



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Rauan Bernardo Pires Klaus

**Protótipo de uma plataforma para monitoramento das condições
físico-químicas do Rio Araranguá**

Araranguá
2020

Rauan Bernardo Pires Klaus

**Protótipo de uma plataforma para monitoramento das condições
físico-químicas do Rio Araranguá**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Jim Lau, Dr.

Coorientador: Prof. Analúcia Schiaffino Morales, Dra.

Araranguá

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pires Klaus, Rauan Bernardo
Protótipo de uma plataforma para monitoramento das
condições físico-químicas do Rio Araranguá / Rauan Bernardo
Pires Klaus ; orientador, Jim Lau, coorientadora,
Analúcia Schiaffino Morales, 2020.
52 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2020.

Inclui referências.

1. Engenharia de Computação. 2. Rede de sensores. 3.
nRF24. 4. Rizicultura. I. Lau, Jim . II. Schiaffino
Morales, Analúcia. III. Universidade Federal de Santa
Catarina. Graduação em Engenharia de Computação. IV. Título.

Rauan Bernardo Pires Klaus

Protótipo de uma plataforma para monitoramento das condições físico-químicas do Rio Araranguá

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Computação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.



Documento assinado digitalmente
Fabricio de Oliveira Ourique
Data: 06/12/2020 15:38:29-0300
CPF: 916.167.860-00

Prof. Dr. Fabrício de Oliveira Ourique
Coordenador de curso

Comissão Examinadora



Documento assinado digitalmente
Jim Lau
Data: 03/12/2020 12:01:06-0300
CPF: 613.464.702-00

Prof. Dr. Jim Lau
Orientador



Documento assinado digitalmente
Analucia Schiaffino Morales
Data: 03/12/2020 12:10:58-0300
CPF: 622.256.420-87

Profa. Dra. Analúcia Schiaffino Morales
Coorientador



Documento assinado digitalmente
Ricardo Alexandre Reinaldo de Moraes
Data: 03/12/2020 16:23:16-0300
CPF: 844.812.939-34

Prof. Dr. Ricardo Alexandre Reinaldo De
Moraes



Documento assinado digitalmente
Carla de Abreu Daquino
Data: 05/12/2020 10:55:48-0300
CPF: 024.586.819-48

Profa. Dra. Carla De Abreu D'aquino

Araranguá, 3 de dezembro de 2020

RESUMO

Segundo estudo realizado pela EPAGRI em 2020, a cultura do arroz irrigado foi responsável pelo maior Valor Bruto da Produção (VBP), correspondendo a aproximadamente 865 milhões de reais para o Estado de Santa Catarina. Entre os municípios produtores, destaca-se Araranguá, que concentra 39% do total da área cultivada, ou aproximadamente 58 mil hectares. Apesar dos bons resultados em relação a esta cultura, o total de produtores, familiares e não familiares, caiu aproximadamente 27% no período de 2006 a 2017, resultado ligado aos altos custos de produção e manutenção das lavouras. Assim sendo, está em desenvolvimento desde 2018, na Universidade Federal de Santa Catarina, o projeto SisMoRA - Sistema de Monitoramento do Rio Araranguá. O projeto SisMoRA têm como objetivo o desenvolvimento de uma plataforma para o monitoramento das condições físico-químicas do Rio Araranguá, em termos de salinidade, pH e temperatura da água. Este trabalho tem como base o projeto SisMoRA, e, especificamente neste trabalho de conclusão de curso, será apresentado um protótipo de uma plataforma capaz de monitorar índices físico-químicos da água do Rio Araranguá e permitir que rizicultores da região tenham acesso facilitado e confiável a esses dados, para melhor tomar decisões em relação a irrigação de suas plantações. Com caráter de protótipo, este trabalho concentrou-se na validação do caminho dos dados, desde a aferição dos dados feita através de uma rede sensorial até a disponibilização desses dados em um aplicativo para *smartphones*. Por fim, verificou-se que a aplicabilidade do protótipo é alta e este chega muito próximo a solucionar completamente o problema em questão. Algumas configurações e ajustes, não menos importantes, ainda estão pendentes e estão listadas como sugestões de trabalhos futuros ao final deste documento.

Palavras-chave: Plataforma de monitoramento. IoT; Redes de sensores sem fio. Aplicativo para smartphone. Agricultura de precisão.

ABSTRACT

According to a study carried out by EPAGRI in 2020, the irrigated rice culture was responsible for the highest Gross Production Value (VBP), corresponding to approximately R\$ 865 million for the State of Santa Catarina. Among the producing municipalities, Araranguá stands out, which concentrates 39% of the total cultivated area, or approximately 58 thousand hectares. Despite the good results in relation to this crop, the total of producers, both family and non-family, fell by approximately 27% in the period from 2006 to 2017, a result linked to the high costs of production and maintenance of crops. Therefore, the SisMoRA - Araranguá's River Monitoring System project is under development at the Federal University of Santa Catarina. The SisMoRA project aims to develop a platform for monitoring the physical and chemical conditions of the Araranguá River, in terms of salinity, pH and water temperature. This work is based on the SisMoRA project, and it will present a prototype of a platform capable of monitoring physical-chemical indexes of the Araranguá River water and allowing rice farmers in the region to have easy and reliable access to these data, to better make decisions in irrigation of their plantations. As a prototype, this work focused on the validation of the data path, from the measurement of the data made through a sensory network to the availability of this data in a smartphone application. Finally, it was found that the applicability of the prototype is high and it comes very close to completely solving the problem in question. Some settings and adjustments, equally important, are still pending and are listed as suggestions for future improvements at the end of this document.

Keywords: Monitoring platform. IoT. Wireless Sensor networks. Smartphone apps. Precision agriculture

LISTA DE FIGURAS

Figura 1 – Paradigma IoT	14
Figura 2 – nRF24L01+ PA/LNA	18
Figura 3 – Shield GSM SIM900	19
Figura 4 – Escala Potencial Hidrogênico (pH)	20
Figura 5 – Sensor SEN0161	21
Figura 6 – Sensor KS0429	22
Figura 7 – Visão macro dos componentes da plataforma	26
Figura 8 – Componentes nó Coletor	27
Figura 9 – Componentes nó Intermediário	28
Figura 10 – Componentes nó Master (Gateway)	28
Figura 11 – Topologia da rede - Árvore	29
Figura 12 – Topologia da rede aplicada no protótipo	31
Figura 13 – Caminho de dados na <i>Application Interface</i> (API)	32
Figura 14 – Exemplo de configuração do documento <i>uniquePositionGroup</i>	35
Figura 15 – Tela inicial - Tela de histórico	36
Figura 16 – Tela de histórico por bomba/sensor - Tela de gráficos	37
Figura 17 – Teste de API	40
Figura 18 – Localização geográfica do ponto de medição	41
Figura 19 – Foto do ponto de medição	42
Figura 20 – Medição com nó coletor	43
Figura 21 – Gráficos das leituras	44
Figura 22 – Adaptador nRF24	52

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Interface</i>
EPAGRI	Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina
LoRa	<i>Long Range</i>
pH	Potencial Hidrogênico
RTC	<i>Real-Time Clock</i>
SDK	<i>Software Development Kit</i>
SisMoRA	Sistema de Monitoramento do Rio Araranguá
TDS	Total de sólidos dissolvidos
UFSC	Universidade Federal de Santa Catarina

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA	10
1.2	PROBLEMÁTICA	11
1.3	OBJETIVOS	12
1.3.1	Objetivo geral	12
1.3.2	Objetivos específicos	12
1.4	METODOLOGIA	12
1.5	ORGANIZAÇÃO DO TRABALHO	13
2	REFERENCIAL TEÓRICO	14
2.1	IOT E AGRICULTURA	14
2.2	REDES DE SENSORES SEM FIO	16
2.2.1	Características dos nós	16
2.3	NRF24L01+	17
2.3.1	nRF24I01+ PA/LNA	17
2.4	GSM-GPRS	18
2.5	SENSORES E PARÂMETROS RELEVANTES	19
2.5.1	pH	19
2.5.2	Sensor de pH - SEN0161	20
2.5.3	Total de sólidos dissolvidos (TDS) - Total de sólidos dissolvidos	21
2.5.4	Sensor de TDS - KS0429	21
2.6	COMPUTAÇÃO EM NUVEM	22
2.6.1	Google Firebase	24
2.7	PLATAFORMA ANDROID	25
2.7.1	React Native	25
3	MODELAGEM DO SISTEMA	26
3.1	REDE DE SENSORES SEM FIO	27
3.2	DESENVOLVIMENTO DA API	31
3.2.1	Rota de entrada - <i>/sensordata</i>	32
3.2.2	Validação dos dados	33
3.2.3	Análise e classificação	33
3.2.4	Envio ao banco de dados	33
3.2.5	Rotas de acesso aos dados	34
3.2.6	Configuração inicial do Banco de Dados	34
3.3	DESENVOLVIMENTO DO APLICATIVO	35
4	TESTES E VALIDAÇÃO DO MODELO	38
4.1	TESTE DA REDE SENSORIAL	38
4.2	TESTE DA API	39

4.3	TESTE DO APLICATIVO MÓVEL	40
4.4	TESTE EM CAMPO	41
5	CONCLUSÃO	45
5.1	TRABALHOS FUTUROS	45
5.1.1	Alimentação dos nós	45
5.1.2	Case para os nós	46
5.1.3	Substituição de sensores	46
	REFERÊNCIAS	47
	ANEXO A – CONFIGURAÇÃO DOS NÓS	50

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

Considerado o segundo maior estado produtor, ficando atrás apenas do Rio Grande do Sul no ranking, Santa Catarina, na última safra (2019), plantou e colheu mais de 149.000 hectares de arroz irrigado. Segundo estudo realizado pela Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina (EPAGRI) em 2020 (EPAGRI... , s.d.), a cultura do arroz irrigado foi responsável pelo maior Valor Bruto da Produção (VBP), correspondendo a aproximadamente 865 milhões de reais para o Estado Catarinense. Entre os municípios produtores, destaca-se Araranguá, que concentra 39% do total da área cultivada, ou aproximadamente 58 mil hectares. O arroz é uma planta hidrófila, ou seja, sua cultura necessita muito de água para se desenvolver. A maior parte da produção em SC é de arroz irrigado onde o seu plantio é feito em terreno de várzea, também chamado de planície de inundação. A várzea é caracterizada pela região à margem de um curso d'água que inunda durante períodos de cheias, permitindo assim uma grande economia da água necessária à irrigação. Em termos de qualidade, as melhores várzeas são aquelas que oferecem um sub-solo impermeável, com uma altura de aproximadamente entre 20 a 25 centímetros da superfície.

No que diz respeito ao início da cultura do arroz na região sul de Santa Catarina, teve início com a chegada dos imigrantes italianos por volta de 1880. Criado em 1981, o Programa de Aproveitamento das Várzeas (PROVÁRZEAS) acelerou muito a produção de arroz. Este programa promovia a utilização de grandes quantidades de produtos químicos além de intensa mecanização do processo de plantio e colheita (PROFILL, 2014). Inicialmente, a forma de cultivo limitava-se ao plantio de arroz sequeiro¹, em contraste ao formato irrigado, que é mais presente atualmente na região. De acordo com relatos, em 1977 os arrozais representavam cerca de 22.529 hectares da área agricultável na bacia, valor que aumentou gradativamente em função de novas tecnologias e insumos vindas em parte do programa PROVÁRZEAS (PROFILL, 2014).

Apesar dos incentivos aos produtores, segundo análises feitas pelo IBGE - Instituto Brasileiro de Geografia e Estatística (IBGE, s.d.[a]) e (IBGE, s.d.[b]), o total de estabelecimentos produtores de arroz no estado de Santa Catarina caiu aproximadamente 27% no ano de 2017 em relação a 2006. O número de famílias, que compunham o grupo de agricultura familiar, também sofreu redução entre os últimos dois recenseamentos, passando de 7,093 famílias em 2006 para 4,845 famílias em 2017. Essa redução geral da produção de arroz em Santa Catarina está ligada principalmente aos altos custos de produção e manutenção do solo, características da rizicultura. Por este motivo acredita-se que novas tecnologias que auxiliem e agilizem o processo produtivo, como é o caso da solução proposta nesse trabalho, possam incentivar a produção de arroz na região.

¹ Arroz produzido em terras altas sem a presença de uma lâmina d'água como no caso do arroz irrigado

Desde 2018, a Universidade Federal de Santa Catarina (UFSC), campus de Araranguá, desenvolve o projeto "Sistema de Monitoramento do Rio Araranguá" ou Sistema de Monitoramento do Rio Araranguá (SisMoRA). Este projeto tem como objetivo a idealização, construção e implantação de um sistema de monitoramento remoto no Rio Araranguá para. O projeto SisMoRA tem como intuito criar uma plataforma que permita aos agricultores da região acompanhar os índices físico-químicos da água do Rio Araranguá relevantes às atividades de produção. A plataforma aqui apresentada tem como objetivo auxiliar na solução do projeto SisMoRA, apresentando um protótipo para a plataforma que possibilite o monitoramento de condições físico-químicas da água do Rio Araranguá. Considerando o amplo escopo do projeto SisMoRA, este trabalho foi construído no intuito de validar uma possível arquitetura de plataforma para sua solução, considerando desde a rede física até o aplicativo para *smartphone*. Com caráter experimental, a plataforma proposta no presente trabalho, não reflete a solução final para o projeto, mas representa uma etapa importante para o andamento das atividades de pesquisa do projeto SisMoRA.

1.2 PROBLEMÁTICA

Fazendo parte da região hidrográfica *RH 10 - Extremo Sul Catarinense*, o Rio Araranguá é formado pela junção dos rios Mãe Luiza e Itaupava. Considerando as atividades de mineração e beneficiamento de carvão, atividades industriais e a utilização de defensivos agrícolas nas áreas cultivadas, a bacia hidrográfica do Rio Araranguá têm suas águas comprometidas em qualidade, fazendo com que seus índices físico-químicos, como salinidade e pH, sofram alterações severas (PROFILL, 2014). Segundo um relatório elaborado pela EPAGRI (EPAGRI.2011, 2011), a salinidade elevada da água de irrigação aumenta o potencial osmótico e reduz o potencial hídrico, dificultando a absorção de água pelas plantas, culminando na redução do aproveitamento agrícola e elevação de custos.

Em relação, a utilização do solo na Bacia hidrográfica do Rio Araranguá, verifica-se que aproximadamente 35% do solo é utilizado para a Agricultura, em sua maioria culturas de arroz e fumo, florestas em estágio médio ou avançado representam 39%, pastagens e campos naturais tomam 18% do solo, áreas urbanizadas equivalem a 3,13%, e a mineração corresponde em até 1% do solo utilizável (PROFILL, 2014). Considerando que o arroz irrigado possui uma tolerância moderada a salinidade da água, a EPAGRI(EPAGRI.2011, 2011) aponta algumas soluções para o problema, entre elas: "Monitoramento Climático: Monitorar continuamente a vazão do Rio e a condutividade elétrica da água e previsão do tempo, associado a um processo eficiente de comunicação, contribuirão no processo preventivo para evitar graves danos aos produtores". Nos moldes atuais, o monitoramento de índices físico-químicos relevantes é feito de maneira manual e arcaica, sendo necessário que os interessados, em sua maioria agricultores da região, se desloquem até a margem do Rio Araranguá e realizem a aferição dos dados manualmente, com o auxílio de medidores portáteis. A troca de informações a respeito dos índices lidos, quando ocorre, é feita através

de telefones e grupos em aplicativos de celular.

Visando agilizar este processo e focando nos índices relevantes para a cultura do arroz, o presente trabalho têm como objetivo a elaboração de um protótipo de uma plataforma que seja capaz de monitorar os índices físico-químicos em questão, e apresentá-los de maneira rápida, confiável e barata aos agricultores da região. Dessa forma, acredita-se que o monitoramento das condições de salinidade e pH na porção do baixo e médio estuário do Rio Araranguá possa favorecer o planejamento para irrigação, evitando a retirada de água do rio quando a mesma não apresenta os índices adequados para a cultura do arroz.

1.3 OBJETIVOS

1.3.1 Objetivo geral

O presente trabalho tem como principal objetivo projetar um protótipo para uma plataforma de monitoramento de dados, capaz de coletar e analisar parâmetros de pH e TDS do Rio Araranguá e disponibilizá-los através do uso de um dispositivo móvel ou *smartphone*.

1.3.2 Objetivos específicos

Considerando o objetivo geral do trabalho, destacam-se os seguintes objetivos específicos:

- Elaborar uma rede experimental com NRF2401 para os testes do protótipo;
- Desenvolver a programação para os nós da rede de sensores;
- Desenvolver o módulo gateway através de rede celular (GSM-GPRS);
- Investigar ferramentas para o desenvolvimento de uma API e banco de dados;
- Construir um aplicativo para acessar os dados da plataforma;
- Testar e validar o modelo proposto.

1.4 METODOLOGIA

Este trabalho é uma pesquisa aplicada, e utilizou recursos de pesquisa em material bibliográfico na literatura científica e tecnológica para as decisões em relação a escolhas ferramentais, e quantitativamente analisar a validade da solução proposta.

Para o desenvolvimento do protótipo da plataforma experimental, a seguinte sequência de atividades foi realizada durante o trabalho:

- Pesquisar soluções na literatura com o uso de TDS para irrigação;

- Pesquisar sobre redes sem fio para solucionar a comunicação interna da rede de sensores, buscando uma solução de baixo custo;
- Modelar e programar a rede de sensores e comunicação via GSM-GPRS;
- Elaborar uma API juntamente a um banco de dados capaz de receber e disponibilizar os dados lidos pela rede de sensores;
- Conceitualizar e construir um aplicativo (*smartphone*) para dispositivos Android ou para permitir o acesso aos dados lidos pela rede sensorial.
- Realizar testes unitários e em campo para validar a eficácia da solução desenvolvida frente ao problema de projeto.

1.5 ORGANIZAÇÃO DO TRABALHO

Esta monografia apresenta o desenvolvimento do trabalho de conclusão de curso de graduação em Engenharia de Computação da UFSC e está organizada em cinco capítulos.

- Capítulo 2 - Apresenta os fundamentos teóricos e ferramentas que foram empregadas no desenvolvimento do trabalho. São apresentados conceitos de *Internet of Things* (IoT), redes de sensores sem fio (RSSF), sensores e parâmetros analisados, paradigma de computação em nuvem e aplicações móveis.
- Capítulo 3 - Os aspectos da modelagem empregada no desenvolvimento do protótipo da plataforma experimental estão detalhados no terceiro capítulo. Destacam-se como módulos principais do sistema: a rede de sensores, a API e a aplicação para *smartphone*.
- Capítulo 4 - No quarto capítulo estão os testes aplicados à cada módulo do sistema, bem como, os testes de integração da plataforma (protótipo), feitos em campo. Também são apresentadas as leituras feitas pela rede de sensores e as dificuldades encontradas no processo de validação do modelo.
- Capítulo 5 - O último capítulo apresenta as considerações finais do trabalho e a conclusão. Além disso, ao final do capítulo, são sugeridos trabalhos futuros que foram identificados ao longo do desenvolvimento do protótipo.

Ao final, encontram-se as referências bibliográficas e um anexo com os códigos de programação utilizados no desenvolvimento do protótipo.

2 REFERENCIAL TEÓRICO

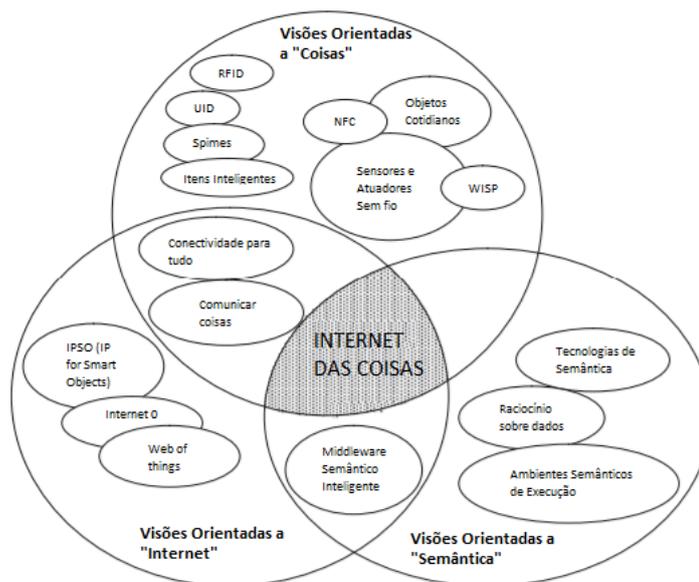
2.1 IOT E AGRICULTURA

Internet das coisas ou, do inglês, Internet of Things (IoT) pode ser definida como uma rede de dispositivos computacionais conectados através da Internet que, trocando dados e informações entre si, cooperam para a realização de um objetivo comum. A base para o funcionamento de uma rede IoT são os sensores e dispositivos acoplados a meios físicos, dos quais deseja-se extrair informação. Uma rede IoT também necessita que existam agentes responsáveis pelo controle e análise das informações coletadas, seja este pertencente a rede dos sensores, como um *gateway* ou que a informação seja processada por um agente externo, como um servidor ou *middleware* (BUDAKOTI, 2018).

A definição de uma rede IOT pode ser muito abrangente, de maneira que diversos sistemas que pouco possuem em comum possam ser enquadrados sob o mesmo nome, desde que estes atendam as premissas básicas de um sistema distribuído. Entre essas premissas podem ser citadas a participação de diversos pontos de comunicação e geração de dados bem como a necessidade de troca de informação entre os mesmos para que o objetivo final da rede seja alcançado. A necessidade de processamento dos dados coletados para futuras inferências de informações relevantes também é característica presente em grande parte das redes IoT operantes.(VANDIKAS, K., TSIATSIS, V., 2014)

A Figura 1 ilustra o paradigma da Internet das Coisas inserida num contexto de visão orientada a internet, tecnologias e comunicação intra-dispositivos.

Figura 1 – Paradigma IoT



Fonte: (ROOSTAEI, 2018)

Entre uma das aplicações mais promissoras para a IoT, destaca-se o uso na agricultura digital. Sistemas compostos por uma rede de sensores e diversos outros recursos, podem ser interligados por mecanismos de comunicação e entregar aos produtores rurais, as informações relacionadas as etapas de produção. Entre as tecnologias mais promissoras para o desenvolvimento destes sistemas destacam-se o LoRA e o uso de telefonia celular, para auxiliar na cobertura de larga escala. A agricultura de precisão ou agricultura digital, pode melhorar o rendimento da produção, reduzir os custos com insumos mal dimensionados, melhorar o controle sobre a produção e minimizar significativamente os riscos e impactos ao meio ambiente (ZHAO *et al.*, 2017).

A infraestrutura da rede normalmente é composta por imensa gama de sensores, redes de comunicação sem fio, veículos aéreos não tripulados e armazenamento dos dados através de *cloud computing*. A integração dos componentes em uma estrutura de IoT melhora significativamente as soluções para monitoramento, modelagem de produção, previsão e tomadas de decisão no ambiente de produção. Os dados de entrada coletados no ambiente alimentam os sistemas especializados para o gerenciamento de processos na agricultura de precisão nas mais variadas culturas. Além disso, tem sido empregado também recursos de aprendizado de máquina para melhorar os resultados de avaliação e comparação dos dados para uma tomada de decisão mais precisa.

Dentre as diversas aplicações para que a tecnologia auxilie na produção agrícola, destacam-se a coleta de informações sobre o clima, as culturas e o solo; o monitoramento do desenvolvimento da cultura na lavoura; o gerenciamento do rodízio das culturas em uma única propriedade; o manejo de fertilizantes e da irrigação para diferentes propriedades de terra; a avaliação de requisitos de culturas para condições climáticas e de solo diferentes, e por fim, a atuação em soluções para campo com características proativas em vez de soluções reativas, visando a sustentabilidade ambiental e social das famílias que dependem do campo (AQEEL-UR-REHMAN, 2014).

Alguns trabalhos, considerando redes de comunicação em agricultura de precisão, consideram que as condições climáticas e o uso de sensores em ambientes abertos exigem diferenciais nas tecnologias, desta forma são propostos protocolos de roteamento para melhorar a eficiência de energia nas baterias de sensores e minimizar os danos causados aos equipamentos utilizados para o sensoriamento (QURESHI, K. N., 2020). Um estudo foi realizado comparando as três tecnologias de comunicação sem fio mais utilizadas na agricultura de precisão: comunicação sem fio de banda estreita (NB-IoT), Long Range (*Long Range* (LoRa)) e ZigBee, medindo tempo de comunicação, consumo de energia e custos. Neste artigo, os autores concluem que para a monitoração em campos abertos, devido sua capacidade de cobertura e os custos associados a manutenção do canal, o LoRa foi apontado como a melhor alternativa (FENG, X., 2019).

No presente trabalho, serão empregados os módulos nRF24L01+ pa/lna para o teste da rede devido o seu baixo custo e o alcance que pode chegar até 1km. A comunicação

entre a rede de sensores e API é feita através de rede celular, por uma placa GSM-GPRS SIM900. Os detalhes técnicos dos módulos serão apresentados nas próximas seções.

2.2 REDES DE SENSORES SEM FIO

Redes de Sensores Sem Fio (RSSF) são um grupo de componentes físicos de natureza computacional, capaz de coletar, processar e transmitir dados em um determinado ambiente. A transmissão dos dados em uma RSSF intercorre através de rádio-frequência, possibilitando que a comunicação entre nós, assim como são chamados os componentes individuais de uma RSSF, seja feita a distância e sem necessidade de cabos ou outros meios físicos para interligá-los (MIGNACO, A. G., 2005).

Usualmente, uma hierarquia mínima é presente na estrutura de uma RSSF, de maneira que diferentes nós na rede sejam responsáveis por operar diferentes funções. Como uma RSSF é altamente orientada à aplicação, esta deve ser mínima frente ao objetivo para qual foi projetada, ou seja, precisa cumprir seu objetivo e ao mesmo tempo apresentar custo de aplicação e operação reduzidos. Por consequência, os nós têm baixo poder computacional e de armazenamento, fazendo com que dados provenientes da rede sejam processados e armazenados em servidores externos. Como costumam operar em protocolos próprios e específicos, RSSFs necessitam de um *Gateway*, mais poderoso computacionalmente, que recebe os dados vindos dos nós e os repassa para um controlador ou servidor externo (YICK, J., 2008).

2.2.1 Características dos nós

Um nó de uma RSSF é frequentemente composto por quatro partes principais. Uma unidade de sensoriamento, responsável pela aquisição dos dados, uma unidade de processamento, responsável pelo controle operacional do nó, uma unidade de energia, encarregada de alimentar o nó, e uma unidade transceptora, que realiza a comunicação entre nós e entre nó e gateway (AKYILDIZ, I., VURAN, M., 2010).

- **Unidade de sensoriamento:** Responsável pela aquisição do dado a partir do ambiente onde o nó está instalado. Alguns nós não possuem o sensor integrado em seu hardware, mas possuem terminais digitais e analógicos que permitem que sensores externos possam ser acoplados.
- **Unidade de processamento:** Responsável pelo controle do comportamento do nó. Processa e organiza os dados e também controla as outras unidades.
- **Unidade de alimentação:** Normalmente uma bateria ou pilha de longa duração. Alimenta todos os componentes conectados ao nó.
- **Unidade transceptora:** Responsável por converter os dados recebidos e processados para rádio-frequência, para que possa deixar a antena e ser transmitida aos

outros nós ou gateway. Essa unidade também faz o inverso, ou seja, recebe dados a ela destinados e os transforma em bits para que a unidade de processamento possa tratá-los (J. P. FONSECA, 2018).

2.3 NRF24L01+

Os módulos Nordic da família nRF24 são transceptores wireless utilizados para receber e transmitir dados por rádio-frequência. Trabalhando na frequência de 2,4GHz ISM (Industrial Scientific and Medic) utilizando 126 canais de 1MHz de largura de banda, os transceptores nRF24 utilizam o protocolo ESB+ para a transmissão de dados, e tem capacidade para transmitir a uma velocidade de até 2Mbps. Como dispositivos Wi-Fi trabalham na maioria dos canais mais baixos dessa banda, para aplicações que utilizam o nRF24, a fim de evitar eventuais interferências, é aconselhável utilizar os 25 canais mais altos do espectro utilizável (H. SAHA, 2017).

Redes nRF24 podem ser configuradas para trabalharem em diversas topologias diferentes. Além de possuir diversos parâmetros configuráveis, como canal, potência e taxa de transmissão, o módulo suporta conexões diretas entre módulos (P2P), assim como estrela ou até mesmo árvore. (SEMICONDUCTOR, 2008) Algumas bibliotecas mais recentes implementam o protocolo Mesh para a arquitetura. Uma das vantagens da rede Mesh é o auto endereçamento e auto roteamento dos pacotes de maneira dinâmica, caso um nodo caia ou seja adicionado a rede, sem a necessidade de reiniciar a rede como um todo. Entre as tecnologias de sensoriamento sem fio, a nRF24 destaca-se pelo seu baixo custo e baixo consumo, sendo ideal para projetos de pequeno porte e construção de protótipos (J. P. FONSECA, 2018).

2.3.1 nRF24I01+ PA/LNA

O módulo transceptor utilizado neste projeto conta com outros dois componentes que auxiliam seu funcionamento e aumentam seu alcance. Em seu nome, a sigla PA significa *Power Amplifier* ou, amplificador de potência, que aumenta a potência do sinal sendo transmitido a partir de um determinado transceptor nRF24I01+. Por outro lado, o LNA, *Low-Noise Amplifier* ou, Amplificador de baixo ruído, serve para amplificar o sinal recebido pela antena de um módulo. Este sinal costuma ser muito baixo, na ordem de microvolts, tornando necessário o uso de um amplificador capaz de amplificar o sinal para níveis mais úteis para o microcontrolador, sem adicionar ruídos que comprometeriam a qualidade da informação recebida (NRF24... , s.d.).

Enquanto o módulo nRF24I01+ apresenta distâncias máximas de comunicação de aproximadamente 100 metros, seu sucessor, o nRF24I01+PA/LNA consegue, trabalhando em potência máxima e taxa de transmissão mínima (250KBPS), comunicar-se adequadamente com outro sensor a uma distância de até 1000 metros, quando considerado um

terreno sem obstáculos. A Figura 2 apresenta o transceptor nRF24L01+PA/LNA.

Figura 2 – nRF24L01+ PA/LNA



Fonte: O Autor

Medições foram realizadas em terrenos com características similares as da costa do Rio Araranguá, que apresenta vegetação ripária e algumas residências, e estimou-se que a comunicação é confiável em até aproximadamente 900 metros entre dois nós. Os testes não foram realizados em clima chuvoso.

2.4 GSM-GPRS

Sistema global para comunicação móvel, ou GSM (*Global System for Mobile Communication*) é uma tecnologia celular utilizada em todo o mundo para a transmissão de voz e dados. Enquanto a tecnologia GSM foi desenvolvida com ênfase em sessões de voz, a tecnologia GPRS foi criada para oferecer acesso a outras redes de dados, como a internet através da integração do protocolo TCP/IP (HALONEN, 2003).

Esta tecnologia foi escolhida para integrar o projeto, sendo responsável pelo envio das informações coletadas pela rede de sensores até a API projetada, através de módulo SIM900. A Figura 3 apresenta o Shield GSM SIM900.

Figura 3 – Shield GSM SIM900



Fonte: O Autor

2.5 SENSORES E PARÂMETROS RELEVANTES

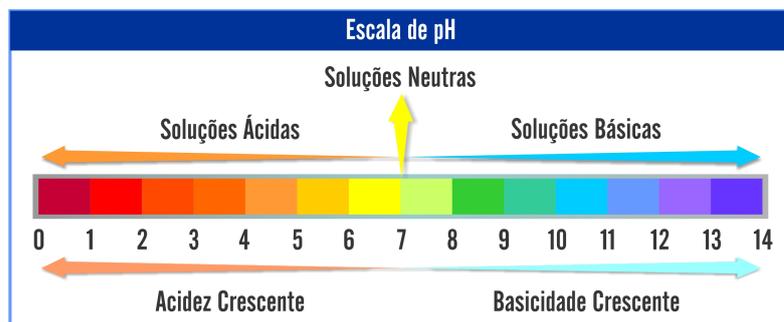
2.5.1 pH

Variando de acordo com a temperatura e composição química de cada substância, o pH ou potencial hidrogênico é uma escala logarítmica que representa a intensidade das condições ácidas ou alcalinas de um meio líquido. A partir da medição da presença de íons de hidrogênio em determinada solução, o valor de pH é classificado em uma escala de 0 - 14. Sendo que, valores inferiores a 7 apontam condições ácidas, superiores a 7 apontam condições alcalinas, e um valor de pH igual a 7 representa uma substância neutra.

Alterações no pH podem ter origem tanto natural, como resultado de dissolução de rochas ou por consequência de processos fotossintéticos, ou pode ser fenômeno consequente de processos de poluição atmosférica ou poluição direta, como despejo de poluentes e substâncias químicas em corpos de água.(FUNASA, 2014)

Considerando, a manutenção da vida aquática, o índice pH deve situar-se na faixa de 6 a 9. Considerando, águas destinadas ao abastecimento, o valor deve estar entre 6.5 a 9.5. Para águas cujo destino é a irrigação de plantações, o pH geralmente deve permanecer nestes mesmos intervalos, atentando para necessidades específicas de cada tipo de cultura (FUNASA, 2014). A Figura 4 apresenta a escala de pH.

Figura 4 – Escala pH



Fonte: <https://www.blog.mcientifica.com.br/wp-content/uploads/2013/10/escala-de-ph-01.jpg> [accessed 28 Oct, 2020]

2.5.2 Sensor de pH - SEN0161

Devido ao baixo custo, facilidade de uso e aferições precisas, o sensor de pH *Gravity:SEN0161* foi escolhido para integrar este protótipo. Com amplo alcance de leitura, aferindo valores de 0 a 14 na escala de pH com erro de aproximadamente ± 0.1 , o sensor apresenta baixo tempo de resposta e fácil integração com o microcontrolador Arduino UNO, utilizado na construção dos nós da rede.(GRAVITY:..., s.d.)

Por outro lado, este modelo é considerado um sensor de laboratório, não podendo ficar imerso em água por longos períodos de tempo. Como a imersão permanente do sensor em água é um dos requisitos do projeto *SisMoRA*, sua substituição será necessária. No capítulo "Trabalhos futuros", segue como sugestão sua substituição pelo modelo industrial, que pode ficar submerso por períodos de até 1 ano antes de receber manutenção.(GRAVITY:..., s.d.)

A Figura 5 apresenta o sensor SEN0161.

Figura 5 – Sensor SEN0161



Fonte: O Autor

2.5.3 TDS - Total de sólidos dissolvidos

O índice de TDS, medido em ppm (partes por milhão), representa a quantidade total de substâncias sólidas que estão dissolvidas em um líquido. Estas substâncias podem incluir tanto componentes orgânicos como inorgânicos, como sais minerais, metais, cálcio, etc. Tais partículas são caracterizadas por apresentarem diâmetro inferior a $10^{-3}\mu m$ e permanecerem em solução mesmo após processo de filtragem. Assim como o índice de pH, o índice de TDS é influenciado por processos naturais assim como por processos antropogênicos (lixo, esgoto e poluentes). (FUNASA, 2014)

Neste projeto, conforme discutido posteriormente, o sensor de TDS foi escolhido para substituir o sensor de condutividade. Por ser financeiramente mais viável, o sensor de TDS foi escolhido e demonstrou-se suficiente para gerar estimativas em relação ao estado físico-químico da água a ser analisada. Análises mais fidedignas aos índices reais da água do Rio Araranguá poderão ser obtidas ao realizar a substituição do sensor de TDS por um condutivímetro. Tal adaptação segue como sugestão para trabalhos futuros que tenham este como base.

2.5.4 Sensor de TDS - KS0429

Para fazer a aferição da quantidade de substâncias orgânicas e inorgânicas diluídas em água, foi escolhido para integrar o protótipo o medidor de TDS *KS0429*. Por ser um sensor barato, de fácil interface com microcontroladores da família do Arduino, este sensor substituiu um condutivímetro durante esta etapa de prototipação.

Embora não seja possível correlacionar de forma direta valores de condutividade e TDS, especialmente para águas naturais, que são soluções complexas, tal correlação é

possível e mais fiel para águas nas quais a predominância de um determinado íon é bem definida (FUNASA, 2014).

Apesar da água do rio ser uma solução complexa, uma conversão para medida de condutividade foi estimada. Para poder classificar o índice de TDS medido, a seguinte fórmula foi utilizada (1 mg/L = 1 ppm):

$$TDS(mg/L) = K * [EC(dS/m)]$$

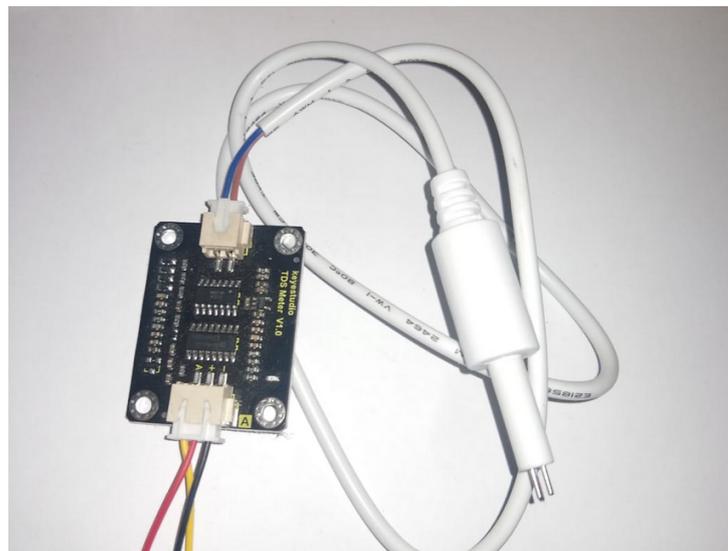
$$K = 640$$

K é uma constante de conversão utilizada quando considerada a análise de águas naturais (A. F. RUSYDI, 2018). Contudo, perde-se precisão no processo de conversão.

Outros trabalhos como (ALI, N.S., MO, K. KIM, M., 2012), que estuda equivalências de valores de TDS e condutividade em águas poluídas por atividades industriais, estima valores similares para essa constante de conversão. Iyasele (IYASELE, J.U, 2015) estima que os valores de conversão variam de 550 a 800, dependendo da solução analisada.

A Figura 6 apresenta o sensor KS0429.

Figura 6 – Sensor KS0429



Fonte: O Autor

2.6 COMPUTAÇÃO EM NUVEM

Segundo Peter Mell (P. MELL, 2011), o modelo de computação em nuvem tem por objetivo criar um ambiente conveniente, de fácil acesso para um conjunto de recursos computacionais, como redes, servidores, bancos de dados e aplicações para que estes possam ser rapidamente configurados e utilizados com esforço e manutenção mínima. Este modelo é composto de cinco características essenciais e cinco modelos de serviço.

Entre as características essenciais estão:

- **Serviço *on-demand***

O consumidor desse serviço pode requisitar e utilizar recursos computacionais de maneira automática, fazendo com que o serviço escale sem interação humana.

- **Alta disponibilidade**

Os recursos podem ser acessados de diversos lugares e dispositivos e deverão estar disponíveis a todo tempo.

- **Agrupamento de recursos**

Os recursos computacionais são agrupados e podem servir diversos clientes ao mesmo tempo. Os recursos físicos e virtuais são delegados e re-atribuídos de acordo com a demanda de cada cliente.

- **Rápida escalabilidade de serviço**

Recursos e serviços deverão escalar rapidamente para suprir as demandas dos clientes. Essa transição deve ser invisível, e o cliente deve ter acesso instantâneo e aparentemente ilimitado.

- **Métricas de serviço**

Os sistemas em nuvem deverão controlar e otimizar recursos, além de manter métricas a respeito do uso e clientes. Essas informações trazem transparência para o cliente e para o provedor do serviço.

Entre os modelos de serviço estão:

- **SaaS - Software como serviço**

O provedor permite que o cliente utilize aplicações sendo executadas no ambiente da nuvem. Essas aplicações são acessadas através de diversos clientes, utilizando uma interface cliente como um navegador ou uma outra aplicação.

- **PaaS - Plataforma como serviço**

O provedor permite que o cliente desenvolva aplicações na nuvem utilizando linguagens de programação, bibliotecas, serviços e ferramentas suportadas pelo provedor.

- **IaaS - Infraestrutura como serviço**

O cliente tem a capacidade de preparar ambientes completos de processamento, armazenamento e redes nos quais poderá executar quaisquer softwares pela infraestrutura suportados.

Outros modelos de serviço, mais recentemente definidos são:

- **DaaS - Banco de dados como serviço**

Nesse modelo o provedor permite que o cliente configure acesse e utilize uma instância de banco de dados, servido na nuvem.

- **FaaS - Função como serviço**

Parecido com o modelo PaaS, o modelo de função como serviço dá vida aos serviços *serverless*. Estes serviços, ou funções, executam sob-demanda, sendo instanciadas em um servidor que melhor pode atender à requisição do cliente no momento que esta foi feita. Após executar, essa função é removida do ambiente de execução, sendo instanciada exclusivamente quando um cliente a requisitar. Como consequência, serviços FaaS não possuem estado, fazendo com que quaisquer dados importantes sejam mantidos em um banco de dados ou serviço externo à função.

Neste trabalho alguns desses modelos de serviço foram utilizados. Enquanto a API é *serverless*, ou seja, foi construída a partir do modelo FaaS, o banco de dados se comporta como um DaaS. Esses serviços são providos pela plataforma Firebase, um serviço do tipo PaaS.

2.6.1 Google Firebase

O Google Firebase é uma plataforma web, voltada para o desenvolvimento de aplicações, que provém diversos serviços baseados em nuvem para o desenvolvedor. O serviço de banco de dados do Firebase, chamado Firestore, é baseado no paradigma NoSQL, utilizando objetos JSON para armazenar e administrar os dados.

Além de banco de dados, o Firebase disponibiliza outros serviços úteis para o desenvolvedor da aplicação (MORONEY, 2017), entre eles:

- **Firestore Analytics:** Busca e relaciona indicadores da utilização do Firestore. Permite ao administrador entender como os clientes estão utilizando seu serviço.
- **Firestore Cloud Messaging:** Uma plataforma que provém soluções de notificações em tempo real para usuários da aplicação. Suporta aplicações Android, iOS e web.
- **Firestore Auth:** Serviço de autenticação. Funciona através de middlewares de autenticação como login com Facebook ou Twitter mas também pode funcionar com e-mail e senha cadastrados no próprio Firestore.
- **Firestore Storage:** Armazena e facilita a troca de arquivos na plataforma. Suporta imagens, vídeos, áudio ou qualquer outro arquivo gerado pelos usuários.

(FIREBASE. . . , s.d.)

O Firestore possui três tipos de planos, um deles é gratuito, para prototipação e testes e dois pagos. O plano gratuito oferece autenticação de usuários, notificações ilimitadas e capacidade de armazenamento de 1 Gigabyte. A banda de dados é de no máximo 10 Gigabytes por mês. Um dos planos pagos opera a partir de um valor fixado de 25 dólares, e são oferecidos 10 Gigabytes de armazenamento, e 50 Gigabytes de banda

de dados, além dos serviços gratuitos já citados. O último plano pago, chamado de Blaze Plan, é pago de acordo com a quantidade de dados armazenados e banda utilizada, com preços de de 0,026 centavos de dólar por Gigabyte utilizado para armazenamento e 15 centavos por Gigabyte de dados transferidos. Considerando projetos em que as medições são esparsas e a quantidade de dados é mínima, como é o caso deste projeto, o plano Blaze é uma boa opção, uma vez que dá acesso a todos os serviços ofertados a um preço acessível. O plano gratuito pode ser utilizado para testes de rede e prototipação, mas considerando que o serviço deve funcionar por bastante tempo, e que os dados antigos deverão ser mantidos, para possibilitar análises e predições, o limite de 1 GigaByte de armazenamento é insuficiente.

2.7 PLATAFORMA ANDROID

Android é o nome de uma plataforma para dispositivos móveis que inclui o sistema operacional, *middlewares* e aplicações base. Lançado em setembro de 2008, a plataforma Android cresceu rapidamente no mercado de sistemas para celulares, e hoje representa aproximadamente 91,6% de todos os *smartphones* vendidos no Brasil(ANDROID. . . , s.d.).

Baseada no sistema operacional *Linux*, a plataforma Android pode ser dividida em quatro subgrupos de organização de software. O software de mais baixo nível é o Kernel do sistema, composto em sua maioria por *drivers*. Estes *drivers* são softwares responsáveis pela comunicação com outros módulos do sistema, como câmera, tela, USB, audio e alimentação. A próxima camada de software é composta por bibliotecas necessárias para o funcionamento correto do sistema, como bibliotecas de rede e banco de dados. A camada de *framework* de aplicação vem logo acima, trazendo componentes macro do sistema, responsáveis pela integração e manejo de recursos. Por último a camada de aplicação, que representa aplicativos instaláveis no smartphone. Estes podem apresentar diversas funções, desde aplicações simples como uma calculadora até aplicações mais complexas como jogos 3D(NISARG GANDHEWAR, s.d.).

2.7.1 React Native

React Native é um *framework* javascript que permite o desenvolvimento de aplicações, renderizadas nativamente, para plataformas Android e iOS. Diferente de outros *frameworks*, como o Cordova, que funcionam em WebView, o React Native permite que o desenvolvedor utilize recursos nativos da plataforma alvo, garantindo à aplicação maior performance (quando comparado a soluções que utilizam WebView) e facilidade de desenvolvimento por parte do programador(AXELSSON *et al.*, 2016).

Esse framework foi utilizado para o desenvolvimento da aplicação móvel, componente deste trabalho.

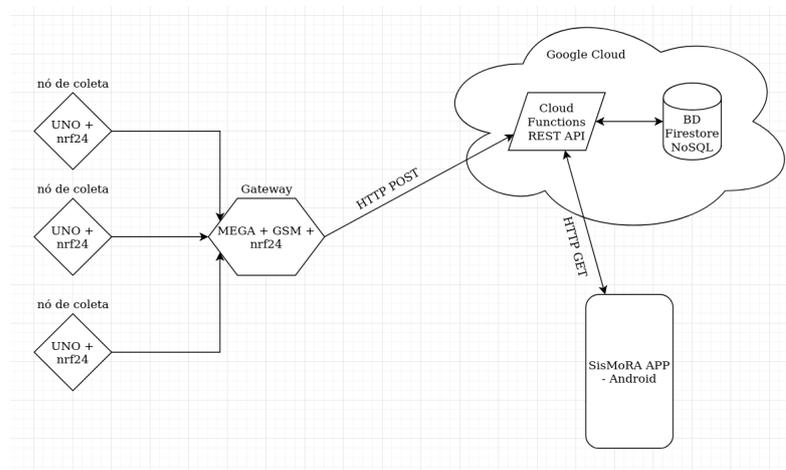
3 MODELAGEM DO SISTEMA

O modelo proposto possui três partes principais: Uma rede de sensores elaborada com transceptores nRF24L01+ e microcontroladores Arduino UNO e Arduino MEGA, responsável pela aquisição e transmissão dos dados coletados. Por segundo, para receber e servir os dados coletados pela rede sensorial, uma API HTTP juntamente a um banco de dados não relacional hospedados nos serviços do Google Firebase foi projetado. Por fim, uma aplicação, construída a partir do framework *React Native*, para dispositivos móveis Android, que serve como interface de acesso para o usuário final.

O caminho da informação tem origem na rede de sensores, que coleta dados de temperatura, pH e TDS (Total de sólidos dissolvidos) da água do Rio Araranguá a partir de sensores específicos para leitura desses dados, e os envia a um nó coordenador da rede, que opera a partir de um módulo GSM, chamado aqui de *gateway*. Após receber os dados dos diversos nós da rede, o *gateway* realiza o pré-processamento, que consiste na validação dos dados e classificação da água amostrada em três categorias, e envio das informações para o banco de dados por intermédio da API, o servidor do serviço, através de uma requisição HTTP POST. Uma vez verificada a integridade dos dados recebidos, a API salva tais informações no banco de dados não-relacional *Firestore*, para que estes sejam acessíveis pela aplicação móvel. Por último, um usuário do serviço pode acessar tais informações através da aplicação móvel desenvolvida para dispositivos Android, e acompanhar os índices físico-químicos da água amostrada. Nesta aplicação, além de acessar dados da leitura mais recente realizada pela rede de sensores, o usuário pode também ter acesso ao histórico de leituras e visualização de gráficos, para melhor acompanhar e compreender as flutuações dos índices apresentados.

A Figura 7 apresenta uma visão geral dos componentes da plataforma.

Figura 7 – Visão macro dos componentes da plataforma



3.1 REDE DE SENSORES SEM FIO

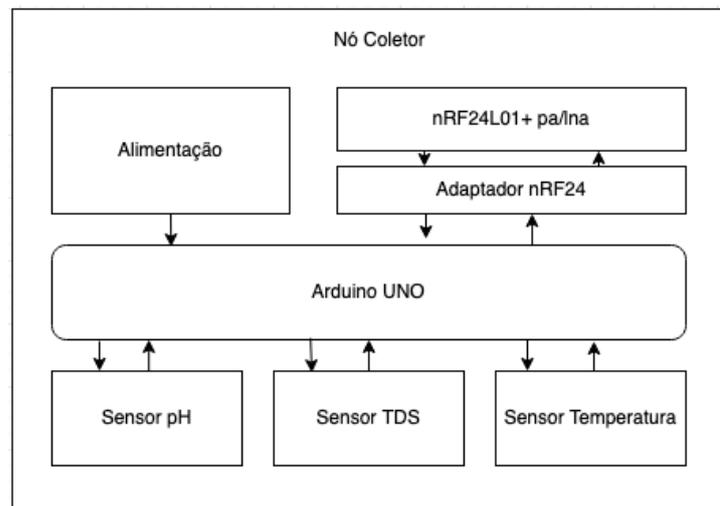
Para estruturar a rede de sensores, além dos sensores específicos medidores, foram utilizados Arduinos, plataforma de prototipagem eletrônica, juntamente a transceptores nRF24L01+ PA/LNA.

A rede possui três tipos de nós, cada um construído de uma maneira e com funcionalidades distintas. Estes são:

- **Nós Coletores:**

São os nós que realizam as aferições dos índices da água do rio. Durante a fase de testes, não foi desenvolvido nenhum tipo de revestimento para os nós. Porém, estes nós estarão em contato com a água e por isso necessitarão de um revestimento a prova d'água. Além disso, por alimentarem também os sensores de temperatura, pH e TDS, necessitarão de um sistema de alimentação mais robusto. A configuração ótima de alimentação dos nós não foi coberta nesse trabalho. A Figura 8 apresenta os componentes dos nós Coletores.

Figura 8 – Componentes nó Coletor

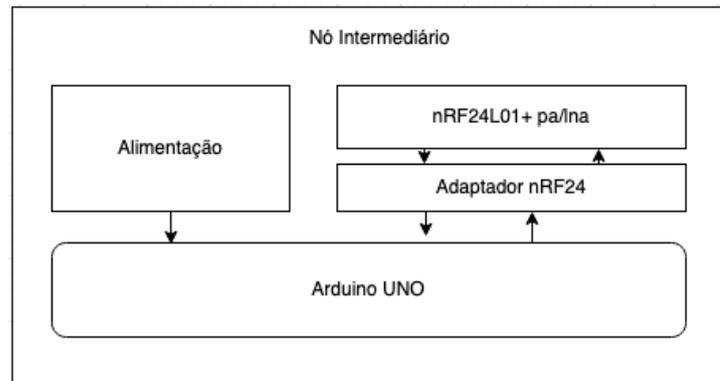


Fonte: O Autor

- **Nós Intermediários:**

Sua função é apenas servir de ponte de comunicação entre dois nós que estejam fisicamente muito distantes para se comunicarem diretamente. São os nós que ficam entre o nó Master e os nós Coletores, ou entre dois nós Coletores. Estes nós não precisaram ficar em contato com a água. A Figura 9 apresenta os componentes dos nós Intermediários.

Figura 9 – Componentes nó Intermediário

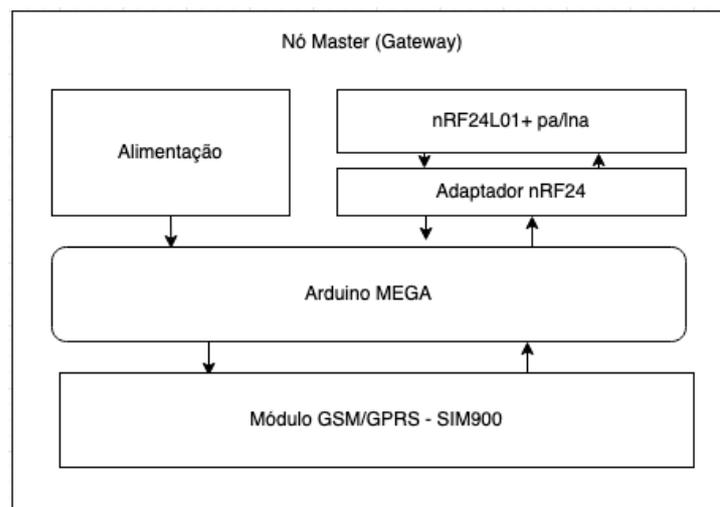


Fonte: O Autor

- **Nó Master (*gateway*):**

Este é o nó responsável por coordenar a rede sensorial. O nó Master têm a responsabilidade de receber e pré-processar os dados lidos pelos nós coletores, além de encaminhar mensagens de configuração aos nós da rede. Este nó é construído a partir de um Arduino MEGA um módulo nRF24L01+ e um shield GSM-GPRS SIM900. Após coletar os dados vindos da rede de sensores nRF24, o nó Master os envia à API através de uma requisição HTTP POST. É esperado que esse nó opere a partir de alimentação contínua, e não de bateria. A Figura 10 apresenta os componentes do nó Master.

Figura 10 – Componentes nó Master (Gateway)



Fonte: O Autor

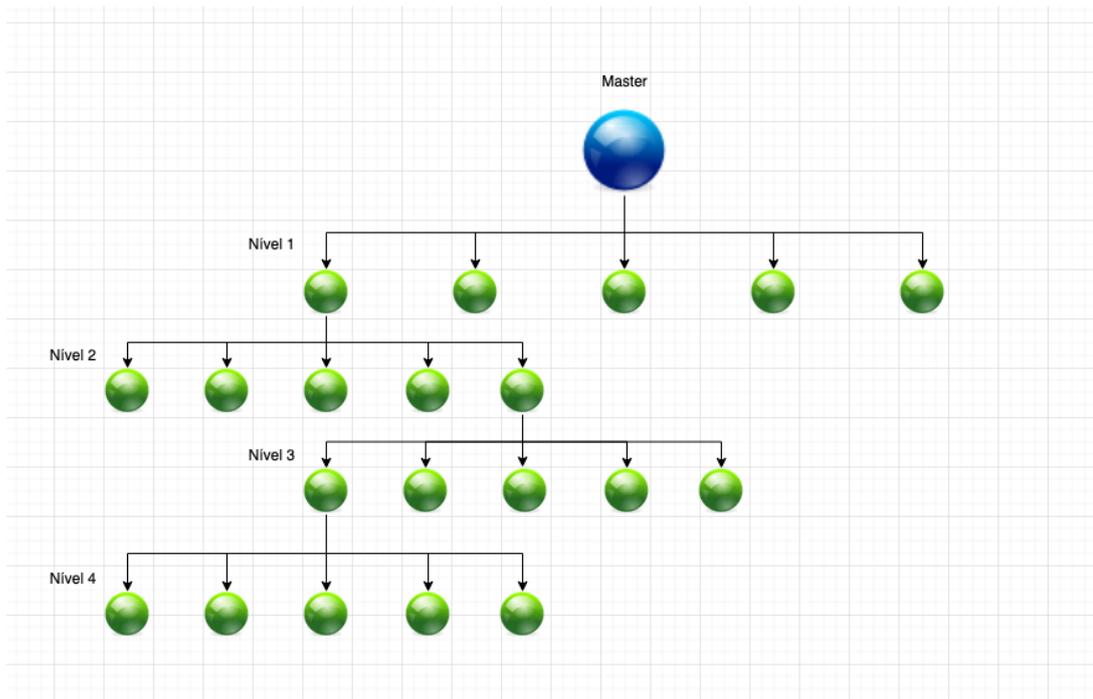
O Anexo A descreve o processo de montagem e componentes utilizados em cada tipo de nó.

No que diz respeito a topologia de rede, os transceptores da família nRF24 têm a capacidade de operar organizados de diversas maneiras. Fazendo uso de bibliotecas previamente desenvolvidas por terceiros, uma rede de transceptores nRF24 pode operar em topologias ponto-a-ponto, árvore, estrela e até mesmo *mesh*.

O modelo utilizado para esse protótipo, por demonstrar alta configurabilidade, foi a topologia do tipo árvore. A rede tem capacidade de endereçar até quatro níveis a partir do nó *Master*, e cada nó pode ter até cinco nós filhos. Considerando a Figura 11, que representa a topologia de árvore aplicada a este projeto, e considerando a distância máxima entre dois nós adjacentes sendo aproximadamente 800 metros, estima-se que uma rede com essas características poderá cobrir até 6,400 metros em linha reta. Este número foi estimado a partir das características observadas durante a fase de testes, e em nenhum momento essa distância foi de fato calculada em operação.

A Figura 11 apresenta a topologia do tipo Árvore.

Figura 11 – Topologia da rede - Árvore



Fonte: O Autor

Levando em consideração a natureza de operação remota do sistema e a localização isolada onde a rede operará, o meio de comunicação escolhido para que o envio de informações coletadas pela rede de sensores seja feito até a API projetada foi o GSM-GPRS, uma vez que redes celulares apresentam boa cobertura na região onde as medições serão

realizadas. Enquanto internamente a rede opera a partir dos transceptores da família nRF24, a comunicação do *gateway* até o serviço em nuvem é feito via rede celular.

Esta escolha possibilitou também a configuração remota da rede a partir de mensagens SMS. Neste momento, o *gateway* aceita apenas configurações em relação à frequência de leitura de cada nó, que funciona da seguinte maneira: A partir de um número de celular previamente configurado no *gateway*, deve ser enviado o seguinte código: *CAI: H,M,S,X*, onde "CAI" significa "Configure Alarm Interval" ou "Configure o intervalo do alarme", "H" indica horas, "M" indica minutos e "S" indica segundos. O parâmetro "X" pode tomar valores "0" ou "1". Quando $X = 0$, a nova configuração será enviada após a próxima leva de leituras, para garantir que todos os sensores estarão ativos e aptos a receber a mensagem. Caso $X = 1$, a nova configuração será enviada imediatamente a todos os nós, opção utilizada para a primeira configuração da rede, ou para forçar uma nova configuração em qualquer momento. Porém, nesse caso, receberão a nova atualização apenas os nós que estiverem ativos no momento do envio.

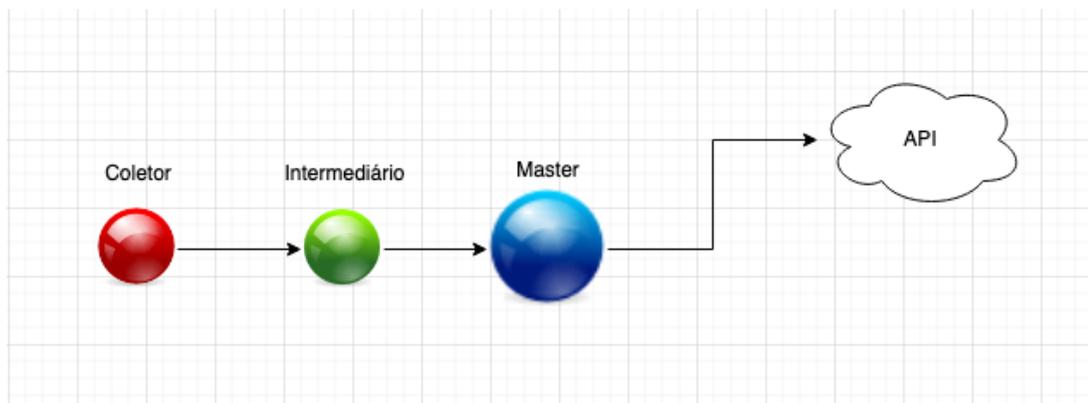
Por exemplo, uma mensagem de configuração do tipo: "CAI: 0,0,20,1" indica que a leitura deverá ser feita a cada vinte segundos, e o nó *gateway* deverá enviar essa configuração aos demais nós imediatamente. Caso a inclusão de módulos *Real-Time Clock* (RTC) seja feita à rede, esse modelo de configuração poderá ser substituído por datas e horários fixos, informados ao nó Master através de mensagem SMS.

A comunicação interna da rede ocorre majoritariamente no sentido *Nó Coletor -> Master*. Após efetuar uma leitura, o nó coletor envia um objeto contendo os índices lidos, informações sobre bateria e um campo para reportar erros até o nó Master.

Nesse modelo, o roteamento interno de pacotes é feito automaticamente a partir da numeração de cada nó, não sendo necessário que a informação siga uma rota manualmente definida pelo programador. No modelo atual os nós estão constantemente em modo de escuta, esperando eventuais configurações vindas do nó master, ou esperando informações que por ele sejam roteadas. No momento de envio de informações o nó deixa de escutar outros nós da rede, fazendo com que eventuais pacotes a ele endereçados sejam perdidos. Deste modo a rede funciona como um sistema *Half-Duplex*, ou seja, a comunicação acontece em ambas direções (*Coletor -> Master* ou *Master -> Coletor*), mas apenas uma por vez. Para mitigar esse problema, a biblioteca utilizada possui um sistema de confirmação de entrega de pacotes que faz o reenvio automático de um pacote caso este tenha sido perdido.

Um esquema da rede final utilizada nesse protótipo pode ser visualizada abaixo, na Figura 12.

Figura 12 – Topologia da rede aplicada no protótipo



Fonte: O Autor

Este modelo representa uma configuração mínima da rede e deverá ser extrapolado para cobrir uma porção mais significativa do rio.

3.2 DESENVOLVIMENTO DA API

Esta API foi projetada utilizando a plataforma Google Firebase. Além de dar suporte a serviços *serverless*, o Firebase também disponibiliza serviços de bancos de dados não relacionais, serviços de autenticação, hospedagem e modelos de *Machine Learning* para auxiliar o processo de desenvolvimento de aplicativos Web ou móveis. Uma das características de aplicações que seguem a arquitetura *serverless* é o funcionamento sob demanda, fazendo com que o serviço escale automaticamente com base na frequência de requisições (P. MELL, 2011).

Assim como seu funcionamento, sua cobrança também é feita sob demanda, fazendo com que o responsável pela aplicação pague um valor proporcional à quantidade de requisições feitas ao serviço. O valor de cada requisição é porém muito pequeno. Considerando a cota gratuita de duas milhões de requisições mensais, estima-se que o responsável pela API seja cobrado apenas após a marca de 6500 usuários mensais, considerando que cada usuário realize até 10 requisições diariamente.

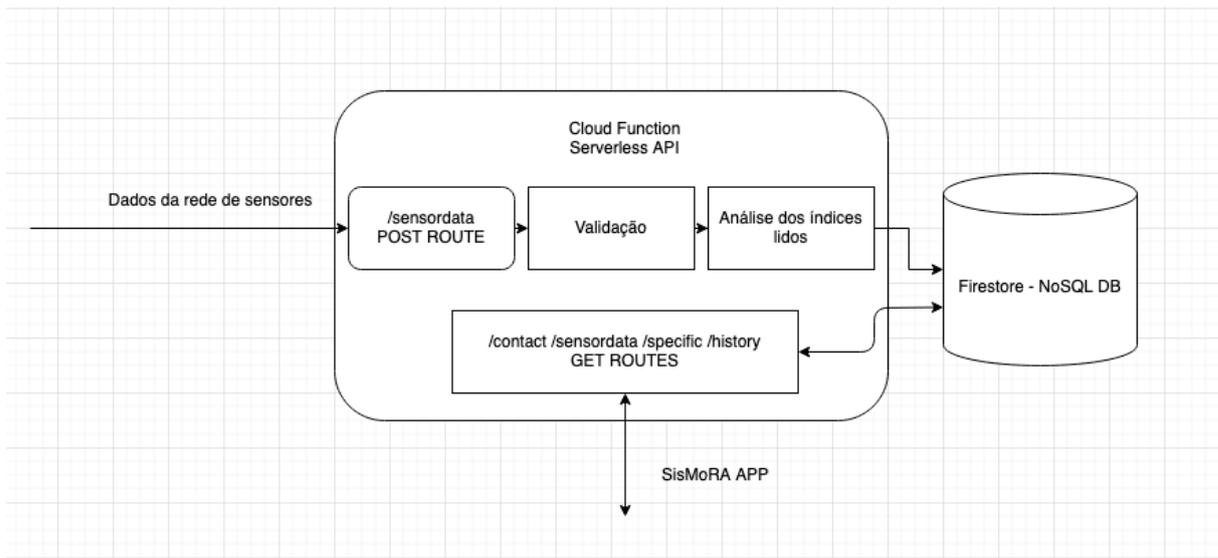
Seguindo o modelo *serverless*, foi elaborado uma API para servir as funcionalidades da plataforma projetada. Utilizando o protocolo HTTP e utilizando-se de "rotas" ou caminhos de comunicação, a API recebe dados através de métodos POST e serve dados através de métodos GET. Considerando o armazenamento dos dados, o banco de dados escolhido foi o Firestore, banco de dados não-relacional (NoSQL), também parte do ecossistema da plataforma Firebase.

Além da rota receber dados da rede de sensores, outras rotas foram projetadas para auxiliar o administrador do sistema. Entre estas estão a rota para atualizar informações de contato e de posicionamento geográfico dos sensores. Estas são acessíveis apenas através de

requisições diretas, não podendo ser acessadas pela aplicação ou pela rede de sensores. No futuro sugere-se a criação de uma página de administração para mais facilmente manipular estes dados.

A Figura 13 ilustra a interação dos módulos da API que serão discutidos nas próximas seções.

Figura 13 – Caminho de dados na API



Fonte: O Autor

3.2.1 Rota de entrada - */sensordata*

Destacada na Figura 13, a rota */sensordata* é a porta de entrada para dados vindos da rede de sensores. Sua função é apenas receber os dados que foram lidos pela rede de sensores e enviados até a API via GSM utilizando uma requisição do tipo HTTP POST. Entre os dados lidos pela rede e posteriormente enviados à API através da rota */sensordata* estão:

- **TDS** - Valor de TDS lido em ppm (partes por milhão)
- **sensor** - Número identificador do sensor em base 8
- **battery** - Tensão da bateria ligada ao nó em Volts
- **error** - *Flag* para indicar erros no sensor
- **PH** - Valor de pH lido pelo sensor
- **EC** - Campo para enviar valores de condutividade (Não utilizado enquanto a troca do sensor de TDS por um condutímetro não for realizada).

- **bomba** - Identificador da bomba de água correspondente ao sensor. É esperado que cada sensor coletor seja posicionado próximo à uma bomba d'água.

* Os dados serão aceitos pela API apenas se estiver no formato JSON.

3.2.2 Validação dos dados

Após receber os dados na rota `/sensordata`, a API realiza uma etapa de validação. Esta etapa é muito simples e serve apenas para verificar que todos os campos que deveriam estar presentes em uma leitura de fato estão presentes na requisição iniciada pela rede de sensores. Requisições que contenham campos desconhecidos ou não que não contenham os campos listados na seção 3.2.1 serão ignoradas. Vale pontuar que todos os campos deverão estar formatados como *string*.

3.2.3 Análise e classificação

Depois de verificada a integridade dos dados recebidos, a API realiza a classificação da água baseando-se nos parâmetros de pH e TDS de cada leitura. Destaca-se que para classificar o estado da água entre "BOM", "CUIDADO" ou "RUIM" estes parâmetros foram classificados conforme a literatura científica. A classificação dá-se a partir da análise do índice de TDS, convertido em métrica de condutividade, e do índice de pH. Apesar da perda de precisão no processo de conversão, para a validação do modelo apresentado, esta conversão demonstrou-se adequada para os testes da plataforma. No entanto, é recomendado a troca do sensor de TDS por uma célula de condutividade para a precisão dos dados de salinidade no projeto.

Segundo (FRAGA *et al.*, 2010), o crescimento da planta de arroz começa a ser afetado pelo nível de salinidade da água quando esta apresenta valores de condutividade de aproximadamente $2dS/m$. Níveis de pH devem manter-se próximos a 6, segundo (J. M.B. PARFITT, 2016).

3.2.4 Envio ao banco de dados

Após o tratamento dos dados recebidos pela rede de sensores, a API os envia para um banco de dados não relacional, também servido na nuvem. Para comunicar-se com o banco de dados de maneira segura, uma chave de acesso foi gerada nos serviços do Google Firebase, fazendo com que a API esteja apta a guardar ou ler informações do banco de dados. A comunicação entre API e banco de dados é abstraída pelo *Software Development Kit* (SDK) `firebase-admin`, disponibilizada pela Google para linguagem NodeJs.

3.2.5 Rotas de acesso aos dados

Entre as rotas que leem informações do banco de dados, através de requisições do tipo GET, estão: */sensordata*, que colhe os dados da última leitura de cada sensor, */specific*, que busca os dados dado uma data específica, */history*, que devolve dados de leituras antigas a partir de uma data específica e */contact*, que retorna dados relacionados ao contato com o administrador da plataforma. Todas essas rotas, porém, não devem ser acessadas diretamente, uma vez que a aplicação já está preparada para utilizá-las conforme demanda do usuário.

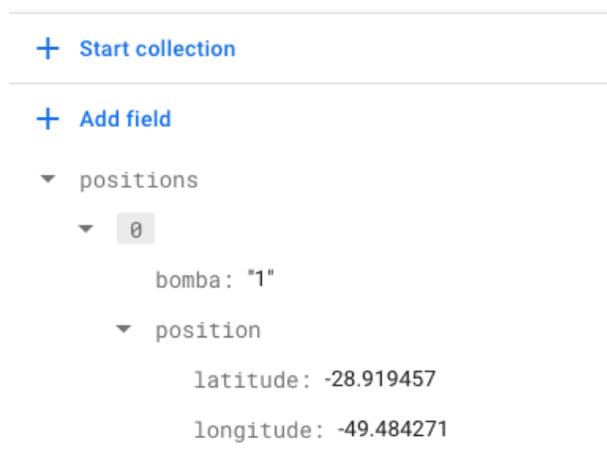
É válido notar que em nenhum momento o banco de dados é manipulado diretamente, seja pela rede de sensores ou pela aplicação. Esta configuração ajuda a manter a integridade dos dados em questão.

3.2.6 Configuração inicial do Banco de Dados

Antes de executar a plataforma pela primeira vez, três coleções deverão ser criadas no banco de dados manualmente através do console do Firebase. Coleções são análogas à tabelas em bancos de dados SQL, com a diferença que podem guardar objetos dinâmicos, não atrelados a um *schema* pré-definido.

Uma coleção chamada *contactInfo* armazenará um documento com informações de contato com o responsável pela plataforma. Dentro desta, um documento será criado. Este documento deverá conter os seguintes campos: *email* (e-mail para contato), *name* (nome do responsável) e *whatsapp* (número de telefone do responsável).

A coleção *sensorPosition* deverá conter um documento chamado *uniquePosition-Group* que por sua vez guarda dados de longitude e latitude do posicionamento de cada sensor. Dessa maneira, a cada leitura, o aplicativo poderá relacionar uma posição geográfica no mapa a um sensor específico. A Figura 14 exemplifica uma possível configuração para este documento.

Figura 14 – Exemplo de configuração do documento *uniquePositionGroup*

Fonte: O Autor

Por último, uma coleção chamada *meshReads* deverá ser criada. Esta coleção não precisa ser inicializada com nenhum documento, uma vez que documentos serão adicionados conforme a rede sensorial reportar novas leituras. Para cada novo grupo de leitura um novo documento é criado.

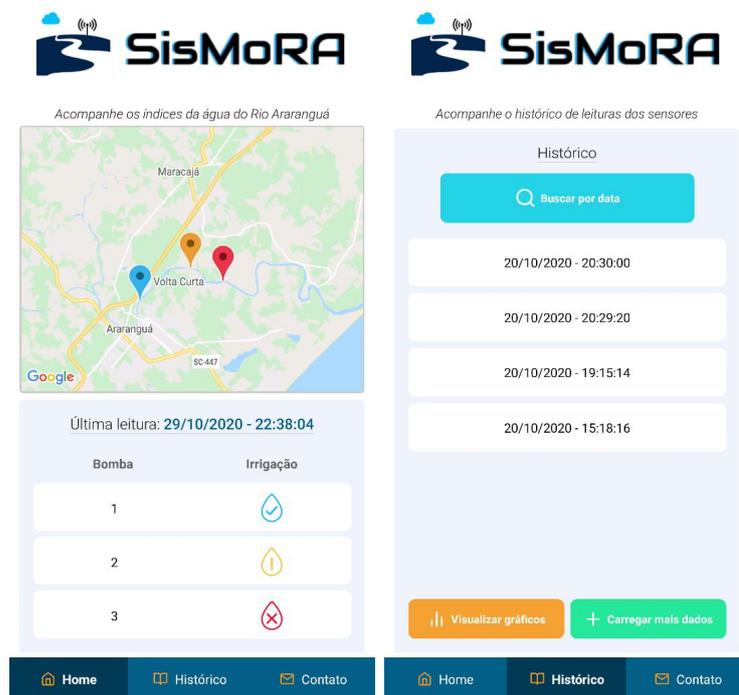
Essas coleções poderão ser adaptadas, tanto em forma quanto em nome, desde que o código da API também seja atualizado para manter a integridade da comunicação entre banco de dados e API.

3.3 DESENVOLVIMENTO DO APLICATIVO

Para a construção do aplicativo *mobile*, interface com o usuário final, foi escolhido o *framework* Javascript React Native, por conta de sua extensa documentação e facilidade de uso, além de prover o desenvolvedor com inúmeras ferramentas que aceleram o processo de elaboração e construção de aplicativos.

Através de requisições HTTP GET em rotas predefinidas na API, backend do projeto, a aplicação tem a capacidade de coletar dados da última leitura feita pela rede, dados das medições feitas por cada sensor, histórico das leituras e até mesmo histórico individual de cada sensor. Cada tipo de dado coletado foi tratado e distribuído em diversas telas do sistema, para que ao mesmo tempo informações mais relevantes tenham mais destaque, mas que outras informações, como histórico e gráficos, também estejam disponíveis e organizadas. Na Figura 15 é possível visualizar as telas projetadas para o sistema e uma breve descrição das funcionalidades das mesmas.

Figura 15 – Tela inicial - Tela de histórico



Fonte: O Autor

Na tela inicial, primeira tela apresentada ao usuário quando o aplicativo é iniciado, estão apresentados os dados referentes à última leitura efetuada pela rede de sensores. Um mapa é exibido com a geolocalização de cada sensor/bomba. Essa localização está salva no banco de dados e não é coletada periodicamente pois não há nenhum tipo de geolocalizador nos sensores.

A qualidade da água, considerando seu uso para irrigação, é representada de forma visual. Um marcador azul significa que a água coletada por um determinado sensor está dentro das condições necessárias para ser utilizada para irrigação. Um marcador amarelo representa características físico-químicas no limiar entre aceitável e imprópria. Por fim, a cor vermelha representa índices fora dos padrões considerados necessários para utilização da água para irrigação de maneira segura. Ao clicar em cada bomba são apresentados os valores dos índices lidos. Também é possível acessar a classificação clicando em cada marcador no mapa.

Na segunda tela, Figura 15, é apresentado um histórico das leituras realizadas pela rede sensorial. Quando o usuário busca pelas leituras feitas em um dia específico, são mostradas em tela, inicialmente, até quatro leituras feitas neste dia. Caso o usuário deseje obter mais dados desse dia ou dados mais antigos poderá utilizar o botão "Carregar mais dados". Essa funcionalidade foi implementada para evitar que dados não relevantes sejam buscados, uma vez que como apresentado na seção sobre a API, a cobrança do serviço é

calculada a partir do número de requisições feitas. Após carregar os dados de interesse, é possível utilizar o botão "Visualizar gráficos" para visualizar a flutuação dos índices lidos por cada sensor. Vale lembrar que integrarão os gráficos apenas dados previamente carregados na página de histórico.

Figura 16 – Tela de histórico por bomba/sensor - Tela de gráficos



Fonte: O Autor

Na primeira tela, Figura 16, é apresentada uma lista dos sensores que integram os dados carregados na página de histórico. Selecionando algum dos sensores da lista, são apresentados ao usuário os gráficos referentes as leituras passadas, Figura 16 segunda tela. Os gráficos apresentados têm dimensões dinâmicas, ou seja, os dados são distribuídos de maneira que o valor mais alto lido esteja sempre no topo e o valor mais baixo representa o ponto zero do eixo vertical. Caso muitos dados sejam carregados, o gráfico permite ao usuário rolar horizontalmente para visualização completa do relatório. Dados mais recentes são incluídos da esquerda para a direita.

No aplicativo foi definido também um sistema de envio de e-mail e mensagens pelo aplicativo *Whatsapp* para o responsável pelo aplicativo.

4 TESTES E VALIDAÇÃO DO MODELO

Considerando a natureza modular do sistema, num primeiro momento, as três principais partes da plataforma foram testadas separadamente. Após testar e validar o funcionamento dos componentes individualmente, foram realizados testes integradores em campo no intuito de examinar o funcionamento da plataforma como um todo.

Durante os testes verificou-se que aplicação e API tem construção robusta, apresentando poucos erros ou comportamentos inesperados. A rede sensorial, por outro lado, demonstrou-se mais sensível a diferentes configurações e perturbações externas. Entre os principais fatores que dificultaram a operação da rede sensorial estão interferências eletromagnéticas, causadas por outros dispositivos operando fisicamente próximos ou nos mesmos canais que os transceptores da rede, *jumpers*, ou fios conectores, de baixa qualidade e soldas precárias. Estes fatores tornaram difícil a identificação da fonte dos problemas na rede, pois muitas vezes os nós falhavam sem motivo aparente.

Entretanto tais dificuldades eram esperadas, levando em consideração que durante o processo de prototipagem o acesso à laboratórios, equipamentos e materiais de qualidade foi limitado, devido principalmente à pandemia do COVID-19 de 2020.

4.1 TESTE DA REDE SENSORIAL

Para testar o funcionamento da rede de sensores, num primeiro momento, foram construídos dois nós, compostos de um Arduino UNO e um transceptor nRF24. Com configuração similar ao nó intermediário, descrito no capítulo 3.1, os nós foram utilizados para testar o funcionamento mais básico da rede, a comunicação ponto-a-ponto. Sem muitos problemas nesse modelo de funcionamento, testes de distâncias máximas de comunicação foram realizados. Considerando perda máxima de pacotes de 5%, as distâncias medidas são apresentadas na Tabela 2.

Tabela 2 – Testes de distância.

Potência	dBm	Distância (m)
RF24_PA_MIN	-18	~5
RF24_PA_LOW	-12	~20
RF24_PA_HIGH	-6	~250
RF24_PA_MAX	0	~900

Em seguida, testes envolvendo a topologia de árvore foram realizados. Durante esses testes, foi adicionado à rede o nó Master, responsável por receber todas as informações aferidas pelos nós coletores. Neste modelo foi utilizado um nó coletor, um nó intermediário e um nó Master. Em relação às distâncias nas quais a rede se mantinha operacional, não houveram mudanças significativas. Houveram porém problemas em relação a perda de

alguns pacotes durante as comunicações. Configurações de reenvio de pacotes em caso de não recebimento foram adicionadas aos nodos, amenizando assim este problema.

Como previamente citado, em alguns momentos a rede não funcionou corretamente sem motivo aparente. Mantendo as mesmas configurações e ambientes de testes, perdas de pacotes em taxas elevadas ocorreram arbitrariamente. Conhecendo a natureza sensível dos componentes transceptores, acredita-se que a reconstrução dos nodos utilizando materiais de melhor qualidade, como fios e soldas, poderão amenizar esse problema.

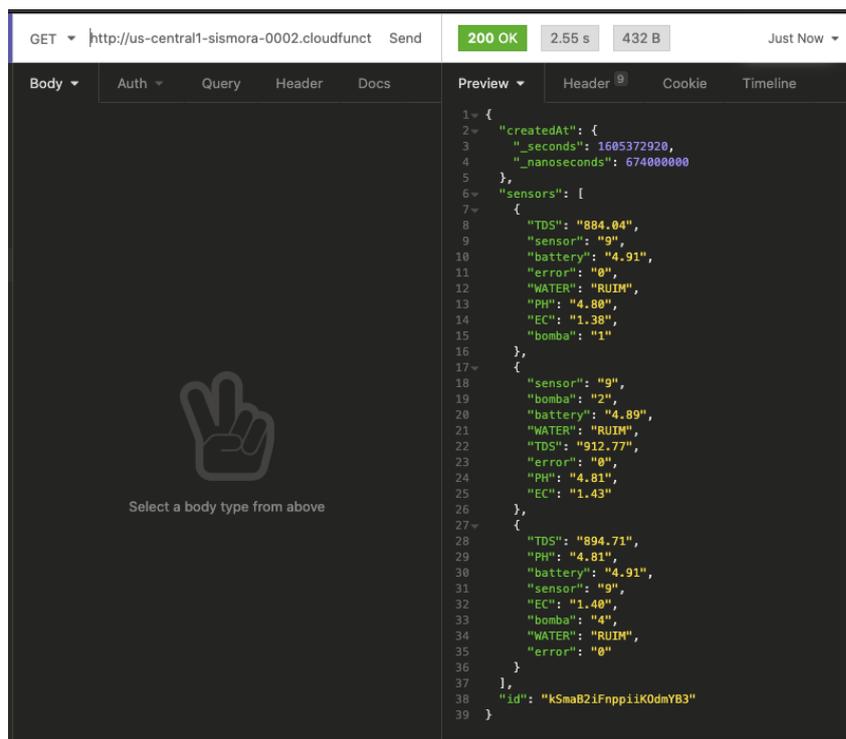
4.2 TESTE DA API

Como o protocolo de comunicação da API é o HTTP, testes puderam ser realizados a partir de requisições manipuladas. Para a elaborar tais requisições, foi utilizado o software *Insomnia*. Requisições similares às que seriam utilizadas de fato, tanto pela rede sensorial, a partir do nó Master através de GSM-GPRS, quanto pelo aplicativo móvel, foram elaboradas. A partir dessas requisições, foi possível cobrir vários tipos de mensagens, como adição de novos dados, leitura de dados antigos e leitura de dados de geolocalização, presentes no banco de dados.

Esta etapa de testes ocorreu sem problemas críticos, validando assim o bom comportamento da API em diversas situações. Observou-se que após alguns minutos sem ser utilizada, a API demora à responder as primeiras requisições. Este comportamento, chamado *cold start*, é esperado, uma vez que é uma característica conhecida de serviços *serverless*. Isso ocorre pois a API funciona sob demanda, e quando requisições são feitas à API, serviços internos da plataforma Firebase precisam iniciar uma instância da API em seus servidores para servir à requisição efetuada. Este tempo, de aproximadamente três segundos, não constitui contratempo significativo aos usuários da aplicação.

A Figura 17 apresenta um exemplo de teste feito para validação da API utilizando o software *Insomnia*. Nesse teste foi feita uma requisição do tipo GET à rota */sensordata*, que deve retornar os dados da última leitura efetuada pela rede sensorial.

Figura 17 – Teste de API



```
GET http://us-central1-sismora-0002.cloudfunc Send 200 OK 2.55 s 432 B Just Now
Body Auth Query Header Docs Preview Header Cookie Timeline
1 {
2   "createdAt": {
3     "_seconds": 1605372920,
4     "_nanoseconds": 674000000
5   },
6   "sensors": [
7     {
8       "TDS": "884.04",
9       "sensor": "9",
10      "battery": "4.91",
11      "error": "0",
12      "WATER": "RUIM",
13      "PH": "4.80",
14      "EC": "1.38",
15      "bomba": "1"
16    },
17    {
18      "sensor": "9",
19      "bomba": "2",
20      "battery": "4.89",
21      "WATER": "RUIM",
22      "TDS": "912.77",
23      "error": "0",
24      "PH": "4.81",
25      "EC": "1.43"
26    },
27    {
28      "TDS": "894.71",
29      "PH": "4.81",
30      "battery": "4.91",
31      "sensor": "9",
32      "EC": "1.40",
33      "bomba": "4",
34      "WATER": "RUIM",
35      "error": "0"
36    }
37  ],
38  "id": "KSmaB2iFnppiik0dmYB3"
39 }
```

Fonte: O Autor

Durante esse teste, foi possível observar as seguintes informações: Código de resposta HTTP 200, que representa que a requisição foi bem sucedida, tempo de resposta de 2.55 segundos, valor esperado de *cold start* e 412B, representando 412 bytes de informação recebida.

4.3 TESTE DO APLICATIVO MÓVEL

Assim como os testes da API, os testes da aplicação não apresentaram problemas críticos. Para agilizar o processo de desenvolvimento e teste unitário de cada nova funcionalidade desenvolvida para aplicação, foi utilizada a plataforma Expo.

A principal vantagem na utilização da plataforma Expo para desenvolver aplicativos React Native é que esta permite que alterações no código sejam refletidas quase instantaneamente na aplicação, não sendo necessária a recompilação do código a cada modificação aplicada (EXPO..., s.d.). A plataforma também provém o desenvolvedor com acesso às APIs nativas do dispositivo móvel, sem a necessidade da instalação de dependências nativas ou alterações de códigos nativos. A plataforma também facilita o processo de desenvolvimento e atualização gradativa, sendo uma boa alternativa para projetos pequenos ou com caráter de protótipo, como este.

Testes foram então realizados após a implementação de cada nova funcionalidade, não havendo um grande teste integrador durante o processo de desenvolvimento dessa

Figura 19 – Foto do ponto de medição



Fonte: Google Maps, Disponível em: <<https://goo.gl/maps/QtWihuaxkK1CSA6DA>> [Acessado em 17/11/2020].

Medições foram realizadas nos dias 14 e 18 de Novembro de 2020. Com intervalos de 1 minuto entre as medições, a rede foi configurada com a seguinte mensagem SMS: *CAI: 0,1,0,1*, ou seja, intervalos de 1 minuto e a configuração deve ser repassada aos nós imediatamente. As Tabelas 3 e 4 descrevem os dados lidos durante os testes da rede.

14/11/20

Tabela 3 – Dados lidos dia 14/11

Data	pH	TDS	Irrigação
1:55:20PM	4.8	884.04	RUIM
1:56:20PM	4.9	879.56	RUIM
1:57:22PM	4.78	880.12	RUIM
1:58:23PM	4.92	882.05	RUIM

18/11/20

Tabela 4 – Dados lidos dia 18/11

Data	pH	TDS	Irrigação
6:07:57PM	5.47	983.45	CUIDADO
6:08:58PM	5.86	990.13	BOA
6:09:58PM	5.67	978.25	BOA
6:11:00PM	5.48	981.03	CUIDADO

Segundo os dados lidos, o pH da água é o principal fator que faz com que a água seja imprópria para a irrigação na maior parte do tempo. Mesmo nas medições em que a plataforma aponta a qualidade água como 'BOA' o índice do pH manteve-se muito próximo ao limiar aceitável de pH. Índices de TDS mantiveram-se dentro do limite adequado.

Durante o processo de medição observou-se a necessidade da utilização de uma boia para manter o sensor na água mesmo durante os testes. Como esta não foi projetada, durante os testes o nó precisou ser suspenso sobre a água enquanto os sensores executavam as aferições. Sugere-se a utilização de uma boia como em (SILVA, s.d.).

A Figura 20 apresenta o momento da medição feitas na etapa de testes.

Figura 20 – Medição com nó coletor



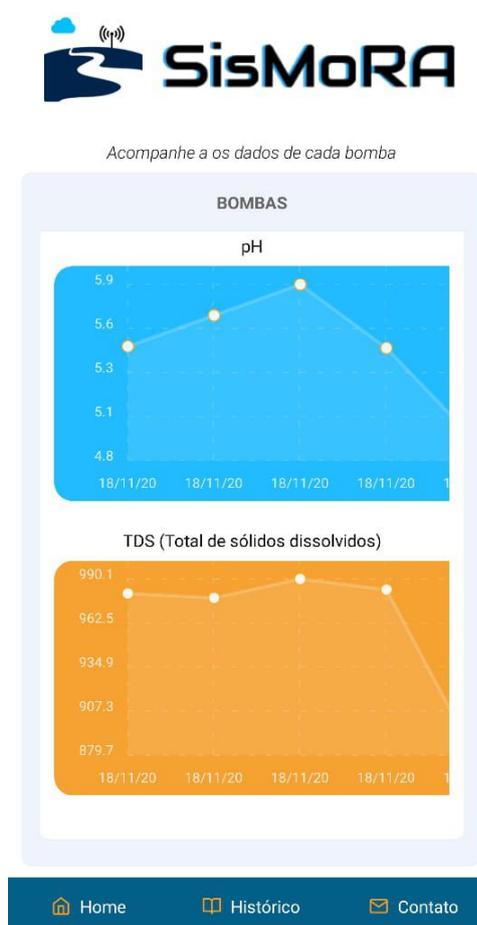
Fonte: O Autor

No que diz respeito à comunicação até o aplicativo, esta aconteceu de maneira imediata após as leituras, validando assim o caminho de dados proposto. Apesar de o aplicativo não atualizar automaticamente a cada nova leitura efetuada pela rede sensorial, ao atualizar a tela, ou reiniciar o aplicativo manualmente, este busca sempre as informações mais recentemente lidas.

Para visualizar os gráficos dos dados lidos, basta acessar o aplicativo e utilizar a aba "Histórico" para carregar os dados desejados. A partir dos dados carregados, é possível visualizar os gráficos utilizando o botão "Visualizar gráficos". Estes apresentam ao usuário gráficos de pH e TDS das leituras previamente carregadas.

A Figura 21 apresenta os gráficos gerados pelo aplicativo após os testes realizados.

Figura 21 – Gráficos das leituras



Fonte: O Autor

Durante os testes, o principal ponto de falha foi a rede sensorial. Como previamente descrito, a rede demonstrou-se sensível a diferentes ambientes, apresentando falhas arbitrárias. Para corrigir tal problema sugere-se a reconstrução de soldas e verificação de conexões físicas entre os componentes dos nós.

5 CONCLUSÃO

O presente trabalho apresentou os resultados do desenvolvimento de um protótipo para a plataforma experimental de monitoração de condições físico químicas do Rio Araranguá. O protótipo foi construído a partir de transceptores nRF24, Arduinos e sensores de pH, TDS e temperatura, em conjunto com uma API *serverless* e uma aplicação móvel para dispositivos Android, validando uma possível solução para o problema proposto.

Levando em consideração o amplo escopo do projeto proposto e por consequência a necessidade de validá-lo em sua totalidade como protótipo, os sensores, microcontroladores e demais periféricos escolhidos para integrar a rede de sensores foram escolhidos por atenderem as demandas técnicas do projeto. Dessa maneira, entende-se que este protótipo não reflete necessariamente as configurações e arquitetura a serem aplicadas na construção da plataforma final, que por sua vez atenderá a todas as demandas do projeto SisMoRA. Isto posto, considerando a filosofia modular aplicada à construção deste protótipo, eventuais alterações na quantidade e tipos de sensores poderão ser feitas a custo de alterações mínimas nos demais componentes da plataforma, uma vez que o backend de serviço e aplicação não dependem diretamente da arquitetura da rede sensorial. Atentando-se aos ajustes e adições propostas nas seções abaixo, este protótipo poderá suprir as necessidades do projeto SisMoRA integralmente.

5.1 TRABALHOS FUTUROS

Haja vista que o trabalho proposto tem caráter de protótipo, alguns pontos relevantes, considerando a aplicação final da rede, foram omitidos. Estes, não menos importantes, ficam aqui sugeridos para continuação e aperfeiçoamento deste trabalho.

5.1.1 Alimentação dos nós

Considerando a natureza do funcionamento da rede de sensores, que operará de maneira remota, distribuída e parcialmente autônoma, a definição de um sistema de alimentação capaz de manter a rede operante por um período de tempo adequado, é considerada muito importante.

Baterias 9v como as utilizadas neste projeto são insuficientes para manter os nós operando por longos períodos de tempo. Medições precisas em relação ao consumo de cada tipo de nó (coletor, intermediário e master), deverão ser feitas em laboratório com equipamentos adequados.

Contudo, algumas mudanças na arquitetura poderão ser feitas na rede no intuito de mitigar esse problema. Um deles é a integração de um RTC (*Real Time Clock*) ou Relógio de Tempo Real. Este serviria para dois propósitos. O primeiro, a funcionalidade de *sleep*, modo de baixo consumo de bateria porém de funcionamento limitado, dos microcontroladores

Atmel328p (presentes na configuração dos nós) poderia ser utilizada, uma vez que o RTC pode ser configurado para, através de interrupções externas, "acordar" os nós em períodos pré-definidos. Segundo, a integração de um RTC facilitaria a sincronização dos nós da rede, uma vez que para a comunicação acontecer de maneira completa através da rede, todos os nós devem estar ativos e operantes. Essa substituição poderia servir também para melhorar a maneira que o intervalo de leitura da rede é configurado.

Outras soluções como a construção de módulos personalizados a partir do microcontrolador Atmel32p, utilizado na placa de desenvolvimento Arduino, também pode ser considerada. Fazendo isso, seria possível evitar que alguns componentes inutilizados da placa, como LEDs ou reguladores de tensão, contribuíssem para a utilização das baterias.

Uma última solução a ser estudada é a aplicação de módulos solares para a alimentação de cada nó.

5.1.2 Case para os nós

Considerando o ambiente de operação da rede de sensores, será necessário o projeto e construção de cases a prova da água para envolver o microcontrolador, sensores e baterias. Boias deverão ser projetadas para abrigar os nós medidores. Nós intermediários e o *gateway* não necessariamente necessitarão desse ajuste pois podem operar fora de contato com a água do rio. Uma possível solução para este problema está proposta em (SILVA, s.d.).

5.1.3 Substituição de sensores

Como previamente discutido, o sensor de pH escolhido para integrar este projeto, o *Gravity:SEN0161* é considerado um sensor de laboratório e não pode ficar imerso em água por longos períodos, a custo de perda de precisão nas aferições. Sugere-se então sua substituição pelo modelo *Gravity:SEN0169*, modelo mais robusto e que pode ficar imerso em água por períodos de até um ano sem comprometer a qualidade de suas leituras. Por ser um sensor do mesmo fabricante, sua substituição poderá ser realizada facilmente, uma vez que conexões físicas e software necessário para sua configuração são similares.

Sugere-se também trocar o sensor de TDS KS0429, por um condutivímetro. Dessa maneira, índices mais relevantes e precisos poderão ser aferidos e melhorarão a confiabilidade de cada informe gerado pela plataforma.

REFERÊNCIAS

- A. F. RUSYDI. Correlation between conductivity and total dissolved solid in various type of water: A review. **IOP Conf. Ser.: Earth Environ.**, 2018.
- AKYILDIZ, I., VURAN, M. Wireless Sensor Networks. **John Wiley Sons**, v. 4, 2010.
- ALI, N.S., MO, K. KIM, M. **A case study on the relationship between conductivity and dissolved solids to evaluate the potential for reuse of reclaimed industrial wastewater**. 2012. F. 708–713. Tese (Doutorado).
- ANDROID está em 91,6% dos celulares vendidos no 1º semestre. [*S.l.: s.n.*]. <http://g1.globo.com/tecnologia/noticia/2014/09/android-esta-em-916-dos-celulares-vendidos-no-1-semester-diz-pesquisa.html>. Accessed: 2020-12-06.
- AQEEL-UR-REHMAN. A review of wireless sensors and networks' applications in agriculture. Computer Standards and Interfaces. **Computer Standards and Interfaces**, **36(2)**, 263–270, 2014.
- AXELSSON, Oscar *et al.* **Evaluation Targeting React Native in Comparison to Native Mobile Development**. eng. [*S.l.: s.n.*], 2016. Student Paper.
- BUDAKOTI, J. **An IoT Gateway Middleware for interoperability in SDN Managed Internet of Things**. 2018. F. 108. Tese (Doutorado).
- EPAGRI conclui mapeamento da área de arroz por imagens de satélite. [*S.l.: s.n.*]. <https://www.epagri.sc.gov.br/index.php/2020/07/31/epagri-conclui-mapeamento-por-imagens-de-satelite-da-area-de-cultivo-de-arroz/>. Accessed: 29-10-20.
- EPAGRI.2011. Parecer sobre a salinidade do rio Araranguá e suas implicações para a cultura do arroz irrigado., 2011.
- EXPO Documentation. [*S.l.: s.n.*]. <https://docs.expo.io/>. Accessed: 2020-12-06.
- FENG, X. Study of Wireless Communication Technologies on Internet of Things for Precision Agriculture. **Wireless Personal Communications**, **108(3)**, 1785–1802., 2019.
- FIREBASE Pricing. [*S.l.: s.n.*]. <https://firebase.google.com/pricing>. Accessed: 2020-12-06.
- FRAGA, Thiago Isquierdo *et al.* Attributes of irrigated rice and soil solution as affected by salinity levels of the water layer. en. **Revista Brasileira de Cia do Solo**, scielo, v. 34, p. 1049–1057, ago. 2010. ISSN 0100-0683. Disponível em:

http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-06832010000400005&nrm=iso.

FUNASA, Fundação Nacional de Saúde. Manual de Controle da Qualidade da Água para Técnicos que Trabalham em ETAS. **Ministério da Saúde, Fundação Nacional de Saúde - Brasília**, 2014.

GRAVITY: Analog pH Sensor / Meter Kit For Arduino. [*S.l.: s.n.*].
<https://www.dfrobot.com/product-1025.html>. Accessed: 29-10-20.

H. SAHA. Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad-hoc Networks. **International Journal of Computer Network and Information Security(IJCNIS)**, Vol.9, No.7, 2017.

HALONEN, T. **GSM, GPRS, and EDGE Performance - Evolution Towards 3G/UMTS**. England: John Wiley Sons Ltd, 2003.

IBGE, - Instituto Brasileiro de Geografia e Estatística. Censo agropecuário de 2006. **Produção de arroz em casca na lavoura temporária, segundo a agricultura familiar e Pronaf - Santa Catarina, year = 2006**.

_____. Censo agropecuário de 2017. **Produção de arroz em casca na lavoura temporária, segundo a agricultura familiar e Pronaf - Santa Catarina, year = 2017**.

IYASELE, J.U. Investigation of the Relationship between Electrical Conductivity and Total Dissolved Solids for Mono-Valent, Di-Valent and Tri-Valent Metal Compounds. **International Journal of Engineering Research and Reviews**, 01 2015.

J. M.B. PARFITT. Soil and Water Management for Sprinkler Irrigated Rice in Southern Brazil. **Advances in International Rice Research - Chapter 1**, 2016.

J. P. FONSECA. Monitoramento hospitalar para avaliação remota de pacientes, 2018.

MIGNACO, A. G. Análise de desempenho de redes de sensores sem fio., 2005.

MORONEY, L. **The Definitive Guide to Firebase**. [*S.l.*]: Apress, 2017.

NISARG GANDHEWAR, Rahila Sheikh. Google Android: An Emerging Software Platform For Mobile Devices. **International Journal on Computer Science and Engineering (IJCSE)**, year = 2010.

NRF24, How nRF24L01+ Wireless Module Works Interface with Arduino. [*S.l.: s.n.*].
<https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>. Accessed: 29-10-20.

- P. MELL, Timothy Grance. The NIST Definition of Cloud Computing. **Recommendations of the National Institute of Standards and Technology - NIST**, 2011.
- PROFILL, Secretaria do Desenvolvimento Econômico Sustentável - SC. Plano de recursos hídricos da bacia do Rio Araranguá. **Relatório B1 - Consolidação das informações sobre recursos hídricos - Etapa B**, 2014.
- QURESHI, K. N. Optimized Cluster-Based Dynamic Energy-Aware Routing Protocol for Wireless Sensor Networks in Agriculture Precision. **Journal of Sensors**, 2020.
- ROOSTAEI, Javad. **IoT-based Edge and Cloud Computing for Smart Environmental Engineering Applications**. Jun. 2018. Tese (Doutorado). DOI: 10.13140/RG.2.2.22089.65121.
- SEMICONDUCTOR, Nordic. nRF24L01+ Single Chip 2.4GHz Transceiver. <https://datasheetspdf.com/pdf-file/906332/Nordic/nRF24L01+/1>, 2008.
- SILVA, Gustavo. Monitoramento remoto do Rio Araranguá para a irrigação no plantio de arroz através de uma rede de sensores sem fio (RSSF). **Trabalho de conclusão de Curso - Engenharia de Computação - UFSC year = 2018**.
- VANDIKAS, K., TSIATSIS, V. Performance evaluation of an IoT platform. **Next Generation Mobile Apps, Services and Technologies (NGMAST)**, 2014.
- YICK, J. Wireless sensor network survey. 52, 2008.
- ZHAO, W. *et al.* Design and Implementation of Smart Irrigation System Based on LoRa. *In: 2017 IEEE Globecom Workshops (GC Wkshps)*. [S.l.: s.n.], 2017. P. 1–6. DOI: 10.1109/GLOCOMW.2017.8269115.

ANEXO A – CONFIGURAÇÃO DOS NÓS

Neste anexo é descrito o processo de montagem dos nodos. Entretanto, por existirem vários terminais com a mesma funcionalidade, tanto no Arduino UNO quando no Arduino MEGA, no que diz respeito as conexões entre os componentes, configurações diferentes a estas também poderão ser utilizadas.

- **Nó intermediário**

Componentes necessários:

- Arduino UNO
- nRF24L01+ pa/lna com antena
- Adaptador nRF24
- Bateria 9V
- Jumpers diversos

Conexões:

Arduino	Adaptador nRF24
5V	VCC
GND	GND
7	CE
8	CSN
13	SCK
11	MOSI
12	MISO

- **Nó Coletor** Componentes necessários:

- Arduino UNO
- nRF24L01+ pa/lna com antena
- Adaptador nRF24
- Bateria 9V
- Jumpers diversos
- Sensor de TDS - KS0429
- Sensor de pH - SEN0161
- Sensor de Temperatura - DS18B20

Conexões:

Arduino	Adaptador nRF24
5V	VCC
GND	GND
7	CE
8	CSN
13	SCK
11	MOSI
12	MISO

Arduino	Sensor
A1	Sensor TDS
A2	Sensor pH
A3	Sensor Temperatura

* Todos os três sensores precisam de alimentação. Dessa maneira conectam-se os terminais equivalentes de cada sensor da seguinte maneira: *GND* -> *GND* e *VCC* -> *5V*

• **Nó Master** Componentes necessários:

- Arduino MEGA
- Shield GSM SIM900
- nRF24L01+ pa/lna com antena
- Adaptador nRF24
- Bateria 9V
- Jumpers diversos

Conexões:

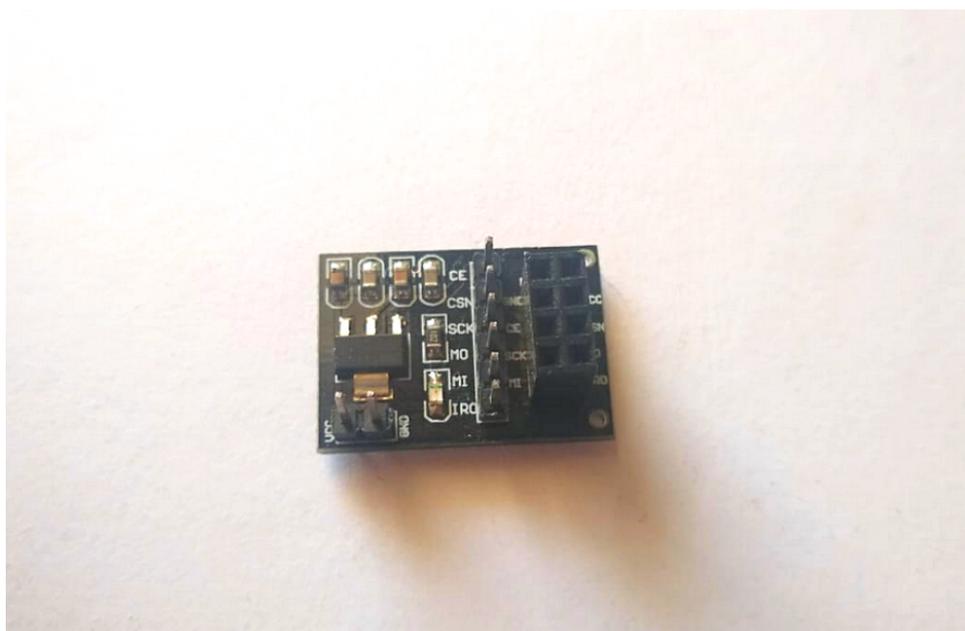
Arduino	Adaptador nRF24
GND	GND
5V	VCC
53	CE
48	CSN
52	SCK
51	MOSI
50	MISO

* Após acoplar o Shield GSM ao Arduino MEGA, deve-se conectar o pino TX (GSM) até o terminal 18 (MEGA) e o pino RX (GSM) até o terminal 19 (MEGA).

Vale lembrar que a utilização do adaptador é de suma importância. Os transceptores nRF24 são muito sensíveis a perturbações eletromagnéticas, e pequenos ruídos são suficientes para comprometer a comunicação entre dois nós. Dessa maneira, o adaptador possui um regulador de tensão, que ajuda manter a tensão de alimentação estável, e um capacitor de desacoplamento, que ajuda a filtrar ruídos entre VCC e GND.

A Figura 22 apresenta o adaptador nRF24.

Figura 22 – Adaptador nRF24



Fonte: O Autor