MEC-SETEC

INSTITUTO FEDERAL MINAS GERAIS – Campus Formiga

Curso de Engenharia Elétrica

# SISTEMA DE MONITORAMENTO DE RESERVATÓRIOS DE ÁGUA COM ABORDAGEM E CONCEITO DE INTERNET DAS COISAS

Rafael Alves Costa

Orientador: Prof. Dr. Carlos Renato Borges dos Santos.

Co-orientadora: Prof. Dra. Ana Flávia Peixoto de Camargos.

FORMIGA-MG.

2019

Rafael Alves Costa

# SISTEMA DE MONITORAMENTO DE RESERVATÓRIOS DE ÁGUA COM ABORDAGEM E CONCEITO DE INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Campus Formiga, como requisito parcial para a obtenção do título de Bacharel em Engenharia elétrica.

Orientador: Prof. Dr. Carlos Renato Borges dos Santos.

Co-orientadora: Prof. Dra. Ana Flávia Peixoto de Camargos.

FORMIGA-MG.

2019

Costa, Rafael Alves.
621.3 Sistema de monitoramento de reservatórios de água com abordagem e conceito de internet das coisas / Rafael Alves Costa. -- Formiga : IFMG, 2019.
50p. : il.

Orientador: Prof. Dr. Carlos Renato Borges dos Santos
Co-Orientador: Prof. Dra. Ana Flávia Peixoto de Camargos.
Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Internet das coisas. 2.Microcontrolador. 3. Sensor de nível. 4. Back-end. 5. Front-end.. I. Título.

CDD 621.3

*Ficha catalográfica elaborada pela Bibliotecária Msc. Naliana Dias Leandro CRB6-1347*

**RAFAEL ALVES COSTA**

# SISTEMA DE MONITORAMENTO DE RESERVATÓRIOS DE ÁGUA COM ABORDAGEM E CONCEITO DE INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais como requisito para obtenção do Título de Bacharel em Engenharia Elétrica.

Avaliado em: 12 de junho de 2014.

Nota: 94

**BANCA EXAMINADORA**

Prof. Carlos Renato Borges dos Santos (Orientador)

Prof. Everthon Valadão dos Santos

Prof. Renan Souza Moura

# AGRADECIMENTOS

Inicialmente agradeço a Deus por ter tornado possível a minha chegada até aqui. Agradeço ao meu orientador, Prof. Dr. Carlos Renato Borges Santos e co-orientadora, Prof. Dra. Ana Flávia Peixoto de Camargos, pela grande contribuição no desenvolvimento deste trabalho.

Agradeço também a toda minha família e amigos principalmente ao meu pai Mauricio Alves Costa e minha mãe Marcia Conceição Costa, que fizeram tudo o que era possível para me oferecer condições para a confecção deste trabalho.

# RESUMO

O presente trabalho tem como objetivo desenvolver um sistema de monitoramento remoto em tempo real de um reservatório de água residencial juntamente com um circuito de acionamento de uma bomba d'água para o reabastecimento, tendo como principal objetivo fornecer ao morador residencial uma forma de conscientizar-se sobre a falta de fornecimento de água. Para o desenvolvimento do projeto utilizou-se o microcontrolador ESP32 juntamente com um sensor de nível automotivo, viabilizando a aquisição do nível do reservatório e a conexão à internet por meio de uma rede Wi-Fi. As medições de níveis são enviadas para um servidor *web* hospedado remotamente, oferecendo ao usuário acesso a uma interface de monitoramento e controle do sistema por meio de qualquer dispositivo conectado à internet.

**Palavras-chaves:** Internet das coisas, microcontrolador, *web*, reabastecimento, sensor de nível, *back-end*, *front-end*.

# ABSTRACT

In this paper aims to present the development of a real time remote monitoring system of a residential water reservoir together with a water pump drive circuit for refueling, having as main objective to provide to the residential dweller a way to become aware of the lack of water supply. For the development of the project was used the microcontroller ESP32 together with an automotive level sensor, making it possible to acquire the level of the reservoir and the connection to the internet through a Wi-Fi network. The level measurements are sent to a remotely hosted web server, giving the user access to a system monitoring and control interface through any device connected to the internet.



**Keywords:** Internet of things, microcontroller, web, resource, level sensor, back-end, front-end.

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# SUMÁRIO

# 1 INTRODUÇÃO

Em meados de 1967, uma concepção de tecnologia de intercomunicação entre dispositivos providos de processamento de dados foi implementada, contemplando uma forma de comunicação descentralizada que utiliza a arquitetura TCP/IP, concebida pela agência de projetos de pesquisa avançados dos Estudos Unidos (ARPA), foi denominada ARPANET, hoje conhecida como internet (SIMON, 1997).

Após a difusão da internet, no ano de 1999, Kevin Ashton desenvolveu uma arquitetura de sensores conectados entre si utilizando-a como intermediadora da conexão (SANTOS, et al., 2017). Este sistema foi denominado como "Internet das Coisas", tendo em vista que viabilizava o tráfego de informações entre dispositivos conectados virtualmente. Contudo, no contexto em que o termo foi concebido, as tecnologias de infraestruturas do meio *web* e os *hardwares* que possibilitavam a conexão à internet não viabilizavam o desenvolvimento de tal sistema de forma comercial.

Devido à contínua contribuição de grandes empresas (por exemplo: Google, Facebook, Espressif, Nordic Semiconductors, dentre outras), ocorreu o surgimento de novas tecnologias no que diz respeito a *hardwares*, infraestrutura *web* e ferramentas de desenvolvimento para aplicações *web*. Essas tecnologias possibilitam o tráfego de informações em tempo real, persistência de informações em banco de dados hospedados remotamente e confecção de *layouts* para interação com usuários com um bom desempenho e responsivos (layouts que respondem ao tamanho da tela, tornando a aplicação utilizável em qualquer dispositivo). Tais tecnologias propiciam o desenvolvimento de sistemas de automação interconectados por meio da rede mundial.

Tendo em mente a viabilidade em desenvolver sistemas de automação utilizando o conceito de internet das coisas (IoT), pode-se aplicar esse conceito em situações nas quais há necessidade de monitorar e controlar alguma variável. No ambiente residencial existem diversas variáveis que podem ser controladas e que possibilitam a aplicação do conceito apresentado, sendo uma delas o nível do reservatório de água.

O monitoramento do reservatório de água é uma demanda crescente devido ao cenário da crise hídrica presente no Brasil nos períodos de seca, pois, a interrupção do fornecimento de água para moradores residenciais faz com que muitas vezes esses reservatórios cheguem a níveis críticos. Assim, muitos moradores tomam ciência da falta do fornecimento somente quando seus reservatórios se esgotem por inteiro.

Estima-se que o Brasil tenha cerca de 12 % da disponibilidade mundial de água. Entretanto, apenas cerca uma parte da população brasileira, cerca de 5% é atendida desse total (AGÊNCIA NACIONAL DE ÁGUAS, 2019).

Em 2017, houve uma das piores crises hídricas do Brasil, a qual foi mencionada no "Fórum Mundial da Água" no ano seguinte por deixar diversos moradores dos estados de São Paulo, Distrito Federal e Ceará sem fornecimento de água (G1, 2018). Tendo em vista o exposto, a crescente falta de fornecimento de água justifica a necessidade de um sistema de monitoramento e controle do nível de reservatório que forneça aos moradores meios preventivos, os quais evitem o nível morto (reservatório vazio) de seus reservatórios e forneçam informações em tempo real da quantidade de água disponível.

Em 2014, após a grave crise hídrica em Formiga-MG, o trabalho de Oliveira, Santos e Marco (2014) iniciou o desenvolvimento de protótipo para a medição do nível de água de reservatórios residenciais, utilizando módulo Bluetooth e um sensor de pressão MPX 5010, obtendo bons resultados em testes.

Mais tarde, em 2015 o, trabalho de Oliveira (2015) deu continuidade ao trabalho apresentado anteriormente, realizando a instalação do sistema em uma caixa d'água, modificando o sensor da pesquisa anterior, pois, este apresentava problemas em longos períodos de medição, substituindo-o por um sensor de nível automotivo. Além disso, este projeto aumentou a distância entre o usuário e o sistema de medição utilizando um módulo transceiver, emitindo o sinal a centenas de metros.

Após mais uma crise hídrica, ocorrida em 2017 na cidade de Formiga, Lopes (2018) atualizou o sistema, conectando-o à internet por módulo ethernet e Arduino. Adicionalmente, testou dois sensores, um ultrassônico e continuando com o de combustível, verificando problemas no ultrassônico, causados por condensação de água.

Com base nos aspectos supracitados sobre a crise hídrica em períodos de seca, o presente trabalho tem como objetivo apresentar o desenvolvimento de um sistema de monitoramento de nível para reservatórios de água residenciais. Este utilizará o conceito de IoT que proporciona ao usuário o acesso às informações por *website* e um sistema de reabastecimento.

## 1.1 Formulação do Problema

A crise hídrica é uma realidade em que toda a população brasileira está inserida, porém, os que são mais afetados são os pequenos consumidores, nesse caso, consumidores residenciais que eventualmente enfrentam situações nas quais seus reservatórios ficam sem fornecimento

de água gerando, um cenário precário de saneamento básico. Um fator agravante dessa situação é a falta de informação sobre o fornecimento em um período de crise, pois, na maioria dos casos os moradores tornam-se cientes apenas quando seus reservatórios atingem o volume morto (vazio). Tal fato ocorre por não existir no mercado um produto acessível que monitore o nível do reservatório e que incorpore uma forma de reabastecimento por meio de um reservatório secundário, o qual possibilite uma maior capacidade de armazenamento de água e autonomia para tomar medidas de racionalização.

## 1.2 Justificativa

Analisando a falta de fornecimento de água como reflexo de uma crise hídrica no Brasil, com base no aumento da densidade demográfica do país, podemos concluir que a tendência é que esse cenário de falta de água venha agravar-se. Ao mesmo tempo que a população cresce, a falta de conscientização e utilização indevida de recursos hídricos contribui para o cenário de desequilíbrio vivenciado pela sociedade.

Um agravante da crise hídrica é o fato de alguns moradores não terem conhecimento da falta do fornecimento de água, pois tendo essa informação o consumidor teria a possibilidade de tomar medidas de racionalização com o objetivo de precaver que seus reservatórios não atinjam o volume morto. Desta forma, o sistema de monitoramento do nível do reservatório, em conjunto com um sistema de reabastecimento, pode fornecer ao consumidor residencial uma maior estabilidade na falta do fornecimento de água tendo uma maior reserva de água e também a capacidade de monitorar a disponibilidade de água em seu reservatório capacitando-o a tomar medidas de racionalização.

## 1.3 Hipótese

Contextualizado o persistente problema, por mais que o sistema de monitoramento e reabastecimento do reservatório de água não atuem de fato na solução da falta do fornecimento, a conscientização do consumidor do nível de seu reservatório faz com que ele tenha conciência do problema e uma maior comodidade em períodos que a seca e a falta de fornecimento de água prolongam-se.

## 1.4 Objetivos

### 1.4.1 Objetivos Gerais

O principal objetivo do trabalho é desenvolver uma forma de monitorar o nível de reservatórios residenciais acoplado a um sistema de reabastecimento, fornecendo ao usuário a autonomia nas tomadas de decisões em situações de falta de fornecimento de água e um aumento na capacidade de armazenar água. O sistema desenvolvido é um aperfeiçoamento do trabalho de conclusão de curso apresentado por Lopes (2018). Assim, este trabalho terá uma nova abordagem no desenvolvimento da infraestrutura exposta no trabalho anterior.

### 1.4.2 Objetivos Específicos

Os objetivos específicos do trabalho, são:

- Desenvolver *firmware* e protótipo de *hardware* capaz de realizar as aquisições de níveis do reservatório enviando-as para um servidor remoto. O *firmware* conterá: acionamento remoto de uma bomba em função do nível, otimização no consumo de energia, conexão à internet por meio de rede Wi-Fi, interface de gerenciamento de conexão (Wi-Fi *Manager*) e variáveis de ambientes configuradas pela interface do usuário.
- Desenvolver aplicação *back-end* (*web-service*), capaz de receber as informações de nível, persistindo-as em uma base de dados, a qual será intermediador da comunicação entre usuário e sensor.
- Desenvolver aplicação *front-end* (*web app*), sendo a interface do usuário, possibilitando que o mesmo tenha acesso às informações de nível em tempo real e interação com o sensor de nível.
- Desenvolver proposta de projeto para automação de um sistema de reabastecimento.
- Aplicar medidas de segurança no tráfego de informações pela internet.

## 1.5 Estrutura do Trabalho

O presente trabalho é dividido em capítulos e subcapítulos, com o intuito de oferecer uma melhor experiência durante a leitura do mesmo.

- Capítulo primeiro, introdução, aborda a contextualização do assunto apresentado no trabalho, objetivos gerais, objetivos específicos e a estrutura do trabalho.
- Capítulo segundo, fundamentação teórica, apresenta os conhecimentos básicos que fundamentaram o desenvolvimento do trabalho.
- Capítulo terceiro, matérias e métodos, elucida as etapas e metodologias utilizadas para a elaboração do projeto proposto.
- Capítulo quarto, resultados e discussões, apresenta os resultados obtidos no desenvolvimento juntamente com as análises realizadas para a otimização do sistema.
- Capítulo quinto, conclusões, descreve o resultado como um todo do trabalho desenvolvido
- Capítulo sexto, trabalhos futuros, enumera possíveis implementações para melhoria do trabalho desenvolvido.
- Capítulo sétimo, referências bibliográficas.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo aborda-se os principais conceitos que foram utilizados como base para a confecção deste trabalho. Inicialmente é apresentado uma forma de converter uma grandeza física e manipulá-la por meio de um transmissor e controlador, em seguida enuncia-se a concepção da arquitetura *web* e as duas formas de interação mais utilizadas em sistemas de IoT. Por fim exemplifica o diagrama de fluxo de processo e os procedimentos necessários para acionar uma bomba d'água.

## 2.1 Transmissor e Controlador de Nível

Transmissor de nível é um dispositivo capaz de transformar a propriedade física que corresponde à quantidade de volume que um fluido ou um sólido ocupa em um recipiente em uma outra grandeza física que é transmitida e interpretada por um controlador, este por sua vez é responsável por receber a "informação" correspondente ao nível, analisá-la e fornecer ao sistema uma saída controlável, viabilizando assim o controle do nível do reservatório. Em sistemas de IoT esses dois conceitos são embarcados em um único dispositivo, muitas vezes

denominado como sensor de nível, no qual este apresenta as competências abordadas anteriormente. A aquisição do nível fornecida pelo transmissor pode ser obtida por:

- Medição indireta: conversão do nível em uma grandeza física a ele relacionada utilizando transdutores;
- Medição direta: medição do nível com relação a uma referência pré-estabelecida;
- Medição descontínua: tendo apenas a indicação de alguns pontos como, por exemplo, nível baixo e nível alto (CASSIOLATO, 2010).

Comumente utiliza-se a medição indireta para aplicações voltadas em IoT por ter a capacidade de converter a grandeza de nível em uma grandeza elétrica que é facilmente interpretada pelos controladores.

O controlador para possuir as funcionalidades apresentadas anteriormente é construído por dispositivos eletrônicos, que possibilitam a conversão do sinal elétrico analógico recebido do transdutor, para um sinal digital de forma a propiciar o tratamento lógico e acionando dispositivos periféricos capazes de realizar alguma tarefa, como proposto no projeto, que é o acionamento de uma bomba para o reabastecimento de um reservatório. Tais dispositivos eletrônicos são denominados microcontroladores.

### 2.1.1 Microcontroladores

São componentes eletrônicos que encapsulam em um único chip: processador, memória RAM, memória EEPROM e periféricos, possibilitando que o mesmo seja programado para realizar uma rotina. Desenvolvidos para o fim de controlar periféricos, possui uma ampla aplicação em sistemas embarcados voltados para automação (PEREIRA, 2002).

A facilidade e praticidade de desenvolver soluções utilizando sistemas microcontrolados acelerou a evolução tecnológica, ao ponto de ser concebida a arquitetura denominada *Advanced RISC Machine* (ARM), sendo uma família de microcontroladores com um alto processamento e um consumo de energia reduzido (OLIVEIRA, 2017).

Dentre os diversos fabricantes e modelos da arquitetura ARM, destacam-se o ESP32, por ter um valor acessível, viabilidade de desenvolvimento por meio do Arduino IDE e um kit de fácil utilização. A Tabela 1 apresenta as principais características da MCU (*Microcontroller Unity*):

Tabela 1 - Especificações ESP32 (ESPRESSIF, 2019).

| **MCU** | **ESP32** |
|---|---|
| Processador | Dual Core - 32 bits |
| Memória ROM | 338KB |
| Memória Flash | 4MB |
| Tensão de operação | 3,3V |
| Corrente de operação | ~120mA |
| Periféricos | Wi-Fi, Bluetooth 4.2, ADC 12 bits |
| Quantidade de Pinos | 48 |

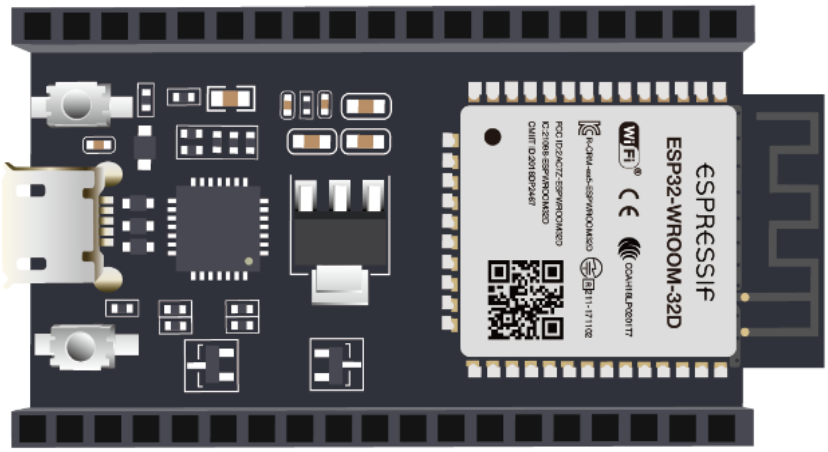



Figura 1 - ESP32 (ESPRESSIF, 2019).

Além das características e vantagens apresentadas anteriormente, o ESP32 viabiliza a utilização de dois conceitos bastantes importantes em aplicações de IoT, que são, *sleep mode* e Wi-Fi *manager.*

*Sleep mode* é a tecnologia que possibilita configurar o MCU em *stand-by*, assim toda vez que o microcontrolador estiver aguardando algum evento de interrupção por *timer* ou interrupção externa, o mesmo entrará em “hibernação”, ficando ativo apenas o subprocessador RTC (*Real Time Controller*) de baixo consumo de energia, reduzindo assim o consumo. Em aplicações de IoT, a parte do hardware responsável por conectar-se à internet faz com que o dispositivo tenha um consumo elevado de corrente elétrica. Analisando o kit de desenvolvimento do ESP32, o consumo pode chegar até 120mA quando a antena Wi-Fi é

ativada. Em contrapartida quando o MCU ativa o modo de *sleep mode,* fica em operação somente o processador RTC e alguns periféricos de baixo consumo de energia, reduzindo o consumo de corrente elétrica para, aproximadamente, 20 µA. (ESPRESSIF, 2019).

Wi-Fi *manager* é um termo utilizado para aplicações de gerenciamento de conexão a redes Wi-Fi. Quando tem-se a necessidade de estabelecer a conexão a uma rede sem fio, faz-se necessário oferecer ao usuário uma forma simplificada de configurar o acesso à rede no *hardware*. Com essa finalidade a aplicação de Wi-Fi *manager* torna o dispositivo portador da tecnologia de conexão a redes sem fio, um roteador e um servidor local. Dessa forma o usuário inicialmente conecta-se na rede gerada pelo *hardware* possibilitando o acesso à aplicação servida. Nela, encontra-se a aplicação que permita ao usuário inserir as credenciais de acesso à rede Wi-Fi desejada. Por fim, a aplicação salva as credenciais informadas na memória flash do microcontrolador, que é uma memória não volátil.

### 2.1.2 Transdutores

Transdutores são dispositivos que têm a finalidade de converter grandezas físicas ou químicas em grandezas elétricas tornando possível a manipulação e amostragem de tais grandezas. Existem diversos tipos de transdutores que possibilitam a conversão do nível de um reservatório em um sinal de tensão ou de corrente. Para aferição de nível em reservatórios de água residenciais destaca-se dois entre os diversos modelos de transdutores: sensores ultrassônicos e sensores de nível automotivo.

Os Sensores ultrassônicos são compostos por um emissor e um receptor de ondas ultrassônicas, permitindo a medição do nível a partir do tempo decorrido entre a transmissão e a recepção do sinal sonoro. Porém como apresentado por Lopes (2018), para aplicações em que o nível a ser medido é de um fluido confinado em um reservatório que é exposto à radiação solar, esse sensor é prejudicado pela condensação de água no sensor, gerando falhas no processo de aquisição de nível.

O sensor de nível automotivo é composto basicamente por quatro elementos, exemplificado na Figura 2.

- Boia: composta por material de baixa densidade de forma que ao entrar em contato com o líquido no reservatório, estabiliza-se na superfície por flutuação. Utilizada para acompanhar o nível do líquido, movimentando a haste do sensor.
- Haste: tem como finalidade variar a resistência elétrica do potenciômetro em função da movimentação da boia, muitas vezes confeccionada com um material antioxidante.
- Resistência variável: conectada à haste, sendo variada em função da variação do nível do fluido.
- Terminais: concedem acesso a resistência variável do sensor.

Figura 2 - Sensor de nível automotivo (MIXAUTO, 2019).

O sensor automotivo converte a variação do nível em uma variação de uma resistência elétrica. Por meio dos terminais é possível inserir o sensor em um circuito divisor de tensão, gerando uma variação linear de tensão, proporcional a variação de nível.

## 2.2 Aplicação *Web*

O ecossistema *web* vem sendo moldado ao longo de décadas, com a crescente demanda do acesso à informação em tempo real e da viralização das redes sociais, em que uma grande

quantidade de usuários acessa simultaneamente uma única plataforma. Isso fez com que as tecnologias *web* evoluíssem para atender a essa necessidade e como consequência surgiu o conceito de *front-end* e *back-end*, que são serviços descentralizados capazes de realizarem tarefas com alto desempenho consumindo menos recursos computacionais dos servidores, tais tecnologias são abordadas nas seções seguintes e contextualizadas para aplicações em IoT.

### 2.2.1 HTTP - HTTPS – WebSocket

O envio de informações na internet pode ser compreendido quando analisada a camada de aplicação da arquitetura TCP/IP, que prevê protocolos para gerenciar tal ação. Destes destaca-se: HTTP, HTTPS e WebSocket, oferecendo formas distintas para realizar o trafego de informações.

O protocolo HTTP (*Hyper Text Transfer Protocol*) provê à aplicação uma forma de enviar informações com o formato de requisição / resposta (RFC 7231, 2014). O envio é concebido pela requisição que é construída por um cabeçalho e por um corpo. O cabeçalho de uma requisição pode conter: o tipo de serviço solicitado (*post*, *get*, *put*, *delete*, dentre outros), normas de autenticação / autorização com intuito de proteger as informações do corpo e o formato da informação contida no corpo, podendo ser formatos do tipo: texto, JSON (*Java Script Object Notation*), form-data, dentre outros. Já o corpo de toda requisição contém as devidas informações que se deseja trafegar.

Em aplicações voltadas para IoT o formato de requisições mais utilizado é o JSON que possibilita o envio de informações organizadas em chave-valor facilitando o tratamento das mesmas.

A estrutura do protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) é semelhante ao HTTP, o que o torna único é a utilização do protocolo criptográfico SSL (*Secure Sockets Layer*) nas requisições, fazendo com que todo o tráfego de informação ocorra de forma criptografada (RFC 2818, 2000). A Figura 3 exemplifica o tráfego de informação apresentado anteriormente.

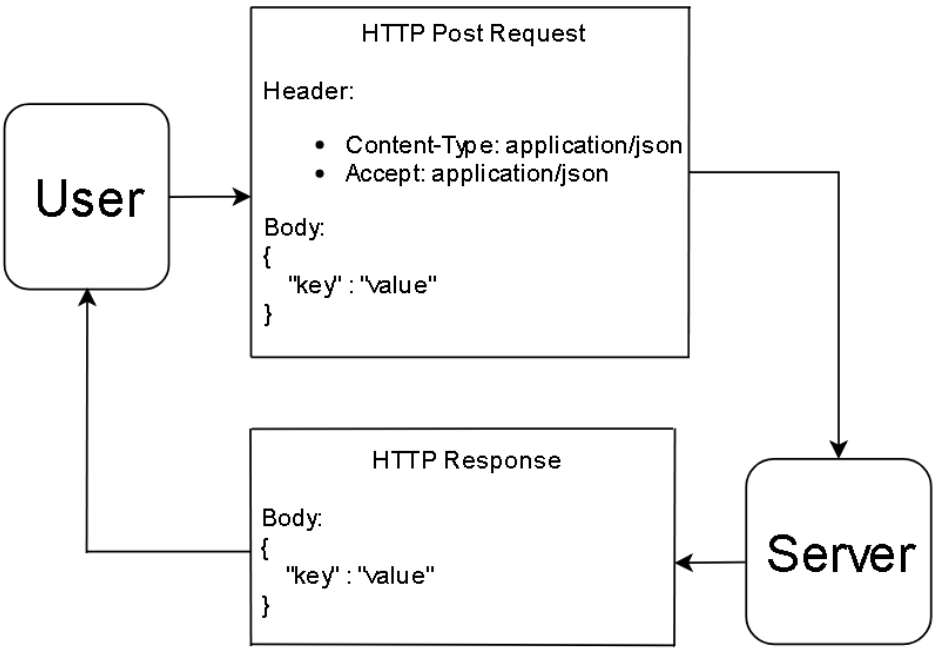



Figura 3 - HTTP post.

*Web Socket* é uma tecnologia de intercomunicação que fornece o tráfego de informações em tempo real entre duas aplicações distintas na internet, a comunicação é baseada em uma arquitetura de cliente / servidor, onde o servidor provê uma instância de intercomunicação permitindo que a aplicação construída como cliente conecte-se e trafegue informações de forma bidirecional por meio dos canais gerados (RFC 6455, 2011).

### 2.2.2 Infraestrutura

A infraestrutura é a camada física que sustenta a aplicação, sendo responsável por armazenar toda a estrutura de arquivos e a base de dados, fornecendo acesso remoto por meio do protocolo IPv4 e / ou IPv6 e provendo um serviço de DNS (*Domain Name System*). Essa infraestrutura muitas vezes é hospedada remotamente por empresas que garantem o funcionamento da instância contratada e conexão à internet de forma contínua com IP fixo. Existem diversos serviços disponíveis no mercado, o mais utilizado é o serviço denominado de máquina virtual, que fornece acesso a uma instância de processamento e armazenamento em que a capacidade de armazenamento e processamento é o fator de análise para avaliar o desempenho da aplicação. Esse tipo de serviço é um dos mais utilizados por fornecer total

controle da instância contratada possibilitando a criação de sistemas descentralizados que são os comumente utilizados em IoT.

Deve-se ter ciência que as instâncias contratadas em serviços de hospedagem fornecem acesso remoto a todos que possuam as credenciais de acesso. Por esse fato existem procedimentos necessários a serem realizados para garantir a segurança da instância e da aplicação hospedada. A configuração de uma chave de conexão SSH (*Secure Shell* – protocolo de acesso remoto) garante que somente o contratante tenha acesso à infra-estrutura porém não garante que as informações trafegadas à instância por meio de outros protocolos de intercomunicação sejam protegidas pois elas podem ser interceptadas. Assim, uma forma de permutar esse grave problema de segurança é por meio do certificado de criptografia TLS, que, quando configurado, garante que a informação trafegada pela instância seja criptografada.

### 2.2.3 Back-end

O termo *back-end* surgiu na arquitetura moderna de desenvolvimento *web,* sendo um serviço descentralizado e especializado, tendo como finalidade gerenciar os dados da aplicação, recebendo as informações provenientes dos "*gateways"* e dos usuários, direcionando-as assim para seus devidos fins e se houver a necessidade, armazenando-as em uma estrutura de banco de dados.

Têm-se no ecossistema *web* diversas formas de se construir tal aplicação, cada uma possui suas vantagens e desvantagens dependendo do contexto em que serão inseridas. Das diversas tecnologias três destacam-se no mercado: Apache, Python e NodeJs.

Apache é uma tecnologia de servidor web multi *thread* (sub processos que dividem-se em diversas tarefas) de gerenciamento de requisições podendo prover conteúdo estático de uma aplicação *web.* O conteúdo estático é todo o conteúdo construído utilizando HTML (*Hyper Text Markup Language*) e CSS (*Cascading Style Sheets*). A linguagem utilizada para desenvolver o *back-end* em Apache é o PHP (*Hypertext Preprocessor*) (APACHE, 2019).

Python é uma linguagem de programação orientada a endentação viabilizando a construção de servidores *web* por meio de *frameworks*, que são bibliotecas de terceiros, criadas com o intuito de otimizar o desenvolvimento. Para a criação de um servidor HTTP multi *thread* o *framework* mais utilizado em phyton é o Django (PYTHON, 2019).

NodeJs é uma tecnologia *single thread* que viabiliza a criação de aplicações *web* por meio de *frameworks* utilizando JavaScript como sua linguagem de desenvolvimento. Dessas três tecnologias citadas, o NodeJs destaca-se na atualidade por possuir uma arquitetura simplificada e uma linguagem de programação de fácil desenvolvimento, no qual empresas como Netflix, Uber, Walmart, PayPal utilizam tal recurso em suas aplicações (SARTORI, 2017), a Figura 4 ilustra a arquitetura do NodeJs.

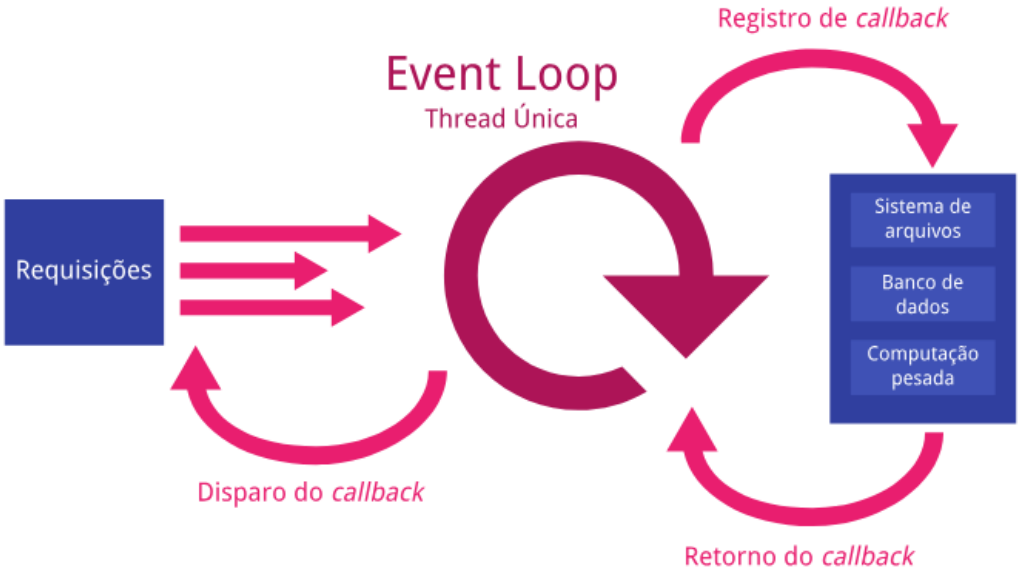



Figura 4 - Arquitetura de requisições NodeJs (ROECKER, 2015).

Por possuir uma estrutura baseada em uma única *thread*, cada requisição feita ao servidor é analisada previamente antes de ser processada. Quando a requisição requer um processamento assíncrono ou requer o envio de um conteúdo estático para renderização de páginas, essas são enviadas para o *Event Loop* e empilhadas na fila de *callbacks*, aguardando que sejam devidamente processadas. Após o processamento as mesmas são retornadas ao *Event Loop* que as envia a *thread* quando a disponibilidade de serviço. Por fim a *thread* retorna a resposta da requisição feita ao servidor. Este formato single *thread* faz com que não haja reserva desnecessária de processamento e armazenamento da instância que hospedará a aplicação (NODE, 2019).

### 2.2.4 Banco de dados

Banco de dados são estruturas que tem a finalidade de persistir informações de forma segura em uma instância hospedada, viabilizando a criação de *backups* automáticos, gerenciamento de armazenamento e otimização no acesso as informações salvas. Na atualidade tem-se basicamente duas tecnologias de banco de dados: relacionais e não relacionais.

As estruturas de dados relacionais, conhecidas como SQL (*Structured Query Language*), dispõem os dados de forma tabular utilizando de *querys* para executar as ações de manipulação dos dados. Banco de dados SQL é utilizado quando a aplicação exige um alto nível de relacionamento entre os dados. O termo relacionamento está associado a dados persistidos em tabelas distintas que fazem sentido quando analisadas em conjunto. Existem dois modos de relacionamento:

- *Has one* quando se tem o intuito de relacionar uma coluna de uma tabela com uma coluna de demais tabelas
- *Has many* utilizando quando há necessidade de relacionar *n* colunas de uma tabela com *m* colunas de demais tabelas (SEQUELIZE, 2019).

Bases de dados não relacionais, denominados no-SQL, utilizam um formato de coleção de dados em sua estrutura, sendo utilizado quando o formato de dados a serem salvos não possui um padrão definido e não há necessidade de relacionamentos complexos entre as coleções. Por ser uma tecnologia recente o custo de hospedagem de uma base de dados não relacional é mais elevado do que a base relacional.

### 2.2.5 Front-end

Front-end é a denominação atribuída à interface gráfica que conecta o usuário à aplicação, tendo uma infinidade de tecnologias e formas para conceber a interface. Em 2012 o Facebook apresentou para toda a comunidade de desenvolvedores uma forma de construir tal aplicação utilizando o conceito de componentização e virtual DOM (*Document Object Model*). Essa abordagem fez com que a interface gráfica reduzisse o consumo de recursos do servidor e possibilitasse a utilização de aplicações em tempo real cujos dados são alterados sem haver a

necessidade de renderizar todo o conteúdo estático. Tal tecnologia é denominada ReactJs (REACT, 2019).

A componentização de aplicações *web* é, na atualidade, a forma mais utilizada para construir a interface de usuário. Grandes empresas como, Netflix, Yahoo, Atlassian, Istagram e o próprio Facebook utilizam ReactJs em todas suas aplicações *web* que possuem interface gráfica (MOCIUN, 2019).

Juntamente com a componentização o desenvolvimento de uma UI (*User Interface*) utiliza-se basicamente de três linguagens:

- HTML, sendo uma linguagem de marcação de texto contendo elementos que são dispostos na arquitetura de árvore, a qual cada elemento contém funcionalidades para realizar apresentação de conteúdo ou de interação com usuário.
- CSS, utilizada para inserir aos elementos HTML uma estilização gráfica tornando a UI visualmente atrativa.
- JavaScript, fornecendo ao navegador formas de realizar interações com usuário; animações, eventos em background dentre outras funcionalidades (W3SCHOOLS, 2019).

## 2.3 Internet das Coisas

O termo “Internet das Coisas” foi concebido por Kevin Ashton em uma proposta de otimização para um sistema de identificação de objetos por rádio frequência (RFID), abordando uma conexão entre dispositivos denominada *Device to Device* (D2D) (SANTOS, et al., 2017).

Atualmente existem diversas arquiteturas para desenvolver um sistema baseado em IoT. O conceito geral consiste em utilizar a internet como forma de intercomunicação entre dois ou mais dispositivos, para que isso seja possível faz-se necessário que os *hardwares* utilizados tenham a capacidade de conectar-se à internet.

Sabendo que sistemas baseados em IoT necessitem do acesso à internet, e que dispositivos com tal tecnologia possuem um custo elevado, uma aplicação que necessita utilizar diversos *hardwares* para a arquitetura do sistema de IoT, ocasiona um alto custo para a implementação deste sistema. Com esta motivação surgiu-se o conceito de *gateway* aplicado a IoT, que são dispositivos capazes de estabelecer conexão à internet e trocar informações com

os demais dispositivos IoT utilizando protocolos de comunicação como: RS232, UART, I2C, Bluetooth, ZigBee ou até mesmo Bluetooth *Low Energy*. Sendo assim os *gateways* intermediam a comunicação entre os demais dispositivos IoT e a internet reduzindo expressivamente o valor do sistema.

### 2.3.1 Interação Dispositivo – Dispositivo

A interação D2D viabiliza a comunicação entre dispositivos intermediada por um *gateway*, possuindo basicamente três elementos: o dispositivo propriamente dito, *gateway* e *back-end*. Os dispositivos localizam-se na *workspace,* que é o local onde são instalados para executar a operação para que foram programados. O *gateway*, sendo o intermediador da comunicação, recebe as informações dos dispositivos e as envia para o *back-end*. Este por sua vez tem a capacidade de receber as informações via requisição HTTP, manipular o corpo desta requisição e a persistir em uma base de dados. A Figura 5, exemplifica a interação D2D de uma forma mais clara.

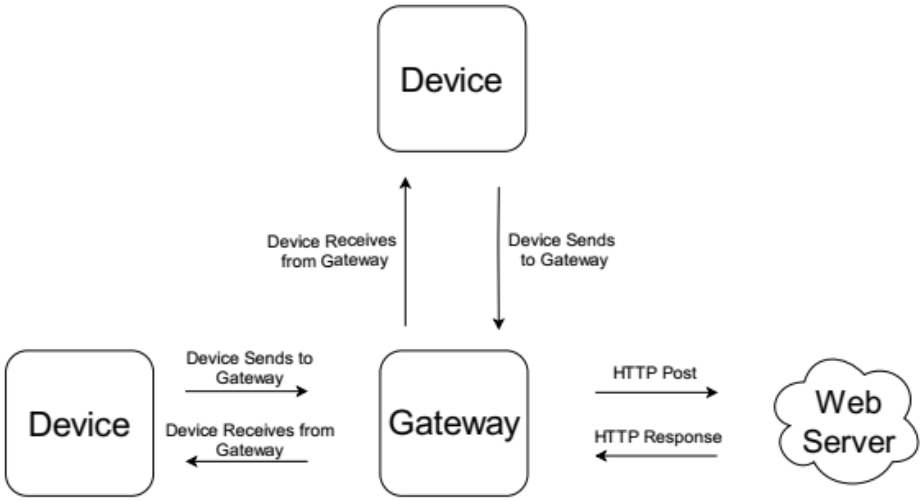



Figura 5 - Arquitetura Device to Device.

Esta arquitetura é bastante utilizada quando se tem apenas o intuito de persistir as informações em um banco de dados e / ou dispô-las para um usuário, tendo assim uma

simplicidade no desenvolvimento da aplicação como um todo. A topologia D2D não possibilita a interação de um usuário com o dispositivo.

### 2.3.2 Interação Usuário – Dispositivo

Quando a necessidade do usuário vai além de receber informações dos dispositivos, mas também enviar informações para o *gateway*, caracteriza uma interação entre usuário e dispositivo. Para essa abordagem existem diversas soluções, cada solução depende do tipo de *gateway* utilizado. Desta forma, pode-se generalizar a arquitetura *user to device* (U2D) como apresentado na Figura 6.

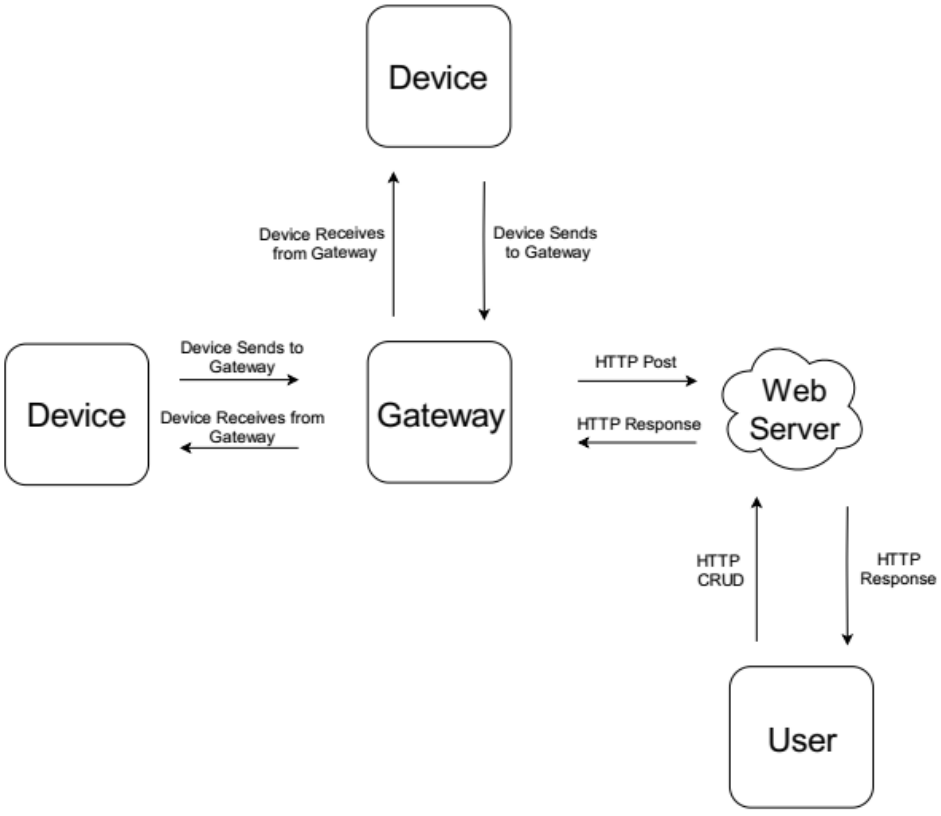



Figura 6 - Arquitetura User to Device.

A interação do usuário é concebida por meio da aplicação *web* denominada REST CRUD, que implementa os métodos *post*, *get*, *put* e *delete* do protocolo HTTP. Analisando a Figura 6, pode-se abstrair o funcionamento da arquitetura onde o objetivo é utilizar o servidor

como intermediador na comunicação do usuário com o *gateway* e este, como intermediador na comunicação entre o servidor e os dispositivos.

Para uma análise mais detalhada, assumindo o início da comunicação na interação do usuário, a UI envia uma requisição do tipo *post* para o servidor contendo as informações com destino ao *gateway* no corpo da requisição. Ao receber essas informações, o servidor salva-as no banco de dados. No momento em que os dispositivos enviarem informações ao *gateway*, este processa as informações e também as envia para o servidor utilizando o método *post*; este ao receber a requisição proveniente do *gateway*, avalia se existe solicitações enviadas pelo usuário, se existir responde a requisição com as devidas informações.

## 2.4 Diagrama de fluxo de processo P&ID

Diagrama de fluxo de processo P&ID (*Piping and Instruments Diagram*) tem como intuito documentar processos a fim de obter detalhes de operação sobre o mesmo, seguindo um compilado de normas da ABNT (Associação Brasileira de Normas Técnicas) e ISA (*Instrument Society of America*), criadas para fim padronizar a simbologia do diagrama (TRIERWEILER, 2019). Assim a confecção de um P&ID é necessária quando se tem o intuito de projetar e / ou automatizar um processo. Muitas vezes esse conceito está aplicado às industrias, porém com a evolução tecnológica e o desenvolvimento de sistemas de IoT, essa abordagem vem sido trazida para realidade de automação residencial.

O diagrama é construído utilizando simbologias que representam cada componente do sistema analisado, tais simbologias são normatizadas pelas normas NBR8190 e ISA 5.1, porém fica a cargo do projetista determinar o padrão a ser utilizado. A Figura 7 ilustra um diagrama P&ID construído com base na planta de nível disposta no laboratório de instrumentação do Instituto Federal de Minas Gerais campus Formiga.

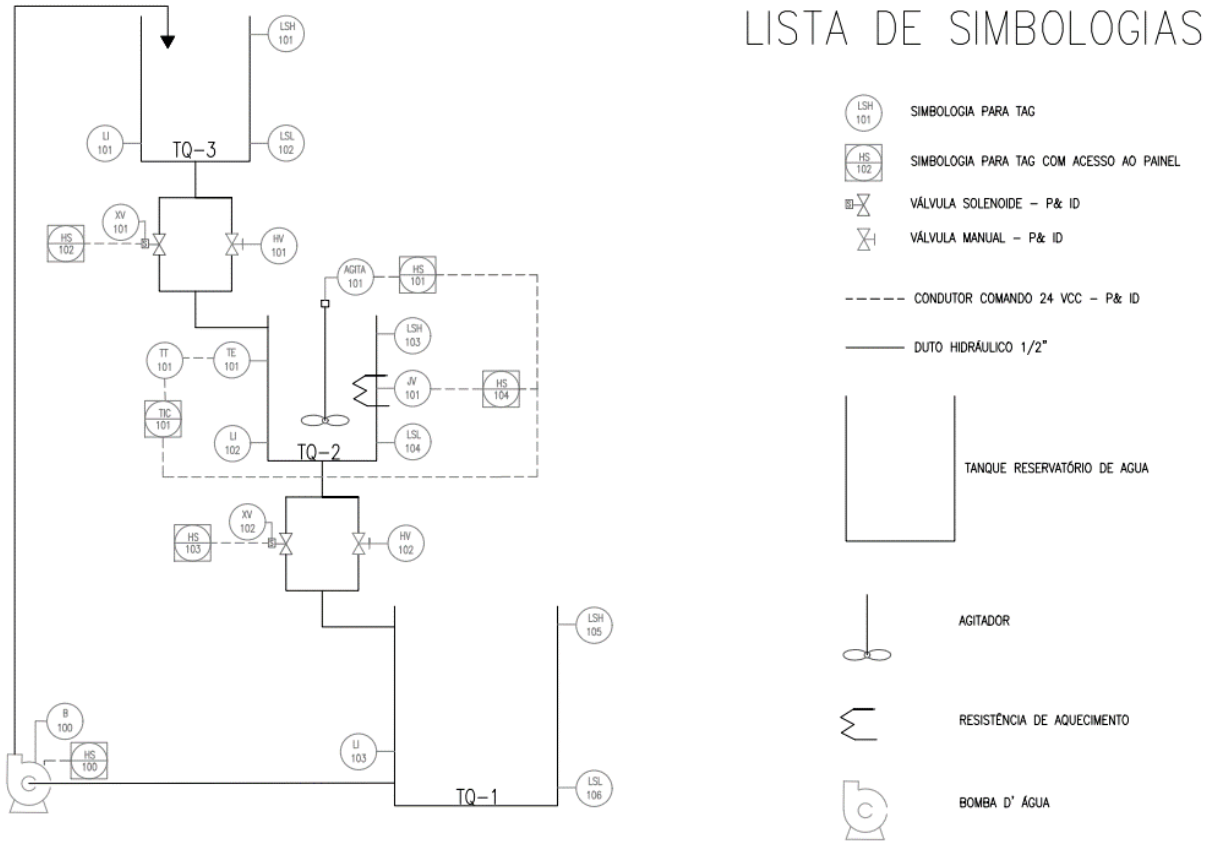



Figura 7 - Diagrama P&ID.

## 2.5 Acionamento de bombas d'água

Bomba d'água é um dispositivo capaz de converter energia elétrica em elevação de pressão e vazão de um líquido, sendo utilizada quando há necessidade de transmitir o líquido de um reservatório para outros reservatórios e/ou para aplicações específicas. A bomba é composta por um mecanismo mecânico e por um motor elétrico de indução. Por motores elétricos serem uma carga não linear e passivo a falhas mecânicas e elétricas, faz-se necessário utilizar de dispositivos de proteção para que no momento do acionamento não ocasionem nenhum dano ao equipamento e à rede em que é instalado (MAMED, 2015).

Em aplicações que se é necessária a utilização de uma bomba é fundamental que a mesma seja devidamente dimensionada. Para tal pode-se utilizar a equação 1 (MAMED, 2015), em que é possível determinar a potência elétrica requerida pela bomba em função da vazão necessária.

$$P_b = \frac{9{,}8\ x\ Q\ x\ \gamma\ x\ H}{\eta} \qquad [1]$$

Onde:

Pb – Potência requerida da bomba para atender a demanda [KW];

Q – Vazão [$m^3/s$];

γ – Peso específico do líquido [$kgf/dm^3$];

H – Altura da elevação [m];

η – Eficiência da bomba, tendo que para bombas pistão varia de 0,87 a 0,9 e para bombas centrífugas varia entre 0,40 a 0,9;

Sabendo que a bomba é composta por um motor elétrico, o momento em que este sai de sua inércia até atingir a velocidade nominal ocorre um fenômeno denominado de corrente de partida, que é um pico momentâneo de corrente podendo chegar até oito vezes o valor da corrente nominal do motor. Desta forma para o acionamento de motores deve-se utilizar dispositivos que possuem uma resposta a picos de corrente de uma forma mais lenta, tendo em mente que o pico de corrente ocasionado na partida ocorre em um intervalo de tempo muito pequeno. Os dispositivos utilizados para o acionamento e proteção de um motor são:

- Disjuntor motor, possuindo a capacidade de proteção em situações de sobrecorrente e sobretensão;
- Fusível diazed e fusível NH, protegendo cada fase do motor contra sobrecorrente;
- Relé térmico, protegendo todas as fases do motor em situações de sobre corrente;
- Contator: utilizado para acionar um motor por comandos remotos (MAMED. 2015).

Desta forma os dispositivos são dimensionados em função das características do motor a ser acionado, tais características são: tensão nominal, corrente nominal, potência, corrente de partida, rendimento e tempo de rotor bloqueado. O dimensionamento é um procedimento bastante delicado de ser realizado, pois um pequeno erro cometido no procedimento pode ocasionar sérios danos durante a operação do equipamento. Por este fato a WEG desenvolveu chaves de partidas em função da potência dos motores a serem acionados, onde apresentam tanto o acionamento, quanto a proteção para partida de motores elétricos monofásicos ou até mesmo trifásicos (WEG, 2019).

# 3 MATERIAIS E MÉTODOS

No presente capítulo aborda-se as metodologias utilizadas para o desenvolvimento do projeto proposto. Para este fim, inicialmente apresenta-se a análise realizada do sistema proposto. Após expõe a concepção e procedimentos utilizados no desenvolvimento do sensor de nível, reabastecimento e *web* site, por fim enumera os materiais necessários para a execução do projeto.

Com intuito de desenvolver um sistema de monitoramento de nível de um reservatório com reabastecimento aprimorando a abordagem apresentada por Lopes (2018), abstraiu-se o diagrama de fluxo de processo apresentado na Figura 8.

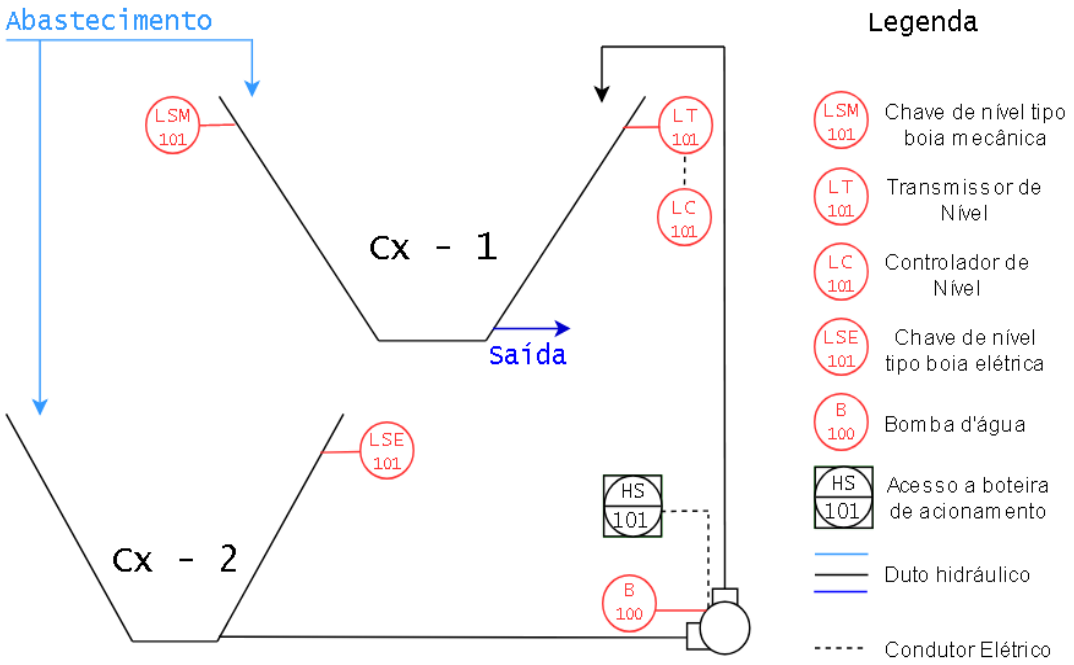



Figura 8 - Sistema a ser controlado.

A confecção de tal digrama possibilita antes mesmo de implementar o sistema, analisar e abstrair as melhores formas para realizar o controle proposto. Optou-se por confeccionar o sistema contendo dois reservatórios, ambos reservatórios abastecidos pela companhia de saneamento básico, porém o principal recebe também o reabastecimento proveniente do segundo reservatório.

O reservatório secundário deve ser instalado em um local com o nível mais próximo ao do solo, pois muita das vezes a baixa pressão do fornecimento de água impede que o reservatório principal receba o fornecimento de água, porem com o reservatório secundário instalado em um nível inferior, este recebera o fornecimento o sistema de bombeamento transferira a água para o reservatório principal.

Para poder monitorar o nível em tempo real construiu-se o sensor de nível, sendo este composto por dois elementos principais: transdutor responsável por converter o nível em uma grandeza elétrica e microcontrolador, permitindo a interpretação e manipulação do sinal elétrico proveniente do transdutor. Utilizou-se o sensor de nível automotivo com base no trabalho apresentado por Lopes (2018), o qual comprovou que o mesmo possui uma resposta linear à variação de nível quando instalado corretamente.

Tendo em mente que o sensor a ser desenvolvido tem a necessidade de conectar-se à internet para enviar as informações de nível para o *web site*, utilizou-se o microcontrolador ESP32 contendo integrado ao devKit driver e antena Wi-Fi. Por ser um processador ARM, viabilizou também a programação utilizando a tecnologia de *Deep Sleep* para otimizar o consumo de energia do sensor e a aplicação de Wi-Fi *Manager* sendo uma forma simples de configurar e registrar o *access point* Wi-Fi ao sensor de nível.

Após definir os dois elementos fundamentais para a confecção do sensor, com intuito de oferecer total controle sobre o sistema ao usuário, implementou-se a configuração por meio do *web* site das seguintes variáveis: dimensão do reservatório principal, tempo de aquisição, tempo de amortização máximo (tendo a finalidade de aumentar o tempo de aquisição quando o nível do reservatório é maior que 70% reduzindo o consumo de energia do sensor), tempo de amortização mínimo (reduzindo o tempo da coleta dos dados quando o nível é inferior a 40% aumentado assim a eficiência do controle) e configuração do acionamento da bomba. Todas essas implementações visaram oferecer maior controle sobre o sistema para o usuário.

Desta forma construiu-se o sensor com base na arquitetura U2D de IoT, o diagrama em blocos apresentado na Figura 9 ilustra a lógica de funcionamento do sensor.

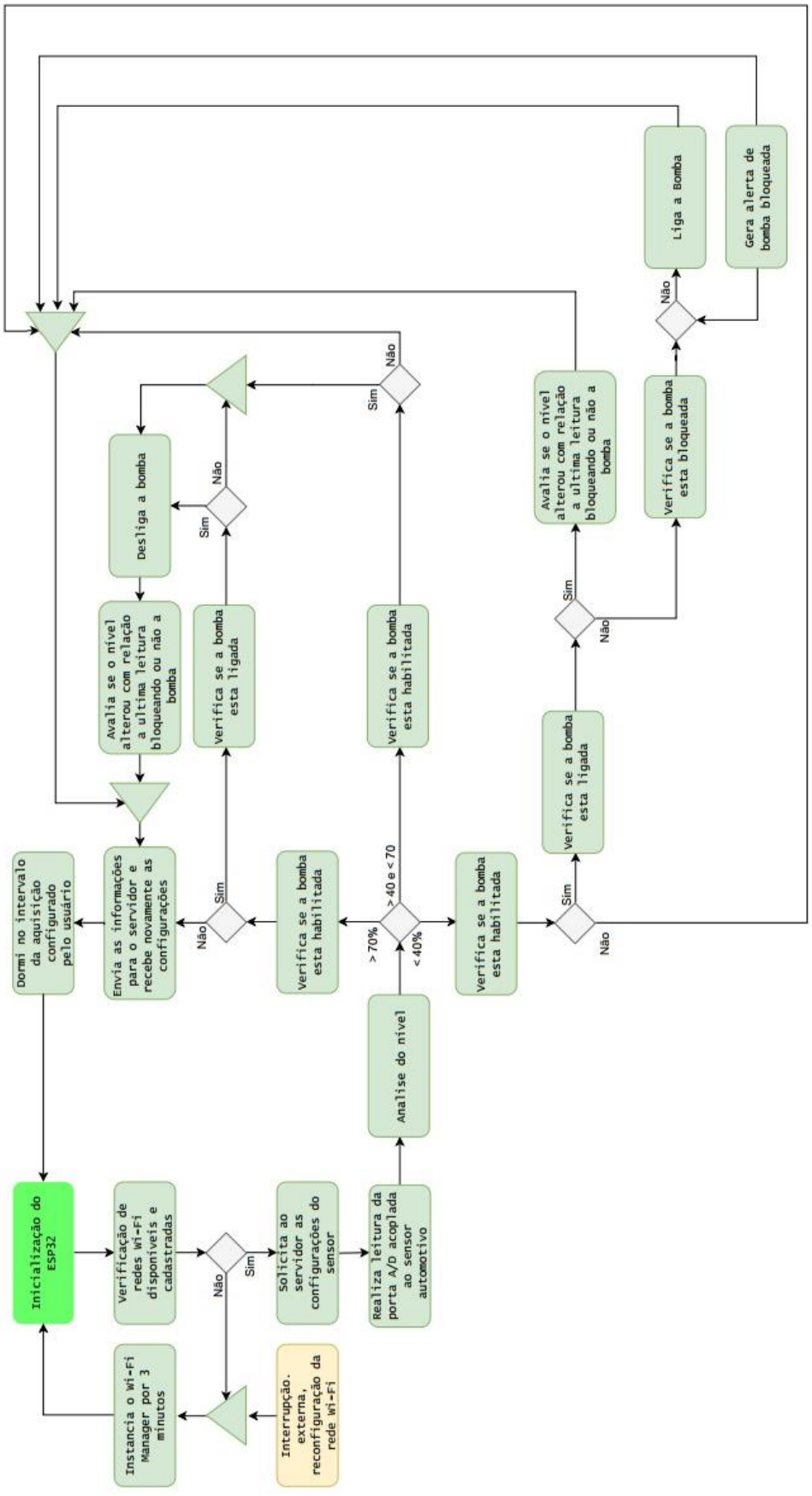



Figura 9 - Diagrama do Sensor de Nível.

Construiu-se o protótipo em uma *protoboard* com intuito de validar o funcionamento, após a constatação do funcionamento projetou-se um protótipo de *hardware* para a implementação do sistema utilizando o software *Eagle* da Autodesk que é uma ferramenta CAD para desenvolvimento de circuitos eletrônicos possuindo uma licença gratuita.

O reabastecimento é composto por dois elementos principais: a bomba e o circuito de acionamento, quando configurado na UI o acionamento da bomba de forma automática o sistema monitorador de nível será encarregado de ligar e desligar a bomba em função do nível da caixa principal. Porém houve a necessidade de inserir duas condições de segurança para o acionamento:

- Avaliação do nível do reservatório secundário, tendo que a bomba não pode ser acionada em caso de nível baixo, desta forma faz-se necessário a inserção de uma chave boia elétrica, na qual o contato normalmente fechado permanecerá neste estado enquanto o nível do reservatório secundário for ideal para o acionamento da bomba, caso contrário o mesmo não permite o acionamento ou até mesmo desenergiza a bomba.
- Possíveis falhas no sistema de reabastecimento, toda vez que o nível do reservatório principal for inferior a 40%, a bomba é acionada até o momento em que o reservatório atinja o nível de 70%. Neste intervalo o sensor armazena sempre a última leitura realizada, possibilitando comparar o nível atual com o nível da última leitura, com isso pode-se constatar se houve um aumento ou não do nível do reservatório principal. Assumindo que a bomba esteja ligada, caso o nível seja igual ou menor de uma leitura anterior, interpreta-se essa condição como uma possível falha no sistema, desligando a bomba e enviando um alerta ao usuário.

Contemplando as especificações apresentadas referente ao acionamento dimensionou-se a bomba e projetou o circuito de acionamento. Tendo em mente que a potência da bomba é relativamente baixa, pode-se utilizar o kit de acionamento WEG para otimizar o processo de dimensionamento dos componentes de acionamento.

A aplicação *web* que amostrará em tempo real as informações do sensor e fornecerá ao usuário a possibilidade de configurar o sistema foi construída de forma descentralizada, contendo: *back-end* e *front-end*.

O *back-end* sendo responsável por intermediar a comunicação do UI com o sensor de nível armazenando as informações em um banco de dados.

*Front-end* fornecendo as informações de todo o sistema, o nível atual e todas as outras leituras armazenadas juntamente com o sistema de notificação em que uma vez acessado a UI são enviadas notificações de nível baixo do reservatório principal.

Tendo em mente que o sistema poderá ser utilizado por diversos usuários simultaneamente fez-se necessário a implementação de uma estrutura de cadastro e login de usuários para a aplicação.

O desenvolvimento do *back-end* foi realizado utilizando a arquitetura do NodeJs juntamente com *frameworks* para propiciar a implementação de um servidor CRUD e relacionamento com uma base de dados relacional denominada MySQL. Optou-se por uma estrutura de dados relacional pelo fato da existência do relacionamento entre o cadastro de usuário e as aquisições realizadas pelo sensor de nível. Para a implementação da base de dados que salvará as informações abstraiu-se a arquitetura apresentada na Figura 10.

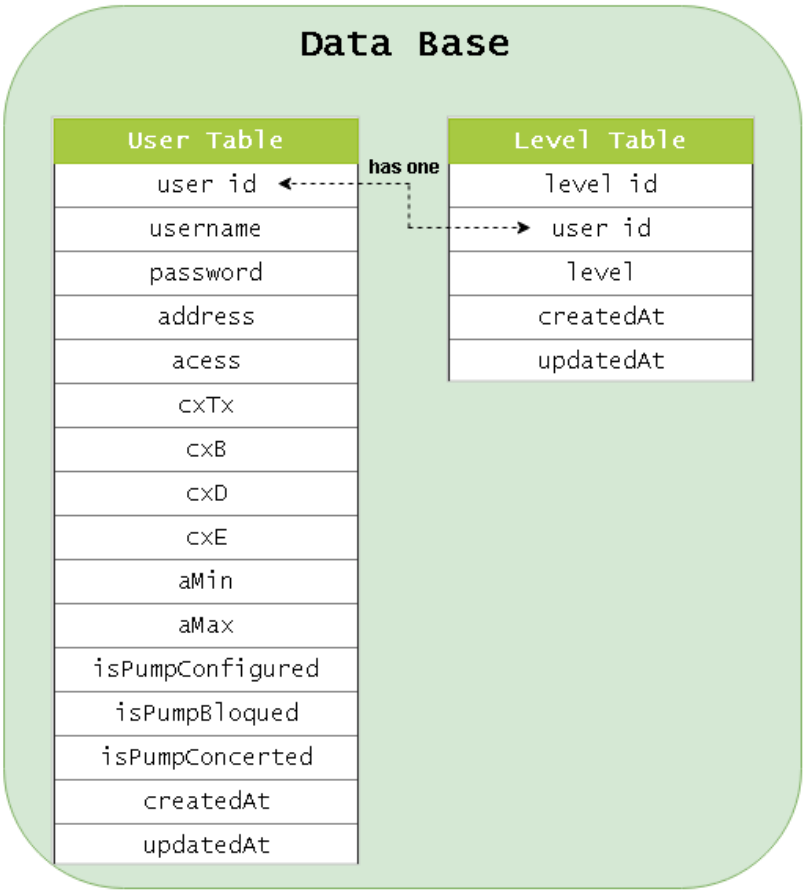



Figura 10 - Modelo das tabelas utilizadas na base de dados.

Desenvolveu-se o *front-end* utilizando uma biblioteca de componentização denominada ReactJs. A UI foi criada para oferecer aos usuários as funcionalidades de: registro com confirmação por e-mail, autenticação das informações do usuário e do sensor utilizado no monitoramento, amostra da última leitura e a data que foi realizada em um quadro de alertas, menu de configuração do sensor, previsão de tempo para a região de Formiga - MG, menu de configuração das informações pessoais do usuário e sistema de notificação em situação de nível abaixo de 40%.

Para hospedar a aplicação *web* optou-se por utilizar um sistema de máquina virtual viabilizando assim a configuração do certificado SSL, garantindo a integridade das informações trafegadas pela aplicação. Optou-se por contratar os serviços da Digital Ocean por ser uma empresa que está no mercado de hospedagem desde 2011 oferecendo preços acessíveis e uma ótima documentação.

Contemplando a confecção do sistema apresentado, utilizou-se os materiais abaixo listados.

Tabela 2 - Materiais utilizados na confecção do trabalho.

| **Item** |
|---|
| Sensor de nível automotivo |
| devKit ESP32  WROOM 32D |
| Fonte HLK PM03 110/220 to 3,3V |
| Relé SDR-03VDC-SL-C |
| BC 548 |
| Resistor 1K |
| Resistor 100 |
| Led 3mm |
| Butão tátil |
| Conector 2 terminais |
| Boia de nível elétrica |
| Bomba d’água externa Ferrari 0,5cv |
| Kit Acionamento WEG |
| Instância Digital Ocean |

# 4 RESULTADOS E DISCUSSÕES

Este capítulo tem a finalidade de apresentar os resultados obtidos no desenvolvimento do trabalho. Inicialmente aborda-se a construção do sensor de nível, sistema de reabastecimento e aplicação *web* e por fim apresenta-se o orçamento dos itens utilizados.

Durante a confecção do protótipo, foram constatadas variações abruptas na aquisição do nível, sendo ocasionadas pelas oscilações da superfície da água. Para solucionar a citada ocorrência, implementou-se uma rotina de leituras realizando-as cem vezes consecutivas calculando assim a média aritmética dessas. Com essa metodologia foi possível reduzir as variações ocasionadas pelo fluido, estabilizando assim as leituras. Além disso, para garantir um comportamento mais estável, implementou-se adicionalmente, um algoritmo de filtro digital passa baixo em cada leitura realizada otimizando ainda mais as aquisições de nível. A Figura 11 ilustra o protótipo utilizado para os testes.

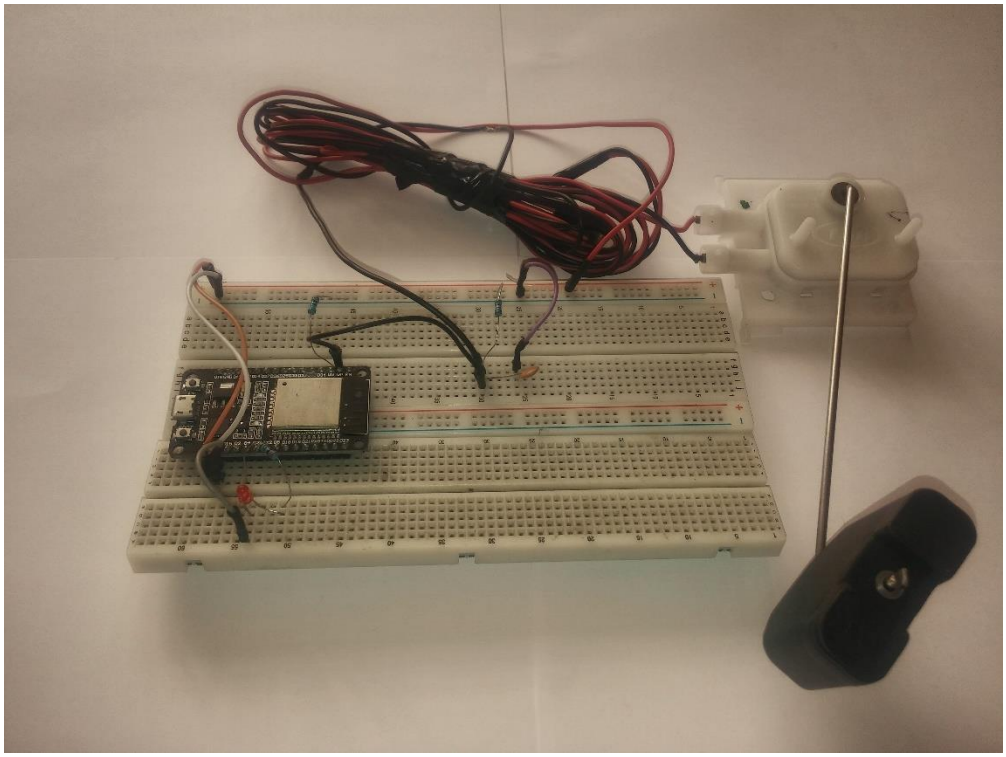
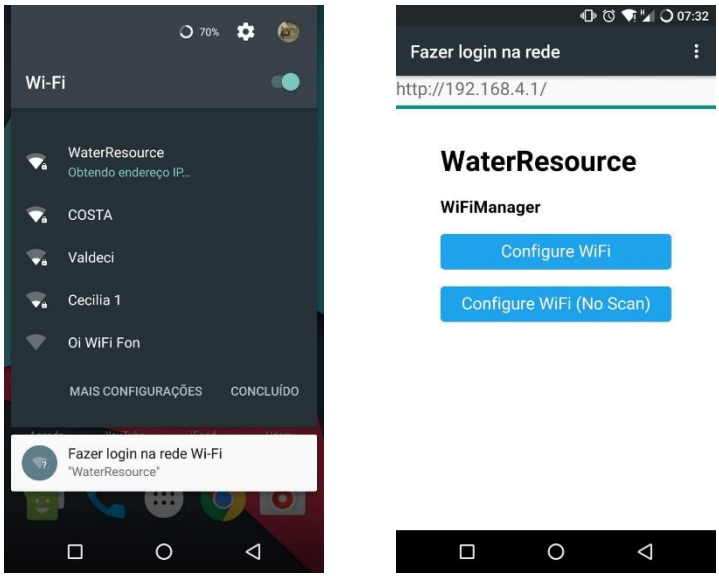
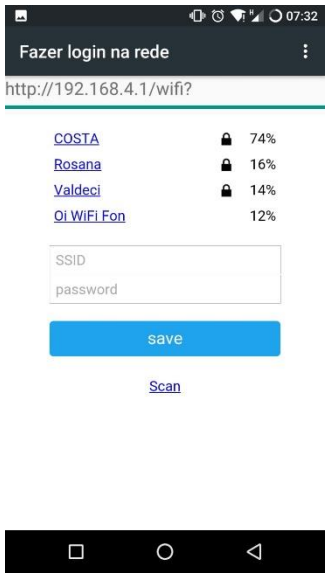


Figura 11 - Protótipo desenvolvido para teste.

Configurou-se a aplicação de Wi-Fi *Manager* para gerar uma rede local com o nome de WaterResource com a senha de acesso de 12345678, ilustrado na Figura 12 a). Ao conectar-se à rede a aplicação de gerenciamento é inicializada Figura 12 b). Quando selecionada a opção *Config WiFi* o gerenciador faz um *scan* de todos os *access points* disponíveis listando-os e fornecendo a capacidade de registrar o acesso aos mesmos Figura 12 c).



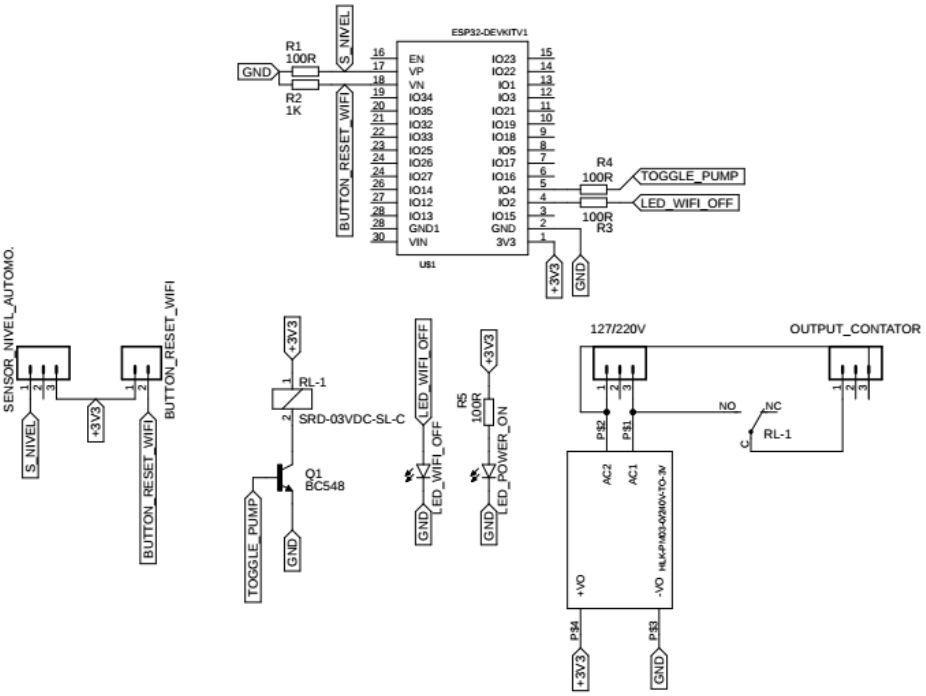

a) b)



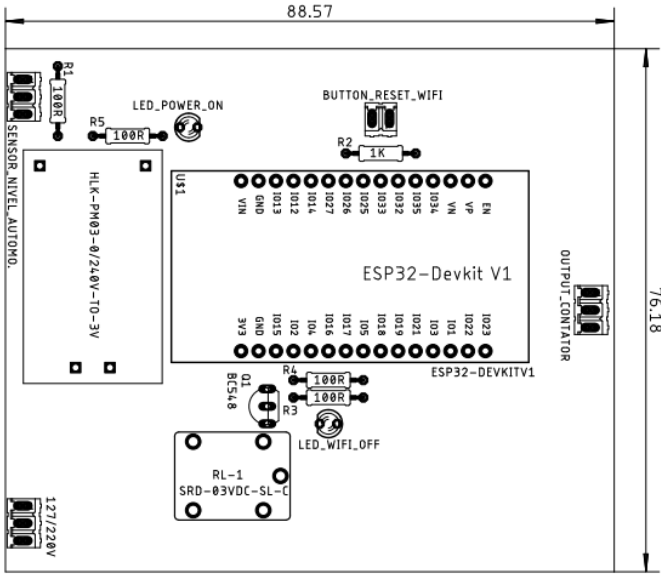

c)

Figura 12 - Wi-Fi Manager: a) Rede local; b) Acesso inicial a aplicação; c) Cadastro do *access point*.

Após a constatação do devido funcionamento do sensor de nível projetou-se o protótipo de *hardware* utilizando o *Eagle* apresentado na Figura 13.




a)




b)

Figura 13 – Protótipo de *Hardware*: a) Esquemático; b) Estampa superior, dimensões do tamanho estão na escala de milímetros.

Para o dimensionamento da bomba faz-se necessário o conhecimento da vazão do sistema. Para a obtenção desta grandeza analisou-se o manual da TIGRE (2019), obtendo uma vazão estimada para uma tubulação de ½", uma vez que tal tubulação é comumente utilizada nas instalações residenciais, para uma velocidade intermediária apresentada no ábaco do fabricante tem-se uma vazão aproximada de 0,0012 $m^3/s$, assumindo a altura de elevação entre os reservatórios de 10 m tem-se:

$$Pb = \frac{(\ 9{,}8\ x\ 0{,}0012\ x\ 1\ x\ 10\ )}{0{,}4} \approx 294\ W$$

Com o valor de potência calculado, pode-se extrapolá-lo para a primeira faixa de potência comercial de motores sendo 0,5 cv. Analisando o manual do fabricante de bombas Ferrari, uma bomba d'água externa com a potência que foi projetada fornece 22 metros de colunas de água (FERRARI, 2019). Desta forma a potência recomendada atende a especificação do projeto. Em seguida projetou-se o circuito de acionamento para o sistema de reabastecimento utilizando o software open source CADeSIMU, apresentado na Figura 14.

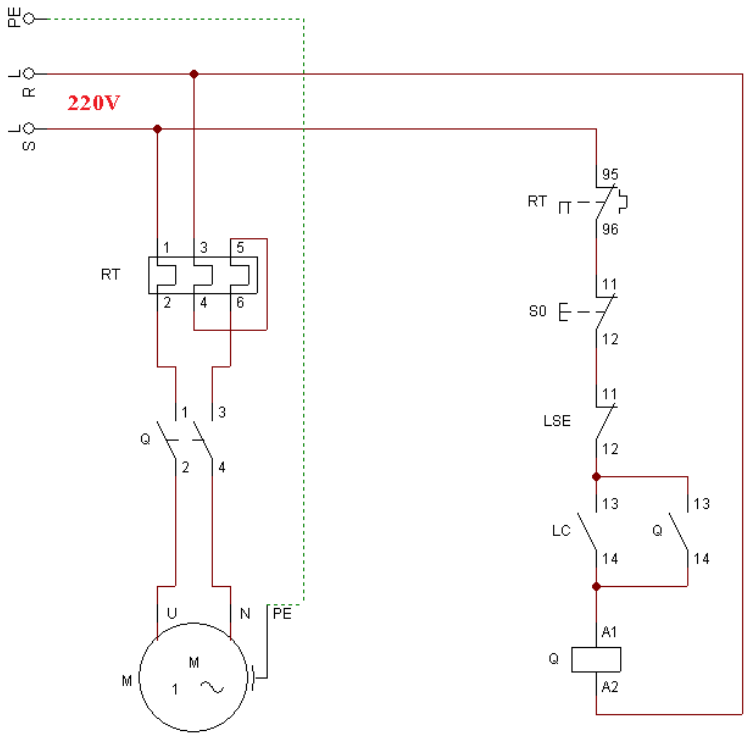



Figura 14 - Diagrama de acionamento do sistema de abastecimento.

Onde:

RT – Relé térmico;

Q – Contator;

M – Motor da bomba;

S0 – Botoeira de emergência;

LSE – Contato NF da bóia de nível do reservatório secundário;

LC – Acionamento da bomba proveniente do sensor de nível;

O diagrama apresentado é facilmente integrado ao kit de acionamento WEG sendo que o mesmo já possui um contator e um relé térmico. O circuito de potência presente no kit é semelhante ao proposto.

As Figuras a seguir ilustram toda a aplicação *web* desenvolvida.

| Nome | Tipo |
|---|---|
| id | int(11) |
| name | varchar(255) |
| email | varchar(255) |
| password | varchar(255) |
| address | varchar(255) |
| acesso | int(11) |
| aquisition | int(11) |
| cxTxt | varchar(255) |
| cxB | float |
| cxC | float |
| cxE | float |
| aMin | float |
| aMax | float |
| isPumpConfigured | int(11) |
| isPumpBlocked | int(11) |
| isPumpConcerted | int(11) |
| createdAt | datetime |
| updatedAt | datetime |

a)

| Nome | Tipo |
|---|---|
| id | int(11) |
| userId | int(11) |
| nivel | float |
| createdAt | datetime |
| updatedAt | datetime |

b)

Figura 15 - Banco de dados: a) Estrutura tabular para cadastro dos usuários; b) Estrutura tabular para registro do nível.


Figura 16 - Interface do usuário contendo a caixa de alertas, gráfico do nível atual e previsão do tempo.

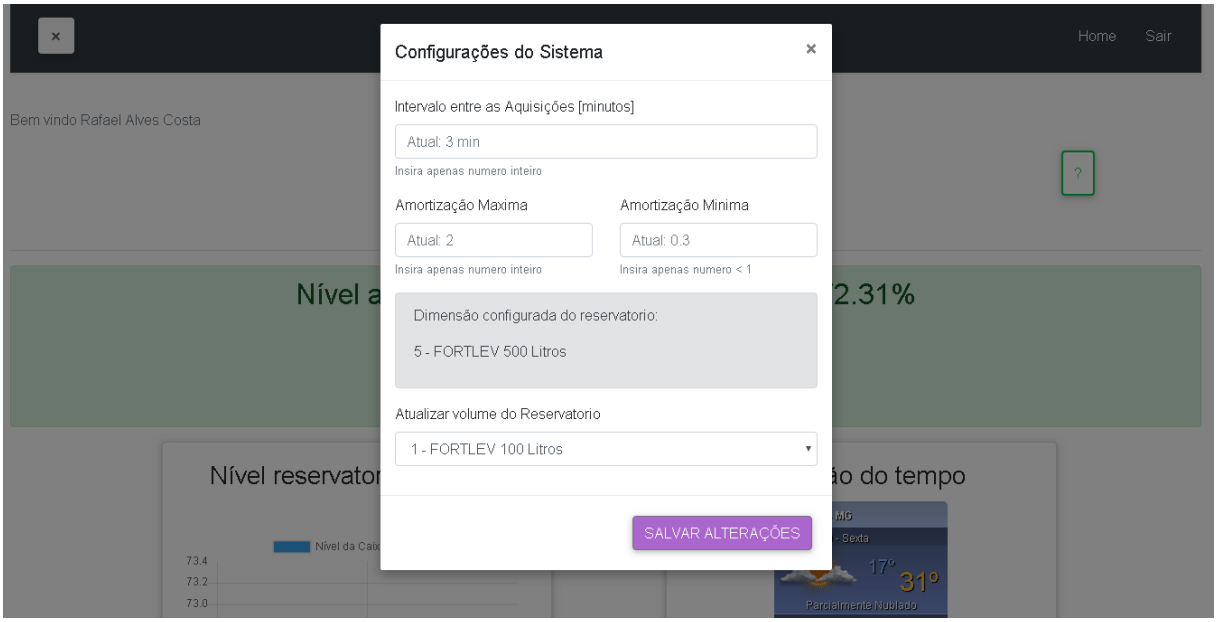



Figura 17 - Menu de configurações do sensor.

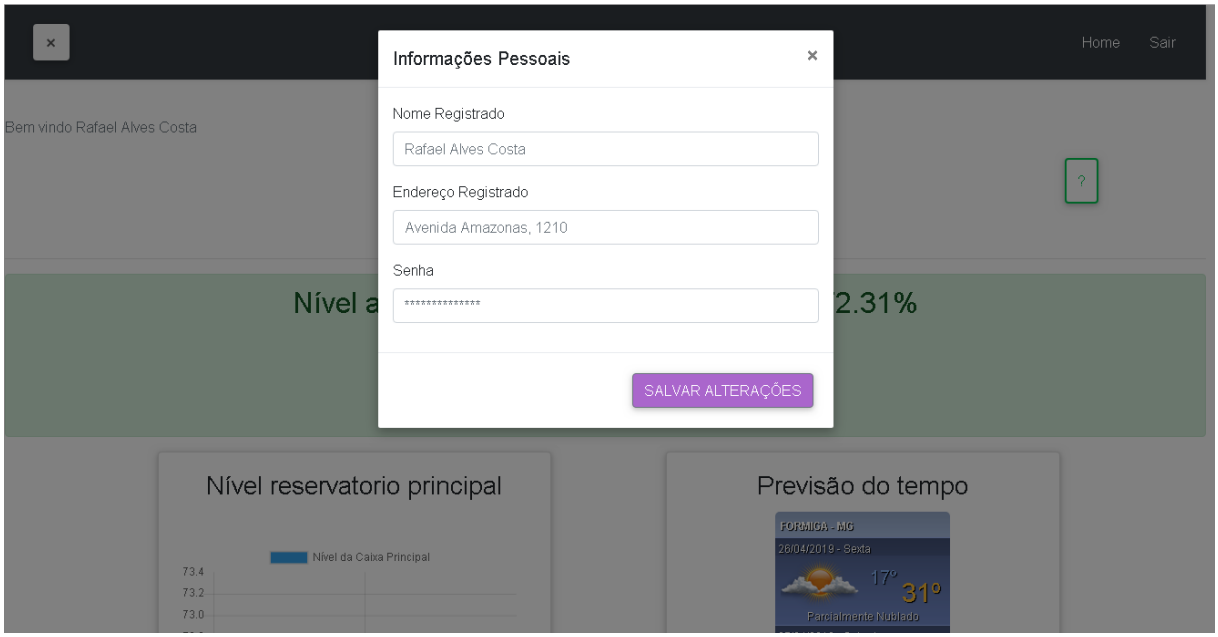



Figura 18 - Menu de configurações pessoais do usuário.

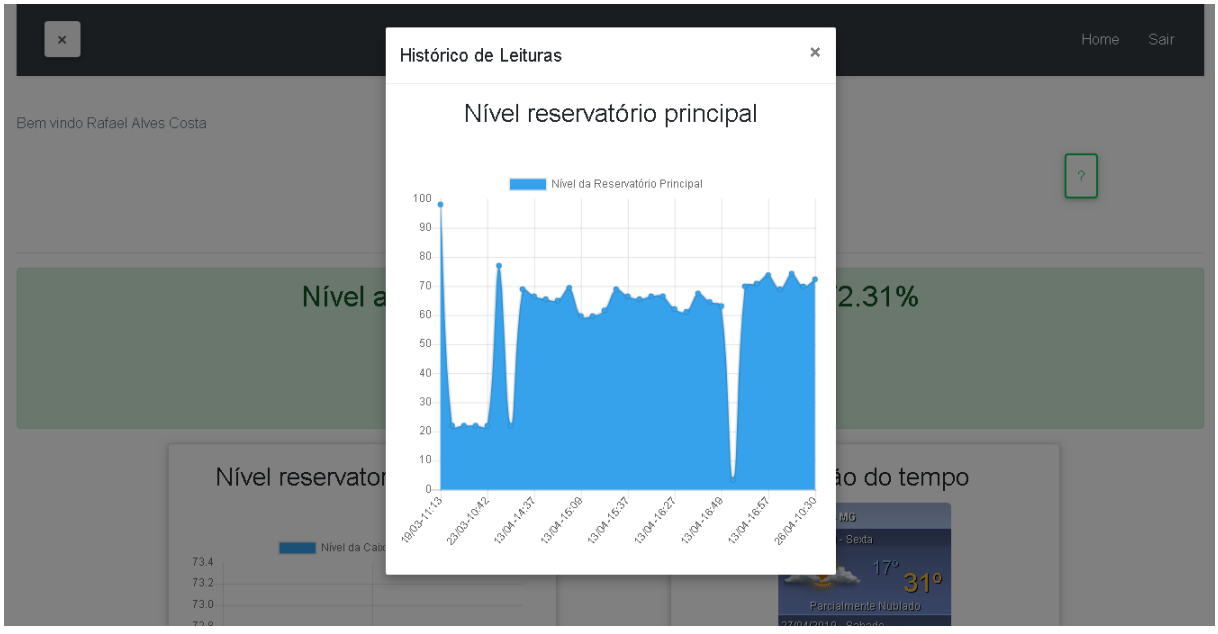



Figura 19 - Gráfico de todas as leituras realizadas

Durante a confecção da aplicação *web* foi estabelecido o menor tempo de aquisição, sendo três minutos, esse tempo foi determinado pelo fato de prevenir a utilização desnecessária de recursos do sensor sendo que o tempo inferior ao estipulado a variação de nível será mínima.

Após a confecção do *web site* realizou-se testes para constatar a capacidade de acessos simultâneos suportada pelo mesmo, utilizando a *framework Artillery* (ARTILLERY, 2019). É possível simular acessos simultâneos podendo então quantificar a capacidade de acessos simultâneos suportada. Inicialmente simulou-se cem acessos simultâneos à UI obtendo resultado de 100% de sucesso das requisições realizadas. Ao simular cento e cinquenta acessos simultâneos o resultado foi que 98% das requisições foram bem-sucedidas enquanto 2% ocorreram algum tipo de falha. Por fim quando simulado trezentos acessos simultâneos obteve apenas 62% de sucesso nas requisições, desta forma constatou-se que o sistema desenvolvido suporta cem usuários conectados simultaneamente utilizando os recursos da UI.

Para estipular o espaço de armazenamento que a aplicação consumirá ao longo de sua utilização assume-se que a aplicação conterá a quantidade máxima de cem usuários registrados tendo em vista que é a quantidade limite suportada de acessos simultâneos. Cada registro de usuário ocupa um espaço de armazenamento no banco de dados de 32 KB desta forma para cem usuários registrados tem-se o consumo de 3,2 MB. Cada registro de nível salvo ocupa 300 Bytes assumindo que será configurado o tempo de aquisição do sensor para o menor possível, em um dia será armazenado 144 KB por sensor de nível. Considerando a quantidade máxima de usuário por dia resultará em 14,4 MB de dados salvos, tendo que o espaço de armazenamento contratado para a aplicação é de 25 GB a aplicação suportará armazenar os dados de aquisição por um período de 4 anos e 8 meses.

A Tabela 3 apresenta o orçamento para a implementação de todo o sistema apresentado no trabalho, tendo em mente que a cotação foi realizada no dia 03 de Maio de 2019 com o valor do dólar a R$ 3,67.

Tabela 3 - Materiais utilizados na confecção do trabalho.

| Quantidade | Item | Valor [U$] |
|---|---|---|
| 1 | Sensor de nível automotivo | 14,87 |
| 1 | devKit ESP32 WROOM 32D | 10,08 |
| 1 | Fonte HLK PM03 110/220 to 3,3V | 6,02 |
| 1 | Relé SDR-03VDC-SL-C | 3,50 |
| 1 | BC 548 | 0,25 |
| 1 | Resistor 1K | 0,03 |
| 3 | Resistor 100 | 0,1 |
| 2 | Led 3mm | 0,29 |
| 1 | Butão tátil | 0,33 |
| 3 | Conector 2 terminais | 1,36 |
| 1 | Boia de nível elétrica | 9,07 |
| 1 | Bomba d'água externa Ferrari 0,5cv | 37,81 |
| 1 | Kit Acionamento WEG | 40,33 |
| Um ano | Instância Digital Ocean | 66,04 |
| **TOTAL** | - | 190,08 |

Pela falta de recursos durante o desenvolvimento do projeto não foi possível confeccionar o sistema de reabastecimento nem a *hardware*. No entanto, todo o sistema foi devidamente testado e validado o funcionamento do conjunto como um todo, ficando então a confecção para uma projeção futura de outros trabalhos.

# 5 CONDISERAÇÕES FINAIS

O objetivo do trabalho desenvolvido foi aprimorar o sistema apresentado por Lopes (2018) em que foi desenvolvido um sistema que utiliza dois sensores de nível, para redundância, um sensor ultrassônico e um sensor de combustível. No sistema citado, a conexão com a internet ocorre com cabos de rede, utilizando um Arduino e um módulo *Ethernet*.

Inicialmente, retirou-se o sistema de redundância de medição por meio de dois sensores, adotando-se, então, apenas o sensor de nível automotivo. A conexão à internet foi mudada para conexão por meio de redes sem fio, simplificando e reduzindo custos de instalação do sistema.

Constatou-se que o funcionamento do sensor de nível foi condizente com o esperado, obtendo-se aquisições de nível conforme o tempo programado e enviando-as ao servidor *web*. A utilização de somente o sensor de nível automotivo trouxe ao sistema uma imprecisão na leitura dos níveis, uma vez que a variação da resistência interna do sensor apresenta uma não linearidade; entretanto, tal variação não prejudica o sistema de forma comprometedora com a proposta de estimativa do nível.

A confecção da aplicação *web* de forma descentralizada viabilizou a construção da UI no formato de *Web App,* ou seja, um site que responde ao tamanho da tela do dispositivo que o acessa, assim fez com que não houvesse a necessidade de construir um aplicativo para dispositivos móveis. Além desse fato a arquitetura construída apresenta fácil manutenção por ser uma estrutura desenvolvida no formato de componentes. A limitação de tempo de funcionamento da aplicação em função do espaço de armazenamento pode ser solucionada implementando um evento de gerenciamento dos dados persistidos no banco de dados. Sabendo-se que o foco do sistema é o monitoramento em tempo real, logo, não existe a necessidade de persistir as leituras de níveis por um período muito longo, desta forma configurando um evento que mantém as leituras salvas durante o prazo de um mês, o tempo máximo de utilização por falta de espaço de armazenamento não existirá mais. O valor orçado para manter o site online pelo prazo de um ano é facilmente recuperado dividindo esse valor para a quantidade de usuários que o sistema suporta, desta forma pode-se extrapolar o orçamento para um tempo maior mantendo assim a aplicação *online* durantes vários anos.

O sistema de reabastecimento projetado contempla medidas de segurança para garantir o perfeito funcionamento do sistema sem danificar os equipamentos e a instalação do cliente. Juntamente com o *hardware* viabiliza uma fácil instalação no sistema de abastecimento já existente nas residências, tendo um custo acessível, propiciando o desenvolvimento de um protótipo comercial.

Desta forma infere-se que o objetivo apresentado neste trabalho foi concluído com êxito, pois o sistema apresentou o funcionamento desejado com todas as funcionalidades dimensionadas. O fato de não ter construído o *hardware* e o sistema de reabastecimento para a constatação da longevidade do sistema e seu comportamento quando submetido à situações de estresses, não invalida o mesmo, uma vez que foge do escopo deste trabalho. Estes testes podem ser realizados em trabalhos futuros.

Todo o conteúdo desenvolvido no trabalho está disponível de maneira *Open Source* no *GitHub* < https://github.com/rafaelbodao/waterresource>.

# 6 TRABALHOS FUTUROS

A partir deste trabalho, pode-se sugerir como temas para trabalhos futuros:

- Confecção do protótipo de hardware e melhora no mesmo desenvolvendo assim um *hardware* baseado no MCU ESP32.
- Confecção do sistema de reabastecimento implementando a bomba juntamente com o circuito de acionamento.
- Otimização na aplicação *web* para que a mesma suporte uma quantidade maior de usuários simultâneos.
- Implementação do sistema em uma residência para validação e constatação do funcionamento do mesmo.
- Desenvolvimento de um sensor de nível utilizando como base um potenciômetro que tenha uma resposta linear à variação de nível.

# 7 REFERÊNCIAS

AGÊNCIA NACIONAL DE ÁGUAS. **Quantidade de água**. Disponível em <http://www3.ana.gov.br/portal/ANA/panorama-das-aguas/quantidade-da-agua>. Acesso em: 18 Mai, 2019.

APACHE. **Apache**. Disponível em: <https://www.apache.org/>. Acesso em: 15 Abr, 2019.

ARTILLERY. **Artillery.io**. Disponível em: <https://artillery.io/>. Acesso em: 06 Mai, 2019.

CASSIOLATO, César. **Medição de Nível.** Disponível em: < http://www.smar.com/newsletter /marketing/index39.html>. Acesso em: 12 Abr, 2019.

DIGITAL OCEAN. **Digital Ocean**. Disponível em: < https://www.digitalocean.com/>. Acesso em: 12 Abr, 2019.

ESPRESSIFI. **ESP 32 a different IoT power and performance.** Disponível em: < https:// www.espressif.com/en/products/hardware/esp32/overview >. Acesso em: 12 Abr, 2019.

FERRARI. **Bombas Periféricas**. Disponível em: <https://www.ferrarinet.com.br/aab 1010009>. Acesso em: 22 Abr, 2019.

G1. **Crise hídrica no Brasil será destaque no 2º dia do Fórum Mundial da Água**. Disponível em: <https://g1.globo.com/df/distrito-federal/forum-mundial-daagua/2018/noticia/crise-hidrica-no-brasil-sera-destaque-no-2-dia-do-forum-mundial-da-agua.ghtml>. Acesso em: 30 Mar, 2019.

LET'S ENCRYPT. **Let's Encrypt**. Disponível em: < https://letsencrypt.org/>. Acesso em: 12 Abr, 2019.

LOPES, G. V. S. **Sistema microcontrolado para monitoramento remoto de nível em reservatórios d´água.** 2018. 88 p. Monografia (Bacharelado em Engenharia Elétrica) – INSTITUTO FEDERAL DE MINAS GERAIS – CAMPUS FORMIGA, Formiga, 2018

MAMEDE, João Filho. **Instalações elétricas industriais.** 8ed. Rio de Janeiro. LTC, 2010. 666p.

MIXAUTO. **Sensor de nível de combustível.** Disponível em: < https://www.mixauto.com.br/ sensor-do-nivel-de-combustivel-citroen-c3-picasso-aircross-11-novo-c3-13-tsa-010218-p3037/>. Acesso em: 12 Abr, 2019.

MOCIUM, Wiktor. ***Big Names using React.js***. Disponível em: < https://reactkungfu.com/2015/07/big-names-using-react-js/>. Acesso em: 30 Abr, 2019.

NGINX. **Nginx**. Disponível em: <https://www.nginx.com/>. Acesso em: 12 Abr, 2019.

NODE. **NodeJs**. Disponível em: < https://nodejs.org/api/esm.html>. Acesso em: 15 Abr, 2019.

OLIVEIRA, I. R. H. **Desenvolvimento de um sistema para monitoramento remoto do nível de água de reservatórios residenciais.** 86 p. Monografia (Bacharelado em Engenharia Elétrica) – INSTITUTO FEDERAL DE MINAS GERAIS – CAMPUS FORMIGA, Formiga, 2015.

OLIVEIRA, I. R. H. MARCO, A. L. R.; SANTOS, C. R. B. **Desenvolvimento de um aplicativo Android para monitoramento microcontrolado do nível de um reservatório de água residencial em tempo real**. XII CEEL, outubro de, 2014.

OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry Pi.** 1ed. São Paulo. Editora Novatec Ltda, 2017. 240p.

PEREIRA, Fábio. **Microcontroladores PIC técnicas avançadas.** 1 ed. São Paulo. Editora ERICA, 2002. 358p.

PYTHON. **Python**. Disponível em: < https://www.python.org/>. Acesso em: 15 Abr, 2019.

REACT. **ReactJs**. Disponível em: < https://reactjs.org/>. Acesso em: 30 Abr, 2019.

RFC 2818. **HTTP Over TLS.** Disponível em <https://tools.ietf.org/html/rfc2818>. Acesso em: 14 Jun, 2019

RFC 6455. **The WebSocket Protocol.** Disponível em <https://tools.ietf.org/html/rfc6455>. Acesso em: 14 Jun, 2019

RFC 7231. **Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content.** Disponível em <https://tools.ietf.org/html/rfc7231>. Acesso em: 14 Jun, 2019

ROECKER, Max Naegeler. **Introdução a Node.js**. Disponível em: <https://maxroecker.github.io/blog/introducao-ao-nodejs/index.html>. Acesso em: 15 Abr, 2019.

SANTOS, B. P, et al. **Internet das Coisas: da Teoria à Prática.** 2017. 50 p. (Departamento de Ciências da Computação) – UFMG – Belo Horizonte – Minas Gerais, 2017. Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>. Acesso em: 10 Abr, 2019.

SARTORI, Gustavo. **Node.js quem já utiliza a tecnologia (e aprova)?**. Disponível em: <https://blog.umbler.com/br/node-js-quem-ja-utiliza-a-tecnologia-e-aprova/>. Acesso em: 15 Abr, 2019.

SEQUELIZE. **Associations**. Disponível em: <http://docs.sequelizejs.com/manual/associations.html#basic-concepts>. Acesso em: 15 Abr, 2019.

SIMON, Imre. **A ARPANET.** Disponível em: <https://www.ime.usp.br/~is/abc/abc/node20.html>. Acesso em: 30 Mar, 2019.

TIGRE. **Orientação para instalações de Água Fria**. Disponível em: <https://www.tigre.com.br/themes/tigre2016/downloads/catalogos-tecnicos/ct-agua-fria.pdf>. Acesso em: 22 Abr, 2019.

TRIERWEILER, Jorge. **Fluxograma de Engenharia (P&I-Diagram)**. Disponível em: < http://www.producao.ufrgs.br/arquivos/disciplinas/492_pei_3.pdf>. Acesso em: 01 Mai, 2019.

WEG. **Chaves de partidas em caixas plásticas (direta, reversa e estrela-triangulo)**. Disponível em: <https://www.weg.net/catalog/weg/BR/pt/Automa%C3%A7%C3%A3o-e-Controle-Industrial/Controls/Partida-e-Prote%C3%A7%C3%A3o-de-Motores/Chaves-de-Partida-em-Caixa/Chave-de-Partida-em-Caixa-Pl%C3%A1stica-%28Direta%2C-Reversora-e-Estrela-Tri%C3%A2ngulo%29/c/BR_WDC_IA_CTL_MPC_ES_ENCLOSEDSTARTERSPLASTIC?h=d2993794>, Acesso em: 01 Mai, 2019.