

Sistema de Monitoramento Automatizado por Meio de Dispositivo Embarcado de Baixo Custo.

Felipe Schubert Costa, Gabriel Araujo, Kele Belloze¹,
Sanderson Gonzaga de Oliveira², Luis Barbosa de Assis Junior¹,
Raphael Guerra³, Diego Brandão¹

¹Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
Rio de Janeiro – RJ – Brasil

²Departamento de Ciência da Computação – Universidade Federal de São Paulo
São José dos Campos – SP – Brasil.

³Instituto de Computação
Universidade Federal Fluminense (UFF) – Niterói, RJ – Brasil

{gabriel.araujo, diego.brandao, }@cefet-rj.br, rguerra@ic.uff.br

Abstract. *The growing use of the IoT concept and the tools that integrate this technology, such as WSNs (Wireless Sensor Networks), decreased the aggregate cost of the devices that provide this technology. However, integrating this technology in the management of sensors in the field is still an implementation challenge. This work presents the construction of an architecture model responsible for implementing an automated monitoring system of embedded devices and constructing a simple interface for the final user. The evaluation of this proposal is done through the implementation of a water quality monitoring system.*

Resumo. *Com o crescente uso do conceito de IoT e das ferramentas que integram essa tecnologia, como as RSSF (Rede de Sensores Sem Fio), o custo agregado dos dispositivos que a disponibilizam tem caído gradativamente. Todavia, a integração dessa tecnologia no gerenciamento de sensores no campo ainda é um desafio de implementação. Este trabalho apresenta a construção de um modelo de arquitetura, responsável pela implementação de um sistema de monitoramento automatizado de dispositivos embarcados, bem como a construção de uma interface simples para o usuário final. A arquitetura é avaliada por meio da implementação de um sistema de monitoramento de qualidade de água.*

1. Introdução

A Internet das Coisas consiste em uma extensão da Internet para o mundo físico, interligando objetos do cotidiano como carros, alarmes, eletrodomésticos e outros objetos em geral [Santos et al. 2017]. Segundo [Pires et al. 2017], IoT também contempla o universo dos sensores e atuadores interligados à uma rede sem fio controlados pela Internet, criando uma rede complexa capaz de processar e capturar variáveis de ambiente e responder a estímulos do meio.

Apesar da evolução dessa tecnologia, sua integração e gerenciamento em campo ainda é um desafio [Othman and Shazali 2012, Rastegari et al. 2023]. Para se obter uma estrutura autoconfigurável, escalável, com capacidade dinâmica de expansão

e com protocolos de comunicação com interoperabilidade, onde as coisas são objetos físicos ou virtuais, utiliza-se a uma plataforma de software definida como um middleware [Bandyopadhyay et al. 2011]. Algumas das razões de se utilizar IoT-middleware consiste em prover soluções satisfatórias relacionadas a crescente variedade de dispositivos associados a aplicações para permitir interoperabilidade fornecendo de uma forma padrão o acesso aos dados e serviços para os elementos conectados a rede [Pires et al. 2017]. Existem diversos middlewares no mercado hoje, exemplos são o ThingsBoard, o NodeRed, Flogo e OpenIoT [Terracher et al. 2023, Bestari and Wibowo 2023, Prompt et al. 2023]. Cada um desses sistemas possui características específicas, permitindo visualização de dados, Edge computing e/ou Fog computing, além de outras funcionalidades.

Apesar da diversidade das plataformas de IoT, raramente é possível a exploração de dados de redes de sensores em uma única ferramenta, devido à diversidade de dispositivos em redes de grande escala [Terracher et al. 2023]. Sem mencionar nas dificuldades oriundas da gestão dos sensores em locais remotos [Othman and Shazali 2012].

Neste contexto, este trabalho apresenta uma proposta de sistema IoT para monitoramento ambiental automatizado que permite agregar diferentes tipos de sensores. A avaliação do sistema é feita utilizando um estudo de caso para o monitoramento de qualidade de água por meio de dispositivos de baixo custo.

O presente trabalho está organizado da seguinte maneira. Uma breve revisão sobre artigos focados em sistemas de monitoramento de água é apresentada na Seção 2. Em seguida, a Seção 3 apresenta a fundamentação teórica de sistemas de monitoramento, sensores e comunicação de dados. A Seção 4 apresenta o sistema proposto com sua versão estática e dinâmica, descrevendo os elementos de hardware que compõem o sistema. A avaliação experimental da proposta deste trabalho é apresentada na Seção 5. Por fim, a Seção 6 apresenta as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

O interesse no desenvolvimento de sistemas para monitoramento da qualidade de água cresceu consideravelmente nos últimos anos, surgindo inúmeras pesquisas sobre o assunto. Diversas pesquisas focam em sistemas para monitoramento de aquários ou criadouros em cativeiros [Han and Noor 2023, Amaliah 2023], outras pesquisas para o monitoramento de piscinas e reservatórios [Yusri et al. 2023] enquanto outras desenvolvem sistemas para o monitoramento de rios e lagos [Misra 2023].

Ainda existem trabalhos focando em aspectos de comunicação de dados, avaliando protocolos de comunicação e tecnologias a serem utilizadas [Prompt et al. 2023, Jáquez et al. 2023]. Abordagens que utilizam análise de dados sobre os dados oriundos de tais sistemas também vem sendo realizadas [Chhipi-Shrestha et al. 2023]. Por fim, o presente trabalho não visa apresentar uma revisão detalhada sobre artigos focados em sistemas para monitoramento de qualidade de água. Mais informações sobre o assunto podem ser obtidas em [Pasika and Gandla 2020, Rozario et al. 2022].

3. Fundamentação Teórica

Como mencionado, a Internet das Coisas contempla também redes de sensores e atuadores controlados pela Internet, sendo esse controle realizado por meio de software. No

contexto de hardware, essa rede de sensores e atuadores consiste em um conjunto de dispositivos capazes de adquirir dados e transmiti-los para um elemento concentrador ou a outros dispositivos da rede. Uma versão simplificada pode ser constituída por dois elementos fundamentais, os nós e os gateways. Os nós são elementos que, mediante a uma solicitação, recolhem os dados dos sensores utilizando seus respectivos protocolos e ao final de cada leitura enviam os dados para o dispositivo que fez a solicitação. Eles também são capazes de fazer a identificação única de cada sensor para aprimorar o controle no nível de gerenciamento pelo software.

Já os gateways consistem em dispositivos responsáveis por interligar duas ou mais interfaces de redes com arquiteturas distintas. Eles desempenham o papel de se comunicar com os nós, fazer a requisição e o recolhimento dos dados na rede e retransmiti-los para um servidor [Ruiz-Garcia et al. 2009].

3.1. Comunicação de dados

Outro aspecto fundamental em um sistema IoT é a comunicação, que pode envolver outras tecnologias além da Internet. Existem diversos protocolos que podem ser utilizados para garantir a troca de informação entre dispositivos, um deles é o protocolo ZigBee (IEEE 802.15.4) [Malik and Kushwah 2023]. Este protocolo visa aplicações com finalidade de economia de bateria, além da transmissão de dados [Litjens 2009]. Os dispositivos de uma rede ZigBee podem ser divididos em dispositivos de função completa e dispositivos de funções reduzidas. O primeiro diz respeito a dispositivos com maior complexidade e que podem se comunicar com gateways (coordenadores) e nós (roteadores). O segundo se refere a dispositivos na borda da rede, que visam otimização de energia (terminais).

3.2. Aspectos de Software

As técnicas de computação consistem nos métodos e tecnologias utilizadas para o desenvolvimento de software e das rotinas nos elementos de rede. Elas podem ser baseadas em técnicas de IoT, de computação em Nuvem, em Névoa, em Borda ou combinações dessas [Mell and Grance 2011, Patil 2015, Bourechak et al. 2023].

A tecnologia de computação na nuvem para sistemas de RSSF desempenha um importante papel no quesito de persistência dos dados. Um dos fatores que torna a essa tecnologia relevante para a arquitetura de redes de sensores sem fios é a possibilidade de deixar toda a parte de processamento de dados brutos ser realizada na nuvem onde a aplicação está hospedada, ou seja, não é necessário consumir recursos de processamento, nem mesmo uma interface para disponibilizar os dados para o usuário no nível da RSSF.

Já a computação em Névoa consiste em uma tecnologia que implementa o processamento, armazenamento e serviços de rede entre os dispositivos da rede e a nuvem. Os dispositivos da rede podem implementar a computação em Borda, realizando processamentos locais, como a aplicação de filtros nos dados [Bourechak et al. 2023].

4. Sistema Proposto

O sistema proposto é composto por dois elementos, os nós e os gateways, implementados em duas arquiteturas. As arquiteturas do sistema consistem nas diferentes formas de como os dados monitorados podem ser recepcionados pela aplicação. Além disso, a aplicação de software desenvolvida abarca aspectos de computação em Névoa (Fog Computing) e de Borda (Edge Computing), sendo hospedada em um servidor em Nuvem.

4.1. Modelo do Sistema

A fim de elaborar uma estrutura fixa que possa ser utilizada pelos usuários de forma remota e sem a necessidade de estar próximo aos sensores, foi construído um modelo de rede estruturado que captura os dados de campo e os disponibiliza ao usuário final. O esquema da Figura 1 demonstra o fluxo de dados nesta rede.

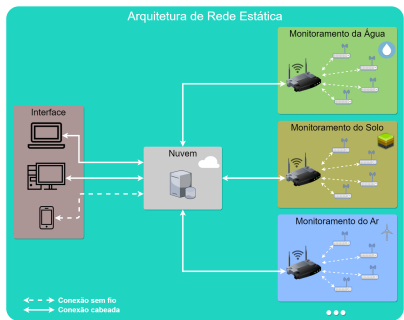


Figura 1. Modelagem estática do sistema.

Os sistemas são separados por blocos que podem ser de diversos tipos: Monitoramento da água, do solo, do ar, da poluição sonora, do consumo de energia, do nível de intensidade de luz, híbridos e dentre outros. O tipo de cada sistema é definido de acordo com os objetivos de cada nó, por exemplo, um sistema com sensores de oxigênio dissolvido, temperatura e salinidade, demonstram um interesse de se saber parâmetros de ambientes aquáticos.

Os Nós são dispositivos concentradores construídos para agrupar os dados de cada sensor. Eles são responsáveis por verificar se os dados estão íntegros e implementam o pré-processamento antes de enviar os dados dos sensores para o gateway utilizando algoritmos que detectam variações, ruídos e caso necessário refazem a requisição, conforme conceitos de Edge Computing. O modelo de nó desenvolvido aqui é apresentado na Figura 2. Nele é possível inserir até 5 sensores digitais, 3 sensores analógicos previamente configurados e até 127 sensores com compatibilidade I2C que podem ser configurados pela interface com o usuário. O consumo dos sensores no nó estão limitados ao fornecimento de energia do próprio, que pode ser modificada para atender a demanda de novos sensores.

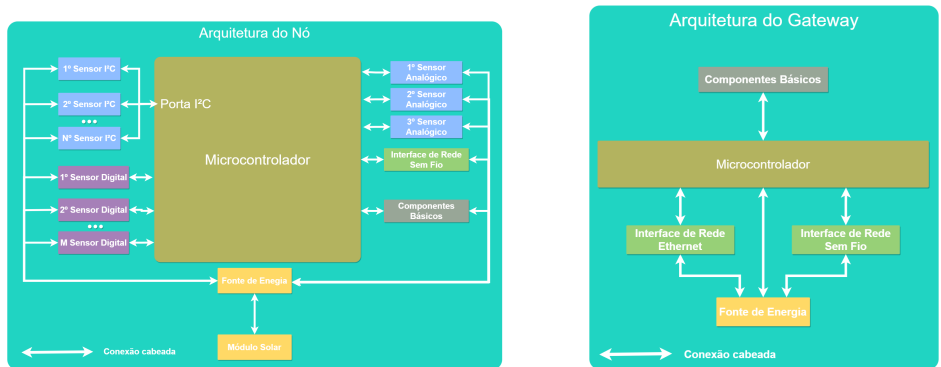


Figura 2. Modelo do Nó e Modelo do Gateway

Os gateways são dispositivos de rede capazes de converter dados de uma interface

de rede sem fio no protocolo Zigbee para uma interface de rede ethernet e vice-versa. Os gateways também são capazes de implementar o roteamento de requisições para nós específicos na rede. A Figura 2 também apresenta a arquitetura do gateway desenvolvido neste trabalho. Os componentes básicos permitem que o microcontrolador ATmega328 trabalhe de forma adequada e efetue a conversão dos dados para os módulos corretamente. A nuvem é responsável pela comunicação com os gateways fazendo as requisições em períodos de tempo bem definidos e persistindo-os no banco de dados, além de hospedar o sistema para acesso do usuário.

Outra arquitetura implementada consiste na abordagem dinâmica, nela é possível utilizar devices para adquirir os dados de cada nó individualmente (Figura 3).

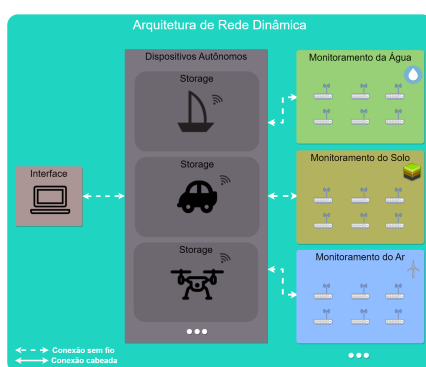


Figura 3. Modelagem dinâmica do sistema.

A arquitetura física do sistema permite o envio de pacotes de dados tanto na arquitetura de rede estática quanto na dinâmica, sem a necessidade de alterar os elementos da rede. No caso da arquitetura dinâmica, não existe a figura de um Gateway de rede para intermediar o fluxo de dados para uma rede ethernet. Essa responsabilidade é entregue aos storages que ficam embarcados nos devices. Os storages são elementos da rede que podem ser embarcados em um device, se comunicam com os nós e armazenam os dados enviados por eles em sua memória interna.

O fluxo de dados segue quando um device, que é controlado por um indivíduo remotamente, se aproxima de um ou mais nós da rede. O device possui um circuito embarcado que é capaz de disparar uma requisição para o nó adquirindo os dados do mesmo. Após receber os dados, eles são armazenados em sua memória interna. Após efetuar a leitura dos dados, o device retorna a base e é possível remover o cartão de memória e acessar os dados em um computador de forma crua ou importá-los para a aplicação. O Storage se comporta como um gateway móvel que possibilita a recepção dos dados em campo sem a necessidade de uma arquitetura de rede com a figura de um servidor fixo. Todos os dados podem ser visualizados de forma crua ou importados para a interface após serem recolhidos do SD card.

4.2. Hardware desenvolvido

Em relação aos dispositivos (things), consistindo especificamente dos Nós e Gateways, eles foram desenvolvidos utilizando a plataforma Arduino. O *gateway* estático pode ser visto na Figura 4.



Figura 4. Gateway estático desenvolvido.

A Figura 5 apresenta o Nó para monitoramento de água, ele é responsável por capturar os dados advindos dos sensores de pH, condutividade da água e do GPS. Os dois sensores de qualidade água utilizados foram desenvolvidos pela Atlas Scientific [Rao et al. 2013].



Figura 5. Nó para Monitoramento de Qualidade de Água.

4.3. Aplicação de Software

A aplicação de software é responsável por fazer a requisição e recepcionar os dados da rede, bem como gerar os relatórios e disponibilizar os dados para o usuário final. A interface de gerenciamento de dados disponibiliza os dados para o usuário de forma amigável através de um dashboard. Com ele, o usuário é capaz de verificar o recebimento dos dados de cada nó através de widgets, permitindo gerar relatórios pelo acesso aos elementos do nó e também configurando novos gateways, nós e sensores na rede. Os dados exibidos para o usuário nessa aplicação não são provenientes diretamente de campo, eles são armazenados e processados pelo servidor na nuvem.

4.3.1. Serviço de requisição

O serviço de requisição é uma aplicação C a parte responsável por popular o banco de dados que o dashboard irá consumir. Esse serviço, através dos protocolos TCP/IP com os gateways, faz periodicamente a requisição dos dados em cada um dos nós e assim popula o banco de dados que fica disponível no servidor. Esse serviço também é passível de

configuração através do dashboard para definir a taxa de requisição e quais gateways e nós serão lidos.

Os intervalos de coleta de dados são definidos manualmente para cada tipo específico de sensor e de acordo com o meio que está inserido, por exemplo, a frequência de leitura de dados em um sensor de temperatura deve ser fixada em um período maior do que a de um sensor de condutibilidade da água quando o ambiente em que estes estão inseridos é um rio. Pois temperatura é um parâmetro que geralmente não muda abruptamente, ou seja, pode ser coletada uma vez a cada 5 minutos, em quanto que a condutância pode ser alterada rapidamente quando algum material for despejado próximo ao sensor e sua concentração em um rio será dissipada de forma rápida.

5. Avaliações Experimentais

O sistema foi implementado em linguagem TypeScript enquanto os sensores, nós e gateways utilizaram as linguagens C e C++, o banco de dados utiliza MySQL. Todos os códigos podem ser obtidos no github do laboratório¹.

Em relação ao custo do sistema, analisando apenas os dispositivos hardware para construção de um Gateway e um Nó, observa-se que os maiores valores são relativos aos sensores profissionais utilizados² e aos componentes de comunicação³, sendo os primeiros com um custo de cerca de 500 dólares e o sistema de comunicação de 100 dólares. Todos os demais componentes são baseados no Arduino, sendo obtidos a um valor menor do que 100 dólares⁴. Tal proposta tem custo bem inferior aos sistemas profissionais, como o da empresa Ag Solve⁵ com sensores de mais de dois mil dólares.

5.1. Interface do sistema

O desenvolvimento da interface seguiu o estilo *dashboard* com a utilização de *widgets* como já mencionado. O framework *ngx-admin* foi utilizado e modificado para fornecer os estilos e a correta disposição dos *widgets* na tela. A tela principal consiste nos *gateways* configurados no sistema (Figura 6).



Figura 6. Tela principal do *dashboard*.

Ao clicar em um dos *widgets*, é possível acessar os seus respectivos nós. O usuário pode posteriormente acessar ao nó de “Monitoramento da Qualidade da Água”, como apresentado na Figura 7. O usuário pode escolher o sensor que deseja monitorar.

Ao acessar o sensor de condutividade elétrica, é possível verificar as medições de cada um dos parâmetros do sensor, a saber: salinidade da água (‰), condutividade

¹<https://github.com/SCICOM-CEFET-RJ/Monitoring>

²<https://atlas-scientific.com/>

³<https://www.digi.com/products/embedded-systems/digi-xbee>

⁴<https://www.huinfinito.com.br/>

⁵<https://www.agsolve.com.br/>



Figura 7. Tela de monitoramento dos sensores.

elétrica ($\mu S/cm$) e sólidos dissolvidos totais (TDS). Um exemplo de medição de salinidade pode ser verificado na Figura 8. Além dos parâmetros relativos as medições do sensor, também é possível verificar sua posição geográfica (Figura 9).



Figura 8. Tela dos dados do sensor de condutividade.

A captura dos dados que alimentam a interface pode ser feita de duas formas. A primeira é utilizando a arquitetura estática do sistema. No entanto, o usuário não precisa tomar nenhuma ação para obter os dados, todo o processo é feito de forma automática desde que a arquitetura de rede esteja devidamente configurada. A segunda forma consiste em utilizar a arquitetura dinâmica do sistema. No entanto, faz-se necessário o usuário importar os dados para dentro da interface de gerenciamento de dados, afim de persistir esses dados, essa funcionalidade está disponível no menu lateral da aplicação.

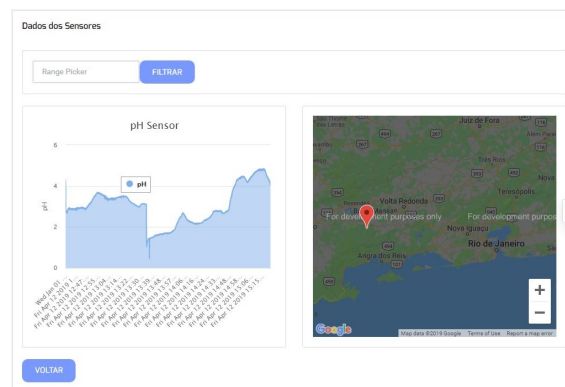


Figura 9. Tela dos dados do sensor de pH.

5.2. Avaliação de computação em borda

A implementação em edge computing permitiu melhorar a qualidade dos dados reduzindo o tráfego de informação incorreta. Um dos sensores que mais apresentou dificuldade em se adquirir dados consistentes foi o de condutividade elétrica da água. A Figura 10 é um exemplo de como os dados advindos desse sensor chegavam no microcontrolador do nó antes e depois da conformação de dados.

330403326533041,17842,21	2154,17363,20
33041,17842,21	32595,17602,21
31500,17010,20	31500,17010,20
31500,17010,20	31284,16894,20
31284,16894,2031935,17245,20	31284,16894,20
33265,17964,21	31935,17245,20
33041,178031500,17010,20	31284,16894,20
31500,17010,20	31284,16894,20
32374,17482,21	31284,16894,20
31500,17010,20	31284,16894,20
33265,17964,21	32817,17722,21
31500,17010,2032817,17722,21	32595,17602,21
333041,17842,21	31935,17245,20
31500,332500,17010,20	31284,16894,20
33041,17842,21	32154,17363,20
32817,17722,21	31284,16894,20
3333265,17964,21	32595,17602,21
332817,17722,232532817,17722,21	32595,17602,21
33041,17842,21	32817,17722,21
32817,17722,21	
32595,17602,21	

Figura 10. Dados de condutividade elétrica. Sem conformação de dados (esq.) V.S. Com conformação de dados (dir.)

6. Considerações Finais

O presente trabalho teve por objetivo automatizar os processos para obtenção de dados em meios físicos com um sistema de baixo custo, a fim de aprimorar o fluxo de coleta de dados em ambientes, o que levou a produção de dois módulos concentradores de dados (gateways dinâmico e estático) e um módulo para monitorar a qualidade da água. De forma geral, foi possível demonstrar que os diferentes sensores comerciais, como das empresas Atlas Scientific e Sparkfun; podem ser utilizados. Essa solução pôde ser gerada utilizando componentes de baixo custo, apresentando uma flexibilidade que permite agregar novos tipos de sensores, como aqueles para monitorar a qualidade do ar, além de outros eventos naturais.

Como trabalho futuro, podem ser utilizados novos sensores para monitorar a qualidade da água. Além disso, uma avaliação em um ambiente real, como um rio ou lago, deverá ser realizada, além de estudos sobre os melhores materiais para encapsulamento dos componentes eletrônicos. Um outro aspecto a ser aprimorado, consiste na avaliação do módulo ethernet ENC28J60 que pode permitir maior robustez da rede. Por fim, a utilização do protocolo LoRa pode garantir um alcance mais abrangente e ainda baixo consumo de energia para o sistema. A análise dos dados gerados por meio de algoritmos de estatística e aprendizado de máquina também será realizada.

Referências

- Amaliah, F. I. (2023). Water quality level for shrimp pond at probolinggo area based on fuzzy classification system. *Jurnal Rekayasa Elektrika*, 19(1).
- Bandyopadhyay, S. et al. (2011). Role of middleware for internet of things: A study. *International Journal of Computer Science & Engineering Survey (IJCES)*.
- Bestari, D. N. and Wibowo, A. (2023). An iot-based real-time weather monitoring system using telegram bot and thingsboard platform. *International Journal of Interactive Mobile Technologies*, 17(6).
- Bourechak, A. et al. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors*, 23(3):1639.
- Chhipi-Shrestha, G. et al. (2023). Digital water: artificial intelligence and soft computing applications for drinking water quality assessment. *Clean Technologies and Environmental Policy*, pages 1–30.

- Han, L. C. and Noor, I. (2023). Automatic aquarium water change system with real-time monitoring through iot. *Journal of Applied Technology and Innovation*, 7(2):11.
- Jáquez, A. et al. (2023). Extension of lora coverage and integration of an unsupervised anomaly detection algorithm in an iot water quality monitoring system. *Water*, 15(7):1351.
- Litjens, O. J. (2009). Automação de estufas agrícolas utilizando sensoriamento remoto e o protocolo zigbee.
- Malik, A. and Kushwah, R. (2023). Energy-efficient scheduling in iot using wi-fi and zigbee cross-technology. *The Journal of Supercomputing*, pages 1–30.
- Mell, P. and Grance, T. (2011). The nist definition of cloud computing.
- Misra, G. (2023). Iot based river water quality monitoring. *Computational Algorithms and Numerical Dimensions*.
- Othman, M. F. and Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. In *International Symposium on Robotics and Intelligent Sensors 2012*, pages 1204–1210, Amsterdã.
- Pasika, S. and Gandla, S. T. (2020). Smart water quality monitoring system with cost-effective using iot. *Heliyon*, 6(7):e04096.
- Patil, P. V. (2015). Fog computing. *International Journal of Computer Applications*.
- Pires, P. et al. (2017). Plataformas para a internet das coisas.
- Prompt, S. et al. (2023). Design and analysis performance of iot-based water quality monitoring system using lora technology. *Network*, 3:16.
- Rao, A. S. et al. (2013). Design of low-cost autonomous water quality monitoring system. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 14–19. IEEE.
- Rastegari, H. et al. (2023). Internet of things in aquaculture: A review of the challenges and potential solutions based on current and future trends. *Smart Agricultural Technology*.
- Rozario, A. R. et al. (2022). Review of water quality monitoring using internet of things (iot). *International Journal of Research and Analytical Reviews*.
- Ruiz-Garcia, L. et al. (2009). A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends.
- Santos, B. et al. (2017). Internet das coisas: da teoria à prática.
- Terracher, A. et al. (2023). Model-driven engineering to ensure interoperability in iot data exploitation. *Available at SSRN*.
- Yusri, W. M. E. W. M. et al. (2023). Iot-based water quality monitoring system and test for swimming pool water physicochemical quality. In *AIP Conference Proceedings*, volume 2609. AIP Publishing LLC.