



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Pedro Alexandre Barradas da Côte

**Sistema de gerenciamento autônomo de consumo de água para casas inteligentes
utilizando IoT**

Florianópolis
2020

Pedro Alexandre Barradas da Côte
**Sistema de gerenciamento autônomo de consumo de água para casas inteligentes
utilizando IoT**

O presente trabalho em nível de bacharelado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof^ª. Carla Merkle Westphall, Dr^ª.
Universidade Federal de Santa Catarina

Bel. Ricardo do Nascimento Boing
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Bacharel em Ciência da Computação.

Prof. Jose Francisco Danilo de Guadalupe Correa Fletes, Dr
Coordenador do Programa

Prof. Carlos Becker Westphall, Dr.
Orientador

Florianópolis, 2020.

Pedro Alexandre Barradas da Côrte

Sistema de gerenciamento autônomo de consumo de água para casas inteligentes utilizando IoT

TCC submetido ao Programa de Graduação em
Ciência da Computação para a obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. Carlos Becker Westphall, Dr.
Coorientador: Prof. Hugo Vaz Sampaio, Me.

Florianópolis

2020

LISTA DE ILUSTRAÇÕES

Figura 1 – Descrição de camadas de um sistema <i>Fog</i>	11
Figura 2 – O balanço de água da IWA para a África do Sul com FBW (<i>Free Basic Water</i>)	14
Figura 3 – Proporção de consumo de água de 18 usuários durante 24 horas	19
Figura 4 – Diagrama Básico de Blocos	20
Figura 5 – Modelo de criação de contexto em três níveis	21
Figura 6 – <i>Homepage</i> da aplicação <i>web</i>	22
Figura 7 – Medidor de fluxo de água YF-S201	23
Figura 8 – Conexão de um sensor YF-S201 com o Arduíno	23
Figura 9 – Leitura serial de 3 sensores YF-S201 conectados no Arduíno	24
Figura 10 – Interface do <i>thingSpeak</i>	24
Figura 11 – <i>Framework</i> do <i>Edge-Based Energy Management System</i> (EEMS)	26
Figura 12 – Ambiente de testes para o EEMS	27
Figura 13 – Resultados dos testes (preferências dos usuários)	27
Figura 14 – Resultados dos testes (preferências do algoritmo)	28
Figura 15 – Comparação dos resultados dos testes (usuários em vermelho, algoritmo em azul)	28
Figura 16 – Consumo de energia da rede e produção de energia solar	29
Figura 17 – Dados coletados com relação à coluna “Total” da Tabela 5	32
Figura 18 – Dados coletados com relação à coluna “Por Hora” da Tabela 5	32
Figura 19 – Dados coletados com relação à coluna “Diário” da Tabela 5	33
Figura 20 – Dados coletados com relação à coluna “Mensal” da Tabela 5	33
Figura 21 – Rede de sensores sem fio	35
Figura 22 – Quatro principais desafios da gestão de água na agricultura moderna	36
Figura 23 – WDS em uma cidade inteligente	40
Figura 24 – Arquitetura do SWD baseado em IoT	42
Figura 25 – Diagrama de Fluxo da SWDN	42
Figura 26 – Calibração do sensor de nível de capacitância	43
Figura 27 – Diagrama de blocos representando a arquitetura do sistema	45
Figura 28 – Visão geral da proposta	47
Figura 29 – Visão do Arduíno e sua implementação na proposta	48
Figura 30 – Visão do dispositivo IoT e sua implementação na proposta	49
Figura 31 – Arduíno e suas conexões físicas com o sensor YF-S201	50
Figura 32 – Visão do Arduíno e sua implementação no dispositivo IoT da proposta	51
Figura 33 – Exemplo do servidor web da proposta	53
Figura 34 – Exemplo da janela da porta serial do Arduíno	55
Figura 35 – Protótipo em ação, conectado à mangueira	55
Figura 36 – Protótipo em ação, com fluxo “grande” de água visível	56
Figura 37 – Protótipo em ação, com fluxo “pequeno” de água visível	56

Figura 38 – Gráfico das leituras individuais	58
Figura 39 – Gráfico das leituras acumuladas	58
Figura 40 – Fluxograma do código do Arduino	69
Figura 41 – Visão das interações do Arduino com o servidor da proposta	70

SUMÁRIO

1	INTRODUÇÃO	9
1.1	MOTIVAÇÃO E JUSTIFICATIVA	9
1.2	OBJETIVOS	10
1.2.1	Objetivos específicos	10
1.3	MÉTODO DE PESQUISA	10
2	CONCEITOS FUNDAMENTAIS	11
2.1	FOG COMPUTING	11
2.2	INTERNET OF THINGS	12
2.3	MEDIDORES INTELIGENTES	12
2.4	SEGURANÇA DE REDES IOT	15
3	TRABALHOS RELACIONADOS	17
3.1	REVISÃO BIBLIOGRÁFICA SISTEMÁTICA	17
3.2	ANÁLISE DE CONSUMO DE ÁGUA EM ÁREAS URBANAS	18
3.3	SMART HOME COM MEDIÇÃO ELETRÔNICA EM TEMPO REAL	19
3.4	GERENCIAMENTO DE ÁGUA BASEADO EM IOT	22
3.5	GERENCIAMENTO DE ENERGIA BASEADO EM EDGE PARA SMART HOMES	25
3.6	SISTEMA AUTÔNOMO PARA MONITORAMENTO DE CONSUMO DE ÁGUA EM APARTAMENTOS	29
3.7	PESQUISA SOBRE DESAFIOS DO GERENCIAMENTO DE ÁGUA NA AGRICULTURA	34
3.8	DISTRIBUIÇÃO E MONITORAMENTO DE TUBULAÇÕES COM IOT PARA CIDADES INTELIGENTES	39
3.9	SISTEMA DE GERENCIAMENTO DE ÁGUA BASEADO EM IOT	43
3.10	DESAFIOS EM SISTEMAS DE MONITORAMENTO	45
4	PROPOSTA DO SISTEMA DE GERENCIAMENTO DE ÁGUA	47
5	DESENVOLVIMENTO DE UM PROTÓTIPO	50
5.1	HARDWARE DO PROTÓTIPO IOT	50
5.2	SOFTWARE DO PROTÓTIPO IOT	51
5.3	SERVIDOR WEB (FOG)	52
6	TESTES E RESULTADOS	54
7	CONCLUSÕES	59
7.1	TRABALHOS FUTUROS	59

REFERÊNCIAS	60
APÊNDICE A – CÓDIGO DO ARDUÍNO	62
APÊNDICE B – FLUXOGRAMA DO ARDUÍNO	69
APÊNDICE C – FLUXOGRAMA DO SERVIDOR	70
APÊNDICE D - ARTIGO NO FORMATO SBC	71

RESUMO

Ambientes residenciais podem ser automatizados com o uso da Internet das Coisas, o que permite que várias tarefas sejam colocadas nas mãos de computadores. Uma dessas tarefas é a verificação de consumo de água de uma residência. Gastos exagerados ou vazamentos podem ocorrer em uma casa, e esses problemas podem passar despercebidos por muito tempo. Atualmente, a forma mais comum de evitar ou remediar tais problemas é ficar atento aos números do medidor de água, e saber interpretá-los. Este TCC propõe um sistema que automatiza a verificação e interpretação do consumo de água de uma casa, assim ajudando a reduzir gastos desnecessários e a identificar problemas (como vazamentos) de forma mais rápida e eficaz. O sistema inclui um sensor que faz as medições de água e as envia para um servidor programado em um Arduino, o qual contabiliza o consumo total para vários períodos e também faz todos os cálculos necessários para identificar possíveis problemas.

Palavras-chave: Gerenciamento Autônomo. Internet das Coisas. Computação em Névoa. Gerenciamento de água. Casas inteligentes.

ABSTRACT

Residential environments can be automated using the Internet of Things, which allows multiple tasks to be handed over to computers. One of these tasks is to check the water consumption of a house. Excessive water usage or water leaks can occur in a house, and these problems can go undetected for a long time. Currently, the most common way to avoid or remedy such problems is to pay attention to the water meter's numbers and to know how to interpret them. This TCC proposes a system that automates the verification and interpretation of water consumption in a home, thus helping to reduce unnecessary expenses and to identify problems (such as leaks) more quickly and effectively. The system includes a sensor that makes water measurements and sends them to a server programmed in an Arduino, which counts the total consumption for several periods and also makes all the necessary calculations to identify possible problems.

Keywords: *Autonomic Management. Internet of Things. IoT. Fog Computing. Water management. Smart homes.*

1 INTRODUÇÃO

A evolução das tecnologias de comunicação e de aparelhos inteligentes trouxe consigo a possibilidade de automatizar diversas tarefas em diversas áreas de trabalho ou pesquisa. O advento de sistemas *Fog* permitiu que tais tarefas não fossem limitadas apenas ao mesmo espaço onde a computação ocorre, trazendo soluções inovadoras para lugares desprovidos de acesso a computadores. Alrawais et al. (2017) explica que a computação *Fog* permite uma nova geração de serviços na borda da rede. A única coisa necessária é uma conexão.

Diversas soluções tem como foco a automatização de ambientes residenciais, com o objetivo de facilitar a vida das pessoas, como pode ser observado no trabalho de Bhilare e Mali (2016). Desde luzes que acendem com um comando verbal, até sistemas que controlam o uso de eletricidade de eletrodomésticos para reduzir gastos, todo o tipo de ideias podem apresentar viabilidade para casas inteligentes.

Um dos gastos que pode ser controlado de forma autônoma em residências é o de recursos hídricos, um dos bens mais preciosos da raça humana. Este trabalho busca propor um sistema que colete dados de consumo de água, guarde-os, analise-os automaticamente, e que tenha a capacidade de avisar os usuários sobre excesso de uso ou vazamentos com antecedência, fazendo um gerenciamento autônomo de consumo de água. Todo esse processo será facilitado pelo uso da Internet das Coisas (*Internet of Things*, ou IoT) e dispositivos inteligentes, garantido aos usuários um acesso fácil e rápido aos dados obtidos.

Hoje, temos que nos preocupar com todo o tipo de problema, e muitas vezes ficamos sobrecarregados. Com isso, a automatização nos oferece uma tranquilidade, pois a preocupação com alguma atividade é colocada em uma máquina que cuida de tudo sozinha, e apenas nos notifica quando precisa de intervenção humana. Uma das preocupações em casas é a possibilidade de vazamentos de água, que por vezes se manifestam apenas no final do mês, por meio de contas altas. Um sistema poderia ajudar a evitar que vazamentos passem despercebidos por muito tempo, e este é o principal motivo da proposta deste trabalho.

Este TCC está organizado em 7 capítulos. Esta é a primeira seção, e em seguida vem a seção dos conceitos fundamentais. A terceira seção contém os dados da revisão bibliográfica realizada para fundamentar este trabalho, e alguns dos trabalhos foram selecionados e analisados em mais detalhes na sequência. Na quarta seção, a proposta deste trabalho é apresentada, e a mesma é desenvolvida na quinta seção. A sexta seção apresenta os resultados dos testes feitos e a sétima seção conclui o trabalho.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Um ponto se destaca como a principal motivação para o desenvolvimento deste trabalho: vazamentos e contas altas de água sem nenhum tipo de aviso prévio já foram problemas experienciados pelo autor. Ao avaliar as possibilidades de automação e gerenciamento que a IoT possibilita, foi possível chegar à conclusão que, para ambientes residenciais, é possível

controlar todo o tipo de recurso que é consumido, como energia e água.

Este projeto visa utilizar automação de gerência para que problemas com vazamentos possam ser evitados ou, pelo menos, para que não sejam tão caros.

1.2 OBJETIVOS

O objetivo principal deste trabalho é propor e desenvolver um sistema de gerenciamento autônomo de medidores de consumo de água de uma residência, que realizará a verificação de níveis de consumo de água em casas, focando na economia de água, bem como facilitar o processo de obtenção de tais dados.

1.2.1 Objetivos específicos

- Automatizar e armazenar as medições de água de uma residência
- Detectar vazamentos com base em padrões de consumo
- Notificar o usuário de anomalias ou limites de consumo alcançados
- Facilitar a obtenção das medições de volume de água

1.3 MÉTODO DE PESQUISA

A princípio, foi feita uma pesquisa e avaliação de trabalhos já publicados que são relevantes ao tema deste TCC, com o objetivo de alcançar um melhor entendimento dos conceitos envolvidos, desde as funcionalidades e estruturas de sistemas IoT até a utilização e instalação dos sensores necessários.

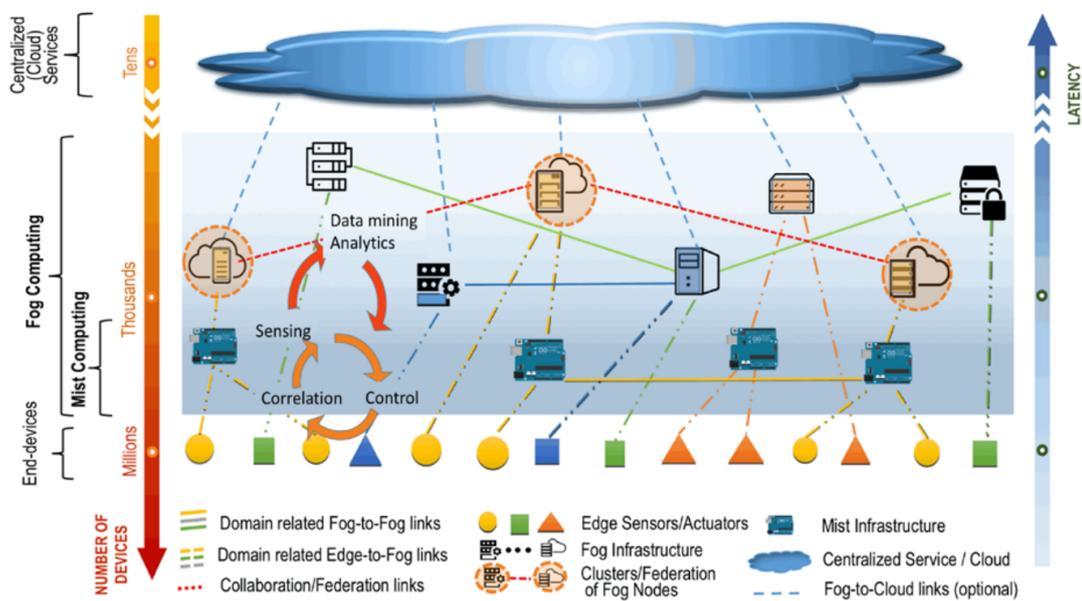
2 CONCEITOS FUNDAMENTAIS

2.1 FOG COMPUTING

A computação em névoa, ou *Fog Computing*, permite descentralizar o gerenciamento e o processamento de dados, executando tais tarefas na própria rede local dos sensores e atuadores implantados (IORGA et al., 2018). Este modelo ajuda a liberar recursos da nuvem, reduzindo a dependência de sistemas para todas as operações, além de resultar em um impacto positivo à qualidade de serviço e reduzir a latência da solução provida (ALRAWAIS et al., 2017).

O conceito *Fog Computing* pode ser definido como um modelo de computação separado em camadas, onde os recursos computacionais são escaláveis, ou seja, podem ser alterados para fornecer mais (ou menos) poder computacional, conforme as necessidades da rede. Na Figura 1, um modelo de *Fog Computing* é definido por Iorga et al. (2018), indicando as separações das camadas. Na camada superior situam-se os grandes centros de processamento, que possuem alto poder computacional e são responsáveis pela computação na nuvem (*Cloud Computing*). A camada intermediária, denominada *Fog*, encontra-se entre a *Cloud* e os dispositivos IoT.

Figura 1 – Descrição de camadas de um sistema *Fog*



Fonte – (IORGA et al., 2018)

Na camada inferior, estão presentes os nós, que são, essencialmente, computadores que recebem e processam os dados dos sensores e atuadores da rede. Os nós *Fog* (tanto físicos quanto virtuais) são sensíveis ao contexto e utilizam todos um sistema comum para gerenciar e comunicar os dados. Eles podem ser organizados em *clusters*, com o objetivo de dar suporte ao isolamento (organizados de forma vertical), de dar suporte à federação (organizados de forma horizontal), ou até mesmo para reduzir a latência (organizados de forma relativa à latência

entre os nós e os dispositivos da rede). Este modelo também pode fornecer, para os sensores, conectividade a serviços centralizados (IORGA et al., 2018).

2.2 INTERNET OF THINGS

Um dos maiores desafios encontrados ao tentar implantar um sistema de Internet das Coisas (*Internet of Things*, IoT) é a complexidade no gerenciamento de dados que são gerados por sensores e atuadores presentes na rede. Os sistemas IoT tradicionais, que utilizam majoritariamente processamento de dados na nuvem, sofrem principalmente com alta latência e heterogeneidade (de dispositivos e tipos de dados) (ALRAWAIS et al., 2017).

Segundo os autores Alrawais et al. (2017), os aparelhos presentes em uma IoT possuem características inerentemente desafiadoras (como baixo poder computacional, baixas taxas de transferência e armazenamento de dados limitado), as quais afetam a experiência do usuário final e também a qualidade do serviço (*Quality of Service*, QoS).

Um dos usos mais discutidos para um sistema IoT, com uso de computação na névoa, é a automatização de ambientes de moradia, em especial casas. O sistema proposto por Bhilare e Mali (2016) utiliza uma aplicação *web* (funcionando na nuvem) em conjunto com o sistema IoT para controlar aparelhos inteligentes dentro de uma casa, além de utilizar tal aplicação para monitorar os gastos com eletricidade. O objetivo é permitir que o usuário faça *login* no sistema para checar o consumo elétrico em tempo real e manualmente controlar seus aparelhos locais.

Um aspecto importante a ser considerado ao implantar um sistema IoT é o consumo de energia (corrente ou bateria) dos dispositivos presentes na rede (tanto os *smart devices* quanto os nós da névoa). Para a maioria dos sistemas IoT, é fundamental que os dispositivos implantados possam continuar ligados pela maior quantidade de tempo possível, o que pode se tornar difícil caso o consumo das baterias destes não sejam gerenciadas de forma efetiva (SAMPAIO et al., 2019). Esta necessidade fica realçada ainda mais com as previsões que haverá mais de 50 bilhões de dispositivos conectados em 2020 (IORGA et al., 2018).

2.3 MEDIDORES INTELIGENTES

Os medidores inteligentes funcionam da mesma forma como os medidores analógicos funcionam, com a principal diferença sendo que os inteligentes convertem as medidas para dados digitais, que podem ser utilizados por computadores ou servidores. Estes medidores também possuem conectividade com alguma rede local, permitindo comunicação com os dispositivos que farão uso dos dados, e classificando-os como dispositivos IoT.

Para realizar a medição automática de consumo de água, um sistema qualquer necessitaria de um desses medidores de água inteligentes. De acordo com Masia e Erasmus (2013), não existe uma definição universal para um medidor destes, mas sim definições com base nas funcionalidades adicionais, as quais vão além da medição e apresentação do consumo de água.

De acordo com os autores, existem algumas grandes categorias de acordo com o nível de sofisticação, sendo estas:

- Leitura Automatizada de Medidores (*Automated Meter Reading (AMR)*)
- Gerenciamento Automatizado de Medidores (*Automated Meter Management (AMM)*)
- Medição de Intervalo com Gerenciamento Automatizado de Medidores (*Interval Metering with Automated Meter Management (AMM)*)
- Medidores de Pré-pagamento (*Prepayment Meters (PM)*)
- Infraestrutura Avançada de Medição (*Advanced Metering Infrastructure (AMI)*)

As diferenças gerais destas 5 categorias foram detalhadas nas Tabelas 1 e 2. A Tabela 1 realiza a comparação das funções de medidores inteligentes diferentes, enquanto que na Tabela 2 são comparados os benefícios dos medidores inteligentes.

Tabela 1 – Comparação das funções de medidores inteligentes diferentes

Funções	AMR	AMM	Medição AMM	PPM	AMI
Possui um link de comunicação do medidor com o fornecedor para ler o medidor remotamente	X	X	X	X	X
Possui um link de comunicação entre fornecedor e medidor		X	X	X	X
O operador de rede pode limitar remotamente o suprimento de energia e desconectar, se necessário		X	X	X	X
Tarifas podem ser modificadas remotamente		X	X	X	X
Dados em tempo real podem ser mostrados para o usuário		X	X	X	X
Proteção contra violação e fraude		X	X	X	X
Medir o consumo de energia intervalos curtos (meia hora ou menos) enviar ao fornecedor			X	X	X
Pode ter várias estruturas tarifárias (tarifas por tempo de uso)			X	X	X
O fornecedor pode alternar o medidor entre crédito ou pagamento antecipado				X	X
Instalação de calibração remota				X	X
Pode fornecer informações detalhadas, como histórico de custo e crédito restante				X	X
Permitir alterar a locação				X	X
Entrada de crédito através de um keypad				X	X
Pode adicionar crédito remotamente					X
Pode controlar eletrodomésticos remotamente					X
Fornecer facilidades para projeto, operação e gerenciamento de rede					X

Tabela 2 – Comparação dos benefícios de medidores inteligentes diferentes

Benefícios	AMR	AMM	Medição AMM	PPM	AMI
A leitura manual do medidor não é necessária, portanto reduz o custo e as dificuldades práticas da leitura do medidor	X	X	X	X	X
Permite produzir faturas com leituras reais, em vez de faturas imprecisas estimadas	X	X	X	X	X
O cliente pode mudar de fornecedor rapidamente, pois estão disponíveis leituras precisas do medidor	X	X	X	X	X
Detectar e notificar fraudes quando um medidor for violado	X	X	X	X	X
Visitas e reajustes manuais do medidor não são necessários quando o preço e/ou as tarifas mudam		X	X	X	X
Conscientizar o cliente quanto à energia, o custo e a eficiência, para que o consumo seja ajustado para reduzir o custo			X	X	X
Instalações aprimoradas para clientes pré-pagos				X	X
Poderia ajudar a evitar apagões em larga escala através de derramamento de carga controlado durante eventos críticos de pico					X

Atualmente se faz necessário que a sociedade revise as políticas de consumo de água para garantir o uso adequado dos recursos hídricos. Masia e Erasmus (2013) explicam as melhores práticas de cálculo de consumo de água, de acordo com a Associação Internacional da Água (*International Water Association, IWA*). Na Figura 2 considera-se uma modificação nos cálculos mencionados para refletir as circunstâncias da África do Sul.

Figura 2 – O balanço de água da IWA para a África do Sul com FBW (*Free Basic Water*)

System Input Volume	Authorised Consumption	Billed Authorised Consumption	Billed Metered Consumption	Free Basic
		Unbilled Authorised Consumption	Billed Unmetered Consumption	Recovered Revenue
			Non-Recovered	
Water Losses	Apparent Losses	Real Losses	Unbilled Metered Consumption	Non Revenue Water
			Unbilled Unmetered Consumption	
	Real Losses	Unauthorised Consumption		
		Customer Meter Inaccuracies		
		Leakage on Transmission and Distribution Mains		
		Leakage and Overflows at Storage Tanks		
			Leakage on Service Connections up to point of Customer Meter	

Fonte – (MASIA; ERASMUS, 2013)

2.4 SEGURANÇA DE REDES IOT

As infraestruturas e serviços da IoT apresentam grandes desafios de segurança, devido ao aumento significativo da superfície de ataque, complexidade, heterogeneidade e número de recursos quando comparado com outros tipos de redes. De acordo com Pacheco e Hariri (2016), a integração de sistemas físicos e cibernéticos, bem como as interações humanas, aumentam drasticamente a vulnerabilidade e a superfície de ataque dos ecossistemas de infraestrutura interdependentes.

A arquitetura mais comum para monitorar e controlar infraestruturas inteligentes, como *Smart Homes*, são os Sistemas de Automação Predial (*Building Automation Systems*, BAS) e os Sistemas de Controle de Supervisão e Aquisição de Dados (*Supervisory Control and Data Acquisition*, SCADA). Pacheco e Hariri (2016) explicam que, à medida que os sistemas BAS e SCADA se interconectam com os recursos e serviços da internet, eles se tornam alvos fáceis para ataques, especialmente porque nunca foram projetados para lidar com ameaças cibernéticas, mas apenas para operar em um ambiente completamente isolado da internet. Isso torna os dados do sistema de controle vulneráveis a ataques de falsificação que levam à entrega incorreta de informações aos usuários, levando-os a tomar decisões erradas e perigosas ou a desconhecer que um ataque está em andamento, como foi o caso do ataque *Stuxnet*. Essa interconexão também permite que ataques executem comandos potencialmente maliciosos em sistemas de controle e dispositivos remotos, causando ações prejudiciais. Portanto, é extremamente importante proteger as operações de IoT de tais sistemas contra ataques cibernéticos.

As soluções tradicionais de segurança de TI não são diretamente aplicáveis à IoT. Suo et al. (2012) explica que isso é devido aos seguintes problemas:

- A IoT estende a internet através da internet tradicional, rede móvel, redes sem endereços de IP, rede de sensores, computação em nuvem e computação em nevoeiro;
- Plataformas de computação com restrição de capacidade de memória e processamento que, conseqüentemente, podem não suportar algoritmos de segurança complexos; e
- Todos os dispositivos da rede se comunicarão. Isso leva a vários pontos de acesso que podem ser usados para explorar vulnerabilidades existentes.

Greengard (2016) diz que as soluções atuais de segurança cibernética estão longe de ser satisfatórias para impedir o crescimento exponencial em número e complexidade de ataques cibernéticos. De acordo com Can e Sahingoz (2015), o esforço e o conhecimento necessários para lançar ataques sofisticados estão diminuindo, enquanto sua propagação foi reduzida de dias (no início dos anos 80) para uma fração de segundos (no novo milênio).

A superfície de ataques nas redes IoT é grande. Hossain, Fotouhi e Hasan (2015) consideram duas ramificações da rede IoT, sendo uma delas as de redes proximais (rede local, privada ou doméstica), onde as entidades são Controladores IoT, *Gateways* IoT, Coordenadores

IoT e objetos inteligentes. Já a outra ramificação é a de redes públicas, que consistem em Controladores IoT, Serviços IoT, *Gateways* IoT e *Cloud*. Um subconjunto dessas superfícies de ataque é herdado do paradigma *Cloud Computing*, por causa dos serviços *Cloud* da IoT. De acordo com Hossain, Fotouhi e Hasan (2015), as superfícies de ataque são:

- Em redes proximais (locais):
 - Aparelho ↔ Aparelho
 - Aparelho ↔ Coordenador
 - Coordenador ↔ *Gateway*
 - Aparelho ↔ Controlador

- Em redes públicas:
 - Controlador ↔ Provedor de Serviço IoT
 - Serviço ↔ Serviço

3 TRABALHOS RELACIONADOS

3.1 REVISÃO BIBLIOGRÁFICA SISTEMÁTICA

Com o objetivo de salientar a necessidade e importância da pesquisa deste trabalho, uma revisão sistemática foi desenvolvida, utilizando três ferramentas de pesquisa acadêmica, sendo estas o Google Scholar, o IEEE Xplore™ e o ScienceDirect. As palavras-chave (em Inglês e sem aspas) foram pesquisadas nestas ferramentas tanto separadas quanto agrupadas, conforme mostrado na Tabela 3, que apresenta o número de ocorrências destas em cada ferramenta. Os artigos pesquisados foram limitados para apenas aqueles publicados após o ano 2015.

Tabela 3 – Resultados da revisão sistemática bibliográfica (atualizados em 30 de Abril de 2020)

Palavras-chave	Google Scholar	IEEE Xplore™	ScienceDirect
AM	39 800	817	39 996
IoT	241 000	31 415	14 811
FC	31 200	3 341	4 769
WM	993 000	4 067	280 673
SH	71 500	6 822	27 202
AM + IoT	9 520	73	1 214
AM + FC	3 510	13	329
AM + WM	18 600	10	9 810
AM + SH	16 600	34	2 343
IoT + FC	17 000	1 339	1 362
IoT + WM	23 200	328	2 595
IoT + SH	16 800	1 973	3 969
FC + WM	14 100	16	932
FC + SH	16 200	88	749
WM + SH	19 800	168	7 271
AM + IoT + FC	4 480	5	214
AM + IoT + WM	1 870	4	317
AM + IoT + SH	4 750	9	526
AM + FC + WM	1 290	0	101
AM + FC + SH	2 150	1	135
AM + WM + SH	5 760	3	885
IoT + FC + WM	6 210	8	266
IoT + FC + SH	13 800	63	617
IoT + WM + SH	17 100	25	1 075
FC + WM + SH	9 230	2	202
AM + IoT + FC + WM	1 530	0	53
AM + IoT + FC + SH	2 150	0	123
AM + IoT + WM + SH	7 350	2	192
AM + FC + WM + SH	714	0	45
IoT + FC + WM + SH	14 500	2	167
AM + IoT + FC + WM + SH	714	0	41

As palavras foram identificadas pelas seguintes siglas:

1. AM - *Autonomic Management*
2. IoT - *Internet of Things* (pesquisada como “*iot*”)
3. FC - *Fog Computing*
4. WM - *Water management*
5. SH - *Smart homes*

Os dados acima ressaltam uma clara falta de artigos nos últimos 5 anos que tratem de *Fog Computing*, em especial nos casos envolvendo gestão autônoma de sistemas (palavra-chave *Autonomic Management*) ou gestão de água (palavra-chave *Water Management*). Nos resultados da ferramenta IEEE Xplore™, quando as três palavras-chave foram pesquisadas juntas, as mesmas não retornaram qualquer resultado, reforçando a necessidade de novas pesquisas envolvendo tais assuntos.

3.2 ANÁLISE DE CONSUMO DE ÁGUA EM ÁREAS URBANAS

Ji et al. (2015) acreditam que os padrões consumo de água de pessoas em áreas urbanas depende muito da estrutura da família, do tipo de trabalho do indivíduo ou do seu estilo de vida. A forma tradicional de medir o consumo de água de uma residência é pela leitura manual de um medidor analógico, que oferece uma precisão de $0,1\text{m}^3$, uma vez por mês. Com os avanços da IoT, a proliferação dos medidores inteligentes cresceu, os quais oferecem uma precisão de medição de $0,001\text{ m}^3$ a cada 15 minutos.

Algumas formas de coletar dados para uma análise parecida dependem de questionários, os quais não garantem integridade, precisão e representatividade, ou dependem de medições coletadas manualmente numa escala de tempo mensal ou anual, ou numa escala de distrito ou cidade, as quais não são precisas o suficiente. Os medidores inteligentes oferecem medições com alta frequência e precisão para uma análise com uma escala de tempo (e espaço) menor.

Para tirar proveito dos dados coletados pelos *Smart Meters*, Ji et al. (2015) propõem um algoritmo de agrupamento não supervisionado para analisar o consumo de água de residências em áreas urbanas. Esse algoritmo é adaptável na escala de tempo diário e pode dividir os residentes por estrutura familiar, tipo de trabalho ou estilo de vida. A pesquisa foi realizada com dados experimentais do volume de consumo de 20 usuários, coletando dados a cada 15 minutos com uma precisão de $0,001\text{m}^3$. A Tabela 4 exhibe o agrupamento de horários em 8 períodos para a realização de medidas individuais.

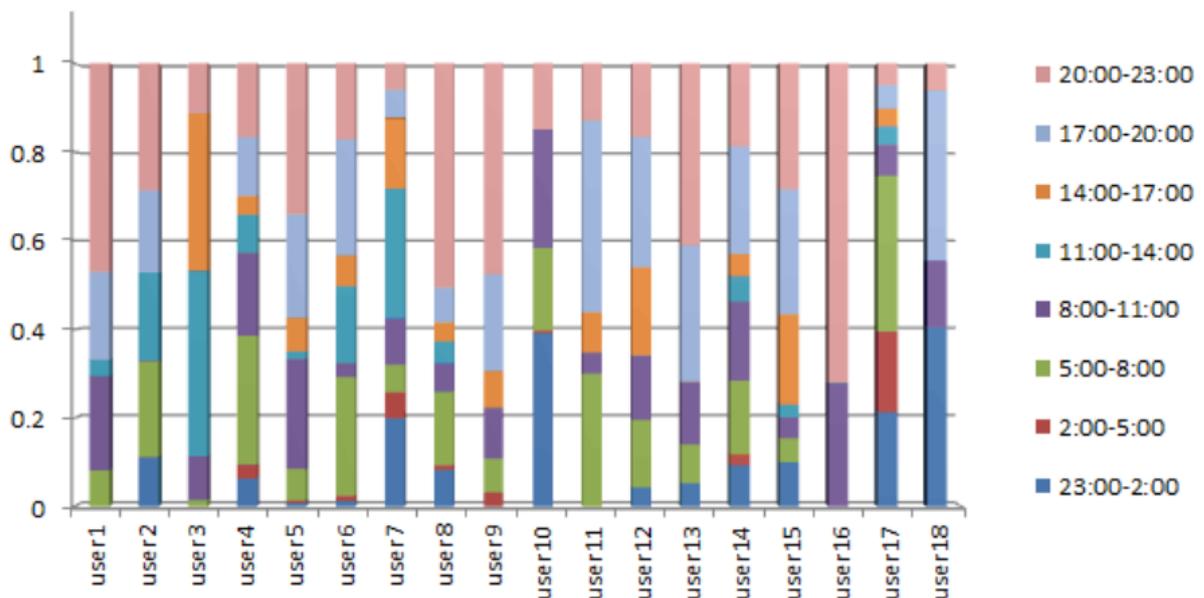
Usuários com média de consumo diário igual a zero foram descartados (2 nesta pesquisa). A análise dos dados ignora os intervalos individuais e os contabiliza como porcentagens do dia inteiro. Para calcular estes valores, os mesmos são comprimidos à escala de 0 a 1 em uma

Tabela 4 – Intervalos de tempo de Ji et al. (2015)

Intervalo de tempo	Tipo
23:00 - 2:00	Meia noite
2:00 - 5:00	Madrugada
5:00 - 8:00	Manhã
8:00 - 11:00	Horário de trabalho
11:00 - 14:00	Horário de almoço
14:00 - 17:00	Horário de trabalho (tarde)
17:00 - 20:00	Horário de jantar
20:00 - 23:00	Noite

matriz fuzzy. O resultado dos cálculos facilita a criação de grafos, como o da Figura 3, para visualizar o consumo de água de cada usuário em porcentagens ao longo de 24 horas, permitindo estudar o significado dos mesmos para quaisquer fins.

Figura 3 – Proporção de consumo de água de 18 usuários durante 24 horas



Fonte – (JI et al., 2015)

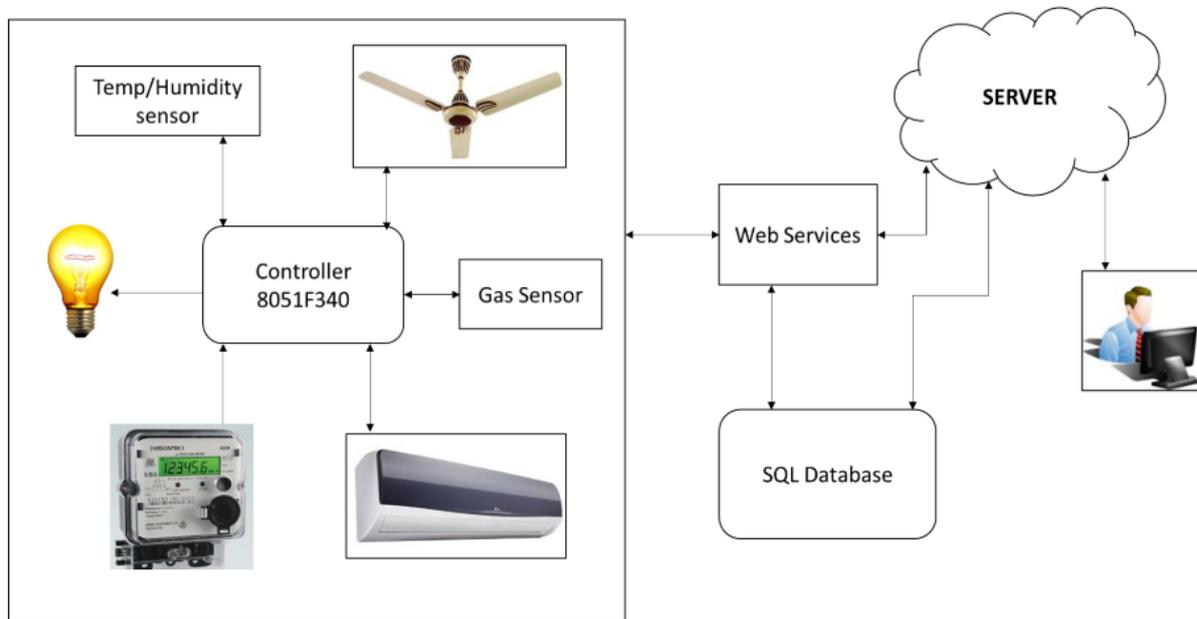
3.3 SMART HOME COM MEDIÇÃO ELETRÔNICA EM TEMPO REAL

Bhilar e Mali (2016) focam sua atenção no desenvolvimento de um sistema *web* que será utilizado para controlar aparelhos e sensores IoT. A ideia principal é facilitar o acesso à informação de medidores inteligentes e outros dados para os usuários finais, oferecendo conforto, eficiência e ajudando a economizar energia.

Eles propõem um sistema distribuído de controle, através do uso de uma aplicação e páginas *web*. Este sistema controla todos os recursos de cobrança da aplicação *web* funcionando

como uma aplicação *Cloud*, além de permitir os usuários a controlar seus eletrodomésticos de qualquer lugar, bem como verificar o gasto elétrico dos mesmos.

Figura 4 – Diagrama Básico de Blocos



Fonte – (BHILARE; MALI, 2016)

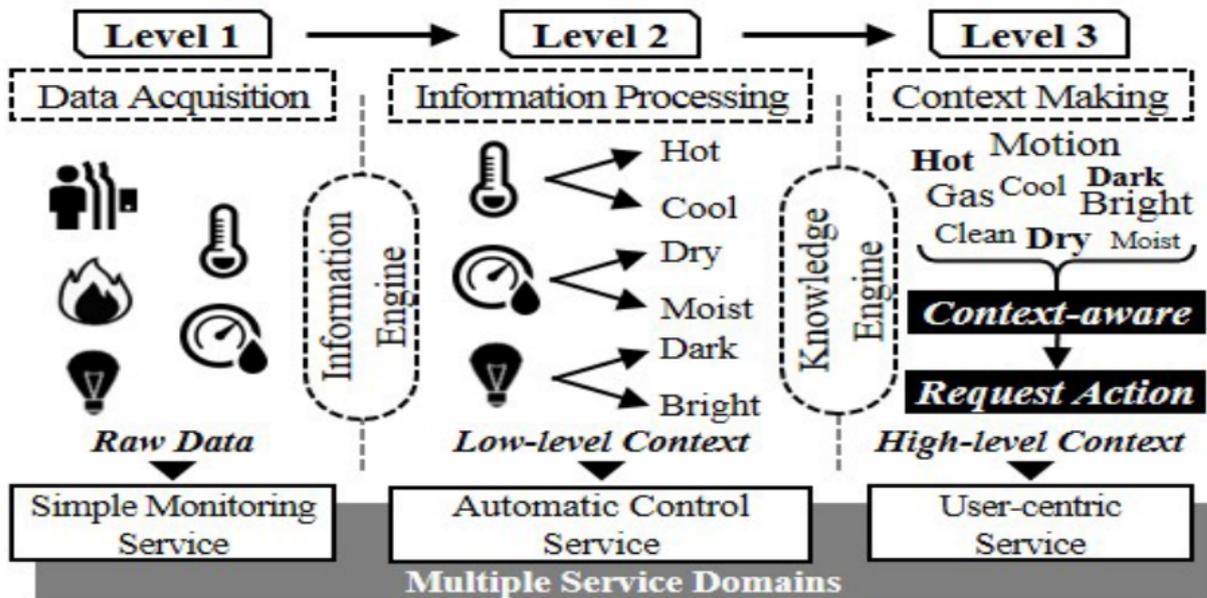
A Figura 4 apresenta o sistema em forma de diagrama básico de blocos, o qual funciona através de um microcontrolador C8051F340, que interage com vários sensores e aparelhos de um ambiente, além de possuir um servidor onde a aplicação *web* é executada. Ele é definido no manual de informações técnicas como um SMD com clock de 80MHz, até 64Kb de memória flash e até 4Kb de RAM.

O microcontrolador coleta dados de vários tipos de sensores, como o ACS712, que mede consumo elétrico, o DHT 11, que mede temperatura e umidade, além de controlar a velocidade de ventoinhas, e o MQ6, que pode identificar vazamento de gases. Todas essas informações são salvas em um database SQL na *Cloud*, permitindo que o usuário visualize os resultados ou seja alertado dos mesmos, através de notificações ou email.

O sistema num todo opera nos princípios do “Modelo de criação de contexto em três níveis”, como demonstrado na Figura 5. A aquisição de dados e processamento de sinais são o baixo nível de contexto, sendo que o contexto de alto nível é usado para gerar consciência da situação. Os sensores são os transdutores que detectam as características do ambiente. O fluxo do sistema pode ser descrito nos seguintes passos¹:

¹ Adaptados de uma imagem no artigo original

Figura 5 – Modelo de criação de contexto em três níveis



Fonte – (BHILARE; MALI, 2016)

1. Começo
2. Sensor coleta os dados com o estado atual dos aparelhos
3. Dados são enviados ao controlador
4. Controlador manda os dados para o servidor *web* através do protocolo HTTP
5. Dados são armazenados na *database SQL*
6. Dados (junto do estado dos aparelhos) é mostrado através de interface *web*
7. Caso o estado não esteja atualizado, volta-se ao passo 2
8. Os valores atualizados são enviados para a *database*, e a mesma é atualizada
9. Os valores atualizados são então enviados para o controlador através do protocolo HTTP
10. O controlador recebe o comando e realiza a ação de controlar os aparelhos
11. Fim

A interface da aplicação *web* é utilizada para controlar e monitorar os dados. O usuário pode encontrar quaisquer avisos urgentes na seção de alertas da página principal. Também existem links para os *websites* de vários provedores de serviços, com os detalhes de suas taxas de consumo de eletricidade. O usuário deve logar com senha para poder monitorar e controlar seus eletrodomésticos.

Figura 6 – Homepage da aplicação web



Fonte – (BHILARE; MALI, 2016)

3.4 GERENCIAMENTO DE ÁGUA BASEADO EM IOT

Rajurkar, Prabakaran e Muthulakshmi (2017) apresentam um projeto onde o foco é monitorar somente o uso de água. A metodologia está focada no uso de um dispositivo IoT, composto por um Arduíno e um sensor de fluxo de água, o qual irá medir o consumo de água em litros por hora. As medições obtidas são enviadas a partir do dispositivo IoT para um servidor na *Cloud*, onde as informações serão disponibilizadas para aplicativos *mobile*, com o objetivo de manter um usuário ou uma empresa informados caso ultrapassem o limite de uso de água (e quais os custos adicionais caso isso ocorra).

Os autores mencionam uma frase do especialista sueco Falken Mark, que diz “Quando a disponibilidade de água é inferior a um metro cúbico por pessoa por dia, ocorre estresse hídrico. O culminar de uma população enorme e crescente e as demandas cada vez maiores por água e o acesso desigual a ela são a principal causa da escassez de água.”. Fora o uso agrícola, a maior parte da água utilizada pelos humanos é destinada às indústrias.

O projeto visa monitorar o consumo de água individual por casa, enviando os dados para um servidor na *Cloud*. Tais dados podem ser acessados posteriormente através de aplicação móvel, a qual ajudará o usuário a verificar se o mesmo ultrapassou o limite de consumo mensal de água, além de informar o custo extra. Rajurkar, Prabakaran e Muthulakshmi (2017) acreditam que, quando os recursos hídricos forem cobrados de forma adequada, incluindo os usos em excesso, as pessoas não irão tirar mais vantagem da água.

O hardware implementado depende principalmente de um Arduíno, que mede o consumo de água a partir de um sensor de fluxo de água que fica situado no cano principal da casa (também é possível utilizar um solenoide). O sensor utilizado é o YF-S201 (Figura 7),

Figura 7 – Medidor de fluxo de água YF-S201



Fonte – (RAJURKAR; PRABAHARAN; MUTHULAKSHMI, 2017)

que funciona com o *Hall Effect*², possui um rotor em forma de pequeno ventilador/hélice, que fica no caminho do fluxo da água. O sensor produz 450 pulsos³ por litro de água passando a cada minuto, e o Arduino converte o número de pulsos em litros por hora. As Figuras 8 e 9 mostram, respectivamente, como conectar o sensor ao Arduino diretamente, bem como a leitura simultânea de 3 sensores.

Figura 8 – Conexão de um sensor YF-S201 com o Arduino



Fonte – (RAJURKAR; PRABAHARAN; MUTHULAKSHMI, 2017)

² É a produção de uma diferença de tensão através de um condutor elétrico, transversal a uma corrente elétrica no condutor e a um campo magnético aplicado perpendicular à corrente.

³ Medida específica do sensor YF-S201.

3.5 GERENCIAMENTO DE ENERGIA BASEADO EM EDGE PARA SMART HOMES

De acordo com Xia et al. (2018), a maioria dos sistemas de gerenciamento de energia residencial (*Home Energy Management System*, ou HEMS) existentes são projetados principalmente para economizar energia e satisfazer o conforto dos clientes, controlando aparelhos inteligentes remotamente, por meio de um método de reconhecimento de local, e alterando de forma inteligente os pontos de ajuste de aquecimento, ventilação e ar condicionado (*Heating, Ventilation and Air Conditioning*, ou HVAC).

Os sistemas HEMS existentes podem ser divididos principalmente em três categorias: tecnologias de habilitação, interfaces gráficas de usuário e dispositivos de controle. Para as tecnologias de habilitação, o sensor e a comunicação são o foco principal. Com a ajuda de medidores inteligentes, sensores de temperatura e sensores de ocupação, as variáveis dinâmicas no ambiente doméstico podem ser coletadas. Para as interfaces gráficas de usuário, seu objetivo é fornecer uma interface *web* ou um aplicativo, para permitir que os usuários tomem conhecimento e monitorem seu uso de energia. Os dispositivos de controle são servidos como portadores da operação do sistema de gerenciamento de energia, ou em aparelho ou no nível doméstico. Esses dispositivos de controle geralmente são executados em PCs de chip único, como Raspberry Pi e Arduíno.

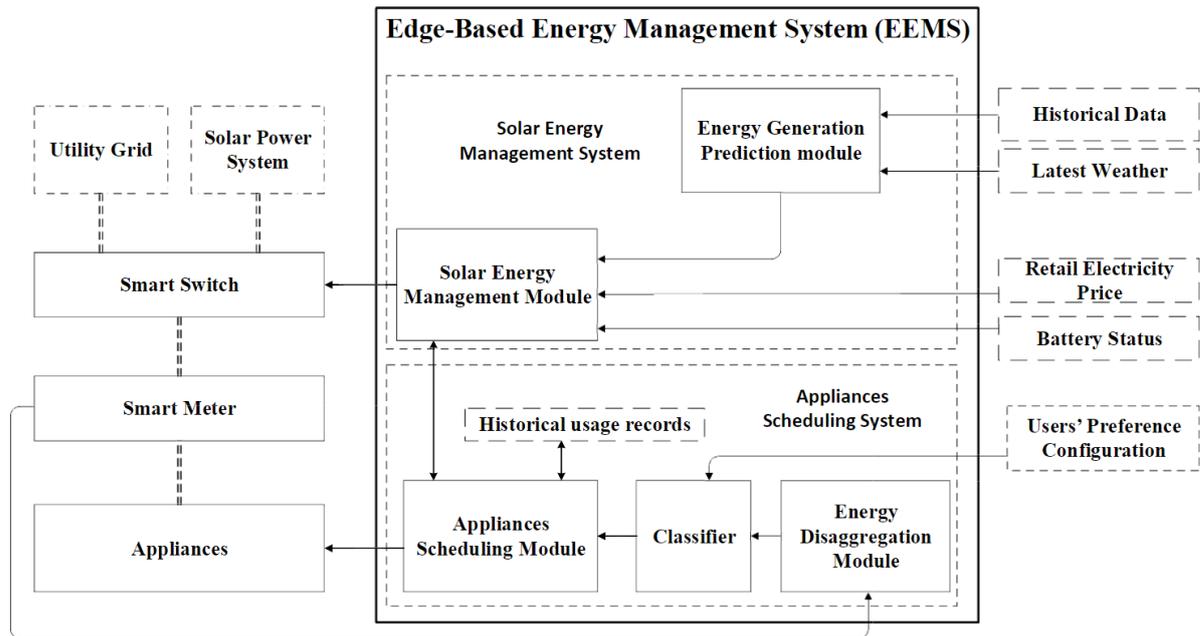
A motivação para o artigo é propor um *framework* de gerenciamento de energia baseada em *Edge*, que permita baixo custo de eletricidade, gerando estratégias de programação ideais para os aparelhos domésticos, e maximizando a utilização de energia solar em comunidades residenciais. Além disso, dispositivos de *Edge* fáceis de obter podem fornecer recursos de computação suficientes para agendamento de dispositivos e tomada de decisões na implantação de energia, basicamente diminuindo o custo de investimento de capital em grande medida.

Existem várias técnicas de resposta à demanda (*Demand Response*, ou DR), e Xia et al. (2018) aprofundam-se principalmente na implementação de *Load Shifting*, pelo qual sistemas sem restrição de tempo podem alterar o funcionamento de dispositivos para fora do horário de pico, de forma a reduzir a demanda de uma casa, sem comprometer o conforto dos usuários.

A técnica de monitoramento de carga de dispositivo (*Appliance Load Monitoring*, ou ALM) é uma ferramenta valiosa para a construção de sistemas domésticos de gerenciamento de energia. Ela permite que os usuários obtenham estatísticas de consumo de energia específicas do equipamento, as quais podem ser usadas para conceber estratégias de agendamento de carga para utilização otimizada da energia. As abordagens do ALM podem ser divididas em dois tipos: Monitoramento de carga intrusiva (*Intrusive Load Monitoring*, ou ILM) e Monitoramento de carga não intrusiva (*Non-Intrusive Load Monitoring*, ou NILM). Para o ILM, vários dispositivos e sensores de medição são implantados para obter o monitoramento de energia em cada dispositivo de maneira distribuída, enquanto que as técnicas NILM conduzem a desagregação de energia apenas com base na leitura geral do consumo de energia de um único medidor elétrico.

Ao adotar o sistema EEMS, mostrado na Figura 11, o tempo de operação dos eletro-

Figura 11 – *Framework do Edge-Based Energy Management System (EEMS)*



Fonte – (XIA et al., 2018)

domésticos passa a ser automaticamente agendado com base nas preferências dos usuários, nos preços de eletricidade do dia seguinte e em históricos de uso, gerados com o tempo. O EEMS possui dois subsistemas: o sistema de gerenciamento de energia solar, que foca em maximizar o uso de eletricidade produzida por painéis solares, e o sistema de agendamento de aparelhos, que evita gastos de energia muito altos durante os horários de pico, de forma a evitar altos custos.

O sistema no geral é composto por alguns componentes:

- O módulo *energy generation prediction*, nos estágios iniciais, carrega dados de localização e previsão do tempo, com o intuito de maximizar a produção de energia de painéis fotovoltaicos
- O módulo *solar energy management*, que controla o uso da energia solar, maximizando seu uso quando a geração da mesma é maior que a demanda de carga, além de guardar todo o excesso em baterias
- O módulo *energy disaggregation*, que monitora e, utilizando uma abordagem NILM sensível à localização, deduz as atividades do usuário ao longo de um dia típico
- O módulo *classifier*, que classifica a carga dos eletrodomésticos inteligentes em agendáveis e não agendáveis. Para descrever flexibilidade quantitativamente, um coeficiente é automaticamente gerado para cada eletrodoméstico com base nos dados do módulo *energy disaggregation*, além das preferências pessoais do usuário
- O módulo *appliances scheduling*, que agenda os horários de início dos eletrodomésticos agendáveis de acordo com todas as informações coletadas pelo módulo *classifier*

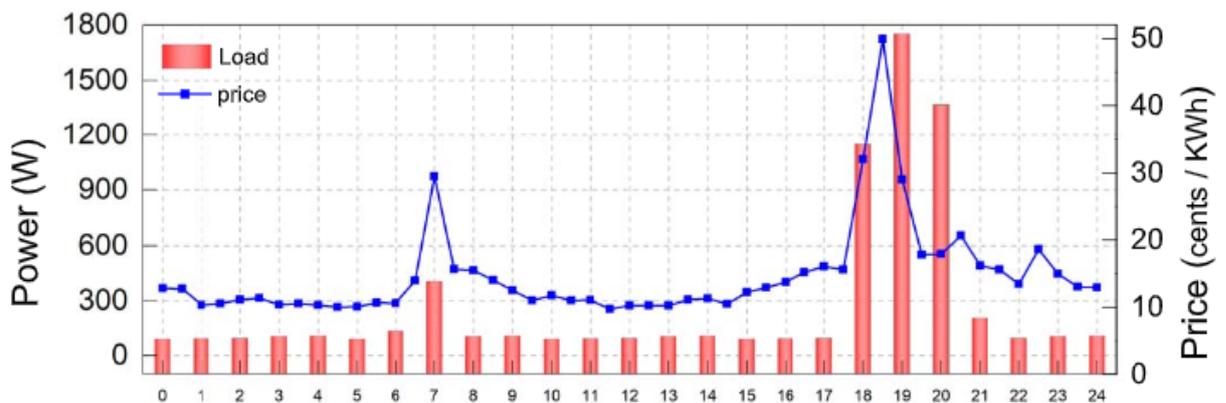
Figura 12 – Ambiente de testes para o EEMS



Fonte – (XIA et al., 2018)

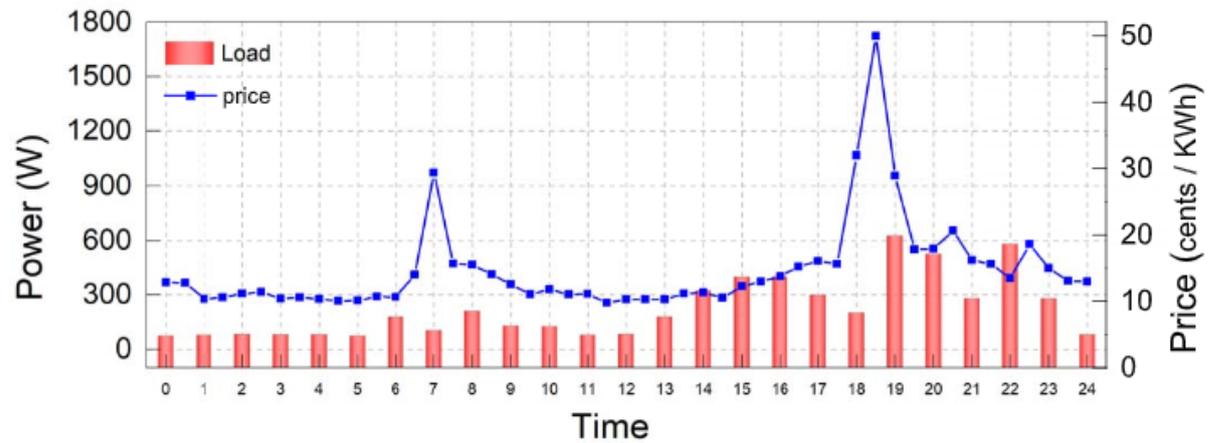
Para verificar a viabilidade do EEMS, Xia et al. (2018) construíram um ambiente de testes real com geradores de energia solar em pequena escala e implementaram a estrutura proposta e o algoritmo de agendamento, para gerenciar o funcionamento de eletrodomésticos como uma geladeira, ar condicionado, TV e máquina de lavar roupa, como mostrado na Figura 12. Os resultados podem ser observados nos gráficos das Figuras 13 e 14, sendo apresentado na Figura 13 o consumo elétrico de acordo com as preferências dos usuários, e na Figura 14, de acordo com o algoritmo implementado. Essas Figuras indicam o consumo diário com barras vermelhas e o custo (centavos por kWh) com uma linha azul. Observa-se na Figura 13 que os horários de maior consumo de energia ocorrem entre as 18 e as 20 horas, onde o preço varia de 15 a 50 centavos por kWh. Também podemos verificar que existem dois períodos de alto custo de energia, sendo estes entre as 6 e 8 horas e entre as 17 e 20 horas. A Figura 15 mostra os dois resultados anteriores comparados entre si. Por fim, a Figura 16 mostra o consumo de energia da rede comparado com a produção de energia solar, onde os períodos de alta produção dessa energia reduzem o consumo da rede para zero.

Figura 13 – Resultados dos testes (preferências dos usuários)



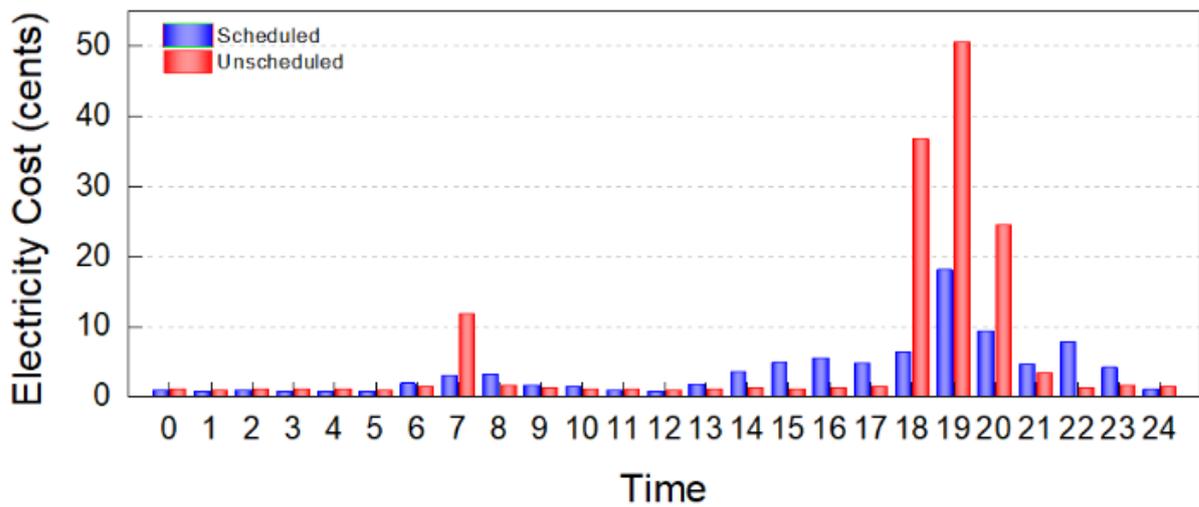
Fonte – (XIA et al., 2018)

Figura 14 – Resultados dos testes (preferências do algoritmo)



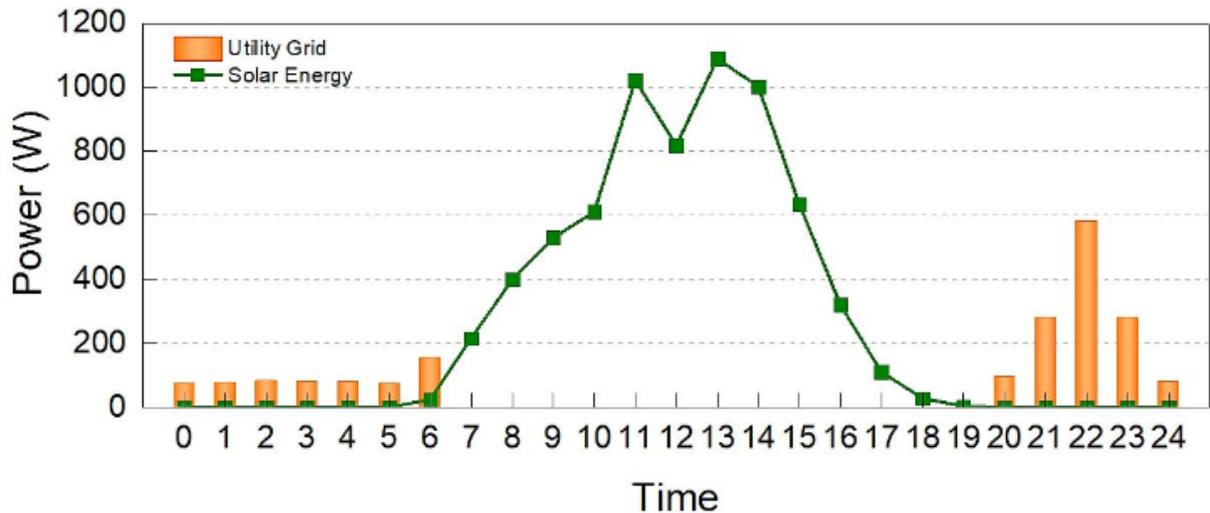
Fonte – (XIA et al., 2018)

Figura 15 – Comparação dos resultados dos testes (usuários em vermelho, algoritmo em azul)



Fonte – (XIA et al., 2018)

Figura 16 – Consumo de energia da rede e produção de energia solar



Fonte – (XIA et al., 2018)

3.6 SISTEMA AUTÔNOMO PARA MONITORAMENTO DE CONSUMO DE ÁGUA EM APARTAMENTOS

Os autores Aggarwal, Chauhan e Prakash (2019) acreditam que o custo de instalação de hidrômetros separados é muito alto. Em seu artigo, eles focam na criação de um dispositivo inteligente de baixo custo, o qual pode ser usado para coletar os dados de consumo de água. Tal dispositivo foi criado com o Arduino, juntamente com alguns sensores e módulos para coletar vários dados de consumo de água de diferentes apartamentos.

O sistema de monitoramento automatizado de baixo custo poderá rastrear e monitorar automaticamente o consumo de água de diferentes apartamentos em um prédio, e exigirá a instalação de um dispositivo que inclua sensores com localização geográfica, para que ele possa rastrear dados diferentes em diferentes locais do edifício, com fluxo contínuo de dados e análises, que podem ajudar no monitoramento e no controle de todo o processo com mais eficiência. Um dispositivo fabricado com Arduino e sensores para coletar esses dados enviar ao servidor para análise, com a ajuda de uma rede neural artificial, poderá gerar resultados com uma precisão muito alta. O sistema enviará notificações de consumo excessivo de água aos usuários com a ajuda do aplicativo *Web* e *Android*.

Este sistema ajudará proprietários de edifícios a gerarem contas de consumo de água, e também ajudará os moradores a reduzirem suas contas, pois poderão conhecer seu consumo diário e por hora de água, o que os ajudará a analisar quando e o quanto de água eles podem salvar.

Em um trabalho correlato, Aggarwal, Chauhan e Prakash (2019) afirmam que o autor propôs desenvolver um modelo no qual eles criaram um sistema para otimizar o consumo de água em uma cidade, que pode ser usado para fornecer suprimento de água 24/7. O sistema decide dinamicamente a cota para cada entidade com base na oferta e demanda. Esse sistema

foi bem eficiente para as cidades, mas seu modelo não pode ser usado em uma escala menor (como a de um edifício).

Outros autores apresentaram um modelo diferente para prever o consumo de água nas cidades urbanas. O principal problema nesses modelos era a falta da análise de dados em tempo real para gerar resultados instantâneos com alta precisão. Além disso, outro problema identificado foi que eles não incluem um sistema de segurança. Não há sistema ou mecanismo que possa ser usado para notificar o usuário quando o dispositivo falhar ao executar qualquer tarefa.

Um dispositivo baseado em IoT foi utilizado para coletar todos os dados em diferentes apartamentos de um edifício. O modelo de Rede Neural Artificial (*Artificial Neural Network*, ou ANN) foi utilizado para analisar os dados coletados. Um aplicativo da *Web* e um aplicativo do Android agem como interface para aqueles que desejam verificar os dados coletados pelo dispositivo IoT com melhor usabilidade.

O dispositivo baseado na IoT foi usado para o envio contínuo de dados ao servidor, para que os dados em tempo real pudessem ser usados para análise pela Rede Neural Artificial. A implementação está usando um Arduíno Mega para controlar muitos sensores ao mesmo tempo. Os autores afirmam que este microcontrolador é três vezes mais barato que outros microcontroladores, como o Raspberry Pie, portanto, o Arduíno Mega é acessível para pessoas comuns e também está disponível em vários modelos.

O dispositivo usará um painel solar com uma bateria para alimentá-lo, para que ele possa funcionar de forma independente. Além do dispositivo, estão sendo utilizados sensores como sensor de fluxo, sensor de umidade, sensor de temperatura (DS18B20), relógio em tempo real (DS3231), módulo WiFi (ESP8266) e módulo Bluetooth, entre outros. A razão por trás do uso do módulo WiFi e Bluetooth é que eles reduzirão o custo da fiação, pois o dispositivo central poderá monitorar o funcionamento de todos os outros dispositivos remotamente e será usado para acionar o sistema de alarme.

Nesse sistema, o dispositivo coletará dados de consumo de água por hora, diariamente, mensalmente e o total de diferentes apartamentos, o que ajudará o usuário a analisar onde a água é desperdiçada e como eles podem reduzir seu consumo, o que os ajudará a reduzir sua conta.

O modelo de Rede Neural Artificial foi usado para a análise e geração de resultados com base nos dados coletados pelo dispositivo na última seção. Ele é capaz de gerar resultados com muita precisão e pode analisar e processar uma grande quantidade de dados muito rapidamente, o que é muito importante para todos os tipos de sistemas que geram resultados instantâneos.

Este modelo foi usado para descobrir todos os padrões de consumo de água, que serão realizados pelo treinamento e teste dos dados coletados pelo dispositivo no servidor. Depois disso, esse modelo poderá identificar todos esses padrões no futuro e enviará uma notificação de consumo excessivo aos usuários com a ajuda do aplicativo *Web* e do aplicativo Android, mencionados na próxima seção. A Tabela 5 possui uma amostra do conjunto de dados coletados

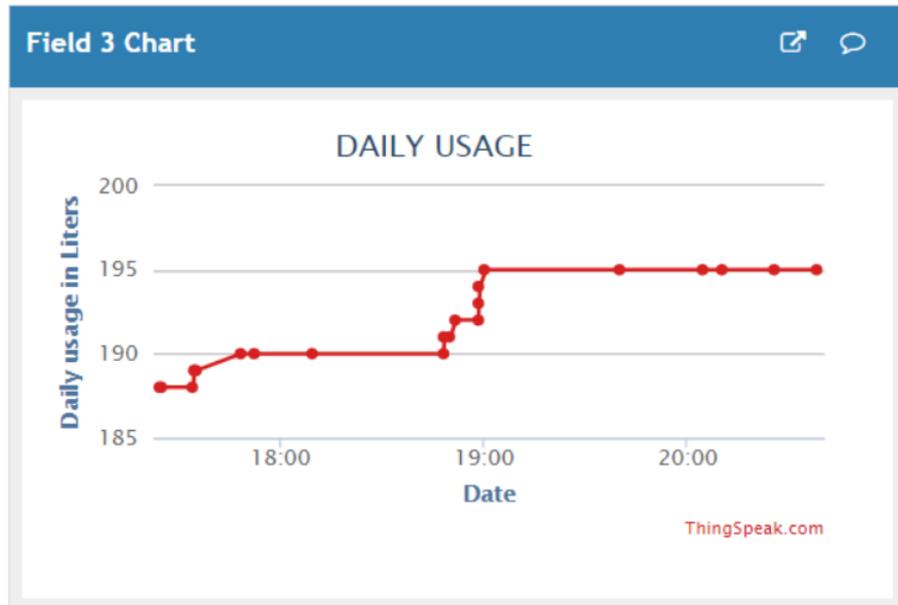
pelo dispositivo em um apartamento, que será analisado pelo modelo de Rede Neural Artificial.

Tabela 5 – Exemplo de dados de consumo de água por litro de Aggarwal, Chauhan e Prakash (2019)

Criado em	ID de entrada	Total	Por Hora	Diário	Mensal
2017-11-14 10:16:38	28216	9334	12	168	174
2017-11-14 10:16:56	28217	9335	13	169	175
2017-11-14 10:17:12	28218	9336	14	170	176
2017-11-14 10:17:27	28219	9337	15	171	177
2017-11-14 10:17:42	28220	9338	16	172	178
2017-11-14 10:17:58	28221	9338	17	173	179
2017-11-14 10:18:14	28222	9339	18	174	180
2017-11-14 10:18:31	28223	9340	19	175	181
2017-11-14 10:19:05	28224	9341	19	176	181
2017-11-14 10:19:36	28225	9341	20	176	182
2017-11-14 10:19:59	28226	9342	20	177	182
2017-11-14 10:24:28	28227	9342	21	177	183
2017-11-14 10:25:11	28228	9343	21	178	184
2017-11-14 10:25:27	28229	9343	22	178	184
2017-11-14 10:25:44	28230	9345	23	179	185
2017-11-14 10:30:43	28231	9345	24	180	186
2017-11-14 10:30:59	28232	9346	1	181	187

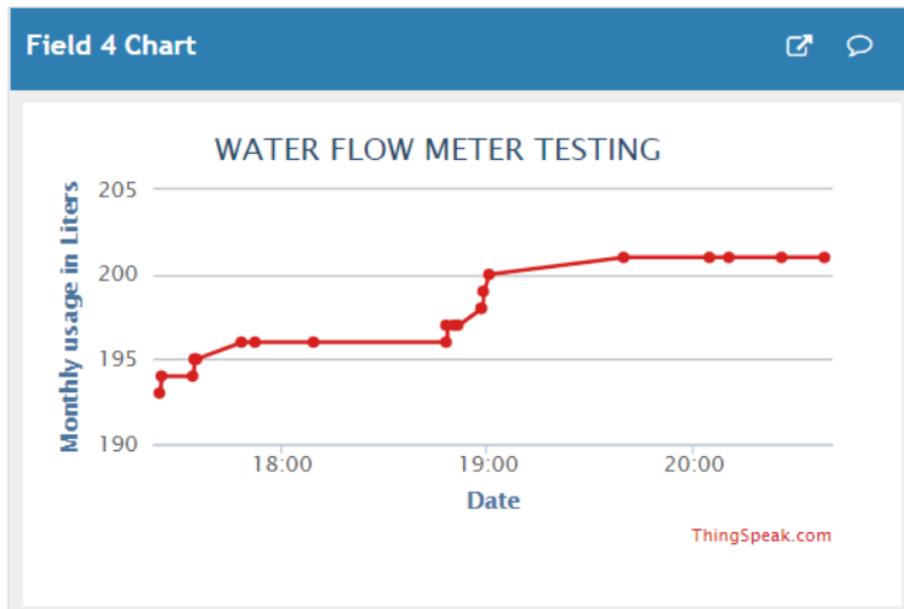
O aplicativo deve ser fácil de usar e estar facilmente disponível para todos. As tarefas deste aplicativo incluem a autenticação adequada de uma pessoa com a ajuda de um sistema OTP móvel, de modo que apenas uma pessoa autorizada possa acessar esses dados, o que fornecerá segurança para todo o sistema. O usuário poderá visualizar todos os dados (anteriores e atuais) coletados pelo dispositivo, o que ajudará a analisá-los facilmente e, com base nisso, o usuário poderá agir de forma a diminuir seu consumo de água, o que o ajudará a reduzir sua conta. Uma amostra dos dados coletados pelo dispositivo, que serão armazenados em um servidor *thingSpeak*, é mostrada na forma de gráficos no trabalho.

Figura 19 – Dados coletados com relação à coluna “Diário” da Tabela 5



Fonte – (AGGARWAL; CHAUHAN; PRAKASH, 2019)

Figura 20 – Dados coletados com relação à coluna “Mensal” da Tabela 5



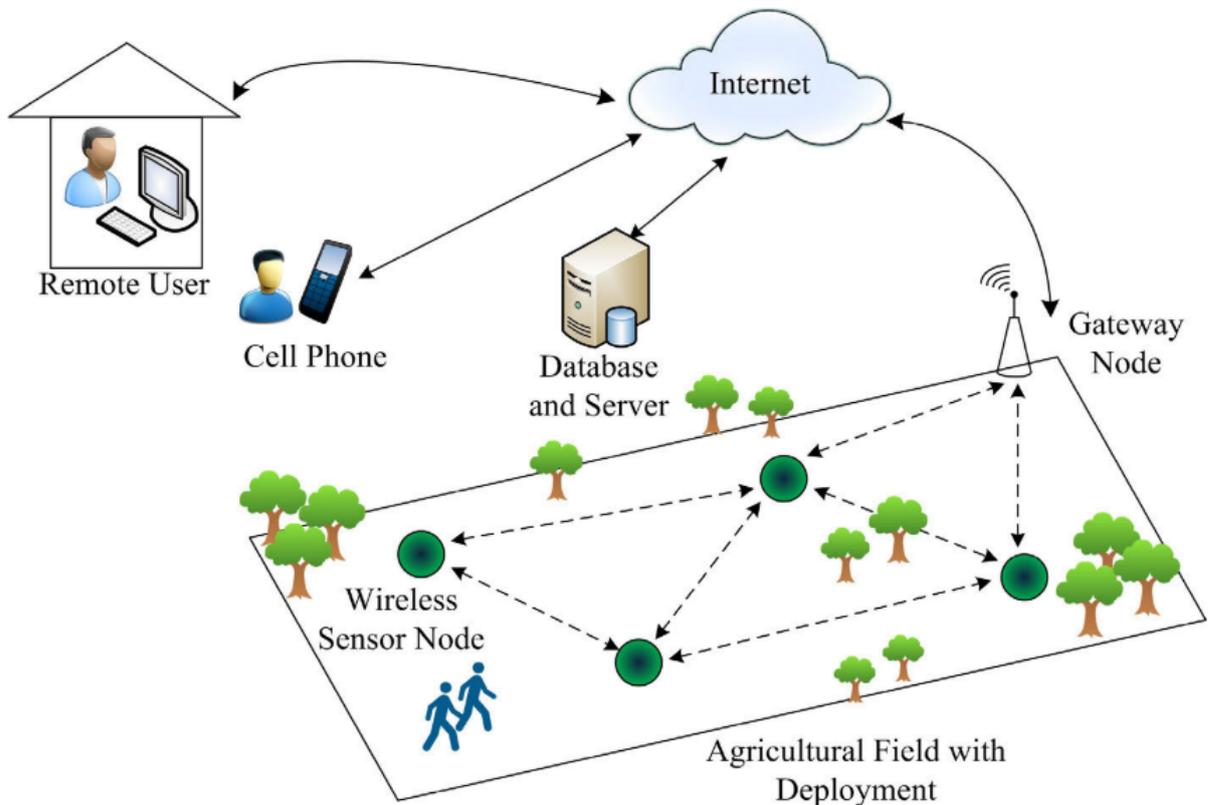
Fonte – (AGGARWAL; CHAUHAN; PRAKASH, 2019)

Para lidar com o problema de falhas no sistema, Aggarwal, Chauhan e Prakash (2019) propõem o uso de um sistema de alarme em todo o modelo que monitore o funcionamento do modelo principal e, se ocorrer algum problema, o modelo fechará o dispositivo principal para que não execute ações erradas. Além disso, esse modelo acionará um pequeno sistema de alarme para enviar notificações, que notificarão a pessoa autorizada, e as ações correspondentes serão tomadas. O dispositivo usará um RTC para armazenar data e hora, para que a pessoa que está verificando os dados no servidor possa saber quando e onde esse valor foi obtido pelo dispositivo. Também deve ajudar os usuários a saber quanta água consumiram nos últimos dias.

3.7 PESQUISA SOBRE DESAFIOS DO GERENCIAMENTO DE ÁGUA NA AGRICULTURA

Neste artigo, Saad, Benyamina e Gamatie (2020) apresentam uma pesquisa sobre o uso de tecnologias inovadoras para desenvolver um conceito inteligente na regulamentação do uso da água no domínio agrícola. Os mesmos começam explicando que existem vários métodos que ajudam a preservar e proteger as fontes de água, como a construção de barragens para armazenar água da chuva, a dessalinização da água do mar, o tratamento de águas residuais e o monitoramento de oleodutos para detectar danos ou vazamentos. Para aprimorar esses métodos e torná-los mais inteligentes, os autores mencionam os principais paradigmas de pesquisa que são utilizados, sendo estes os sistemas ciber-físicos (*Cyber-Physical Systems*, ou CPS), as redes de sensores sem fio (*Wireless Sensor Networks*, ou WSN, representado na Figura 21), a IoT e outras tecnologias em nuvem. Esses paradigmas entraram no setor agrícola como forma de criar um sistema automatizado e integrado. Sensores ambientais (por exemplo, sensores de umidade, pressão e temperatura) são implantados nas infraestruturas CPS e WSN. Esses sensores geram dados espaço-temporais maciços e heterogêneos, que são armazenados e processados em larga escala. Essas tecnologias também permitem reduzir o desperdício de água, pois é importante para a sustentabilidade geral de recursos limitados de água doce. A implementação de soluções inteligentes para o monitoramento da água oferece a possibilidade de aprimorar a produção agrícola e facilitar o gerenciamento, porque ter a capacidade de monitorar a água através de sensores fornece aos agricultores o poder de aumentar o crescimento das culturas. O uso desses sensores fornece informações precisas e em tempo real sobre os diferentes parâmetros relacionados à água, para que os agricultores possam intervir efetivamente no momento certo.

Figura 21 – Rede de sensores sem fio



Fonte – (SAAD; BENYAMINA; GAMATIE, 2020)

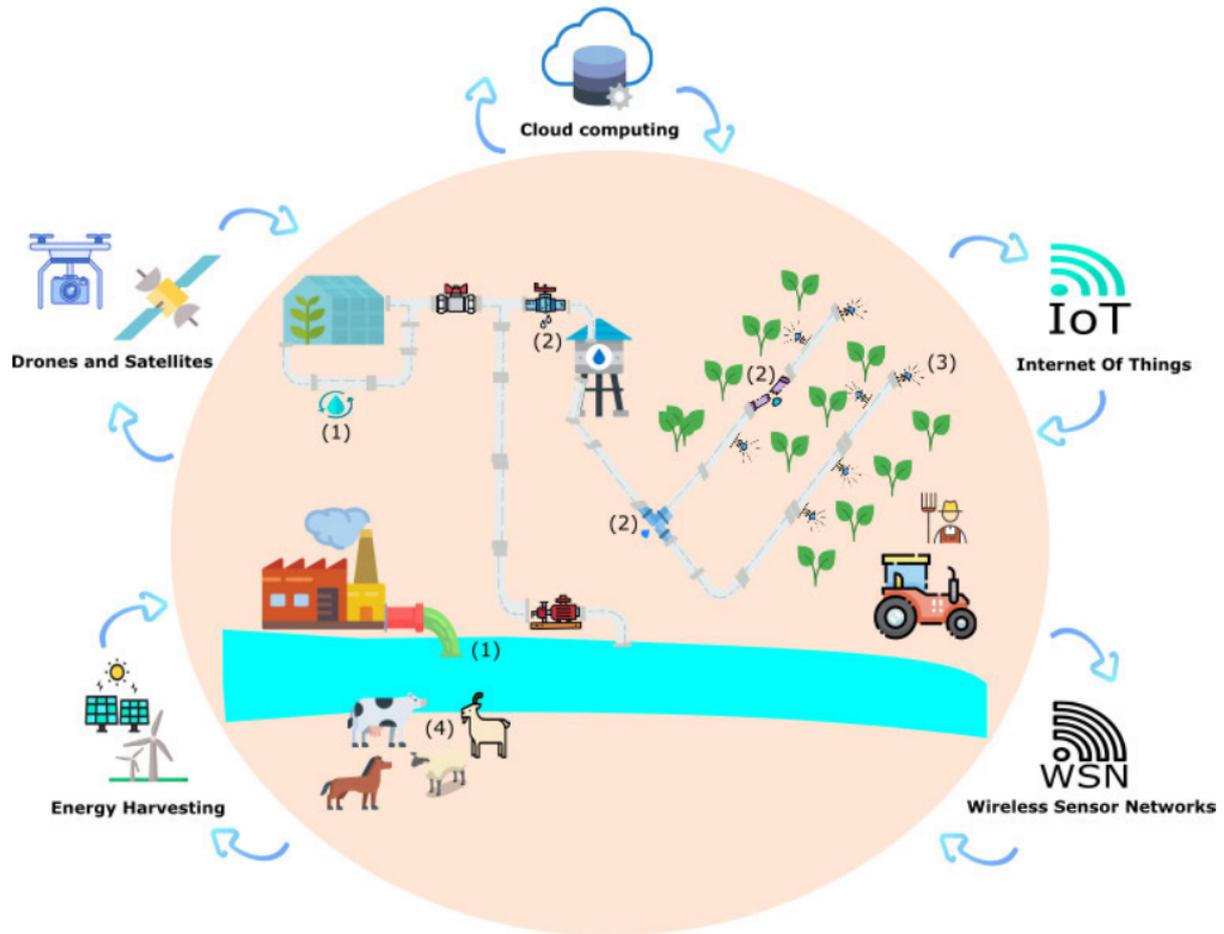
Os autores discutem a necessidade de preservar os recursos hídricos, o meio ambiente, e melhorar as lavouras usando tecnologias avançadas. Para esse fim, eles identificam quatro campos de aplicação na agricultura, apresentando desafios reais. Esses campos de aplicação foram cuidadosamente considerados para melhorar o gerenciamento da água como um fator *premium* para o desenvolvimento no domínio agrícola. Foram enfatizadas várias abordagens e métodos selecionados que aproveitam as tecnologias modernas para o gerenciamento inteligente da água.

Na Figura 22, os quatro principais desafios da gestão de água na agricultura moderna estão representados, sendo estes:

- A reutilização da água e monitoramento da poluição da água (representado por (1));
- O monitoramento de dutos de água (representado por (2));
- A irrigação por água (representado por (3)); e
- A disponibilidade de água potável para o gado (representado por (4)).

Dos quatro desafios apresentados, o monitoramento de dutos de água apresenta a maior utilidade para este trabalho. Este desafio está focado em problemas como a idade do tubo, a

Figura 22 – Quatro principais desafios da gestão de água na agricultura moderna



Fonte – (SAAD; BENYAMINA; GAMATIE, 2020)

sobreprensão, a instalação inadequada, o mau funcionamento do atuador mecânico (ou seja, válvulas, bombas, pulverizadores, etc) e desastres naturais. Estes são os fatores mais importantes que podem causar vazamentos e danos na rede de distribuição de dutos.

Um vazamento de água na rede de irrigação pode causar uma escassez na produtividade da produção agrícola, devido a uma quantidade insuficiente de água para o crescimento das culturas. Mecanismos de monitoramento e controle em tempo real ajudam a superar esses problemas relacionados à distribuição de água.

Com base nos paradigmas apresentados no primeiro parágrafo, os autores selecionaram alguns estudos para revisão de acordo com a aplicabilidade dessas tecnologias na gestão de água. A maior parte desses estudos trabalham com monitoramento, gestão e reutilização de água, e os considerados exploram dados de várias fontes:

- sensores no local (por exemplo, estações meteorológicas, dispositivos de detecção química, biossensores, sondas, etc.) fornecem coleta de dados;
- tecnologias de sensoriamento remoto (por exemplo, UAVs e satélites) oferecem a possi-

bilidade de coletar dados de mídia (ou seja, imagens e vídeos); e

- serviços online na *Web*, fornecendo dados valiosos, como informações sobre plantas e condições climáticas.

Saad, Benyamina e Gamatie (2020) também apresentam uma tabela com os parâmetros mais medidos nos desafios de monitoramento de água, replicada na Tabela 6.

Tabela 6 – Parâmetros mais medidos nos desafios de monitoramento de água

Parâmetro	Objetivo do uso	Grau de envolvimento
pH	O pH da solução mede o grau de acidez ou alcalinidade em relação à ionização da água	+++++
Potencial de oxidação e redução (ORP)	ORP é uma medida que indica o grau em que uma substância é capaz de oxidar ou reduzir outra substância	+++
Condutividade elétrica (EC)	Condutância é a capacidade de uma substância transmitir e conduzir uma corrente elétrica	++++
Temperatura	É usado para medir a temperatura do solo e da água	+++++
Umidade do solo	É uma medida do teor volumétrico real de água	+++
Contaminantes	Monóxido de carbono, dióxido de carbono, amônia, metano, etanol e hidrogênio	++++
Turbidez	É usado para medir a intensidade da luz dispersa a 90 graus quando um feixe de luz passa através de uma amostra de água	+++
Oxigênio dissolvido (DO)	É uma medida da quantidade de moléculas de oxigênio livre na água	++++
Teor de matéria orgânica	Mede nutrientes muito importantes do ponto de vista do manejo da fertilidade do solo, como pequenas quantidades de enxofre, nitrogênio, fósforo, potássio, cálcio e magnésio	++++
Nível de água da pia	Determina a quantidade de água na pia	+++
Pressão	Refere-se à quantidade de força que a água exerce ao sair do tubo	++
Parâmetros do tempo (<i>Weather</i>)	Os sensores são usados para medir a velocidade e orientação do vento, chuva, luz do sol, temperatura do ar, umidade, etc	++
Obstáculos	GPS e câmera são instalados em instalações autônomas para realizar atividades agrícolas	++
Fluxo de água	Refere-se à quantidade de água que sai do tubo em um determinado período de tempo	+++
Cloro	Mede a quantidade de desinfetante na água	+++

Na seção IV do artigo, os autores listam vários outros trabalhos que resultaram do artigo *survey* realizado, os quais apresentam possíveis soluções para cada um dos quatro desafios mencionados anteriormente. Focando novamente no desafio de monitoramento de dutos de água, Saad, Benyamina e Gamatie (2020) apresentam 7 trabalhos, dentre os quais está um trabalho já mencionado pelos autores em outra seção, intitulado “*SmartPipes: Smart wireless sensor networks for leak detection in water pipelines*”, onde é proposto um protótipo de hardware para detecção de vazamento de água com baixo consumo de energia. A principal preocupação do protótipo é permitir uma instalação fácil, sem influenciar o desempenho do tubo, usando um método de detecção relativa baseado em resistores sensíveis à força (*Force-Sensitive Resistors*, ou FSR) para derivar medições de pressão no WSN subterrâneo proposto para monitoramento

de tubulação. Uma bancada de testes foi desenvolvida e o protótipo foi testado dentro de um laboratório.

Na penúltima seção do artigo, são apresentadas diversas tabelas com as principais características dos trabalhos apresentados para cada problema. A tabela comparativa dos trabalhos relacionados ao monitoramento de dutos de água foi recriada na Tabela 7 e na Tabela 8.

Tabela 7 – Trabalhos relacionados ao monitoramento de dutos de água (Parte 1)

Trabalhos correlatos	Abordagem de validação	Interno/ Externo ao duto	Tecnologias	Aparelhos de baixo custo	Monitoramento em tempo real	Interface de transmissão de mensagens
Sadeghioon et al. [17]	Protótipo de hardware (curta distância)	Externo	WSN subterrânea	Sim	Sim	433Mhz RF
Karray et al. [75]	Modelo matemático híbrido e protótipo de hardware	Externo	WSN	Sim	Sim	Bluetooth
McDougle et al. [50]	Protótipo de hardware e modelo analítico	Externo	IoT	Conjunto fornecido pela Veolia	Sim	GSM ou sinais de rádio
Natividad e Palaoag [76]	Protótipo de hardware	Interno	IoT, Cloud	Sim	Sim	Redes celulares
Dhulavvagol et al. [77]	Protótipo de hardware	Ambos	IoT, Cloud	Sim	Sim	IEEE 802.11, Redes celulares
Rahmat et al. [74]	Protótipo de hardware	Externo	IoT, Cloud	Sim	Sim	Ethernet
Sun et al. [78]	Modelo analítico	Ambos	WSN subterrânea	Sim	Sim	Guia de ondas de indução magnética

Tabela 8 – Trabalhos relacionados ao monitoramento de dutos de água (Parte 2)

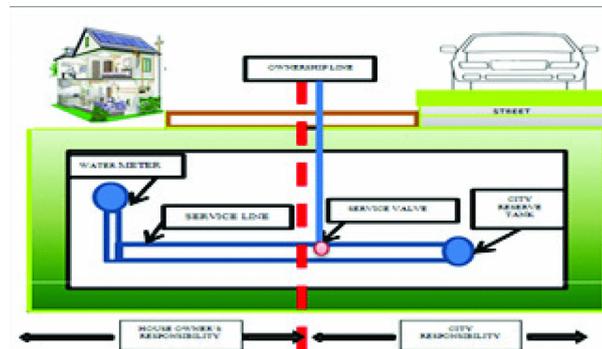
Trabalhos correlatos	Armazenamento dos dados	Plataformas de Hardware	Análise dos dados	Parâmetros medidos	Aplicativo Web/Mobile de acessibilidade
Sadeghioon et al. [17]	Local	Microcontrolador PIC16LF1827, Transceptor de RF	Sim	FSR para medições de pressão	NM
Karray et al. [75]	Local	Processador ARM, Transceptor RF, placa para Arduíno	Sim	FSR para medições de pressão	Aplicativo web
McDougle et al. [50]	Local	Sonda KAPTA 3000 AC4	Sim	Cloro, EC, pressão, temperatura	NM
Natividad e Palaoag [76]	Remoto (usando database MySQL)	Raspberry Pi, módulo GSM, Microcontrolador Arduíno UNO	Não	Pressão e nível da água	Aplicativo web
Dhulavvagol et al. [77]	Sim	Módulo GSM, módulo GPS, Microcontrolador Arduíno	Sim	Fluxo, pressão, estado do duto (usando sensor de vibração)	NM
Rahmat et al. [74]	Local e remoto	Microcontrolador Arduíno	Sim	Fluxo	Aplicativo web
Sun et al. [78]	Local	NM	Sim	Pressão, propriedades do solo, estado do duto (usando sensor acústico)	NM

3.8 DISTRIBUIÇÃO E MONITORAMENTO DE TUBULAÇÕES COM IOT PARA CIDADES INTELIGENTES

Narayanan e Sankaranarayanan (2019) afirmam que a integração da tecnologia de informação e comunicação no sistema de distribuição de água resolverá muitos problemas no sistema de distribuição de água atual e tradicional. A formação de redes inteligentes de distribuição de água (*Smart Water Distribution Network*, ou SWDN) para cidades inteligentes é projetada com reservatórios de armazenamento, estação de bombeamento de reforço, hidrantes e linhas de serviço ao consumidor em mente, e a redundância da rede é fornecida via redes e loops de água inteligentes. Essas redes inteligentes de água são integradas com tecnologias avançadas para detectar o fluxo, a pressão e o consumo de água, a fim de fornecer uma distribuição eficiente de água e aumentar a eficiência do sistema de distribuição para atender a demanda. A gestão da distribuição de água faz parte da iniciativa de *Smart Cities* e ajuda a prever a demanda de abastecimento, além de também cobrir a operação de monitoramento e proteção da rede de distribuição de água por meio do SCADA (*Supervisory Control and Data Acquisition*, ou Controle de Supervisão e Aquisição de Dados).

A Figura 23 ilustra a construção de uma SWDG (*Smart Water Distribution Grid*, ou Rede Inteligente de Distribuição de Água) que interliga o recurso hídrico e o consumidor de

Figura 23 – WDS em uma cidade inteligente



Fonte – (NARAYANAN; SANKARANARAYANAN, 2019)

uma cidade inteligente. A responsabilidade de manutenção da rede de tubos é dividida entre o engenheiro SCADA e o proprietário da casa com base na colocação da tubulação. Com isso, Narayanan e Sankaranarayanan (2019) propõem um sistema de gerenciamento de distribuição de água baseado em IoT e monitoramento de integridade de dutos integrado com computação em nuvem e neblina.

Ao estabelecer um WDN para o sistema de gerenciamento de água, o sistema baseado na gravidade deve ser escolhido, uma vez que é mais preferível e confiável. O WDS (*Water Distribution System*, ou sistema de distribuição de água) deve ser projetado de forma redundante, interligando tubos em loops. As válvulas devem ser colocadas de forma que os reparos possam ser feitos com o mínimo de perturbação no sistema. O SCADA coleta os dados de vários sensores que são implantados na rede em locais remotos e processam os dados de maneira correspondente.

Em uma das pesquisas relacionadas à automação da distribuição de água usando sensores sem fio, foi desenvolvido um sistema de enchimento de fluido que se concentra principalmente no monitoramento do enchimento de um tanque por meio de sensores de nível de água, e o vazamento na tubulação é detectado por meio de um sensor de umidade. Um servidor DWEET é usado na nuvem para buscar os dados e um aplicativo JSOVP é usado para ler os dados do servidor na nuvem. Este sistema é projetado principalmente como um sistema de monitoramento de nível.

A automação da distribuição de água tem a capacidade de misturar a automação em áreas locais e remotas para o controle de válvulas de vazão, pressão, etc. A tomada de decisão centralizada permite soluções econômicas, tornando o sistema eficiente e coeso enquanto opera uma arquitetura para o sistema de distribuição de água. Essa automação também oferece soluções como diagnóstico de falhas, detecção de falhas, regulação e controle da variação da pressão de fluxo, isolamento e restauração de tubulações na rede, aumentando a eficiência e melhorando a eficiência e desempenho do sistema de distribuição de água.

O sistema proposto faz parte da iniciativa de cidades inteligentes. Existem muitos modelos independentes para previsão da demanda de água, monitoramento da saúde da tubu-

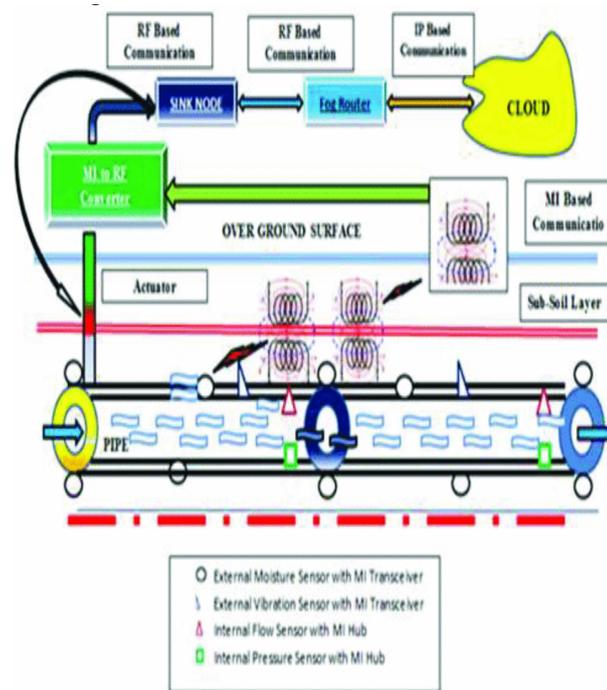
lação, distribuição de água, estimativa do abastecimento de água, avaliação da qualidade da água fornecida, etc. Mas todos esses sistemas são projetados para realizar uma tarefa específica sozinhos, e apenas alguns sistemas estão disponíveis para monitorar sistemas de distribuição de água subterrânea. Com o objetivo final de fornecer um sistema de baixo custo e digno de preço para cidades inteligentes, Narayanan e Sankaranarayanan (2019) propõem este SWDG integrado com IoT e Fog Computing.

Os objetivos básicos que satisfazem as necessidades essenciais das cidades inteligentes são:

1. O sistema deve compreender a necessidade de água (demanda) e as necessidades de pressão (que dependem da topologia do terreno).
2. O WMS deve conhecer os princípios básicos dos requisitos de fornecimento (os requisitos variam dependendo da finalidade da aplicação, como irrigação, casa, segurança contra incêndio, etc.)
3. O WMS deve conhecer sua origem e sua estrutura de recebimento, e também deve ter conhecimento detalhado sobre o layout do sistema de distribuição.
4. O sistema deve ter um entendimento completo do tipo de tubos e também das válvulas de controle.

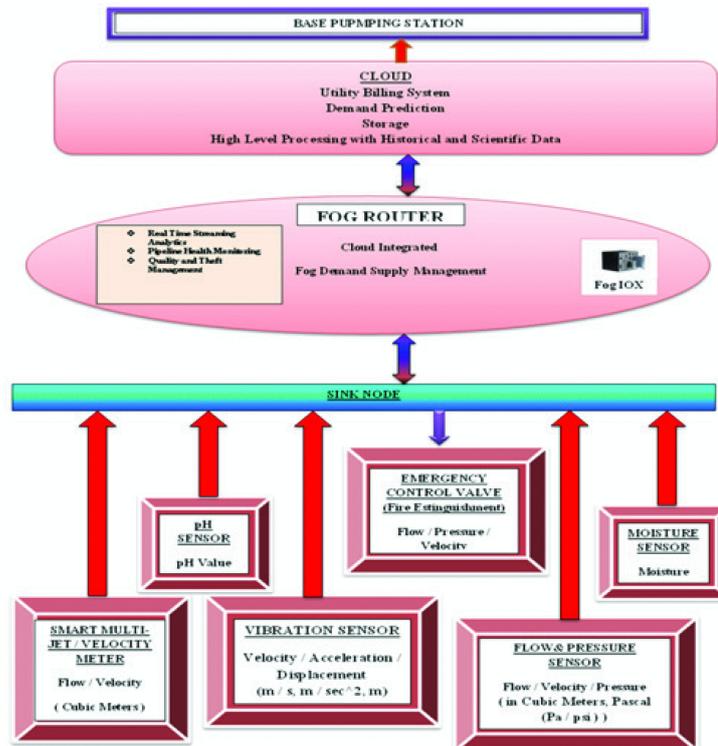
A Figura 24 mostra o fluxo de processo usado pelo SWDG baseado em IoT integrado com a tecnologia *Fog* e *Cloud Computing*. Na Figura 25 é apresentado um diagrama de fluxo para ilustrar a SWDN.

Figura 24 – Arquitetura do SWD baseado em IoT



Fonte – (NARAYANAN; SANKARANARAYANAN, 2019)

Figura 25 – Diagrama de Fluxo da SWDN



Fonte – (NARAYANAN; SANKARANARAYANAN, 2019)

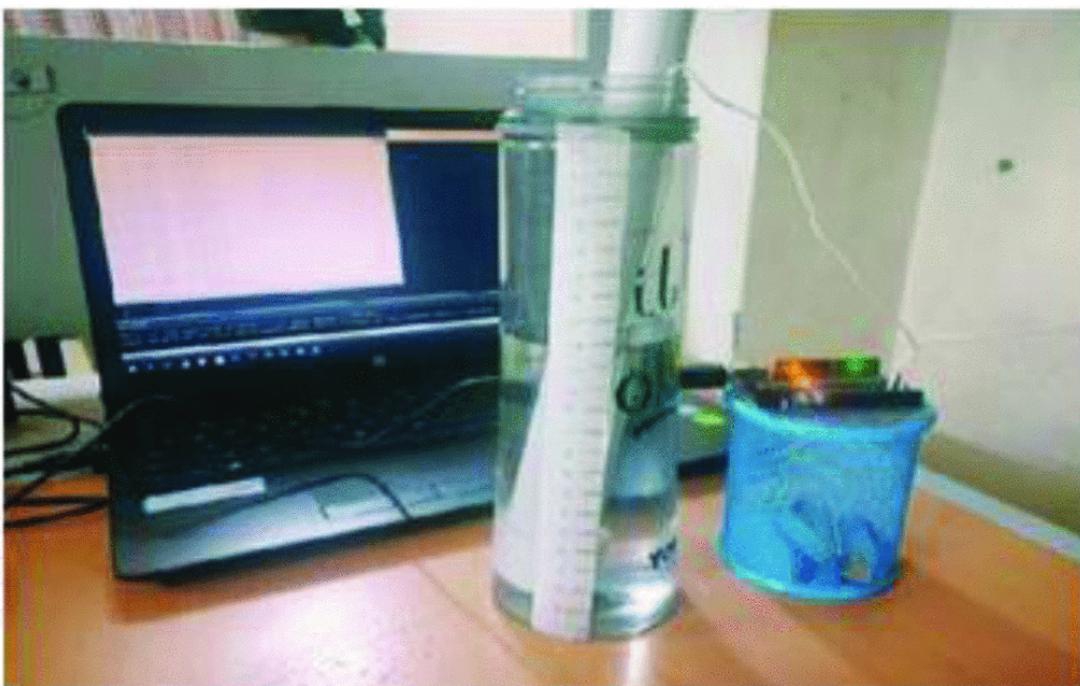
3.9 SISTEMA DE GERENCIAMENTO DE ÁGUA BASEADO EM IOT

Outro artigo, dos autores Jeurkar et al. (2020), apresenta um sistema onde os três componentes principais, que são o medidor de vazão, o sensor de temperatura e o sensor de nível, têm interface com um módulo WiFi por meio de um microcontrolador. O medidor de vazão é instalado dentro do tubo através do qual a água flui em direção ao tanque. Isso ajuda a detectar o fluxo de água para obter o consumo total de água.

Um dos módulos essenciais deste sistema é um sensor de nível baseado em capacitância. A capacitância de um capacitor é diretamente proporcional à área das “placas”. Um eletrodo é colocado no centro do tanque, e sua altura é igual à do tanque. O eletrodo atua como uma placa do capacitor e a água no tanque atua como a segunda placa. Conforme o nível da água sobe, a área da placa do capacitor aumenta, aumentando assim a capacitância do sensor. Não há peças móveis expostas aos fluidos. Este é um tipo de sensor muito sensível e adequado para indicação ou controle contínuo.

Os dados de temperatura e vazão da água, juntamente com o uso, são exibidos nos smartphones por meio de uma plataforma IoT chamada Thingsboard, utilizando o módulo WiFi. Um alarme é disparado se a temperatura da água ultrapassar um determinado nível. O nível da água é mostrado de forma gráfica nos smartphones. O sensor monitora continuamente o nível e ativa um alarme quando o tanque está cheio.

Figura 26 – Calibração do sensor de nível de capacitância

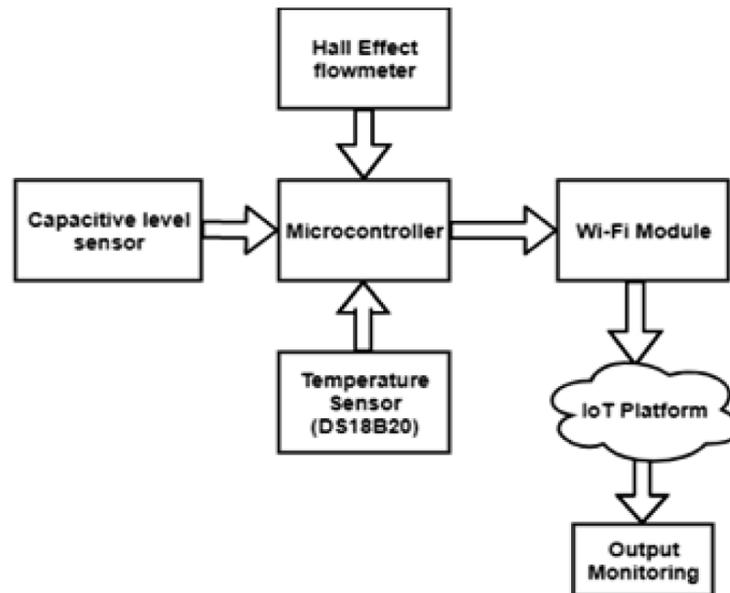


Fonte – (JEURKAR et al., 2020)

O sistema é construído a partir de vários componentes, e os mais relevantes para este trabalho são:

1. Sensor de fluxo de água: este sensor é instalado dentro do tubo para medir a taxa de fluxo de água e calcular a quantidade de água que flui através do tubo. O sensor consiste em uma válvula de plástico, por onde passa a água. A válvula em si consiste em um rotor ao lado de um sensor de *Hall Effect*, usado para detectar e medir o fluxo de água. Quando a água flui pela válvula, ela gira o rotor e as mudanças são observadas de acordo com a velocidade do rotor. Essa alteração é então calculada como saída do sinal de pulso do sensor de *Hall Effect*, portanto, a taxa de fluxo de água pode ser medida.
2. Sensor de nível: este sensor indica o nível da água em um determinado instante. O eletrodo e a água atuam como duas placas do capacitor. A capacitância muda conforme o nível da água sobe ou desce. Um microcontrolador é usado para medir a capacitância entre o eletrodo e a água. A capacitância é medida usando um método RC e a saída é dada em termos de nível. O microcontrolador fornece os dados ao módulo WiFi e os dados são exibidos por meio da plataforma IOT. Na Figura 26, pode-se observar a calibração deste sensor, onde a garrafa é usada como tanque e o tubo de PVC branco inserido é um eletrodo.
3. Microcontrolador: é usado para medir a capacitância e calibrar o sensor de nível. Ele também atua como um meio de comunicação para os sensores e o módulo WiFi. Os dados obtidos de todos os sensores são enviados para o microcontrolador e cálculos são realizados para obter a saída desejada. A saída é enviada ao módulo WiFi pelo microcontrolador.
4. Módulo WiFi: o módulo WiFi usado é o ESP8266, e o protocolo usado para se comunicar com a nuvem é o Transporte de Telemetria de Enfileiramento de Mensagens (*Message Queuing Telemetry Transport*, ou MQTT). Este módulo é usado para exibir os dados recebidos do microcontrolador na plataforma IOT.

Figura 27 – Diagrama de blocos representando a arquitetura do sistema



Fonte – (JEURKAR et al., 2020)

3.10 DESAFIOS EM SISTEMAS DE MONITORAMENTO

Diante de todos os estudos e artigos analisados, é possível concluir que existem muitas formas de se analisar e estudar as medições de consumo de água em casas, apartamentos e até mesmo fazendas, porém, há apenas uma maneira prática de coletar tais dados, que é através do uso de medidores presentes ao longo da tubulação no sistema de distribuição de água utilizado. Estes medidores podem ser utilizados em várias quantidades, mas tipicamente apenas um é o suficiente (especialmente para casas ou apartamentos). As medidas obtidas por estes aparelhos podem variar em precisão das medições (Ji et al. (2015)) e nos métodos usados para contabilizar tais medições (como, por exemplo, utilizando ciclos do medidor, como observado por Rajurkar, Prabakaran e Muthulakshmi (2017) e Jeurkar et al. (2020)), que precisam ser convertidas para medidas úteis aos programas ou algoritmos, os quais visam automatizar o resto do processo.

Para obter as medições, todos os artigos fazem uso de sensores, os quais transmitem os dados obtidos através da IoT ou por computação em névoa (*Fog Computing*), método que também é observado em todos os artigos. O monitoramento é inteiramente feito pelo sistema de forma automática, mas nem todos os sistemas são similares. Os trabalhos de Ji et al. (2015) e Xia et al. (2018) apresentam algoritmos para trabalhar com os dados, sendo, respectivamente, para análise e automatização do sistema. Contudo, apenas Xia et al. (2018) apresenta uma *framework* para implementar tal sistema. Quatro dos oito trabalhos propõem um aplicativo para facilitar a análise dos dados por parte dos usuários, porém apenas os aplicativos de Bhilare e Mali (2016) e Xia et al. (2018) oferecem algum controle do sistema através do aplicativo, permitindo que os usuários controlem, respectivamente, o estado de diversos eletrodomésticos ou definam “*guidelines*” para o algoritmo que automatiza o sistema (respectivamente). O traba-

lho de pesquisa de Saad, Benyamina e Gamatie (2020) é o único que busca soluções para o ambiente de *Smart Agriculture*, enquanto o trabalho de Narayanan e Sankaranarayanan (2019) é o único com foco em *Smart Cities*, já que Ji et al. (2015) faz uma análise apenas de áreas urbanas em geral. Todos os artigos focam em soluções que envolvem água e seus usos, com a única exceção sendo o artigo de Xia et al. (2018), o qual foca seus esforços para trabalhar com energia e seus usos. O trabalho de Bhilare e Mali (2016) é o único que trabalha com água e energia simultaneamente. Todas estas informações podem ser observadas na Tabela 9, onde foram compiladas.

Tabela 9 – Comparativo com os trabalhos relacionados

Artigos	Tipos de redes utilizadas	Usa sensores?	Possui monitoramento automático?	Software	Permite controle manual pelo aplicativo?	Oferece solução para:	Medições
Ji et al. (2015)	IoT, Fog Computing	Sim	Sim	Algoritmo	Não aplica	Nenhum ambiente	Volume de água
Bhilare e Mali (2016)	IoT, Fog Computing, Cloud	Sim	Sim	Aplicativo	Sim	Smart Home	Volume de água e consumo elétrico
Rajurkar, Prabakaran e Muthulakshmi (2017)	IoT, Fog Computing, Cloud	Sim	Sim	Aplicativo	Não	Smart Home	Volume de água
Xia et al. (2018)	IoT, Edge, Fog Computing	Sim	Sim	Algoritmo, Framework e Aplicativo	Sim	Smart Home	Consumo elétrico
Aggarwal, Chauhan e Prakash (2019)	IoT, Fog Computing	Sim	Sim	Aplicativo	Não	Smart Home	Volume de água
Saad, Benyamina e Gamatie (2020)	IoT, Fog Computing, Cloud	Sim	Sim	Não apresenta software	Não aplica	Smart Agriculture	Volume de água
Narayanan e San-karanarayanan (2019)	IoT, Fog Computing, Cloud	Sim	Sim	Não apresenta software	Não aplica	Smart City	Volume de água (dentre outros)
Jeurkar et al. (2020)	IoT, Fog Computing	Sim	Sim	Aplicativo	Não aparenta	Smart Home	Volume e nível de água
Côrte (2020)	IoT, Fog Computing	Sim	Sim	Aplicativo Web	Sim	Smart Home	Volume de água

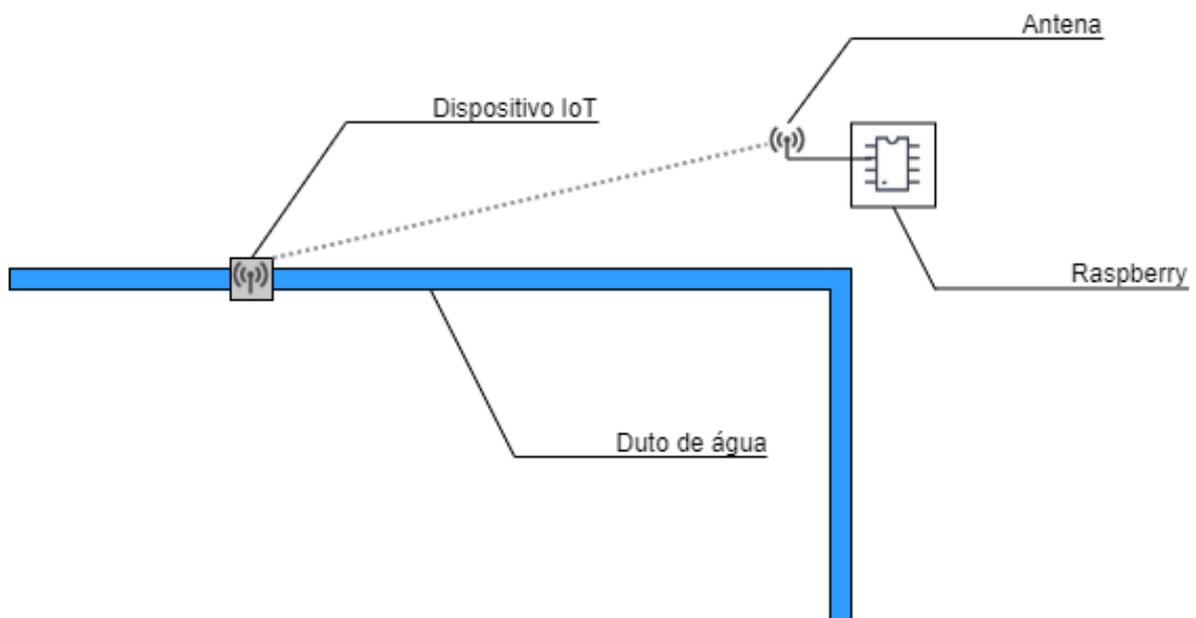
4 PROPOSTA DO SISTEMA DE GERENCIAMENTO DE ÁGUA

Automação tem a capacidade de facilitar a vida de muitas pessoas, reduzindo diversas atividades a simples avisos ou notificações, que servem para alertar que as mesmas foram concluídas. Tendo em vista todos os conteúdos explorados até o momento, com a evolução tecnológica cada vez mais acelerada e presente em nosso cotidiano, desenvolver um sistema que ajude diversos tipos de usuários a acompanhar e controlar seus gastos de um dos recursos mais importantes para o ser humano parece ser inevitável.

Muitas coisas podem dar errado com os sistemas de distribuição de água que são encontrados em casas, apartamentos ou fábricas, desde tubos furados ou quebrados, até torneiras que ficam abertas por muito tempo ou eletrodomésticos que consomem água em demasia. É muito fácil as pessoas demorarem a perceber qualquer um desses problemas, causando desperdício que poderia ser reduzido ou, até mesmo, evitado.

Para este trabalho, é proposto um sistema que combina um medidor de volume de água com um servidor, tendo por objetivo monitorar e automaticamente avisar o usuário quando há algo de errado, ou seja, o usuário deve ser notificado quando o consumo de água está perto do limite mensal ou se está muito alto quando comparado a medições de dias anteriores, ou até quando um vazamento pode estar acontecendo. Para tanto, a medição de água é frequentemente atualizada e salva em um servidor local, o qual pode verificar os dados para alertar se o consumo de água está perto do limite mensal, ou se está muito alto quando comparado às medições dos dias anteriores.

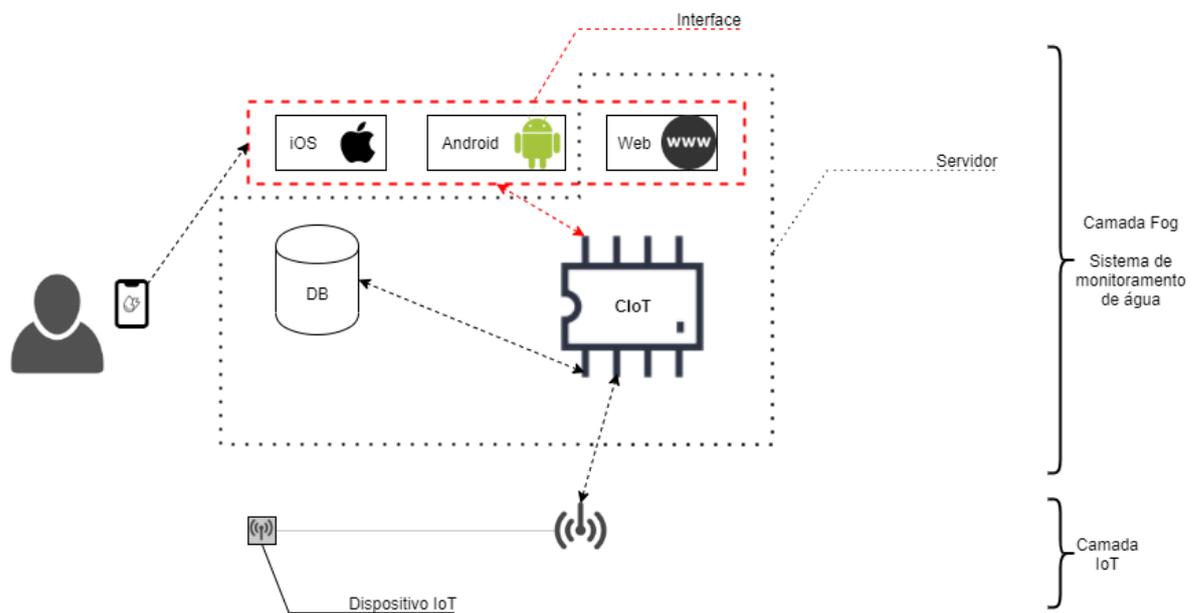
Figura 28 – Visão geral da proposta



Fonte – Produção original

O sistema inclui um sensor que pode ser colocado em qualquer duto de água da residência (de preferência o de entrada), que pode ser observado na Figura 28. Este dispositivo faz medições de água várias vezes por hora (os intervalos são configuráveis) e envia essas medições para um servidor local, através de uma rede sem fio. A combinação do servidor local, do dispositivo que faz as medições e da rede, a qual permite a conexão e interação entre os aparelhos mencionados, resulta em uma rede da Internet das Coisas (IoT). O paradigma IoT é a *cornerstone* (ou pedra angular) que permite que sistemas automatizados atinjam seu potencial.

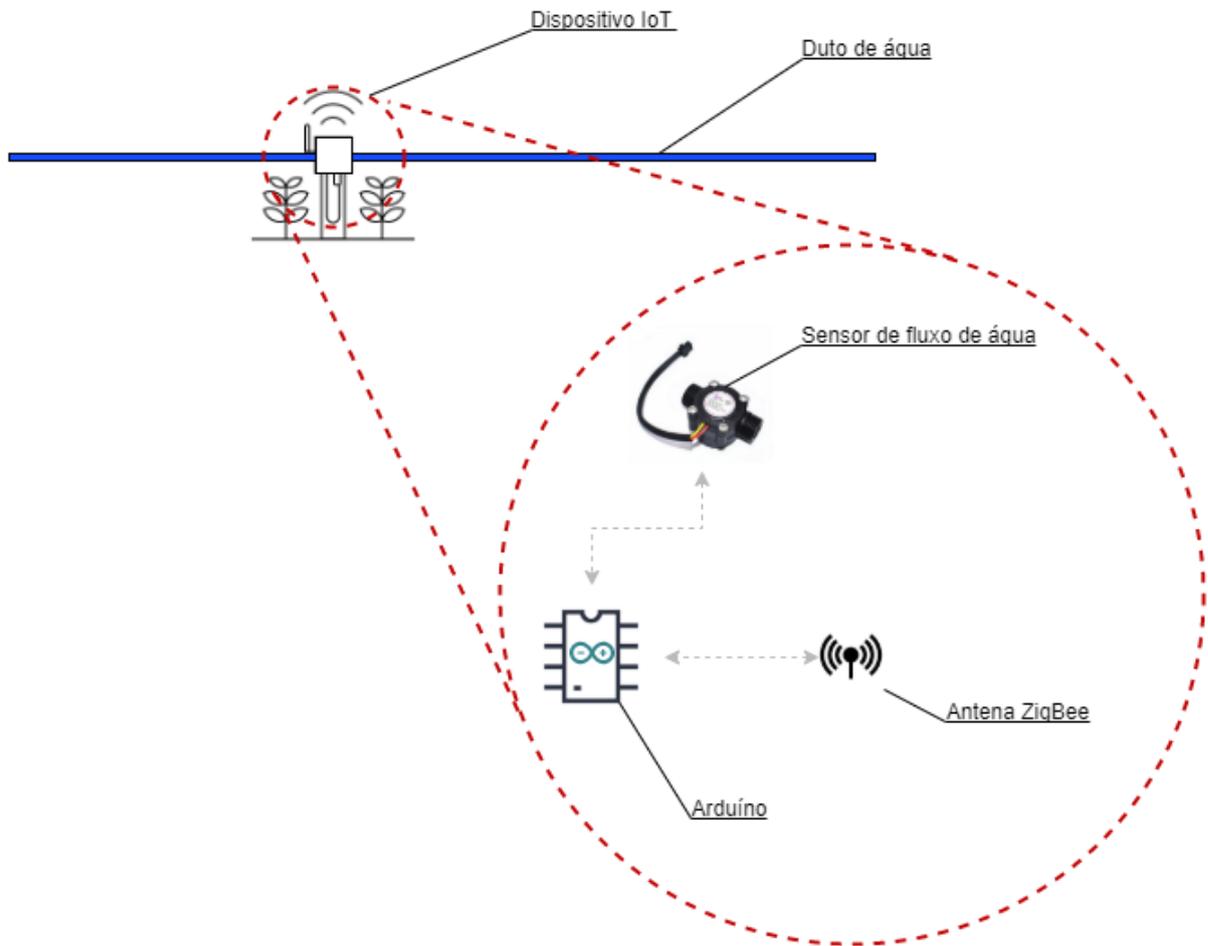
Figura 29 – Visão do Arduíno e sua implementação na proposta



Fonte – Produção original

O servidor roda o sistema CIoT (ou *Central Internet of Things*, desenvolvido no Laboratório de Redes e Gerência (LRG) da UFSC) que controla vários subsistemas e dispositivos de uma casa inteligente, além de disponibilizar funcionalidades para o sistema de monitoramento de água, como avisos de consumo elevado (ou vazamento), disponibilização de dados para análise de média de consumo, estimação de custo da conta de água do mês de acordo com padrões de consumo anteriores, “orçamento” para gasto de água com certas atividades, dentre outras. Exemplificamos esse servidor sendo executado em um Raspberry Pi 3B, detalhado na Figura 29, que contém uma *database* onde são armazenados os dados do medidor, de onde é possível comparar dados novos com anteriores e manter um histórico dos mesmos. Também é possível acessar uma aplicação via *website* hospedado no próprio Raspberry, através de qualquer computador ou smartphone conectado à internet.

Figura 30 – Visão do dispositivo IoT e sua implementação na proposta



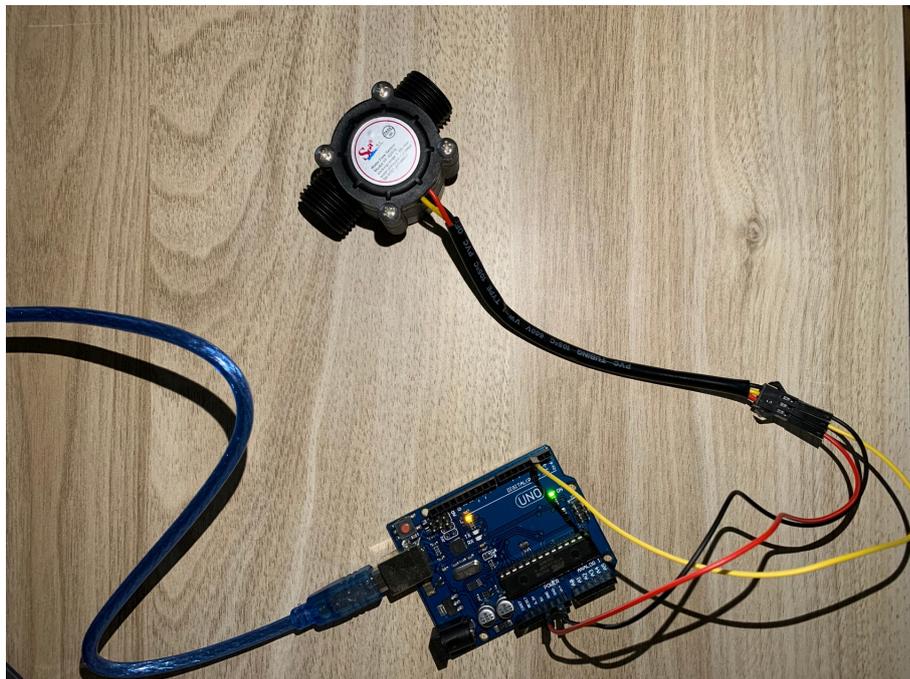
Fonte – Produção original

O dispositivo IoT, exemplificado na Figura 30, é composto por um Arduino e um sensor de fluxo de água. O Arduino lê as medições do sensor a cada intervalo de tempo configurado pelo usuário (por padrão, este intervalo é de 1 minuto). Não é recomendado configurar um intervalo de leitura muito curto, pois o objetivo do dispositivo é ser instalado em locais que costumam não possuir nenhum tipo de tomada elétrica por perto, o que significa que o sistema irá depender de uma bateria na maioria dos casos. A comunicação do Arduino com o servidor é feita através de uma conexão sem fio, utilizando o protocolo *ZigBee*, para ser compatível com a CIoT. Todos os dados obtidos pelo Arduino devem ser apenas enviados (em formato legível) para o servidor, sendo que o processamento lógico dos mesmos deverá ser feito pelo servidor (como mencionado anteriormente).

5 DESENVOLVIMENTO DE UM PROTÓTIPO

Para validar a proposta deste trabalho, foi desenvolvido um pequeno protótipo, composto por um Arduino e por um sensor de fluxo de água simples. Este protótipo não incorpora o aspecto de conexão sem fio da proposta, já que um módulo para testes (ou seja, uma antena) não estava disponível. O mesmo foi instalado na boca de uma mangueira, para simular a medição de gasto de água. A Figura 31 mostra uma foto do protótipo, com o sensor, o Arduino e as conexões físicas estabelecidas entre os mesmos utilizando 3 *jumpers*.

Figura 31 – Arduino e suas conexões físicas com o sensor YF-S201

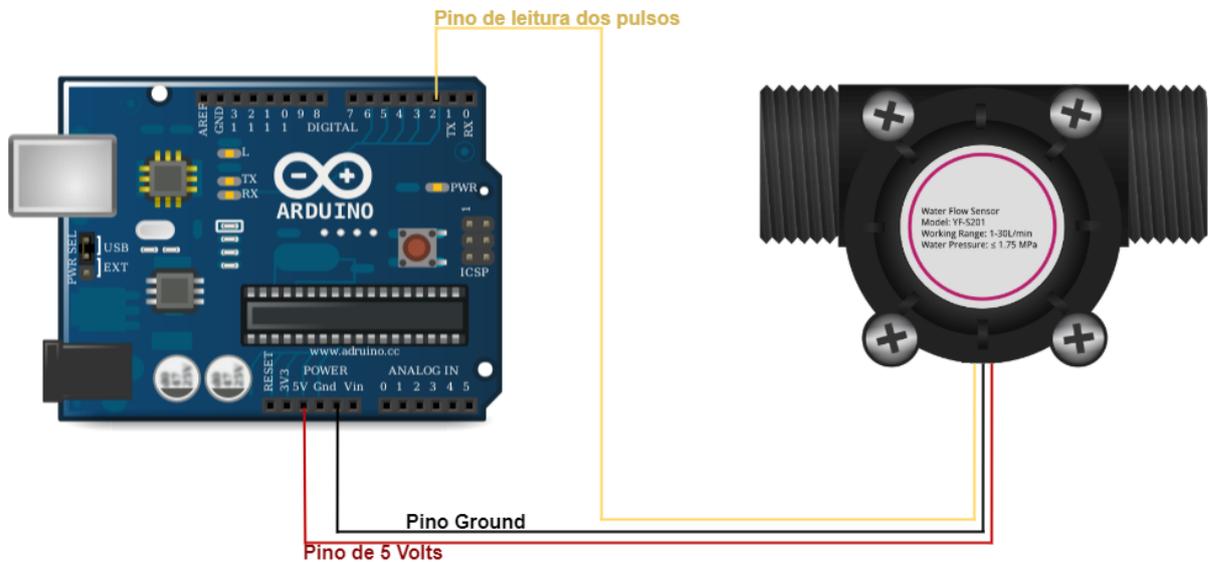


Fonte – Produção original

5.1 HARDWARE DO PROTÓTIPO IOT

O Arduino é uma placa Uno Rev3 que, para facilitar os testes, ficou conectada diretamente a um laptop durante sua operação. Esta placa opera em 5 volts e possui 14 pinos digitais, dos quais 6 possuem *output* PWM (*Pulse-width modulation*, ou Modulação de Largura de Pulso) e outros 6 são analógicos. Três destes pinos foram utilizados para conectar a placa com o sensor do protótipo, como pode ser observado na Figura 32. O cabo vermelho conecta com a porta de 5 volts da Uno Rev3, para alimentar o circuito elétrico do sensor de fluxo de água, o qual é fechado pelo cabo preto, que está conectado a uma das portas *Ground* da placa. Por fim, o cabo amarelo fica conectado à porta digital 2 do Arduino, por onde o mesmo lê os pulsos do sensor.

Figura 32 – Visão do Arduino e sua implementação no dispositivo IoT da proposta



Fonte – Produção original

O sensor de fluxo de água utilizado foi o YF-S201, como pode ser observado nas Figuras 31 e 32. Este modelo possui um limite de funcionamento que varia de 1 a 30 litros de água por minuto, aguentando uma pressão de até 2 Mpa (Mega Pascal), que é equivalente a 2 N/mm^2 (Newton por milímetro quadrado). O aparelho opera com 5 Volts de eletricidade e transmite pulsos que o Arduino pode interpretar, de forma a obter as medições desejadas do sensor.

5.2 SOFTWARE DO PROTÓTIPO IOT

Para permitir que o Arduino faça a leitura do (e interaja com o) sensor de fluxo YF-S201, um pequeno programa foi escrito. Esse programa utiliza algumas variáveis e um método de interrupção, junto com um método `update_system()` e os métodos padrões do Arduino, `setup()` e `loop()`. No método `update_system()`, a porta serial é lida para atualizar duas das variáveis do programa, a que controla o *threshold* de aviso de consumo e a que controla o intervalo de medição customizável, permitindo um terceiro intervalo junto dos predefinidos “minuto” e “hora” (este intervalo não afeta o ciclo de *sleep* e *wakeup* do dispositivo¹). No método `setup()`, o pino 2 é configurado como *input*, o *baud rate* é definido em 9600 e a interrupção é acoplada ao pino 2, para chamar o método `rpm()` toda vez que o valor deste pino subir. Ainda no `setup()`, as interrupções são habilitadas, as variáveis de controle de tempo são inicializadas e, por fim, é impresso na porta serial os valores iniciais do *threshold* e do intervalo de medição customizado. Em seguida temos o método `loop()`, que é o mais complexo deste programa. O fluxograma da Figura 40 representa todo o código (descrito no apêndice A) do Arduino.

¹ O dispositivo está sempre acordado, devido à forma que as leituras são realizadas

O Arduíno envia diversas mensagens pela sua porta serial, codificadas em um formato específico. Essas mensagens podem vir escritas como “m2345”, por exemplo. O primeiro carácter da mensagem que o Arduíno envia serve para identificar o tipo de mensagem sendo impressa, e é sempre uma letra (descritas abaixo). Em uma versão anterior do código, as mensagens sempre terminavam com a letra “z”, porém isto foi substituído pelo uso de quebra de linha (`\n`) através do comando `println()`. As mensagens também contém uma quantidade variada de números, às vezes com casas decimais separadas por um ponto, que podem representar medidas de volume ou tempo decorrido. As letras que definem os diferentes tipos de resposta são:

1. Letra `t`: Quanto tempo passou desde o último aviso de consumo, no formato HHMMSS. Devido ao formato, o número de horas pode eventualmente ficar maior que 100, alterando o formato.
2. Letra `m`: Uma leitura do volume de consumo do último intervalo customizado (por padrão, um minuto).
3. Letra `c`: Confirmação que o *threshold* de aviso foi configurado com sucesso, contendo o número lido.
4. Letra `i`: Confirmação que o intervalo de medição customizado foi configurado com sucesso, contendo o número lido.

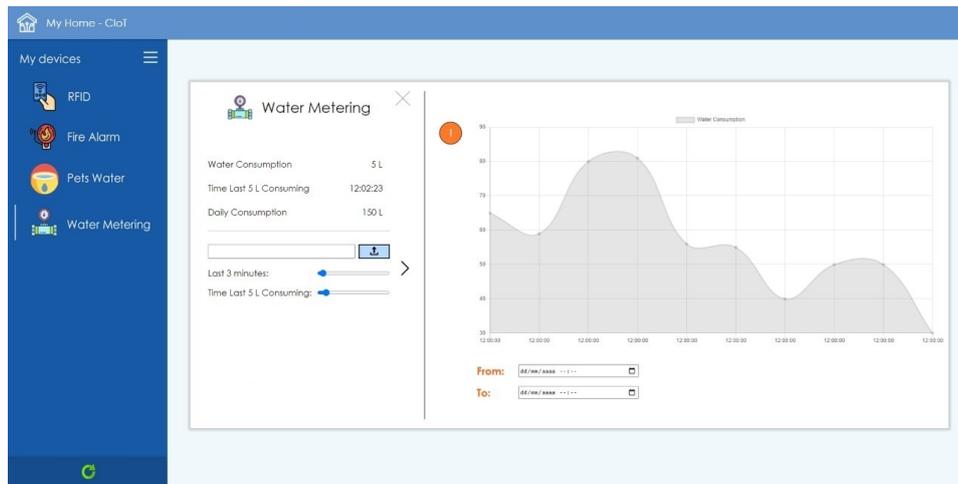
Como mencionado anteriormente, o programa do Arduíno conta com um método para configurar o *threshold* ou para configurar o intervalo de leitura customizável do dispositivo, o `update_system()`. Para invocar esse método, é necessário enviar uma mensagem através da porta serial com um formato específico, e tal mensagem serve para configurar um dos valores de cada vez, não sendo possível alterar ambos simultaneamente com apenas uma mensagem. Há dois formatos que são aceitos pelo Arduíno, e eles seguem as seguintes expressões regulares:

1. `a[0-9]+`: Configura o *threshold* de aviso de consumo para o número definido.
2. `m[0-9]+`: Configura o intervalo de medição customizado para o número definido.

5.3 SERVIDOR WEB (FOG)

Como mencionado na proposta, juntamente com os colegas do Laboratório de Redes e Gerência (LRG) da UFSC, foi desenvolvido um website que oferece uma interface para controlar diversos dispositivos de uma casa inteligente através de uma *Central Internet of Things*, ou CIoT. Parte deste sistema interage com o protótipo desenvolvido para este trabalho, de forma a oferecer as funções detalhadas na proposta do mesmo. A Figura 33 mostra uma janela do website, na aba referente à medição de água da casa.

Figura 33 – Exemplo do servidor web da proposta



Fonte – Produção original

A interface do website conta com alguns elementos interativos para facilitar a navegação. À esquerda, há uma lista com todos os sistemas disponíveis na casa, e os dispositivos presentes em cada sistema são fruto de trabalhos do LRG. Na interface de medição de água, há primeiramente três informações distintas, sendo a primeira, *Water Consumption*, o valor referente ao volume medido no último intervalo de tempo (o qual pode ser customizado pelo usuário). A segunda informação mostra o *timestamp* do último aviso de *threshold*, e a terceira informação mostra o volume total do dia, o qual é calculado com base nas leituras feitas pelo Arduino. Em seguida, há um campo onde pode-se escrever e enviar um comando pela porta serial do Arduino, com dois *sliders* abaixo que permitem configurar os dois valores customizáveis de uma maneira mais intuitiva, sendo o *slider* superior utilizado para configurar o intervalo de leitura e o inferior para controlar o *threshold* (na Figura 33, ambos estão configurados para 3 minutos e 5 litros, respectivamente). Por fim, há um gráfico à direita da página que mostra um intervalo de tempo (que pode ser editado nas caixas de texto abaixo do mesmo) com todas as leituras de volume (que estão salvas na *database*) alinhadas a seus *timestamps*. As leituras no gráfico que forem representadas por um ponto diferente são os momentos onde o *threshold* foi alcançado.

O funcionamento de *backend* do website pode ser analisado pelo fluxograma do apêndice C. Quando o Arduino envia uma mensagem para o servidor, ele precisa enviar um pacote de dados com alguma informação. Esses pacotes são montados de acordo com a explicação no final da subseção 5.2, porém a CIoT requer uma informação adicional, um código hexadecimal no começo da mensagem, e este código não foi implementado no protótipo (logo, o protótipo atual não se comunica com a CIoT). De forma a identificar a qual sistema pertencem os dados das mensagens dos dispositivos, o servidor requer que os mesmos incluam um código hexadecimal no começo de suas mensagens e, caso o protótipo desenvolvido seja atualizado para enviar esse código, ele enviaria o valor hexadecimal 0x13 para identificar dados que pertencem ao sistema de monitoramento de água.

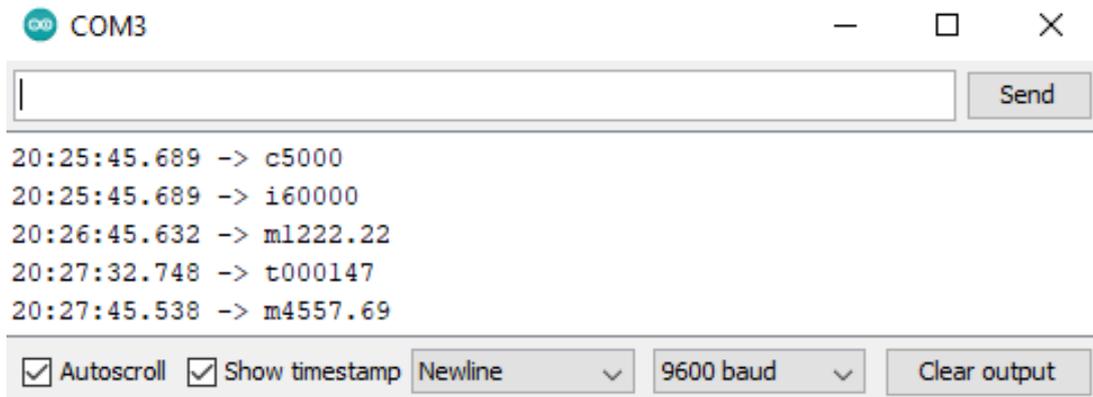
6 TESTES E RESULTADOS

O código do apêndice A foi compilado e rodou no Arduíno para testar a viabilidade do dispositivo base da proposta. Durante o período de uma hora, o sensor de fluxo de água ficou acoplado a uma mangueira de água, normalmente utilizada junto com uma pequena hidropistola para limpeza do vaso sanitário. A torneira da mangueira foi aberta algumas vezes de forma controlada, para permitir que água fluísse pelo sensor, de forma a produzir dados de consumo. Um exemplo das leituras que podem ser obtidas foi demonstrado na Figura 34, e a Tabela 10 possui as leituras de um teste de uma hora, onde é possível ver todas as leituras não nulas desse período (os períodos com muitas leituras nulas em sequência foram omitidos para melhorar a legibilidade da tabela, restando apenas saltos nos minutos onde duas leituras nulas se encontram).

Tabela 10 – Teste do protótipo durante uma hora

Número	Data e hora da leitura	Mensagem	Medição	Tipo de mensagem (identificado pela letra)
01	07/11/2020 - 10:17:05.791	m1193	1193ml	Leitura após 1 minuto (configurado)
02	07/11/2020 - 10:18:05.702	m0	0ml	Leitura após 1 minuto (configurado)
03	07/11/2020 - 10:19:05.664	m320	320ml	Leitura após 1 minuto (configurado)
04	07/11/2020 - 10:20:05.559	m0	0ml	Leitura após 1 minuto (configurado)
05	07/11/2020 - 10:21:05.506	m2177	2177ml	Leitura após 1 minuto (configurado)
06	07/11/2020 - 10:22:05.419	m0	0ml	Leitura após 1 minuto (configurado)
07	07/11/2020 - 10:24:05.255	m0	0ml	Leitura após 1 minuto (configurado)
08	07/11/2020 - 10:24:35.751	a000830	8m e 30s	Aviso de <i>threshold</i> de 5 litros
09	07/11/2020 - 10:25:05.210	m2235	2235ml	Leitura após 1 minuto (configurado)
10	07/11/2020 - 10:26:05.102	m0	0ml	Leitura após 1 minuto (configurado)
11	07/11/2020 - 10:37:04.304	m0	0ml	Leitura após 1 minuto (configurado)
12	07/11/2020 - 10:38:04.218	m1959	1959ml	Leitura após 1 minuto (configurado)
13	07/11/2020 - 10:38:43.469	a001409	14m e 9s	Aviso de <i>threshold</i> de 5 litros
14	07/11/2020 - 10:39:04.150	m3251	3251ml	Leitura após 1 minuto (configurado)
15	07/11/2020 - 10:40:04.057	m73	73ml	Leitura após 1 minuto (configurado)
16	07/11/2020 - 10:41:04.020	m0	0ml	Leitura após 1 minuto (configurado)
17	07/11/2020 - 10:46:03.639	m0	0ml	Leitura após 1 minuto (configurado)
18	07/11/2020 - 10:47:03.547	m946	946ml	Leitura após 1 minuto (configurado)
19	07/11/2020 - 10:48:03.492	m1593	1593ml	Leitura após 1 minuto (configurado)
20	07/11/2020 - 10:48:51.970	a001009	10m e 9s	Aviso de <i>threshold</i> de 5 litros
21	07/11/2020 - 10:49:03.424	m1562	1562ml	Leitura após 1 minuto (configurado)
22	07/11/2020 - 10:50:03.319	m337	337ml	Leitura após 1 minuto (configurado)
23	07/11/2020 - 10:51:03.241	m0	0ml	Leitura após 1 minuto (configurado)
24	07/11/2020 - 10:54:03.052	m0	0ml	Leitura após 1 minuto (configurado)
25	07/11/2020 - 10:55:02.969	m428	428ml	Leitura após 1 minuto (configurado)
26	07/11/2020 - 10:55:47.113	a000656	6m e 56s	Aviso de <i>threshold</i> de 5 litros
27	07/11/2020 - 10:56:02.889	m5284	5284ml	Leitura após 1 minuto (configurado)
28	07/11/2020 - 10:57:02.807	m300	300ml	Leitura após 1 minuto (configurado)
29	07/11/2020 - 10:58:02.745	m0	0ml	Leitura após 1 minuto (configurado)
30	07/11/2020 - 11:16:01.417	m0	0ml	Leitura após 1 minuto (configurado)

Figura 34 – Exemplo da janela da porta serial do Arduino



Fonte – Produção original

O *threshold* padrão de 5 litros não foi alterado nos testes, bem como o intervalo customizado de medições, que forneceu as leituras por minuto (sua configuração padrão), as quais produziram os resultados da Tabela 10. A Figura 35 mostra uma visão geral do local onde o protótipo foi instalado, enquanto as Figuras 36 e 37 apresentam o protótipo em ação. A Figura 36 representa um grande fluxo de água, enquanto a Figura 37 representa um pequeno fluxo de água.

Figura 35 – Protótipo em ação, conectado à mangueira



Fonte – Produção original

Figura 36 – Protótipo em ação, com fluxo “grande” de água visível



Fonte – Produção original

Figura 37 – Protótipo em ação, com fluxo “pequeno” de água visível



Fonte – Produção original

Os testes feitos com a mangueira servem apenas para demonstrar que o dispositivo IoT proposto funciona. Os resultados obtidos e demonstrados na Tabela 10 são meramente ilustrativos, porém nos permitem chegar a algumas conclusões. De acordo com os resultados, pode-se concluir que o sensor de fluxo utilizado tem uma taxa de erro em torno dos 25%, o que torna este sensor impraticável em aplicações que necessitem de alta precisão. Alguns *outliers* podem ser observados, como um erro de 8.85% e outro de 56.2%. As medições manuais para validar a precisão das leituras foram feitas com ajuda de um pequeno medidor de cozinha de 250 mililitros. As várias leituras (dos resultados) foram discutidas na lista abaixo:

- A medida 01, de 1193 mililitros, resultou de um gasto controlado de 1 litro, apresentando uma margem de erro de 19,3%.
- A medida 03, de 320 mililitros, resultou de um gasto controlado de 250 mililitros, apresentando uma margem de erro de 28%.
- As medidas 05 e 09, de 2177 e 2235 mililitros respectivamente, resultaram de dois gastos controlados de 2 litros cada, apresentando erros de 8.85% e 11.75%.
- As medidas 12, 14 e 15, de 1959, 3251 e 73 mililitros respectivamente, não foram medidas manualmente. Considerando um erro de 10% para as duas primeiras medidas, pode-se estimar que o gasto foi de 1763 e 2926 mililitros respectivamente. Para a terceira medida, considera-se um erro de 25%, logo resultando em uma estimativa de 55 mililitros. As taxas de erro para estas estimativas foram escolhidas com base nos itens anteriores desta lista.
- As medidas 18, 19, 21 e 22 são resultantes de um fluxo contínuo, e apenas a medida 21 de 1562 mililitros foi medida manualmente. Ao longo do minuto onde essa medida foi obtida, o medidor utilizado foi manualmente enchido e despejado 4 vezes, sem interromper o fluxo de água. Assim, pode-se concluir que há um erro de 56.2% para um fluxo “pequeno” e contínuo.
- A medida 27, de 5284 mililitros, foi resultado de um fluxo contínuo “grande”. O medidor foi utilizado novamente e, desta vez, foi enchido e despejado 16 vezes. Pode-se concluir, portanto, que houve um erro de 32% nesta medição.
- Ao final das leituras, exatamente uma hora após a inicialização do dispositivo (no mesmo momento do *timestamp* da leitura 30), pode-se calcular a leitura de uma hora com base nas leituras parciais. Logo, temos uma medida de 21666.9 mililitros (21.666 litros). A medida real, considerando um erro médio de 25%, deveria ser mais próxima dos 16250.175 mililitros.

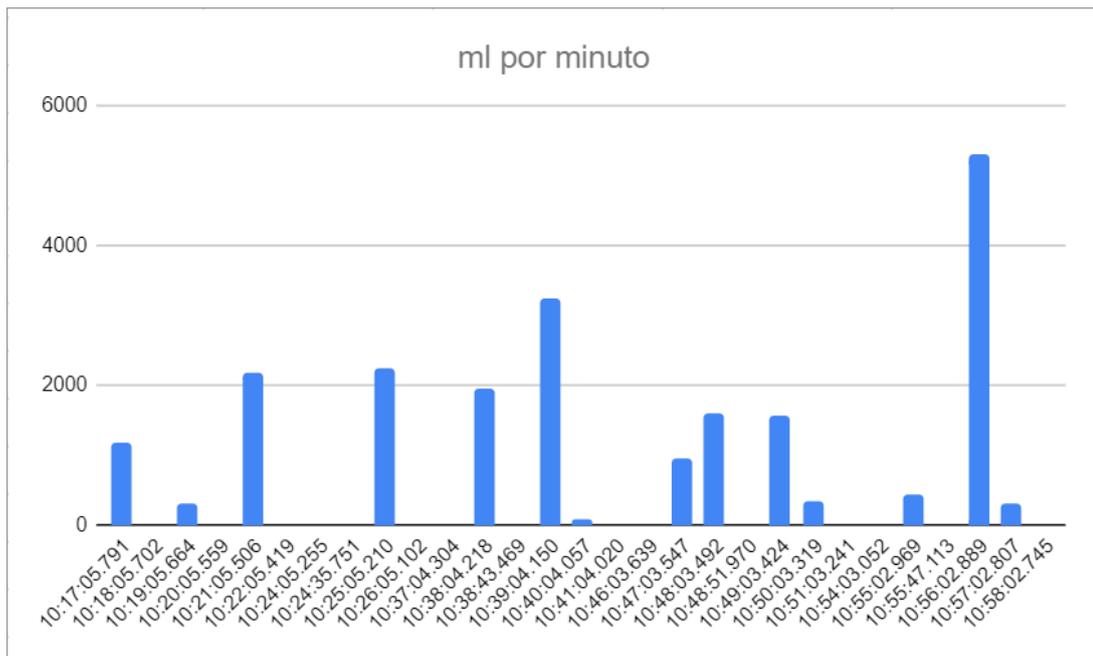
Devido à necessidade de manter o dispositivo conectado a um computador, não foi possível testar outras funcionalidades da proposta. Há um pequeno desvio de milissegundos a cada leitura que resulta (a longo prazo) em variação nos horários exatos das leituras.

Com as medidas obtidas pelo Arduíno, o servidor pode utilizar os valores dos volumes e os *timestamps* das medições para viabilizar as várias funções mencionadas no capítulo 4. Esses *timestamps* e medições permitem que o servidor vá gerando gradualmente um gráfico do consumo de água, que pode ser utilizado para diversas funções, como permitir que o servidor identifique possíveis vazamentos a partir de leituras irregulares (por “leituras irregulares” entende-se que o servidor, ao analisar os dados de consumo anteriores, pode deduzir matematicamente que uma leitura x com um *timestamp* y não é comum para os padrões de uso de água da residência). Também é possível identificar quais medições são de qual mês e calcular

uma aproximação do gasto total do mesmo, viabilizando os alertas de limite de gasto mensal. O *threshold* customizável facilita a definição de limites menores para permitir o controle de períodos mais variados.

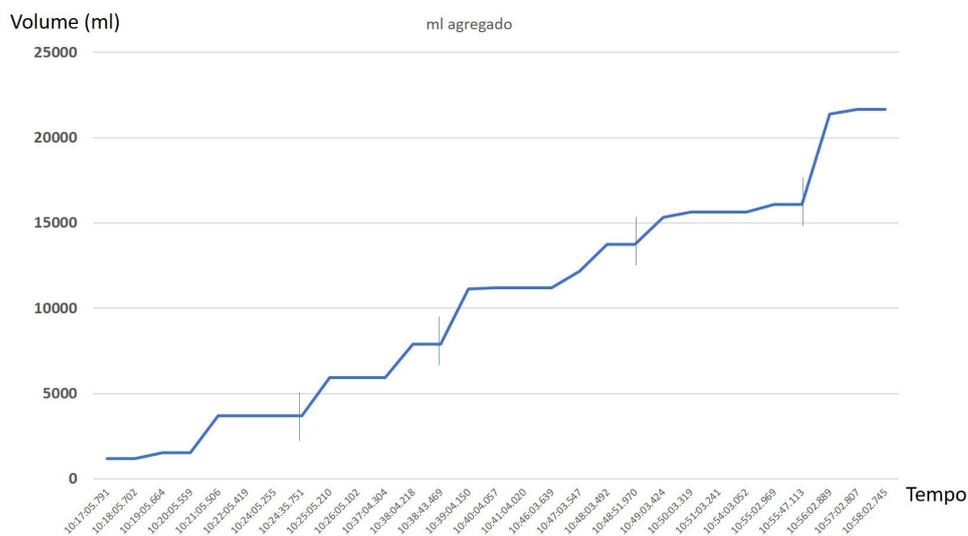
As Figuras 38 e 39 apresentam os gráficos que a aplicação web, da Figura 33, pode produzir com leituras como as deste teste. Na Figura 39, os avisos de *threshold* estão denotados por uma linha vertical presente em algumas leituras.

Figura 38 – Gráfico das leituras individuais



Fonte – Produção original

Figura 39 – Gráfico das leituras acumuladas



Fonte – Produção original

7 CONCLUSÕES

O objetivo deste trabalho era demonstrar como uma solução para gerenciamento autônomo do consumo de água de uma residência poderia ser implementado, levando em consideração o paradigma *Internet of Things*. A pesquisa realizada permitiu a obtenção de um entendimento mais avançado sobre como as redes IoT podem ser utilizadas para se implementar a proposta eventualmente desenvolvida. O protótipo produzido foi capaz de obter medições simples necessárias para implementar todas as outras funcionalidades da proposta (as quais ficaram reservadas ao servidor da rede local).

Os resultados dos testes do protótipo ressaltaram o quão impreciso o sensor de fluxo YF-S201 é, indicando até 56% de erro em baixo fluxo e 32% em alto fluxo. Pela própria natureza do sensor, as precisões obtidas pelos testes deixam muito a desejar quando se leva em conta o objetivo inicial da proposta, a qual pretendia alcançar um sistema com alta confiabilidade e autonomia. Mesmo que o protótipo tenha se demonstrado perfeitamente capaz de ser autônomo, verifica-se a necessidade de substituir o sensor de fluxo por outro modelo com maior precisão, já que a confiabilidade do sistema depende quase que inteiramente dessa precisão.

7.1 TRABALHOS FUTUROS

Observa-se que existem melhorias a serem feitas no protótipo, e para exemplificar mais detalhadamente essa afirmação, nota-se que o sistema pode:

- ser atualizado para medir uma quantidade mais variada de dados, como pressão de água, temperatura, “pureza” da água, dentre outros;
- ser ampliado para monitorar condomínios inteiros, ou talvez até cidades;
- ser multiplicado para monitorar diversas saídas de água de uma casa, capacitando-o a medir os gastos específicos de torneiras, chuveiros e/ou mangueiras individualmente;
- ser otimizado para não precisar ficar ligado 100% do tempo para ler o sensor;
- adicionar uma antena para aprimorar a comunicação do dispositivo com a CIoT;
- ser atualizado com um sensor mais robusto.

REFERÊNCIAS

- AGGARWAL, S.; CHAUHAN, S.; PRAKASH, R. J. An Automated System to Monitor the Usage of Water in Apartments Using IOT and Artificial Neural Network. In: **2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)**. [S.l.]: IEEE, 2019. p. 821–825. ISBN 978-1-7281-0282-5.
- ALRAWAIS, A. et al. Fog Computing for the Internet of Things: Security and Privacy Issues. **IEEE Internet Computing**, IEEE, v. 21, n. 2, p. 34–42, 2017.
- BHILARE, R.; MALI, S. IoT based smart home with real time E-metering using E-controller. **12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015**, p. 1–6, 2016.
- CAN, O.; SAHINGOZ, O. K. A survey of intrusion detection systems in wireless sensor networks. In: **2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)**. [S.l.]: IEEE, 2015. p. 1–6. ISBN 978-1-4673-6601-4.
- GREENGARD, S. Cybersecurity gets smart. **Communications of the ACM**, v. 59, p. 29–31, 2016. ISSN 15577317.
- HOSSAIN, M. M.; FOTOUHI, M.; HASAN, R. Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things. In: **2015 IEEE World Congress on Services**. [S.l.]: IEEE, 2015. p. 21–28. ISBN 978-1-4673-7275-6.
- IORGA, M. et al. Fog computing conceptual model. **NIST Special Publication**, n. 500-325, p. 11, 2018.
- JEURKAR, V. et al. IOT Based Water Management System. **2020 International Conference on Industry 4.0 Technology, I4Tech 2020**, p. 141–144, 2020.
- JI, Y. et al. Analysis of urban residential water consumption based on smart meters and fuzzy clustering. **Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se**, p. 1295–1301, 2015.
- MASIA, O. A.; ERASMUS, L. A study into the implementation of Water Conservation/Water Demand Management in Gauteng municipalities. **2013 Proceedings of PICMET 2013: Technology Management in the IT-Driven Services**, PICMET, p. 2660–2673, 2013.
- NARAYANAN, L. K.; SANKARANARAYANAN, S. IoT Enabled Smart Water Distribution and Underground Pipe Health Monitoring Architecture for Smart Cities. In: **2019 IEEE 5th International Conference for Convergence in Technology (I2CT)**. [S.l.]: IEEE, 2019. p. 1–7. ISBN 978-1-5386-8075-9.
- PACHECO, J.; HARIRI, S. IoT security framework for smart cyber infrastructures. **Proceedings - IEEE 1st International Workshops on Foundations and Applications of Self-Systems, FAS-W 2016**, IEEE, p. 242–247, 2016.
- RAJURKAR, C.; PRABAHARAN, S. R.; MUTHULAKSHMI, S. IoT based water management. **2017 International Conference On Nextgen Electronic Technologies: Silicon to Software, ICNETS2 2017**, p. 255–259, 2017.

SAAD, A.; BENYAMINA, A. E. H.; GAMATIE, A. Water Management in Agriculture: A Survey on Current Challenges and Technological Solutions. **IEEE Access**, p. 38082–38097, 2020. ISSN 21693536.

SAMPAIO, H. V. et al. Autonomic IoT Battery Management with Fog Computing. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 11484 LNCS, n. Cc, p. 89–103, 2019.

SUO, H. et al. Security in the internet of things: A review. **Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012**, IEEE, v. 3, p. 648–651, 2012.

XIA, C. et al. Edge-based energy management for smart homes. **Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3**, IEEE, p. 842–848, 2018.

APÊNDICE A – CÓDIGO DO ARDUÍNO

Listing 1 – Código do Arduino do protótipo

```

1 // Variáveis
2 // Contador de pulsos
3 volatile int pulsos;
4
5 // Contadores de volume para várias funções (todos em mililitros)
6 float volume_instantaneo = 0;
7 float volume_aviso = 0;
8 float volume_custom = 0;
9
10 // Variáveis para mapear e auxiliar funções do Arduino
11 unsigned char pino_sensor_fluxo = 2;
12 char serial_reader_char = 0;
13 unsigned long serial_reader_long = 0;
14
15 // Contadores de tempo para várias funções
16 unsigned long tempo_atual = 0;
17 unsigned long intervalo_custom_medicao = 60000; // Configurável,
    ↪ 1 minuto por padrão
18 unsigned long loop_custom = 0;
19 unsigned long loop_de_um_segundo = 0;
20 unsigned long tempo_segundos_aviso = 0;
21 unsigned long tempo_minutos_aviso = 0;
22 unsigned long tempo_horas_aviso = 0;
23
24 // Variável que define os n litros de threshold (em mililitros)
25 unsigned long threshold_aviso = 5000; // Configurável, 5L por
    ↪ padrão
26
27 // Identificadores de mensagens
28 char identificador_medicao_aviso_threshold = 't';
29 char identificador_medicao_custom = 'm';
30 char identificador_threshold_configurado = 'c';
31 char identificador_intervalo_configurado = 'i';
32
33 // Métodos -----
34

```

```

35 // Método que é chamado via interrupções
36 void rpm() {
37     pulsos++;
38 }
39
40 // Método que lê a porta serial para atualizar o threshold ou o
    ⇨ intervalo de medição custom
41 void update_system() {
42     serial_reader_char = Serial.read();
43
44     if(serial_reader_char == 'a') { // Update do volume que gera
        ⇨ aviso
45         serial_reader_long = Serial.parseInt();
46
47         // Caso o valor seja não nulo:
48         if(serial_reader_long > 0) {
49             // Atualizar o threshold
50             threshold_aviso = serial_reader_long;
51
52             // Responder com um sinal de OK, informando o valor
                ⇨ configurado
53             Serial.print(identificador_threshold_configurado);
54             Serial.println(threshold_aviso);
55         }
56     } else if(serial_reader_char == 'm') { // Update do intervalo
        ⇨ de medição
57         serial_reader_long = Serial.parseInt();
58
59         if(serial_reader_long > 0) {
60             // Atualizar o intervalo
61             intervalo_custom_medicao = serial_reader_long;
62
63             Serial.print(identificador_intervalo_configurado);
64             Serial.println(intervalo_custom_medicao);
65         }
66     } else { // Comando formatado incorretamente
67         Serial.println("Input incorreto. Utilize a ou m seguido
            ⇨ de números. Exemplo: a21000 configura um aviso de 21
            ⇨ litros");

```

```
68     }
69
70     // Esvaziar o buffer
71     while(Serial.available() > 0) {
72         Serial.read();
73     }
74 }
75
76 // Setup do Arduíno
77 void setup() {
78     // Inicializar o pino do sensor no modo INPUT
79     pinMode(pino_sensor_fluxo, INPUT);
80
81     // Inicializar a porta serial com baud rate de 9600
82     Serial.begin(9600);
83
84     // Acoplar o interrupt à porta do sensor
85     attachInterrupt(digitalPinToInterrupt(pino_sensor_fluxo),
86         ↪ rpm, RISING);
87
88     // Habilitar as interrupções
89     sei();
90
91     // Definir o tempo atual para as variáveis necessárias
92     tempo_atual = millis();
93     loop_de_um_segundo = tempo_atual;
94     loop_custom = tempo_atual;
95
96     // Informar o threshold de aviso de consumo
97     Serial.print(identificador_threshold_configurado);
98     Serial.println(threshold_aviso);
99
100    // Informar o intervalo custom
101    Serial.print(identificador_intervalo_configurado);
102    Serial.println(intervalo_custom_medicao);
103 }
104 // Loop do Arduíno
105 void loop() {
```

```

106 // Atualizar as variáveis de controle caso necessário
107 if(Serial.available() > 0) {
108     update_system();
109 }
110
111 // Atualizar o tempo_atual a cada novo loop
112 tempo_atual = millis();
113
114 // Calcular o volume deste loop (considerando que um pulso
115     ↪ vale aproximadamente 2,222222 mililitros)
116 volume_instantaneo = pulsos * 2.222222;
117
118 // Zerar o contador de pulsos
119 pulsos = 0;
120
121 // Aumentar os volumes
122 volume_aviso = volume_aviso + volume_instantaneo;
123 volume_custom = volume_custom + volume_instantaneo;
124
125 // Caso um segundo tenha passado:
126 if(tempo_atual >= (loop_de_um_segundo + 1000)) {
127     loop_de_um_segundo = tempo_atual; // Atualizar para o
128     ↪ próximo segundo
129     tempo_segundos_aviso = tempo_segundos_aviso + 1; //
130     ↪ Atualizar o contador de segundos
131 }
132
133 // Caso o intervalo configurado tenha passado:
134 if(tempo_atual >= (loop_custom + intervalo_custom_medicao)) {
135     loop_custom = tempo_atual; // Atualizar para o próximo
136     ↪ loop
137     Serial.print(identificador_medicao_custom); // Sinalizar
138     ↪ o tipo de mensagem
139     Serial.println(volume_custom); // Imprimir o valor da
140     ↪ leitura do último intervalo
141     volume_custom = 0; // Medição do intervalo customizado é
142     ↪ zerada
143 }
144
145

```

```

138     // Caso o consumo tenha atingido n litros
139     if(volume_avisos > threshold_avisos) {
140         if(tempo_segundos_avisos < 60) { // Caso este fato tenha
141             ↪ ocorrido em menos de um minuto
142             tempo_horas_avisos = 0; // Zera as horas a serem
143             ↪ impressas
144             tempo_minutos_avisos = 0; // Zera os minutos a serem
145             ↪ impressos
146         } else if (tempo_segundos_avisos < 3600) { // Caso este
147             ↪ fato tenha ocorrido em menos de uma hora
148             tempo_horas_avisos = 0; // Zera as horas a serem
149             ↪ impressas
150
151             // Calcular os minutos corretos
152             tempo_minutos_avisos = tempo_segundos_avisos / 60;
153             for (int i = 0; i < tempo_minutos_avisos; i++) {
154                 tempo_segundos_avisos = tempo_segundos_avisos - 60;
155                 ↪ // Corrigir o valor de segundos a serem
156                 ↪ impressos
157             }
158
159         } else { // Caso este fato tenha ocorrido após uma hora
160             // Calcular as horas corretas
161             tempo_horas_avisos = tempo_segundos_avisos / 3600;
162             for (int i = 0; i < tempo_horas_avisos; i++) {
163                 tempo_segundos_avisos = tempo_segundos_avisos -
164                 ↪ 3600; // Corrigir o valor de segundos a serem
165                 ↪ impressos
166             }
167
168             if(tempo_segundos_avisos > 60) { // Caso true, minutos
169                 ↪ não será igual a zero
170                 // Calcular os minutos corretos
171                 tempo_minutos_avisos = tempo_segundos_avisos / 60;
172
173                 for (int i = 0; i < tempo_minutos_avisos; i++) {
174                     tempo_segundos_avisos = tempo_segundos_avisos -
175                     ↪ 60; // Corrigir o valor de segundos a
176                     ↪ serem impressos

```

```

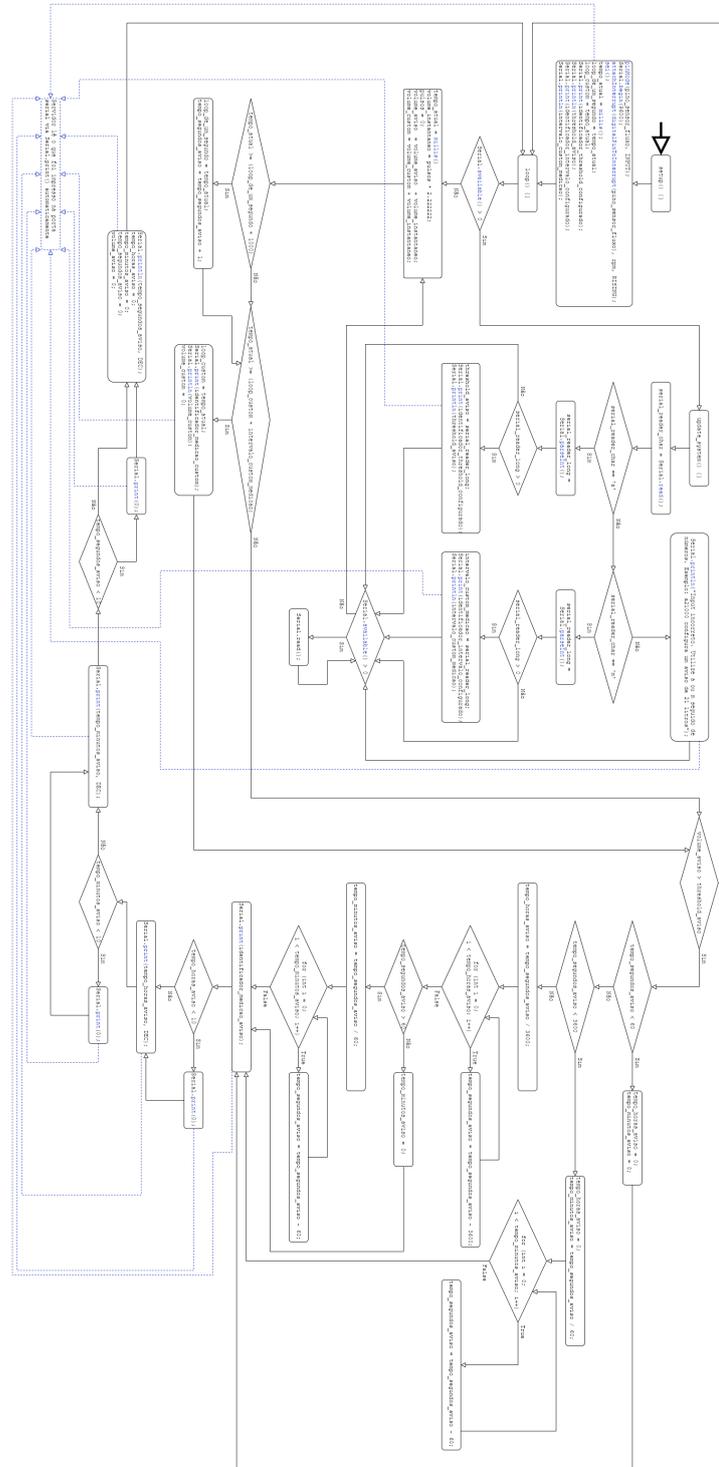
165         }
166
167     } else { // Caso false, minutos será igual a zero
168         tempo_minutos_avisos = 0; // Zera os minutos a
169         ↪ serem impressos
170     }
171 }
172
173 // Imprimir o aviso de consumo de n litros em m horas, w
174 ↪ minutos e z segundos
175 Serial.print(identificador_medicao_avisos_threshold); //
176 ↪ Sinalizar o tipo da mensagem
177
178 if(tempo_horas_avisos < 10) {
179     Serial.print(0); // Zero à esquerda nos dígitos da
180     ↪ hora
181 }
182 Serial.print(tempo_horas_avisos, DEC); // Imprimir as
183 ↪ horas do aviso
184
185 if(tempo_minutos_avisos < 10) {
186     Serial.print(0); // Zero à esquerda nos dígitos dos
187     ↪ minutos
188 }
189 Serial.print(tempo_minutos_avisos, DEC); // Imprimir os
190 ↪ minutos do aviso
191
192 if(tempo_segundos_avisos < 10) {
193     Serial.print(0); // Zero à esquerda nos dígitos dos
194     ↪ segundos
195 }
196 Serial.println(tempo_segundos_avisos, DEC); // Imprimir os
197 ↪ segundos do aviso
198
199 // Zerar os contadores de tempo
200 tempo_horas_avisos = 0;
201 tempo_minutos_avisos = 0;
202 tempo_segundos_avisos = 0;
203
204

```

```
195         // Zerar o contador de aviso de n litros
196         volume_aviso = 0;
197     }
198 }
```

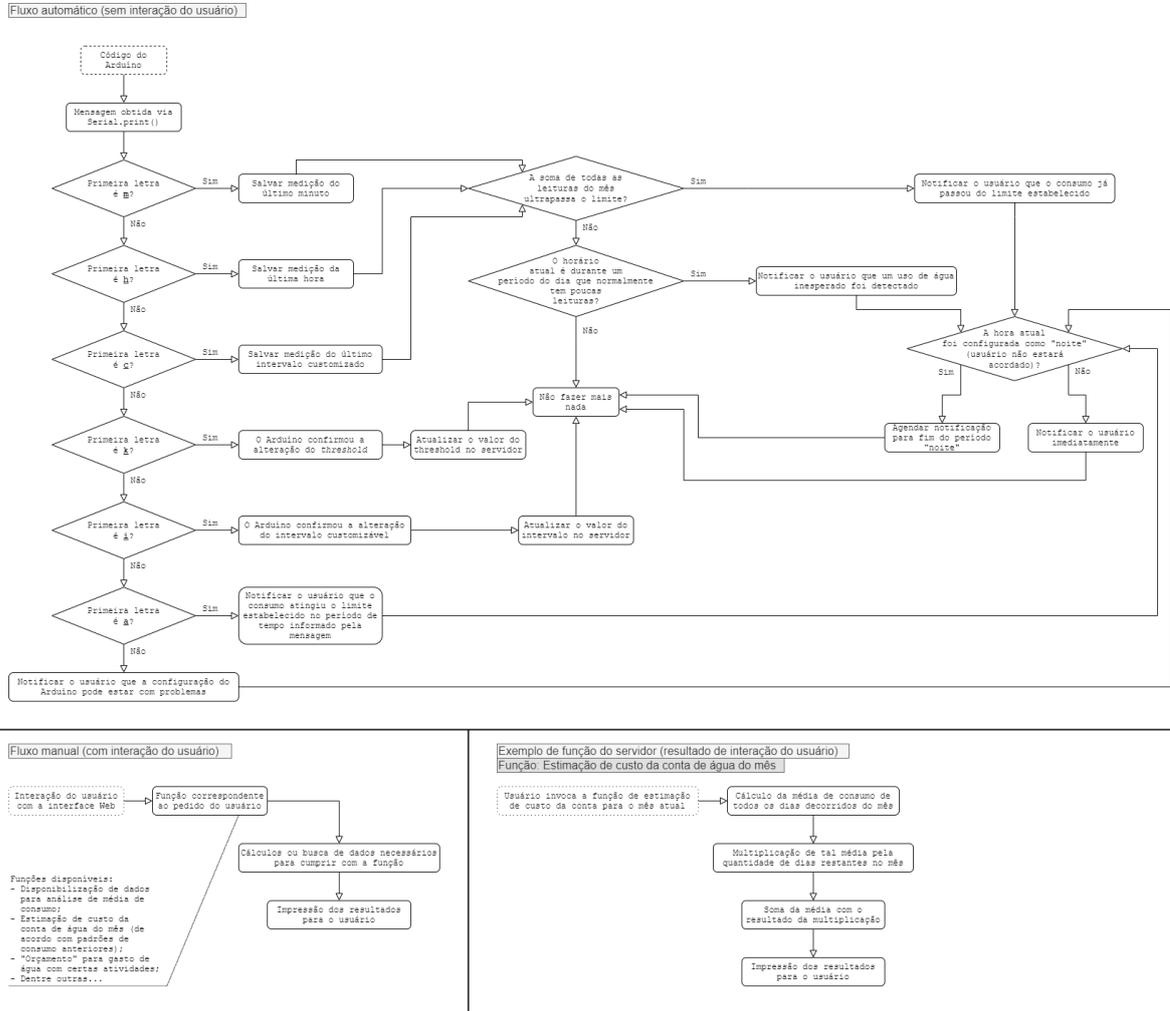
APÊNDICE B – FLUXOGRAMA DO ARDUÍNO

Figura 40 – Fluxograma do código do Arduino



APÊNDICE C – FLUXOGRAMA DO SERVIDOR

Figura 41 – Visão das interações do Arduino com o servidor da proposta



Fonte – Produção original

APÊNDICE D - ARTIGO NO FORMATO SBC

Sistema de gerenciamento autônomo de consumo de água para casas inteligentes utilizando IoT

Pedro A. B. da Côte¹

¹Departamento de Informática e Estatísticas
Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC - Brasil

pedroabcorte@gmail.com

Abstract. Residential environments can be automated using the Internet of Things, which allows multiple tasks to be handed over to computers. One of these tasks is to check the water consumption of a house. Excessive water usage or water leaks can occur in a house, and these problems can go undetected for a long time. Currently, the most common way to avoid or remedy such problems is to pay attention to the water meter's numbers and to know how to interpret them. This TCC proposes a system that automates the verification and interpretation of water consumption in a home, thus helping to reduce unnecessary expenses and to identify problems (such as leaks) more quickly and effectively. The system includes a sensor that makes water measurements and sends them to a server programmed in an Arduino, which counts the total consumption for several periods and also makes all the necessary calculations to identify possible problems.

Resumo. Ambientes residenciais podem ser automatizados com o uso da Internet das Coisas, o que permite que várias tarefas sejam colocadas nas mãos de computadores. Uma dessas tarefas é a verificação de consumo de água de uma residência. Gastos exagerados ou vazamentos podem ocorrer em uma casa, e esses problemas podem passar despercebidos por muito tempo. Atualmente, a forma mais comum de evitar ou remediar tais problemas é ficar atento aos números do medidor de água, e saber interpretá-los. Este TCC propõe um sistema que automatiza a verificação e interpretação do consumo de água de uma casa, assim ajudando a reduzir gastos desnecessários e a identificar problemas (como vazamentos) de forma mais rápida e eficaz. O sistema inclui um sensor que faz as medições de água e as envia para um servidor programado em um Arduino, o qual contabiliza o consumo total para vários períodos e também faz todos os cálculos necessários para identificar possíveis problemas.

1. Introdução

A evolução das tecnologias de comunicação e de aparelhos inteligentes trouxe consigo a possibilidade de automatizar diversas tarefas em diversas áreas de trabalho ou pesquisa. O advento de sistemas *Fog* permitiu que tais tarefas não fossem limitadas apenas ao mesmo espaço onde a computação ocorre, trazendo soluções inovadoras para lugares desprovidos de acesso a computadores. [Alrawais et al. 2017] explica que a computação *Fog* permite uma nova geração de serviços na borda da rede. A única coisa necessária é uma conexão.

Diversas soluções tem como foco a automatização de ambientes residenciais, com o objetivo de facilitar a vida das pessoas, como pode ser observado no trabalho de [Bhilare and Mali 2016]. Desde luzes que acendem com um comando verbal, até sistemas que controlam o uso de eletricidade de eletrodomésticos para reduzir gastos, todo o tipo de ideias podem apresentar viabilidade para casas inteligentes.

Um dos gastos que pode ser controlado de forma autônoma em residências é o de recursos hídricos, um dos bens mais preciosos da raça humana. Este trabalho busca propor um sistema que colete dados de consumo de água, guarde-os, analise-os automaticamente, e que tenha a capacidade de avisar os usuários sobre excesso de uso ou vazamentos com antecedência, fazendo um gerenciamento autônomo de consumo de água. Todo esse processo será facilitado pelo uso da Internet das Coisas (*Internet of Things*, ou IoT) e dispositivos inteligentes, garantido aos usuários um acesso fácil e rápido aos dados obtidos.

Hoje, temos que nos preocupar com todo o tipo de problema, e muitas vezes ficamos sobrecarregados. Com isso, a automatização nos oferece uma tranquilidade, pois a preocupação com alguma atividade é colocada em uma máquina que cuida de tudo sozinha, e apenas nos notifica quando precisa de intervenção humana. Uma das preocupações em casas é a possibilidade de vazamentos de água, que por vezes se manifestam apenas no final do mês, por meio de contas altas. Um sistema poderia ajudar a evitar que vazamentos passem despercebidos por muito tempo, e este é o principal motivo da proposta deste trabalho.

Este TCC está organizado em 7 capítulos. Esta é a primeira seção, e em seguida vem a seção dos conceitos fundamentais. A terceira seção contém os dados da revisão bibliográfica realizada para fundamentar este trabalho, e alguns dos trabalhos foram selecionados e analisados em mais detalhes na sequência. Na quarta seção, a proposta deste trabalho é apresentada, e a mesma é desenvolvida na quinta seção. A sexta seção apresenta os resultados dos testes feitos e a sétima seção conclui o trabalho.

1.1. Motivação e Justificativa

Um ponto se destaca como a principal motivação para o desenvolvimento deste trabalho: vazamentos e contas altas de água sem nenhum tipo de aviso prévio já foram problemas experienciados pelo autor. Ao avaliar as possibilidades de automação e gerenciamento que a IoT possibilita, foi possível chegar à conclusão que, para ambientes residenciais, é possível controlar todo o tipo de recurso que é consumido, como energia e água.

Este projeto visa utilizar automação de gerência para que problemas com vazamentos possam ser evitados ou, pelo menos, para que não sejam tão caros.

1.2. Objetivos

O objetivo principal deste trabalho é propor e desenvolver um sistema de gerenciamento autônomo de medidores de consumo de água de uma residência, que realizará a verificação de níveis de consumo de água em casas, focando na economia de água, bem como facilitar o processo de obtenção de tais dados.

1.2.1. Objetivos específicos

- Automatizar e armazenar as medições de água de uma residência
- Detectar vazamentos com base em padrões de consumo
- Notificar o usuário de anomalias ou limites de consumo alcançados
- Facilitar a obtenção das medições de volume de água

1.3. Método de pesquisa

A princípio, foi feita uma pesquisa e avaliação de trabalhos já publicados que são relevantes ao tema deste TCC, com o objetivo de alcançar um melhor entendimento dos conceitos envolvidos, desde as funcionalidades e estruturas de sistemas IoT até a utilização e instalação dos sensores necessários.

2. Conceitos Fundamentais

2.1. Fog Computing

A computação em névoa, ou *Fog Computing*, permite descentralizar o gerenciamento e o processamento de dados, executando tais tarefas na própria rede local dos sensores e atuadores implantados [Iorga et al. 2018]. Este modelo ajuda a liberar recursos da nuvem, reduzindo a dependência de sistemas para todas as operações, além de resultar em um impacto positivo à qualidade de serviço e reduzir a latência da solução provida [Alrawais et al. 2017].

O conceito *Fog Computing* pode ser definido como um modelo de computação separado em camadas, onde os recursos computacionais são escaláveis, ou seja, podem ser alterados para fornecer mais (ou menos) poder computacional, conforme as necessidades da rede. Na Figura 1, um modelo de *Fog Computing* é definido por [Iorga et al. 2018], indicando as separações das camadas. Na camada superior situam-se os grandes centros de processamento, que possuem alto poder computacional e são responsáveis pela computação na nuvem (*Cloud Computing*). A camada intermediária, denominada *Fog*, encontra-se entre a *Cloud* e os dispositivos IoT.

Na camada inferior, estão presentes os nós, que são, essencialmente, computadores que recebem e processam os dados dos sensores e atuadores da rede. Os nós *Fog* (tanto físicos quanto virtuais) são sensíveis ao contexto e utilizam todos um sistema comum para gerenciar e comunicar os dados. Eles podem ser organizados em *clusters*, com o objetivo de dar suporte ao isolamento (organizados de forma vertical), de dar suporte à federação (organizados de forma horizontal), ou até mesmo para reduzir a latência (organizados de forma relativa à latência entre os nós e os dispositivos da rede). Este modelo também pode fornecer, para os sensores, conectividade a serviços centralizados [Iorga et al. 2018].

2.2. Internet of Things

Um dos maiores desafios encontrados ao tentar implantar um sistema de Internet das Coisas (*Internet of Things*, IoT) é a complexidade no gerenciamento de dados que são gerados por sensores e atuadores presentes na rede. Os sistemas IoT tradicionais, que utilizam majoritariamente processamento de dados na nuvem, sofrem principalmente com alta latência e heterogeneidade (de dispositivos e tipos de dados) [Alrawais et al. 2017].

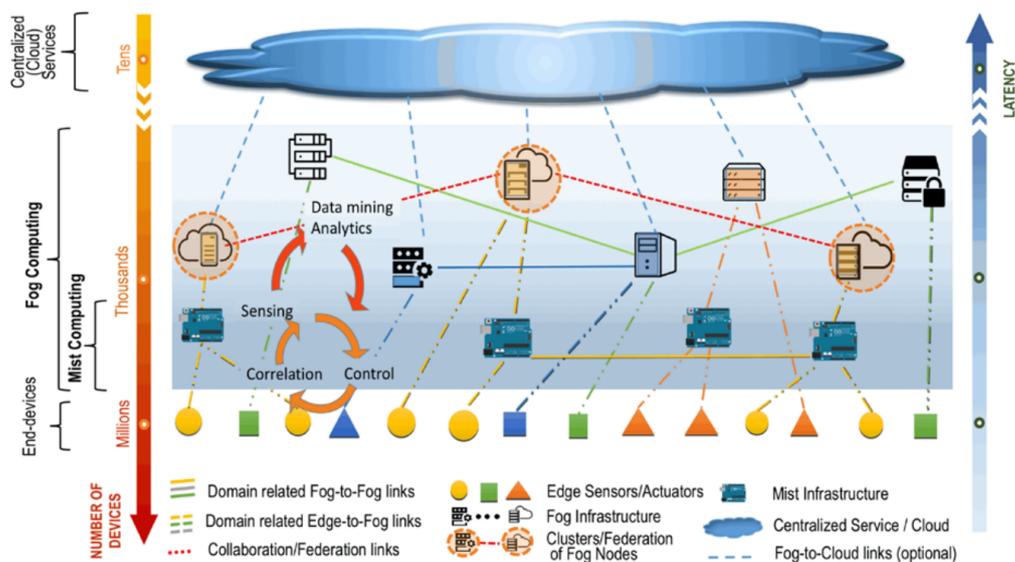


Figura 1. Descrição de camadas de um sistema Fog
Fonte: [Iorga et al. 2018]

Segundo os autores [Alrawais et al. 2017], os aparelhos presentes em uma IoT possuem características inerentemente desafiadoras (como baixo poder computacional, baixas taxas de transferência e armazenamento de dados limitado), as quais afetam a experiência do usuário final e também a qualidade do serviço (*Quality of Service, QoS*).

Um dos usos mais discutidos para um sistema IoT, com uso de computação na névoa, é a automatização de ambientes de moradia, em especial casas. O sistema proposto por [Bhilare and Mali 2016] utiliza uma aplicação *web* (funcionando na nuvem) em conjunto com o sistema IoT para controlar aparelhos inteligentes dentro de uma casa, além de utilizar tal aplicação para monitorar os gastos com eletricidade. O objetivo é permitir que o usuário faça *login* no sistema para checar o consumo elétrico em tempo real e manualmente controlar seus aparelhos locais.

Um aspecto importante a ser considerado ao implantar um sistema IoT é o consumo de energia (corrente ou bateria) dos dispositivos presentes na rede (tanto os *smart devices* quanto os nós da névoa). Para a maioria dos sistemas IoT, é fundamental que os dispositivos implantados possam continuar ligados pela maior quantidade de tempo possível, o que pode se tornar difícil caso o consumo das baterias destes não sejam gerenciadas de forma efetiva [Sampaio et al. 2019]. Esta necessidade fica realçada ainda mais com as previsões que haverá mais de 50 bilhões de dispositivos conectados em 2020 [Iorga et al. 2018].

2.3. Medidores inteligentes

Os medidores inteligentes funcionam da mesma forma como os medidores analógicos funcionam, com a principal diferença sendo que os inteligentes convertem as medidas para dados digitais, que podem ser utilizados por computadores ou servidores. Estes medidores também possuem conectividade com alguma rede local, permitindo comunicação com os dispositivos que farão uso dos dados, e classificando-os como dispositivos IoT.

Para realizar a medição automática de consumo de água, um sistema qual-

quer necessitaria de um desses medidores de água inteligentes. De acordo com [Masia and Erasmus 2013], não existe uma definição universal para um medidor destes, mas sim definições com base nas funcionalidades adicionais, as quais vão além da medição e apresentação do consumo de água. De acordo com os autores, existem algumas grandes categorias de acordo com o nível de sofisticação, sendo estas:

- Leitura Automatizada de Medidores (*Automated Meter Reading (AMR)*)
- Gerenciamento Automatizado de Medidores (*Automated Meter Management (AMM)*)
- Medição de Intervalo com Gerenciamento Automatizado de Medidores (*Interval Metering with Automated Meter Management (AMM)*)
- Medidores de Pré-pagamento (*Prepayment Meters (PM)*)
- Infraestrutura Avançada de Medição (*Advanced Metering Infrastructure (AMI)*)

Na Tabela 1 são comparados os benefícios dos medidores inteligentes.

Tabela 1. Comparação dos benefícios de medidores inteligentes diferentes

Benefícios	AMR	AMM	Medição AMM	PPM	AMI
A leitura manual do medidor não é necessária, portanto reduz o custo e as dificuldades práticas da leitura do medidor	X	X	X	X	X
Permite produzir faturas com leituras reais, em vez de faturas imprecisas estimadas	X	X	X	X	X
O cliente pode mudar de fornecedor rapidamente, pois estão disponíveis leituras precisas do medidor	X	X	X	X	X
Detectar e notificar fraudes quando um medidor for violado	X	X	X	X	X
Visitas e reajustes manuais do medidor não são necessários quando o preço e/ou as tarifas mudam		X	X	X	X
Conscientizar o cliente quanto à energia, o custo e a eficiência, para que o consumo seja ajustado para reduzir o custo			X	X	X
Instalações aprimoradas para clientes pré-pagos				X	X
Poderia ajudar a evitar apagões em larga escala através de derramamento de carga controlado durante eventos críticos de pico					X

Atualmente se faz necessário que a sociedade revise as políticas de consumo de água para garantir o uso adequado dos recursos hídricos. [Masia and Erasmus 2013] explicam as melhores práticas de cálculo de consumo de água, de acordo com a Associação Internacional da Água (*International Water Association, IWA*). Na Figura ?? considera-se uma modificação nos cálculos mencionados para refletir as circunstâncias da África do Sul.

3. Trabalhos Relacionados

3.1. Revisão Bibliográfica Sistemática

Com o objetivo de salientar a necessidade e importância da pesquisa deste trabalho, uma revisão sistemática foi desenvolvida, utilizando três ferramentas de pesquisa acadêmica, sendo estas o Google Scholar, o IEEE Xplore™ e o ScienceDirect. As palavras-chave (em

Inglês e sem aspas) foram pesquisadas nestas ferramentas tanto separadas quanto agrupadas, conforme mostrado na Tabela 2, que apresenta o número de ocorrências destas em cada ferramenta. Os artigos pesquisados foram limitados para apenas aqueles publicados após o ano 2015.

Tabela 2. Resultados da revisão sistemática bibliográfica (30 de Abril de 2020)

Palavras-chave	Google Scholar	IEEE Xplore™	ScienceDirect
AM + IoT + FC	4 480	5	214
AM + IoT + WM	1 870	4	317
AM + IoT + SH	4 750	9	526
AM + FC + WM	1 290	0	101
AM + FC + SH	2 150	1	135
AM + WM + SH	5 760	3	885
IoT + FC + WM	6 210	8	266
IoT + FC + SH	13 800	63	617
IoT + WM + SH	17 100	25	1 075
FC + WM + SH	9 230	2	202
AM + IoT + FC + WM	1 530	0	53
AM + IoT + FC + SH	2 150	0	123
AM + IoT + WM + SH	7 350	2	192
AM + FC + WM + SH	714	0	45
IoT + FC + WM + SH	14 500	2	167
AM + IoT + FC + WM + SH	714	0	41

As palavras foram identificadas pelas seguintes siglas:

1. AM - *Autonomic Management*
2. IoT - *Internet of Things* (pesquisada como “iot”)
3. FC - *Fog Computing*
4. WM - *Water management*
5. SH - *Smart homes*

Os dados acima ressaltam uma clara falta de artigos nos últimos 5 anos que tratem de *Fog Computing*, em especial nos casos envolvendo gestão autônoma de sistemas (palavra-chave *Autonomic Management*) ou gestão de água (palavra-chave *Water Management*). Nos resultados da ferramenta IEEE Xplore™, quando as três palavras-chave foram pesquisadas juntas, as mesmas não retornaram qualquer resultado, reforçando a necessidade de novas pesquisas envolvendo tais assuntos.

3.2. Análise de consumo de água em áreas urbanas

[Ji et al. 2015] acreditam que os padrões consumo de água de pessoas em áreas urbanas depende muito da estrutura da família, do tipo de trabalho do indivíduo ou do seu estilo de vida. A forma tradicional de medir o consumo de água de uma residência é pela leitura manual de um medidor analógico, que oferece uma precisão de $0,1\text{m}^3$, uma vez por mês. Com os avanços da IoT, a proliferação dos medidores inteligentes cresceu, os quais oferecem uma precisão de medição de $0,001\text{m}^3$ a cada 15 minutos.

Para tirar proveito dos dados coletados pelos *Smart Meters*, [Ji et al. 2015] propõem um algoritmo de agrupamento não supervisionado para analisar o consumo de água de residências em áreas urbanas. Esse algoritmo é adaptável na escala de tempo diário e pode dividir os residentes por estrutura familiar, tipo de trabalho ou estilo de vida. A pesquisa foi realizada com dados experimentais do volume de consumo de 20 usuários, coletando dados a cada 15 minutos com uma precisão de $0,001\text{m}^3$. A Tabela 3 exibe o agrupamento de horários em 8 períodos para a realização de medidas individuais.

Tabela 3. Intervalos de tempo de [Ji et al. 2015]

Intervalo de tempo	Tipo
23:00 - 2:00	Meia noite
2:00 - 5:00	Madrugada
5:00 - 8:00	Manhã
8:00 - 11:00	Horário de trabalho
11:00 - 14:00	Horário de almoço
14:00 - 17:00	Horário de trabalho (tarde)
17:00 - 20:00	Horário de jantar
20:00 - 23:00	Noite

3.3. Smart Home com medição eletrônica em tempo real

[Bhilare and Mali 2016] focam sua atenção no desenvolvimento de um sistema *web* que será utilizado para controlar aparelhos e sensores IoT. A ideia principal é facilitar o acesso à informação de medidores inteligentes e outros dados para os usuários finais, oferecendo conforto, eficiência e ajudando a economizar energia.

Eles propõem um sistema distribuído de controle, através do uso de uma aplicação e páginas *web*. Este sistema controla todos os recursos de cobrança da aplicação *web* funcionando como uma aplicação *Cloud*, além de permitir os usuários a controlar seus eletrodomésticos de qualquer lugar, bem como verificar o gasto elétrico dos mesmos.

O sistema num todo opera nos princípios do “Modelo de criação de contexto em três níveis”. A aquisição de dados e processamento de sinais são o baixo nível de contexto, sendo que o contexto de alto nível é usado para gerar consciência da situação. Os sensores são os transdutores que detectam as características do ambiente. O fluxo do sistema pode ser descrito nos seguintes passos:

1. Começo
2. Sensor coleta os dados com o estado atual dos aparelhos
3. Dados são enviados ao controlador
4. Controlador manda os dados para o servidor *web* através do protocolo HTTP
5. Dados são armazenados na *database* SQL
6. Dados (junto do estado dos aparelhos) é mostrado através de interface *web*
7. Caso o estado não esteja atualizado, volta-se ao passo 2
8. Os valores atualizados são enviados para a *database*, e a mesma é atualizada
9. Os valores atualizados são então enviados para o controlador através do protocolo HTTP
10. O controlador recebe o comando e realiza a ação de controlar os aparelhos
11. Fim

A interface da aplicação *web* é utilizada para controlar e monitorar os dados. O usuário pode encontrar quaisquer avisos urgentes na seção de alertas da página principal. Também existem links para os *websites* de vários provedores de serviços, com os detalhes de suas taxas de consumo de eletricidade. O usuário deve logar com senha para poder monitorar e controlar seus eletrodomésticos.

3.4. Gerenciamento de água baseado em IoT

[Rajurkar et al. 2017] apresentam um projeto onde o foco é monitorar somente o uso de água. A metodologia está focada no uso de um dispositivo IoT, composto por um Arduíno e um sensor de fluxo de água, o qual irá medir o consumo de água em litros por hora. As medições obtidas são enviadas a partir do dispositivo IoT para um servidor na *Cloud*, onde as informações serão disponibilizadas para aplicativos *mobile*, com o objetivo de manter um usuário ou uma empresa informados caso ultrapassem o limite de uso de água (e quais os custos adicionais caso isso ocorra).

O projeto visa monitorar o consumo de água individual por casa, enviando os dados para um servidor na *Cloud*. Tais dados podem ser acessados posteriormente através de aplicação móvel, a qual ajudará o usuário a verificar se o mesmo ultrapassou o limite de consumo mensal de água, além de informar o custo extra. [Rajurkar et al. 2017] acreditam que, quando os recursos hídricos forem cobrados de forma adequada, incluindo os usos em excesso, as pessoas não irão tirar mais vantagem da água.

3.5. Gerenciamento de energia baseado em Edge para Smart Homes

De acordo com [Xia et al. 2018], a maioria dos sistemas de gerenciamento de energia residencial (*Home Energy Management System*, ou HEMS) existentes são projetados principalmente para economizar energia e satisfazer o conforto dos clientes, controlando aparelhos inteligentes remotamente, por meio de um método de reconhecimento de local, e

alterando de forma inteligente os pontos de ajuste de aquecimento, ventilação e ar condicionado (*Heating, Ventilation and Air Conditioning*, ou HVAC).

A motivação para o artigo é propor um *framework* de gerenciamento de energia baseada em *Edge*, que permita baixo custo de eletricidade, gerando estratégias de programação ideais para os aparelhos domésticos, e maximizando a utilização de energia solar em comunidades residenciais. Além disso, dispositivos de *Edge* fáceis de obter podem fornecer recursos de computação suficientes para agendamento de dispositivos e tomada de decisões na implantação de energia, basicamente diminuindo o custo de investimento de capital em grande medida.

Existem várias técnicas de resposta à demanda (*Demand Response*, ou DR), e [Xia et al. 2018] aprofundam-se principalmente na implementação de *Load Shifting*, pelo qual sistemas sem restrição de tempo podem alterar o funcionamento de dispositivos para fora do horário de pico, de forma a reduzir a demanda de uma casa, sem comprometer o conforto dos usuários.

O sistema no geral é composto por alguns componentes:

- O módulo *energy generation prediction*, nos estágios iniciais, carrega dados de localização e previsão do tempo, com o intuito de maximizar a produção de energia de painéis fotovoltaicos
- O módulo *solar energy management*, que controla o uso da energia solar, maximizando seu uso quando a geração da mesma é maior que a demanda de carga, além de guardar todo o excesso em baterias
- O módulo *energy disaggregation*, que monitora e, utilizando uma abordagem NILM sensível à localização, deduz as atividades do usuário ao longo de um dia típico
- O módulo *classifier*, que classifica a carga dos eletrodomésticos inteligentes em agendáveis e não agendáveis. Para descrever flexibilidade quantitativamente, um coeficiente é automaticamente gerado para cada eletrodoméstico com base nos dados do módulo *energy disaggregation*, além das preferências pessoais do usuário
- O módulo *appliances scheduling*, que agenda os horários de início dos eletrodomésticos agendáveis de acordo com todas as informações coletadas pelo módulo *classifier*

3.6. Sistema autônomo para monitoramento de consumo de água em apartamentos

Os autores [Aggarwal et al. 2019] acreditam que o custo de instalação de hidrômetros separados é muito alto. Em seu artigo, eles focam na criação de um dispositivo inteligente de baixo custo, o qual pode ser usado para coletar os dados de consumo de água. Tal dispositivo foi criado com o Arduíno, juntamente com alguns sensores e módulos para coletar vários dados de consumo de água de diferentes apartamentos.

O sistema de monitoramento automatizado de baixo custo poderá rastrear e monitorar automaticamente o consumo de água de diferentes apartamentos em um prédio, e exigirá a instalação de um dispositivo que inclua sensores com localização geográfica, para que ele possa rastrear dados diferentes em diferentes locais do edifício, com fluxo contínuo de dados e análises, que podem ajudar no monitoramento e no controle de todo o processo com mais eficiência. Um dispositivo fabricado com Arduíno e sensores para

coletar esses dados enviar ao servidor para análise, com a ajuda de uma rede neural artificial, poderá gerar resultados com uma precisão muito alta. O sistema enviará notificações de consumo excessivo de água aos usuários com a ajuda do aplicativo *Web* e *Android*.

3.7. Pesquisa sobre desafios do gerenciamento de água na agricultura

Neste artigo, [Saad et al. 2020] apresentam uma pesquisa sobre o uso de tecnologias inovadoras para desenvolver um conceito inteligente na regulamentação do uso da água no domínio agrícola. Os mesmos começam explicando que existem vários métodos que ajudam a preservar e proteger as fontes de água, como a construção de barragens para armazenar água da chuva, a dessalinização da água do mar, o tratamento de águas residuais e o monitoramento de oleodutos para detectar danos ou vazamentos. Para aprimorar esses métodos e torná-los mais inteligentes, os autores mencionam os principais paradigmas de pesquisa que são utilizados, sendo estes os sistemas ciber-físicos (*Cyber-Physical Systems*, ou CPS), as redes de sensores sem fio (*Wireless Sensor Networks*, ou WSN), a IoT e outras tecnologias em nuvem. Esses paradigmas entraram no setor agrícola como forma de criar um sistema automatizado e integrado. Sensores ambientais (por exemplo, sensores de umidade, pressão e temperatura) são implantados nas infraestruturas CPS e WSN. Esses sensores geram dados espaço-temporais maciços e heterogêneos, que são armazenados e processados em larga escala. Essas tecnologias também permitem reduzir o desperdício de água, pois é importante para a sustentabilidade geral de recursos limitados de água doce. A implementação de soluções inteligentes para o monitoramento da água oferece a possibilidade de aprimorar a produção agrícola e facilitar o gerenciamento, porque ter a capacidade de monitorar a água através de sensores fornece aos agricultores o poder de aumentar o crescimento das culturas. O uso desses sensores fornece informações precisas e em tempo real sobre os diferentes parâmetros relacionados à água, para que os agricultores possam intervir efetivamente no momento certo.

Os quatro principais desafios da gestão de água na agricultura moderna são:

- A reutilização da água e monitoramento da poluição da água;
- O monitoramento de dutos de água;
- A irrigação por água; e
- A disponibilidade de água potável para o gado.

3.8. Distribuição e monitoramento de tubulações com IoT para cidades inteligentes

[Narayanan and Sankaranarayanan 2019] afirmam que a integração da tecnologia de informação e comunicação no sistema de distribuição de água resolverá muitos problemas no sistema de distribuição de água atual e tradicional. A formação de redes inteligentes de distribuição de água (*Smart Water Distribution Network*, ou SWDN) para cidades inteligentes é projetada com reservatórios de armazenamento, estação de bombeamento de reforço, hidrantes e linhas de serviço ao consumidor em mente, e a redundância da rede é fornecida via redes e loops de água inteligentes. Essas redes inteligentes de água são integradas com tecnologias avançadas para detectar o fluxo, a pressão e o consumo de água, a fim de fornecer uma distribuição eficiente de água e aumentar a eficiência do sistema de distribuição para atender a demanda. A gestão da distribuição de água faz parte da iniciativa de *Smart Cities* e ajuda a prever a demanda de abastecimento, além de também cobrir a operação de monitoramento e proteção da rede de distribuição de água por meio do SCADA (*Supervisory Control and Data Acquisition*, ou Controle de Supervisão e

Aquisição de Dados). [Narayanan and Sankaranarayanan 2019] propõem um sistema de gerenciamento de distribuição de água baseado em IoT e monitoramento de integridade de dutos integrado com computação em nuvem e neblina.

A automação da distribuição de água tem a capacidade de misturar a automação em áreas locais e remotas para o controle de válvulas de vazão, pressão, etc. A tomada de decisão centralizada permite soluções econômicas, tornando o sistema eficiente e coeso enquanto opera uma arquitetura para o sistema de distribuição de água. Essa automação também oferece soluções como diagnóstico de falhas, detecção de falhas, regulação e controle da variação da pressão de fluxo, isolamento e restauração de tubulações na rede, aumentando a eficiência e melhorando a eficiência e desempenho do sistema de distribuição de água.

3.9. Sistema de gerenciamento de água baseado em IoT

Outro artigo, dos autores [Jeurkar et al. 2020], apresenta um sistema onde os três componentes principais, que são o medidor de vazão, o sensor de temperatura e o sensor de nível, têm interface com um módulo WiFi por meio de um microcontrolador. O medidor de vazão é instalado dentro do tubo através do qual a água flui em direção ao tanque. Isso ajuda a detectar o fluxo de água para obter o consumo total de água.

Um dos módulos essenciais deste sistema é um sensor de nível baseado em capacitância. A capacitância de um capacitor é diretamente proporcional à área das “placas”. Um eletrodo é colocado no centro do tanque, e sua altura é igual à do tanque. O eletrodo atua como uma placa do capacitor e a água no tanque atua como a segunda placa. Conforme o nível da água sobe, a área da placa do capacitor aumenta, aumentando assim a capacitância do sensor. Não há peças móveis expostas aos fluidos. Este é um tipo de sensor muito sensível e adequado para indicação ou controle contínuo.

O sistema é construído a partir de vários componentes, e os mais relevantes para este trabalho são:

1. Sensor de fluxo de água: este sensor é instalado dentro do tubo para medir a taxa de fluxo de água e calcular a quantidade de água que flui através do tubo.
2. Sensor de nível: este sensor indica o nível da água em um determinado instante.
3. Microcontrolador: é usado para medir a capacitância e calibrar o sensor de nível.
4. Módulo WiFi: este módulo é usado para exibir os dados recebidos do microcontrolador na plataforma IOT.

3.10. Desafios em sistemas de monitoramento

Diante de todos os estudos e artigos analisados, é possível concluir que existem muitas formas de se analisar e estudar as medições de consumo de água em casas, apartamentos e até mesmo fazendas, porém, há apenas uma maneira prática de coletar tais dados, que é através do uso de medidores presentes ao longo da tubulação no sistema de distribuição de água utilizado. Estes medidores podem ser utilizados em várias quantidades, mas tipicamente apenas um é o suficiente (especialmente para casas ou apartamentos). As medidas obtidas por estes aparelhos podem variar em precisão das medições ([Ji et al. 2015]) e nos métodos usados para contabilizar tais medições (como, por exemplo, utilizando ciclos do medidor, como observado por [Rajurkar et al. 2017] e [Jeurkar et al. 2020]), que

precisam ser convertidas para medidas úteis aos programas ou algoritmos, os quais visam automatizar o resto do processo.

Para obter as medições, todos os artigos fazem uso de sensores, os quais transmitem os dados obtidos através da IoT ou por computação em névoa (*Fog Computing*), método que também é observado em todos os artigos. O monitoramento é inteiramente feito pelo sistema de forma automática, mas nem todos os sistemas são similares. Os trabalhos de [Ji et al. 2015] e [Xia et al. 2018] apresentam algoritmos para trabalhar com os dados, sendo, respectivamente, para análise e automatização do sistema. Contudo, apenas [Xia et al. 2018] apresenta uma *framework* para implementar tal sistema. Quatro dos oito trabalhos propõem um aplicativo para facilitar a análise dos dados por parte dos usuários, porém apenas os aplicativos de [Bhilare and Mali 2016] e [Xia et al. 2018] oferecem algum controle do sistema através do aplicativo, permitindo que os usuários controlem, respectivamente, o estado de diversos eletrodomésticos ou definam “*guidelines*” para o algoritmo que automatiza o sistema (respectivamente). O trabalho de pesquisa de [Saad et al. 2020] é o único que busca soluções para o ambiente de *Smart Agriculture*, enquanto o trabalho de [Narayanan and Sankaranarayanan 2019] é o único com foco em *Smart Cities*, já que [Ji et al. 2015] faz uma análise apenas de áreas urbanas em geral. Todos os artigos focam em soluções que envolvem água e seus usos, com a única exceção sendo o artigo de [Xia et al. 2018], o qual foca seus esforços para trabalhar com energia e seus usos. O trabalho de [Bhilare and Mali 2016] é o único que trabalha com água e energia simultaneamente.

4. Proposta do sistema de gerenciamento de água

Automação tem a capacidade de facilitar a vida de muitas pessoas, reduzindo diversas atividades a simples avisos ou notificações, que servem para alertar que as mesmas foram concluídas. Tendo em vista todos os conteúdos explorados até o momento, com a evolução tecnológica cada vez mais acelerada e presente em nosso cotidiano, desenvolver um sistema que ajude diversos tipos de usuários a acompanhar e controlar seus gastos de um dos recursos mais importantes para o ser humano parece ser inevitável.

Muitas coisas podem dar errado com os sistemas de distribuição de água que são encontrados em casas, apartamentos ou fábricas, desde tubos furados ou quebrados, até torneiras que ficam abertas por muito tempo ou eletrodomésticos que consomem água em demasia. É muito fácil as pessoas demorarem a perceber qualquer um desses problemas, causando desperdício que poderia ser reduzido ou, até mesmo, evitado.

Para este trabalho, é proposto um sistema que combina um medidor de volume de água com um servidor, tendo por objetivo monitorar e automaticamente avisar o usuário quando há algo de errado, ou seja, o usuário deve ser notificado quando o consumo de água está perto do limite mensal ou se está muito alto quando comparado a medições de dias anteriores, ou até quando um vazamento pode estar acontecendo. Para tanto, a medição de água é frequentemente atualizada e salva em um servidor local, o qual pode verificar os dados para alertar se o consumo de água está perto do limite mensal, ou se está muito alto quando comparado às medições dos dias anteriores.

O sistema inclui um sensor que pode ser colocado em qualquer duto de água da residência (de preferência o de entrada), que pode ser observado na Figura 2. Este dispositivo faz medições de água várias vezes por hora (os intervalos são configuráveis) e

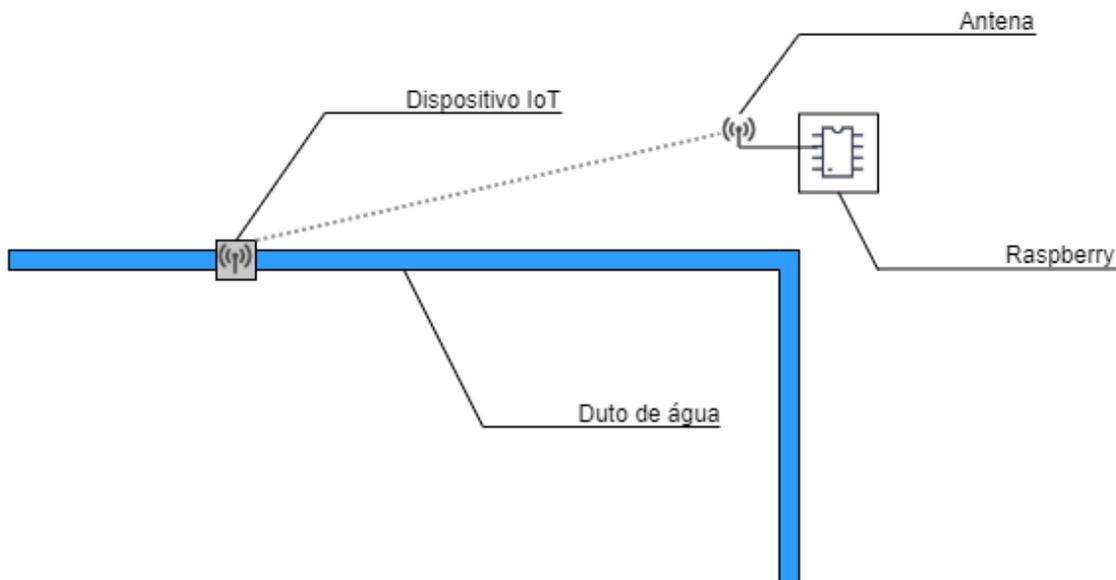


Figura 2. Visão geral da proposta
Fonte: Produção original

envia essas medições para um servidor local, através de uma rede sem fio. A combinação do servidor local, do dispositivo que faz as medições e da rede, a qual permite a conexão e interação entre os aparelhos mencionados, resulta em uma rede da Internet das Coisas (IoT). O paradigma IoT é a *cornerstone* (ou pedra angular) que permite que sistemas automatizados atinjam seu potencial.

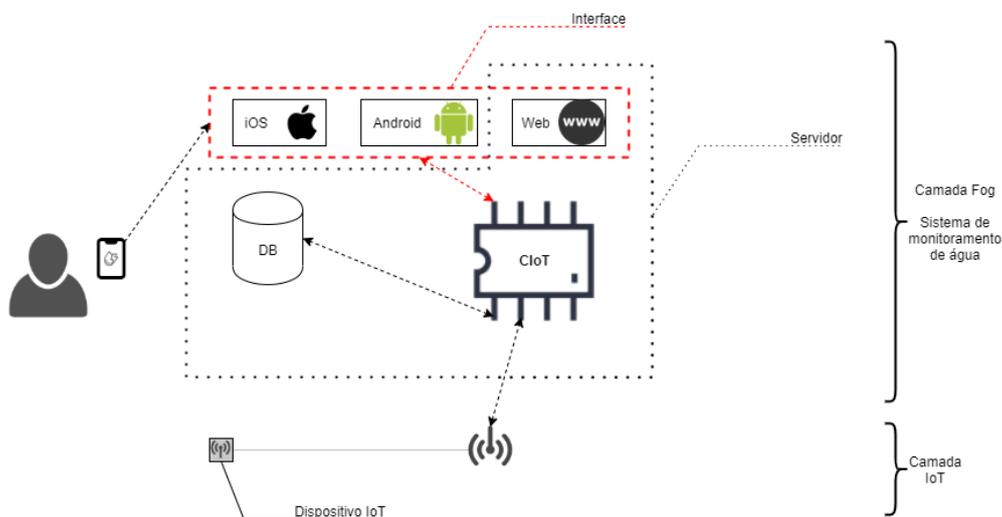


Figura 3. Visão do Arduino e sua implementação na proposta
Fonte: Produção original

O servidor roda o sistema CIoT (ou *Central Internet of Things*, desenvolvido no Laboratório de Redes e Gerência (LRG) da UFSC) que controla vários subsistemas e dispositivos de uma casa inteligente, além de disponibilizar funcionalidades para o sistema de monitoramento de água, como avisos de consumo elevado (ou vazamento),

disponibilização de dados para análise de média de consumo, estimação de custo da conta de água do mês de acordo com padrões de consumo anteriores, “orçamento” para gasto de água com certas atividades, dentre outras. Exemplificamos esse servidor sendo executado em um Raspberry Pi 3B, detalhado na Figura 3, que contém uma *database* onde são armazenados os dados do medidor, de onde é possível comparar dados novos com anteriores e manter um histórico dos mesmos. Também é possível acessar uma aplicação via *website* hospedado no próprio Raspberry, através de qualquer computador ou smartphone conectado à internet.

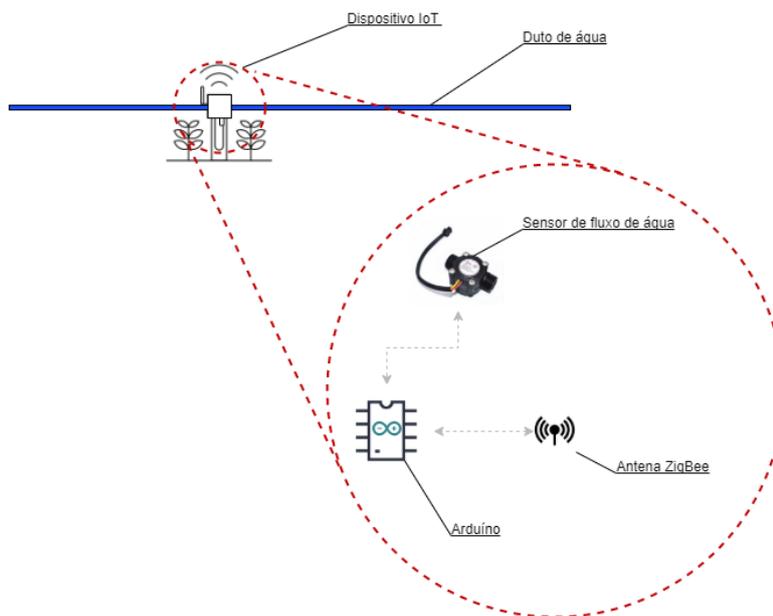


Figura 4. Visão do dispositivo IoT e sua implementação na proposta
Fonte: Produção original

O dispositivo IoT, exemplificado na Figura 4, é composto por um Arduino e um sensor de fluxo de água. O Arduino lê as medições do sensor a cada intervalo de tempo configurado pelo usuário (por padrão, este intervalo é de 1 minuto). Não é recomendado configurar um intervalo de leitura muito curto, pois o objetivo do dispositivo é ser instalado em locais que costumam não possuir nenhum tipo de tomada elétrica por perto, o que significa que o sistema irá depender de uma bateria na maioria dos casos. A comunicação do Arduino com o servidor é feita através de uma conexão sem fio, utilizando o protocolo *ZigBee*, para ser compatível com a CIoT. Todos os dados obtidos pelo Arduino devem ser apenas enviados (em formato legível) para o servidor, sendo que o processamento lógico dos mesmos deverá ser feito pelo servidor (como mencionado anteriormente).

5. Desenvolvimento de um protótipo

Para validar a proposta deste trabalho, foi desenvolvido um pequeno protótipo, composto por um Arduino e por um sensor de fluxo de água simples. Este protótipo não incorpora o aspecto de conexão sem fio da proposta, já que um módulo para testes (ou seja, uma antena) não estava disponível. O mesmo foi instalado na boca de uma mangueira, para simular a medição de gasto de água. O protótipo conta com o sensor, o Arduino e as conexões físicas estabelecidas entre os mesmos utilizando 3 *jumpers*.

5.1. Hardware do protótipo IoT

O Arduino é uma placa Uno Rev3 que, para facilitar os testes, ficou conectada diretamente a um laptop durante sua operação. Esta placa opera em 5 volts e possui 14 pinos digitais, dos quais 6 possuem *output* PWM (*Pulse-width modulation*, ou Modulação de Largura de Pulso) e outros 6 são analógicos. Três destes pinos foram utilizados para conectar a placa com o sensor do protótipo, como pode ser observado na Figura 5. O cabo vermelho conecta com a porta de 5 volts da Uno Rev3, para alimentar o circuito elétrico do sensor de fluxo de água, o qual é fechado pelo cabo preto, que está conectado a uma das portas *Ground* da placa. Por fim, o cabo amarelo fica conectado à porta digital 2 do Arduino, por onde o mesmo lê os pulsos do sensor.

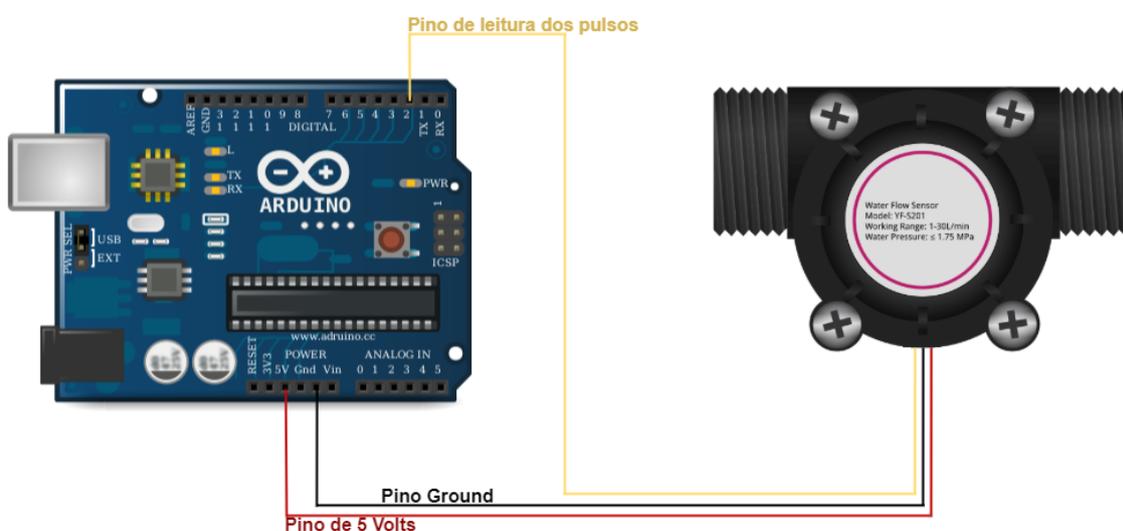


Figura 5. Visão do Arduino e sua implementação no dispositivo IoT da proposta
Fonte: Produção original

O sensor de fluxo de água utilizado foi o YF-S201. Este modelo possui um limite de funcionamento que varia de 1 a 30 litros de água por minuto, aguentando uma pressão de até 2 Mpa (Mega Pascal), que é equivalente a 2 N/mm^2 (Newton por milímetro quadrado). O aparelho opera com 5 Volts de eletricidade e transmite pulsos que o Arduino pode interpretar, de forma a obter as medições desejadas do sensor.

5.2. Software do protótipo IoT

Para permitir que o Arduino faça a leitura do (e interaja com o) sensor de fluxo YF-S201, um pequeno programa foi escrito. Esse programa utiliza algumas variáveis e um método de interrupção, junto com um método `update_system()` e os métodos padrões do Arduino, `setup()` e `loop()`. No método `update_system()`, a porta serial é lida para atualizar duas das variáveis do programa, a que controla o *threshold* de aviso de consumo e a que controla o intervalo de medição customizável, permitindo um terceiro intervalo junto dos predefinidos “minuto” e “hora” (este intervalo não afeta o ciclo de *sleep* e *wakeup* do dispositivo¹). No método `setup()`, o pino 2 é configurado como *input*, o *baud rate* é definido em 9600 e a interrupção é acoplada ao pino 2, para chamar o

¹O dispositivo está sempre acordado, devido à forma que as leituras são realizadas

método `rpm()` toda vez que o valor deste pino subir. Ainda no `setup()`, as interrupções são habilitadas, as variáveis de controle de tempo são inicializadas e, por fim, é impresso na porta serial os valores iniciais do *threshold* e do intervalo de medição customizado. Em seguida temos o método `loop()`, que é o mais complexo deste programa. O fluxograma da Figura 40 representa todo o código (descrito no apêndice A) do Arduino.

O Arduino envia diversas mensagens pela sua porta serial, codificadas em um formato específico. Essas mensagens podem vir escritas como “m2345”, por exemplo. O primeiro carácter da mensagem que o Arduino envia serve para identificar o tipo de mensagem sendo impressa, e é sempre uma letra (descritas abaixo). Em uma versão anterior do código, as mensagens sempre terminavam com a letra “z”, porém isto foi substituído pelo uso de quebra de linha (`\n`) através do comando `println()`. As mensagens também contém uma quantidade variada de números, às vezes com casas decimais separadas por um ponto, que podem representar medidas de volume ou tempo decorrido. As letras que definem os diferentes tipos de resposta são:

1. Letra `t`: Quanto tempo passou desde o último aviso de consumo, no formato HHMMSS. Devido ao formato, o número de horas pode eventualmente ficar maior que 100, alterando o formato.
2. Letra `m`: Uma leitura do volume de consumo do último intervalo customizado (por padrão, um minuto).
3. Letra `c`: Confirmação que o *threshold* de aviso foi configurado com sucesso, contendo o número lido.
4. Letra `i`: Confirmação que o intervalo de medição customizado foi configurado com sucesso, contendo o número lido.

Como mencionado anteriormente, o programa do Arduino conta com um método para configurar o *threshold* ou para configurar o intervalo de leitura customizável do dispositivo, o `update_system()`. Para invocar esse método, é necessário enviar uma mensagem através da porta serial com um formato específico, e tal mensagem serve para configurar um dos valores de cada vez, não sendo possível alterar ambos simultaneamente com apenas uma mensagem. Há dois formatos que são aceitos pelo Arduino, e eles seguem as seguintes expressões regulares:

1. `a[0-9]+`: Configura o *threshold* de aviso de consumo para o número definido.
2. `m[0-9]+`: Configura o intervalo de medição customizado para o número definido.

5.3. Servidor Web (Fog)

Como mencionado na proposta, juntamente com os colegas do Laboratório de Redes e Gerência (LRG) da UFSC, foi desenvolvido um website que oferece uma interface para controlar diversos dispositivos de uma casa inteligente através de uma *Central Internet of Things*, ou CIoT. Parte deste sistema interage com o protótipo desenvolvido para este trabalho, de forma a oferecer as funções detalhadas na proposta do mesmo. A Figura 6 mostra uma janela do website, na aba referente à medição de água da casa.

A interface do website conta com alguns elementos interativos para facilitar a navegação. À esquerda, há uma lista com todos os sistemas disponíveis na casa, e os dispositivos presentes em cada sistema são fruto de trabalhos do LRG. Na interface de

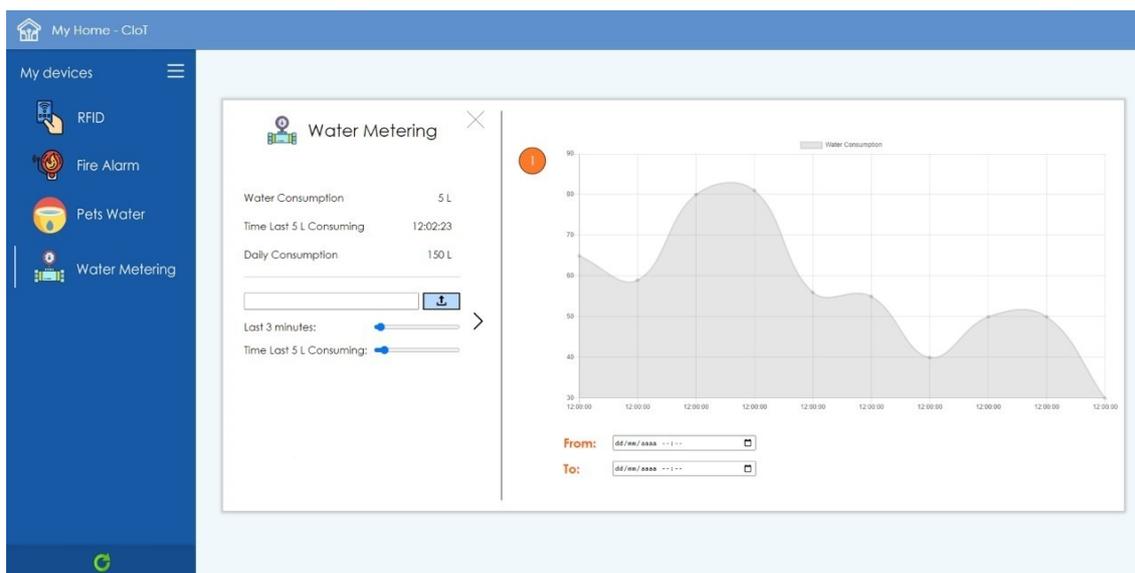


Figura 6. Exemplo do servidor web da proposta
Fonte: Produção original

medição de água, há primeiramente três informações distintas, sendo a primeira, *Water Consumption*, o valor referente ao volume medido no último intervalo de tempo (o qual pode ser customizado pelo usuário). A segunda informação mostra o *timestamp* do último aviso de *threshold*, e a terceira informação mostra o volume total do dia, o qual é calculado com base nas leituras feitas pelo Arduino. Em seguida, há um campo onde pode-se escrever e enviar um comando pela porta serial do Arduino, com dois *sliders* abaixo que permitem configurar os dois valores customizáveis de uma maneira mais intuitiva, sendo o *slider* superior utilizado para configurar o intervalo de leitura e o inferior para controlar o *threshold* (na Figura 6, ambos estão configurados para 3 minutos e 5 litros, respectivamente). Por fim, há um gráfico à direita da página que mostra um intervalo de tempo (que pode ser editado nas caixas de texto abaixo do mesmo) com todas as leituras de volume (que estão salvas na *database*) alinhadas a seus *timestamps*. As leituras no gráfico que forem representadas por um ponto diferente são os momentos onde o *threshold* foi alcançado.

O funcionamento de *backend* do website pode ser analisado pelo fluxograma do apêndice C. Quando o Arduino envia uma mensagem para o servidor, ele precisa enviar um pacote de dados com alguma informação. Esses pacotes são montados de acordo com a explicação no final da subseção 5.2, porém a CIoT requer uma informação adicional, um código hexadecimal no começo da mensagem, e este código não foi implementado no protótipo (logo, o protótipo atual não se comunica com a CIoT). De forma a identificar a qual sistema pertencem os dados das mensagens dos dispositivos, o servidor requer que os mesmos incluam um código hexadecimal no começo de suas mensagens e, caso o protótipo desenvolvido seja atualizado para enviar esse código, ele enviaria o valor hexadecimal 0×13 para identificar dados que pertencem ao sistema de monitoramento de água.

6. Testes e Resultados

O código do apêndice A foi compilado e rodou no Arduíno para testar a viabilidade do dispositivo base da proposta. Durante o período de uma hora, o sensor de fluxo de água ficou acoplado a uma mangueira de água, normalmente utilizada junto com uma pequena hidropistola para limpeza do vaso sanitário. A torneira da mangueira foi aberta algumas vezes de forma controlada, para permitir que água fluísse pelo sensor, de forma a produzir dados de consumo. Um exemplo das leituras que podem ser obtidas foi demonstrado na Figura 7, e a Tabela 4 possui as leituras de um teste de uma hora, onde é possível ver todas as leituras não nulas desse período (os períodos com muitas leituras nulas em sequência foram omitidos para melhorar a legibilidade da tabela, restando apenas saltos nos minutos onde duas leituras nulas se encontram).

Tabela 4. Teste do protótipo durante uma hora

Número	Data e hora da leitura	Mensagem	Medição	Tipo de mensagem (identificado pela letra)
01	07/11/2020 - 10:17:05.791	m1193	1193ml	Leitura após 1 minuto (configurado)
02	07/11/2020 - 10:18:05.702	m0	0ml	Leitura após 1 minuto (configurado)
03	07/11/2020 - 10:19:05.664	m320	320ml	Leitura após 1 minuto (configurado)
04	07/11/2020 - 10:20:05.559	m0	0ml	Leitura após 1 minuto (configurado)
05	07/11/2020 - 10:21:05.506	m2177	2177ml	Leitura após 1 minuto (configurado)
06	07/11/2020 - 10:22:05.419	m0	0ml	Leitura após 1 minuto (configurado)
07	07/11/2020 - 10:24:05.255	m0	0ml	Leitura após 1 minuto (configurado)
08	07/11/2020 - 10:24:35.751	a000830	8m e 30s	Aviso de <i>threshold</i> de 5 litros
09	07/11/2020 - 10:25:05.210	m2235	2235ml	Leitura após 1 minuto (configurado)
10	07/11/2020 - 10:26:05.102	m0	0ml	Leitura após 1 minuto (configurado)
11	07/11/2020 - 10:37:04.304	m0	0ml	Leitura após 1 minuto (configurado)
12	07/11/2020 - 10:38:04.218	m1959	1959ml	Leitura após 1 minuto (configurado)
13	07/11/2020 - 10:38:43.469	a001409	14m e 9s	Aviso de <i>threshold</i> de 5 litros
14	07/11/2020 - 10:39:04.150	m3251	3251ml	Leitura após 1 minuto (configurado)
15	07/11/2020 - 10:40:04.057	m73	73ml	Leitura após 1 minuto (configurado)
16	07/11/2020 - 10:41:04.020	m0	0ml	Leitura após 1 minuto (configurado)
17	07/11/2020 - 10:46:03.639	m0	0ml	Leitura após 1 minuto (configurado)
18	07/11/2020 - 10:47:03.547	m946	946ml	Leitura após 1 minuto (configurado)
19	07/11/2020 - 10:48:03.492	m1593	1593ml	Leitura após 1 minuto (configurado)
20	07/11/2020 - 10:48:51.970	a001009	10m e 9s	Aviso de <i>threshold</i> de 5 litros
21	07/11/2020 - 10:49:03.424	m1562	1562ml	Leitura após 1 minuto (configurado)
22	07/11/2020 - 10:50:03.319	m337	337ml	Leitura após 1 minuto (configurado)
23	07/11/2020 - 10:51:03.241	m0	0ml	Leitura após 1 minuto (configurado)
24	07/11/2020 - 10:54:03.052	m0	0ml	Leitura após 1 minuto (configurado)
25	07/11/2020 - 10:55:02.969	m428	428ml	Leitura após 1 minuto (configurado)
26	07/11/2020 - 10:55:47.113	a000656	6m e 56s	Aviso de <i>threshold</i> de 5 litros
27	07/11/2020 - 10:56:02.889	m5284	5284ml	Leitura após 1 minuto (configurado)
28	07/11/2020 - 10:57:02.807	m300	300ml	Leitura após 1 minuto (configurado)
29	07/11/2020 - 10:58:02.745	m0	0ml	Leitura após 1 minuto (configurado)
30	07/11/2020 - 11:16:01.417	m0	0ml	Leitura após 1 minuto (configurado)

O *threshold* padrão de 5 litros não foi alterado nos testes, bem como o intervalo customizado de medições, que forneceu as leituras por minuto (sua configuração padrão), as quais produziram os resultados da Tabela 4. A Figura 8 mostra uma visão geral do local onde o protótipo foi instalado.

Os testes feitos com a mangueira servem apenas para demonstrar que o dispositivo IoT proposto funciona. Os resultados obtidos e demonstrados na Tabela 4 são meramente ilustrativos, porém nos permitem chegar a algumas conclusões. De acordo com os resultados, pode-se concluir que o sensor de fluxo utilizado tem uma taxa de erro em torno

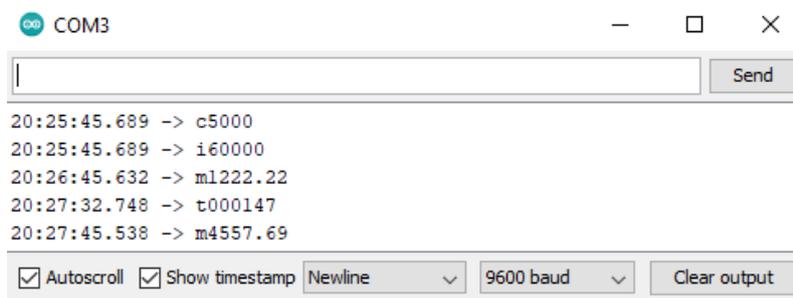


Figura 7. Exemplo da janela da porta serial do Arduino
Fonte: Produção original



Figura 8. Protótipo em ação, conectado à mangueira
Fonte: Produção original

dos 25%, o que torna este sensor impraticável em aplicações que necessitem de alta precisão. Alguns *outliers* podem ser observados, como um erro de 8.85% e outro de 56.2%. As medições manuais para validar a precisão das leituras foram feitas com ajuda de um pequeno medidor de cozinha de 250 mililitros. As várias leituras (dos resultados) foram discutidas na lista abaixo:

- A medida 01, de 1193 mililitros, resultou de um gasto controlado de 1 litro, apresentando uma margem de erro de 19,3%.
- A medida 03, de 320 mililitros, resultou de um gasto controlado de 250 mililitros, apresentando uma margem de erro de 28%.
- As medidas 05 e 09, de 2177 e 2235 mililitros respectivamente, resultaram de dois gastos controlados de 2 litros cada, apresentando erros de 8.85% e 11.75%.
- As medidas 12, 14 e 15, de 1959, 3251 e 73 mililitros respectivamente, não foram medidas manualmente. Considerando um erro de 10% para as duas primeiras medidas, pode-se estimar que o gasto foi de 1763 e 2926 mililitros respectivamente.

Para a terceira medida, considera-se um erro de 25%, logo resultando em uma estimativa de 55 mililitros. As taxas de erro para estas estimativas foram escolhidas com base nos itens anteriores desta lista.

- As medidas 18, 19, 21 e 22 são resultantes de um fluxo contínuo, e apenas a medida 21 de 1562 mililitros foi medida manualmente. Ao longo do minuto onde essa medida foi obtida, o medidor utilizado foi manualmente enchido e despejado 4 vezes, sem interromper o fluxo de água. Assim, pode-se concluir que há um erro de 56.2% para um fluxo “pequeno” e contínuo.
- A medida 27, de 5284 mililitros, foi resultado de um fluxo contínuo “grande”. O medidor foi utilizado novamente e, desta vez, foi enchido e despejado 16 vezes. Pode-se concluir, portanto, que houve um erro de 32% nesta medição.
- Ao final das leituras, exatamente uma hora após a inicialização do dispositivo (no mesmo momento do *timestamp* da leitura 30), pode-se calcular a leitura de uma hora com base nas leituras parciais. Logo, temos uma medida de 21666.9 mililitros (21.666 litros). A medida real, considerando um erro médio de 25%, deveria ser mais próxima dos 16250.175 mililitros.

Devido à necessidade de manter o dispositivo conectado a um computador, não foi possível testar outras funcionalidades da proposta. Há um pequeno desvio de milissegundos a cada leitura que resulta (a longo prazo) em variação nos horários exatos das leituras.

Com as medidas obtidas pelo Arduíno, o servidor pode utilizar os valores dos volumes e os *timestamps* das medições para viabilizar as várias funções mencionadas no capítulo 4. Esses *timestamps* e medições permitem que o servidor vá gerando gradualmente um gráfico do consumo de água, que pode ser utilizado para diversas funções, como permitir que o servidor identifique possíveis vazamentos a partir de leituras irregulares (por “leituras irregulares” entende-se que o servidor, ao analisar os dados de consumo anteriores, pode deduzir matematicamente que uma leitura x com um *timestamp* y não é comum para os padrões de uso de água da residência). Também é possível identificar quais medições são de qual mês e calcular uma aproximação do gasto total do mesmo, viabilizando os alertas de limite de gasto mensal. O *threshold* customizável facilita a definição de limites menores para permitir o controle de períodos mais variados.

A Figura 9 apresenta o gráfico que a aplicação web, da Figura 6, pode produzir com leituras como as deste teste. Os avisos de *threshold* estão denotados por uma linha vertical presente em algumas leituras.

7. Conclusões

O objetivo deste trabalho era demonstrar como uma solução para gerenciamento autônomo do consumo de água de uma residência poderia ser implementado, levando em consideração o paradigma *Internet of Things*. A pesquisa realizada permitiu a obtenção de um entendimento mais avançado sobre como as redes IoT podem ser utilizadas para se implementar a proposta eventualmente desenvolvida. O protótipo produzido foi capaz de obter medições simples necessárias para implementar todas as outras funcionalidades da proposta (as quais ficaram reservadas ao servidor da rede local).

Os resultados dos testes do protótipo ressaltaram o quão impreciso o sensor de fluxo YF-S201 é, indicando até 56% de erro em baixo fluxo e 32% em alto fluxo. Pela

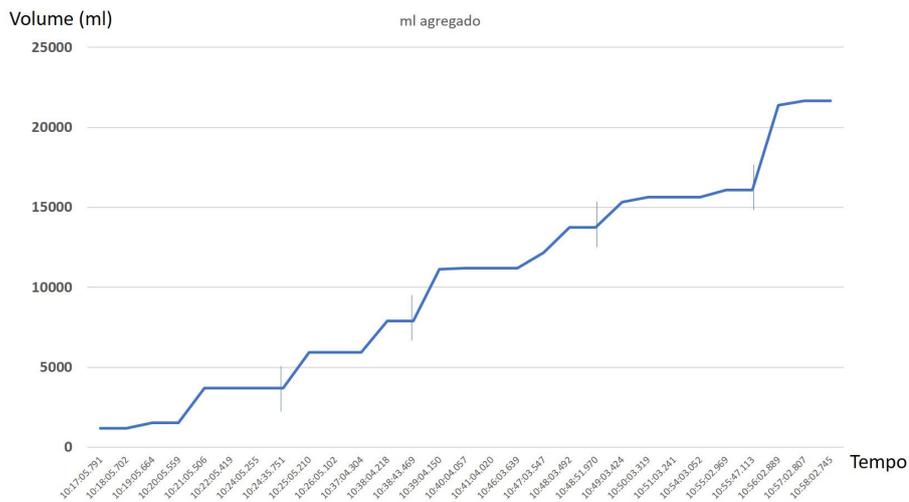


Figura 9. Gráfico das leituras acumuladas
Fonte: Produção original

própria natureza do sensor, as precisões obtidas pelos testes deixam muito a desejar quando se leva em conta o objetivo inicial da proposta, a qual pretendia alcançar um sistema com alta confiabilidade e autonomia. Mesmo que o protótipo tenha se demonstrado perfeitamente capaz de ser autônomo, verifica-se a necessidade de substituir o sensor de fluxo por outro modelo com maior precisão, já que a confiabilidade do sistema depende quase que inteiramente dessa precisão.

7.1. Trabalhos futuros

Observa-se que existem melhorias a serem feitas no protótipo, e para exemplificar mais detalhadamente essa afirmação, nota-se que o sistema pode:

- ser atualizado para medir uma quantidade mais variada de dados, como pressão de água, temperatura, “pureza” da água, dentre outros;
- ser ampliado para monitorar condomínios inteiros, ou talvez até cidades;
- ser multiplicado para monitorar diversas saídas de água de uma casa, capacitando-o a medir os gastos específicos de torneiras, chuveiros e/ou mangueiras individualmente;
- ser otimizado para não precisar ficar ligado 100% do tempo para ler o sensor;
- adicionar uma antena para aprimorar a comunicação do dispositivo com a CIoT;
- ser atualizado com um sensor mais robusto.

Referências

- Aggarwal, S., Chauhan, S., and Prakash, R. J. (2019). An Automated System to Monitor the Usage of Water in Apartments Using IOT and Artificial Neural Network. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pages 821–825. IEEE.
- Alrawais, A., Alhothaily, A., Hu, C., and Cheng, X. (2017). Fog Computing for the Internet of Things: Security and Privacy Issues. *IEEE Internet Computing*, 21(2):34–42.

- Bhilare, R. and Mali, S. (2016). IoT based smart home with real time E-metering using E-controller. *12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015*, pages 1–6.
- Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N., and Mahmoudi, C. (2018). Fog computing conceptual model. *NIST Special Publication*, (500-325):11.
- Jeurkar, V., Inamdar, M., Kale, P., Khandelwal, A., Daware, S., and Patel, R. (2020). IOT Based Water Management System. *2020 International Conference on Industry 4.0 Technology, I4Tech 2020*, pages 141–144.
- Ji, Y., Yong, L., Jingfeng, Y., Ke, K., Yuehua, H., and Wenguang, Q. (2015). Analysis of urban residential water consumption based on smart meters and fuzzy clustering. *Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se*, pages 1295–1301.
- Masia, O. A. and Erasmus, L. (2013). A study into the implementation of Water Conservation/Water Demand Management in Gauteng municipalities. *2013 Proceedings of PICMET 2013: Technology Management in the IT-Driven Services*, pages 2660–2673.
- Narayanan, L. K. and Sankaranarayanan, S. (2019). IoT Enabled Smart Water Distribution and Underground Pipe Health Monitoring Architecture for Smart Cities. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–7. IEEE.
- Rajurkar, C., Prabakaran, S. R., and Muthulakshmi, S. (2017). IoT based water management. *2017 International Conference On Nextgen Electronic Technologies: Silicon to Software, ICNETS2 2017*, pages 255–259.
- Saad, A., Benyamina, A. E. H., and Gamatie, A. (2020). Water Management in Agriculture: A Survey on Current Challenges and Technological Solutions. *IEEE Access*, pages 38082–38097.
- Sampaio, H. V., de Jesus, A. L. C., do Nascimento Boing, R., and Westphall, C. B. (2019). Autonomic IoT Battery Management with Fog Computing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11484 LNCS(Cc):89–103.
- Xia, C., Li, W., Chang, X., Delicato, F. C., Yang, T., and Zomaya, A. (2018). Edge-based energy management for smart homes. *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3*, pages 842–848.