

# Uso de IoT no Monitoramento de Reservatórios de Água

João Porto<sup>1</sup>, Mateus Ceccagno<sup>1</sup>, Robson Costa<sup>1</sup>

<sup>1</sup>Instituto Federal de Santa Catarina (IFSC)

Rua Heitor Vila Lobos, 225 - São Francisco - 88506-400 - Lages - SC - Brasil

{joao.porto47, mateus.cecc}@gmail.com, robson.costa@ifsc.edu.br

**Abstract.** *The Internet of Things (IoT) is expanding greatly in computing, bringing technology to all areas of society, increasing efficiency, productivity and security in everyday processes. The aim of this work was to propose a solution based on IoT architecture to monitor the water reservoirs at the Instituto Federal de Santa Catarina Campus Lages, thus making it possible to keep track of their water levels and flows. Microcontrollers and pressure transducers were used, as well as a platform for managing IoT devices and visualizing data via dashboards. The results showed that their use allowed the team to monitor operations more safely, avoiding access to high roasts, as well as greater sustainability, allowing any leaks to be identified quickly.*

**Resumo.** *A Internet das Coisas (IoT) tem sido um campo de grande crescimento na computação, trazendo tecnologia para todas as áreas da sociedade, aumentando a eficiência, a produtividade e a segurança em processos habituais. Este trabalho teve como objetivo propor uma solução baseada na arquitetura IoT para realizar o monitoramento dos reservatórios de água do Instituto Federal de Santa Catarina Câmpus Lages, permitindo assim acompanhar os níveis e fluxos de água dos mesmos. Microcontroladores e transdutores de pressão foram utilizados, bem como uma plataforma de gestão de dispositivos IoT e visualização de dados através de dashboards. Os resultados demonstraram que o seu uso permitiu uma maior segurança nas operações de monitoramento por parte da equipe, evitando o acesso em torra altas, bem como uma maior sustentabilidade permitindo que rapidamente sejam identificados eventuais vazamentos.*

## 1. Introdução

A ideia de conectar dispositivos entre si utilizando redes sem fio e sensores eletrônicos tem sido um grande avanço na sociedade em geral. Ideia essa que cresceu e contribuiu com a quarta Revolução Industrial. O termo Indústria 4.0, criado na Alemanha em 2011 (Bassi, 2017), resume-se ao uso integrado de Internet das Coisas, 5G e *Big Data* com a indústria da transformação, fazendo com que a digitalização do processo industrial facilite o desempenho, trazendo velocidade e economia, tal qual, maior controle do que é produzido e utilizado (máquinas, matéria-prima, etc) (Chen e Zhou, 2018). Desse modo tal evolução não vem apenas mudando o mundo empresarial, mas também cidades e residências.

O conceito de *Smart Cities* na década de 2020 é visto de duas formas pela comunidade de pesquisadores, alguns dizem que o avanço tecnológico possibilita a sustentabilidade dessas cidades e outros que tais avanços precisam se tornar mais presentes no

cotidiano das pessoas, assim possibilitando a chegada em nossas casas, veículos, móveis, objetos e entre outros itens (Founoun et al., 2019).

Esse avanço possibilitou ampliar a pesquisa e popularidade do conceito de Internet das Coisas (IoT, do inglês *Internet of Things*). “*Data is the new oil*”, frase dita pelo matemático Clive Humby em 2006, refere-se ao valor e importância que os dados estão recebendo no Século XXI. Empresas, instituições, cientistas, buscam extraí-los em todos os processos atribuídos em seus produtos, pesquisas, consumidores, mas, tal qual o petróleo, dados precisam ser processados e assim transformados em informação, em conhecimento permitindo assim o aumento da produtividade, velocidade e lucratividade. Em paralelo computadores, redes e algoritmos tornam-se cada dia mais rápidos (Provost e Fawcett, 2013). Essa corrida gera competitividade e aumenta o investimento em pesquisa não só na ciência de dados, mas em toda a esfera tecnológica.

Espera-se que até 2025 o número de objetos conectados entre si seja em torno de 50 bilhões gerando cerca de 4,5 milhões de empregos (Angrisani et al., 2018). IoT é a referência do poder da computação de integrar humanos com dispositivos inteligentes, tornando o manual em automático.

Aplicações que utilizam IoT, têm como foco obtenção de dados analíticos ou até mesmo o monitoramento de sensores e equipamentos implantados em indústrias, cidades, casas, etc. Tais aplicações podem ser sistemas autônomos e inteligentes.

No contexto de aplicações hídras baseadas em soluções IoT, destacam-se a automatização do funcionamento de bomba de água com base no fluxo e no nível de água. Esta solução é composta pela integração de um microcontrolador com sensores e atuadores.

A preocupação por melhoria na gestão hídrica é um interesse de contexto mundial, como pode ser observado em 2015, ano que a Organização das Nações Unidas (ONU) publicou a Agenda 2030, composta de 17 Objetivos do Desenvolvimento Sustentável (ODS). Água potável e saneamento é o objetivo de número 6 (seis), o qual é definido por metas que destacam a necessidade de melhoria nos processos de gestão da água e de saneamento, ademais, esse conjunto de metas se destinam a todas as pessoas e a todos os setores (United Nations e Development, 2015). Analisando esse prospecto em detalhes, destaca-se o monitoramento como uma das importantes etapas da gestão hídrica, a qual permite agir de forma preventiva, ou seja, detectar problemas precocemente. Diante disso, a Internet das Coisas pode exercer o papel de monitoramento de água em reservatórios, visto que, a IoT abrange assuntos relativos à coleta, transmissão e análise de dados. Sendo essas, características que possibilitam a automação para emissão de alertas e geração de relatórios sobre o que está sendo monitorado.

Além de contribuir para o não desperdício de água, o uso de IoT neste tipo de aplicação, em alguns casos, também permite reduzir o risco de acidentes de trabalho em altura. No caso específico dos campus do Instituto Federal de Santa Catarina (IFSC), existem reservatórios verticais com grande altitude que necessitam de manutenção, sendo necessário uma pessoa com certificação na Norma Regulamentadora 35 (NR 35).

Diante do exposto, através da Tecnologia da Informação e Comunicação (TIC), é possível desenvolver uma ferramenta que contribua para a redução de acidentes de trabalho em altura. Uma das maneiras de alcançar isso é com o uso de sensoriamento e

uma interface humano-computador que apresente os dados sobre os reservatórios, assim reduzindo o número de vezes que uma pessoa precisa subir em um reservatório para verificar manualmente/visualmente o nível de água. Ainda fruto desse modelo de ferramenta, outro benefício é o acréscimo de tempo para concertar algum cenário indesejado nos reservatórios, um caso de uso hipotético de cenário indesejado seria o volume de água de um determinado reservatório estar inadequado. Neste caso, alertas seriam emitidos para que o funcionário competente fique ciente de que é necessário agir o quanto antes para reverter essa situação e impedir a escassez de água.

Servindo-se dos casos de uso expostos, incluindo problemas e possíveis soluções apresentadas, o presente trabalho se dispõe a construir uma aplicação que forneça alertas especializados, gráficos com informações sobre níveis de água em reservatórios e outras funcionalidades que serão detalhadas na seção 3. Para o planejamento e construção do sistema foi levado em consideração o conceito de *Smart Campus*, que objetiva o uso de tecnologias de maneira integrada e eficiente, permitindo que os sistemas, dispositivos e pessoas interajam entre si, a principal contribuição do desenvolvimento do campus inteligente é facilitar a vida em todos os aspectos do campus ??.

Dessa forma, o presente trabalho tem por objetivo principal desenvolver uma solução de monitoramento dos reservatórios de água do câmpus Lages do IFSC a qual deva ser baseada no conceito de Internet das Coisas. De forma a alcançar este objetivo, os seguintes objetivos específicos foram estabelecidos:

- Realizar o estudo da infraestrutura hídrica do câmpus;
- Fazer o levantamento dos requisitos funcionais e não funcionais do sistema;
- Estudar as diversas tecnologias existentes para selecionar a mais adequada;
- Desenvolver o sistema de monitoramento de recursos hídricos;
- Realizar testes e eventuais ajustes necessários.

A metodologia utilizada neste trabalho lançou mão de um estudo transversal, exploratório e descritivo com uma abordagem mista (quantitativa e qualitativa), dividida em 4 (quatro) etapas (Fortin et al., 2009). Na primeira etapa foi realizada uma análise técnica da infraestrutura de abastecimento e armazenamento de água do IFSC/Lages. A segunda etapa consistiu no levantamento de requisitos necessários para o desenvolvimento de um sistema de monitoramento da infraestrutura analisada na fase anterior. A terceira etapa consistiu no desenvolvimento da solução IoT, passando pela construção de um *hardware* e o desenvolvimento do seu respectivo *firmware*, instalação e configuração de uma plataforma de gestão IoT e desenvolvimento de um aplicativo móvel para interação com o sistema. Por fim, a quarta e última etapa consistiu na realização de testes e eventuais ajustes que fizeram-se necessários, no levantamento de resultados quantitativos e qualitativos, bem como na análise, comparação e compreensão dos resultados obtidos.

Este documento está organizado com a seguinte estrutura: na seção 2 é apresentado o referencial teórico, na seção 3 é descrita a solução proposta; na seção 4 são descritos os testes e resultados obtidos; ao final, na seção 5 são apresentadas as conclusões e propostas de trabalhos futuros.

## **2. Referencial Teórico**

Nesta seção são apresentadas os principais conceitos relacionados com soluções IoT.

## 2.1. Indústria 4.0

A indústria passou por diversos saltos tecnológicos no decorrer na história humana (Figura 1). A primeira revolução industrial aconteceu na Inglaterra no final do século XVIII e início do século XIX, entre 1760 e 1860, quando houve a mecanização dos processos e o uso de da energia gerada pelo carvão em máquinas à vapor, sendo esta um dos marcos do início do capitalismo (Sakurai e Zuchi, 2018).

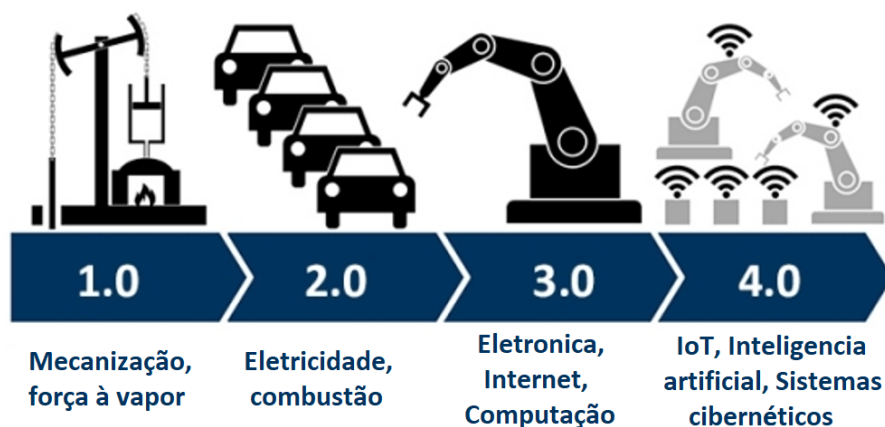


Figura 1. Revoluções industriais (Gislam, 2021).

Com o advento e a utilização intensiva da energia elétrica, iniciou-se em 1870 a segunda revolução industrial. Indústrias com foco no aumento da lucratividade, desenvolvimento do aço e da química. Como grande marco desta podemos citar o Fordismo, a introdução de uma esteira rolante, fazendo com que o trabalhador ficasse em posição fixa, aumentando muito a produtividade (Ribeiro, 2015).

Tais revoluções mudaram muito o estilo de vida, o capitalismo cresceu, houve alta na produtividade e um aumento populacional global. Isso trouxe uma aumento das necessidades, até mesmo, outras foram criadas, tornando assim necessária a utilização de tipos de energia alternativas. Tecnologias foram aprimoradas, como a robótica, transportes, química, telecomunicações, e a informática em geral, caracterizando assim a terceira revolução industrial, entre o século XX e XXI.

O aumento da digitalização industrial em conjunto com a Internet e sua conectividade com objetos inteligentes, está tornando os processos integrados em todo o ambiente manufatureiro. Tais mudanças já são vistas como um marco na produção industrial, customizações modulares e informações em tempo real, integrando matéria-prima, máquinas, operadores e gestores nos diferentes processos industriais. Evoluções estas que a partir de 2011 foram caracterizadas pelo governo da Alemanha como a Indústria 4.0 (Lasi et al., 2014).

## 2.2. Cidades Inteligentes (*Smart Cities*)

O conceito de Cidades Inteligentes (*Smart Cities*) trabalha com a conectividade de regras, processos e dispositivos a partir de múltiplas redes das quais informam dados sobre o fluxo de decisões de pessoas e materiais da cidade (Batty et al., 2012). Desde fluxo de celulares parados em uma avenida totalizando informações de um engarrafamento, ou, carnês de IPTU serem disponibilizados e pagos de forma *online*.

Tal conceito de inteligência não se resume em apenas extrair tais dados, e sim, entender como essas informações podem melhorar questões sociais e econômicas, trazendo eficiência em diferentes áreas. Conforme a Figura 2 demonstra, diversos escopos transformam uma cidade em uma cidade inteligente: 1) Sustentabilidade: exemplo coleta de lixo programada, bem como, sua separação em resíduos e recicláveis; 2) Locomoção, como dito anteriormente na questão de tráfego, ou, com pontos de ônibus programados para informar a existência ou não de passageiros; 3) Saúde, integralizando o atendimento ao número de leitos e níveis de traumas; 4) Indústria, trazendo tecnologia a processos do dia a dia, integrando dados que trazem mais segurança e eficiência para a produção; 5) Governança, facilitando a comunidade se comunicar com seus gestores, opinando e enviando informações da cidade; 6) Infraestrutura, englobando todos os pontos anteriores *Smart Cities* trazem uma infraestrutura integrativa, juntando todos os escopos de uma sociedade sustentável e tecnológica. Entre tantos outros pontos importantes em uma cidade, assim aumentando a qualidade de vida de seus habitantes.

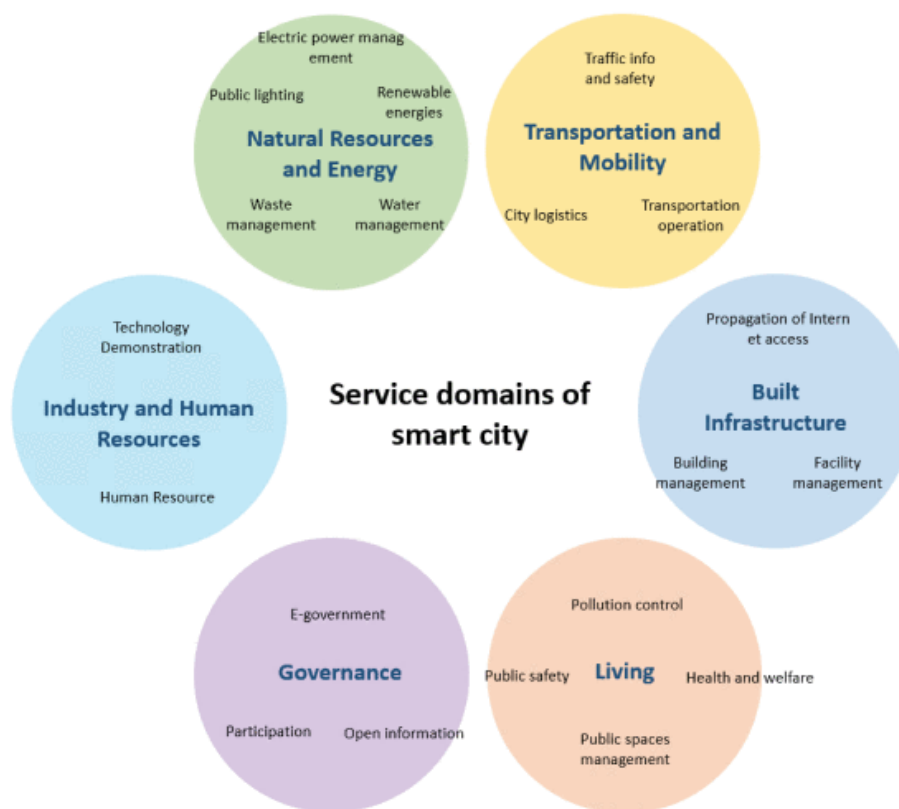


Figura 2. Domínios de serviço de uma cidade inteligente (Yang et al., 2021).

### 2.3. Câmpus Inteligente (*Smart Campus*)

O conceito de Câmpus Inteligente (*Smart Campus*) se refere à um ambiente que utiliza a Tecnologia da Informação e Comunicação (TIC) para oferecer serviços em tempo hábil ou para reduzir o esforço e custos operacionais dos serviços já ofertados (Figura 3). Câmpus Inteligente contém soluções de IoT que podem ser aplicadas em várias áreas, incluindo educação, eficiência energética, segurança, transporte e saúde (Abuarqoub et al., 2017) que melhoram a experiência dos indivíduos que possuem alguma relação com o câmpus em questão.

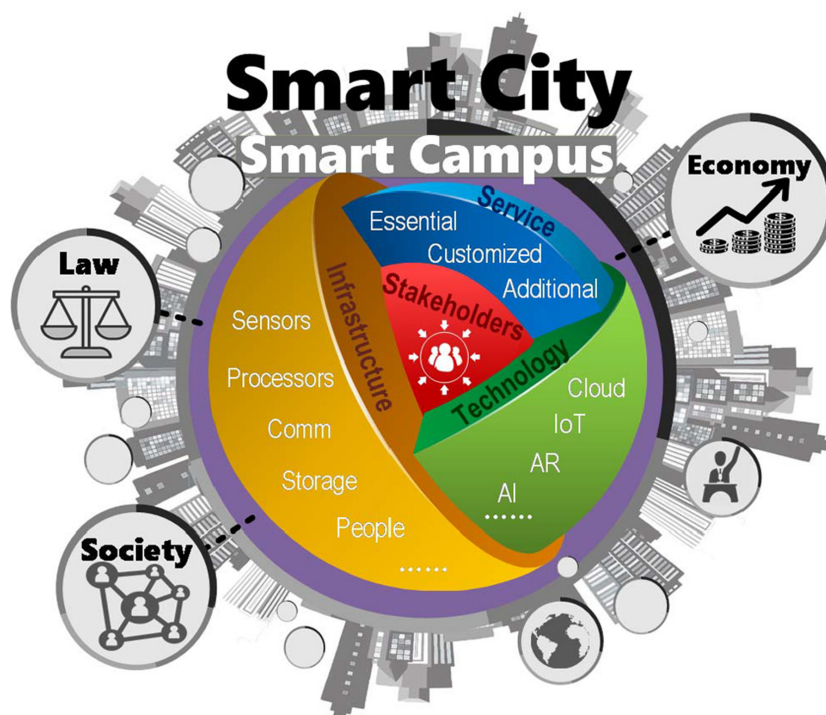


Figura 3. Relação *Smart Campus* e *Smart City* (Dong et al., 2020).

As TIC são essenciais para a troca de dados de medição e comandos de controle, permitindo também a informação e a integração das pessoas. Especialmente o conceito IoT permite o desenvolvimento de dispositivos de comunicação e possibilita atividades no monitoramento e controle em infraestruturas (Botta et al., 2014). Por exemplo, sensores em combinação com comunicação com ou sem fio podem ser facilmente instaladas mesmo em estruturas remotas e subterrâneas, suportando assim uma implementação até mesmo em larga escala de equipamentos de medição (Atzori et al., 2010).

#### 2.4. Sistema de gestão hídrica inteligente (*Smart Water*)

Um sistema de gestão hídrica inteligente (*Smart Water*) é projetado para coletar dados significativos sobre fluxo, pressão e distribuição de água (Batty et al., 2012). Essa coleta de dados pode ser útil para companhias de água, bem como outras entidades, a fim de resolver problemas por meio de automação ou análise de dados. O termo *Smart* presente em *Smart Water* consiste em tornar inteligente as tarefas relacionadas a recursos hídricos através do uso de IoT.

A conectividade, nesse contexto, permite otimizar as estruturas de reservatórios de água detectando vazamentos na rede, vazões, pressões ou controle da distribuição de água na rede e permite aos operadores tomar decisões mais fundamentadas sobre a gestão dos recursos hídricos bem como acerca de parâmetros qualitativos (Mazhelis e Tyrväinen, 2014).

Tal monitoramento, comunicação em tempo real e automação aprimorada é possível devido a quantidade e variedade de sensores e plataformas de Internet das Coisas disponíveis no mercado. A nova era da indústria fornece soluções para controles de *Smart Water*, que se for possível a introdução do mesmo nas companhias municipais ou

particulares de distribuição de água, podem auxiliar não só na prevenção de vazamentos ou racionamento de água, como em um controle maior do fluxo e qualidade da água potável, trazendo assim segurança estrutural e sanitária para a comunidade (Saravanan et al., 2021).

## 2.5. RFC (*Request for Comments*)

As RFCs (*Request for Comments*) são documentos gerenciados pela IETF (*Internet Engineering Task Force*), órgão responsável por padronizar tecnologias utilizadas na Internet. Neste contexto, RFC 7452 (*Architectural Considerations in Smart Object Networking*) (Tschofenig et al., 2015) abrange aspectos sobre modelos de comunicações que podem ser estabelecidos por dispositivos inteligentes em projetos IoT. Abaixo são apresentados os modelos com maior relevância para este trabalho:

- ***Device-to-Device Communication Pattern***: comunicação recíproca entre dispositivos; necessita que os fabricantes fornecedores desses dispositivos estejam de acordo em utilizar pilhas de protocolos compatíveis para que a comunicação seja efetiva entre os mesmos;
- ***Device-to-Cloud Communication Pattern***: comunicação entre dispositivo e serviços de nuvem (*cloud*); não exige interoperabilidade direta entre esses, visto que os mesmos estabelecem comunicações com um provedor, o qual atua como entidade intermediária, assim sendo o responsável por garantir o fluxo de dados entre o dispositivo inteligente e o serviço de nuvem;
- ***Device-to-Gateway Communication Pattern***: comunicação entre dispositivo e *gateway* garante a interoperabilidade de comunicação com diversos destinatários dos dados; para que isso seja possível, um *gateway* é implementado na infraestrutura, a qual um dispositivo inteligente pode utilizar protocolos que não condizem com a pilha de protocolos do destinatário final dos dados, mas sim com os protocolos do *gateway*; os casos mais frequentes dessa abordagem ocorrem quando pelo menos um dispositivo utiliza tecnologias para comunicação que não são amplamente utilizadas no mercado alvo.

## 2.6. Fluxo de dados IoT

Parte das soluções baseadas em IoT processam grande quantidade de dados gerados por uma ampla variedade de sensores. Diante disso, a computação em nuvem oferece níveis de processamento e armazenamento para os dispositivos de IoT, visto que a maioria desses não possuem *hardware* suficiente para executar aplicativos robustos. Contudo, a computação em nuvem baseada em servidores de centros de computação remota pode causar latências de execução e custos operacionais que são inaceitáveis para muitos domínios de aplicativos de IoT (Geihs et al., 2020).

Por exemplo, em redes que consistem em diferentes tipos de dispositivos, como veículos autônomos, os requisitos de latência são baixos para que os dispositivos ou sistemas tenham tempo hábil para agir. Sendo assim, a computação de borda e a computação em névoa trazem a capacidade de processamento e armazenamento mais próximo das fontes de dados (Geihs et al., 2020). A Figura 4 ilustra uma arquitetura geral de três camadas a respeito desses modelos de processamento: *edge* (borda), *fog* (névoa) e *cloud* (nuvem).

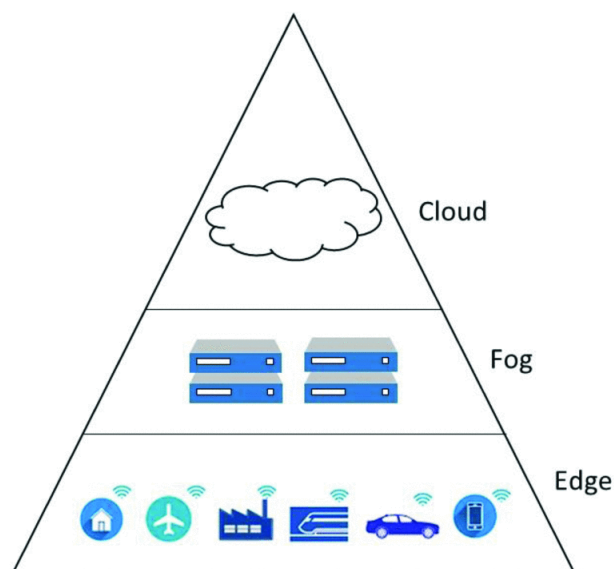


Figura 4. Arquitetura *Edge-Fog-Cloud* (Geihs et al., 2020).

### 2.6.1. Computação de borda (*Edge Computing*)

A computação de borda (*edge computing*) consiste em um nível da arquitetura que possui os dispositivos inteligentes ou dispositivos de borda que obterão as informações dos sensores e poderão interagir com os atuadores, mas que normalmente têm recursos de computação limitados (Ortiz et al., 2022). Nesta camada a ênfase é nos dados críticos, ou seja, aqueles que são fontes para a obtenção de informações que devem ser atualizadas rapidamente, em tempo real ou próximo disso. Portanto, o processamento é feito pelos *edge-nodes*, que estão mais próximos dos usuários, o que reduz a latência de resposta e os custos de rede.

### 2.6.2. Computação em névoa (*Fog Computing*)

A computação em névoa (*fog computing*) permite coletar informações de vários dispositivos de borda e atuar como um *gateway* para comunicação com outros *gateways* e com o terceiro nível da arquitetura (Ortiz et al., 2022). Na computação em névoa são feitas análises em tempo real de baixa latência de dados que não podem ser processados por dispositivos da *Edge*, uma vez que os mesmos geralmente são limitados em computação e armazenamento (Silveira e Dantas, 2022). Portanto, com relação ao próximo nível (computação em nuvem) o modelo em névoa permite a pré-análise dos dados para posteriormente enviar até a próxima camada, se houver necessidade.

### 2.6.3. Computação em nuvem (*Cloud Computing*)

A computação em nuvem (*computing cloud*) tem um sistema de comunicação bidirecional com camada - ou nós - *Fog*. Por outro lado, o nó de nuvem se inscreve nos tópicos de mensagens dos nós de *Fog*, para que possa obter informações de todos esses nós e tomar decisões de nível superior. Essas informações são adquiridas junto com outras fontes

de dados heterogêneas e processadas (Ortiz et al., 2022). Ademais, a responsabilidade de processamento e armazenamento de dados, na prática, é destinado para servidores hospedados em *data-centers* que podem ser acessados via Internet.

## 2.7. A Arquitetura da Internet das Coisas

A arquitetura da Internet das Coisas (IoT) é definida como uma estrutura em camadas na qual cada camada representa um conjunto coerente de responsabilidades. Ademais, até o momento, não existe uma padronização quanto a isso sendo possível encontrar diversas propostas de arquiteturas de soluções IoT. Algumas são compostas de 3 (três) camadas (Silva et al., 2018), 4 (quatro) camadas (Leite et al., 2017), 5 (cinco) camadas Khan et al. (2012) e entre outras. Para o desenvolvimento deste trabalho será levado em consideração a definição de 5 (cinco) camadas conforme proposto por Khan et al. (2012) e apresentado na Figura 5.

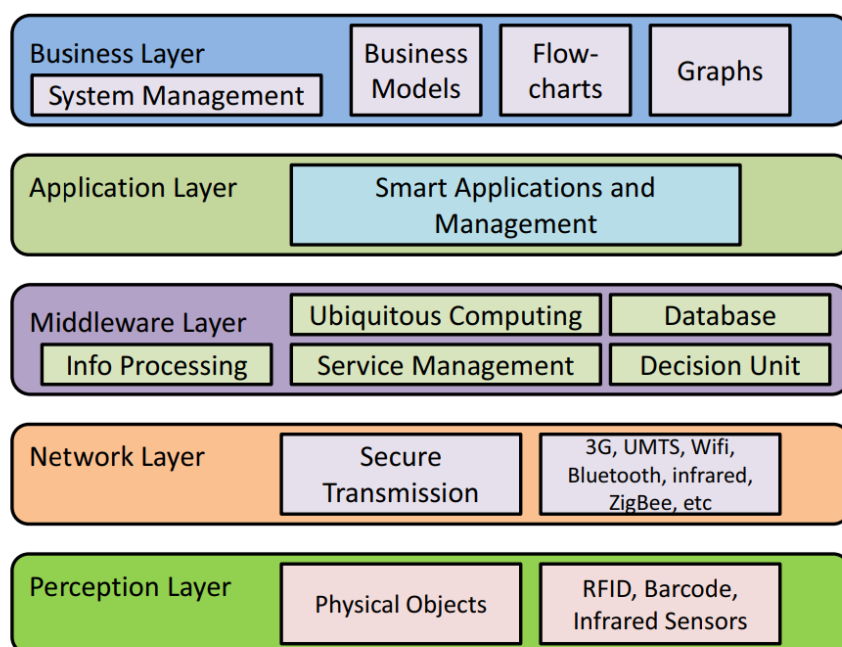


Figura 5. Camadas da arquitetura IoT (Khan et al., 2012).

### 2.7.1. Camada de Percepção

A primeira camada da IoT - a camada mais próxima do ambiente físico - tem como objetivo coletar informações no ambiente. Utilizando sensores e atuadores a camada é responsável por obter informações tais como localização, medições de temperatura, umidade, pressão e outras medições semelhantes (Kaur e Kaur, 2017). Essa camada pode ser compreendida como a camada de dispositivos, justamente por ser composta de objetos físicos e sensores (Khan et al., 2012).

Diante dessa camada, uma variedade de sensores e dispositivos, como etiquetas de identificação por radiofrequência (RFID), códigos de barras e sensores de infravermelho

possibilitam a digitalização dos dados para que possam ser transmitidos até as camadas seguintes da arquitetura (Khan et al., 2012).

Microprocessadores, microcontroladores, dispositivos lógicos programáveis conhecidos por FPGA (*Field Programmable Gate Array*), sensores, atuadores, interfaces de comunicação embarcada e sistemas operacionais destinados a sistemas embarcados são ferramentas implementadas nesta camada da arquitetura.

### **2.7.2. Camada de Comunicação**

Os dados nesta camada são transmitidos de forma segura para a próxima camada, ou seja, para a camada de *middleware*, que irá realizar o fluxo adequado para o armazenamento e processamento de dados (Khan et al., 2012).

A camada de comunicação usufrui de protocolos de comunicação cabeada ou de comunicação sem fio, sendo exemplos de comunicação cabeada: IEEE 802.3 (*Ethernet*) e CAN (*Controller Area Network*). Já para os protocolos de comunicação sem fio são exemplos: IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth), NB-IoT (*Narrow Band IoT*), LoRa, ZigBee, Thread, Sigfox, entre outros.

### **2.7.3. Camada Middleware**

Os dispositivos da IoT implementam diferentes tipos de serviços e cada dispositivo comunica-se apenas com dispositivos que implementam o mesmo tipo de serviço. Dessa forma, um *middleware* recebe as informações da camada de comunicação para que sejam armazenadas em um banco de dados, se for o caso. Ademais, nesta camada também é efetuado o tratamento da informação, a computação ubíqua, e a tomada de decisões automáticas com base nos resultados (Khan et al., 2012).

Nesse sentido, a computação ubíqua é o núcleo da Internet das Coisas, o que significa incorporar a computação e a conectividade em todas as coisas ao nosso redor garantindo interoperabilidade desses dispositivos heterogêneos através de padrões bem definidos. Sendo assim, a camada de *middleware* exerce a gestão de serviços abstraindo o *hardware* e fornecendo uma Interface de Programação de Aplicativos (API) para comunicação, gerenciamento de dados, computação, segurança e privacidade (Sethi e Sarangi, 2017).

Os objetivos da camada de *middleware* envolvem: interoperabilidade e abstrações de programação; descoberta e gerenciamento de dispositivos; escalabilidade; *big data* e análise de dados; segurança e privacidade; detecção de contexto quanto a origem dos dados (Sethi e Sarangi, 2017).

### **2.7.4. Camada de Aplicação**

A camada de aplicação fornece serviços aos usuários finais e é responsável pela formatação, apresentação dos dados. Esta camada, na Internet, geralmente é baseada em HTTP, no entanto, o HTTP não é adequado para ambientes com recursos limitados porque é bastante detalhado por natureza e, portanto, incorre em uma grande sobrecarga

de análise. Muitos protocolos alternativos foram desenvolvidos para ambientes de IoT, como o CoAP (*Constrained Application Protocol*) e o MQTT (*Message Queue Telemetry Transport*) (Sethi e Sarangi, 2017). Comumente, esses protocolos operam em *request and response* (requisição e resposta) ou *publish and subscribe* (publicação e assinatura)

Na presente camada inicia-se a especialização das soluções com base nos objetivos e contextos finais que serão utilizadas, portanto, as soluções abrangem ambientes e espaço inteligentes em domínios como: transporte, construção civil, cidade, varejo, agricultura, indústria, cadeia de suprimentos, emergência, saúde, cultura e turismo, meio ambiente e energia (Patel et al., 2016).

### **2.7.5. Camada de Negócios**

Nesta camada é possível definir os modelos comerciais, gráficos, fluxogramas e entre outros. Ademais, diante desta camada é definido as regras de negócios e a interação com os usuários finais.(Khan et al., 2012). Além disso, alarmes, *plugins*, ferramentas de visualização de dados são empregados na camada final da arquitetura. Algoritmos de *data mining*, *machine learning* e de inteligência artificial podem ser empregados neste nível para gerar melhor aproveitamento dos dados, para geração de *insights* a respeito de possibilidades de negócios e para otimização de fluxos de negócios.

### **2.8. Firmware**

*Firmwares* são desenvolvidos em uma abordagem modularizada, ou seja, as funcionalidades de um firmware são divididas em pequenas partes, o que exige definir funções individuais para, por exemplo, controlar módulos de hardware, funcionalidade de protocolo e assim por diante. Quando o desenvolvimento de um *firmware* é feito em alto nível de abstração, isso corrobora para a minimização das partes que exigem codificação em assembly e implementação das funcionalidades em C ou em linguagens de alto nível. Ademais, dividir a funcionalidade de um *firmware* em unidades únicas permite a aplicabilidade de métodos de desenvolvimento ágeis, como o desenvolvimento orientado por testes (TDD). Com isso, cada uma das unidades do *firmware* é submetido a testes de unidade, que testam se a unidade de software implementa o comportamento correto em diferentes condições (Kleeberger et al., 2015).

### **2.9. ThingsBoard Mobile Application**

ThingsBoard Mobile Application é um projeto de código aberto baseado no framework Flutter, o qual também possui código aberto, mantido pela empresa Google e se destina à criação de aplicativos multiplataforma. A aplicação móvel ThingsBoard possui gerenciamento de alarmes, possibilidade de interligar-se com os dispositivos físicos da camada de percepção, capacidades de tirar fotos, digitalizar código QR e funcionalidades de processamento utilizando os serviços da plataforma ThingsBoard<sup>1</sup>.

### **2.10. Trabalhos Correlatos**

Nesta subseção são apresentados projetos com a intenção de obter informações tais como possíveis problemas, êxitos e acertos análogos ao uso de IoT no âmbito da gestão de recursos hídricos. Sendo assim, entre os diversos trabalhos da área de *smart water* que podem

<sup>1</sup><https://thingsboard.io/products/mobile/>

ser encontrados nos periódicos científicos e na indústria, elencamos 3 (três) trabalhos que possuem correlação com a proposta apresentada.

### 2.10.1. Smart Water Tank

Publicado na Conferência Internacional de I-SMAC (IoT in Social, Mobile, Analytics and Cloud) em 2017, possui como motivação o desperdício de água que ocorre devido ao transbordamento dos reservatórios de água, que são gerenciados manualmente por um operador. Além disso, é detalhado na seção de introdução que na ausência do operador, a água continua transbordando até que o motor seja desligado. Nesse sentido, existem outros projetos que são implementados com sensores de imersão, mas por estarem em contato com a água, há uma grande probabilidade de enferrujamento do material utilizado no sensor conforme afirmam os autores em (Shah et al., 2017).

O trabalho trata-se de um projeto utilizando um microcontrolador ESP8266, interligado a um sensor ultrassônico, que lê a distância da superfície da água em um reservatório de água. O microcontrolador, quando ligado à Internet, envia os valores de leitura para a base de dados na nuvem. Além disso, esse mesmo microcontrolador recupera alguns valores da base de dados, estabelecendo fluxo de dados bidirecional, que são apresentados para o usuário num aplicativo móvel desenvolvido para a plataforma Android. Por conseguinte, existe um motor que é controlado pelo usuário, através do aplicativo desenvolvido (Figura 6).

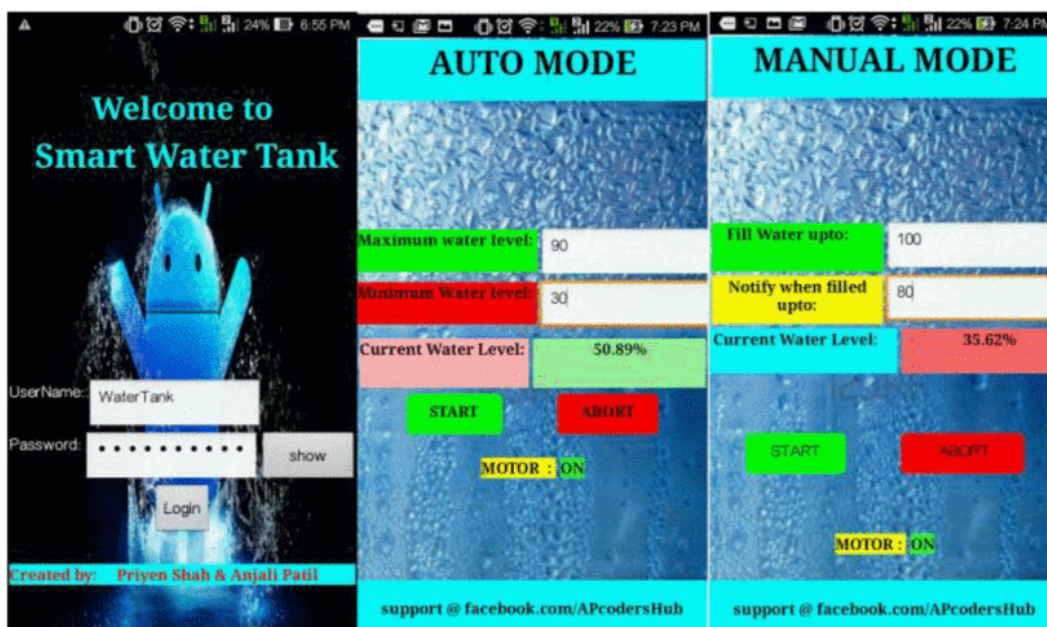


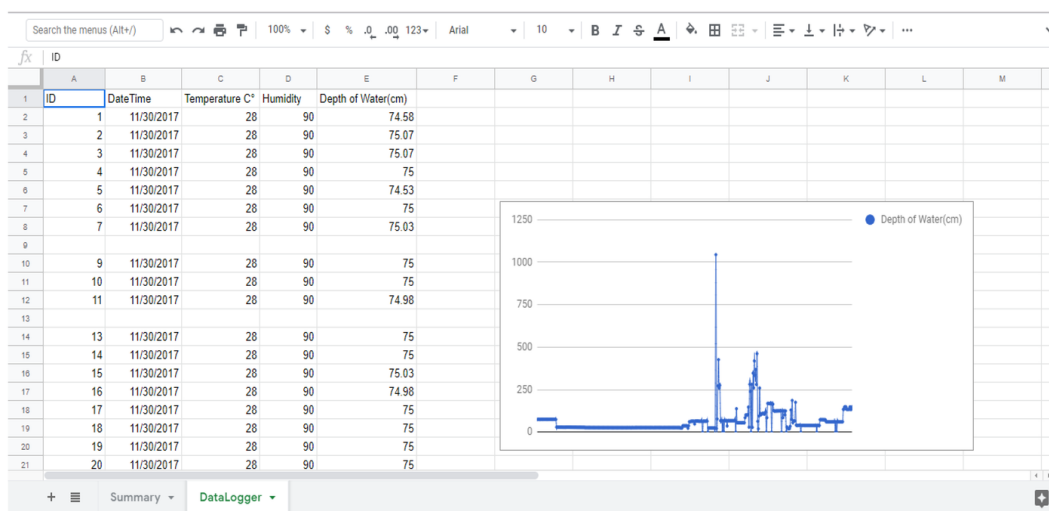
Figura 6. Capturas de tela da aplicação (Shah et al., 2017).

Outras ferramentas utilizadas são: nuvem Firebase e MIT App Inventor-Android Application (plataforma de programação projetada pelo MIT para ajudar os programadores a desenvolver e testar aplicativos Android).

## 2.10.2. Real-time monitoring of water level and storage dynamics of irrigation tank using IoT

A motivação deste trabalho surge com base no grande número de reservatórios de água na Índia, visto que esse país é dotado de precipitação de chuva anual (1.215mm por ano) não bem distribuída ao longo do período. A respeito desses reservatórios, são amplamente utilizados na agricultura e geralmente geridos pelas comunidades locais como um recurso de propriedade comum (Shah et al., 2017).

Sendo assim, o projeto tem por principal objetivo estabelecer monitoramento remoto de temperatura, umidade e dinâmica do nível de água de um determinado reservatório de água para estimar o volume de água disponível para processo de irrigação. O dispositivo baseado em IoT utiliza um microcontrolador desenvolvido para a plataforma NodeMCU, ademais, utiliza sensores e redes de alimentação. Os dados recolhidos na camada de percepção são enviados para a rede de computação em nuvem por meio de WiFi para posteriormente a análise de dados. O nível de água presente no reservatório, a temperatura e a umidade são apresentados no programa *Google Sheets* armazenadas na nuvem do empresa Google (Figura 7). Ademais, o dispositivo de monitorização do nível de água do tanque de irrigação baseado na IoT desenvolvido mede, analisa e fornece uma plataforma de apoio à decisão para os agricultores (Krishnaveni et al., 2020).



**Figura 7. Monitoramento da dinâmica do nível de água através de aplicação web (Krishnaveni et al., 2020).**

## 2.10.3. Smart two-tank water quality and level detection system via IoT

A qualidade de recursos hídricos também é um objetivo amplamente presente em soluções *Smart Water*. Não sendo exclusividade de medições quantitativas como nos outros trabalhos correlatos já abordados.

Neste trabalho proposto por (Olisa et al., 2021) foi desenvolvida uma aplicação móvel Android integrada e um sistema de controle para avaliar a qualidade da água, efetuar a verificação do nível no reservatório superior e ativar o controle inteligente da bombagem. Foi utilizada uma técnica ultrassônica de pulso eco para verificar o nível da água,

enquanto os sinais de turbidez e pH da água foram utilizados para verificar a qualidade da água (Figura 8).

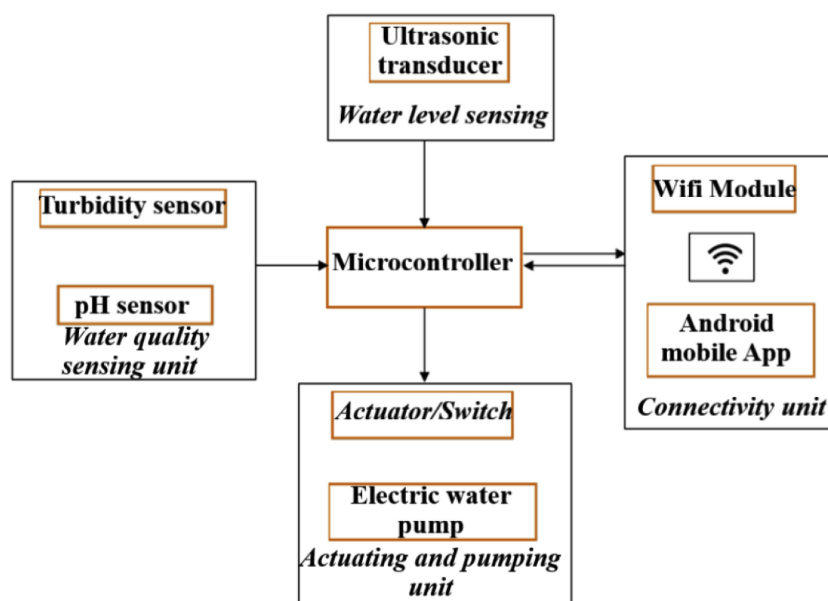


Figura 8. Diagrama de blocos do sistema (Olisa et al., 2021).

Conforme apresentado na Tabela 1, foram analisados três trabalhos correlatos abordados nas subseções anteriores 2.10.1, 2.10.2, 2.10.3. Esses estão representados no quadro abaixo como 1\*, 2\* e 3\* respectivamente.

Tabela 1. Trabalhos Correlatos

Projeto	MCU	Sensor	Atua	Interface	Alarmes	Dados	Outras Ferramentas
1*	ESP8266	Ultrassom	Sim	Aplicativo Android	-	Firestore	MIT App Inventor
2*	ESP8266	Ultrassom; Temperatura; LIDAR	-	Google Sheets	-	Google Sheets; Google Cloud	AutoCAD; ArcGIS
3*	ESP32	Ultrassom; Turbidez; pH	Sim	Aplicativo Android	SMS	Firestore	LabVIEW

### 3. Solução Proposta

A solução proposta visa coletar dados por meio de sensores, a fim de apresentar ao usuário séries temporais sobre os reservatórios de água do do Câmpus Lages do IFSC. O acesso aos dados tratados pode ser feito pelo usuário através de navegadores web ou por um aplicativo móvel desenvolvido neste trabalho (baseado no software de código aberto *ThingsBoard Mobile Application*). Além disso, os usuários administradores podem definir os

intervalos de emissão dos alarmes de nível de água. Dessa forma, a solução provê funcionalidades de aquisição, apresentação e configuração de dados hídricos, por meio de uma infraestrutura de sensores, aplicação web e aplicativo móvel.

O projeto baseou-se na arquitetura IoT de 5 camadas, conforme detalhado na subseção 2.7, para seu planejamento e implementação. As subseções seguintes descrevem os componentes eletrônicos utilizados, conexões Wi-Fi, serviços da plataforma ThingsBoard e um aplicativo móvel derivado do projeto ThingsBoard Mobile Application, bem como o serviço web. A Figura 9 apresenta a arquitetura da solução proposta.

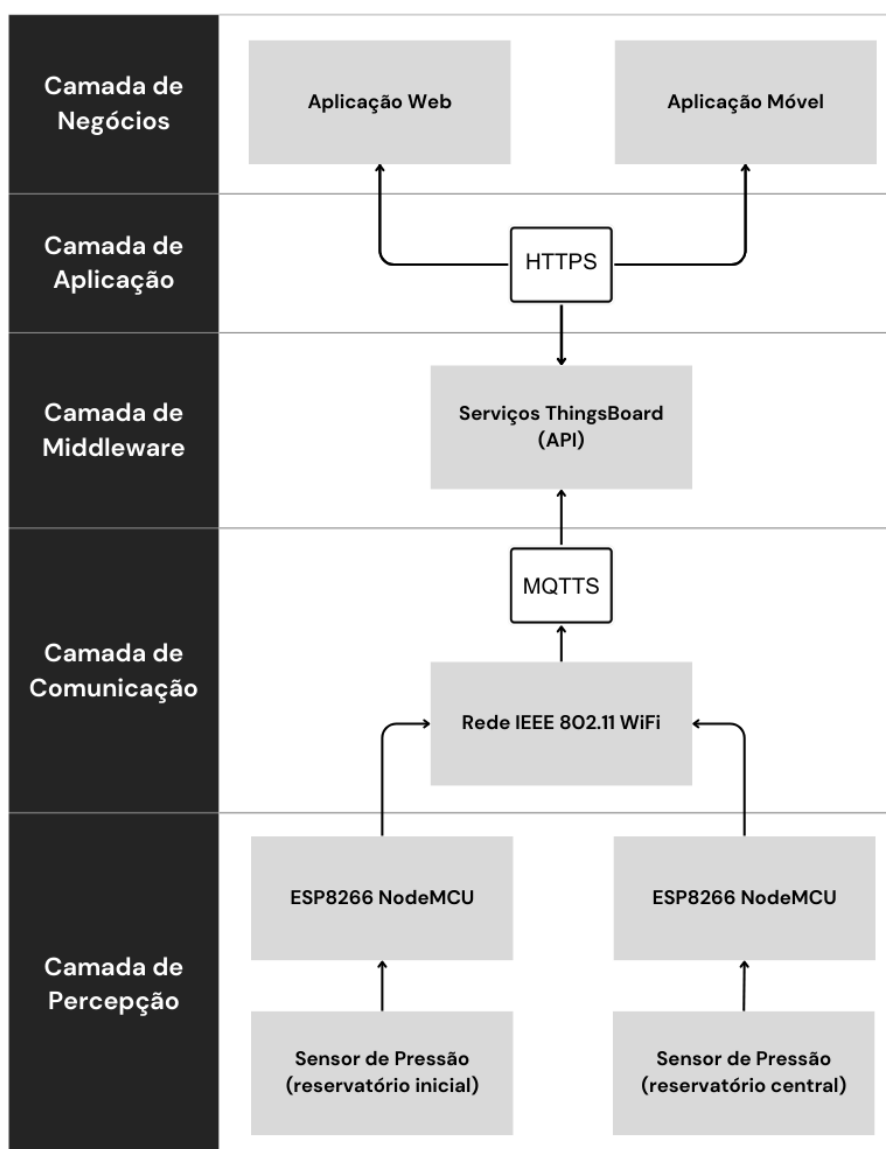


Figura 9. Arquitetura organizacional da solução proposta.

### 3.1. Camada de Percepção

A camada de percepção foi implementada para coletar dados sobre a pressão exercida pelas colunas de água em dois reservatórios do campus Lages. O primeiro reservatório

trata-se de um reservatório de entrada que recebe água potável através do serviço municipal de água e saneamento da cidade, já o segundo reservatório dispõe maior capacidade de armazenamento em relação ao primeiro e conta com 3 (três) divisões, sendo elas: água para serviço de bombeiros, água cinza e água potável.

Para cada reservatório foi elaborado um conjunto de componentes eletrônicos. Esses componentes tem por objetivo realizar leituras periódicas de pressão, e enviar os dados obtidos para um servidor web do campus, através de conexão Wi-Fi.

Cada conjunto de componentes eletrônicos contém os seguintes dispositivos:

- 1 (uma) fonte de alimentação 24V para fornecer tensão necessária para o sistema os sensores<sup>2</sup> e microcontroladores;
- 1 (um) regulador de tensão para converter a tensão de 24V para 3V3 de entrada no microcontrolador;
- 1 (um) transdutor de pressão (4 - 20mA) para medir a pressão das colunas de água;
- 1 (um) conversor analógico/digital (ADC) para converter o sinal de entrada recebido do transdutor em sinal digital a ser utilizado pelo microcontrolador;
- 1 (um) microcontrolador NodeMCU ESP8266-12E que recebe os dados digitais e os envia ao servidor web baseado na plataforma ThingsBoard.

### 3.1.1. Fonte 24V 5A

Afim de estabilizar a alimentação elétrica do nosso sistema, a Fonte 24V 5A (Figura 10) utiliza o chaveamento, incorporando um regulador de comutação que converte energia elétrica (AC 220V para DC 24V) de forma eficiente.



Figura 10. Fonte 24V 5A.

### 3.1.2. Microcontrolador NodeMCU ESP8266-12E

O ESP8266 integra em um único chip um microprocessador LX105 de 32 bits, memória flash, SRAM, interface Wi-Fi 802.11 e GPIOs programáveis. Possui interface USB-

<sup>2</sup>Foi utilizada uma fonte de 24V para compatibilizar o sistema com o padrão de alimentação industrial comumente encontrado no mercado em sensores de 4 - 20mA.

Serial e o regulador de tensão 3,3V. Tal qual utiliza a comunicação via cabo micro-usb. Contando com uma antena embutida, 11 pinos de I/O e um conversor analógico-digital. Ademais, trata-se de um microcontrolador popular, amplamente utilizado em projetos de código aberto e comerciais que requerem monitoramento e controle remoto. Na Figura 11 são apresentados alguns componentes deste microcontrolador.

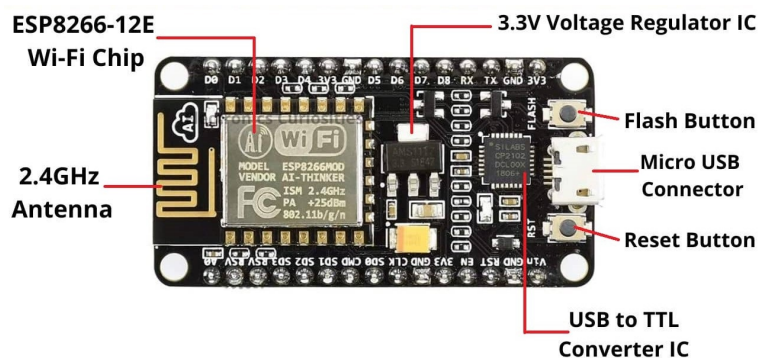


Figura 11. Microcontrolador NodeMCU ESP8266-12E.

### 3.1.3. Regulador de tensão

Reguladores de tensão são amplamente utilizados em sistemas eletrônicos que exigem uma fonte de alimentação estável. Para esse projeto foi utilizado o modelo LM2596 (Figura 12), o qual possui capacidade de fornecer tensão de saída regulada entre 0,8V e 20V, através de uma fonte de entrada entre 4,5V - 28V, portanto, se enquadra como um regulador de tensão *step-down*, converte uma tensão de entrada em uma tensão de saída menor.

O uso deste regulador de tensão tem por objetivo fornecer uma solução econômica para garantir os níveis ideais de tensão para cada componente, o componente possibilita reduzir a tensão de 24V (valor de saída da fonte de alimentação deste projeto) para até 1.5V e com corrente de saída de até 2A. Além disso, o componente dispensa circuitos extras, basta conectá-lo à fonte e ajustar a tensão com um potenciômetro.

### 3.1.4. ADC 4mA - 20mA - Módulo para medições de sensores

O módulo de 4mA a 20mA é um dispositivo eletrônico projetado para medição e controle de processos industriais. O dispositivo funciona convertendo um sinal de corrente entre 4-20 mA, comumente utilizado na indústria. O módulo conta com um resistor integrado, por onde a corrente de entrada percorre e gera a tensão correspondente. Essa tensão é disponibilizada em pinos que podem ser lidos por um microcontrolador, por exemplo. O módulo é calibrado de fábrica para fornecer uma saída entre 0 e 5 V, correspondente a uma entrada de 4-20 mA. Isso possibilita certa praticidade na leitura do sinal de corrente e integração com sistemas de aquisição de dados e controle. Sendo assim, o componente em questão é uma solução compacta e versátil para aplicações de medição e controle em automação industrial. No caso em específico do modelo utilizado neste trabalho (Figura



Figura 12. Regulador de tensão LM2596 DC-DC.

13) o mesmo baseia-se no chip ADS1115 que fornece os dados convertidos através de uma interface de comunicação I2C (*Inter-Integrated Circuit*).

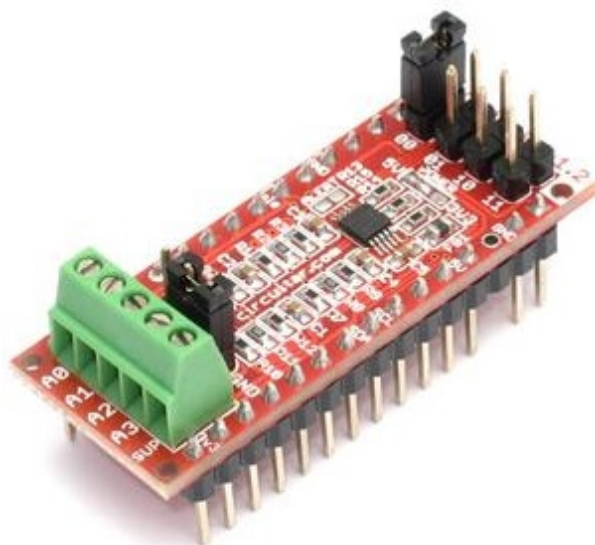


Figura 13. Nanoshield ADC 4-20.

### 3.1.5. Transdutor de pressão

Utilizados transdutores de pressão de 3 e 30 MCA (Metros de Coluna D'Água) através de uma alta impedância de 5000 ohms ou mais. Esses dispositivos eletromecânicos convertem valores de pressão em corrente, que variam de 4 mA até 20 mA para o fundo de escala, dos quais geram os dados lidos pelo microcontrolador. Na Figura 14, os 24 V

representam a tensão de operação do componente, 4 a 20 mA é o intervalo de valores possíveis para o sinal de saída e a malha é útil para proteger os fios do componente.

O uso de transdutores de pressão para este trabalho é adequado para os reservatórios do campus, visto que os modelos destes equipamentos são suficientes para realizarem medições lineares com baixa histerese, operação em reservatórios que exercem alta pressão, e resistência à corrosão provocada pelo contato frequente com a água.



Figura 14. Transdutor de pressão 3 MCA.

### 3.2. Camada de Comunicação

A comunicação deu-se por meio da tecnologia Wi-Fi (IEEE 802.11), visto que o local de estudo já oferece pontos com cobertura para este protocolo, facilitando assim a conectividade de soluções IoT no câmpus. O *firmware* foi executado nos microcontroladores para que os dados coletados na camada de percepção pudessem ser enviado à plataforma ThingsBoard.

O protocolo de comunicação que foi utilizado é o MQTT, um protocolo de comunicação para a IoT que foi desenvolvido para realizar o transporte de mensagens pelo modo *publish/subscribe* (publicação/ assinatura). Ademais, outra característica desse protocolo é sua leveza, ideal para conectar dispositivos remotos com uma largura de rede estreita. Mesmo assim, o protocolo é capaz de proporcionar comunicações bidirecionais e escalabilidade com milhões de dispositivos IoT.

Juntamente com o protocolo MQTT foi utilizado o TLS, que é a sigla para *Transport Layer Security*, uma ferramenta de segurança digital usada para comunicar utilizando certificados, verificando a identidade dos servidores e garantindo que somente o servidor possa descriptografar e ler informações enviadas pelo *client* e reciprocamente. Tal junção cria o MQTTS, do qual é utilizado para mensagens leves de dispositivos IoT, comunicando dispositivo para servidor, de forma segura.

### 3.3. Camada de Middleware

Na camada de *middleware* deste projeto foi utilizada a plataforma ThingsBoard, a qual é amplamente utilizada em soluções IoT, permitindo coletar e processar volumes de dados de sensores. A arquitetura do ThingsBoard é escalável em microsserviços e provê funcionalidade como *dashboards*, regras de alarmes e integrações.

Além disso, a plataforma possui uma versão chamada *Community* a qual pode ser utilizada gratuitamente. Outro ponto a destacar é que apesar da versão utilizada neste projeto não possuir custos, a comunidade do ThingsBoard é ativa, assim corroborando para uma ampla documentação e suporte. Do ponto de vista de integrações do Thingsboard é possível utilizar múltiplos protocolos de comunicação (MQTT, HTTP, CoAP e outros) assim como diferentes bancos de dados (SQL e NoSQL).

Dessa forma, a escolha pelo ThingsBoard para este trabalho se dá pelo fato de que essa plataforma foi possível ser utilizada em diversas camadas da aplicação. Na camada de middleware foi estabelecido comunicações do tipo Publish/Subscribe através do protocolo MQTTS, interligando os microcontroladores e o servidor responsável por tratar os dados recebidos. Essas comunicações estabelecidas com a camada de middleware atuam como um hub centralizado, recebendo e processando os dados enviados através de dois microcontroladores ESP 8266, cada um desses representa um reservatório do campus.

Além disso, o ThingsBoard foi crucial para a Camada de Aplicação e Camada de Negócios, a forma que o ThingsBoard foi utilizado nessas camadas estão detalhadas nas subseções 3.4 e 3.5.

### 3.4. Camada de Aplicação

A camada de aplicação estabeleceu a comunicação entre a API descrito na camada de *middleware* e um aplicativo móvel, que deu-se utilizado por funcionários competentes ao gerenciamento de água no campus Lages.

Para essa camada foi utilizado o protocolo HTTP do qual é o principal em acesso e distribuição de dados *World Wide Web*, focado em eficiência, e novamente o TLS que conforme explicado anteriormente é uma tecnologia que oferece segurança por meio de criptografia. Com tal junção obtém-se o protocolo HTTPS, do qual é utilizado para criar um canal seguro, assim utilizando os certificados do TLS, e estabelecendo uma comunicação que cliente e servidor utilizam o mesmo conjuntos de cifras e chaves.

### 3.5. Camada de Negócios

A aplicação móvel conta com a visualização de gráficos que representam o nível de água nos reservatórios ao longo do tempo, os valores desses gráficos são oriundos das medições de pressões e cálculos matemáticos. Ademais, para o desenvolvimento do aplicativo móvel foi utilizado a uma solução de código aberto conhecida por *ThingsBoard Mobile Application*, a qual customizado de acordo com as definições deste trabalho.

Conforme os dados foram recebidos pela plataforma ThingsBoard tornou-se possível realizar a detecção de intervalos de valores específicos que implicam em ações automatizadas. Para este projeto, essa automação ocorreu nos alarmes, que foram emitidos com base em valores de pressão exercido pelas colunas de água dos reservatórios. Ademais, os alarmes foram destinados aos funcionários competentes pela gestão dos reservatórios, acessando os mesmos via *App Mobile* ou via navegador web.

## 4. Testes e Resultados

Nesta seção são apresentados os principais parâmetros utilizados nos testes realizados bem como são apresentados e discutidos os resultados obtidos.

### 4.1. Parâmetros de testes

A fim de que a realização dos testes, foram utilizados *mocks*<sup>3</sup> (objetos que simulam o comportamento de objetos reais de forma controlada) representando os dois reservatórios de água do Campus IFSC Lages, definidos como Reservatório Central e Reservatório Entrada, ambos portando microcontroladores e sensores de monitoramento de nível de água. Tais *mocks* são implementados no *firmware* dos microcontroladores e simulam informações que seriam obtidas pelos sensores. Após isso, os dados obtidos são transmitidos via Wi-Fi ao servidor do ThingsBoard instalado no Campus, para posteriormente ser realizada a etapa de tratamento de dados e apresentação de *dashboard* informativos.

### 4.2. Resultados

#### 4.2.1. Firmware

O *firmware* foi desenvolvido através da Arduino IDE, o seu código-fonte<sup>4</sup> é encontrado em um arquivo de extensão INO contendo instruções nas linguagens C e C++ para os *sketches* do Arduino. Após a compilação, é obtido o código objeto e o mesmo convertido em um arquivo binário que, esse por sua vez, é gravado via Micro-USB na memória *flash* do microcontrolador ESP8266-12E.

Além da biblioteca padrão do Arduino IDE, a biblioteca instalada na implementação do *firmware* foi a “ThingsBoard 0.6” mantida por ThingsBoard Team e suas respectivas dependências, as quais foram automaticamente detectadas pela IDE e instaladas após a permissão do usuário para essa ação. Outra biblioteca instalada para a execução e desenvolvimento do código foi a dos módulos da placa, ou seja, o pacote “ESP8266 2.7.4” mantida pela ESP8266 Community. Todas as dependências mencionadas podem ser obtidas através da Arduino IDE com acesso à Internet.

O conteúdo do *firmware* inclui funções para conectar e reconectar ao Wi-Fi, configuração de taxa de transmissão serial para depuração, valores de SSID e senha da rede Wi-Fi para que o ESP8266 possa se conectar à mesma. A identificação para a autenticação dos dispositivos no servidor ThingsBoard foi realizada através de *tokens* únicos de acesso, os quais foram gerados na etapa de cadastramento dos microcontroladores no ThingsBoard versão Web do câmpus.

Como ponto de partida para o início da execução/simulação, foi definido o valor de pressão em 16 mA, após isso, a execução do programa inicia um laço de repetição sem condição de parada. Nesse laço de repetição, as primeiras instruções envolvem a verificação se a conexão entre o MCU e o servidor continua ativa e, se necessário é realizada uma tentativa de reconexão. As verificações de conexão e tentativas de reconexão envolvem tanto aspectos de Wi-Fi, quanto de conexão MQTTS com o servidor ThingsBoard. Ainda sobre o laço de repetição, nele ocorre a geração de um valor numérico inteiro

<sup>3</sup>Foram utilizados *mocks* de dados em função da indisponibilidade do IFSC/Lages realizar a aquisição dos transdutores de pressão em tempo hábil para a conclusão deste trabalho.

<sup>4</sup>Acessível em <https://github.com/iot-reservatorios/firmware>

e aleatório entre -1 e 1, este valor aleatório é multiplicado por um décimo e o resultado dessa operação é incrementado ao último valor de pressão e enviado.

Após o envio da mensagem, a próxima instrução envolve deixar o microprocessador em espera. Isso foi definido em virtude da oscilação do nível de água ocorrer de forma gradual e vagarosamente se observado da perspectiva de poucos segundos. Portanto, caso o programa fosse executado sem instrução de tempo de espera, ocasionaria em consumo desnecessário de banda de rede, processamento e armazenamento, visto que, a obtenção de leitura de pressão de segundo em segundo não iria refletir maior qualidade no serviço a ser oferecido para o monitoramento de água do campus. Diante dos fatos apresentados, foi definido o intervalo de 120 segundos entre uma mensagem enviada e o início da geração de valores aleatórios para a próxima mensagem. Esse valor representou um intervalo de tempo confortável para visualizar e monitorar o comportamento do sistema durante a etapa de desenvolvimento.

### 4.3. Tipos de Usuários ThingsBoard

A plataforma ThingsBoard fornece diversos tipos de usuários com funcionalidades e permissões distintas. Estão disponíveis os seguintes tipos de usuários para o sistema desenvolvido:

- Administrador de Sistema (*System Administrator*), possui controle total sobre o servidor ThingsBoard, incluindo todas as configurações de sistema. Portanto, pode gerenciar usuários e *tenants*<sup>5</sup>. Neste trabalho, o Administrador de Sistema representa a Coordenadoria de Tecnologia da Informação e Comunicação (CTIC) do campus.
- Administrador *Tenant* (*Tenant Administrator*), pode administrar os recursos de um *tenant* específico. Sendo assim, esse tipo de usuário gerencia os itens do seu respectivo *tenant*, inclusive para ações de criação, alteração e exclusão. Neste trabalho, os administradores *tenants* são os autores do projeto.
- Usuário Cliente (*Customer User*), pertence à um *tenant*, pode visualizar e interagir com os recursos disponíveis no *tenant* conforme permissões concedidas. Neste trabalho, o tipo Usuário Cliente foi projetado para a equipe de administração do campus.
- Usuário Somente Leitura (*Read Only User*), tipo de usuário que tem permissão de visualizar *dashboards* e dados, mas sem a permissão de criar, editar ou excluir. Neste trabalho, os Usuários Somente Leitura seriam utilizados pelas pessoas que exercem a atividade de zeladoria do campus, especificamente de recursos hídricos.

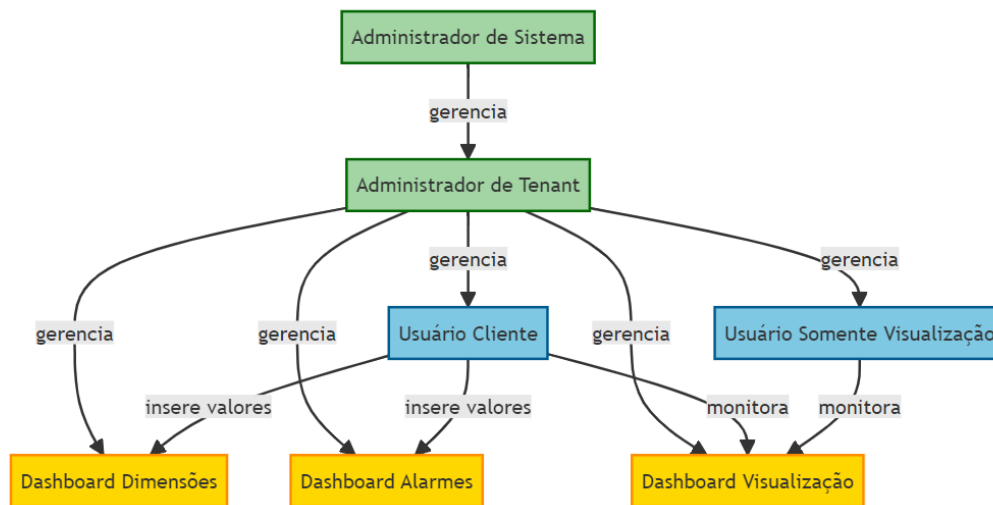
A Figura 15 exemplifica as principais atribuições designadas aos tipos de usuários presentes neste projeto.

#### 4.3.1. Dashboard Alarmes

Este painel é útil para definir em quais intervalos de metros de coluna de água serão acionados os alarmes, portanto, para os casos de excesso de água é possível definir dois

---

<sup>5</sup>No ThingsBoard, um *tenant* representa uma unidade organizacional que possui seus próprios usuários, dispositivos, *dashboards* e demais recursos.



**Figura 15. Atribuições de Administradores e Usuários.**

alarmes. O primeiro alarme, trata-se de um nível de atenção, ou seja, este alarme ocorre antes do segundo alarme, que foi nomeado como alarme de nível crítico.

Para casos de escassez, o mesmo acontece, é possível definir um valor em mca para o alarme de nível de atenção e outro valor em mca para o alarme de nível crítico. Sendo assim, entre o excesso e a escassez, existe um intervalo numérico que não representa risco ao que está sendo monitorado, esse é o intervalo seguro. Sempre que o valor de leitura pertencer ao intervalo seguro, se houver alarmes ativos, esses alarmes serão desativados, visto que o valor atual não representa risco de Excesso ou Escassez. O responsável técnico pelos reservatórios pode definir a emissão de alarmes através de um *widget* (Figura 16). Ademais, os alarmes podem ser visualizados conforme apresentados na subseção 4.3.3 e visualmente através da Figura 24. A emissão de sons de alarmes depende da permissão concedida pelos usuários ao aplicativo para acessar as notificações, bem como do modo de notificação corrente no dispositivo móvel (silencioso, *vibracall*, com som).

A histerese - diferença entre os limiares de transição de estado - não fora definida por intervalos quantitativos para os alarmes, portanto, o sistema fica a mercê de não emitir alarmes excessivamente com base nas leituras de pressão, que ocorrem a cada 120 segundos para cada reservatório.

#### 4.3.2. Dashboard Dimensões

A *dashboard* apresentada na Figura 17 foi desenvolvida para inserir valores quantitativos a respeito da estrutura física dos reservatórios.

Cada *widget* apresentado na Figura 18 contém parâmetros que caracterizam as dimensões físicas de cada reservatório. O campo “Altura Máxima de Armazenamento”

Reservatório	Excesso - Nivel Crítico	Excesso - Nivel Atenção	Escassez - Nivel Atenção	Escassez - Nivel Crítico
Reservatório Entrada	2.75	2.25	1.25	0.75
Reservatório Central	7.75	7.0	4.25	3.5

**Figura 16. Configuração de Alarmes.**

Reservatório	Altura Máxima de Armazenamento (m)	Altura do Encanamento (m)	Raio do Reservatório (m)	Volume Máximo (m³)
Reservatório Entrada	2.3	0	2	28.90
Reservatório Central	7.5	8.7	2.25	119.28

**Figura 17. Dimensões dos Reservatórios.**

leva em consideração apenas a altura útil interna do reservatório. O campo “Raio do Reservatório” representa o raio interno da estrutura. Já o campo “Altura do Encanamento” corresponde à coluna vertical de encanamento que, na maioria dos casos, inicia no fundo do reservatório e termina no nível do solo, próximo de onde se encontra o sensor de pressão.

Para os casos em que o transdutor de pressão não seja instalado no encanamento logo ao fundo do reservatório, a altura e conseqüente volume do encanamento de saída até o ponto de instalação do transdutor de pressão será desconsiderada. Para que essa característica seja desconsiderada nos cálculos e nas demais operações do sistema, basta digitar o número 0 na entrada de valor “Altura do Encanamento” e salvar. Caso o usuário prefira incluir essa característica, deve ser digitado o valor numérico de medida entre a distância do fundo do reservatório e a borda superior do encanamento. Por exemplo, caso um reservatório possua o encanamento de saída rente ao fundo do reservatório e o

Reservatório Entrada	Reservatório Central
Altura Máxima de Armazenamento (m): 2,3	Altura Máxima de Armazenamento (m): 7,5
Altura do Encanamento (m): 0	Altura do Encanamento (m): 8,7
Raio do Reservatório (m): 2	Raio do Reservatório (m): 2,25
Volume Máximo (m³): 28,90	Volume Máximo (m³): 119,28

**Figura 18. Dimensões dos Reservatórios.**

encanamento possui 5 centímetros, deve ser digitado o valor "-0,05".

Com base nos *widgets* apresentados, o campo Volume Máximo (m<sup>3</sup>) é calculado automaticamente. Por ser um campo resultado, o Volume Máximo é somente de leitura, sendo assim, o campo não pode ser editado pelo usuário.

A equação utilizada é a seguinte:

$$Volume\ Máximo\ do\ Reservatório = \pi \times r^2 \times h \quad (1)$$

onde  $\pi$  é a constante matemática PI,  $r$  é o raio do reservatório e  $h$  é a altura do reservatório.

A unidade de medida utilizada para Raio do Reservatório e Altura do Reservatório é metro (m), já a unidade de medida utilizada para o Volume Máximo do Reservatório é de metro cúbico (m<sup>3</sup>).

#### 4.3.3. Dashboard Visualização

O *Dashboard* de visualização apresentado na Figura 19, trata-se do principal painel deste projeto. Nesse há diversos *widgets* que compõem uma gama de informações sobre os níveis de água no câmpus.

A Figura 20 apresenta os reservatórios de entrada e central. A interface gráfica é dinâmica, sendo assim, conforme o volume de água oscila, essa oscilação é representada no *widget* aumentando ou diminuído seu preenchimento na cor azul. Ainda na figura, no momento em que foi capturada, o Reservatório Entrada estava com 80% de sua capacidade total utilizada, enquanto o Reservatório Central estava com 57%. Essa visualização é alterada conforme os novos valores de pressão são persistidos no banco de dados da aplicação.

Na Figura 21 dois *widgets* são apresentados. O *widget* à esquerda representa os dados obtidos através da última mensagem recebida, as informações presentes no quadro são

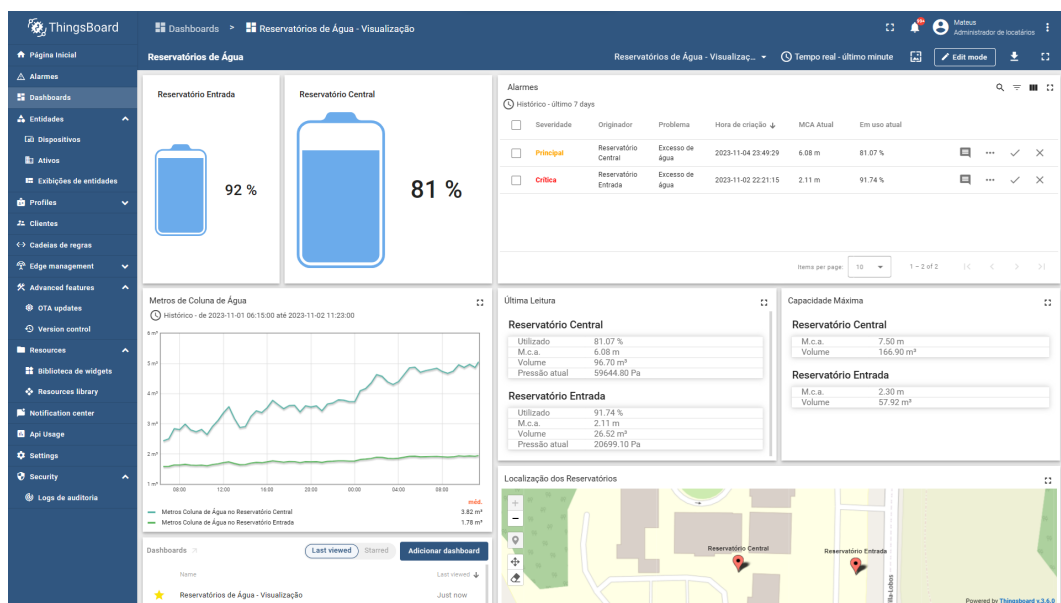


Figura 19. Visão Geral do *Dashboard* de Visualização.

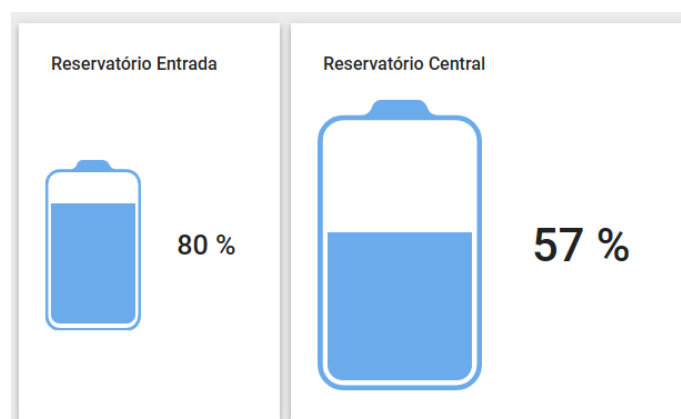
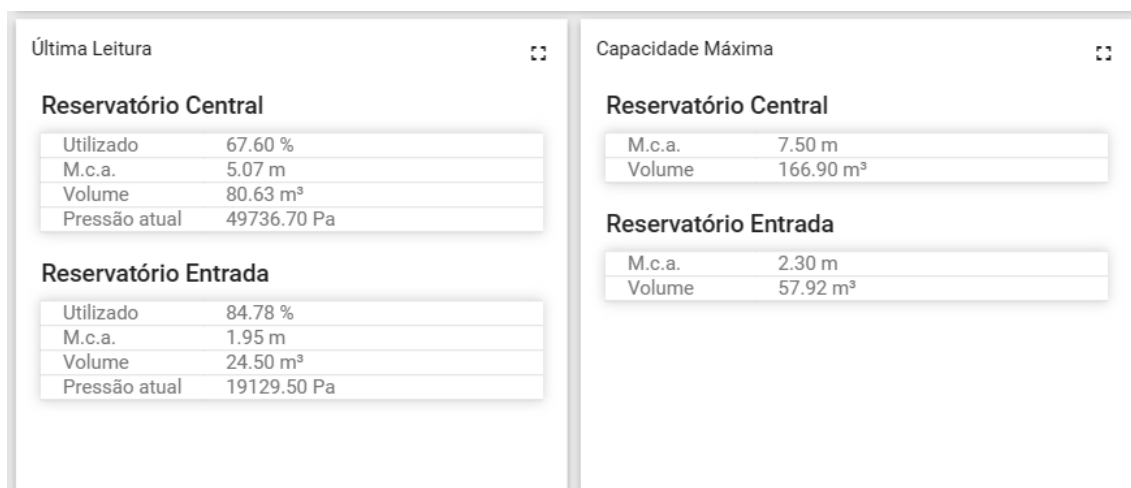


Figura 20. Volume de água nos reservatórios.

as seguintes: “Utilizado” (valor percentual que corresponde ao volume atual); “M.c.a.” (valor de metros de coluna de água presente em cada reservatório); “Volume” (metros cúbicos sobre a quantidade ocupada por água no interior do reservatório); “Pressão atual” (pressão – em Pascal – exercida no encanamento de saída). No *widget* à direita são disponibilizadas informações sobre a estrutura física de cada reservatório cadastrado.

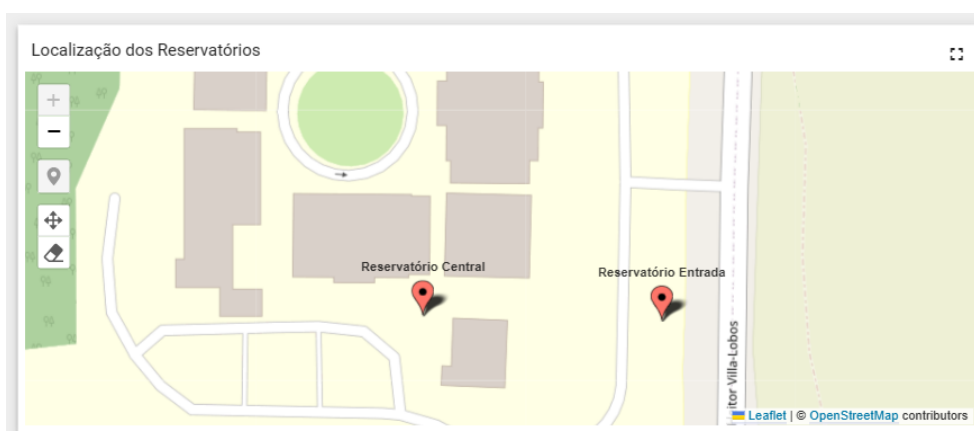
Sendo assim, a utilização combinada desses dois *widgets* permite uma visão abrangente dos níveis de água do câmpus, mostrando indicadores atuais e de capacidade total. Isso fornece informações que podem ser úteis em momentos decisivos, como por exemplo, em momentos que sejam necessários alterar os valores de vazão dos reservatório.

Na Figura 22 é visível a forma com que o *dashboard* trata a localização dos reservatórios. A posição geográfica de cada reservatório é indicada por marcadores na cor vermelho e preto. Esses marcadores são gerados através da biblioteca Leaflet juntamente



**Figura 21. Última Leitura e Capacidade Máxima dos Reservatórios.**

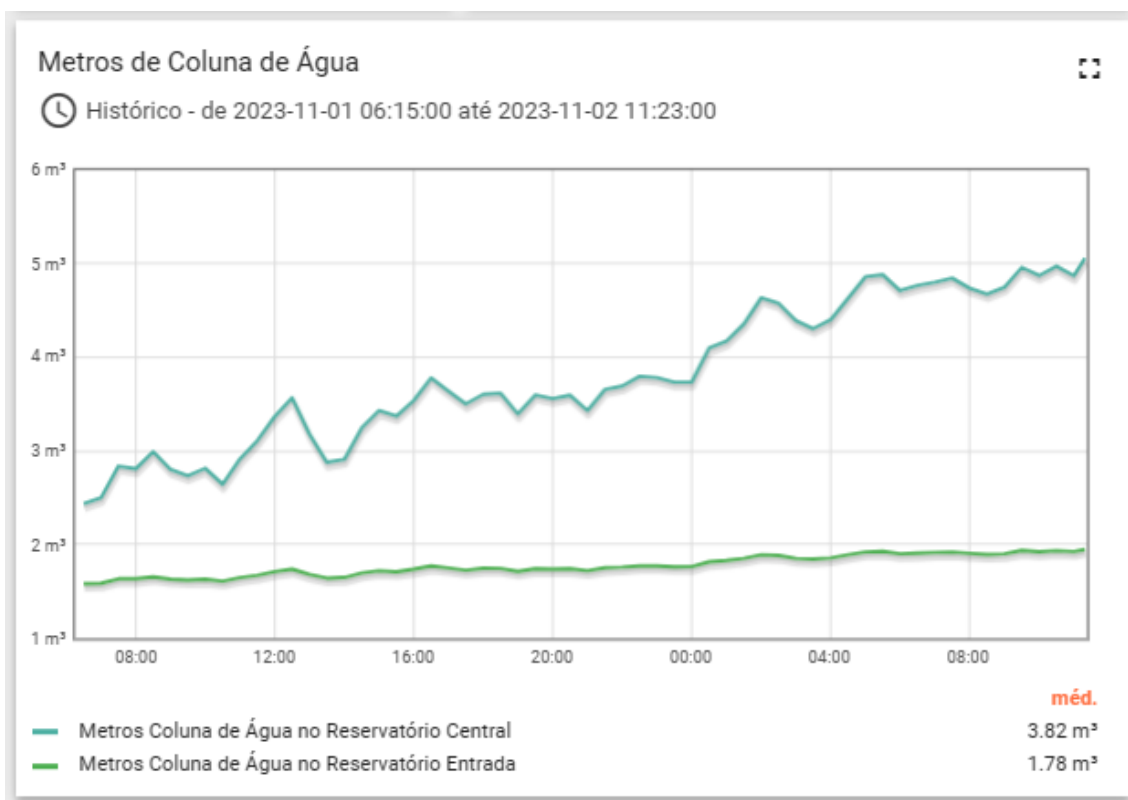
com a integração de mapas Open Street Map. A posição do mapa e dos marcadores são atualizadas conforme os valores numéricos dos campos latitude e longitude de cada dispositivo cadastrado. As coordenadas foram inseridas manualmente e podem ser alteradas na aba “Atributos” localizada na seção “Dispositivos”, tanto na versão para navegador quanto no aplicativo móvel.



**Figura 22. Localização dos reservatórios.**

A Figura 23 apresenta as variações históricas de metros de coluna de água para cada reservatório. Sendo que o eixo x representa o tempo, enquanto o eixo y representa os valores de metros de coluna de água (mca). O gráfico contém duas linhas: uma na cor azul representando os valores de mca do Reservatório Central e uma na cor verde representando os valores de mca do Reservatório Entrada durante as datas de 01/11/2023 às 06 horas e 15 minutos até 02/11/2023 às 11 horas e 23 minutos. O gráfico conta com a funcionalidade de ajustar o intervalo de tempo, seja entre datas específicas ou últimos minutos, últimas horas e últimos dias. Dessa forma, o gráfico se apresentou funcional para a visualização histórica do nível de água, fornecendo dados quantitativos para o monitoramento.

O *widget* para visualização de últimos alarmes, apresentado na Figura 24, lista os



**Figura 23. Gráfico de Metros de Coluna de Água.**

últimos alarmes ativos. Essa interface contém as seguintes informações e funcionalidades:

- Severidade - Indica o nível de severidade do alarme emitido, os valores possíveis são crítica e principal, sendo que o tipo crítica representa um cenário de maior emergência se comparado ao tipo principal. Os termos crítica e principal tratam-se de traduções decorrentes do processo de internacionalização desempenhado pelo ThingsBoard, sendo assim, tais termos são respectivamente traduções diretas das palavras em inglês *Critical* e *Major*.
- Originador - Identifica o dispositivo que gerou o alarme. Para este trabalho, os valores possíveis são Reservatório Entrada e Reservatório Central.
- Problema - Descreve o tipo de alarme. Para este trabalho, os valores possíveis são Excesso de água e Escassez de água.
- Data de criação - Exibe a data e hora em que o alarme foi gerado.
- MCA atual - Mostra a leitura mais recente de metros de coluna de água do reservatório em questão.
- Em uso atual - Expresso em porcentagem, representa o último volume enviado à plataforma.
- Atividade - Permite aos usuários do sistema adicionarem comentários, os comentários realizados são visíveis a todos que possuem acesso ao *dashboard*.
- Detalhes - Mostra informações como duração do alarme, status do alarme que podem assumir os valores de Ativo, Limpo, Confirmado ou Não confirmado).
- Confirmar - Trata-se de um botão que permite ao usuário marcar o alarme como “confirmado”, indicando que ele foi revisado.

- Limpar - Botão que possibilita limpar o alarme depois que o problema foi solucionado.

Severidade	Originador	Problema	Hora de criação ↓	MCA Atual	Em uso atual				
<input type="checkbox"/> Crítica	Reservatório Entrada	Excesso de água	2023-11-02 22:21:15	2.03 m	88.26 %				
<input type="checkbox"/> Crítica	Reservatório Central	Escassez de água	2023-10-29 16:41:54	5.53 m	73.73 %				
<input type="checkbox"/> Principal	Reservatório Entrada	Excesso de água	2023-10-28 22:36:48	2.03 m	88.26 %				

**Figura 24. Últimos alarmes.**

Dessa forma, esse *dashboard* fornece uma visão abrangente do sistema de reservatórios do campus. Visto que os *Widgets* apresentam importantes métricas históricas úteis para o gerenciamento eficiente de recursos hídricos.

#### 4.3.4. Cadeia de Regras

As cadeias de regras, em inglês *rule chains*, é um recurso do ThingsBoard que permite automatizar fluxos lógicos e regras de negócio. Através das cadeias de regras é possível conectar blocos lógicos em sequências que definem ações a serem executadas em resposta a eventos. Nesse sentido, para este trabalho, foi desenvolvida uma cadeia de regra que reage quando uma mensagem de telemetria do tipo *post telemetry* é recebida pelo servidor, esse tipo de mensagem é útil para enviar dados através de dispositivos IoT para servidores web por meio de protocolos como o MQTT e HTTP.

As cadeias de regras são compostas por nós, que são unidades lógicas reutilizáveis, alguns exemplos de nós são: selecionar quais tipos de mensagens irão desencadear uma regra; transformação e conversão de dados; enriquecimento de mensagem e de metadados; integrações com serviços externos; gerenciamento de alarmes.

Neste trabalho, para toda mensagem do tipo *post telemetry* foi definida a seguinte sequência de nós:

1. Enriquecimento de atributos: nesta etapa são recuperados os valores como altura de encanamento e metros de coluna de água, ambos os valores são úteis para emissão de alarme.
2. *Script* de transformação: a transformação ocorre da unidade de medida em mA para dados como: volume, pressão e metros de coluna de água. As Figuras 25 e 26 apresentam os cálculos realizados e parte do código-fonte em JavaScript.
3. *Save Timeseries*: este é um bloco padrão de cadeias de regras do ThingsBoard. Quando esse bloco é acionado, inicia-se o processo de armazenamento dos valores de telemetria. Dessa forma, os dados se tornam persistentes e podem ser

```

function calcularMetrosColunaAgua(miliAmpere, alturaUtil, alturaEncanamento) {
    var alturaTotal = alturaUtil + alturaEncanamento;
    var mca = (miliAmpere - 4) * alturaTotal / 16;
    mca = mca - alturaEncanamento;

    return mca.toFixed(2);
}

msg.mcaAtual = calcularMetrosColunaAgua(msg.miliAmpere,
                                        Number(metadata.ss_alturaUtil),
                                        Number(metadata.ss_alturaEncanamento));

return {msg: msg, metadata: metadata, msgType: msgType};

```

Figura 25. Trecho de código-fonte para obtenção de mca.

```

function calcularPercentualUtilizado(mca, alturaUtil) {
    return (mca / alturaUtil) * 100;
}

function calcularPressaoHidrostatica (mca) {
    return 1000 * 9.81 * mca;
}

function calcularVolume (mca, raio) {
    return Math.PI * Math.pow(raio, 2) * mca;
}

msg.percentualAtual = calcularPercentualUtilizado(msg.mcaAtual, Number(metadata.ss_alturaUtil));
msg.pressaoAtual = calcularPressaoHidrostatica(msg.mcaAtual);
msg.volumeAtual = calcularVolume(msg.mcaAtual, Number(metadata.ss_raio));

return {msg: msg, metadata: metadata, msgType: msgType};

```

Figura 26. Trecho de código-fonte para obtenção de pressão e volume.

consultados posteriormente para análise, visualização de *dashboards*, relatórios e demais funcionalidades relacionadas a estes dados.

4. Conjunto de blocos *Script* e blocos de alarmes: este conjunto de nós tem como objetivo verificar, emitir e limpar alarmes. Nesta etapa, o valor atual de metros de coluna de água é verificado se está dentro do intervalo ideal, se for o caso, os alarmes serão removidos ou limpadados; por outro lado, se o valor estiver fora do intervalo ideal, será emitido um alarme de excesso ou de escassez.

A Figura 27 representa o que foi descrito nos itens 1, 2 e 3 na lista acima. Enquanto a Figura 28 representa o que foi descrito no item 4 da lista.

#### 4.4. Aplicativo

As Figuras 29 e 30 apresentam telas obtidas a partir de um dispositivo móvel *Android*. Tal aplicativo recebe os *Dashboards* da versão *web* redimensionados para resolução e

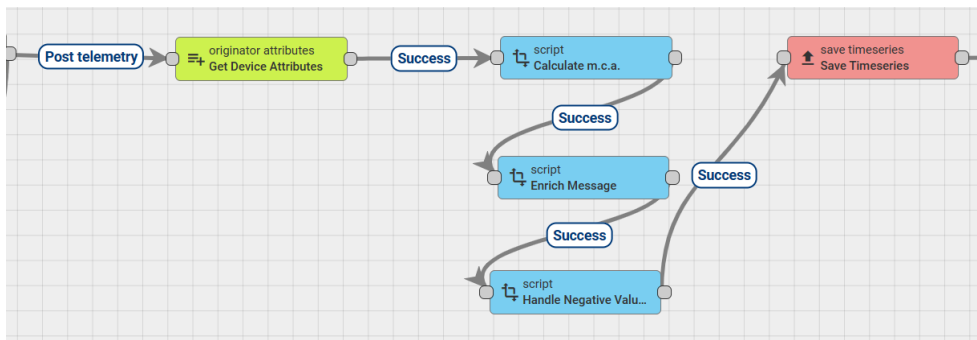


Figura 27. Cadeia de regras para *post telemetry*.

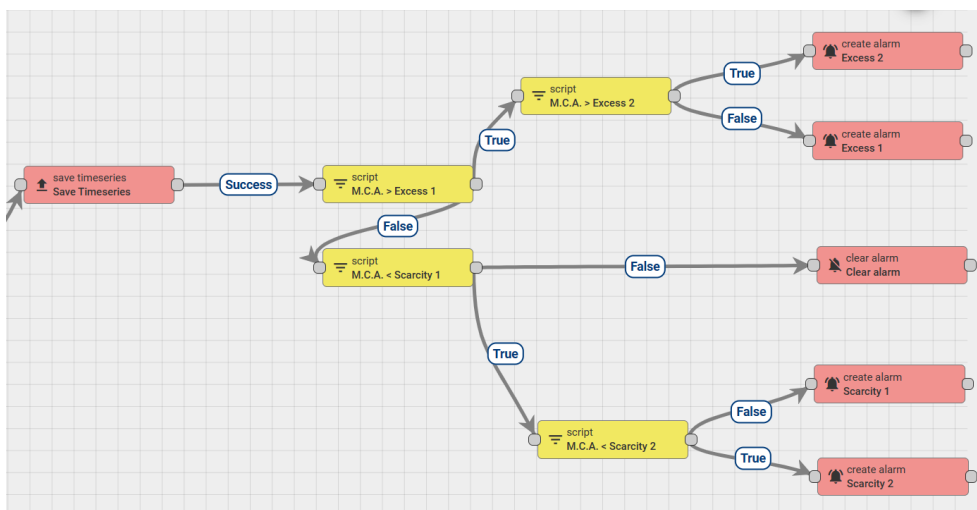


Figura 28. Cadeia de regras para alarmes.

funcionalidade *mobile*. Essas capturas foram registradas durante a navegação de usuário pelo aplicativo.

O aplicativo proporciona agilidade na checagem histórica sobre os níveis de água dos reservatórios, tendo como usuário-alvo os funcionários responsáveis pela gestão hídrica do Campus. As informações ou mudanças feitas no aplicativo refletem na versão *web*.

A versão *mobile* utiliza o *Framework Flutter*, sendo assim proporciona um ambiente de desenvolvimento multiplataforma, que na prática resulta em aplicativos para diferentes sistemas operacionais móveis (*Android* e *iOS*).

## 5. Conclusões e Trabalhos Futuros

Este trabalho alcançou parcialmente seu objetivo principal, uma vez que não foi possível realizar a efetiva instalação dos equipamentos em campo por motivos já descritos anteriormente e que fogem do controle dos autores. No que tange os objetivos específicos, foram concluídos com êxito o desenvolvimento de um *firmware* e sua execução em um microcontrolador NodeMCU EPS8266-12E, a confecção de *dashboards* na plataforma ThingsBoard que integram dados vindos do *firmware* gerando assim o sistema de visualização de dados e o desenvolvimento de uma aplicação móvel que integre o sistema de gestão e

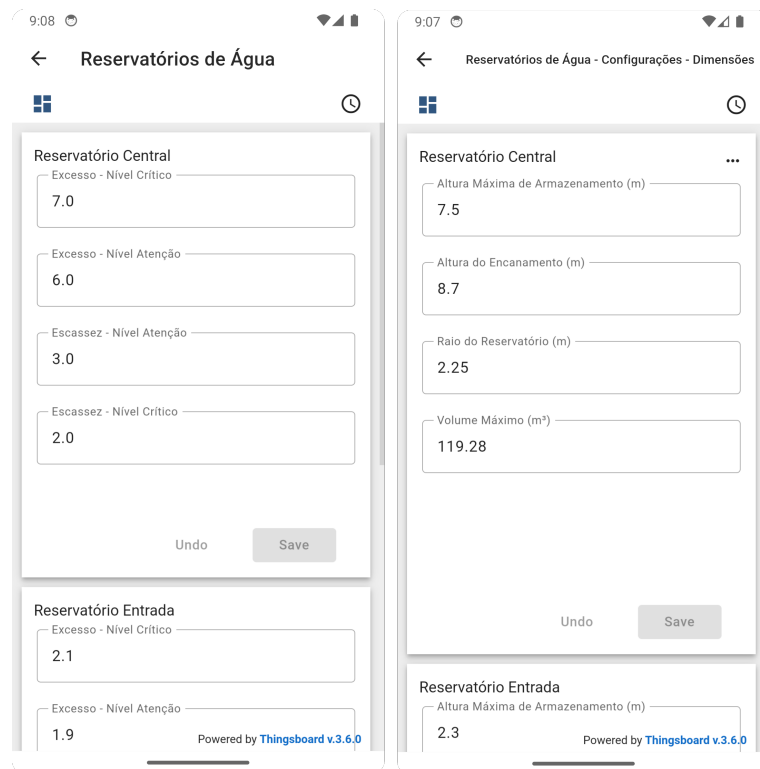


Figura 29. Configuração de Alarmes e Dimensões Estruturais.

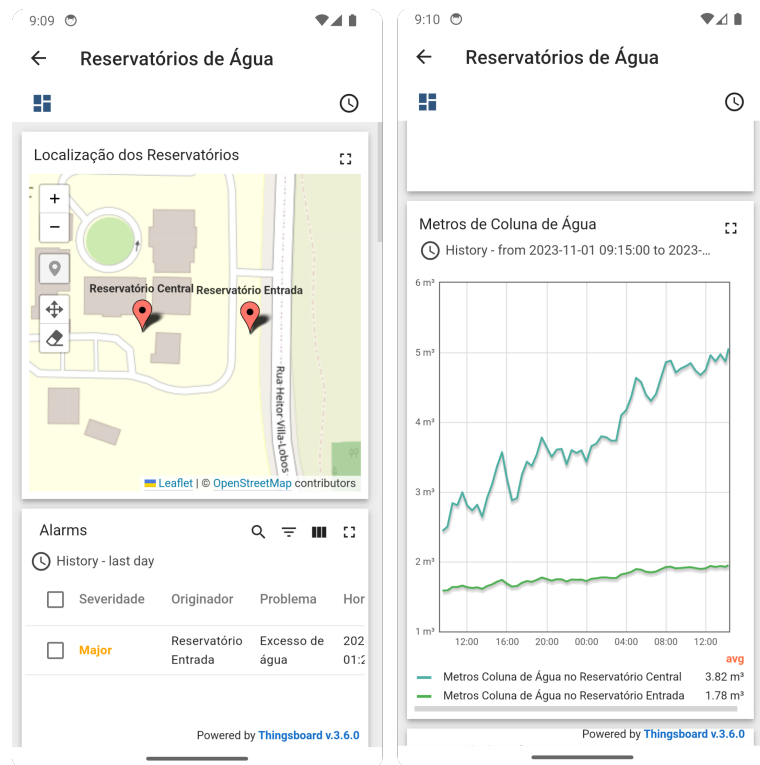


Figura 30. Painel de Visualização dos Reservatórios.

visualização de dados.

A adoção de dispositivos inteligentes agilizou significativamente a coleta de dados, enquanto a plataforma modulada para receber essas informações permitiu uma análise mais eficaz das operações da IoT. Tais resultados foram apresentados de forma clara e intuitiva por meio de *dashboards*, que incluíram gráficos, mapas e alertas bem como interfaces de parametrização do sistema.

Essas conquistas refletem o sucesso em atingir o objetivo macro do projeto. No entanto, é importante observar que as simulações utilizadas para demonstrar o funcionamento do aplicativo, *firmware* e *dashboards* foram baseadas em dados estruturais dos reservatórios de água, os quais foram obtidos a partir de informações disponíveis, como as plantas dos mesmos.

Como trabalhos futuros, sugerimos a implantação dos equipamentos na estrutura física do reservatório, instalando todos os dispositivos informados nesse documento, isso traria a prova real de todas as simulações efetuadas nesse trabalho, bem como traria todos os benefícios citados na problemática.

Outro avanço sugerido é a implementação de um método ajuste no dispositivo para permitir ligar e desligar a bomba de água dos reservatórios, assim trazendo controle para uma solução que agora é de monitoramento. Tal melhoria, faz com que em casos de vazamentos detectados, ou até mesmo qualquer tipo de manutenção necessária seja possível realizar esta operação de forma remota.

Por fim, sugerimos também a instalação de fontes ininterruptas de energia, popularmente conhecidas por *nobreak*, para oferecer maior proteção e funcionamento contínuo em situações de oscilação ou ausência de energia proveniente da rede elétrica. Relativo ao roteamento Wi-Fi do campus, este fica conectado via *nobreak* apenas por 30 minutos diante de falta de energia, tendo isso em vista, sugerimos a instalação de uma rede de telefonia celular para que se torne possível disparar notificações via SMS para o usuário ser notificado no aplicativo e na versão web.

## Referências

- Abuarqoub, A., Abusaimh, H., Hammoudeh, M., Uliyan, D., Abu-Hashem, M. A., Murad, S., Al-Jarrah, M., e Al-Fayez, F. (2017). A survey on internet of things enabled smart campus applications. In *Proceedings of the International Conference on Future Networks and Distributed Systems, ICFNDS '17*, New York, NY, USA. Association for Computing Machinery.
- Angrisani, L., Arpaia, P., Bonavolontà, F., e Schiano Lo Moriello, R. (2018). Academic fablab at university of naples federico ii: New research and development opportunities in the fields of iot and industry 4.0. In *2018 Workshop on Metrology for Industry 4.0 and IoT*, pages 23–27.
- Atzori, L., Iera, A., e Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.
- Bassi, L. (2017). Industry 4.0: Hope, hype or revolution? In *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–6.
- Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., e Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214:481–518.

- Botta, A., de Donato, W., Persico, V., e Pescapé, A. (2014). On the integration of cloud computing and internet of things. In *2014 International Conference on Future Internet of Things and Cloud*, pages 23–30.
- Chen, J. e Zhou, J. (2018). Revisiting industry 4.0 with a case study. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1928–1932.
- Dong, Z. Y., Zhang, Y., Yip, C., Swift, S., e Beswick, K. (2020). Smart campus: definition, framework, technologies, and services. *IET Smart Cities*, 2(1):43–54.
- Fortin, M. F., Côté, J., e Filion, F. (2009). *Fundamentos e Etapas do Processo de Investigação*. Lusodidacta.
- Founoun, A., Hayar, A., e Haqiq, A. (2019). The textual data analysis approach to assist the diagnosis of smart cities initiatives. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 150–153.
- Geihs, K., Baraki, H., e de la Oliva, A. (2020). Performance analysis of edge-fog-cloud architectures in the internet of things. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pages 374–379.
- Gislam, S. (2021). Industry 4.0 - fourth time's the charm? Disponível em: <https://industryeurope.com/sectors/technology-innovation/industry-4-0-fourth-times-the-charm/>. Acesso em 09 fev. 2022.
- Kaur, J. e Kaur, K. (2017). Internet of things: A review on technologies, architecture, challenges, applications, future trends. *International Journal of Computer Network & Information Security*, 9(4).
- Khan, R., Khan, S. U., Zaheer, R., e Khan, S. (2012). Future internet: The internet of things architecture, possible applications and key challenges. In *2012 10th International Conference on Frontiers of Information Technology*, pages 257–260.
- Kleeberger, V. B., Rutkowski, S., e Coppens, R. (2015). Design verification of automotive soc firmware. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6.
- Krishnaveni, M., Praveen Kumar, S. K., Arul Muthusamy, E., Kowshick, J., e Arunya, K. G. (2020). Real-time monitoring of water level and storage dynamics of irrigation tank using IoT. *H2Open Journal*, 3(1):392–400.
- Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., e Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering*, 6:239–242.
- Leite, J. E., Martins, P. S., e Ursini, E. L. (2017). A internet das coisas (iot): Tecnologias e aplicações. *School of Technology, University of Campinas (UNICAMP)*.
- Mazhelis, O. e Tyrväinen, P. (2014). A framework for evaluating internet-of-things platforms: Application provider viewpoint. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 147–152.
- Olisa, S. C., Asiegbu, C. N., Olisa, J. E., Ekengwu, B. O., Shittu, A. A., e Eze, M. C. (2021). Smart two-tank water quality and level detection system via iot. *Heliyon*, 7(8):e07651.
- Ortiz, G., Zouai, M., Kazar, O., de Prado, A. G., e Boubeta-Puig, J. (2022). Atmosphere: Context and situational-aware collaborative iot architecture for edge-fog-cloud computing. *Computer Standards Interfaces*, 79:103550.
- Patel, K. K., Patel, S. M., e Scholar, P. (2016). Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5).

- Provost, F. e Fawcett, T. (2013). *Data Science for Business*. O'Reilly.
- Ribeiro, A. d. F. (2015). Taylorismo, fordismo e toyotismo. *Lutas Sociais*, 19(35):65–79.
- Sakurai, R. e Zuchi, J. D. (2018). As revoluções industriais até a indústria 4.0. *Revista Interface Tecnológica*, 15(2):480–491.
- Saravanan, S., Renugadevi, N., Sudha, C. N., e Tripathi, P. (2021). *Industry 4.0: Smart Water Management System Using IoT*, pages 1–14.
- Sethi, P. e Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017.
- Shah, P. P., Patil, A. A., e Ingleshwar, S. S. (2017). Iot based smart water tank with android application. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 600–603.
- Silva, B. N., Khan, M., e Han, K. (2018). Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical Review*, 35(2):205–220.
- Silveira, W. e Dantas, M. (2022). Ambiente domiciliar assistido para o monitoramento remoto de idosos em zonas rurais. In *Anais Estendidos do XXIII Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 1–8, Porto Alegre, RS, Brasil. SBC.
- Tschofenig, H., Arkko, J., Thaler, D., e McPherson, D. R. (2015). Architectural Considerations in Smart Object Networking. RFC 7452.
- United Nations, D. o. E. e Development, S. A. S. (2015). Transforming our world: the 2030 agenda for sustainable development.
- Yang, J., Kwon, Y., e Kim, D. (2021). Regional smart city development focus: The south korean national strategic smart city program. *IEEE Access*, 9:7193–7210.