



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

MATHEUS FERNANDES DO NASCIMENTO DANTAS

**UM SISTEMA DE TELEMONITORAMENTO E AUTOMAÇÃO BASEADO EM
REDE LORA PARA CRIAÇÃO DE CAMARÃO.**

QUIXADÁ

2020

MATHEUS FERNANDES DO NASCIMENTO DANTAS

UM SISTEMA DE TELEMONITORAMENTO E AUTOMAÇÃO BASEADO EM REDE
LORA PARA CRIAÇÃO DE CAMARÃO.

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Paulo Armando Cavalcante Aguiar

QUIXADÁ

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D214s Dantas, Matheus Fernandes do Nascimento.

Um sistema de telemonitoramento e automação baseado em rede LoRa para criação de camarão / Matheus Fernandes do Nascimento Dantas. – 2020.
65 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Engenharia de Computação, Quixadá, 2020.

Orientação: Prof. Dr. Paulo Armando Cavalcante Aguilár.

1. Camarão. 2. LoRa. 3. Automação. I. Título.

CDD 621.39

MATHEUS FERNANDES DO NASCIMENTO DANTAS

UM SISTEMA DE TELEMONITORAMENTO E AUTOMAÇÃO BASEADO EM REDE
LORA PARA CRIAÇÃO DE CAMARÃO.

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 22 / 10 / 2020

BANCA EXAMINADORA

Prof. Dr. Paulo Armando Cavalcante
Aguilar (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Ms. Roberta Dutra de Andrade
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcio Espíndola Freire Maia
Universidade Federal do Ceará (UFC)

Em memória de Francisco Abreu Pinheiro, meu
avô "Chico", por fazer o homem que sou.

AGRADECIMENTOS

A Deus, por ter permitido que eu tivesse saúde e determinação para não desanimar durante a realização deste trabalho.

Sou grato à minha família pelo apoio que sempre me deram durante toda a minha vida. Em especial a minha mãe Daniela. Seus ensinamentos me mantiveram no bom caminho, sua dedicação ímpar me sustentou nos momentos mais difíceis e seu amor me deu a motivação para seguir.

A minha futura esposa Larissa Brito pelo amor, pelo incentivo nos momentos mais difíceis e por compreender a minha ausência enquanto eu me dedicava à realização deste trabalho.

Ao Prof. Dr. Paulo Armando pela orientação neste trabalho, conselhos, motivação e por sua amizade em bons e maus momentos.

A Roberta Dutra, pelas correções, ensinamentos, companheirismo e apoio em meus projetos profissionais e em meu desenvolvimento pessoal. Obrigado por tudo coração!

Ao Prof. Dr. Joel Ramiro por sua amizade sincera e por sempre acreditar em mim, mesmo quando eu mesmo não acreditava.

Agradeço a todos os meus professores, em especial aos professores: David Romero, Andréia Libório, Paulo de Tarso, Viviane de Menezes, Francisco Hélder, David Sena, Valdemir Queiroz, Márcio Maia e Regis Magalhães por me proporcionar o conhecimento não apenas racional, mas construtor de caráter. Por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

Aos meus amigos, com quem convivi intensamente durante os últimos anos, em especial a: Lucas Cruz, Mateus Lima, André Marcelino, Marianna Severo, Victória Maia, Mateus Sousa, Iana Mary, Rodrigo Machado, Victor Lucas, Marcelo Martins, Rafaella Fernandes, -Ian Torres, Josman Cruz, Leonardo Mariz, Guilherme Rodrigues, Mateus Rios, André Conjo e Mizael Lima. Por compartilharem comigo tantos momentos de descobertas e aprendizado e por todo o companheirismo ao longo deste percurso que me permitiu crescer não só como formando, mas também como pessoa.

A todos os servidores e servidoras técnico-administrativos da Universidade Federal do Ceará Campus Quixadá por todo auxílio nas mais diversas atividades acadêmicas e extracurriculares.

“Cada coisa tem sua hora e cada hora o seu cuidado.”

(Rachel de Queiroz)

RESUMO

Este trabalho teve como principal objetivo desenvolver e implementar um sistema de telemonitoramento da qualidade da água e automação de viveiros de camarão e derivados, usando como tecnologia de comunicação o protocolo LoRaWAN. Dada a necessidade de um sistema eficiente, preciso e versátil para o monitoramento e automação dos tanques de cultivo de camarão, foi desenvolvido um protótipo baseado em um microcontrolador ESP32 e sensores de temperatura e pH alimentados com energia solar. O protótipo foi posto a prova em ambientes aquáticos onde os dados sensoriais foram recolhidos e transmitidos para uma estrutura em nuvem via LoRa e WiFi. Armazenados em um banco de dados e usados para notificações de status da qualidade da água via e-mail. O protótipo também se mostrou capaz de controlar atuadores para correção de parâmetros da qualidade da água. Assim o sistema desenvolvido mostrou-se adequado para a supervisão dos tanques de cultivo de camarão.

Palavras-chave: Camarão. LoRa. Automação.

ABSTRACT

This work had as main objective to develop and implement a system of telemonitoring of water quality and automation of shrimp ponds and derivatives, using the LoRaWAN protocol as communication technology. Given the need for an efficient, precise and versatile system for monitoring and automation of shrimp farming tanks, a prototype was developed based on an ESP32 microcontroller and temperature and pH sensors powered by solar energy. The prototype was put to the test in aquatic environments where sensory data was collected and transmitted to a cloud structure via LoRa and WiFi. Stored in a database and used for water quality status notifications via e-mail. The prototype is also capable of controlling the actuators to correct water quality parameters. Thus, the developed system proved to be suitable for the supervision of shrimp farming tanks.

Keywords: Shrimp. LoRa. Automation.

LISTA DE FIGURAS

Figura 1 – Ciclo de vida da maioria dos camarões peneídeos	20
Figura 2 – Principais parâmetros de qualidade da água, a importância/viabilidade de monitoramento no dia a dia de uma carcinicultura e o grau potencial de impacto ambiental causado pela carcinicultura (* muito baixo/baixo, ** médio, *** alto).	21
Figura 3 – Balanço de oxigênio dissolvido e de gás carbônico em um viveiro de cultivo de camarões. Os processos acelerados ou aumentados pelo aumento de matéria orgânica são indicados através das setas vermelhas.	22
Figura 4 – Escala de pH	23
Figura 5 – Disco de Secchi	25
Figura 6 – Elementos fundamentais de uma rede LoRaWan	26
Figura 7 – Classes do LoRaWAN	27
Figura 8 – Topologia do sistema de telemonitoramento com comunicação LoRa	28
Figura 9 – Exemplificação dos relacionamentos de automação proposto	29
Figura 10 – Sensor DS18B29	31
Figura 11 – Sensor ph-4502c	32
Figura 12 – Pinout do módulo BNC ph-4502c	32
Figura 13 – Divisor de tensão necessário a leitura de pH	33
Figura 14 – Projeto de hardware do protótipo do nó sensor	34
Figura 15 – Projeto de hardware do protótipo do nó sensor	35
Figura 16 – Nó sensor 01	35
Figura 17 – Nó sensor 02	36
Figura 18 – Nó atuador	36
Figura 19 – Gateway	37
Figura 20 – Sistema de alimentação solar	37
Figura 21 – Fluxograma do software do protótipo do nó sensor	39
Figura 22 – Fluxograma do software do protótipo do gateway	40
Figura 23 – Fluxograma do software do protótipo do nó atuador	41
Figura 24 – Estrutura de Cloud	42
Figura 25 – Gráfico de leitura de temperatura	43
Figura 26 – Gráfico de leitura de baixas e altas temperaturas	43

Figura 27 – Amostra de notificações via email	44
Figura 28 – Gráfico de leituras do pH	44
Figura 29 – Alcance de comunicação LoRa do protótipo em ambiente com alta densidade de construções civis	45
Figura 30 – Alcance de comunicação LoRa do protótipo em ambiente com menor densi- dade de construções civis	45
Figura 31 – Gráfico de intensidade do sinal recebido	46
Figura 32 – Tabela do tópico MQTT no banco de dados Dynamo DB	46

LISTA DE QUADROS

Quadro 1 – Comparação entre os trabalhos relacionados e o proposto	19
Quadro 2 – Comparação de microcontroladores e módulos LoRa dispostos.	30

LISTA DE ABREVIATURAS E SIGLAS

AWS	Amazon Web Services
IaaS	Infrastructure as a service
IAM	Identity and Access Management
IoT	Internet of Things
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWA	Low-Power Wide-Area
MAC	Medium Access Control
MQTT	Message Queue Telemetry Transport
OD	Oxigênio Dissolvido
OSI	Open Systems Interconnection
pH	Potencial Hidrogeniônico
RSSI	Received signal strength indication
SNS	Simple Notification Service

SUMÁRIO

1	INTRODUÇÃO	15
1.1	MOTIVAÇÃO	15
1.2	OBJETIVOS	16
1.2.1	OBJETIVOS GERAIS	16
1.2.2	OBJETIVOS ESPECÍFICOS	16
2	TRABALHOS RELACIONADOS	18
3	FUNDAMENTAÇÃO TEORICA	20
3.1	Carcinicultura	20
3.1.1	<i>Litopenaeus Vannamei e seu cultivo</i>	20
3.1.2	<i>Impacto da Qualidade da Água na Carcinicultura</i>	21
3.1.2.1	<i>Oxigênio Dissolvido</i>	22
3.1.2.2	<i>pH</i>	23
3.1.2.3	<i>Salinidade</i>	23
3.1.2.4	<i>Temperatura</i>	24
3.1.2.5	<i>Transparência</i>	24
3.1.3	<i>LoRa</i>	25
3.1.4	<i>LoRaWAN</i>	25
4	METODOLOGIA	28
4.1	Definição da topologia e arquitetura do sistema	28
4.1.1	<i>Telemonitoramento</i>	28
4.1.2	<i>Automação</i>	29
4.2	Definição dos componentes	29
4.2.1	<i>Microcontrolador e módulo LoRa</i>	30
4.2.2	<i>Sensores</i>	30
4.2.2.1	<i>Temperatura</i>	31
4.2.2.2	<i>pH</i>	31
4.2.3	<i>Estrutura de Cloud</i>	32
4.3	Desenvolvimento do protótipo	33
4.3.1	<i>Hardware</i>	33
4.3.2	<i>Software</i>	37

4.3.3	<i>Infraestrutura Cloud</i>	41
4.4	Testes e resultados obtidos	42
4.4.1	<i>Teste de leitura da temperatura, controle do atuador e notificação</i>	42
4.4.2	<i>Teste de leitura do pH</i>	44
4.4.3	<i>Teste de transmissão de pacotes</i>	45
4.4.4	<i>Teste multiplos nós de integração do sistema embarcado com estrutura em nuvem</i>	46
5	CONCLUSÕES E TRABALHOS FUTUROS	47
	REFERÊNCIAS	48
	APÊNDICES	51
	APÊNDICE A – Códigos-fonte utilizado no protótipo do nó sensor	51
	APÊNDICE B – Códigos-fonte utilizado no protótipo do gateway	55
	APÊNDICE C – Códigos-fonte utilizado no protótipo do nó atuador	63

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

A aquicultura (produção de organismos aquáticos) desempenha papel significativo na eliminação da fome, promoção da saúde e redução da pobreza. Nunca antes, pessoas consumiram tantos alimentos oriundos da aquicultura ou dependeram tão significativamente do setor para o seu bem-estar (ORGANIZATION, 2014). Sendo a carcinicultura, cultivo de camarões em viveiros, um dos ramos da aquicultura que mais tem crescido no mundo. O camarão é extremamente nutritivo, fonte vital de proteínas e nutrientes, especialmente para os membros mais pobres da nossa comunidade global (SEABRA, 2011).

Iniciada na década de 1970, a carcinicultura brasileira (MEIRELES *et al.*, 2007) apresenta durante sua história métricas de produção admiráveis. Em 2011 representou cerca de 80% da produção mundial de camarão (ORGANIZATION, 2014). Neste mesmo ano, a cadeia produtiva de camarão representou R\$ 1 bilhão, afetando consideravelmente a economia nacional, principalmente da região Nordeste; gerando empregos (3,75 por hectare) e por consequentes impactos sociais positivos (ARAUJO *et al.*, 2018).

Não obstante, há trabalhos que determinam a eficiência técnica de carcinicultores, evidenciando que para a perpetuação e expansão desse setor, o controle da qualidade de água é essencial. Estando diretamente relacionado a manutenção de condições ideais de saúde e crescimento das formas de vida aquática, presentes nas fazendas de criação (BEGUM *et al.*, 2013).

Atualmente, o controle da qualidade da água é desempenhado, convencionalmente, por meio da coleta manual de amostras de água e entrega dessas amostras a laboratórios, ou ainda, por instrumentos manuais de medição. Ambos os métodos apresentam desvantagens: mão de obra dispendiosa, alta possibilidade de erro humano, monitoramento descontínuo e impossibilidade de acompanhamento a distância de modo automático.

Tais processos, associados à busca de maior produtividade por hectare cultivado e ao surgimento de mazelas como o vírus da síndrome da mancha branca (*Nimaviridae*), tem motivado a busca por tecnologias mais sofisticadas: equipamentos, instrumentos e práticas de monitoramento (ORMOND *et al.*, 2004). Objetivando melhorar a qualidade da água e obter custos menores na produção, como a redução do consumo de energia elétrica, oriundos, inclusive,

do uso de aeradores¹.

Uma promissora tecnologia para essa problemática são as redes Long Range (LoRa). Uma rede LoRa é uma rede composta por hosts terminais e gateways, que mantém comunicação sem fio de longo alcance e baixo consumo elétrico. Possui baixo custo de manutenção e expansão à medida que não detém infraestrutura fixa. (SEMTECH, 2015).

Neste trabalho é proposto o desenvolvimento de uma plataforma de telemonitoramento e automação de viveiros de camarão e derivados, que possa atender os requisitos do mercado brasileiro. O sistema visa auxiliar carcinicultores, engenheiros de pesca e colaboradores das fazendas de criação no cultivo de camarões.

1.2 OBJETIVOS

1.2.1 OBJETIVOS GERAIS

O trabalho propõe o desenvolvimento de um sistema de telemonitoramento da qualidade da água e automação de viveiros de camarão e derivados, usando como tecnologia de comunicação o protocolo Long Range Wide Area Network (LoRaWAN) atendendo as seguintes características:

- a) Fácil instalação, onde qualquer operador com a mínima instrução seja capaz colocar o equipamento em operação.
- b) Flexível, podendo ser modificado a qualquer momento, seja na sua localização ou substituição dos sensores nele contidos.
- c) Sem fios de comunicação e alimentação elétrica.
- d) Capacidade de monitorizar os viveiros, recolhendo e armazenando parâmetros da água de forma a melhorar o controle da qualidade da água.

1.2.2 OBJETIVOS ESPECÍFICOS

- a) Construir um protótipo funcional do sistema, incluso comunicação via LoRaWAN, sendo capaz de:
 - Manipular sensores analógicos com resolução não inferior a 0,1 determinando perioticamente suas respectivas grandezas correlacionadas a qualidade da

¹ Equipamento, usado na aquicultura, colocado na superfície da água dos tanques. Tem a finalidade de realizar a incorporação de oxigênio na água.

- água.
- Determinar periodicamente a temperatura da água, operando entre 10°C e 50°C , com uma resolução nunca inferior a $0,5^{\circ}\text{C}$;
 - Controlar o acionamento e desligamento de atuadores relacionados a qualidade da água.
- b) Detalhar as características da rede LoRa e como esta se adaptaria ao monitoramento de ambientes aquáticos, no caso sistemas de cultivo de camarão.
- Analisando:
- Viabilidade de implementação do sistema relativa a abrangência de sinal LoRa atingido com o protótipo;
 - Intensidade de recepção do sinal LoRa produzido pelo protótipo.
- c) Construir uma estrutura em nuvem capaz de:
- Armazenar os dados sensoriais coletados;
 - Notificar o usuário quanto a parâmetros de qualidade da água inadequados.

2 TRABALHOS RELACIONADOS

Após extensa pesquisa, leitura, estudo e levantadas as devidas comparações, pode-se identificar, dentre a literatura disposta em repositórios online, trabalhos análogos. Semelhantes quanto a problemática e solução proposta neste projeto de pesquisa. Comparados com o trabalho proposto no Quadro 1. Destacando-se:

- a. **A Dissertação de Daniel Pozzatti** (POZZATTI, 2014), onde é exposto o desenvolvimento de um sistema de monitoramento da qualidade da água de baixo custo, utilizando rede de sensores sem fio. Aplicado em centros de criação de esturjão Beluga, em Portugal. Dada a necessidade de um sistema de bom custo-benefício, foi desempenhado um estudo detalhado sobre sensores, rede de sensores sem fio e sobre o protocolo ZigBee, utilizado na comunicação entre os nós da rede. Assim, foi possível o desenvolvimento de um protótipo capaz de efetuar o pretendido. O sistema foi capaz de monitorar tanques de criação de esturjão Bejuga, estabelecer comunicação sem fio via protocolo ZigBee bem como o envio dessas informações para nuvem e armazenamento delas em um banco de dados. Contudo o projeto, limitado pelo protocolo ZigBee, não era capaz de estabelecer abrangência de comunicação sem fio de longa distância.
- b. **O projeto de Bing Shi, Victor Sreeram, Dean Zhao, Suolin Duan, Jianming Jiang** (SHI *et al.*, 2018), que implementaram um sistema de telemonitoramento da qualidade da água com comunicação ZigBee e Wifi capaz de monitorar oxigênio dissolvido e temperatura. Além de comparar sistemas de monitoramento de ambiente baseados em redes de sensores sem fio e redes cabeadas. Sendo o sistema desenvolvido aplicado a piscicultura.
- c. **O trabalho Indonês de Kurniawan, Mahmudin, Maulana, Syamsu e Wiranto** (MAULANA *et al.*, 2018), que descreve o projeto e construção de um sistema de monitoramento da qualidade da água online, baseado em rede de sensores sem fio com protocolo ZigBee. O objetivo era criar condições adequadas relativas a qualidade da água, na carcinicultura. Além de reduzir custos, referentes ao consumo de energia elétrica, usando um sistema automatizado de aeração. O protótipo desenvolvido se mostrou apto ao telemonitoramento de oxigênio dissolvido, pH e temperatura, apresentando os dados coletados de modo visual em uma aplicação web.

Quadro 1 – Comparação entre os trabalhos relacionados e o proposto

Conteúdo	Trabalhos relacionados			Proposta
	Pozzati (2014)	Shi et al. (2018)	Maulana et al. (2018)	
Telemonitoramento da qualidade da água	Sim	Sim	Sim	Sim
Automação de viveiros de criação	Não	Não	Sim	Sim
Uso de LoRa/LoRaWAN	Não	Não	Não	Sim
Foco em carcinicultura	Não	Não	Sim	Sim

Fonte: elaborado pelo autor

Uma vez que projetos que adotem sistemas de telemonitoramento e automação, simultaneamente, ainda são escassos para esse nicho (carcinicultura), se faz necessário o fomento de iniciativas que venham suprir tal carência por tecnologias mais robustas deste setor. Além da ausência de trabalhos sobre o uso de LoRa como tecnologia vigente na área em questão. Dado o exposto, é admissível a relevância do trabalho proposto.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Carcinicultura

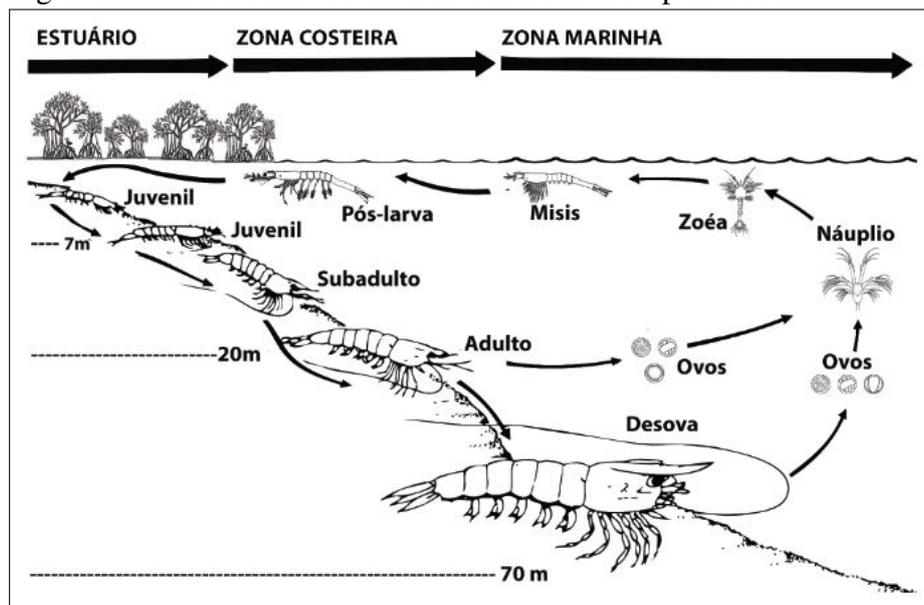
As subseções a seguir explanam aspectos necessários sobre a contextualização do setor alvo deste trabalho: a carcinicultura, cultivo de camarões em viveiros.

3.1.1 *Litopenaeus Vannamei* e seu cultivo

A *litopenaeus vannamei*, nativa do Oceano Pacífico, é a espécie de camarão mais produzida no Brasil (NUNES *et al.*, 2011). Popularmente conhecido no Nordeste do Brasil como "camarão cinza". Introduzido na década de 80 no Brasil, uma vez que possui grandes vantagens em comparação às espécies nativas, como informações zootécnicas e biológicas bem definidas. Trata-se de uma espécie marinha que se adapta muito bem em baixas salinidades (HERNANDEZ; NUNES, 2001), sendo possível seu cultivo em água doce.

Pertencente a família Penaeidae, o camarão cinza apresenta um ciclo de desenvolvimento dividido nos seguintes estágios: larva (náuplio), protozoa (zoéa), misis, pós-larva, juvenil, sub-adulto e adulto (OSTRENSKY A. *et al.*, 2017). Atingindo até 23 cm de comprimento, em condições naturais. Ciclo ilustrado na Figura 1.

Figura 1 – Ciclo de vida da maioria dos camarões peneídeos



Fonte: Ostrensky A. *et al.* (2017)

A maioria das fazendas de cultivo do camarão cinza no Brasil utilizam pós-larvas

(PLs) produzidas em laboratórios especializados, onde são adequadas à salinidade, temperatura e pH da água para onde serão transferidas. Esta é a primeira fase nomeada de larvicultura, etapa de produção de pós-larvas. A segunda fase é a engorda, responsável pelo crescimento do camarão até a idade de comercialização. É nessa etapa que os carcinicultores atuam ativamente e por consequente ela será o foco do sistema proposto neste trabalho. Atualmente, em condições ideais, a metodologia utilizada para a engorda é o cultivo semi-intensivo, caracterizado por densidades populacionais moderadas (aproximadamente 30 camarões/m²) (PONTES; ARRUDA, 2005).

3.1.2 Impacto da Qualidade da Água na Carcinicultura

Segundo a **Resolução CONAMA nº 312/2002**, dentre os parâmetros de qualidade de água, pode-se elencar como mais prioritários, quanto ao monitoramento para esse nicho: o oxigênio dissolvido (OD); pH, salinidade; temperatura e transparência (sintetizados na Figura 2).

Figura 2 – Principais parâmetros de qualidade da água, a importância/viabilidade de monitoramento no dia a dia de uma carcinicultura e o grau potencial de impacto ambiental causado pela carcinicultura (* muito baixo/baixo, ** médio, *** alto).

Parâmetro	Importância de monitoramento da carcinicultura	Potencial Impacto ambiental
Acidez	*	*
Amônia	**	***
Clorofila "a"	**	***
Coliformes totais	*	*
Cor	*	**
DBO	*	***
Dureza	*	*
Fosfato	**	***
Material em suspensão	**	***
Nitrato	**	***
Nitrito	**	**
OD	***	***
pH	***	***
Salinidade	***	*
Silicato	**	*
Sólidos dissolvidos	*	**
Temperatura	***	*
Transparência	***	***
Turbidez	**	***

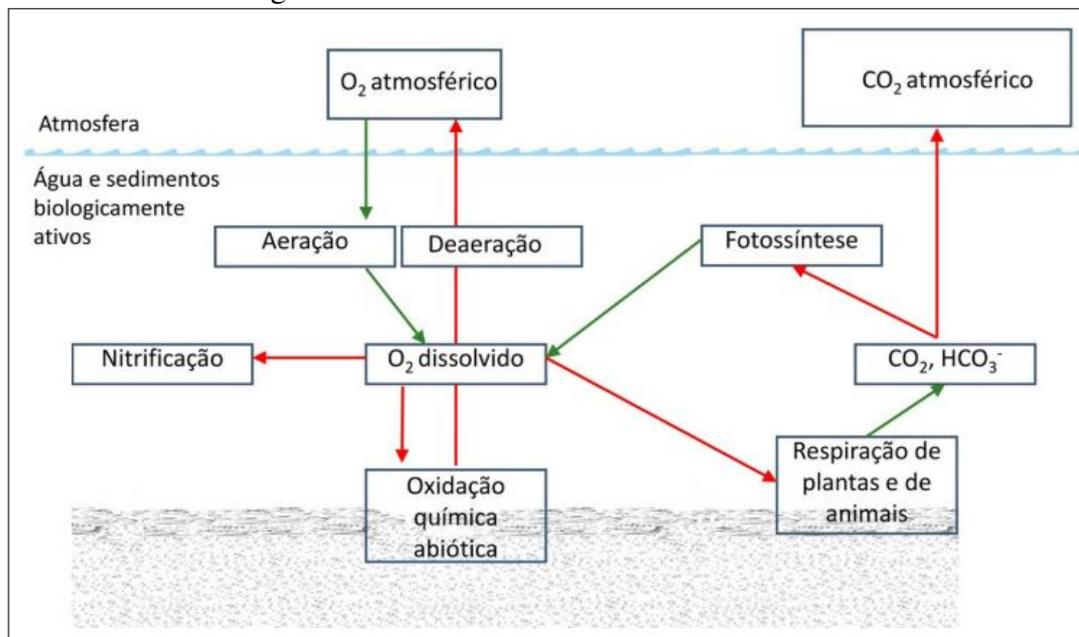
Fonte: Ostrensky A. e Silva (2017)

3.1.2.1 Oxigênio Dissolvido

Quando o oxigênio presente no ar está em contato com a água, o mesmo se dissolve até haver igualdade entre as pressões relativas do oxigênio no ar e na água. A tal oxigênio infiltrado na água damos a nomenclatura de Oxigênio Dissolvido (OD). Elemento vital para os camarões, sendo essencial à quase todos os organismos aquáticos. Corresponde ao elemento fundamental no processo de respiração dos mesmos.

As principais fontes de OD na água são, em ordem: a fotossíntese, a água de renovação e a atmosfera, excluído o uso de aeradores. Contudo, durante o processo de cultivos, diversos processos o consomem. Os processos que mais demandam OD são: decomposição da matéria orgânica, nitrificação, respiração dos organismos aquáticos e a oxidação química abiótica. Quanto mais intensivo for o cultivo maior será o consumo de oxigênio (OSTRENSKY A.; SILVA, 2017). A Figura 3 fornece em maiores detalhes o processo de geração e consumo do oxigênio dissolvido.

Figura 3 – Balanço de oxigênio dissolvido e de gás carbônico em um viveiro de cultivo de camarões. Os processos acelerados ou aumentados pelo aumento de matéria orgânica são indicados através das setas vermelhas.



Fonte: Ostrensky A. e Silva (2017)

A condição ideal de oxigênio para o cultivo de camarões são concentrações superiores a 4 mg/L. Não significando, necessariamente, mortalidade em concentrações menores, uma vez que a maioria das espécies tolera 1,5 mg/L por curtos períodos de tempo. Entretanto baixas concentrações desse elemento (1,5-3,0mg/L) ocasionam danos ao cultivo como:

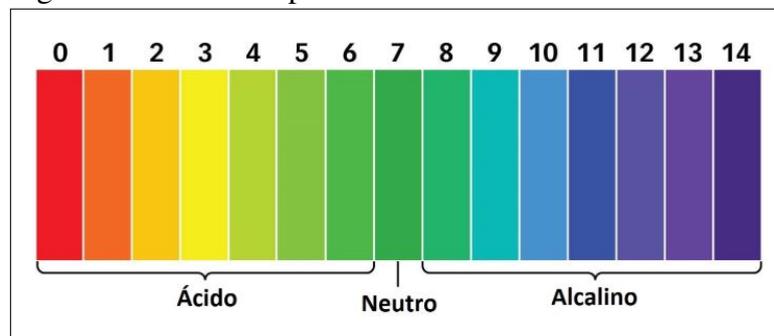
- Atraso no crescimento;
- Redução na eficiência alimentar
- Aumento na incidência de doenças e na mortalidade

Causando perda na produtividade e por conseqüente prejuízos ao carcinicultor. Entender os fatores que afetam a dinâmica do oxigênio nos viveiros de criação e monitorar este elemento são tarefas fundamentais ao manejo econômico da produção carcinicultora (EMBRAPA, 2019).

3.1.2.2 *pH*

Potencial Hidrogeniônico (pH) é a representação do níveis de acidez, neutralidade e alcalinidade de determinada solução. Isto é, o pH expressa o grau de "intensidade" de íons de hidrogênio na água. Sua escala, de 0 a 14, é logarítmica, ou seja, em uma mesma solução, se o pH medido for 7, significa que a concentração de íons hidrogênio nela será 10 vezes maior que em uma solução com pH 6. Representando os estado de ácido ($\text{pH} < 7$), neutro ($\text{pH} = 7$) e básico ($\text{pH} > 7$) (BLOG 2 ENGENHEIROS, 2019). Escala ilustrada na Figura 4.

Figura 4 – Escala de pH



Fonte: Acquanativa (2019)

Valores inadequados de pH afetam, na carcinicultura, gerando danos que vão desde a respiração dos camarões (por afetar suas branquias) até a mortalidades massivas. Os valores aceitáveis de pH na carcinicultura variam entre 6,5 e 8,5.

3.1.2.3 *Salinidade*

Mais do que a densidade de cloreto de sódio na água, salinidade refere-se a concentração de íons de sais dissolvidos na água, como: cloretos, sódio, magnésio, sulfato, cálcio, potássio, bicarbonatos e outros tantos, mesmo que em quantidades minúsculas. Há duas uni-

dades de medidas principais: partes de sal por mil partes de água (ppmi) e Unidade Prática de Salinidade(UPS).

Comparando com outras espécies de camarões, *L. vannamei* destaca-se pela sua grande capacidade de osmorregulação², já sendo cultivado inclusive em água praticamente doce com salinidade acima de 0,5 UPS (HENNIG; ANDREATTA, 1998).

3.1.2.4 *Temperatura*

O camarão é um animal ectotérmico, isto é, seu metabolismo e temperatura corporal são diretamente dependentes da temperatura ambiente. Todas as atividades fisiológicas dos camarões: respiração, digestão, excreção, alimentação e movimentos estão intimamente ligadas à temperatura da água.

Em temperaturas elevadas seus metabolismos aceleram ao passo que em baixas temperaturas seus metabolismos ficam mais devagar, afetando o consumo de ração e crescimento (HENNIG; ANDREATTA, 1998). Logo, é essencial ao cultivo o monitoramento desse fator. Para o camarão pequeno (<5 gramas), a temperatura adequada é superior a 30°C, enquanto que para o camarão grande, a temperatura ideal é de cerca de 27°C (WYBAN *et al.*, 1995).

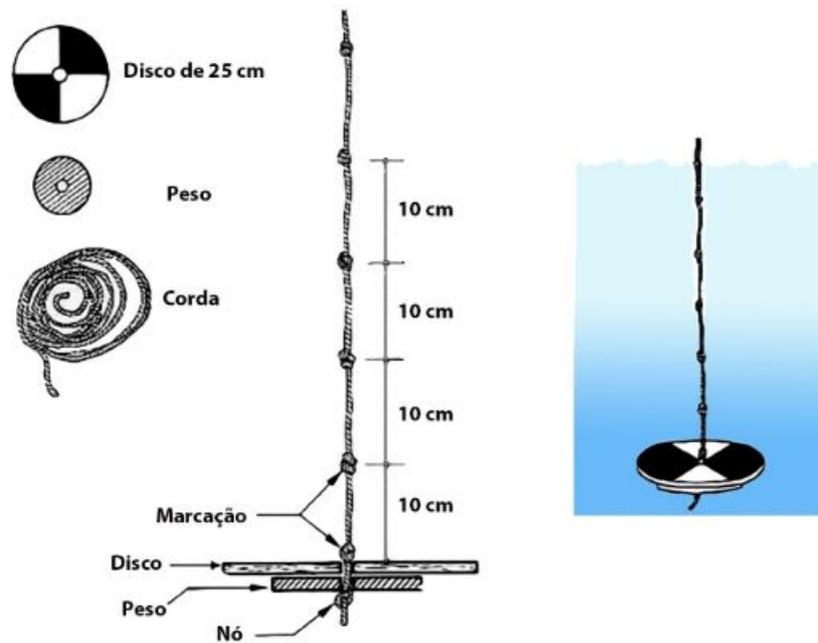
3.1.2.5 *Transparência*

Representa o índice de penetração da luz na água. Quanto mais transparência maior é a capacidade de penetração da luz na água, beneficiando organismos como microalgas (bloom fitoplanctônico). São seres como esses que por meio da fotossíntese representam grande parte do volume de oxigênio dissolvido na água. Atualmente, o instrumento mais usado em sua medição é o disco de Secchi: um disco plástico pintado de preto e branco, de modo alternado em quadrantes, amarrado a uma corda, normalmente, com divisões de 10cm indicadas por nós. Além da presença de um peso anexado ao disco para fazê-lo afundar. Mais detalhes são fornecido pela Figura 5.

A recomendação é que a transparência seja monitoradas entre 10h e 14h, devido a maior incidência de luz solar no dia. Os valores ideais estão entre 30 e 45cm. Logo, transparência é parâmetro primordial no processo de geração de oxigênio dissolvido.

² Controle das concentrações de sais nos tecidos ou células vivas a fim de manter as condições adequadas à atividade metabólica

Figura 5 – Disco de Secchi



Fonte: Ostrensky A. e Silva (2017)

3.1.3 LoRa

Desenvolvida pela *Semtech Corporation* e viabilizada pela *LoRa™ Alliance*, a tecnologia LoRa é uma especificação privada que define uma modulação de comunicação sem fio. Fundamentada em espalhamento espectral, oriunda da modulação CSS, se direciona para aplicações de baixo consumo elétrico e comunicação sem fio de longo alcance, atingindo abrangências de até 5Km em áreas urbanas e 15Km em áreas abertas (SEMTECH, 2015; TEIXEIRA; ALMEIDA, 2017).

Comumente utilizada em sistemas Internet of Things (IoT) (ORTIZ *et al.*, 2018). A LoRa é implementada na camada física do modelo Open Systems Interconnection (OSI), sendo absorva em relação às implementações de camadas superiores. Assim, mostra-se versátil coexistindo e interoperando com uma gama de arquiteturas de rede existentes (SEMTECH, 2015).

3.1.4 LoRaWAN

LoRaWAN é o protocolo de rede Low-Power Wide-Area (LPWA) que define parâmetros de comunicação e de acesso ao meio (Medium Access Control (MAC)). Implementado

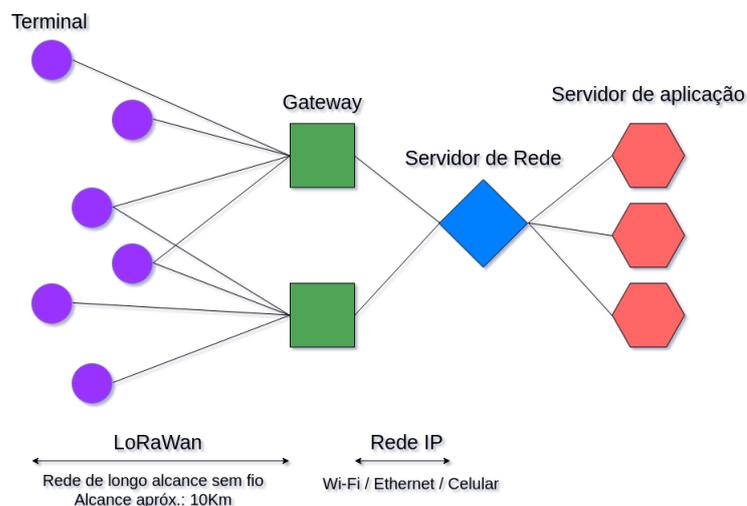
sobre a modulação LoRa™, ele oferta comunicação bidirecional de baixo custo, móvel e segura para soluções de IoT, M2M e aplicações industriais.

Ao contrário da modulação LoRa, as especificações do protocolo LoRaWAN são abertas ao público (ORTIZ *et al.*, 2018) operando em um espectro de rádio não licenciado, isto é, qualquer um pode utilizar tal tecnologia sem a necessidade de uma licença ou autorização. Comunicação de rádio esta que utiliza baixas frequências, possibilitando maior alcance.

A arquitetura do LoRaWAN é composto por três elementos: terminais (*endpoints*), gateways e servidores. Os terminais são os elementos que operam sobre o meio onde a rede é presente. Os terminais enviam informações/requisições aos gateways que por sua vez concentram e processam essas informações, estabelecendo elo com os servidores.

Os servidores recebem essas informações/requisições provenientes dos terminais e as repassam as devidas aplicações. Podendo, posteriormente, enviar os devidos comandos/respostas aos terminais, também via gateways. Conforme expõe a Figura 6.

Figura 6 – Elementos fundamentais de uma rede LoRaWan



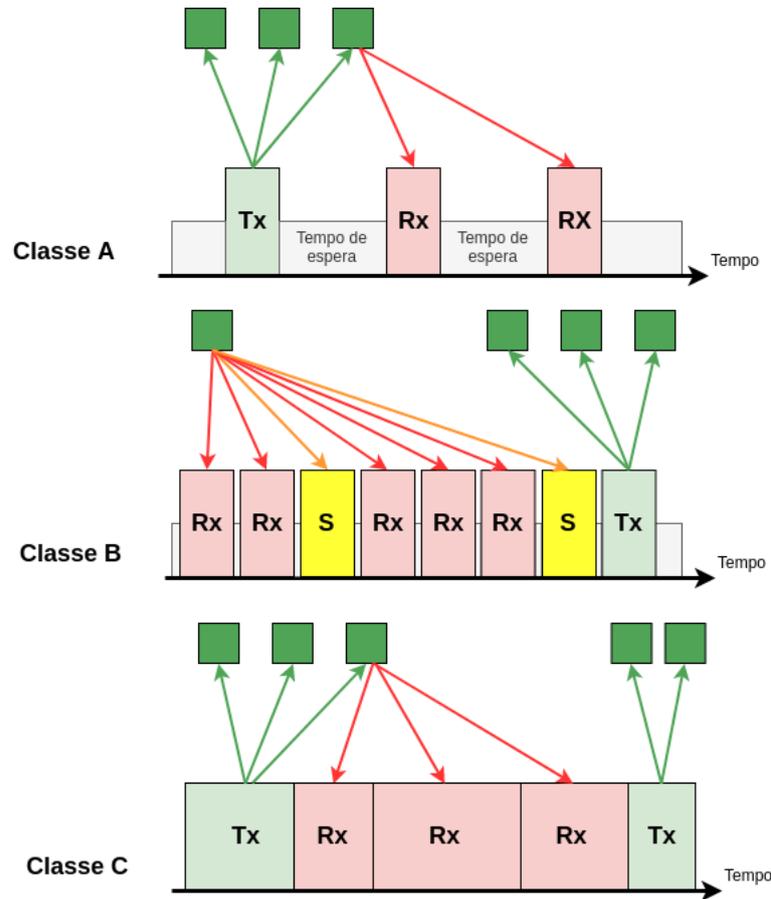
Fonte: elaborada pelo autor

Nos terminais, o protocolo define três classes de comunicação bidirecional: a classe básica A e as classes opcionais B e C. Os terminais da classe A definem dois espaços de tempo para recepção de mensagens (transmissão downlink) após o envio de mensagens (transmissão de uplink). Havendo tempos de espera programados entre intervalos de envio e recebimento, que tem como fim compensar possíveis atrasos e limitações de velocidade na comunicação.

A classe B é uma derivação da classe A onde há a possibilidade de agendamento de

espaço extra para recepção. Sincronizada por um sinal de rede. Na classe C, também derivação da classe A, o espaço de recepção é contínuo, exceto nos períodos quando o terminal precisa realizar o envio de mensagens. Os intervalos de tempo dessas classes são expostos na Figura 7.

Figura 7 – Classes do LoRaWAN



Legenda:

- Tx Intervalo de envio
- Rx Intervalo de recebimento
- S Intervalo de sincronização
- Gateway
- Uplink
- Potencial Downlink
- Sinal de sincronização

Fonte: elaborado pelo autor

No mais, o protocolo LoRaWAN é adequado para redes que necessitem se comunicar com aplicativos conectados à Internet, em conexões sem fio de longo alcance.

4 METODOLOGIA

4.1 Definição da topologia e arquitetura do sistema

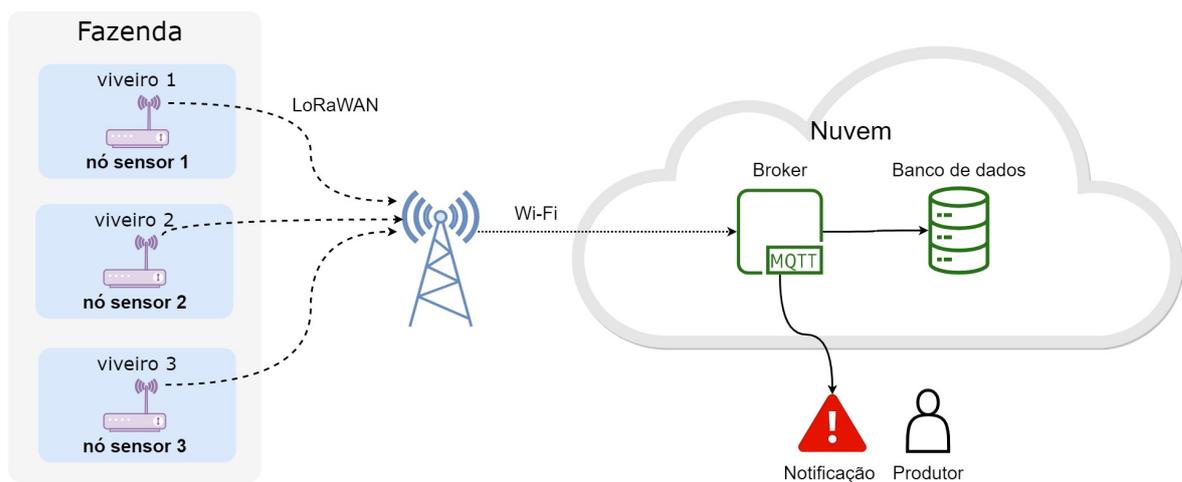
4.1.1 Telemonitoramento

O sistema operará em topologia estrela, inerente a redes LoRa. Os nós sensores são os terminais deste sistema que, não obstante as definições anteriores, coletam dados do ambiente. Assim, esses nós são responsáveis pelo sensoriamento de parâmetros indicadores de qualidade da água. Um sistema de captação e conversão de energia solar é encarregado pela alimentação elétrica dos nós sensores.

A transferência dos dados para o gateway LoRa será realizada via protocolo LoRaWAN. Todo nó é passível a comunicação direta com o gateway, desde que esteja em sua área de abrangência. O gateway enviará os dados de leitura para nuvem via rede sem fio. Os dados serão publicados em um broker Message Queue Telemetry Transport (MQTT).

A estrutura em nuvem usará os dados presentes no tópico MQTT para alimentar um banco de dados e notificar o usuário acerca de parâmetros de qualidade da água inadequados ao cultivo em questão. Essa topologia é exibida na Figura 8.

Figura 8 – Topologia do sistema de telemonitoramento com comunicação LoRa



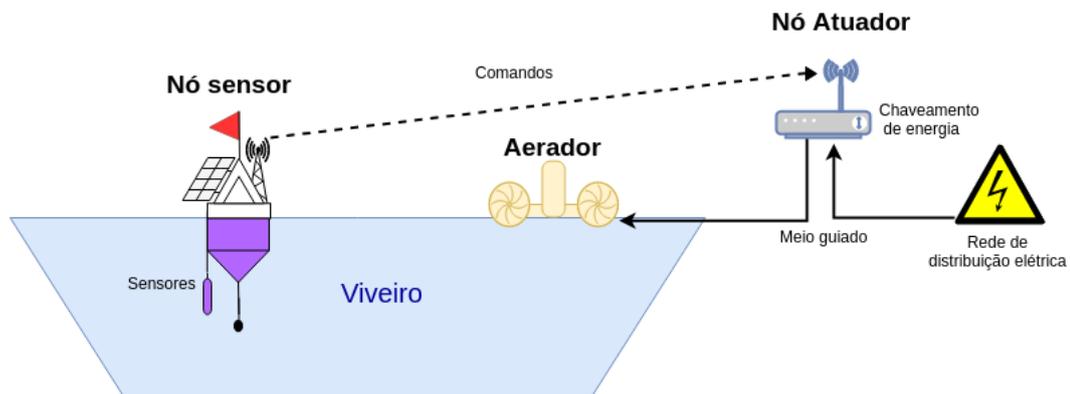
Fonte: elaborada pelo autor

4.1.2 Automação

Em simultâneo ao telemonitoramento, os nós sensores se utilizarão das leituras de ambiente para aplicar as devidas ações de controle. Diante de índices que evidenciem parâmetros inadequados, os nós enviarão mensagens de comando aos atuadores responsáveis a correção dos respectivos parâmetros. A comunicação entre nós sensores e atuadores ocorrerá via protocolo LoRaWAN, por intermédio do gateway LoRa.

Deste modo, por exemplo, um nó ao detectar baixos níveis de OD, em determinado viveiro, acionará o aerador do viveiro em questão. Com igualdade, ao ler índices adequados de OD, o nó sensor enviará o comando de desligamento do aerador. A Figura 9 ilustra tais eventos.

Figura 9 – Exemplificação dos relacionamentos de automação proposto



Fonte: elaborada pelo autor

Os demais parâmetros são restritos a correções químicas e/ou que necessitam de infraestrutura prévia nos viveiros (tubulações, sistemas de drenagem e reposição de água, etc). Assim, de posse das informações dispostas pelo telemonitoramento, caberá ao usuário as devidas adequações.

4.2 Definição dos componentes

Nesta seção é explanado o processo de escolha dos componentes e modelos necessários ao desenvolvimento de um protótipo do sistema bem como suas devidas justificativas. Cabendo a ressaltar a relevância dos critérios de desempenho e custo, pois é desejável a viabilidade de escalonamento do sistema.

4.2.1 Microcontrolador e módulo LoRa

Objetivando um sistema versátil, resistente e de baixo custo e consumo elétrico, seleções e comparações, destes componentes, serão discutidos nesta subseção. Foram selecionados dois dispositivos LoRa para possível utilização no protótipo:

1. Things UNO

Desenvolvido pela *The Things Network*³ baseado no Arduíno Leonardo, é composto por um Arduíno e um módulo *Microchip* LoRaWAN. Sendo opcional uma antena externa. Esse dispositivo de hardware aberto, conta com um alcance de 10km e compatibilidade com a Arduíno IDE, fora bibliotecas LoRaWAN nativas (THE THINGS NETWORK, 2017).

2. WiFi LoRa 32

Essa placa, projetada pela *Heltec Automation*TM, é formulada por uma placa ESP32 atrelada a um módulo LoRaWAN SX127x. Possui funções Wi-Fi, BLE e LoRaWAN, suportando o ambiente de desenvolvimento Arduíno. Além de OLED e antena inclusos.

Quadro 2 – Comparação de microcontroladores e módulos LoRa dispostos.

Especificações	Things UNO	WiFi LoRa 32
MCU	Arduíno Leonardo	ESP32
Processador	ATmega32u4 - 16 MHz	Tensilica LX6 dual-core + 1 ULP - 240MHz
Chip LoRa	RN2483 433 MHz ~868 MHz	SX1276 915MHz
Antena inclusa	Não	Sim
Alimentação	7 ~12V	3,3 ~6V
Custo (US\$)	61,69	18,80

Fonte: elaborado pelo autor

O Quadro 2 levanta o comparativo entre ambos, evidenciando vantagem no uso do nó WIFI LoRa 32, da Heltec. Por seu consumo de energia e processamento superior, além do valor exacerbadamente menor do que o concorrente.

4.2.2 Sensores

Ambicionando monitorar parâmetros de qualidade da água de grande importância e integrar sensores digitais e analógicos, foram eleitos sensores de temperatura e pH para construção do sistema.

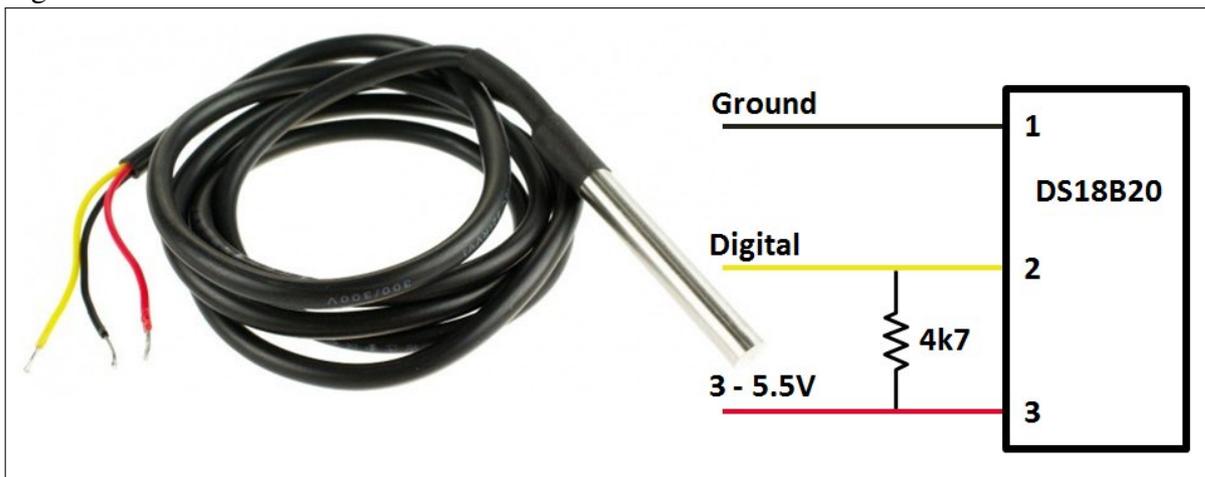
³ Segundo o site oficial (<https://www.thethingsnetwork.org/country/brazil/>): "A 'The Things Network' (TTN) é um projeto global cuja missão é construir uma rede de Internet das Coisas aberta, global e crowdsourced, ou seja, construída, mantida e operada pelos seus usuários".

4.2.2.1 Temperatura

Destaque de mercado no sensoriamento de temperatura da água, o DS18B20 é um sensor do tipo sonda revestido por material impermeável com ponta em aço inoxidável. O DS18B20 se mostra versátil monitorando também ambientes secos. Composto com apenas um único cabo (OneWire) usado para alimentação e envio de dados, seu período de atualização é menor que 750ms.

O sensor possui precisão de +/- 0,5°C entre -10°C e +85°C, conseguindo operar entre -55°C a +125°C. O mesmo trabalha com a tensão de operação de 3 a 5,5V corroborando em economia de energia. O sensor DS18B20 se mostra apropriado ao sistema proposto por sua usabilidade, precisão, baixo custo financeiro e energético. A Figura 10 apresenta o sensor DS18B20.

Figura 10 – Sensor DS18B29



Fonte: Rototron (2013)

4.2.2.2 pH

Dentre os mais variados modelos de destaque na academia e mercado, tais como os sensores digitais de temperatura DS1820 e de pH Atlas Ph Probe (fabricado pela *Atlas Scientific*), elencados em trabalhos análogos como melhor opção (POZZATTI, 2014; PARRA *et al.*, 2018), há um alto valor de custo (US\$ 76,38 ao todo nos relatados).

Não obstante a suas configurações, o sensor ph-4502c monitora pH em precisão adequada ao visado pelo projeto proposto neste trabalho (precisão superior à 0,1). O ph-4502 custa em média US\$33,42, apresentando por um adaptador BNC, tensão de alimentação de 5v e

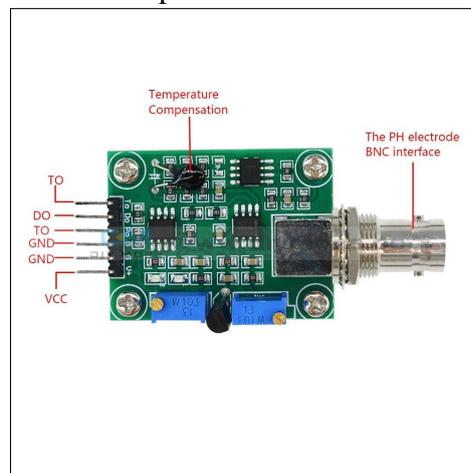
demais características apresentadas na tabela 3 o sensor não somente se mostra de fácil manuseio e versátil como mostra-se a solução adequada ao protótipo relatado. A imagens das Figuras 11 e 12 retratam o sensor bem como a descrição de seus pinos.

Figura 11 – Sensor ph-4502c



Fonte: Autocore Robotica (2020)

Figura 12 – Pinout do módulo BNC ph-4502c



Fonte: Diy More (2020)

4.2.3 Estrutura de Cloud

Dentre os critérios de eleição da estrutura em nuvem, foram elencados: número de serviços disponíveis na plataforma; nível de integração com aplicações externas; preço por serviços e disponibilização de serviços de IoT. Assim, a solução da Amazon para estrutura de cloud: Amazon Web Services (AWS) foi a preferida dentre as opções dispostas.

A AWS dispõe não somente um número de serviços (inclusive de IoT) e integração com soluções externas superiores a concorrentes como também boa viabilidade financeira. Havendo em seu pool de serviços um programa educacional (AWS Educate) que oferta cursos, tutoriais e cotas para acesso aos seus serviços mais utilizados de modo gratuito. Mantendo-se como líder de mercado em Infrastructure as a service (IaaS) com 31% de gastos de cliente do setor (CANALYS, 2020).

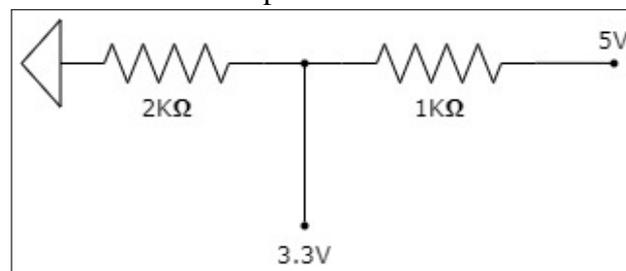
4.3 Desenvolvimento do protótipo

4.3.1 Hardware

O sistema físico do protótipo foi construído com a placa WIFI LoRa 32 (V2) e um conjunto de itens periféricos que divergiam compondo a estrutura de nó sensor e nó atuador. No desenvolvimento do protótipo de nó sensor foram utilizados os sensores de temperatura e pH, anteriormente citados. Um resistor de 4.7K ohms foi inserido entre os fios de dados e tensão do sensor de temperatura para melhor precisão.

O sensor de pH em seu pino de dados (Po) retorna valores de tensão entre 0 e 5V ao passo que os ADCs da WIFI LoRa 32 são aptos a receberem valores até 3.3V. Assim para leitura dos dados de pH foi necessário implementar um divisor de tensão. Esse divisor foi composto por dois resistores: um de 1K ohms e um de 2K ohms, limitando a tensão de entrada no ADC em até 3.3V. O divisor de tensão é exposto na Figura 13.

Figura 13 – Divisor de tensão necessário a leitura de pH

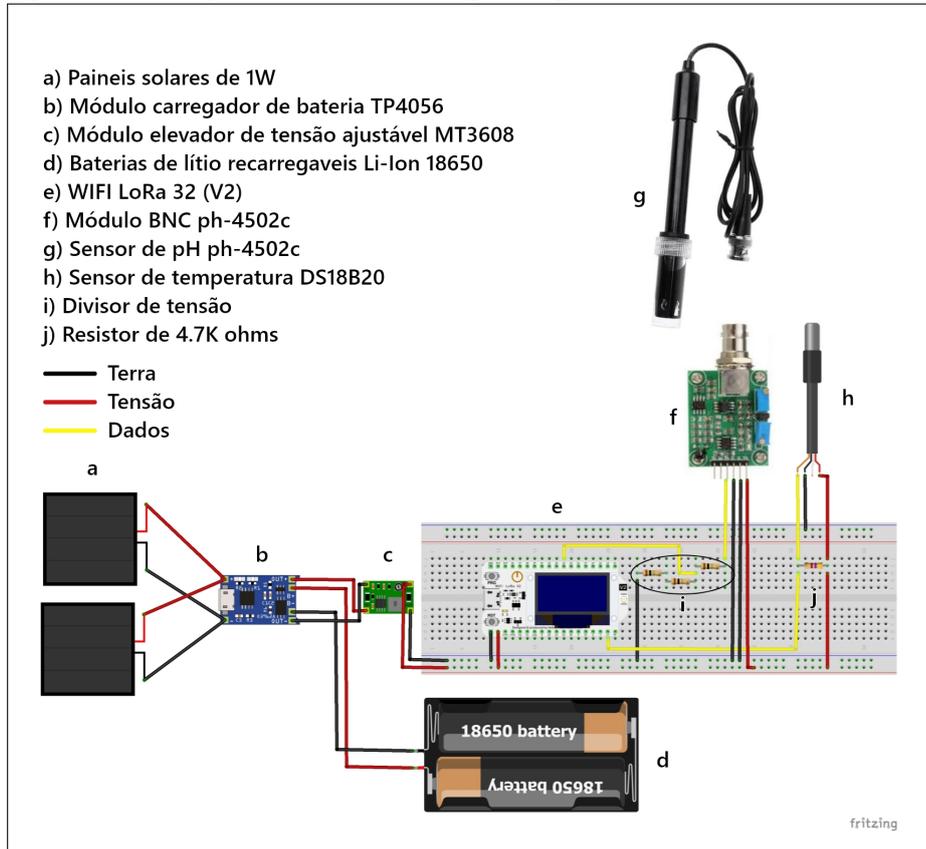


Fonte: elaborada pelo autor

Ambicionando um protótipo energeticamente autossuficiente, foi construído um sistema de alimentação contínua baseado em energia solar. Esse sistema de alimentação foi constituído por dois painéis solares de 1W conectados a um módulo carregador de bateria (modelo TP4056). O módulo carregador fornece energia a placa WIFI LoRa 32 (V2) e sensores

por meio de um módulo elevador de tensão ajustável (modelo MT3608). Em paralelo, o módulo carregador alimenta baterias de lítio recarregáveis de 2.200mAh (modelo Li-Ion 18650). A Figura 14 ilustra a estrutura física do protótipo do nó sensor.

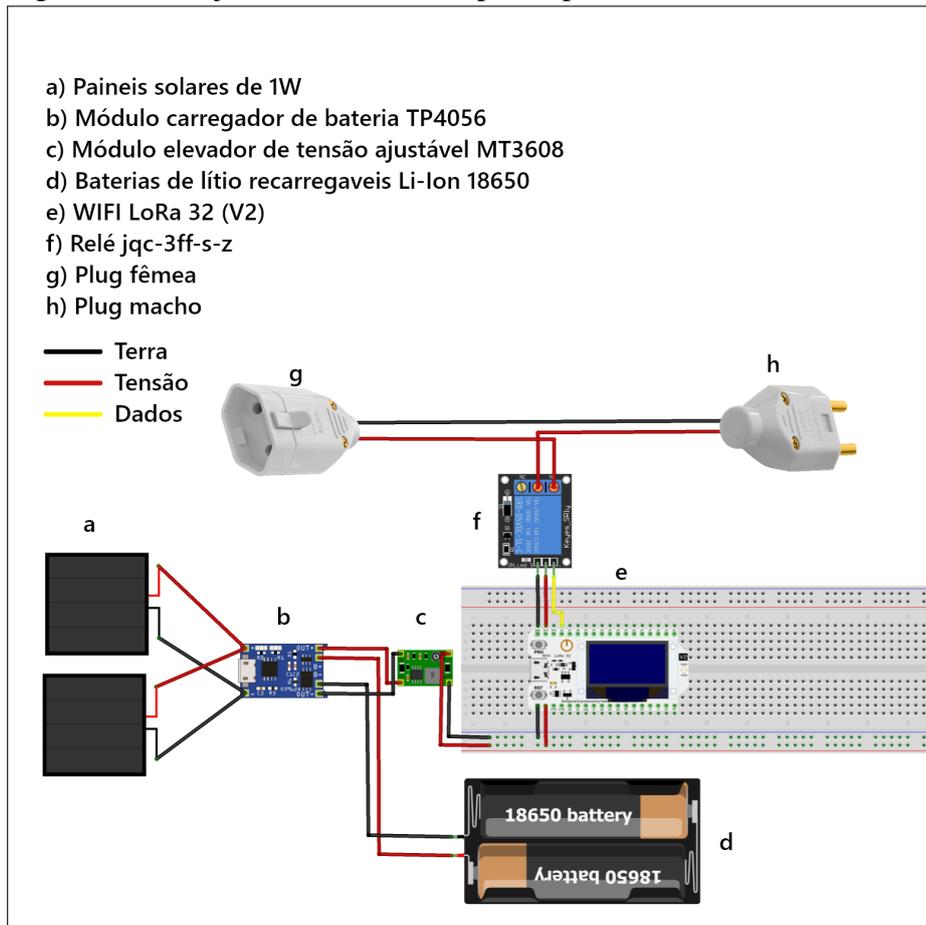
Figura 14 – Projeto de hardware do protótipo do nó sensor



Fonte: elaborada pelo autor

De modo similar, o protótipo do nó atuador conta com uma placa WIFI LoRa 32 (V2) e um sistema de alimentação solar. O nó atuador se diferencia ao possuir um relé (modelo jqc-3ff-s-z) ligado a um plug macho e um plug fêmea. Assim é possível o encaixe de equipamentos elétricos para correção de alguns parâmetros de qualidade da água através do chaveamento de uma tensão de 110v/220v, controlado pela placa microcontroladora. A Figura 15 retrata a estrutura física do protótipo do nó atuador. O gateway é composto exclusivamente pela WIFI LoRa 32 (V2) e case.

Figura 15 – Projeto de hardware do protótipo do nó sensor



Fonte: elaborada pelo autor

As Figuras 16, 17, 18 e 19 exibem nós sensores, nó atuador e gateway respectivamente. A Figura 20 exibe o sistema de alimentação solar sem encapsulamento.

Figura 16 – Nó sensor 01



Fonte: elaborada pelo autor

Figura 17 – Nó sensor 02



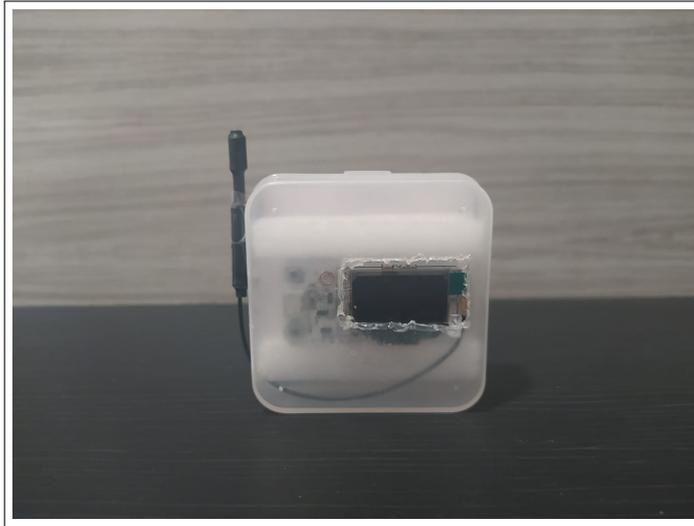
Fonte: elaborada pelo autor

Figura 18 – Nó atuador



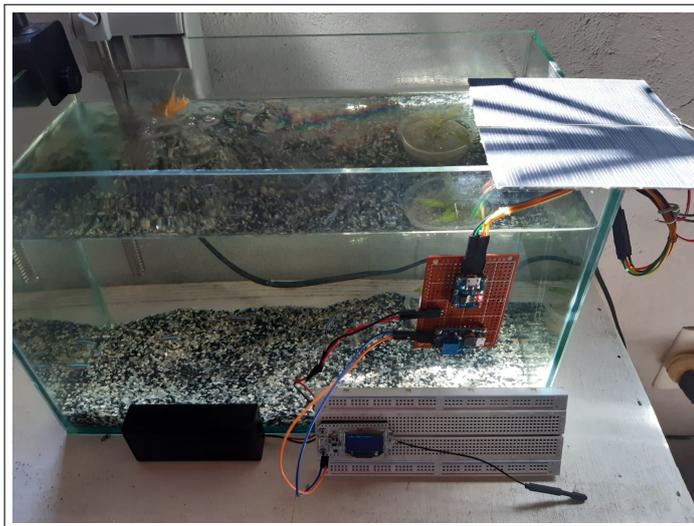
Fonte: elaborada pelo autor

Figura 19 – Gateway



Fonte: elaborada pelo autor

Figura 20 – Sistema de alimentação solar



Fonte: elaborada pelo autor

4.3.2 Software

Todos os softwares embarcados foram escritos em linguagem C++ utilizando a IDE Arduino com auxílio da biblioteca da Heltec, fabricante da WIFI LoRa 32 (V2). Os códigos estão disponíveis nos apêndices A,B e C. Os fluxogramas retratados nas Figuras 21, 22 e 23 ilustram em linhas gerais o funcionamento do protótipo quanto sua programação.

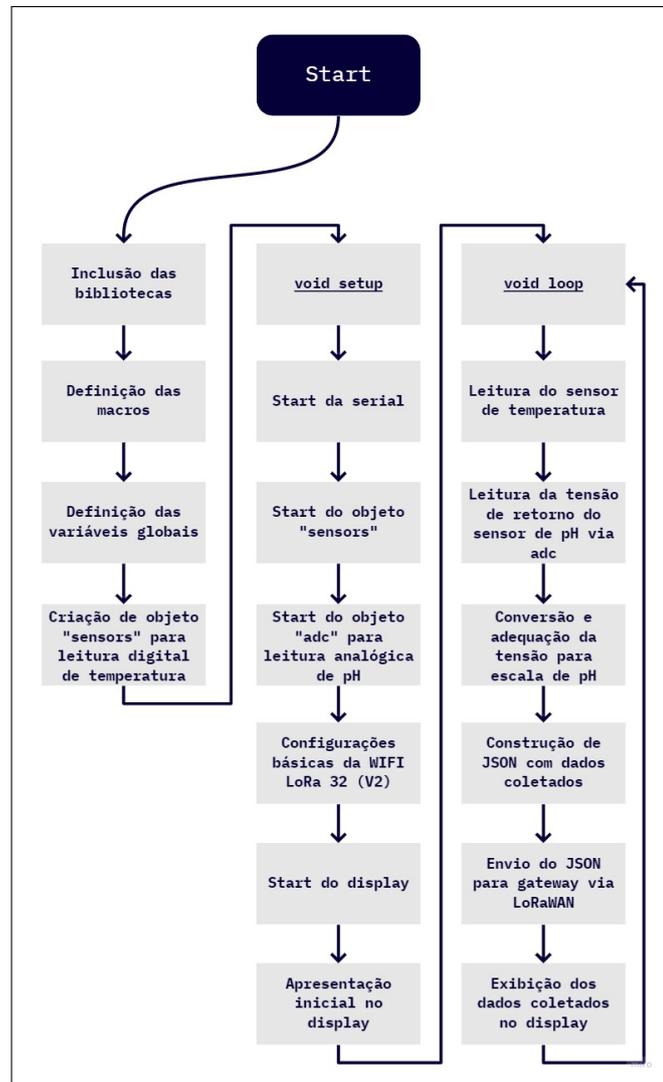
Seguindo o fluxograma da Figura 21, podemos observar que logo após a inicialização e configurações essenciais o dispositivo executa a função **setup**.Essa função no escopo do projeto tem como objetivos: iniciar o objeto **sensors** necessário a leitura da temperatura; iniciar o objeto

adc, necessário para leitura de valores analógicos do sensor de pH, definindo um pino físico para o mesmo; configurar a placa WIFI LoRa 32(V2) definindo dentre outros fatores a frequência de comunicação LoRa a ser utilizada, no caso 915MHz, e por fim a inicialização do display agregada de uma apresentação inicial para o usuário.

Adiante, ao executar a função **loop** é efetuado a leitura dos sensores de temperatura e pH. Na leitura de pH é realizada uma conversão de valores de tensão para valores da escala de pH. Nessa etapa é utilizada uma variável de compensação/calibragem denominada **OFFSET** definida perante testes químicos comparativos. Após a leitura dos sensores é escrita uma mensagem em formato JSON² contendo os valores de temperatura, pH e identificação do nó. Após a mensagem JSON é enviada ao gateway via LoRaWAN e seus dados de sensoriamento exibidos no display seguido do retorno ao início do loop.

² Padrão para armazenamento e transmissão de informações em formato texto amplamente usados em aplicações Web.

Figura 21 – Fluxograma do software do protótipo do nó sensor



Fonte: elaborada pelo autor

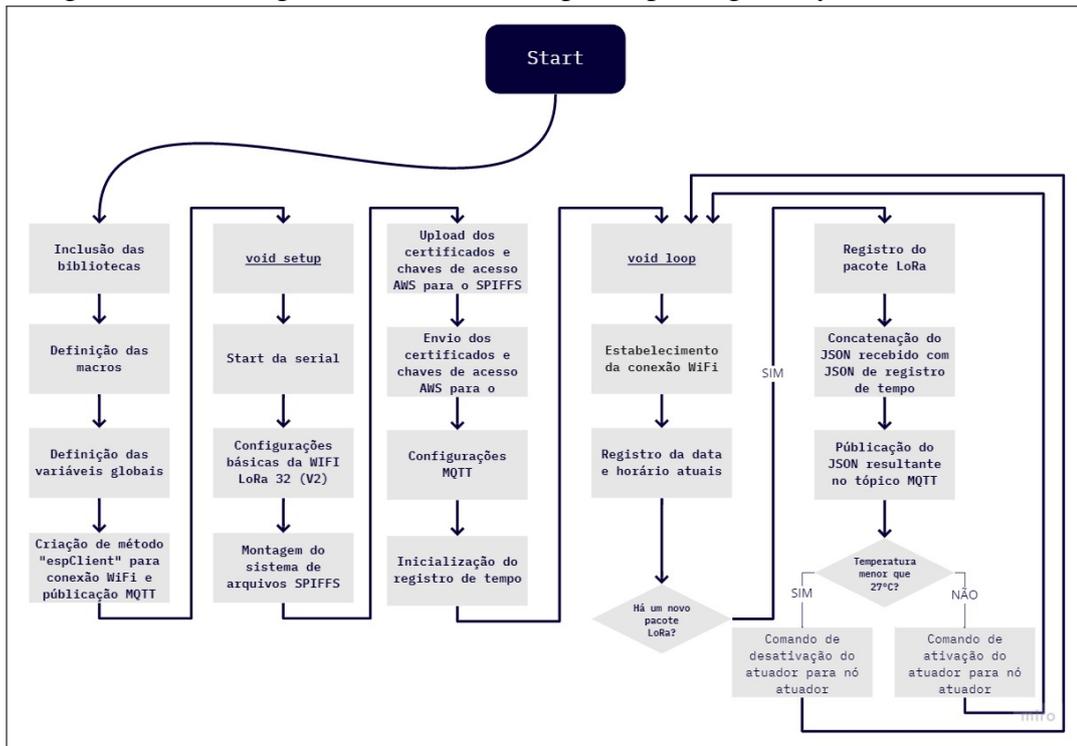
De maneira similar ao software do nó sensor, o programa do gateway, após iniciar e efetuar configurações essenciais como a instância para publicação MQTT, executa a função **setup**. Nessa função é realizado a configuração da placa estabelecendo a mesma faixa de frequência do nó sensor (915MHz). Logo após é montado o sistema de arquivos SPIFFS incumbido de armazenar os certificados e chaves de acesso na memória flash da placa. Esses arquivos são necessários para conexão com a AWS. Em seguida é configurado a comunicação MQTT e iniciado o registro de tempo.

Chegando na função **loop** é estabelecida a conexão WiFi e a captura do tempo atual. Adiante o programa verifica se o nó sensor enviou algum novo pacote. Caso tenha enviado, o pacote LoRa e o JSON enviado pelo nó sensor é concatenado com outro JSON agora gerado

pelo próprio gateway contendo o registro de tempo. O JSON resultante é publicado no tópico MQTT via WiFi.

Prontamente é verificado se a temperatura do nó sensor está adequada (maior de 27° para carcinicultura), caso não esteja, o gateway envia um comando ao nó atuador para que o mesmo acione o aquecedor. Caso a temperatura esteja adequada o gateway enviará o comando de desligamento do aquecedor ao nó atuador. Imediatamente é retornado ao início do loop. a Figura 22 retrata o fluxo o relatado.

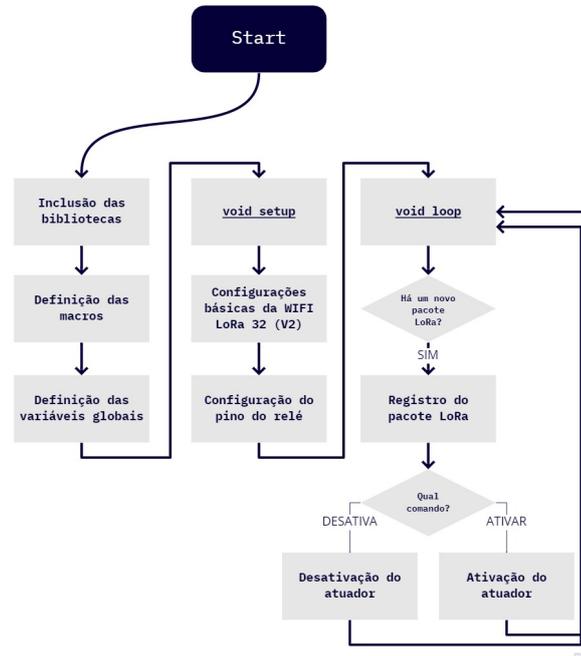
Figura 22 – Fluxograma do software do protótipo do gateway



Fonte: elaborada pelo autor

O programa do nó atuador é simplório, porém, atinge seu objetivo de controle do atuador em questão (aquecedor). Sua rotina basicamente consiste em configurações básicas, alocação de um pino físico para controle do relé e um loop onde é verificado se o gateway enviou algum comando. Se foi enviada alguma instrução ela é lida e então utilizada para o acionamento ou desligamento do atuador. A Figura 23 apresenta em mais detalhes a rotina descrita.

Figura 23 – Fluxograma do software do protótipo do nó atuador



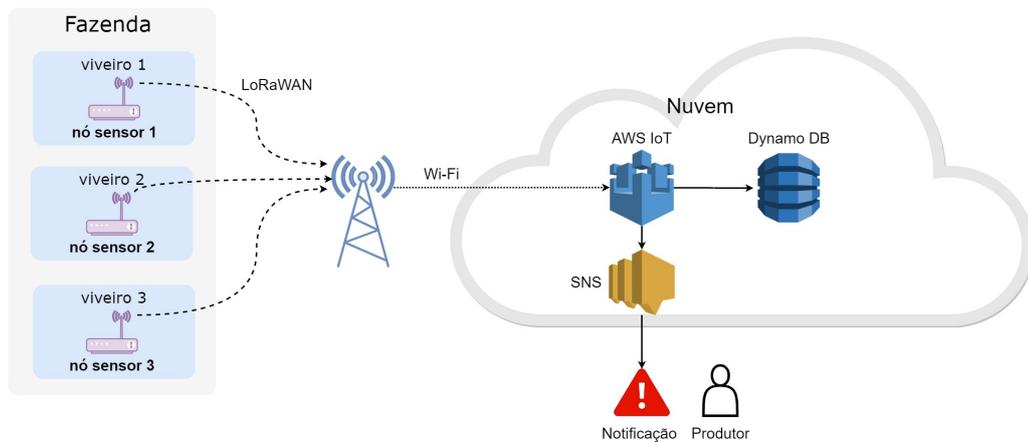
Fonte: elaborada pelo autor

4.3.3 Infraestrutura Cloud

No console AWS foi inicialmente utilizado o serviço Identity and Access Management (IAM) para criação e gestão de usuários e políticas necessárias a integração AWS. Posteriormente foi criada uma thing no serviço AWS IoT. No processo de criação da thing foram gerados certificados e chaves de acesso para vincular o dispositivo físico a thing AWS. Uma vez carregados para flash do dispositivo, como visto no código do gateway, o dispositivo pode atuar na infraestrutura AWS. Logo viabilizando a publicação no tópico MQTT no broker presente na AWS IoT.

Neste mesmo serviço foi implementado três regras ligadas ao broker MQTT. Duas regras são ativadas na presença de valores inadequados de temperatura: uma para valores baixos e outro para valores alto. Essas regras notificam o usuário por e-mail através do serviço Simple Notification Service (SNS). A terceira regra é usada para armazenamento das mensagens JSON no tópico MQTT em um banco de dados DynamoDB, também presente na AWS. A Figura 24 apresenta os serviços AWS inclusos na topologia do sistema proposto.

Figura 24 – Estrutura de Cloud



Fonte: elaborada pelo autor

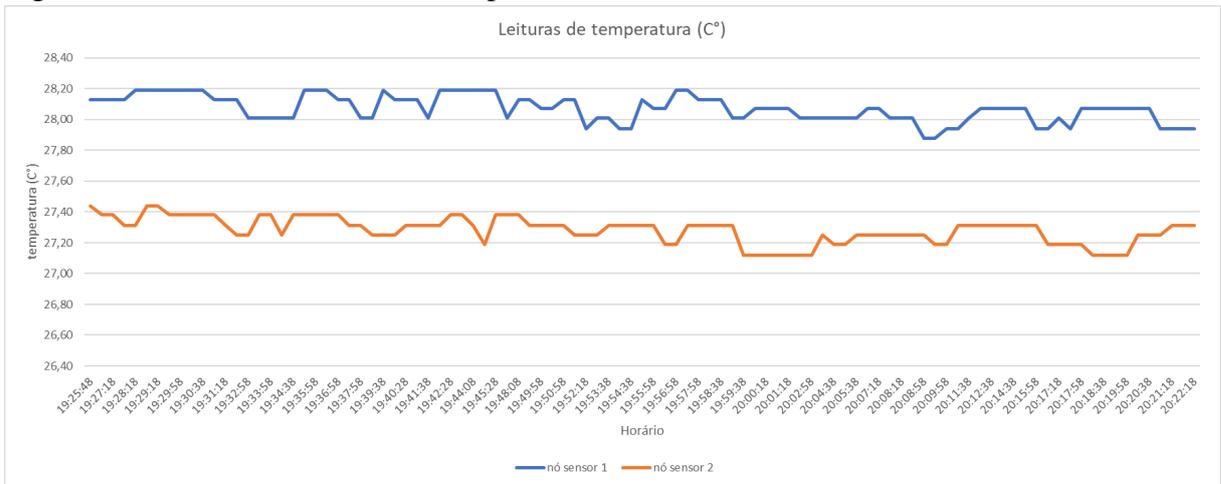
4.4 Testes e resultados obtidos

Nessa seção é descrito o ensaio dos protótipos do sistema. Foram construídos quatro protótipos: um nó gateway, um nó atuador e dois nós sensores a fim de melhor representar uma versão minimal do projeto que esse trabalho propõe. Os nós sensores só se diferenciaram quanto ao encapsulamento e ao uso de uma placa fenolite perfurada no nó sensor 2 em detrimento de uma protoboard, utilizada no nó sensor 1.

4.4.1 *Teste de leitura da temperatura, controle do atuador e notificação*

Para testar a precisão da leitura de temperatura do sistema proposto, os nós sensores foram implantados em aquários de água doce de pequeno porte, onde as temperaturas registradas pelos mesmos foram confrontadas com leituras de um termômetro analógico. As leituras apresentaram precisão de duas casas decimais oscilando em média menos que $0,5^{\circ}\text{C}$ operando em conformidade com as medições do termômetro analógico. A Figura 25 aborta algumas leituras de temperatura. Na Figura é possível nota a baixa oscilação das temperaturas medidas.

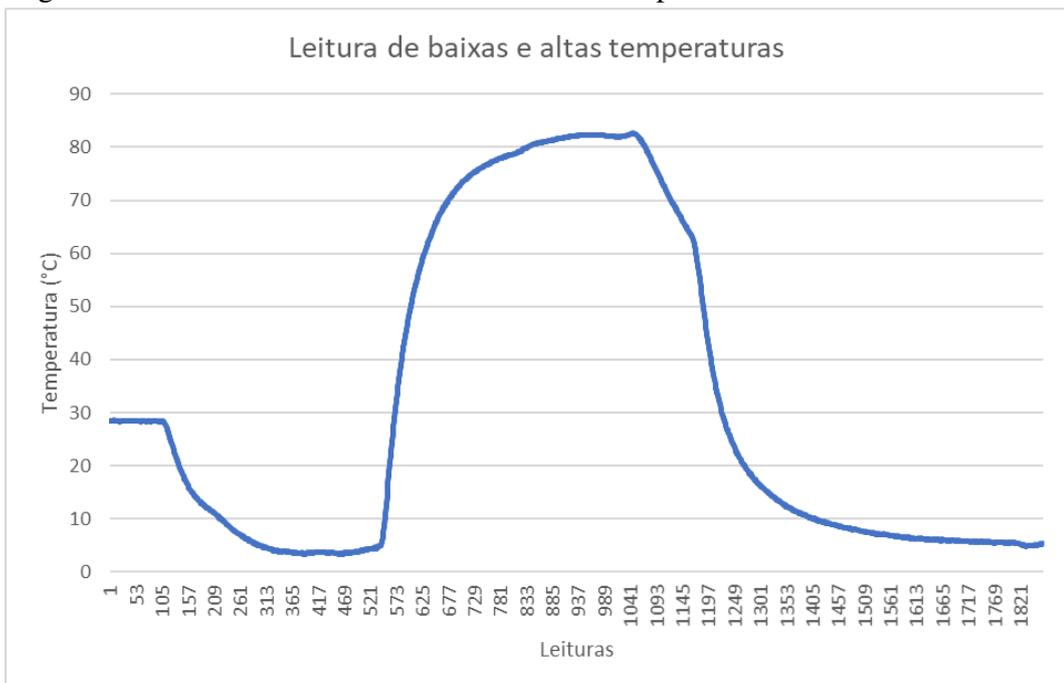
Figura 25 – Gráfico de leitura de temperatura



Fonte: elaborada pelo autor

Ambicionando analisar a abrangência de valores que o sistema está apto a ler, o sensor de temperatura foi submerso em dois recipientes com água. O primeiro contendo água em baixa temperatura, quase em estado sólido, no segundo com água recém fervida. O sensor foi colocado na água fria, em seguida em água quente, depois novamente em água fria. Como ilustrado no gráfico da Figura 26, o sistema foi capaz de registrar temperaturas próximas de 0°C (3,31°C) até temperatura superiores a 80°C (82,62°C).

Figura 26 – Gráfico de leitura de baixas e altas temperaturas

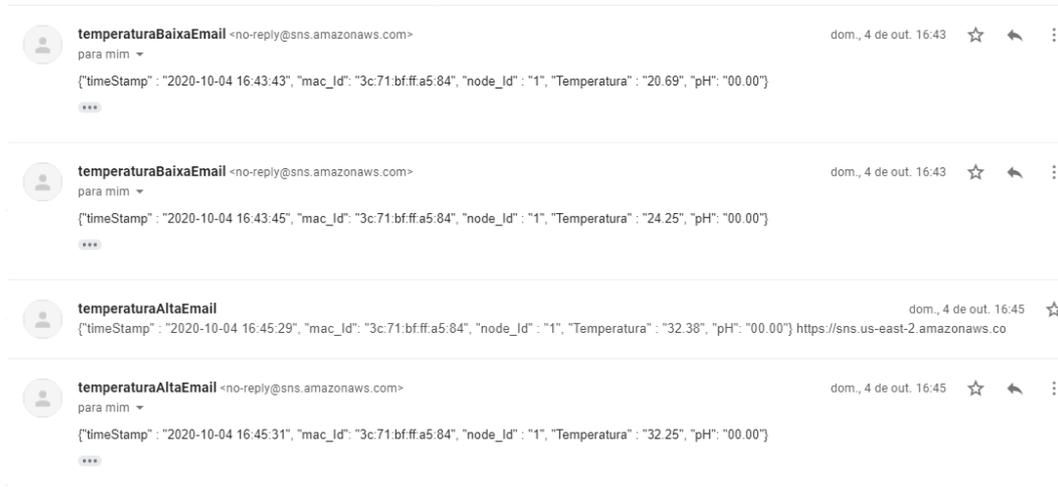


Fonte: elaborada pelo autor

Diante dessas variações também foi analisado o acionamento do atuador pelo sistema.

O mesmo foi ativado e desativado com exatidão. De igual modo o sistema de notificações operou adequadamente alertando o usuário via email, diante de parâmetros inadequados. A Figura 27 retrata alguns notificações via email.

Figura 27 – Amostra de notificações via email

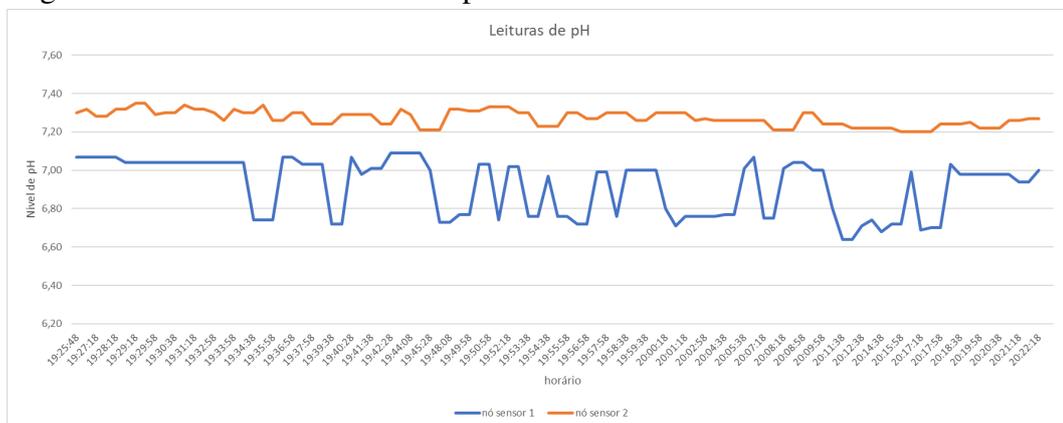


Fonte: elaborada pelo autor

4.4.2 Teste de leitura do pH

O sensoriamento de pH foi testado comparando o resultado de testes químicos e leituras do sensor de pH presente no protótipo. O teste químico empregado foi o Labcon Test Ph Tropical comumente utilizado por carcinicultores. O teste químico agindo sobre amostras de água do aquário número 2 apontou pH entre 7,2 e 7,5. O nó sensor 1 apresentou níveis de pH entre 6,64 e 7,09 ao passo que o nó sensor 2 relatou pH entre 7,2 e 7,35. A Figura 28 mostra os valores de pH registrados por cada nó.

Figura 28 – Gráfico de leituras do pH



Fonte: elaborada pelo autor

4.4.3 Teste de transmissão de pacotes

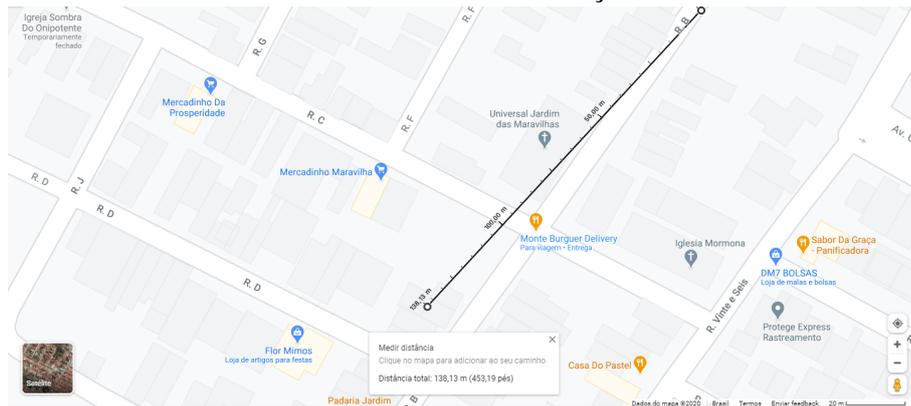
Para saber o alcance da comunicação LoRa um nó sensor e o nó gateway estabeleceram comunicação via LoRaWAN com antenas de 915MHz de pequeno porte. Uma vez em comunicação os dispositivos foram distanciados gradativamente observando-se até quais valores de RSSI permanecia a recepção de pacotes. As Figuras 29 e 30 destacam em mapas a distância alcançadas em ambientes urbanos com densidades de construções civis diferentes.

Figura 29 – Alcance de comunicação LoRa do protótipo em ambiente com alta densidade de construções civis



Fonte: elaborada pelo autor

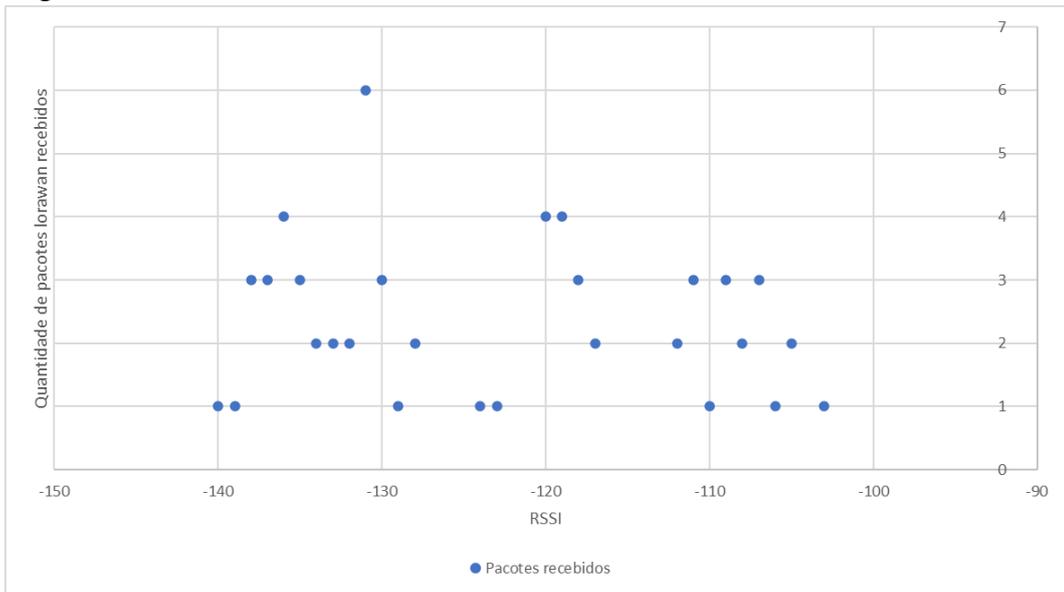
Figura 30 – Alcance de comunicação LoRa do protótipo em ambiente com menor densidade de construções civis



Fonte: elaborada pelo autor

No cenário mais hostil, com alta densidade de edificações, o sistema alcançou aproximadamente 56 metros de abrangência. No cenário com menor número de construções civis entre dispositivos o alcance chegou a 138 metros. A intensidade do sinal recebido chegou ao Received signal strength indication (RSSI) de -140. A Figura 31 ilustra o relatado.

Figura 31 – Gráfico de intensidade do sinal recebido



Fonte: elaborada pelo autor

4.4.4 Teste múltiplos nós de integração do sistema embarcado com estrutura em nuvem

Nessa etapa os dois nós sensores foram implantados em aquários e enviaram dados sensoriais ao gateway. Esse por sua vez foi conectado a estrutura AWS. Na estrutura AWS pode-se constatar a publicação de mensagens no broker MQTT bem como o armazenamento das informações no banco de dados Dynamo DB. Parte tabela do banco de dados referente ao tópico MQTT pode ser observada na Figura 32.

Figura 32 – Tabela do tópico MQTT no banco de dados Dynamo DB

timeStamp	Temperatura	atuador	pH	viveiro
2020-10-16 19:25:48	27.44	OFF	7.30	2
2020-10-16 19:26:18	27.38	OFF	7.32	2
2020-10-16 19:27:18	27.38	OFF	7.28	2
2020-10-16 19:27:38	28.13	OFF	7.07	1
2020-10-16 19:28:18	27.31	OFF	7.32	2
2020-10-16 19:28:38	28.19	OFF	7.04	1
2020-10-16 19:29:18	27.44	OFF	7.35	2

Fonte: elaborada pelo autor

5 CONCLUSÕES E TRABALHOS FUTUROS

Mesmo não sendo uma solução completa para o monitoramento da qualidade da água em carciniculturas, os testes efetuados com o protótipo apresentaram resultados satisfatórios. O alcance de comunicação LoRa relatado de 138 metros, apesar de abaixo do potencial da tecnologia, é mais que suficiente para produções carcinicultoras de pequeno e médio porte, uma vez que estão localizadas em ambientes abertos. Senso possível a implantação de mais gateways ou antenas com maior potência a fim de aumentar a abrangência LoRa tornando o sistema apto para fazendas maiores.

O sistema foi eficiente na medição por meio de sensores analógicos e digitais nas resoluções desejadas. Apresentando-se versátil e autossuficiente energeticamente devido ao sistema fotovoltaico. Sua integração com a estrutura AWS o consolidou, ainda, como solução IoT ao passo que: coleta de dados, os armazena, notifica o usuário, atua no ambiente e torna viável a análise dos dados. Assim, os objetivos impostos no início do projeto foram alcançados.

Contudo, tratando-se de um protótipo, há muita coisa para aprimorar. Assim sendo, algumas propostas para trabalhos futuros podem ser observadas:

- a) Melhor dimensionar o sistema fotovoltaico visando prolongar sua autonomia;
- b) Adicionar mais sensores de qualidade da água como sensores de oxigênio dissolvido, turbidez, salinidade etc;
- c) Realizar o controle de aeradores;
- d) Desenvolver um circuito composto por um microcontrolador ESP32, um módulo LoRa e demais componentes necessários a leitura de sensores de qualidade da água, como divisores de tensão, plugs BNC, portas seriais etc.
- e) Implementação de plicações mobile e web, viabilizando melhor usabilidade do sistema.
- f) Implementação de programas multithreads nos terminais e gateway.
- g) Criptografia das mensagens LoRa e WiFi.

Proponho também o estudo do emprego de outras topologias de rede para a aplicação em questão como redes mesh, por exemplo.

REFERÊNCIAS

- ACQUANATIVA. **A escala de valores de pH**. [S.l.], 2019. Disponível em: <https://www.acquanativa.com.br/aplicacoes/imagens/qualidade-agua/pHscale.jpg>. Acesso em: 02 mai. 2019.
- ARAUJO, J. A.; NORÕES, A. K. M.; MONTEIRO, J. V.; ARAÚJO, R. C. P. d.; SILVA, F. P. d. Eficiência produtiva das fazendas de carcinicultura no estado do ceará. **Revista de Economia e Sociologia Rural**, SciELO Brasil, v. 56, n. 1, p. 35–50, 2018.
- AUTOCORE ROBOTICA. **Sonda Eletrodo Sensor de PH com Módulo PH4502C**. [S.l.], 2020. Disponível em: <https://www.autocorerobotica.com.br/sonda-eletrodo-sensor-de-ph-com-modulo-PH4502C>. Acesso em: 05 out. 2020.
- BEGUM, M. E. A.; HOSSAIN, M. I.; PAPANAGIOTOU, E. Technical efficiency of shrimp farming in bangladesh: an application of the stochastic production frontier approach. **Journal of the world aquaculture society**, Wiley Online Library, v. 44, n. 5, p. 641–654, 2013.
- BLOG 2 ENGENHEIROS. **O que significa pH e Qual sua influência na Qualidade da Água?** [S.l.], 2019. Disponível em: <https://2engenheiros.com/2018/03/20/ph-e-qualidade-da-agua/>. Acesso em: 14 abr. 2019.
- CANALYS. **Global cloud services market Q2 2020**. [S.l.], 2020. Disponível em: <https://canalys.com/newsroom/worldwide-cloud-infrastructure-services-Q2-2020>. Acesso em: 05 out. 2020.
- DIY MORE. **PH4502C PH 4502C Liquid PH Value Detection Detect Sensor Module Monitoring Control For Arduino**. [S.l.], 2020. Disponível em: <https://www.diy-more.cc/products/diy-more-liquid-ph-value-detection-detect-sensor-module-monitoring-control-for-arduino-m>. Acesso em: 05 out. 2020.
- EMBRAPA. **Oxigênio Dissolvido**. [S.l.], 2019. Disponível em: <http://www.cnpma.embrapa.br/projetos/ecoagua/eco/oxigdiss.html>. Acesso em: 14 abr. 2019.
- HENNIG, O.; ANDREATTA, E. Effect of temperature in an intensive nursery system for penaeus paulensis (pérez farfante, 1967). **Aquaculture**, Elsevier, v. 164, n. 1-4, p. 167–172, 1998.
- HERNANDEZ, J.; NUNES, A. Biossegurança no cultivo de camarão marinho: qualidade da água e fatores ambientais. **Revista da ABCC**, v. 3, n. 2, p. 55–59, 2001.
- MAULANA, Y. Y.; WIRANTO, G.; KURNIAWAN, D.; SYAMSU, I.; MAHMUDIN, D. *et al.* Online monitoring of shrimp aquaculture in bangka island using wireless sensor network. **International Journal on Advanced Science, Engineering and Information Technology**, INSIGHT-Indonesian Society for Knowledge and Human Development, v. 8, n. 2, p. 358–364, 2018.
- MEIRELES, A. J. de A.; CASSOLA, R. S.; TUPINAMBÁ, S. V.; QUEIROZ, L. de S. Impactos ambientais decorrentes das atividades da carcinicultura ao longo do litoral cearense, nordeste do brasil. **Mercator-Revista de Geografia da UFC**, Universidade Federal do Ceará, v. 6, n. 12, p. 83–106, 2007.

NUNES, A.; MADRID, R.; ANDRADE, T. Carcinicultura marinha no brasil: passado, presente e futuro. **Panorama da Aqüicultura**, v. 21, p. 26–33, 2011.

ORGANIZATION, A. **The State of World Fisheries and Aquaculture 2014: Opportunities and Challenges**. [S.l.]: Food & Agriculture Org, 2014.

ORTIZ, F. M.; CRUZ, P.; COUTO, R. d. S.; COSTA, L. H. M. Caracterização de uma rede sem-fio de baixa potência e longo alcance para internet das coisas. In: SBC. **Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. [S.l.], 2018.

OSTRENSKY A., C. N. d. A. U.; SILVA, T. da. **A produção integrada na carcinicultura brasileira: vol.2**. Pinhais, Paraná: GIA, 2017. 352 p. (A produção integrada na carcinicultura brasileira, 2). Bibliografia: p. 102 - 119.

OSTRENSKY A., J. D.; STEVANATO DAL PONT. G., G. G.; CASTILHO-WESTPHAL FIER, M. V.; GIROTO COZER, N. F. R.; GARCÍA-MADRIGAL DE ASSIS, U. de A.; SILVA, T. da. **A produção integrada na carcinicultura brasileira: vol.1**. Pinhais, Paraná: GIA, 2017. 288 p. (A produção integrada na carcinicultura brasileira, 1). Bibliografia: p. 55.

PARRA, L.; SENDRA, S.; GARCÍA, L.; LLORET, J. Design and deployment of low-cost sensors for monitoring the water quality and fish behavior in aquaculture tanks during the feeding process. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 3, p. 750, 2018.

PONTES, C.; ARRUDA, M. d. F. Comportamento de *litopenaeus vannamei* (boone)(crustacea, decapoda, penaeidae) em função da oferta do alimento artificial nas fases clara e escura do período de 24 horas. **Revista Brasileira de Zoologia**, v. 22, n. 3, p. 648–652, 2005.

POZZATTI, D. R. **Desenvolvimento de uma rede de sensores sem fios para monitorização de peixes**. 2014. Dissertação (Mestrado em Engenharia Eletrônica e Telecomunicações) — Universidade do Algarve, Faculdade de Ciências e Tecnologia, Universidade do Algarve Campus de Gambelas, Faro, Portugal, 2014.

ROTOTRON. **Pi Temperature Monitor**. [S.l.], 2013. Disponível em: <https://www.rototron.info/projects/pi-temperature-monitor/>. Acesso em: 05 out. 2020.

SEABRA, L. M. A. J. **Camarão *Litopenaeus vannamei*: componentes de importância nutricional na carne e nos resíduos do beneficiamento**. 2011. 79 f. Tese (Doutorado em Ciências da Saúde) — Universidade Federal do Rio Grande do Norte, Natal, 2011.

SEMTECH. **LoRa™ Modulation Basics**. [S.l.], 2015. Disponível em: <https://www.semtech.com/uploads/documents/an1200.22.pdf>. Acesso em: 01 jun. 2019.

SHI, B.; SREERAM, V.; ZHAO, D.; DUAN, S.; JIANG, J. A wireless sensor network-based monitoring system for freshwater fishpond aquaculture. **Biosystems engineering**, Elsevier, v. 172, p. 57–66, 2018.

TEIXEIRA, G. B.; ALMEIDA, J. V. P. de. **Rede LoRa® e protocolo LoRaWAN® aplicados na agricultura de precisão no Brasil**. 2017. 76 f. Trabalho de Conclusão de Curso (Engenharia Eletrônica) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017.

THE THINGS NETWORK. **The Things Uno**. [S.l.], 2017. Disponível em: <https://www.thethingsnetwork.org/marketplace/product/the-things-uno>. Acesso em: 05 jun. 2019.

WYBAN, J.; WALSH, W. A.; GODIN, D. M. Temperature effects on growth, feeding rate and feed conversion of the pacific white shrimp (*penaeus vannamei*). **Aquaculture**, Elsevier, v. 138, n. 1-4, p. 267–279, 1995.

APÊNDICE A – CÓDIGOS-FONTE UTILIZADO NO PROTÓTIPO DO NÓ SENSOR

Código-fonte 1 – Sensoriamento de ambientes aquáticos em C++

```
1 #include "heltec.h"
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5 #include <OneWire.h>
6 #include <DallasTemperature.h>
7 #include <ESP32AnalogRead.h>
8
9 #define OLED_ADDR 0x3c
10 #define OLED_LINE1 0
11 #define OLED_LINE2 10
12 #define OLED_LINE3 20
13 #define OLED_LINE4 30
14 #define OLED_LINE5 40
15 #define OLED_LINE5 60
16 #define SCREEN_WIDTH 128
17 #define SCREEN_HEIGHT 64
18 #define BAND 915E6
19 #define NODE_ID 2
20 #define PACKET_LEN 256
21 #define OFFSET 12.4
22
23 String status_atuador = "OFF";
24 char LoRaPacket[PACKET_LEN];
25 const int onewireBus = 17;
26 float temperatureCelsius = 0;
27 float voltage, phValue = 404;
28
```

```
29 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire
    , 16);
30
31 ESP32AnalogRead adc;
32
33 OneWire oneWire(onewireBus);
34 DallasTemperature sensors(&oneWire);
35
36 void homeScreen () {
37     Heltec.display->setContrast(255);
38     Heltec.display->clear();
39     Heltec.display->setFont(ArialMT_Plain_24);
40     Heltec.display->drawString(0, 0, "UFC Qxd");
41     Heltec.display->setFont(ArialMT_Plain_10);
42     Heltec.display->drawString(0, 25, "Eng. de Computa o")
        ;
43     Heltec.display->setFont(ArialMT_Plain_10);
44     Heltec.display->drawString(0, 45, "Matheus Fernandes");
45     Heltec.display->display();
46     delay(2000);
47 }
48
49 void showData (float temperatureCelsius, float pHValue) {
50     display.clearDisplay();
51     display.setCursor(0, OLED_LINE1);
52     display.println("Temperatura:");
53     display.setCursor(0, OLED_LINE2);
54     display.print(temperatureCelsius);
55     display.setCursor(0, OLED_LINE3);
56     display.println("pH:");
57     display.setCursor(0, OLED_LINE4);
58     display.print(pHValue);
```

```
59   display.display();
60 }
61
62 void sendData (String LoRaPacket) {
63   LoRa.beginPacket();
64   LoRa.print(LoRaPacket);
65   LoRa.endPacket();
66 }
67
68 void setup () {
69
70   Serial.begin(115200);
71   sensors.begin();
72   adc.attach(33);
73   Heltec.begin(true, true, true, true, BAND);
74
75   Wire.begin(4, 15);
76   if(!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)){
77     Serial.println("Display OLED: falha ao inicializar");
78   }else{
79     Serial.println("Display OLED: inicializacao ok");
80   }
81
82   homeScreen();
83
84   display.clearDisplay();
85   display.setTextSize(1);
86   display.setTextColor(WHITE);
87 }
88
89 void loop () {
90
```

```
91 sensors.requestTemperatures();
92 float temperatureCelsius = sensors.getTempCByIndex(0);
93
94 float pHVol = adc.readVoltage();
95 float pHValue = -3.0 * pHVol + OFFSET;
96
97 if (temperatureCelsius < 27.00){
98     status_atuador = "ON";
99 }else{
100     status_atuador = "OFF";
101 }
102 snprintf (LoRaPacket, PACKET_LEN, "\"atuador\": \"%s\",
103           \"viveiro\" : \"%d\", \"Temperatura\" : \"%.2f\", \"pH
104           \": \"%.2f\"}  ", NODE_ID, temperatureCelsius, pHValue
105           , status_atuador.c_str());
106 sendData(LoRaPacket);
107
108 showData(temperatureCelsius, pHValue);
109 }
```

APÊNDICE B – CÓDIGOS-FONTE UTILIZADO NO PROTÓTIPO DO GATEWAY

Código-fonte 2 – Recepção de dados sensoriais, publicação no tópico MQTT e envio de comando ao nó atuador em C++

```
1 #include "SPIFFS.h"
2 #include <WiFiClientSecure.h>
3 #include <PubSubClient.h>
4 #include "heltec.h"
5 #include "ESPDateTime.h"
6 #include "time.h"
7
8 #define BAND 915E6
9 #define BUFFER_LEN 256
10
11 const char* ssid = "SSID";
12 const char* password = "12345678";
13 const char* mqtt_server = "xxxxxxxxxxxxxxxx-ats.iot.region.
    amazonaws.com";
14 const char* ntpServer = "a.st1.ntp.br";
15 const int mqtt_port = 8883;
16 const int daylightOffset_sec = 0;
17 const long gmtOffset_sec = -10800;
18 String Read_rootca;
19 String Read_cert;
20 String Read_privatekey;
21 String loraData = "";
22 String NewData = "";
23 char msg[BUFFER_LEN];
24 char timeStamp[20];
25 int count = 1;
26 int packetSize = 0;
27 long lastMsg = 0;
```

```
28
29 WiFiClientSecure espClient;
30 PubSubClient client(espClient);
31
32 void setup_wifi() {
33     delay(10);
34     Serial.println();
35     Serial.print("Conectando ");
36     Serial.println(ssid);
37
38     WiFi.begin(ssid, password);
39
40     while (WiFi.status() != WL_CONNECTED) {
41         delay(500);
42         Serial.print(".");
43     }
44
45     randomSeed(micros());
46
47     Serial.println("");
48     Serial.println("WiFi conectado");
49     Serial.println("Endere o IP: ");
50     Serial.println(WiFi.localIP());
51 }
52
53 void callback(char* topic, byte* payload, unsigned int
    length) {
54     Serial.print("Mensagem recebida [");
55     Serial.print(topic);
56     Serial.print("] ");
57     for (int i = 0; i < length; i++) {
58         Serial.print((char)payload[i]);
```

```
59     }
60     Serial.println();
61 }
62 void reconnect() {
63     // Loop until we're reconnected
64     while (!client.connected()) {
65         Serial.print("Tentando conex o MQTT ...");
66         // Create a random client ID
67         String clientId = "tcc-sensorX";
68         clientId += String(random(0xffff), HEX);
69         // Attempt to connect
70         if (client.connect(clientId.c_str())) {
71             Serial.println("conectado");
72             // Once connected, publish an announcement...
73             client.publish("telemetria", "hello world");
74             // ... and resubscribe
75             client.subscribe("telemetria-in");
76         } else {
77             Serial.print("Falha, rc=");
78             Serial.print(client.state());
79             Serial.println(" Tente novamente em 5 segundos");
80             // Wait 5 seconds before retrying
81             delay(5000);
82         }
83     }
84 }
85
86 String receiveData(){
87     loraData = "";
88     while(LoRa.available ()) {
89         loraData = LoRa.readString();
90     }
```

```
91     return loraData;
92 }
93
94 void setup() {
95     Serial.begin(115200);
96     Heltec.begin(true, true, true, true, BAND);
97     Serial.setDebugOutput(true);
98     setup_wifi();
99     delay(1000);
100
101     if (!SPIFFS.begin(true)) {
102         Serial.println("Erro ao montar SPIFFS");
103         return;
104     }
105
106     File file2 = SPIFFS.open("/AmazonRootCA1.pem", "r");
107     if (!file2) {
108         Serial.println("Falha ao abrir /AmazonRootCA1.pem para
109             leitura");
110         return;
111     }
112     Serial.println("Root CA File cont m:");
113     while (file2.available()) {
114         Read_rootca = file2.readString();
115         Serial.println(Read_rootca);
116     }
117
118     File file4 = SPIFFS.open("/012345678-certificate.pem.crt"
119         , "r");
120     if (!file4) {
121         Serial.println("Falha ao abrir /012345678-certificate.
122             pem.crt para leitura");
123     }
124 }
```

```
120     return;
121 }
122 Serial.println("Cert File cont m:");
123 while (file4.available()) {
124     Read_cert = file4.readString();
125     Serial.println(Read_cert);
126 }
127
128 File file6 = SPIFFS.open("/012345678-private.pem.key", "r
129     ");
129 if (!file6) {
130     Serial.println("Falha ao abrir /012345678-private.pem.
131         key para leitura");
131     return;
132 }
133 Serial.println("privateKey File cont m:");
134 while (file6.available()) {
135     Read_privatekey = file6.readString();
136     Serial.println(Read_privatekey);
137 }
138
139 char* pRead_rootca;
140 pRead_rootca = (char *)malloc(sizeof(char) * (Read_rootca
141     .length() + 1));
141 strcpy(pRead_rootca, Read_rootca.c_str());
142
143 char* pRead_cert;
144 pRead_cert = (char *)malloc(sizeof(char) * (Read_cert.
145     length() + 1));
145 strcpy(pRead_cert, Read_cert.c_str());
146
147 char* pRead_privatekey;
```

```
148 pRead_privatekey = (char *)malloc(sizeof(char) * (
    Read_privatekey.length() + 1));
149 strcpy(pRead_privatekey, Read_privatekey.c_str());
150
151 Serial.println("=====");
152 Serial.println("Certificados passados para o m todo
    espClient");
153 Serial.println();
154 Serial.println("Root CA:");
155 Serial.write(pRead_rootca);
156 Serial.println("=====");
157 Serial.println();
158 Serial.println("Cert:");
159 Serial.write(pRead_cert);
160 Serial.println("=====");
161 Serial.println();
162 Serial.println("privateKey:");
163 Serial.write(pRead_privatekey);
164 Serial.println("=====");
165
166 espClient.setCACert(pRead_rootca);
167 espClient.setCertificate(pRead_cert);
168 espClient.setPrivateKey(pRead_privatekey);
169
170 client.setServer(mqtt_server, mqtt_port);
171 client.setCallback(callback);
172
173 configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
174 }
175
176 void loop() {
177
```

```
178 if (!client.connected()) {
179     reconnect();
180 }
181 client.loop();
182
183 long now = millis();
184 if (now - lastMsg > 10000) {
185     lastMsg = now;
186
187     struct tm timeinfo;
188     if(!getLocalTime(&timeinfo)){
189         Serial.println("Falha ao obter registro de tempo");
190     }else{
191         strftime(timeStamp,20, "%Y-%m-%d %H:%M:%S", &timeinfo
192             );
193
194         String timeStr = "{\"timeStamp\" : \"" + String(
195             timeStamp) + "\", ";
196
197         packetSize = LoRa.parsePacket();
198         if(packetSize){
199             NewData = receiveData();
200             String aux = NewData;
201             NewData = timeStr + NewData;
202             NewData.toCharArray(msg, 256);
203
204             Serial.print("Publicar mensagem: ");
205             Serial.print(count);
206             Serial.println(msg);
207             client.publish("telemetria", msg);
```

```
208     aux.toCharArray(msg, 256);
209     String sub = strtok(msg, ",");
210     if (sub == "atuador: ON"){
211         LoRa.beginPacket();
212         LoRa.print(1);
213         LoRa.endPacket();
214     }else{
215         LoRa.beginPacket();
216         LoRa.print(0);
217         LoRa.endPacket();
218     }
219     count = count + 1;
220 }
221 }
222 }
```

APÊNDICE C – CÓDIGOS-FONTE UTILIZADO NO PROTÓTIPO DO NÓ ATUADOR

Código-fonte 3 – Controle de atuador C++

```
1 #include "heltec.h"
2
3 #define BAND 915E6
4
5 int rele = 36;
6 String NewData = "";
7
8 String receiveData(){
9     String loraData = "";
10    while(LoRa.available ()) {
11        loraData = LoRa.readString();
12    }
13    return loraData;
14 }
15
16 void setup() {
17     Heltec.begin(true, true, true, true, BAND);
18     pinMode(rele, OUTPUT);
19 }
20
21 void loop() {
22     int packetSize = LoRa.parsePacket();
23     if(packetSize){
24         NewData = receiveData();
25     }
26     if(NewData == "1"){
27         digitalWrite(rele, HIGH);
28     }else{
29         digitalWrite(rele, LOW);
```

30	}
31	}