

---

Desenvolvimento de um sistema embarcado de monitoramento de  
nível de água aplicado ao reservatório da Pequena Central  
Hidrelétrica do Lobo em Itirapina - SP

*Gustavo Benetti Delai*

---



Desenvolvimento de um sistema embarcado de monitoramento de  
nível de água aplicado ao reservatório da Pequena Central  
Hidrelétrica do Lobo em Itirapina - SP

**Gustavo Benetti Delai**

*Orientador:* **Prof. Dr. Frederico Fábio Mauad**

Monografia apresentada à Escola de Engenharia de São  
Carlos, como parte dos requisitos para obtenção do título  
de Engenheiro Eletricista.

**USP – São Carlos**  
**Dezembro de 2018**

D334d Delai, Gustavo Benetti  
Desenvolvimento de um sistema embarcado de monitoramento de nível de água aplicado ao reservatório da Pequena Central Hidrelétrica do Lobo em Itirapina - SP / Gustavo Benetti Delai; orientador Frederico Fábio Mauad. - São Carlos - SP, 2018.  
74 p.

Monografia (Engenheiro - Eletricista) - Escola de Engenharia de São Carlos, Universidade de São Paulo, 2018.

1. Internet das Coisas. 2. IoT. 3. Raspberry Pi. 4. Pequenas Centrais Hidrelétricas. 5. MQTT. 6. sensoriamento. I. Mauad, Frederico Fábio, orient. II. Título.

# FOLHA DE APROVAÇÃO

Nome: Gustavo Benetti Delai

Título: "Desenvolvimento de um sistema embarcado de monitoramento de nível de água aplicado ao reservatório da Pequena Central Hidrelétrica do Lobo em Itirapina - SP"

Trabalho de Conclusão de Curso defendido e aprovado  
em 27/11/18,

com NOTA 10,0 (dez), pela Comissão Julgadora:

*Prof. Associado Frederico Fabio Mauad - Orientador - SHS/EESC/USP*

*Prof. Associado José Carlos de Melo Vieira Júnior - SEL/EESC/USP*

*Mestre Paulo Roberto Bairros da Silva - Doutorando - EESC/USP*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Rogério Andrade Flauzino



*Gostaria de agradecer sobretudo à população do estado de São Paulo, que permite, através do pagamento de impostos, pagos com um dinheiro conquistado com suor, que eu tenha realizado o curso de Engenharia Elétrica em uma prestigiada universidade.*

*Também agradeço à minha mãe, pai e irmão.*

*Por final, agradeço com muita alegria todos os que, de alguma forma, estiveram presentes comigo em todos estes anos de graduação. Ao meu orientador, Frederico, e ao Carlos, sem os quais este trabalho não poderia ter sido feito, ao menos da forma como foi.*

*A todos esses, muito obrigado.*



# RESUMO

DELAI, G.B.. **Desenvolvimento de um sistema embarcado de monitoramento de nível de água aplicado ao reservatório da Pequena Central Hidrelétrica do Lobo em Itirapina - SP.** 2018. 74 f. Monografia (Engenheiro Eletricista) – Escola de Engenharia de São Carlos (EESC/USP), São Carlos – SP.

Este trabalho teve por objetivo desenvolver uma solução integrada de monitoramento de nível de água e aplicá-la a uma pequena central hidrelétrica. Para tal, utilizou-se de um microcontrolador Raspberry Pi B, um sensor de distância e a plataforma de internet das coisas IBM Bluemix. O sensor lê os dados, que são enviados à plataforma na internet das coisas por meio de protocolo MQTT. Ali são processados e armazenados em um banco de dados, além de exibidos em um *Dashboard*, para visualização do usuário. O trabalho foi testado sob duas condições (testes controlados e uma aplicação em uma pequena central hidrelétrica (Represa do Lobo) ) para garantir a eficácia do sensor frente às condições estabelecidas pela Agência Nacional de Águas (ANA). Nos testes realizados, mostrou-se que o erro médio estava dentro do valor esperado pela ANA, sendo assim, o sensor pode ser utilizado nas aplicações para as quais foi desenvolvido. Devido ao baixo custo de implementação das tecnologias utilizadas, permite a expansão para outras áreas de utilização, abrindo caminho para trabalhos futuros.

**Palavras-chave:** Internet das Coisas, IoT, Raspberry Pi, Pequenas Centrais Hidrelétricas, MQTT, sensoriamento.



# ABSTRACT

DELAI, G.B.. **Desenvolvimento de um sistema embarcado de monitoramento de nível de água aplicado ao reservatório da Pequena Central Hidrelétrica do Lobo em Itirapina - SP.** 2018. 74 f. Monografia (Engenheiro Eletricista) – Escola de Engenharia de São Carlos (EESC/USP), São Carlos – SP.

This paper had as objective the development of an integrated water level monitoring solution and its application on a small hydro. For that, a Raspberry Pi B microcontroller, a distance sensor and the IBM Bluemix Internet of Things platform were used. The sensor reads the distance data, which are then sent to the IoT platform through MQTT protocol, where they are processed and stored in a database. The data is also available for the user through a Dashboard. This work was tested in two different conditions (controlled tests and an application in a small hydro (Represa do lobo) ) to show the accuracy of the sensor is within the limits specified by the regulations agency (ANA - Agência Nacional de Águas). In the realized tests, the mean error was within the specified. Thus improving the sensor may be used for the application to which it was designed. Since the used technologies have a low implementation cost, its expansion to another areas is achievable, allowing further work.

**Key-words:** Internet of Things, IoT, Raspberry Pi, Small hydro, MQTT, sensing.



# LISTA DE ILUSTRAÇÕES

---

---

Figura 1 – Vista da Represa do Lobo. . . . .	25
Figura 2 – Placa Raspberry Pi 3. . . . .	33
Figura 3 – Ilustrativo indicando o funcionamento do sensor. . . . .	35
Figura 4 – Esquema indicando os níveis e interfaces entre os dispositivos, a nuvem e o usuário. . . . .	38
Figura 5 – Ilustrativo indicando a conexão do sensor ao Raspberry PI. . . . .	39
Figura 6 – Conexão realizada entre o sensor e o Raspberry PI. . . . .	39
Figura 7 – Fluxo construído na plataforma Node-RED. . . . .	43
Figura 8 – Tela de exibição de dados para o usuário ( <i>Dashboard</i> ). . . . .	44
Figura 9 – Testes feitos na barragem na posição 1 . . . . .	46
Figura 10 – Testes feitos na barragem na posição 2 . . . . .	47
Figura 11 – Tela de inserção de dados para conexão ao Broker MQTT. . . . .	52



# LISTA DE TABELAS

---

---

Tabela 1 – Classificação das PCHs quanto à potência e quanto à queda de projeto. . . . .	24
Tabela 2 – Características básicas do reservatório do Lobo. . . . .	25
Tabela 3 – Parâmetros nominais do sensor HC-SR04. . . . .	35
Tabela 4 – Especificações determinadas do sensor através das medições realizadas no apêndice D . . . . .	46
Tabela 5 – Comparação entre as medidas realizadas pelo sensor e as aferidas através de régua limnimétrica . . . . .	47
Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1). . . . .	63
Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1). . . . .	64
Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1). . . . .	65
Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1). . . . .	66
Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2). . . . .	67
Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2). . . . .	68
Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2). . . . .	69
Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3). . . . .	70
Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3). . . . .	71
Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3). . . . .	72



# LISTA DE ABREVIATURAS E SIGLAS

---

---

ANA	.....	Agência Nacional de Águas
ANEEL	..	Agência Nacional de Energia Elétrica
API	.....	<i>Application Programming Interface</i>
BIG	.....	Banco de Informações de Energia da ANEEL
CGH	.....	Central Geradora Hidrelétrica
DSI	.....	<i>Display Serial Interface</i>
GPIO	.....	<i>General purpose input/output</i>
IaaS	.....	Infraestrutura como Serviço
IOT	.....	<i>Internet of Things</i>
JSON	....	<i>JavaScript Object Notation</i>
M2M	.....	Máquina a Máquina
MQTT	...	Message Queuing Telemetry Transport
MQTT	...	<i>Message Queue Telemetry Transport</i>
PaaS	.....	Plataforma como Serviço
PCH	.....	Pequena Central hidrelétrica
PROINFA		Programa de Incentivo às Fontes Alternativas de Energia Elétrica
RRNN	...	Referência de Nível
SaaS	.....	<i>Software</i> como Serviço
TCP/IP	...	<i>Transmission Control Protocol/Internet Protocol</i>
UHE	.....	Usina Hidrelétrica de Energia
USB	.....	<i>Universal Serial Bus</i>



# SUMÁRIO

---

---

1	<b>INTRODUÇÃO</b>	17
1.1	<b>Motivação</b>	20
1.2	<b>Objetivos</b>	20
2	<b>MATERIAIS E MÉTODOS</b>	21
2.1	<b>Energia Hidrelétrica</b>	21
2.1.1	<i>Penetração da Energia Hidrelétrica na Matriz Elétrica Brasileira</i>	22
2.1.2	<i>Potencial Hidrelétrico Brasileiro</i>	22
2.2	<b>Pequenas Centrais Hidrelétricas</b>	23
2.2.1	<i>Classificação quanto à capacidade de regularização</i>	23
2.2.1.1	<i>PCH a Fio d'Água</i>	23
2.2.1.2	<i>PCH de acumulação, com regularização diária</i>	24
2.2.1.3	<i>PCH de acumulação, com regularização mensal</i>	24
2.2.2	<i>Classificação quanto à potência instalada</i>	24
2.3	<b>Usina do Lobo</b>	24
2.4	<b>Regulamentação da Energia Hidrelétrica</b>	25
2.4.1	<i>Agência Nacional de Energia Elétrica</i>	26
2.4.2	<i>Agência Nacional de Águas</i>	26
2.4.3	<i>Monitoramento Hidrológico no Setor Elétrico</i>	27
2.4.3.1	<i>Estações Limnimétricas Automáticas</i>	27
2.4.3.2	<i>Seções de réguas Limnimétricas</i>	28
2.4.3.3	<i>Referências de Nível</i>	28
2.5	<b>Tecnologia de Computação em Nuvem</b>	28
2.5.1	<i>Software como Serviço</i>	29
2.5.2	<i>Plataforma como Serviço</i>	29
2.5.2.1	<i>IBM Bluemix</i>	30
2.6	<b>Infraestrutura como Serviço</b>	30
2.7	<b>Node RED</b>	30
2.8	<b>Tecnologia de Comunicação MQTT (IoT)</b>	31
2.8.1	<i>Publicante</i>	32
2.8.2	<i>Assinante</i>	32
2.8.3	<i>Broker</i>	32
2.9	<b>Microcontrolador Raspberry Pi B</b>	32

2.10	Sistema operacional <i>Raspbian</i> . . . . .	34
2.11	Biblioteca Eclipse Paho-MQTT . . . . .	34
2.12	Sensor de distância HC-SR04 . . . . .	34
2.12.1	<i>Diagrama de temporização</i> . . . . .	35
3	DESENVOLVIMENTO DO SENSOR E CONEXÃO COM IOT . . . . .	37
3.1	Conexão do sensor HC-SR04 e Raspberry PI . . . . .	38
3.1.1	<i>Interface elétrica HC-SR04 e Raspberry PI</i> . . . . .	38
3.1.2	<i>Interface Software entre o HC-SR04 e o Raspberry PI</i> . . . . .	39
3.2	Configuração da interface IoT com o servidor e <i>Dashboard</i> . . . . .	40
3.3	Integração do sensor à <i>IoT</i> e comunicação com <i>Broker MQTT</i> . . . . .	41
3.3.1	<i>Conexão do sensor à IoT</i> . . . . .	41
3.3.2	<i>Utilização da ferramenta Node-Red</i> . . . . .	41
3.3.3	<i>Visualização de Dados</i> . . . . .	44
4	TESTES PRELIMINARES E APLICAÇÃO EM PEQUENA CENTRAL HIDRELÉTRICA . . . . .	45
4.1	Testes de eficácia . . . . .	45
4.2	Medidas em Campo . . . . .	46
5	CONCLUSÕES . . . . .	49
A	CONFIGURAÇÃO DOS NÓS NA FERRAMENTA NODE-RED . . . . .	51
B	ALGORITMO UTILIZADO NO RASPBERRY PI . . . . .	55
C	UTILIZAÇÃO DA BIBLIOTECA PAHO-MQTT . . . . .	59
D	TABELA DE RESULTADOS - TESTE DE EFICÁCIA DO SENSOR . . . . .	63
	REFERÊNCIAS . . . . .	73

---

## INTRODUÇÃO

---

Há séculos a humanidade vem utilizando variadas formas de energia para a realização de tarefas das mais diversas. Desde a utilização do fogo e da tração animal, nos tempos antigos, o homem vem mostrando como a energia é capaz de realizar tarefas das mais básicas, como aquecer alimentos e a moagem de grãos, até os tempos modernos, em que a energia elétrica é responsável por grande parte das atividades que realizamos diariamente, tornando-se impensável a vida sem ela, ao menos da forma em que a sociedade hoje existe.

Dentre as diversas maneiras de gerar energia elétrica o Homem encontra na água um grande potencial. Segundo o Atlas da Energia Elétrica do Brasil (ANEEL, 2008), a água é o recurso natural mais abundante no mundo, com um volume estimado de 1,63 bilhões de quilômetros cúbicos (km<sup>3</sup>). Sendo também, ainda segundo o Atlas, uma das poucas fontes de energia elétrica não poluente e renovável.

Para produzir energia elétrica através da água é necessária a integração da vazão do rio, a quantidade de água disponível em determinado período de tempo e os desníveis do relevo (naturais ou artificiais). Essa integração e utilização do potencial energético da água é realizado através de usinas, constituídas normalmente por uma barragem, sistema de captação e sistema de adução de água, casa de força e vertedouro (ANEEL, 2008).

A barragem tem o objetivo de interromper o fluxo da água e dar condições à formação do reservatório. Além de ser responsável por criar as condições pelo acúmulo de água, o qual gera armazenamento de energia potencial (que será convertida em elétrica), os reservatórios também exercem função de regularizar o regime dos rios, permitindo o armazenamento de água em tempos em que o fluxo do rio é superior ao necessário para uso em períodos de estiagem. Portanto, a energia hidrelétrica também possui como vantagem a capacidade de armazenar energia através dos reservatórios e dispêndia-la quando necessário (MAUAD; FERREIRA; TRINDADE, 2017).

Sendo uma parte considerável das obras civis de uma usina hidrelétrica, os reservatórios

de água merecem então, especial atenção. Assim, a resolução conjunta ANEEL/ANA nº 03, de 10 de agosto de 2010, publicada em 20 de outubro de 2010, é responsável por estabelecer os procedimentos e condições que devem ser observados pelos concessionários e autorizados de geração de energia elétrica para a instalação, operação e manutenção de estações pluviométrico, limnimétrico, fluviométrico, sedimentométrico e de qualidade da água associado a aproveitamentos hidrelétricos (ANA, 2014).

Dentre os procedimentos citados pela resolução está o acompanhamento do nível de água do rio (limnometria), realizado através de réguas limnimétricas, grandes réguas cujo objetivo é permitir a leitura visual do nível de água dos rios, e também por estações automáticas, que permitem a leitura contínua do nível de água do rio, assim garantindo maior monitoramento do nível de água.

Ainda no que tange reservatórios de água, o risco de ruptura da barragem por galgamento (quando a água ultrapassa a altura máxima da barragem) faz necessário o monitoramento do nível de água, permitindo ações rápidas quando este nível cresce acima do esperado (SANTOS; FRANCA; ALMEIDA, 2007). Ações essas incluem, por exemplo, a abertura das comportas dos vertedouros, eliminando o volume de água acima do suportado pela barragem. É evidente que ações como essa geram desperdício de energia, pois a água que passa pelos vertedouros não gera energia elétrica.

O monitoramento de um reservatório é demonstrado por Li e Wang (2017). Em seu trabalho, os autores demonstram como dispositivos conectados à Internet das Coisas (do inglês: *Internet of Things* (IOT)) permitem a agregação de segurança a processos de manutenção de um reservatório de água. Os autores descrevem o desenvolvimento e implementação de um sensor integrado à *IoT*, capaz de medir em tempo real variáveis como nível da água e *pH*, garantindo não só a manutenção da qualidade da água como a supervisão da segurança da barragem, emitindo alertas quando o nível do reservatório ultrapassa limites pré estabelecidos.

A Internet das Coisas é um conceito que considera a crescente entrada de dispositivos conectados e interligados e que por meio de esquemas de conexão diversos são capazes de interagir entre si e com o meio, criando uma rede de objetos (coisas) capazes de cooperar entre si em tempo real com o intuito de atingir um objetivo comum (VERMESAN; FRIESS, 2015).

Nesse contexto, a possibilidade de conectar praticamente todo e qualquer dispositivo a uma rede de objetos e torná-los inteligentes abre portas para um mundo em que os processos de geração de energia, transporte, sustentabilidade entre outros se tornam mais velozes, eficientes e conectados, tornando a cadeia entrelaçada de objetos e eventos uma cadeia inteligente e capaz de adaptar-se ao ambiente e a si própria.

*IoT* tem por objetivo permitir que as coisas estejam conectadas o tempo todo entre si e com as pessoas, através de qualquer rede ou caminho existente e através de qualquer serviço. Assim, a Internet das Coisas se configura como uma revolução da internet como um

todo. Os objetos são capazes de se reconhecer, reconhecer aos outros e passar informações valiosas sobre seu estado próprio, ao mesmo tempo em que recebem informações do ambiente e de outros objetos, dessa forma, sendo capazes de processar mais informação, gerando dados mais complexos e úteis, em uma sociedade em que a informação cada vez mais possui valor (VERMESAN; FRIESS, 2015).

O potencial da Internet das Coisas é mostrado pelo seu rápido crescimento. Até 2020 espera-se que o número de objetos conectados à internet atinja 38 bilhões, representando mais do que 4 vezes o número de seres humanos na Terra (SHAH; YAQOOB, 2016). Esse dado é capaz de dar amostra da capacidade de utilização de uma rede de objetos conectados, fato este representado pela, já atual, aplicabilidade da Internet das Coisas a tarefas rotineiras e não-rotineiras, tais como em processos de logística, estacionamento inteligente de veículos, monitoramento de dados e controle de reservatórios e mananciais (SHAH; YAQOOB, 2016).

Alguns trabalhos podem ser citados como exemplos, demonstrando campos em que a *IoT* já vem trazendo benefícios, entre eles, Lekić e Gardašević (2018) demonstram, ao utilizar-se de um Raspberry PI e de um sensor de temperatura e umidade, ambos anteriormente *offline* e então conectados à Internet das Coisas, a viabilidade de utilizar-se de dispositivos integrados à Internet das Coisas como meio de se obter medidas de variáveis do ambiente, por meio de um sistema eficiente e de baixo custo.

Jamborsalamati *et al.* (2017) demonstram a implementação de um sistema conectado à *IoT* como forma de adicionar inteligência a um sistema de controle de alimentação de dispositivos conectados a uma fonte de energia fotovoltaica. Através da integração de conexão bi-direcional entre dispositivos medidores, um dispositivo capaz de processar dados e controladores, os autores evidenciam a capacidade que a adição de inteligência aos processos tem de aumentar a eficiência de processos energéticos, agregando valor à informação pura por meio de processamento de dados em tempo real. Dados esses colhidos em diferentes pontos e com processamento unificado através da comunicação via protocolo *Message Queue Telemetry Transport* (MQTT), o protocolo de estado da arte de *IoT*.

Tendo como foco a utilização da internet das coisas como meio para transporte, processamento e utilização de dados, Dhandre, Kamalasekaran e Pandey (2016) demonstram a capacidade de se utilizar dispositivos conectados a *IoT* como meio de aumentar a confiabilidade e a velocidade com que os dados são gerados, permitindo tomada de decisões mais rápidas ou mesmo viabilizando decisões que anteriormente não eram possíveis. No artigo, os autores demonstram o desenvolvimento de um sensor integrado de temperatura, umidade, nível de água e condições climáticas aplicado a um reservatório de água. Obtendo como resultado a disponibilização em tempo real de dados e a geração de alertas.

Através das pesquisas aqui citadas, em especial Li e Wang (2017) e Dhandre, Kamalasekaran e Pandey (2016), fica evidente a capacidade que a interligação de dispositivos anteriormente *offline* tem de corresponder às necessidades geradas pela resolução conjunta aqui citada, bem

como aumentar a segurança através do monitoramento constante do nível de água em uma barragem.

## 1.1 Motivação

Este trabalho encontra, então, motivação na crescente e já grande gama de aplicações que a Internet das Coisas demonstra apresentar.

Tendo em vista a possibilidade de se medir e utilizar de dados de reservatórios de água, o projeto propõe o desenvolvimento e o teste de um sistema sensor de nível de água conectado à Internet das Coisas. Sistema este a ser aplicado e testado em condições reais de operação, em uma barragem de uma pequena central hidrelétrica, a represa do Lobo, em Itirapina - SP.

Busca-se, através deste sistema, a geração de valor não só através da medição e disponibilização em tempo real de dados de nível do reservatório, mas também a capacidade de processar esses dados em sistemas hospedados na nuvem, como a emissão de alertas para o caso em que o sistema ultrapasse o limites de segurança estabelecidos.

## 1.2 Objetivos

Como objetivos específicos deste trabalho, citam-se:

1. Criação de um sensor de nível de água conectado à Internet das Coisas via protocolos de comunicação Message Queuing Telemetry Transport (MQTT).
2. Desenvolver um sistema de processamento hospedado na nuvem com o objetivo de processar, armazenar e disponibilizar os dados medidos.
3. Agregar ao sistema de processamento baseado em nuvem a capacidade de emitir alertas quando as variáveis medidas ultrapassarem valores definidos pelo usuário, através de e-mail ou outra forma de comunicação.
4. Disponibilizar um *Dashboard* interativo *online*, permitindo o acesso aos dados bem como a emissão de relatórios dos dados medidos.

---

## MATERIAIS E MÉTODOS

---

Este capítulo tem por objetivo descrever os materiais e métodos utilizados no presente trabalho, assim, pretende-se descrever aqui as tecnologias utilizadas bem como a utilização destas.

O capítulo se divide nas seguintes subseções, organizadas conforme tecnologia, método ou procedimento utilizado.

- a) Energia Hidrelétrica
- b) Pequenas Centrais Hidrelétricas
- c) Usina do Lobo
- d) Regulamentação da Energia Elétrica
- e) Tecnologia de Computação em Nuvem
- f) Node RED
- g) Microcontrolador Raspberry Pi B
- h) Sistema Operacional Rasbian
- i) Biblioteca Eclipse Paho-MQTT
- j) Sensor de Distância HC-SR04

### 2.1 Energia Hidrelétrica

O aproveitamento do fluxo de água em usinas é responsável pela geração de energia elétrica. A este aproveitamento dá-se o nome de Energia Hidrelétrica. Porém, diferentemente de outros tipos de aproveitamento, a utilização das energias das águas para geração de energia necessita de empreendimentos em que a construção de obras civis chega a ser tão ou até mais importante que o maquinário utilizado (ANEEL, 2008).

Tendo origem no final do século XIX, as usinas hidrelétricas tiveram seu tamanho e eficiência altamente aumentados ao longo das décadas. Inicialmente com 0,5MW (Potência da primeira usina hidrelétrica brasileira), as usinas atualmente atingem capacidades de geração de até 14 mil MW, como é o caso de Itaipu Binacional, uma das maiores usinas hidrelétricas do mundo (ANEEL, 2008).

É, então, a capacidade instalada de uma usina hidrelétrica que determina sua classificação em porte. Assim, segundo ANEEL (2008), as usinas hidrelétricas podem ser classificadas em médio ou grande porte ou em Pequenas Centrais Hidrelétricas (PCHs). Sendo adotado:

- Central Geradora Hidrelétrica (CGH): até 1MW de potência instalada;
- Pequena Central hidrelétrica (PCH): entre 1,1MW e 30MW de potência instalada;
- Usina Hidrelétrica de Energia (UHE): potência instalada superior a 30MW.

### ***2.1.1 Penetração da Energia Hidrelétrica na Matriz Elétrica Brasileira***

Segundo o Banco de Informações de Energia da ANEEL (BIG) da Agência Nacional de Energia Elétrica (ANEEL), em 2018 (acesso em setembro/2018) havia 7.125 empreendimentos de geração de energia elétrica no Brasil, totalizando 159.986.609 kW de potência instalada. Desta potência instalada, 63,88% é gerada por meio de aproveitamentos hidrelétricos, distribuídos em 686 CGHs (0,42% da capacidade instalada), 429 PCHs (3,2% da capacidade instalada) e 220 UHEs (60,26% da capacidade instalada).

Os dados da ANEEL indicam predominância de UHEs na matriz hidrelétrica (94,3%), o que, entretanto, não descaracteriza a importância das PCHs ou das CGHs, ainda mais frente a iniciativas como Programa de Incentivo às Fontes Alternativas de Energia Elétrica (PROINFA), criado pela Lei nº 10.438/2002, cujo intuito foi incentivar a participação de fontes alternativas renováveis (pequenas centrais hidrelétricas, usinas eólicas e empreendimentos termelétricos a biomassa) na produção de energia elétrica, privilegiando empreendedores que não tinham vínculos societários com concessionárias de geração, transmissão ou distribuição.

### ***2.1.2 Potencial Hidrelétrico Brasileiro***

Segundo o Plano 2015 da Eletrobrás, o Brasil é o país com maior potencial hidrelétrico do mundo, com 260 mil MW disponíveis para utilização, dos quais aproximadamente 101 mil MW estão utilizados em potência fiscalizada (Banco de Informações de Geração (BIG) da Aneel - 2018).

Espera-se que até 2030 sejam aproveitados 126.000 MW deste potencial ((BRASIL), 2007), dos quais 70% se encontram nas bacias do Amazonas e do Tocantins/Araguaia (ANEEL, 2008).

## 2.2 Pequenas Centrais Hidrelétricas

A estrutura básica das microcentrais hidrelétricas é formada por elementos que têm a função de captar e conduzir a água para uma casa de máquinas, onde se processará a transformação da energia hidráulica em elétrica, e um canal que restituirá a água ao rio. Seus componentes básicos são: barragem, tomada d'água, sistema de adução, câmara de carga, chaminé de equilíbrio, casa de máquinas, turbina e gerador. (MAUAD; FERREIRA; TRINDADE, 2017)

Segundo resolução da ANEEL, 673, de 4 de agosto de 2015,

Serão considerados empreendimentos com características de PCH aqueles empreendimentos destinados a autoprodução ou produção independente de energia elétrica, cuja potência seja superior a 3.000 kW e igual ou inferior a 30.000 kW e com área de reservatório de até 13 km<sup>2</sup> excluindo a calha do leito regular do rio.(ANEEL, 2015).

As PCHs também podem ser classificadas quanto à capacidade de regularização e quanto à queda de projeto.

### 2.2.1 Classificação quanto à capacidade de regularização

Faz-se a classificação das PCHs em 3 tipos, de acordo com a capacidade de regularização do reservatório, sendo tais (ELETROBRÁS, 2000):

- a) a Fio d'Água;
- b) de Acumulação, com Regularização Diária do Reservatório;
- c) de Acumulação, com Regularização Mensal do Reservatório.

#### 2.2.1.1 PCH a Fio d'Água

São classificadas com PCHs a Fio d'Água aquelas PCHs cuja vazão de estiagem do rio é maior ou igual que a descarga necessária à potência a ser instalada para atender à demanda máxima prevista.

Assim, o volume do reservatório criado pela barragem da PCH é desprezível, seu aproveitamento energético é parcial e o vertedouro funciona em quase toda totalidade do tempo.

Classificação das Centrais	Potência - P (kW)	Queda de Projeto - Hd (m)		
		Baixa	Média	Alta
Micro	$P < 100$	$Hd < 15$	$15 < Hd < 50$	$Hd > 50$
Mini	$100 < P < 1.000$	$Hd < 20$	$20 < Hd < 100$	$Hd > 100$
Pequena	$1.000 < P < 30.000$	$Hd < 25$	$25 < Hd < 130$	$Hd > 130$

Fonte: ELETROBRÁS (2000).

Tabela 1 – Classificação das PCHs quanto à potência e quanto à queda de projeto.

### 2.2.1.2 PCH de acumulação, com regularização diária

Este tipo de PCH é empregado quando as vazões de estiagem do rio são inferiores ao necessário para fornecer a potência à demanda máxima. Nesse caso, o reservatório fornece o adicional de vazão regularizada.

### 2.2.1.3 PCH de acumulação, com regularização mensal

Quando o projeto da PCH considera dados de vazões mensais no dimensionamento energético, analisando os dados de vazões mensais, pressupõe-se uma regularização mensal das vazões médias diárias.

## 2.2.2 Classificação quanto à potência instalada

As PCHs também podem ser classificadas de acordo com a potência instalada e a altura de queda do projeto, sendo necessário considerar os dois parâmetros conjuntamente, segundo dados da tabela 1.

Centrais com alta e média queda, com alto desnível natural, possuem a casa de força afastada da estrutura de barragem. Dessa forma, é comum que o circuito hidráulico de adução envolva canal ou conduto de baixa pressão com longa extensão.

As centrais de baixa queda possuem normalmente, em contrapartida, casa de força junto à barragem, sendo a adução realizada através de tomada d'água incorporada ao barramento.

## 2.3 Usina do Lobo

A Usina do Lobo (Figura 1) (também conhecida por Usina do Broa) foi construída em 1936 com a finalidade de produzir energia. Desde a década de 70 o entorno da usina tem sido loteado, transformando-a também em um polo regional de lazer e de pesquisa (MATOS, 2012). Segundo ANEEL (2018) a usina é pertencente atualmente à Aratu Geração S/A, com potência de 1.272 kW, sendo portanto uma PCH.

A tabela 2 indica as principais características do reservatório.



Figura 1 – Vista da Represa do Lobo.

Fonte: Matos (2012).

Latitude	22°15'S
Longitude	47°49'O
Comprimento máximo	8 km
Largura máxima	2 km
Largura média	0,9 km
Profundidade máxima	12,0 m
Profundidade média	3,0 m
Área de superfície	6,8 km <sup>2</sup>
Perímetro	21 km
Volume	22x10 <sup>6</sup> m <sup>3</sup>
Altitude máxima	940 m
Altitude média	770 m
Altitude mínima	680 m

Fonte: Adaptada de Tundisi *et al.* (2004).

Tabela 2 – Características básicas do reservatório do Lobo.

## 2.4 Regulamentação da Energia Hidrelétrica

De forma a regulamentar o setor energético brasileiro e a utilização das águas em posse da União, foram criadas no final do século XX e início do XXI agências reguladoras especializadas, a Agência Nacional de Energia Elétrica (ANEEL) e a Agência Nacional de Águas (ANA) (BRASIL, 1997b),(BRASIL, 1996).

Essas agências possuem entre suas principais atribuições a regulamentação e controle das principais atividades relacionadas à Energia Elétrica e aos corpos de água em posse da

União, respectivamente, além de serem responsáveis, conjuntamente, pelo monitoramento hidrológico no setor elétrico.

### **2.4.1 Agência Nacional de Energia Elétrica**

De forma a regulamentar o setor energético brasileiro, foi criada em 1997, por meio da Lei nº 9.427/1996 e do Decreto nº 2.335/1997 a ANEEL, autarquia em regime especial vinculada ao Ministério de Minas e Energia (BRASIL, 1996), (BRASIL, 1997a).

A ANEEL tem suas atividades e competências regulamentadas pela lei que aprova sua criação, assim sendo suas principais atribuições(ANEEL, Entre 2010 e 2018):

- a) Regular a geração, transmissão, distribuição e comercialização de energia elétrica no Brasil;
- b) Fiscalizar diretamente ou mediante convênios as concessões, premissões e os serviços de energia elétrica no Brasil;
- c) Implementar as políticas e diretrizes do governo federal relativas à exploração da energia elétrica e ao aproveitamento dos potências hidráulicos;
- d) Estabelecer a metodologia para o cálculo tarifário;
- e) Mediar conflitos entre concessionárias, permissionárias, autorizadas, produtores independentes e autoprodutores, bem como entre esses agentes e seus consumidores;
- f) Promover as atividades de outorgas de concessão, permissão e autorização de empreendimentos e serviços de energia elétrica, por delegação do Governo Federal.

### **2.4.2 Agência Nacional de Águas**

Criada pela lei nº 9.984 de 2000 (BRASIL, 1997b), a Agência Nacional de Águas (ANA) é uma agência reguladora vinculada ao Ministério do Meio Ambiente (MMA) dedicada a fazer cumprir os objetivos e diretrizes da Lei das Águas do Brasil, a lei nº 9.433 de 1997 (BRASIL, 2000).

A Agência Nacional de Águas possui em suas principais atribuições (ANA, Entre 2010 e 2018):

- a) Regular o uso e acesso aos recursos hídricos em domínio da União, assim como os serviços públicos de irrigação e adução de água bruta;
- b) Monitorar a situação dos recursos hídricos do Brasil. Coordena a Rede Hidrometeorológica Nacional que capta, com o apoio dos estados e outros parceiros, informações como nível, vazão e sedimentos dos rios ou quantidade de chuvas;
- c) Coordenar a implementação da Política Nacional de Recursos Hídricos, realizando e dando apoio a programas e projetos, órgãos gestores estaduais e à instalação de comitês e agências de bacias;

- d) Elaborar ou participar de estudos estratégicos, como os Planos de Bacias Hidrográficas, Relatórios de Conjuntura dos Recursos Hídricos, entres outros, em parceria com instituições e órgãos do poder público.

### **2.4.3 Monitoramento Hidrológico no Setor Elétrico**

A Resolução Conjunta ANEEL/ANA nº 03, de 10 de agosto de 2010, publicada em 20 de outubro de 2010 (ANEEL; ANA, 2010), é responsável por estabelecer os procedimentos e condições que devem ser observados pelos concessionários e autorizados de geração de energia elétrica para a instalação, operação e manutenção de estações pluviométrico, limnimétrico, fluviométrico, sedimentométrico e de qualidade da água associado a aproveitamentos hidrelétricos.

Assim, a ANA adquire a função de orientar os agentes do setor elétrico sobre procedimentos de coleta, tratamento, armazenamento e envio dessas informações.

Dentre os documentos orientativos disponibilizados junto a esta resolução conjunta está o Manual "ORIENTAÇÕES PARA ELABORAÇÃO DO RELATÓRIO DE INSTALAÇÃO DE ESTAÇÕES HIDROMÉTRICAS"(ANA, 2014).

A ANA determina os parâmetros e especificações a serem adotados na instalação de estações limnimétricas, responsáveis pela medição do nível do reservatório. Assim, no Manual, a ANA especifica a instalação dos seguintes itens.

- a) Estações Limnimétricas Automáticas;
- b) Seções de Régua;
- c) Referências de Nível.

#### *2.4.3.1 Estações Limnimétricas Automáticas*

As estações limnimétricas automatizadas devem ser instaladas de acordo com o tipo de sensor e características do local de instalação, considerando robustez suficiente para garantir a qualidade e continuidade das medições.

São definidos dois métodos automatizados para medição do nível da água, sendo esses:

- a) Transdutor de pressão;
- b) Sensor ultrassônico.

Os dados de nível devem ser referenciados em relação ao nível do mar (nível absoluto) ou ao nível mínimo do corpo d'água (nível relativo). Para tal, deve ser instalado um marco de referência de nível, conforme seção 2.4.3.3.

O tipo, modelo e tecnologia das estações limnimétricas de coleta de dados é de livre escolha, devendo respeitar os seguintes requisitos(ANA, 2014):

- Resolução: igual ou inferior a 5mm;
- Faixa de medição: Compatível com a variação do corpo d'água;
- Exatidão: +- 1cm para corpos d'água com variação máxima de nível de até 10m ou 0,1% da faixa de variação máxima do nível para corpos d'água com faixa de variação acima de 10m;
- Condições ambientais de operação: Temperatura (-10°C a +55°C) e umidade relativa de 0% a 100%.

#### 2.4.3.2 Seções de réguas Limnimétricas

As estações fluviométricas devem possuir lances de réguas limnimétricas, cujos dados possam ser registrados pela leitura de observadores, a fim de aferir a precisão das medidas do sensor de nível, por meio de redundância, bem como assegurar a continuidade das medições quando o equipamento automático apresentar problemas.

Normalmente são utilizadas réguas em modelos de alumínio, fibra de vidro ou PVC, enumeradas de 2 em 2 cm.

As réguas são instaladas em alinhamento perpendicular ao eixo do rio. Para o caso de estações limnimétricas construídas em barragens, a orientação é que as réguas sejam construídas ou fixadas diretamente na estrutura da barragem (ANA, 2014).

#### 2.4.3.3 Referências de Nível

Nas estações fluviométricas devem ser instaladas e mantidas Referência de Nível (RRNN) padronizadas, construídas em chapas de bronze, alumínio ou substituto similar, numeradas e com indicação clara do desnível geométrico em relação ao zero da régua.

Sua instalação deve ser feita em base de concreto, ou se possível, chumbada em estrutura natural ou artificial de caráter permanente.

Devem ser instaladas no mínimo duas RRNN em cada estação, sendo que a RRNN principal deverá ser instalada acima da cota de enchente máxima.

Uma vez por ano deve ser realizado o nivelamento de todos os lances de réguas das estações fluviométricas, a partir das RRNNs, com o objetivo de garantir a correta leitura das oscilações dos níveis do rio (ANA, 2014).

## 2.5 Tecnologia de Computação em Nuvem

Tecnologias de computação em nuvem representam a entrega de recursos computacionais e capacidade de armazenamento como um serviço para uma variada gama de usuários finais. Seu conceito se baseia no compartilhamento destes recursos através de alguma

rede de comunicação, normalmente a internet. O usuário acessa o sistema através de um dispositivo local, como um microcontrolador, um computador ou um celular, enquanto que os serviços de processamento e armazenamento de dados são realizados em uma localização remota (LEKIĆ; GARDAŠEVIĆ, 2018).

O serviço de computação em nuvem permite ao usuário final, em qualquer tempo, o acesso a dados, processamento e serviços de armazenamento sem que a localização física do sistema tenha que ser conhecida. Essa tecnologia é, assim, muito utilizada em serviços como redes sociais, e-mails ou *Smartphones*. Outrossim, os serviços de computação em nuvem disponibilizam uma grande quantidade de poder de processamento e memória quando estes são necessários, adaptando-se às necessidades do consumidor final (LEKIĆ; GARDAŠEVIĆ, 2018).

Um serviço de IoT pode se classificar em uma de três categorias (IBM, 2017):

- a) *Software* como Serviço (SaaS);
- b) Plataforma como Serviço (PaaS);
- c) Infraestrutura como Serviço (IaaS).

A base da tecnologia de computação em nuvem é uma infraestrutura convergente de diferentes serviços de tecnologia da informação ligados a uma unidade lógica funcional, como abstração de recursos físicos pela virtualização bem como o compartilhamento de recursos.

Distinguem-se duas partes do serviço, o *back-end*, que inclui a infraestrutura dos provedores de serviço de nuvem, e o *front-end*, abrangendo todas as partes da infraestrutura controladas pelo usuário final bem como sua própria forma de acessar o sistema (LEKIĆ; GARDAŠEVIĆ, 2018).

Em seguida se descrevem as 3 categorias de serviço existentes:

### **2.5.1 Software como Serviço**

Software como Serviço provêem acesso a aplicações operando na nuvem. Normalmente, muitos clientes têm acesso compartilhado aos serviços e bancos de dados encontrados na nuvem. Assim, um dos principais atributos de uma solução de SaaS é que o *software*, o *hardware* e os dados podem ser usados seguramente por vários usuários ao mesmo tempo. Aplicações comuns de SaaS são: *Softwares* de planejamento de recursos e outras aplicações específicas de indústrias (IBM, 2017).

### **2.5.2 Plataforma como Serviço**

Plataforma como serviço envolve a disponibilização pelo provedor de um ambiente de desenvolvimento bem como os subsistemas de *Software* requeridos. O usuário pode de-

envolver, testar e distribuir suas próprias aplicações que são executadas na infraestrutura de nuvem disponibilizada pelo provedor do serviço. O provedor é responsável por disponibilizar a plataforma e o ambiente executável, normalmente envolvendo servidores, infraestrutura de redes, *data-centers*, sistemas operacionais e linguagens de programação. O usuário tem controle sobre as aplicações e a camada média de processamento, normalmente podendo escolher a estrutura do ambiente.

Plataformas como serviço implicam o sistema operacional na nuvem, sem necessidade de *download* e instalação. Nesse tipo de serviço as ferramentas de desenvolvimento estão localizadas na nuvem e acessadas via navegador (LEKIĆ; GARDAŠEVIĆ, 2018).

### 2.5.2.1 IBM Bluemix

O IBM Bluemix é um serviço de Plataforma como Serviço baseado no *Cloud Foundry*, um projeto *open source*. Suporta muitas línguas de programação diferentes, assim como serviços integrados para construir, executar e distribuir aplicações na nuvem (IBM, Entre 2012 e 2018).

## 2.6 Infraestrutura como Serviço

IaaS proveem ao usuário recursos computacionais básicos como servidores, redes, armazenamento e espaço em *data-centers*. IaaS pode ser considerada como a base para a computação em nuvem. IaaS pode ser utilizado por usuários ou provedores de serviço que desenvolvem soluções mais avançadas e específicas em cima de uma infraestrutura básica fornecida por uma serviço da categoria IaaS (IBM, 2017).

## 2.7 Node RED

Node RED é uma ferramenta *open-source* baseada em fluxo para a integração através de IoT de *hardwares*, *Application Programming Interface* (API) e serviços *online* desenvolvida pela *IBM Emerging Technology*. NODE RED é uma ferramenta grátis, baseada em Java Script, construída em uma plataforma Node.js, que provêem uma plataforma *online*, baseada em navegador, para a edição de fluxos. O sistema contém nós representados por ícones e pode funcionar de duas formas distintas: pela adição dos ícones correspondentes ou pela importação de código JavaScript. Os nós proveem funções variadas, como uma função *debug* para leitura *online* de variáveis, nós para conexão direta com portas *General purpose input/output* (GPIO) de um Raspberry Pi. Os nós criados são armazenados na notação *JavaScript Object Notation* (JSON) (NODE-RED, 2017).

NODE RED permite aos desenvolvedores conectar nós de entrada, saída e processamento de dados, controlando *hardwares* e enviando alertas. Funciona através da conexão dos nós

entre si ou com objetos, permitindo o envio de alertas através de e-mail, protocolos de *IoT* (como MQTT) e facilmente realizando o processamento de dados.

NODE RED também permite a fácil implementação de protótipos, especialmente aqueles envolvidos com tecnologia *IoT*, revelando a capacidade desta ferramenta de permitir a criação de ferramentas em tempo real para dispositivos fim, como a placa Raspberry Pi e os sensores a ser utilizados no presente trabalho (LEKIĆ; GARDAŠEVIĆ, 2018).

## 2.8 Tecnologia de Comunicação MQTT (IoT)

O protocolo de comunicação *Message Queue Telemetry Transport* (MQTT) foi inventado em 1999 de forma a por Andy Stanford-Clark e Arlen Nipper (IBM e Arcom respectivamente). O objetivo foi criar um protocolo leve, de forma a conectar transmissão de óleo através de comunicação pela internet. Os objetivos para esta tecnologia, eram, basicamente, um protocolo capaz de estabelecer comunicação de dados com seção contínua, simples de implementar, com boa qualidade de serviço e com boa utilização de largura de banda. Mesmo que o foco deste protocolo tenha mudado para implementações em *IoT*, estas continuam sendo suas características principais (VELEZ *et al.*, 2018).

O protocolo MQTT se baseia em um modelo de comunicação de publicação e subscrição de dados e é atualmente um modelo estado da arte para a transmissão de mensagem publicador/assinante. Existem nesse protocolo 3 principais elementos, a serem descritos: Assinante, Publicante e *Broker* (JAMBORSALAMATI *et al.*, 2017).

O principal aspecto do sistema publicação/assinante é o desacoplamento entre o publicante e o assinante. Os dois podem trabalhar simultaneamente sem conhecer a existência um do outro e a operação de cada componente não é interrompida. O terceiro elemento, o *Broker*, é responsável por distribuir as mensagens e fazer com que cheguem ao destino correto.

O compartilhamento de dados através de MQTT acontece via tópicos específicos e customizáveis. Assim, o publicante envia dados para um tópico pré-determinado e o assinante deve conhecer previamente quais tópicos pretende assinar. Em suma, pode-se dizer que o MQTT é um protocolo bi-direcional, o que significa que um dispositivo pode ser tanto um assinante quanto um publicante (JAMBORSALAMATI *et al.*, 2017).

A filtragem das mensagens por meio de tópicos permite que o assinante receba suas mensagens de interesse, desacoplando-o do publicante. Outrossim, este protocolo permite ganho de escala, uma vez que é possível utilizar um *broker* de forma paralela, distribuindo as mensagens para uma grande quantidade de assinantes.

A conexão MQTT é realizada através de *Transmission Control Protocol/Internet Protocol* (TCP/IP), assim, tanto o cliente quanto o *broker* devem possuir conexão TCP/IP. A conexão é sempre realizada entre um cliente e o *broker*, e nunca entre dois clientes (VELEZ *et al.*, 2018).

Os três elementos participantes da comunicação podem ser então melhor definidos:

### 2.8.1 *Publicante*

Um publicante MQTT pode ser qualquer dispositivo, sendo assim, tanto um microcontrolador quanto um servidor hospedando um serviço MQTT. Um publicante deve definir dois elementos antes de enviar uma mensagem, o conteúdo (*payload*) da mensagem assim como seu tópico.

Um tópico é uma *string* que pode conter diferentes níveis e atributos, separados por barras. É o tópico que define quais assinantes receberão a mensagem enviada (JAMBORSALAMATI *et al.*, 2017).

### 2.8.2 *Assinante*

Assim como um publicante, um assinante pode ser qualquer dispositivo que possua conexão TCP/IP e esteja conectado a alguma rede, além de ser capaz de se comunicar através de MQTT. O assinante deve conhecer previamente os tópicos que deseja assinar para ser capaz de receber as mensagens.(JAMBORSALAMATI *et al.*, 2017)

### 2.8.3 *Broker*

O *Broker* é a conexão central, responsável por receber e filtrar todas as mensagens e as distribuir para os assinantes correspondentes. Um *Broker* MQTT pode lidar com milhares de assinantes e publicantes, sendo também responsável pelo processo de autorização dos clientes, como por meio de uma senha, se desejado.

É o *broker* que é diretamente exposto à internet, realizando a conexão entre publicante e assinante, já que esta nunca é realizada entre dois clientes.

A conexão MQTT é inicializada por um procedimento de *hand-shake* com um cliente. Na inicialização, o cliente envia ao *broker* uma mensagem solicitando a conexão, a qual o *broker* responde com uma mensagem de status. Após iniciada a conexão, esta é mantida até que uma mensagem de desconexão seja enviada pelo cliente (JAMBORSALAMATI *et al.*, 2017).

## 2.9 Microcontrolador Raspberry Pi B

O Raspberry Pi é uma placa de circuito integrado construído pela Raspberry Pi Foundation. Seu desenvolvimento começou por Eben Upton, cujo principal propósito com suas placas iniciais era o de incentivar crianças e jovens a se interessar mais por programação.

No centro de toda placa Raspberry Pi está um *chip* integrado Broadcom BCM2835, responsável pelo processamento geral da placa, renderização gráfica e capacidades de entrada e saída.

Além deste chip as placas contêm módulos de memória RAM, responsáveis por armazenar a memória temporária enquanto os programas são executados. RAM se traduz como Memória de Acesso Aleatório, significando que o processador pode ler e escrever dados nessa memória a qualquer hora. Pela sua característica volátil, a memória RAM tem seus dados perdidos quando a placa Raspberry Pi é desenergizada.

Mais ainda, o Raspberry Pi possui uma porta HDMI, uma porta GPIO, uma porta *Display Serial Interface* (DSI), uma porta CSI (*Camera Serial Interface*), um conector micro-USB utilizado para alimentação da placa e um *slot* para um cartão SD, responsável pelo armazenamento da memória não-volátil (como o sistema operacional, dados, programas e outros arquivos). O modelo utilizado também possui 4 portas *Universal Serial Bus* (USB), para conexão de periféricos, como teclado e mouse e uma porta *Ethernet* (UPTON; HALFACREE, 2016).

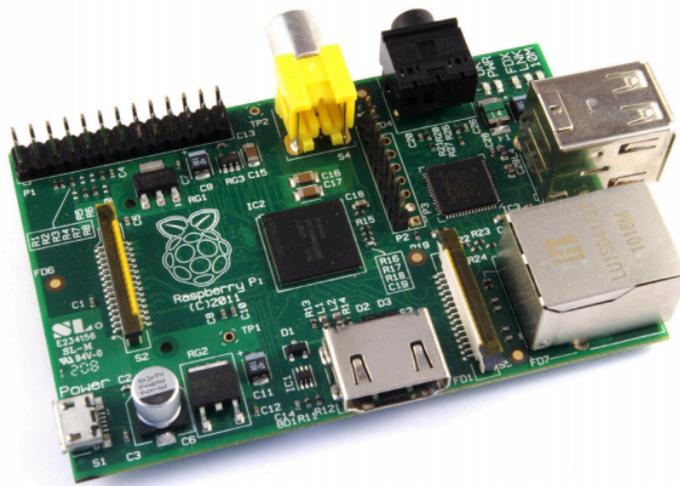


Figura 2 – Placa Raspberry Pi 3.

Fonte: Upton e Halfacree (2016).

## 2.10 Sistema operacional *Raspbian*

O sistema operacional *Raspbian* é um sistema baseado em Debian, otimizado ao *hardware* do Raspberry PI. Um sistema operacional é um conjunto básico de programas e utilidades que permite que o Raspberry PI opere. Entretanto, *Raspbian* é mais do que um simples sistema operacional: *Raspbian* já possui 35000 pacotes pré compilados de software, facilmente instaláveis no Raspberry PI.

A *build* inicial do *Raspbian*, completada em 2012, possui mais de 35000 pacotes otimizados para a melhor performance no Raspberry PI, tendo o lançamento de novas versões focado cada vez mais na estabilidade e performance da maior quantidade de pacotes Debian possível (RASPBIAN, entre 2012 e 2017).

## 2.11 Biblioteca Eclipse Paho-MQTT

De forma a permitir a efetiva comunicação do sistema *Raspbian* com o *broker* MQTT é necessária uma biblioteca implementando o protocolo MQTT. Essa biblioteca é denominada Eclipse Paho - MQTT.

A biblioteca Eclipse Paho é um projeto *open source* sobre o projeto Eclipse Technology. O escopo do projeto Paho é promover implementações de código aberto para protocolos de comunicação padrões e abertos que suportam as necessidades emergentes de integração Máquina a Máquina (M2M) com aplicações da Web e Softwares.

De forma a integrar a comunicação M2M entre dispositivos e clientes, Paho endereça o desenvolvimento de *frameworks* e código que permitam o teste e desenvolvimento de conectividade *end-to-end* com um servidor. O escopo do projeto Paho inclui o desenvolvimento de um ferramental que permita o efetivo uso, integração e teste de componentes de mensagem via protocolo MQTT (ECLIPSE-PAHO, 2015).

A utilização da biblioteca Paho MQTT é melhor descrita no apêndice C.

## 2.12 Sensor de distância HC-SR04

O sensor HC-SR04 é um sensor de distância que realiza medições baseadas em pulsos de ultrassom, com distância entre 2cm e 400cm e precisão de até 3mm. Um módulo inclui transmissores ultrassônicos, receptor e circuito de controle. Os princípios básicos de trabalho: (ELEC FREAKS, 2011)

- a) Ativar a porta *trigger* IO por pelo menos 10ms com um sinal de nível alto;
- b) O módulo automaticamente envia 8 pulsos de 40kHz e aciona a saída na porta *echo*;
- c) Se houve detecção de volta, a porta de saída (*echo*) é desativada.

Assim, a distância entre o medidor e o objeto é então calculada pela fórmula:

$$d_{medida} = \Delta_t * v_{som} / 2 \quad (2.1)$$

Onde,

- $d_{medida}$  é a distância medida;
- $\Delta_t$  é a diferença entre o tempo de início e o final do sinal de *echo*;
- $v_{som}$  é a velocidade do som;

Os parâmetros nominais do sensor são explicitados como na tabela 3.

Tensão de trabalho	DC 5 V
Frequência de trabalho	40Hz
Distância máxima	4m
Distância mínima	2cm
Ângulo de medição	15°

Fonte: Elec Freaks (2011).

Tabela 3 – Parâmetros nominais do sensor HC-SR04.

### 2.12.1 Diagrama de temporização

Para aferição da distância através do sensor HC-SR04, envia-se um sinal de *trigger* ao sensor, dando início a um ciclo de medição. O módulo, então, enviará um trem contendo 8 pulsos de ultrassom a 40kHz e alterará o *status* da porta *echo* para ligado. Após detectar a reflexão dos pulsos enviados, o sensor altera o *status* da porta *echo* para desligado, permitindo o cálculo do intervalo de tempo de ida e volta dos pulsos e, conseqüentemente, a distância entre o emissor e o objeto refletor.

O funcionamento pode ser visualizado na figura 3.

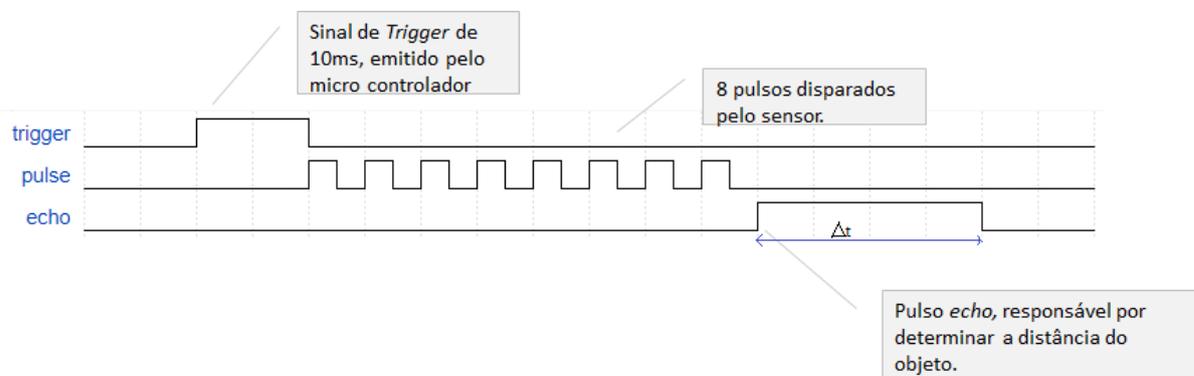


Figura 3 – Ilustrativo indicando o funcionamento do sensor.

Fonte: Adaptada de Elec Freaks (2011).



---

## DESENVOLVIMENTO DO SENSOR E CONEXÃO COM IOT

---

Para o desenvolvimento do sistema sensor embarcado foram utilizadas ferramentas envolvendo *hardware* e *software*, com o objetivo de interfacear diferentes dispositivos e permitir a correta medição e visualização dos dados.

Para tal, os seguintes itens são enumerados:

- Raspberry PI 3 B
- Sistema operacional Raspbian
- Aplicação Nuvem IoT
- Ferramenta Node-RED
- Protocolo MQTT
- Sensor ultrassônico HC-SR04

A interface entre os diversos dispositivos pode ser vista conforme o esquema na figura 4. Como pode-se ver, existem 3 níveis de dados. O nível físico onde são feitas as medições, representado pelo sensor e pelo Raspberry PI, o nível virtual, representado pela ferramenta Node-RED e pela aplicação em nuvem (*IoT*), onde os dados são processados e armazenados, e o nível físico em contato com o utilizador, onde os dados podem ser lidos e trabalhados com *input* do usuário, via *download* ou *dashboard*. Tais interfaces são alcançadas através da utilização de bibliotecas diversas de comunicação, dando atenção especial à biblioteca Eclipse Paho, responsável pela comunicação do Raspberry PI à nuvem, via MQTT.

Nas seções seguintes será desdobrada a utilização dos diversos elementos responsáveis pelo funcionamento do sistema, dando foco especial à comunicação entre o dispositivo e a nuvem.

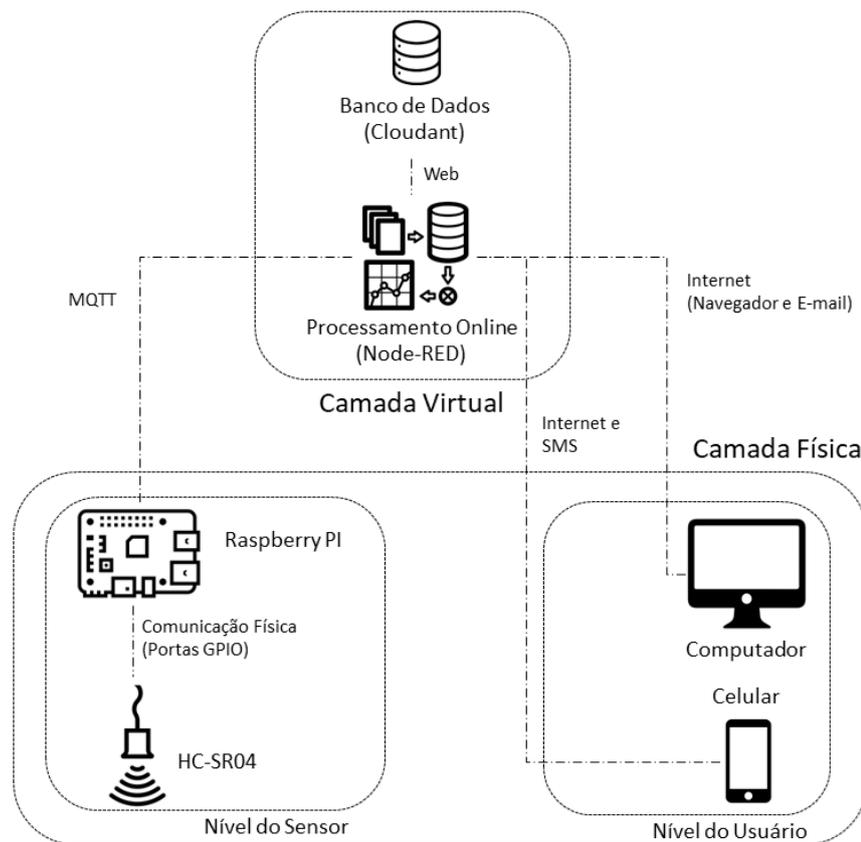


Figura 4 – Esquema indicando os níveis e interfaces entre os dispositivos, a nuvem e o usuário.

Fonte: Elaborada pelo autor.

## 3.1 Conexão do sensor HC-SR04 e Raspberry PI

A efetiva conexão do sensor HC-SR04 exige tanto uma interface elétrica quanto uma interface via *software*, sendo a interface elétrica responsável por permitir o envio e a recepção de sinais pelas portas GPIO do Raspberry PI ao sensor e o *software* responsável pelo processamento desses dados e integração com as bibliotecas de comunicação com o servidor IoT.

### 3.1.1 Interface elétrica HC-SR04 e Raspberry PI

Conforme exemplificado na seção 2.12, o sensor HC-SR04 é alimentado via tensão DC 5V, além de possuir saída 5V. Como as portas GPIO do Raspberry PI operam à tensão 3.3V, é necessário um divisor de tensão para a correta conexão do pino *echo* com a GPIO. Sendo assim, a conexão é realizada conforme o esquemático ilustrado na figura 5.

A montagem realizada pode ser vista na figura 6.

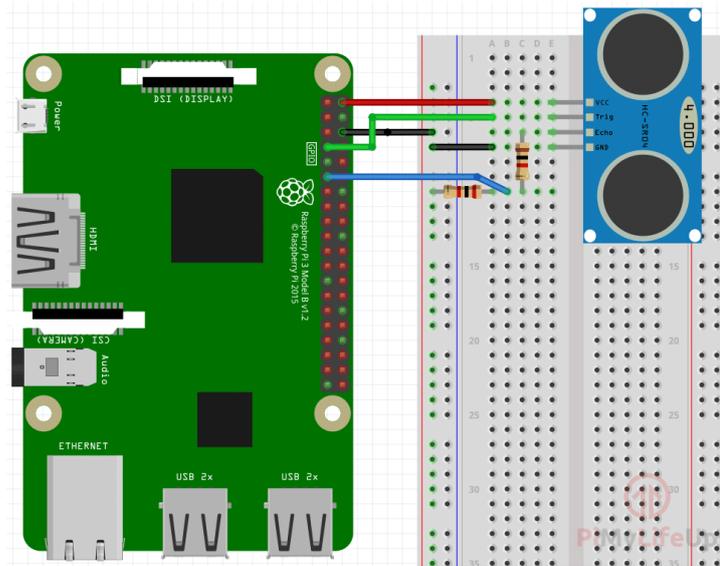


Figura 5 – Ilustrativo indicando a conexão do sensor ao Raspberry PI.

Fonte: pimylifeup (2018).

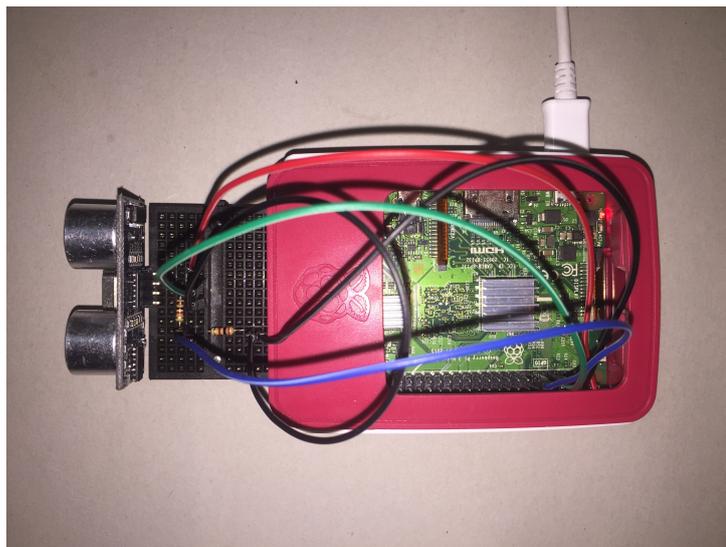


Figura 6 – Conexão realizada entre o sensor e o Raspberry PI.

Fonte: Elaborada pelo autor.

### 3.1.2 Interface Software entre o HC-SR04 e o Raspberry PI

A interface entre o sensor HC-SR04 e o Raspberry PI é realizada baseada no funcionamento do sensor, conforme elucidado na seção 2.12.

Sendo assim, é necessário configurar o Raspberry PI, de forma a emitir pulsos periódicos na porta *trigger* do sensor e então medir o tempo transcorrido entre a ativação e a desativação da porta *echo*.

Dessa forma, inicialmente, deve-se inicializar as bibliotecas responsáveis pela utilização das portas GPIO e de funções de temporização e definir quais portas serão utilizadas. Em

seguida, configuram-se as portas que atuarão como *input* ou *output*. As portas utilizadas serão as portas GPIO 7, para *trigger* e 11 para *echo*.

Foi definida então uma rotina de medição de distância, a qual retorna um valor *double* sempre que chamada, referente à distância (em centímetros) entre o objeto sensor e o objeto defletor (lâmina de água). Esta rotina é responsável pela emissão do pulso *trigger* de 10ms e posterior medição do tempo em que a porta *echo* permanece ligada, resultando, então, na distância aferida.

Ao final da execução do programa, deve-se reiniciar o status das portas GPIO para o padrão, evitando problemas na placa Raspberry PI. A execução conjunta das instruções elucidadas nas seções 2.11, 3.1 e 3.2 permitirá a efetiva medição de distâncias pelo sensor e a publicação periódica dessas distâncias ao *broker IoT*, completando então, o sensor conectado à *IoT*.

## 3.2 Configuração da interface IoT com o servidor e Dashboard

A configuração do servidor *IoT* e do *Broker* MQTT é realizada através da plataforma IBM Watson IoT Platform aliada à ferramenta Node-RED, referenciada na seção 3.2. Essa ferramenta permite tanto o recebimento dos dados enviados pelo sensor (Raspberry PI), quanto o armazenamento, o processamento e a exibição desses dados.

Para a utilização confortável e prática, foi desenvolvida uma interface amigável ao usuário, que permita a exibição dos dados em tempo real através de um *dashboard* acessível por qualquer dispositivo com acesso à internet e navegador, um sistema de emissão de alertas (E-mail), capaz de alertar o usuário quando o nível de água ultrapassar um valor pré-estabelecido, e um banco de dados, responsável pelo armazenamento dos dados, suportando armazenar medições por até um mês.

Também foi disponibilizada a possibilidade de descarregar o banco de dados para o computador, em formato .csv (Microsoft Excel), de forma a permitir uma posterior análise dos dados, por parte do usuário.

Para tal, faz-se uso dos seguintes nós, detalhados no apêndice A:

- a) *ibmiot in*;
- b) *change*;
- c) *json*;
- d) *function*;
- e) *switch*;
- f) *cloudant out*;

- g) *cloudant in*;
- h) *ui\_toast*;
- i) *e-mail*;
- j) *delay*;
- k) *join*;
- l) *json-2-csv*;
- m) *chart*;
- n) *gauge*;
- o) *smooth*.

### **3.3 Integração do sensor à IoT e comunicação com Broker MQTT**

A integração dos dispositivos, interfaces e ferramentas utilizadas permite, então, o desenvolvimento da solução proposta no texto. Nesta seção serão mostrados os passos tomados para leitura de dados de distância do sensor ao objeto refletor e sua efetiva publicação ao *broker IoT*.

Assim, propõe-se a divisão do texto a seguir conforme o enunciado: Na seção 3.3.1 será detalhado o funcionamento do programa responsável por realizar a leitura periódica dos dados de distância e fazer o *upload* para a nuvem, na seção 3.3.2, será mostrada a interface em Node-RED, responsável pelo processamento e armazenamento dos dados, e na seção 3.3.3 serão exibidas as telas de exibição de dados e interface com o usuário.

#### **3.3.1 Conexão do sensor à IoT**

A comunicação do sensor ao Raspberry PI, e concomitantemente à Internet das Coisas é realizada através de um programa escrito em linguagem *Python* e pela biblioteca Eclipse Paho MQTT. No apêndice B está demonstrado o algoritmo completo utilizado no Raspberry PI.

O algoritmo é responsável por se conectar ao *broker* e então realizar medidas periódicas da distância do objeto sensor ao objeto refletor (via portas GPIO), posteriormente fazendo *upload* destes dados à nuvem.

#### **3.3.2 Utilização da ferramenta Node-Red**

Assim como detalhado na seção 3.2, Node-RED é uma plataforma, baseada em NODE.js, através do qual se constroem algoritmos por meio de fluxos de dados.

O fluxo construído pode ser visto na figura 7 e funciona conforme o seguinte:

Os dados publicados pelo Raspberry PI são recebidos pelo fluxo hospedado em nuvem. Estes dados então são tratados, de forma a ser convertidos em números inteiros e armazenados em um banco de dados junto à uma etiqueta de tempo. Paralelo a este processo, os dados são exibidos no *dashboard* e alertas são emitidos quando alguma medição ultrapassa os limites estabelecidos, disparando os fluxos de alerta no *dashboard* e de e-mail.

Também, pode-se realizar uma solicitação de e-mail pelo *dashboard*, produzindo um arquivo .csv contendo os dados armazenados no banco de dados e enviado-o para o e-mail desejado.

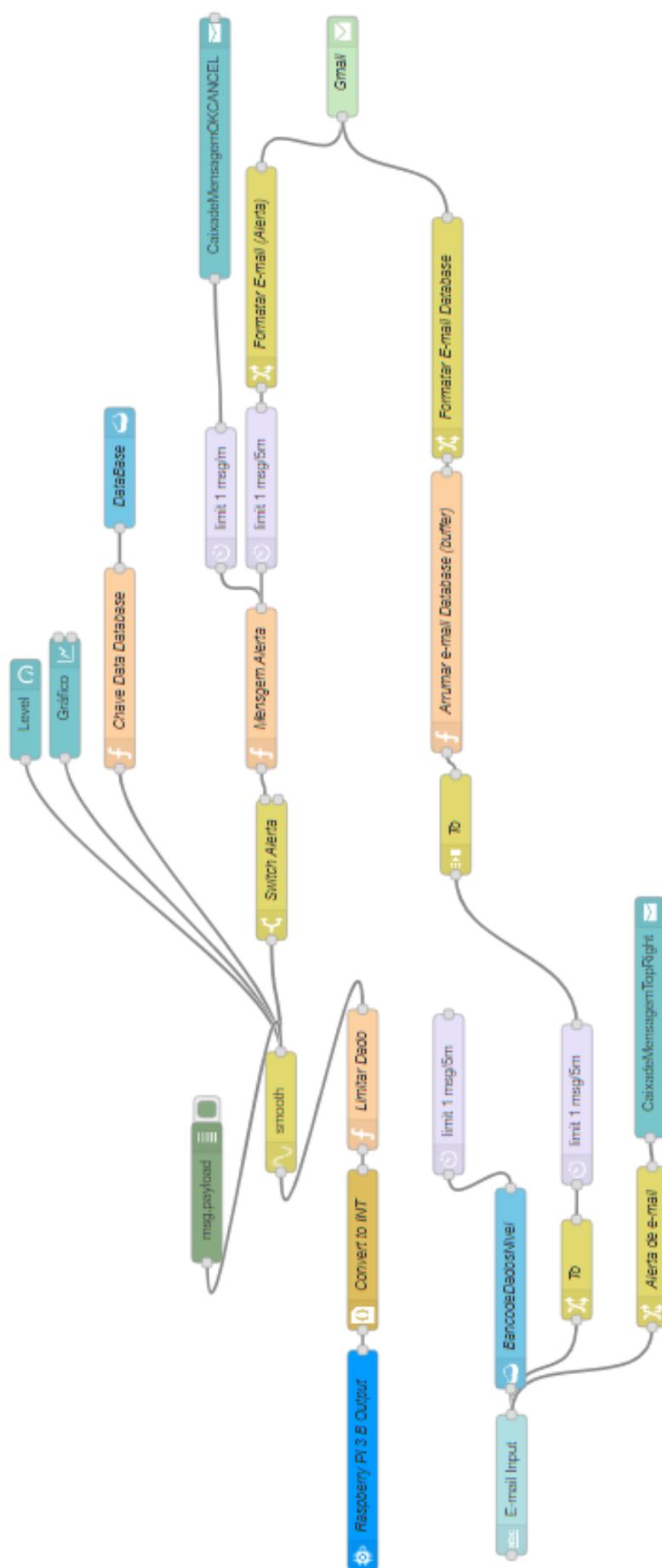


Figura 7 – Fluxo construído na plataforma Node-RED.

Fonte: Elaborada pelo autor.

### 3.3.3 Visualização de Dados

Por fim, os nós de *dashboard* (*chart* e *gauge*) adicionados ao fluxo em Node-RED permitem a visualização dos dados em um *dashboard*, como na figura 8, em tempo real, e os nó e-mail permite o envio de e-mails com alerta. Dessa forma, através da utilização do

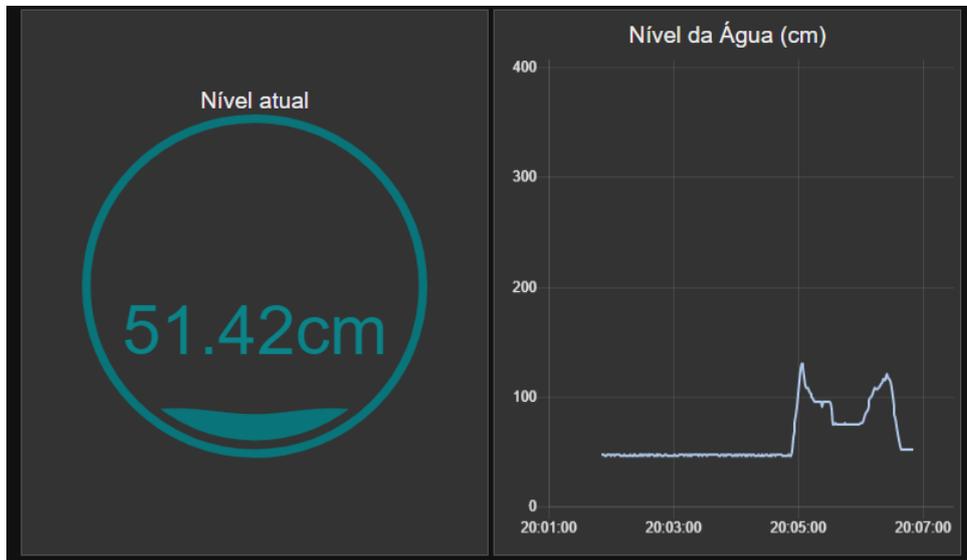


Figura 8 – Tela de exibição de dados para o usuário (*Dashboard*).

Fonte: Elaborada pelo autor.

interfaceamento dos dispositivos, algoritmos e plataformas citados, é possível a medição e armazenamento em tempo real de distância, conforme proposto.

---

## TESTES PRELIMINARES E APLICAÇÃO EM PEQUENA CENTRAL HIDRELÉTRICA

---

Tendo o sensor devidamente montado e com sua conexão à Internet das Coisas realizada, pode-se realizar medições que comprovem sua eficácia, assim como testar sua aplicação no caso proposto, uma pequena central hidrelétrica.

### 4.1 Testes de eficácia

Os primeiros testes possuem como objetivo verificar a eficácia do sensor, em um ambiente controlado, e definir se o sensor é compatível com o desempenho desejado.

Para a realização destes testes, configura-se o sensor para iniciar medições e compara-se as medições com uma régua, de forma a verificar-se se os dados gerados pelo sensor estão com acurácia desejada. Os dados foram amostrados através de uma média móvel com intervalo de 15 entradas. Essa amostragem foi utilizada de forma a suavizar as oscilações do sensor.

Para tal, em um ambiente interno, foram realizados 3 testes iguais, com 88 amostras cada, com distâncias entre 10cm e 445cm. Tais testes foram realizados para aumentar a confiança nos dados medidos e seus valores podem ser encontrados no apêndice D.

Os testes identificaram erros absolutos médios de 1,0 cm , 0,9 cm e 0,8cm. O maior erro médio sendo, portanto, 1.0 cm. O erro está, assim, dentro das especificações determinadas em (ANA, 2014).

As especificações determinadas do sensor são então, segundo a tabela 4.

Especificação	Valor
Erro Médio	1,0cm
Resolução	0,1cm
Faixa de Medição	10cm a 450cm

Fonte: Elaborada pelo autor.

Tabela 4 – Especificações determinadas do sensor através das medições realizadas no apêndice D

## 4.2 Medidas em Campo

De forma a garantir que o dispositivo é capaz de realizar medidas em campo, foi necessário instalá-lo *in loco* e comparar sua eficácia com as réguas limnimétricas instaladas na barragem. Para permitir a realização dos testes, uma rede de internet 4G foi utilizada.

Como a variação do nível da barragem é muito pequeno em apenas um dia, este teste é complementar ao teste controlado, de forma que seu objetivo é garantir que o dispositivo (com acurácia comprovada) seja capaz de operar nas condições para o qual foi designado. Os testes foram realizados em duas posições diferentes na barragem, como nas figuras 9 e 10.



Figura 9 – Testes feitos na barragem na posição 1

Fonte: Elaborada pelo autor.

Assim, instalou-se o sensor na barragem, obtendo medidas de distância (nível) de acordo com o medido nas réguas limnimétricas.



Figura 10 – Testes feitos na barragem na posição 2

Fonte: Elaborada pelo autor.

Medida Sensor (cm)	Medida Régua Limnimétrica (cm)
163,2	160
204,7	200

Fonte: Elaborada pelo autor.

Tabela 5 – Comparação entre as medidas realizadas pelo sensor e as aferidas através de régua limnimétrica

Como a escala da régua limnimétrica não permite medições com a precisão necessária para garantia do erro, este teste garante apenas o funcionamento do sensor nas condições de campo.

Dessa forma, a combinação dos testes de eficácia demonstrados na seção 4.1 junto aos testes em campo, seção 4.2 são suficientes para demonstrar a eficácia da solução proposta.



---

## CONCLUSÕES

---

O objetivo deste trabalho foi desenvolver uma solução capaz de medir o nível de água de uma barragem e disponibilizar os dados em tempo real, através da integração de soluções de Internet das Coisas e sensoriamento.

Em seu desenvolvimento foram utilizadas ferramentas de fácil acesso, como o *hardware* Raspberry PI, com baixo preço de venda, e linguagens de programação de código aberto, como Python, além da plataforma *IBM Bluemix*. Todas essas ferramentas podem ser facilmente acessadas, fazendo com que a solução desenvolvida seja de fácil replicação, quando desejado.

O desenvolvimento deste projeto permitiu aplicar os conhecimentos adquiridos ao longo da graduação de Engenharia Elétrica, em especial das matérias de Linguagens de Programação e Aplicações, Aplicação de Microprocessadores I, Aplicação de Microprocessadores II, Aproveitamentos Hidrelétricos e Redes de Computadores. Este projeto permitiu o desenvolvimento ainda maior das disciplinas aprendidas nas matérias, contribuindo para a formação do aluno.

A eficácia da solução foi comprovada através dos testes realizados, tanto em ambiente controlado quanto em condições de operação (barragem). Os erros estavam dentro da faixa especificada resolução da ANA (ANA, 2014). Assim, compreende-se que o projeto atingiu os seus objetivos de desenvolvimento de uma solução integrada de monitoramento do nível de água aplicado à uma barragem, assim como os objetivos específicos, de garantir uma interface amigável com o usuário e permitir o armazenamento dos dados em um banco de dados.

Para a realização deste trabalho foi desembolsado um valor total de 260 reais, frente a valores de 400 ou mais reais para uma solução de aplicação similar. Mostra-se então a viabilidade econômica, além de técnica, da solução proposta.

Posteriores trabalhos podem se aproveitar deste projeto, seja para aperfeiçoar ainda mais

a solução, seja para utilizar as tecnologias em soluções diferentes, como monitoramento em tempo real dos mais diferentes tipos de variáveis, como temperatura, pressão, fluxo de água, ou qualquer outra variável que possa ser medida por um sensor.

---

## CONFIGURAÇÃO DOS NÓS NA FERRAMENTA NODE-RED

---

A ferramenta Node-RED foi configurada através de "nós", cada qual executando uma função específica. A configuração do nó de conexão com o servidor (nó *ibiot*) e a descrição dos outros nós utilizados será elucidada nas seções seguintes deste apêndice.

### Configuração do nó de conexão com o servidor (nó *ibmiot*)

Para tanto, é necessário a criação, na ferramenta Node-RED de nós responsáveis por receber dados do sensor. Assim, criou-se um nó do tipo "ibmiot". Esse nó é configurado como uma aplicação, que também solicitará conexão ao servidor *IoT*, hospedado na plataforma IBM Watson IoT Platform. A configuração é realizada através da inserção dos dados de usuário, cadastrados na plataforma IBM Watson.

Na figura 11 encontram-se as telas de configuração, onde pode-se inserir os dados da aplicação, responsável por receber os dados enviados pelo *Broker*.

Os dados a ser inseridos são:

- Name : Um nome dado ao cliente;
- API Key : Uma chave gerada ao cadastrar a aplicação no servidor;
- API Token : Um token único gerado no momento de cadastro no servidor;
- Server Name : Endereço do servidor IoT.

Já os itens *Device Type*, *Device ID*, *Event* e *Format* são opcionais e permitem a filtragem dos dados que se deseja receber.

Figura 11 – Tela de inserção de dados para conexão ao Broker MQTT.

Fonte: Elaborada pelo autor.

## Configuração do processamento, armazenamento e exibição de dados

Os nós de processamento, armazenamento e exibição de dados serão aqui detalhados.

### *change*

Nó responsável por alterar alguma propriedade do objeto passado, podendo alterar, por exemplo, a mensagem contida, o tópico do objeto, ou qualquer outra propriedade desejada.

### *json*

Nó utilizado para alterar o tipo de dado, convertendo um dado JSON string para um objeto Javascript.

### *function*

Permite executar qualquer função em Javascript, dessa forma, dando liberdade ao usuário escolher o que fazer com o dado.

### *switch*

Roteia a mensagem de entrada entre duas ou mais saídas, de acordo com o conteúdo da mensagem.

***cloudant out***

Armazena o dado de entrada em um repositório de dados hospedado no servidor cloudant.

***cloudant in***

Permite a leitura de um repositório de dados, hospedado no servidor Cloudant.

***ui\_toast***

Responsável pela exibição de alertas no *dashboard*.

***e-mail***

Nó, cuja função é o envio de e-mails, podendo conter um anexo em formato binário ou uma mensagem de texto.

***delay***

Atrasa uma sequência de dados por um determinado tempo, podendo apagar ou não os dados recebidos durante o intervalo de espera.

***join***

Responsável por unir dois objetos em um só, podendo combinar propriedades dos objetos.

***json-2-csv***

Converte um objeto JSON em um objeto binário configurado como CSV. utilizado para gerar o arquivo para *download* e utilização em programas de planilha.

***chart***

Indica, em um gráfico de linha, com eixo X de tempo, o dado recebido na entrada.

***gauge***

Indica o valor do último dado recebido na entrada.

***smooth***

Calcula a média móvel dos dados inseridos.



---

## ALGORITMO UTILIZADO NO RASPBERRY PI

---

---

O algoritmo completo utilizado no Raspberry PI para medição e *upload* dos dados de distância ao *broker* está aqui transcrito:

---

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3
4 import RPi.GPIO as GPIO
5 import time
6 import paho.mqtt.client as mqtt
7 import sys
8
9 #definicoes:
10 Broker = "89khha.messaging.internetofthings.ibmcloud.com"
11 PortaBroker = 1883
12 KeepAliveBroker = 60
13 TopicoSubscribe = "iot-2/cmd/status/fmt/text" #dica: troque
    o nome do topico por algo "unico",
14                                     #Dessa maneira, ninguem
    ira saber seu topico de
15                                     #subscribe e interferir
    em seus testes
16 GPIO.setmode(GPIO.BOARD)
17
18
19 PIN_TRIGGER = 7
```

```
20 PIN_ECHO = 11
21
22 GPIO.setup(PIN_TRIGGER, GPIO.OUT)
23 GPIO.setup(PIN_ECHO, GPIO.IN)
24
25 GPIO.output(PIN_TRIGGER, GPIO.LOW)
26
27 print "[STATUS] Solicitando conexão ao broker: " + Broker
28
29 time.sleep(2)
30
31 #Callback – conexão ao broker realizada
32 def on_connect(client, userdata, flags, rc):
33     if rc == 0 :
34         print("[STATUS] Conexão concluída com sucesso!")
35     if rc != 0 :
36         print("[STATUS] Falha ao conectar ao Broker.
37             Resultado de conexão: "+str(rc))
38
39     #faz subscribe automatico no topico
40     client.subscribe(TopicoSubscribe)
41
42 #Callback – mensagem recebida do broker
43 def on_message(client, userdata, msg):
44     MensagemRecebida = str(msg.payload)
45     print("Distance "+MensagemRecebida)
46
47 #Callback – escrever mensagem quando publicar
48 def on_publish(client, userdata, mid):
49     print("mid: "+str(mid))
50
51
52 def distance():
53     GPIO.output(PIN_TRIGGER, GPIO.HIGH)
54     time.sleep(0.00001)
55     GPIO.output(PIN_TRIGGER, GPIO.LOW)
56
57     while GPIO.input(PIN_ECHO)==0:
```

---

```
58     pulse_start_time = time.time()
59     while GPIO.input(PIN_ECHO)==1:
60         pulse_end_time = time.time()
61
62         pulse_duration = pulse_end_time - pulse_start_time
63
64         distance = round(pulse_duration * 17150, 2)
65         return distance
66
67 #programa principal:
68 try:
69     print("[STATUS] Inicializando MQTT...")
70     #inicializa MQTT:
71     client = mqtt.Client("d:89khha:Placa:RaspberryPI")
72     #client ID
73     client.username_pw_set("use-token-auth", "98264500")
74     client.on_connect = on_connect
75     client.on_message = on_message
76     #client.on_publish = on_publish
77
78     client.connect(Broker, PortaBroker, KeepAliveBroker
79 )
80     client.loop_start()
81     print "[STATUS] Iniciando leitura de dados..."
82     while True:
83         dist = distance()
84         client.publish("iot-2/evt/status/fmt/text", dist, qos
85 =1)
86         time.sleep(1)
87 except KeyboardInterrupt:
88     print "[STATUS] Ctrl+C pressionado, encerrando
89 aplicacao e saindo..."
90     GPIO.cleanup()
91     sys.exit(0)
```

---



---

# UTILIZAÇÃO DA BIBLIOTECA PAHO-MQTT

---

A utilização da biblioteca Paho MQTT é descrita nas seções seguintes.

## Inicialização da biblioteca Paho e construção de uma instância *client*

De forma a permitir a utilização da biblioteca Paho no Raspberry PI é necessária sua utilização e a construção de uma instância *client*, sob a qual serão chamados os métodos de conexão com o servidor, envio e recebimento de mensagens.

---

```
1 import paho.mqtt.client as mqtt
2
3 client = mqtt.client(client_id="", clean_session=True,
    userdata=None, protocol=MQTTv311, transport="tcp")
```

---

onde:

- `client_id` : Uma *string* única identificando o cliente ao servidor;
- `clean_session` : Um dado lógico que determina o tipo de cliente. Se verdadeiro, o *broker* removerá toda informação sobre este cliente quando ele se desconecta. Se falso, o cliente é um cliente durável e as informações de subscrição e mensagens em fila serão retidas;
- `userdata` : Dados de qualquer tipo do usuário, passados como parâmetros para *callbacks*;
- `protocol` : Versão do protocolo MQTT utilizada, podendo ser tanto MQTTv31 ou MQTTv311;

- `transport` : possui duas opções, "tcp", configurando a transmissão de dados como TCP pura, ou "websocket", para utilizar a tecnologia WebSocket na transmissão de dados.

## Definição dos dados de usuário e conexão ao servidor

Para se conectar ao servidor MQTT utilizado é necessária a definição dos dados de usuário a serem fornecidas no momento da conexão, além da chamada da função de conexão propriamente dita e do início da *thread* que permite o recebimento contínuo de informações do servidor.

---

```
1 client.username_pw_set(username , password=None)
2 client.connect(host , port=1883, keepalive=60, bind_address=
  "")
3 client.loop_start()
```

---

Onde tem-se:

- `username` : nome de usuário fornecido ao servidor;
- `password` : senha de usuário fornecida ao servidor;
- `host` : endereço do *broker* ao qual se tenta a conexão;
- `port` : A porta de rede do servidor ao qual se tenta conectar. O padrão é 1883;
- `keepalive` : máximo período de tempo em segundos permitido entre conexões com o *broker*. Se nenhuma outra mensagem é trocada, este parâmetro controla a taxa com que o cliente envia mensagens de *ping* ao *broker*;
- `bind_adress` : O endereço IP de uma interface de rede local para a qual associar este cliente, assumindo que existam várias interfaces.

Após estas rotinas, o cliente estará devidamente conectado ao *broker*, podendo então executar suas funções de postagem e recebimento de dados.

## Publicando dados ao *broker*

Após a conexão realizada, é possível publicar dados ao *broker* através da função *publish*, responsável por enviar uma mensagem do cliente ao *broker*.

---

```
1 client.publish(topic , payload=None , qos=0 , retain=False)
```

---

Onde:

- `topic` : tópico para qual se deseja publicar a mensagem;

- `payload` : Conteúdo da mensagem a ser publicada;
- `qos` : nível de qualidade de serviço a ser utilizado;
- `retain` : se verdadeiro, a mensagem será configurada como última mensagem retida para este tópico.

## Assinando um tópico

Para ser capaz de receber mensagens, o cliente deve primeiro conectar-se a um tópico do *broker*, e então, a partir deste momento, passa a receber todas as mensagens enviadas a este tópico.

Este procedimento é realizado através da função *subscribe*, assim definida:

---

```
1 client.subscribe(topic, qos=0)
```

---

Onde:

- `topic` : tópico desejado;
- `qos` : nível de qualidade de serviço a ser utilizado.

## Funções *Callback*

Além das funções de conexão, subscrição e publicação, existem também as funções *callbacks*, chamadas quando um dos procedimentos acima citado são realizados. Assim, as funções *on\_connect()*, *on\_publish()*, *on\_message()*, *on\_subscribe()* e *on\_disconnect()* são chamadas ao final da conexão, publicação de uma mensagem, recebimento de uma mensagem, subscrição de um tópico e desconexão.

Assim definidas:

---

```
1 def on_connect(client, userdata, flags, rc):
2     #Descrição da função on_connect
3     mqttc.on_connect = on_connect
4
5 def on_message(client, userdata, message):
6     #Descrição da função on_message
7     mqtt.on_message = on_message
8
9 def on_publish(client, userdata, mid):
10    #Descrição da função on_publish
11    mqtt.on_publish = on_publish
12
```

```
13 def on_subscribe(client , userdata , mid , granted_qos ) :
14     #Descrição da função on_subscribe
15 mqtt.on_subscribe = on_subscribe
16
17 def on_disconnect(client , userdata , rc ) :
18     #Descrição da função on_disconnect
19 mqttc.on_disconnect = on_disconnect
```

---

Onde tem-se:

- *client* : a instância de cliente para o *callback*;
- *userdata* : os dados privados de usuários definidos na função *connect*;
- *flags* : *flags* de resposta enviadas pelo *broker*;
- *rc* : o resultado da conexão (ou desconexão);
- *message* : Uma instância da mensagem MQTT recebida. É uma classe com os membros: *topic*, *payload*, *qos* e *retain*;
- *mid* : variável utilizada para rastrear a mensagem enviada;
- *granted\_qos* : Lista de inteiros indicando o nível de qualidade de serviço oferecido pelo servidor para cada subscrição realizada.

## TABELA DE RESULTADOS - TESTE DE EFICÁCIA DO SENSOR

Este apêndice exhibe as medições realizadas para aferir a precisão do sensor.

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
1	11,6	10,0	-1,6	1,6
2	15,8	15,0	-0,8	0,8
3	21,3	20,0	-1,3	1,3
4	26,3	25,0	-1,3	1,3
5	30,8	30,0	-0,8	0,8
6	36,1	35,0	-1,1	1,1
7	41,1	40,0	-1,1	1,1
8	45,8	45,0	-0,8	0,8
9	50,5	50,0	-0,5	0,5
10	55,7	55,0	-0,7	0,7
11	60,4	60,0	-0,4	0,4
12	65,3	65,0	-0,3	0,3
13	69,9	70,0	0,1	0,1
14	74,9	75,0	0,1	0,1
15	80,1	80,0	-0,1	0,1
16	85,5	85,0	-0,5	0,5
17	89,5	90,0	0,5	0,5
18	94,7	95,0	0,3	0,3
19	99,5	100,0	0,5	0,5

...continua

Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1).

... continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
20	104,5	105,0	0,5	0,5
21	109,5	110,0	0,5	0,5
22	114,8	115,0	0,2	0,2
23	119,1	120,0	0,9	0,9
24	123,9	125,0	1,1	1,1
25	128,6	130,0	1,4	1,4
26	133,6	135,0	1,4	1,4
27	139,4	140,0	0,6	0,6
28	144,0	145,0	1,0	1,0
29	148,9	150,0	1,1	1,1
30	152,7	155,0	2,3	2,3
31	157,6	160,0	2,4	2,4
32	163,1	165,0	1,9	1,9
33	168,0	170,0	2,0	2,0
34	173,1	175,0	1,9	1,9
35	177,8	180,0	2,2	2,2
36	183,1	185,0	1,9	1,9
37	189,1	190,0	0,9	0,9
38	194,5	195,0	0,5	0,5
39	198,7	200,0	1,3	1,3
40	203,0	205,0	2,0	2,0
41	208,3	210,0	1,7	1,7
42	214,3	215,0	0,7	0,7
43	219,8	220,0	0,2	0,2
44	225,2	225,0	-0,2	0,2
45	230,5	230,0	-0,5	0,5
46	236,5	235,0	-1,5	1,5
47	240,7	240,0	-0,7	0,7
48	245,6	245,0	-0,6	0,6
49	250,2	250,0	-0,2	0,2
50	255,2	255,0	-0,2	0,2
51	260,1	260,0	-0,1	0,1
52	265,1	265,0	-0,1	0,1
53	270,5	270,0	-0,5	0,5

... continua

Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1).

...continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
54	278,0	275,0	-3,0	3,0
55	282,2	280,0	-2,2	2,2
56	290,0	285,0	-5,0	5,0
57	293,8	290,0	-3,8	3,8
58	297,0	295,0	-2,0	2,0
59	302,5	300,0	-2,5	2,5
60	306,4	305,0	-1,4	1,4
61	311,4	310,0	-1,4	1,4
62	314,7	315,0	0,3	0,3
63	319,9	320,0	0,1	0,1
64	324,4	325,0	0,6	0,6
65	331,0	330,0	-1,0	1,0
66	334,6	335,0	0,4	0,4
67	339,6	340,0	0,4	0,4
68	345,2	345,0	-0,2	0,2
69	351,1	350,0	-1,1	1,1
70	355,1	355,0	-0,1	0,1
71	360,3	360,0	-0,3	0,3
72	367,1	365,0	-2,1	2,1
73	368,4	370,0	1,6	1,6
74	374,0	375,0	1,0	1,0
75	379,6	380,0	0,4	0,4
76	383,5	385,0	1,5	1,5
77	389,9	390,0	0,1	0,1
78	396,5	395,0	-1,5	1,5
79	398,9	400,0	1,1	1,1
80	405,0	405,0	0,0	0,0
81	409,1	410,0	0,9	0,9
82	416,0	415,0	-1,0	1,0
83	421,8	420,0	-1,8	1,8
84	424,1	425,0	0,9	0,9
85	429,1	430,0	0,9	0,9
86	435,6	435,0	-0,6	0,6
87	439,4	440,0	0,6	0,6

...continua

Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1).

...continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
88	444,7	445,0	0,3	0,3

Tabela 6 – Testes feitos na para aferir a precisão do sensor (Medidas 1).

Fonte: Elaborada pelo autor.

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
1	11,0	10,0	-1,0	1,0
2	15,8	15,0	-0,8	0,8
3	21,0	20,0	-1,0	1,0
4	25,6	25,0	-0,6	0,6
5	29,6	30,0	0,4	0,4
6	35,1	35,0	-0,1	0,1
7	40,6	40,0	-0,6	0,6
8	46,3	45,0	-1,3	1,3
9	49,9	50,0	0,1	0,1
10	54,8	55,0	0,3	0,3
11	59,8	60,0	0,2	0,2
12	64,9	65,0	0,1	0,1
13	69,6	70,0	0,4	0,4
14	74,9	75,0	0,1	0,1
15	80,4	80,0	-0,4	0,4
16	84,0	85,0	1,0	1,0
17	89,0	90,0	1,0	1,0
18	93,8	95,0	1,2	1,2
19	99,0	100,0	1,1	1,1
20	104,1	105,0	0,9	0,9
21	109,1	110,0	0,9	0,9
22	114,2	115,0	0,8	0,8
23	119,4	120,0	0,6	0,6
24	125,0	125,0	0,0	0,0
25	129,8	130,0	0,2	0,2
26	134,1	135,0	0,9	0,9
27	138,6	140,0	1,4	1,4
28	143,6	145,0	1,4	1,4
29	149,1	150,0	0,9	0,9
30	155,3	155,0	-0,3	0,3
31	160,2	160,0	-0,2	0,2
32	165,2	165,0	-0,2	0,2
33	169,4	170,0	0,6	0,6
34	174,6	175,0	0,4	0,4
35	180,1	180,0	-0,1	0,1

... continua

Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2).

... continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
36	185,5	185,0	-0,4	0,4
37	190,8	190,0	-0,8	0,8
38	195,6	195,0	-0,6	0,6
39	199,6	200,0	0,4	0,4
40	205,6	205,0	-0,6	0,6
41	210,5	210,0	-0,5	0,5
42	215,3	215,0	-0,3	0,3
43	220,7	220,0	-0,7	0,7
44	225,6	225,0	-0,6	0,6
45	230,2	230,0	-0,2	0,2
46	235,8	235,0	-0,8	0,8
47	241,1	240,0	-1,1	1,1
48	245,7	245,0	-0,7	0,7
49	249,0	250,0	1,0	1,0
50	255,7	255,0	-0,7	0,7
51	261,3	260,0	-1,3	1,3
52	266,1	265,0	-1,1	1,1
53	270,4	270,0	-0,4	0,4
54	275,4	275,0	-0,4	0,4
55	280,8	280,0	-0,8	0,8
56	286,4	285,0	-1,4	1,4
57	290,7	290,0	-0,7	0,7
58	295,9	295,0	-0,9	0,9
59	301,3	300,0	-1,3	1,3
60	305,8	305,0	-0,8	0,8
61	309,9	310,0	0,1	0,1
62	315,1	315,0	-0,1	0,1
63	319,8	320,0	0,3	0,3
64	325,7	325,0	-0,7	0,7
65	330,7	330,0	-0,7	0,7
66	335,9	335,0	-0,9	0,9
67	340,9	340,0	-0,9	0,9
68	345,9	345,0	-0,9	0,9
69	350,7	350,0	-0,7	0,7

... continua

Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2).

...continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
70	357,0	355,0	-2,0	2,0
71	360,2	360,0	-0,2	0,2
72	365,3	365,0	-0,3	0,3
73	370,3	370,0	-0,3	0,3
74	375,8	375,0	-0,8	0,8
75	380,8	380,0	-0,8	0,8
76	386,8	385,0	-1,8	1,8
77	391,9	390,0	-1,9	1,9
78	396,9	395,0	-1,9	1,9
79	403,0	400,0	-3,0	3,0
80	406,6	405,0	-1,6	1,6
81	412,7	410,0	-2,6	2,6
82	417,3	415,0	-2,3	2,3
83	421,9	420,0	-1,9	1,9
84	428,6	425,0	-3,6	3,6
85	432,7	430,0	-2,7	2,7
86	438,1	435,0	-3,1	3,1
87	443,8	440,0	-3,8	3,8
88	447,2	445,0	-2,2	2,2

Tabela 7 – Testes feitos na para aferir a precisão do sensor (Medidas 2).

Fonte: Elaborada pelo autor.

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
1	10,3	10,0	-0,3	0,3
2	15,1	15,0	-0,1	0,1
3	19,9	20,0	0,1	0,1
4	24,9	25,0	0,1	0,1
5	29,0	30,0	1,0	1,0
6	33,3	35,0	1,7	1,7
7	39,8	40,0	0,2	0,2
8	44,8	45,0	0,3	0,3
9	49,0	50,0	1,0	1,0
10	55,1	55,0	-0,1	0,1
11	59,2	60,0	0,8	0,8
12	65,2	65,0	-0,2	0,2
13	70,1	70,0	-0,1	0,1
14	73,9	75,0	1,1	1,1
15	79,1	80,0	0,9	0,9
16	83,1	85,0	1,9	1,9
17	88,5	90,0	1,5	1,5
18	93,4	95,0	1,6	1,6
19	98,6	100,0	1,4	1,4
20	103,7	105,0	1,3	1,3
21	109,1	110,0	1,0	1,0
22	113,9	115,0	1,1	1,1
23	118,8	120,0	1,2	1,2
24	123,8	125,0	1,2	1,2
25	129,3	130,0	0,8	0,8
26	133,8	135,0	1,2	1,2
27	137,9	140,0	2,2	2,2
28	143,1	145,0	1,9	1,9
29	148,1	150,0	1,9	1,9
30	155,2	155,0	-0,2	0,2
31	160,1	160,0	-0,1	0,1
32	165,0	165,0	0,0	0,0
33	170,3	170,0	-0,3	0,3
34	174,6	175,0	0,4	0,4
35	180,2	180,0	-0,2	0,2

... continua

Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3).

...continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
36	185,3	185,0	-0,3	0,3
37	190,1	190,0	-0,1	0,1
38	195,3	195,0	-0,3	0,3
39	200,4	200,0	-0,4	0,4
40	205,4	205,0	-0,3	0,3
41	210,2	210,0	-0,2	0,2
42	215,6	215,0	-0,6	0,6
43	220,2	220,0	-0,2	0,2
44	224,0	225,0	1,0	1,0
45	230,6	230,0	-0,6	0,6
46	235,9	235,0	-0,9	0,9
47	240,5	240,0	-0,5	0,5
48	245,1	245,0	-0,1	0,1
49	250,5	250,0	-0,5	0,5
50	255,9	255,0	-0,9	0,9
51	260,8	260,0	-0,8	0,8
52	265,5	265,0	-0,4	0,4
53	271,0	270,0	-1,0	1,0
54	275,3	275,0	-0,3	0,3
55	280,1	280,0	-0,1	0,1
56	286,4	285,0	-1,4	1,4
57	291,6	290,0	-1,6	1,6
58	296,0	295,0	-1,0	1,0
59	301,1	300,0	-1,1	1,1
60	306,2	305,0	-1,2	1,2
61	310,0	310,0	0,0	0,0
62	314,6	315,0	0,4	0,4
63	320,0	320,0	0,1	0,1
64	325,4	325,0	-0,4	0,4
65	331,2	330,0	-1,2	1,2
66	335,7	335,0	-0,6	0,6
67	341,3	340,0	-1,3	1,3
68	345,2	345,0	-0,2	0,2
69	350,7	350,0	-0,7	0,7

...continua

Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3).

... continuação

Número	Medido no Sensor (cm)	Medido na Régua (cm)	Erro (cm)	Erro Absoluto (cm)
70	355,9	355,0	-0,9	0,9
71	360,2	360,0	-0,2	0,2
72	365,7	365,0	-0,6	0,6
73	369,4	370,0	0,6	0,6
74	374,7	375,0	0,3	0,3
75	380,6	380,0	-0,6	0,6
76	385,6	385,0	-0,6	0,6
77	391,4	390,0	-1,4	1,4
78	395,5	395,0	-0,5	0,5
79	401,2	400,0	-1,1	1,1
80	405,1	405,0	-0,1	0,1
81	410,9	410,0	-0,9	0,9
82	416,1	415,0	-1,1	1,1
83	420,7	420,0	-0,7	0,7
84	428,3	425,0	-3,3	3,3
85	432,4	430,0	-2,4	2,4
86	436,8	435,0	-1,8	1,8
87	441,3	440,0	-1,3	1,3
88	445,8	445,0	-0,8	0,8

Tabela 8 – Testes feitos na para aferir a precisão do sensor (Medidas 3).

Fonte: Elaborada pelo autor.

## REFERÊNCIAS

---



---

ANA. **Orientações para elaboração do relatório de instalação de estações hidrométricas**. 2014.

\_\_\_\_\_. **Site Oficial da Agência Nacional de Águas**. Entre 2010 e 2018. <<http://www3.ana.gov.br>>. Acesso: 2018-08-07.

ANEEL. **Atlas de energia elétrica do Brasil**. [S.l.]: ANEEL, 2008.

\_\_\_\_\_. **RESOLUÇÃO NORMATIVA Nº 673, DE 4 DE AGOSTO DE 2015**. 2015.

\_\_\_\_\_. **ANEEL Banco de Informações de Geração**. 2018. <<http://www2.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>>. Acesso: 2018-08-07.

\_\_\_\_\_. **Site Oficial da ANEEL**. Entre 2010 e 2018. <<http://www.aneel.gov.br>>. Acesso: 2018-08-07.

ANEEL; ANA. **Resolução Conjunta ANEEL/ANA nº03 de 10 de Agosto de 2010**. 2010.

BRASIL. **LEI Nº 9.427, DE 26 DE DEZEMBRO DE 1996**. 1996.

\_\_\_\_\_. **DECRETO Nº 2.335, DE 6 DE OUTUBRO DE 1997**. 1997.

\_\_\_\_\_. **LEI Nº 9.433, DE 8 DE JANEIRO DE 1997**. 1997.

\_\_\_\_\_. **LEI Nº 9.984, DE 17 DE JULHO DE 2000**. 2000.

(BRASIL), M. de Minas e E. **Plano Nacional de Energia 2030**. [S.l.]: MME, 2007.

DHANDRE, N. M.; KAMALASEKARAN, P. D.; PANDEY, P. Dam parameters monitoring system. In: **2016 7th India International Conference on Power Electronics (IICPE)**. [S.l.: s.n.], 2016. p. 1–5.

ECLIPSE-PAHO. **Site Oficial Eclipse Paho**. 2015. <<https://www.eclipse.org/paho/>>. Acesso: 2018-07-09.

ELEC FREACKS. **Ultrasonic Ranging Module HC - SR04**. [S.l.], 2011.

ELETROBRÁS. **Diretrizes para Estudos e Projetos de Pequenas Centrais Hidrelétricas**. [S.l.], 2000.

IBM. **IBM Cloud Web-Site**. 2017. <<https://www.ibm.com/developerworks/cloud/library/cl-cloud-technology-basics/index.html>>. Acesso: 2018-08-08.

\_\_\_\_\_. **Site Oficial IBM**. Entre 2012 e 2018. <<https://www.ibm.com>>. Acesso: 2018-06-23.

JAMBORSALAMATI, P.; FERNANDEZ, E.; HOSSAIN, M. J.; RAFI, F. H. M. Design and implementation of a cloud-based iot platform for data acquisition and device supply management in smart buildings. In: **2017 Australasian Universities Power Engineering Conference (AUPEC)**. [S.l.: s.n.], 2017. p. 1–6.

- LEKIĆ, M.; GARDAŠEVIĆ, G. Iot sensor integration to node-red platform. In: **2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)**. [S.l.: s.n.], 2018. p. 1–5.
- LI, Y.; WANG, Y. Design and implementation of reservoir dam safety monitoring platform based on asp.net. In: **2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)**. [S.l.: s.n.], 2017. p. 2644–2648.
- MATOS, A. J. S. **Melhorias qualitativas na modelagem de levantamentos batimétricos em reservatórios por meio da ferramenta computacional "CAVN-NH"**. Tese (Doutorado) — Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, 2012. (Doutorado em Ciências da Engenharia Ambiental).
- MAUAD, F. F.; FERREIRA, L. d. C.; TRINDADE, T. C. G. **Energia Renovável no Brasil - Análise das principais fontes energéticas renováveis brasileiras**. [S.l.]: EESC/USP, 2017.
- NODE-RED. **Guia oficial Node-RED**. 2017. <<http://www.nodered.org>>. Acesso: 2018-06-25.
- PIMYLIFEUP. **Raspberry Pi Distance Sensor: How to setup the HC-SR04**. 2018. <<https://pimylifeup.com/raspberry-pi-distance-sensor/>>. Acesso: 2018-07-23.
- RASPBIAN. **Site Oficial RASPBIAN**. entre 2012 e 2017. <<http://www.raspbian.org/>>. Acesso: 2018-07-09.
- SANTOS, J. Brito dos; FRANCA, M.; ALMEIDA, A. Betâmio de. Risco associado à ruptura de barragens por galgamento. **Ingenieria del Agua**, Fundación para el Fomento de la Ingenieria del Agua, v. 14, 12 2007.
- SHAH, S. H.; YAQOOB, I. A survey: Internet of things (iot) technologies, applications and challenges. In: **2016 IEEE Smart Energy Grid Engineering (SEGE)**. [S.l.: s.n.], 2016. p. 381–385.
- TUNDISI, J. G.; MATSUMURA-TUNDISI, T.; JUNIOR, J. D. A.; TUNDISI, J. E. M.; MANZINI, N. F.; DUCROT, R. The response of Carlos Botelho (Lobo, Broa) reservoir to the passage of cold fronts as reflected by physical, chemical, and biological variables. **Brazilian Journal of Biology**, scielo, v. 64, p. 177 – 186, 02 2004. ISSN 1519-6984. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1519-69842004000100020&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1519-69842004000100020&nrm=iso)>.
- UPTON, E.; HALFACREE, G. **Raspberry Pi User Guide**. Wiley, 2016. Disponível em: <<https://www.amazon.com/Raspberry-User-Guide-Eben-Upton-ebook/dp/B01K5WSIPQ?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B01K5WSIPQ>>.
- VELEZ, J.; TRAFFORD, R.; PIERCE, M.; THOMSON, B.; JASTRZEBSKI, E.; LAU, B. Ieee 1451-1-6: Providing common network services over mqtt. In: **2018 IEEE Sensors Applications Symposium (SAS)**. [S.l.: s.n.], 2018. p. 1–6.
- VERMESAN, O.; FRIESS, P. **Internet of things - from research and innovation to market deployment**. [S.l.]: River Publishers, 2015.