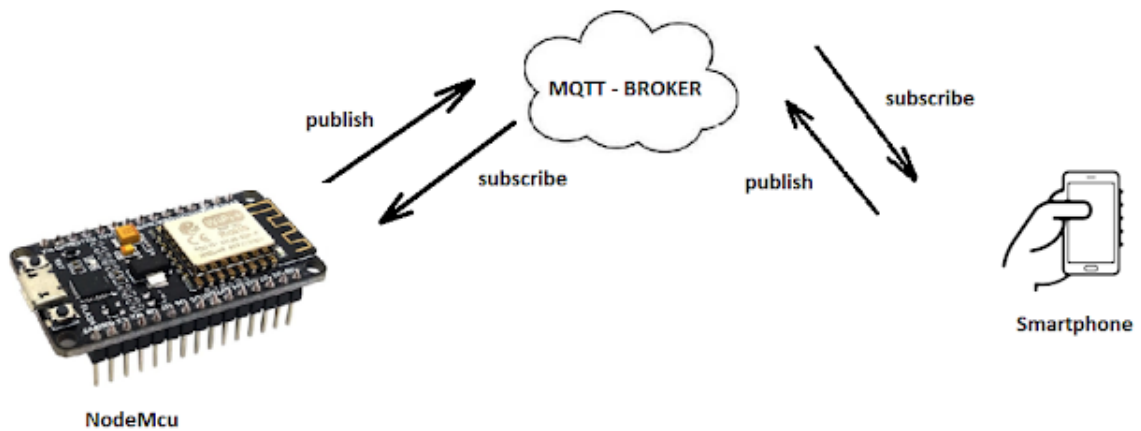


**Fonte:** o autor (2021).

### 3.6 Broker MQTT

O servidor/*Broker* escolhido foi a plataforma *Mosquitto*, um serviço *open source*; para o lado cliente, utilizou-se a biblioteca Paho-MQTT que também é *open source*, mantida pela Eclipse Foundation. Por meio da biblioteca Paho-MQTT, é possível inscrever em tópicos para receber dados e enviar publicações ao servidor. A Figura 8 exibe o fluxo dos dados entre o microcontrolador que está atuando como *Publisher*, o *Broker* e o aplicativo no *smartphone* que atua como *Subscriber*.

**Figura 8:** Comunicação entre *Publisher*, *Broker* e *Subscriber*



**Fonte:** o autor (2021).

### 3.7 Aplicativo Android

O aplicativo foi desenvolvido utilizando o *framework* Ionic e Angular; as linguagens de programação utilizadas foram TypeScript, HTML, CSS e JavaScript, e a IDE utilizada foi o Visual Studio Code. Implementou-se o banco de dados com o SQLite — banco de dados interno e oficial da plataforma Android.

O aplicativo é o responsável por armazenar e exibir os dados coletados pelo sensor enviados ao *Broker*. A Tabela 1 exibe os dados persistidos para compor as informações do gráfico e a Tabela 2 exibe os parâmetros relacionados ao reservatório.

**Tabela 1:** Persistência dos dados

Tabela_Dados	
int	id
float	distancia
float	nivel
float	volume
DateTime	data

**Fonte:** o autor (2021).

**Tabela 2:** Persistência dos parâmetros

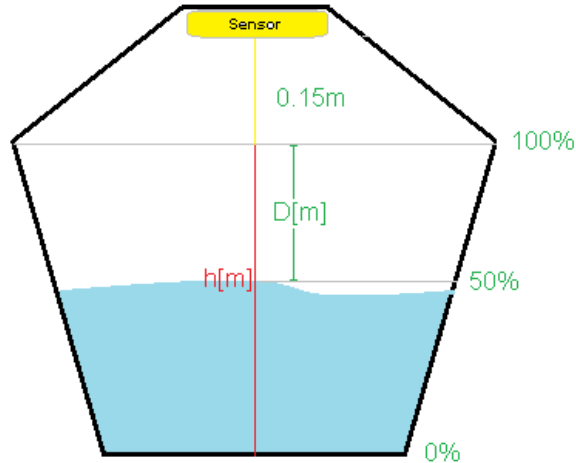
Tabela_Reservatorio	
int	id
charvar	nome
float	altura_max_agua
float	altura_sensor
float	diametro_inferior
float	diametro_superior

**Fonte:** o autor (2021).



A Figura 9 ilustra o posicionamento do sensor no reservatório e as medidas utilizadas para calcular volume e nível. O sensor deve estar afastado 15 cm do espelho de água para evitar danos.

**Figura 9:** Disposição do sensor no reservatório e medidas utilizadas nos cálculos



**Fonte:** o autor (2021).

O sensor mede a distância entre ele e o espelho de água e publica — via protocolo MQTT — este dado ao *Broker*. O app subscrito neste mesmo tópico recebe os dados do *Broker* e, de posse deste valor, o aplicativo converte a distância para nível de água, variando de [100% – 0%] por meio da Equação 3.

$$Nivel[\%] = 100 - \left( \frac{100 * (D[m])}{h[m]} \right) \quad (\text{Equação 3})$$

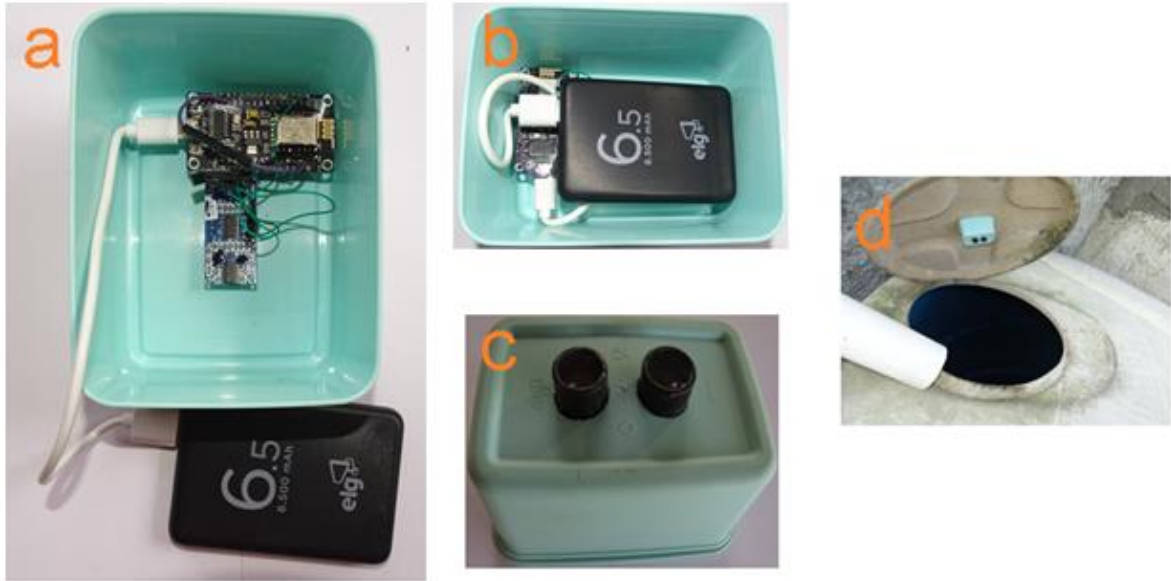
O volume também é calculado por meio da variação da distância do espelho de água ( $D[m]$ ) pela Equação 4.

$$Volume[L] = \left[ \int_0^{h-D} \pi * (0.1125 * r + 1.2)^2 dr \right] * 1000 \quad (\text{Equação 4})$$

#### 4 Resultados e Discussão

A Figura 10(a) exibe o *hardware* com o sensor de distância HC-SR04, em que o NodeMCU é alimentado por uma bateria 5V/ 6500mA. Na Figura 10(b), apresenta-se a acomodação do *hardware* na caixa responsável por protegê-lo da umidade. A Figura 10(c) exibe a parte do sensor que fica exposta e, por ser metálica, foi necessário revesti-la com fita isolante líquida para proteger da umidade; já a Figura 10(d) exibe o dispositivo instalado na tampa do reservatório.

**Figura 10:** a) Hardware do protótipo construído. b) Caixa de proteção. c) Acomodação do hardware na caixa. d) Dispositivo instalado no reservatório

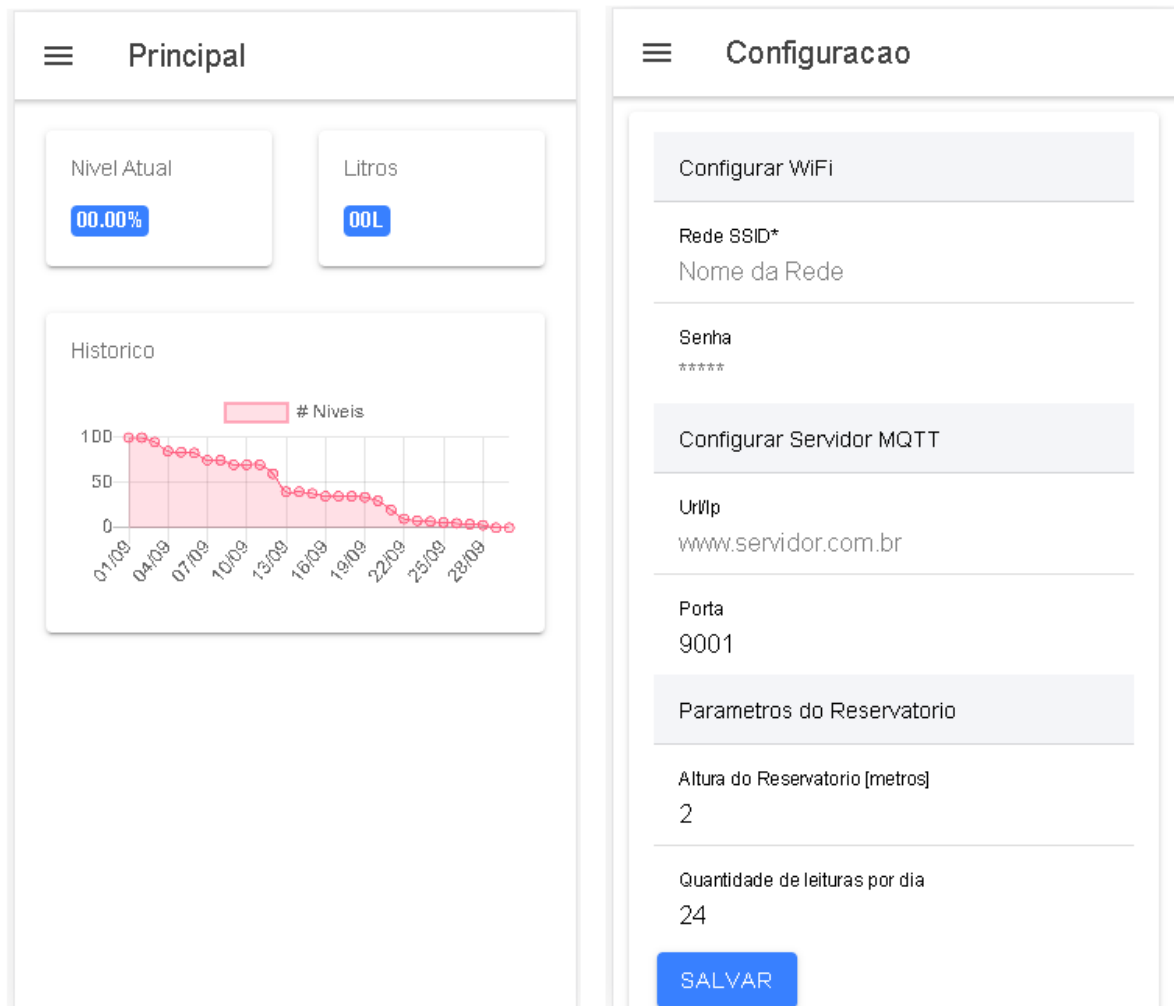


**Fonte:** o autor (2021).

O protótipo apresentou um consumo de corrente de 80mA com todos os periféricos ativados.

Para a exibição dos dados previamente coletados ao usuário, desenvolveu-se um aplicativo para dispositivo móvel, que possui duas telas. A Figura 11 exibe a tela principal e a tela de configuração; a tela principal é responsável por exibir ao usuário o gráfico do consumo de água, além do nível atual e o volume em litros, enquanto que a tela de configuração contém campos para configuração das características do reservatório, conexão ao Wi-Fi e também servidor MQTT.

**Figura 11:** Telas do aplicativo para Android



**Fonte:** o autor (2021).

A Tabela 3 apresenta os dados coletados do reservatório durante o período de 30 dias, ao longo de setembro de 2019. Foram coletadas uma amostra por dia, com intervalos de 24 horas. Pode-se observar, por meio do gráfico apresentado na Figura 12, que o consumo se mantém quase constante durante o mês todo; assim, evidencia-se que há uma rotina no consumo doméstico.

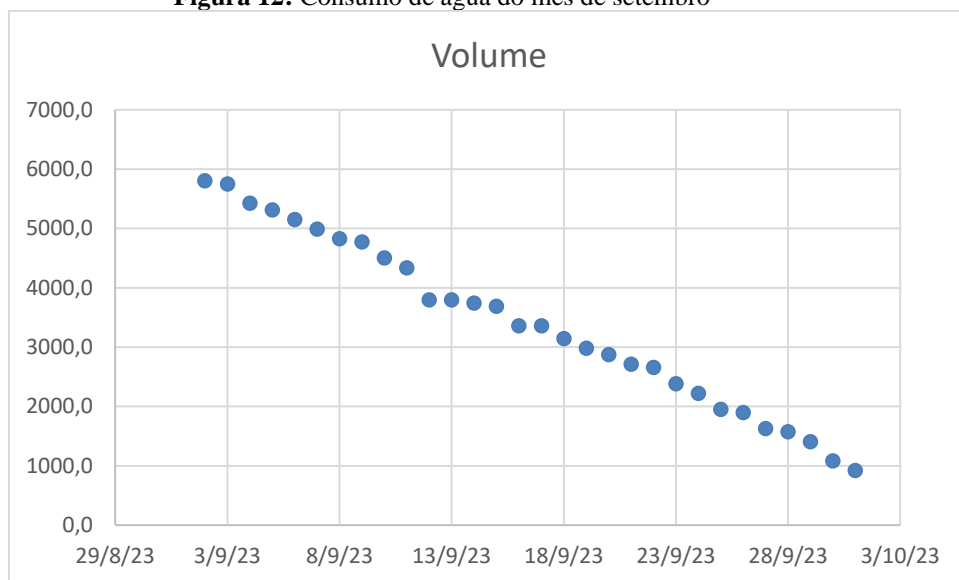
**Tabela 3:** Tabela de dados referente ao mês de setembro de 2019.

Amostra	Data	Altura h [cm]	Nível [%]	Volume [L]
1	1/9/19	107	53.5	5804.9
2	2/9/19	103	53	5750.6
3	3/9/19	100	50	5425.1
4	4/9/19	98	49	5316.6
5	5/9/19	95	47.5	5153.9
6	6/9/19	92	46	4991.1
7	7/9/19	89	44.5	4828.4

8	8/9/19	88	44	4774.1
9	9/9/19	83	41.5	4502.9
10	10/9/19	80	40	4340.1
11	11/9/19	70	35	3797.6
12	12/9/19	70	35	3797.6
13	13/9/19	69	34.5	3743.3
14	14/9/19	68	34	3689.1
15	15/9/19	62	31	3363.6
16	16/9/19	62	31	3363.6
17	17/9/19	58	29	3146.6
18	18/9/19	55	27.5	2983.8
19	19/9/19	53	26.5	2875.3
20	20/9/19	50	25	2712.6
21	21/9/19	49	24.5	2658.3
22	22/9/19	44	22	2387.1
23	23/9/19	41	20.5	2224.3
24	24/9/19	36	18	1953.0
25	25/9/19	35	17.5	1898.8
26	26/9/19	30	15	1627.5
27	27/9/19	29	14.5	1573.3
28	28/9/19	26	13	1410.5
29	29/9/19	20	10	1085.0
30	30/9/19	17	8.5	922.3

Fonte: o autor (2021).

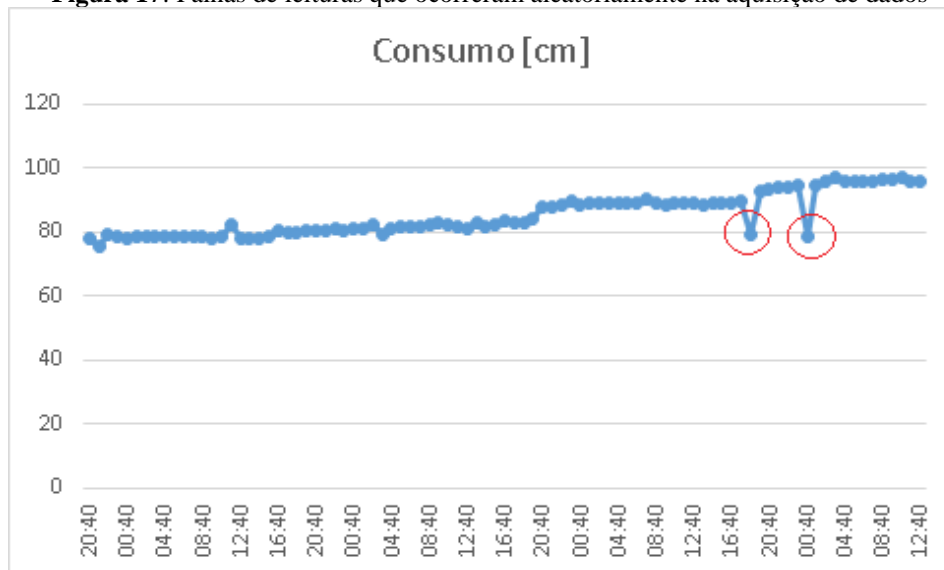
Figura 12: Consumo de água do mês de setembro



Fonte: o autor (2021).

Durante o processo de aquisição de dados, alguns valores obtidos pelo sensor apresentaram erros de leitura. Destarte, a Figura 13 exibe 2 picos de leituras incorretas obtidas pelo sensor.

**Figura 17:** Falhas de leituras que ocorreram aleatoriamente na aquisição de dados



**Fonte:** o autor (2021).

Para aumentar o tempo de autonomia da bateria, o sistema passa a maior parte do tempo hibernando no modo *Deep-Sleep*, mantendo consumo de 0,18mA, por aproximadamente 59 minutos; então, há um pico de 80mA com duração de até 1 minuto em média, que representa o período de quando o módulo está em ativo. O tempo aproximado de descarga da bateria em horas pode ser calculada por meio da Equação 5.

$$t_D[h] = \left( \frac{(mA.h)/I_{carga}[mA]}{t_A} \right) * f_e \quad \text{(Equação 5)}$$

Onde:

- $t_D$  é o Tempo de Descarga da bateria em horas;
- mA.h é a capacidade da bateria em fornecer corrente, neste caso 6.500mA.h;
- $I_{carga}$  é o consumo de corrente da carga;
- $t_A$  é o percentual de tempo que o módulo permanece ativo;
- $f_e$  é um fator de eficiência que pode ser considerado de 90% ou 0.9.

O tempo teórico de autonomia da bateria pode ser calculado substituindo na Equação 5:

$$t_D[h] = \left( \frac{6.500 \text{ mAh}/80 \text{ mA}}{\frac{1}{60}} \right) * 0.9$$

Desta forma, verifica-se que a média de tempo de descarga da bateria será de 4387 horas de autonomia ou aproximadamente 180 dias de bateria.

O fator de eficiência é necessário porque a bateria não entrega 100% de sua potência para a carga, uma parte da energia é desperdiçada na forma de calor, por exemplo.

## 5 Considerações finais

Os resultados indicaram que o sistema é eficiente e tecnicamente viável para monitoramento de reservatórios domésticos.

O protocolo MQTT é simples, leve e de fácil uso, porém, por característica do protocolo, ele não guarda os dados enviados pelos dispositivos, sendo possível apenas reter o último valor enviado por um dispositivo quando  $QoS = 2$ . Esta característica dificulta até certo ponto a aquisição de dados em tempo real que precisam ser persistidos, sendo necessário manter um sistema conectado ao servidor em tempo integral, aguardando a chegada destes dados.

A bateria escolhida para alimentar o sistema se apresentou suficiente para manter a autonomia do dispositivo por um período aceitável de aproximadamente 180 dias; no entanto, o fato de manter o sistema em modo de hibernação impede que o usuário tenha acesso ao dispositivo em tempo real, o que limita a coleta dos dados que já foram disponibilizados até aquele momento.

Eventualmente, o sensor apresenta leituras incorretas, mas não chega a comprometer o desempenho do sistema; contudo, este é um ponto do projeto que demanda atenção e melhoria.

Para trabalhos futuros, sugere-se pesquisar formas alternativas de sensores de nível que permitam maior precisão nas leituras e a viabilidade econômica do produto.

## Referências

HWANG, H.C., Park, J. & Shon, J.G. Design and Implementation of a Reliable Message Transmission System Based on MQTT Protocol in IoT. **Wireless Pers Commun**, [S.l.], v. 91, p. 75-177, jun. 2016. DOI: <https://doi.org/10.1007/s11277-016-3398-2>.

INTERNACIONAL BUSINESS MACHINES CORPORATION; EUROTECH. **MQTT specification version 3.1**. 2010. Disponível em: [public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html](http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html). Acesso em: 17 maio 2019.

MQTT Protocol Tutorial. **SWA**, [S.l.], 2016. Disponível em: <https://www.survivingwithandroid.com/mqtt-protocol-tutorial/>. Acesso em: 26 set. 2019.

PERUMAL, T.; SULAIMAN, M. N.; LEONG, C. Y. Internet of Things (IoT) enabled water monitoring system. In: IEEE GLOBAL CONFERENCE ON CONSUMER ELECTRONICS (GCCE), 4., 2015, Osaka. **Proceedings** [...]. Osaka: IEEE, 2015. p. 86-87.

SACOMANO, J. *et al.* **Indústria 4.0**. 1. ed. São Paulo: Blucher, 2018.

SHANKAR, S.; DAKSHAYINI, M. IoT-Mobile Enabled Smart Water Level Controlling System to Regulate Water Wastage. *In*: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS (ICACCI), Bangalore, 2018. **Proceedings** [...]. Bangalore: IEEE, 2018. Disponível em: <https://www.semanticscholar.org/paper/IoT-Mobile-Enabled-Smart-Water-Level-Controlling-to-Shankar-Dakshayini/d2fb2e0912bea16e909d14fe506ad7007aaf01bb>. Acesso em: 17 set. 2021.

TOP 10 technology trends signal the digital mesh. **Gartner**, [S.l.], 2015. Disponível em: <http://www.gartner.com/smarterwithgartner/top-ten-technology-trends-signal-the-digital-mesh/>. Acesso em: 29 set. 2019.

UNITED NATIONS. Ensure access to water and sanitation for all. **United Nations**, [S.l.], 2014. Disponível em: <https://www.un.org/sustainabledevelopment/water-and-sanitation/>. Acesso em: 10 maio 2019.

WITKOWSKI. **Internet of Things, Big Data, Industry 4.0: Innovative Solutions in Logistics and Supply Chains Management**. Poland: Procedia Engineering, 2017