

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS – INSTITUTO  
POLITÉCNICO DA PUC MINAS  
Curso de Engenharia Elétrica

Gabriel Leonardo Rodrigues da Silva

**DESENVOLVIMENTO DE UM SISTEMA IOT DE MONITORAMENTO DO  
CONSUMO DE ÁGUA RESIDENCIAL COM FOCO NA INFORMAÇÃO AO  
USUÁRIO**

Belo Horizonte  
2022

Gabriel Leonardo Rodrigues da Silva

**DESENVOLVIMENTO DE UM SISTEMA IOT DE MONITORAMENTO DO  
CONSUMO DE ÁGUA RESIDENCIAL COM FOCO NA INFORMAÇÃO AO  
USUÁRIO**

Trabalho de Conclusão de Curso apresentado à  
Pontifícia Universidade Católica de Minas Gerais  
como requisito para obtenção do título de  
Engenheiro Eletricista.

Orientador: Dr. Gustavo Luís Soares

Belo Horizonte

2022

Gabriel Leonardo Rodrigues da Silva

**DESENVOLVIMENTO DE UM SISTEMA IOT DE MONITORAMENTO DO  
CONSUMO DE ÁGUA RESIDENCIAL COM FOCO NA INFORMAÇÃO AO  
USUÁRIO**

Trabalho de Conclusão de Curso apresentado à  
Pontifícia Universidade Católica de Minas  
Gerais como requisito básico para obtenção do  
título de Engenheiro Eletricista.

Área de Concentração: Protótipo de sistema.

---

Prof. Dr. Gustavo Luís Soares – PUC Minas (Orientador)

---

\_\_\_\_\_ – PUC Minas (Banca Examinadora)

Belo Horizonte, \_\_ de dezembro de 2022.

*À minha família, namorada, amigos e colegas de curso, pelo apoio.*

## **AGRADECIMENTOS**

*À Deus, pelo fôlego de vida, sua presença e amor incondicionais.*

*Aos meus pais, Evanio Rodrigues Silva e Regina do Carmo Rodrigues da Silva, por todo amor, apoio e incentivo depositados desde o meu nascimento. Também agradeço em especial pela ajuda na coleta dos dados de consumo realizada durante uma semana.*

*À minha namorada Danielle Perez da Silva, pela paciência, compreensão e carinho ao longo dos últimos anos.*

*À toda a minha família pela minha formação humanística e apoio.*

*Ao professor e orientador Gustavo Luís Soares pelo excelente trabalho.*

*Ao engenheiro Helvécio Britto de Oliveira, por fornecer o hidrômetro ultrassônico, o osciloscópio, e dar dicas específicas durante o desenvolvimento do sistema.*

*Ao meu sogro Eduardo Alexandre Silva pela instalação do hidrômetro ultrassônico.*

*Ao corpo docente do curso de engenharia elétrica na Pontifícia Universidade Católica de Minas Gerais, pelos ensinamentos.*

## RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de monitoramento do consumo de água residencial, centrado em informar o usuário de maneiras diferentes acerca dos seus dados de consumo. O sistema desenvolvido é composto por um hidrômetro ultrassônico, um circuito eletrônico com um microcontrolador conectado à rede wi-fi e um sistema especialista simples. O acompanhamento dos dados de consumo pôde ser feito pelo smartphone, computador e pela assistente virtual, que é capaz de estimar a fatura da concessionária de água. O sistema de monitoramento foi aplicado em uma residência e dados foram coletados durante o período de uma semana. Os dados informados pelo sistema seguiram o consumo da residência, possuindo uma diferença de 16% em relação à medição da concessionária de água. A diferença encontrada se deve aos dois tipos de tecnologia dos hidrômetros, sendo um mecânico e outro ultrassônico, além da incapacidade de identificação da direção do fluxo de água por parte do circuito eletrônico construído.

Palavras-chave: Monitoramento do consumo de água. Internet das coisas. Automação residencial. Informação ao usuário.

## **ABSTRACT**

This work presents the development of a residential water consumption monitoring system, centered at informing the user about his consumption data in different manners. The system is composed of an ultrasonic hydrometer, an electronic circuit with a microcontroller connected to wi-fi and a simple specialist system. The monitoring can be done using a smartphone, computer and by the virtual assistant that is capable of estimating the price of the water provider bill. The system was applied in a residence and data was collected during the course of one week. The data provided by the system followed the residence consumption, having a difference of 16% compared to the water provider measurements. The difference was found due to the difference between the measurement technologies, one being mechanical and the other being ultrasonic, as well as the incapacity of the built circuit in identifying the direction of the water flow.

Keywords: Water Consumption monitoring. Internet of Things. Residential automation. User information.

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>10</b>
<b>1.1. Contextualização .....</b>	<b>10</b>
<b>1.2. Justificativa.....</b>	<b>11</b>
<b>1.3. Objetivos.....</b>	<b>12</b>
1.3.1. Geral.....	12
1.3.2. Específicos.....	12
<b>1.4. Metodologia .....</b>	<b>12</b>
<b>1.5. Delimitações.....</b>	<b>13</b>
<b>1.6. Organização.....</b>	<b>13</b>
<b>2. REVISÃO BIBLIOGRÁFICA .....</b>	<b>14</b>
<b>2.1. Internet das coisas .....</b>	<b>14</b>
<b>2.2. Automação residencial.....</b>	<b>15</b>
<b>2.3. Sistema especialista .....</b>	<b>16</b>
<b>2.4. Microcontrolador ESP8266 .....</b>	<b>18</b>
<b>2.5. Hidrômetro ultrassônico .....</b>	<b>19</b>
<b>2.6. API e JSON .....</b>	<b>20</b>
<b>2.7. Trabalhos relacionados .....</b>	<b>21</b>
<b>3. DESENVOLVIMENTO DO SISTEMA .....</b>	<b>23</b>
<b>3.1. Funcionamento.....</b>	<b>23</b>
<b>3.2. Circuito eletrônico .....</b>	<b>24</b>
3.2.1. Alimentação.....	24
3.2.2. RTC3232 .....	28
3.2.3. Comunicação com o hidrômetro ultrassônico .....	29
3.2.4. Código base em Arduino e compatibilização com o circuito .....	31
<b>3.3. Plataformas para comunicação com o ESP-01 .....</b>	<b>33</b>
3.3.1. Blynk IoT.....	33
3.3.2. ThingSpeak.....	35
<b>3.4. Sistema Especialista com VoiceFlow .....</b>	<b>37</b>
3.4.1. Primeira função - Informação do consumo.....	37
3.4.2. Segunda função - Estimativa de gastos.....	39
<b>4. APLICAÇÃO .....</b>	<b>41</b>
<b>4.1. Instalação do hidrômetro e locação do circuito .....</b>	<b>41</b>
<b>4.2. Coleta e análise dos dados .....</b>	<b>42</b>
4.2.1. Resultados do consumo informado pela Blynk IoT .....	42
4.2.2. Avaliação da skill baseada no sistema especialista .....	45



<b>5. CONCLUSÃO.....</b>	<b>47</b>
<b>REFERÊNCIAS.....</b>	<b>48</b>
<b>APÊNDICE A – CÓDIGO DO MICROCONTROLADOR ESP-01.....</b>	<b>51</b>

## 1. INTRODUÇÃO

### 1.1. Contextualização

A Internet das coisas é um conceito amplo que envolve a conexão digital de dispositivos físicos. A comunicação entre esses dispositivos fornece novas possibilidades de aplicação para eles, pois o dado coletado por qualquer dispositivo conectado à rede pode ficar disponível a todos, desta forma, o dado coletado torna-se informação útil de formas diferentes para dispositivos distintos. Na era digital, a internet das coisas tem sido aplicada em diversos ambientes, inclusive nas residências.

A automação residencial consiste, entre outras coisas, na utilização da tecnologia para a criação de casas inteligentes. Com a automação, tarefas antes feitas pelo homem são substituídas pela máquina. Coisas simples como acender uma lâmpada ou fechar uma janela antes de sair, são feitas automaticamente. Não se limitando a aplicações simples, a criação de sistemas personalizados usando a estrutura *IF This Then That* (IFTTT), permite sequências de ações únicas, realizadas em momentos específicos de acordo com o gosto do usuário. “O objetivo do IFTTT é automatizar tarefas que se prestam à automação, sem exigir nenhuma habilidade formal de programação”. (OVADIA, 2014). Além disso, o uso de assistentes virtuais como a *Amazon Alexa* e *Google assistant* tem promovido a automação residencial.

Um sistema especialista tenta emular a análise de um especialista humano sobre determinado assunto, ou seja, é um sistema especializado em uma determinada área de conhecimento, não é genérico. O sistema é capaz de aprender, e possui o desafio de reconhecer exceções, podendo utilizar respostas exatas ou aproximadas. No contexto da automação residencial, sistemas especialistas podem ser aplicados para auxiliar o morador na gestão dos seus recursos.

O controle do consumo de água em residências, depende, entre outros fatores, do monitoramento dos dados de consumo. Conhecer os gastos e reconhecer o impacto de cada um deles na totalidade da fatura, torna-se uma informação útil para tomar ações corretivas em hábitos específicos. Um dos fatores que colaboram para o descontrole nos gastos de água é o costume de olhar os dados de consumo apenas quando a fatura da concessionária chega na residência. O período do ciclo de faturamento costuma ser longo para um monitoramento efetivo.

Entre os 17 Objetivos de Desenvolvimento Sustentável estabelecidos pela cúpula das Nações Unidas, está o Objetivo 12.2: “Até 2030, alcançar a gestão sustentável e o uso eficiente dos recursos naturais” (ONU, 2015). Devido à emergente preocupação mundial em gerir corretamente os recursos naturais, torna-se cada vez mais importante o desenvolvimento de novos protótipos e ferramentas que permitam o monitoramento destes recursos.

Neste contexto, este trabalho aborda o desenvolvimento de um sistema de monitoramento *lot* do consumo de água residencial. Desta forma, pretende-se utilizar os conhecimentos técnicos para a construção de um protótipo aplicável, podendo ser utilizado como uma ferramenta para o controle do consumo de água em residências com acesso ao *wi-fi*.

## **1.2. Justificativa**

O monitoramento do consumo de água residencial pode ser realizado de diversas formas, seja pela consulta humana ao hidrômetro da concessionária de água ou até mesmo utilizando hidrômetros inteligentes, que enviam dados de consumo automaticamente para um servidor exclusivo.

Entretanto, o monitoramento manual exige várias leituras do consumo, e a coleta de dados nem sempre pode ser realizada, visto que no ambiente residencial, nem sempre o usuário está presente.

Em um contexto de internet das coisas e automação residencial, sistemas personalizáveis surgem como uma necessidade, permitindo a adaptação do monitoramento para cada indivíduo. Os hidrômetros inteligentes que se conectam à um servidor próprio, podem ser limitados em termos de informações fornecidas, onde nem sempre a apresentação da informação é realizada da melhor forma.

Neste trabalho será desenvolvido um protótipo de sistema para o monitoramento do consumo de água residencial, com características específicas focadas na informação ao usuário, sendo assim um sistema totalmente personalizável, podendo ser adaptado de maneiras diferentes para cada aplicação. Assim, a solução apresentada é alterável de acordo com as preferências do desenvolvedor.

### 1.3. Objetivos

#### 1.3.1. Geral

Desenvolver um sistema de monitoramento do consumo de água residencial utilizando um microcontrolador conectado à rede *wi-fi*, informando o usuário de diferentes formas acerca dos seus dados de consumo.

#### 1.3.2. Específicos

- Desenvolver um circuito capaz de disponibilizar os dados do consumo de água da residência via *wi-fi* através do microcontrolador ESP8266;
- Desenvolver um código (skill) para a assistente virtual Alexa, baseado em um sistema especialista;
- Possibilitar o monitoramento do consumo via smartphone;
- Analisar a confiabilidade do sistema de monitoramento.

### 1.4. Metodologia

O desenvolvimento do sistema *iot* é dividido em quatro partes. A primeira consiste no planejamento do circuito eletrônico, acrescido da elaboração do código inicial do microcontrolador. A segunda trata-se da integração do microcontrolador com as plataformas de exibição dos dados. A terceira é o desenvolvimento do código do sistema especialista. A última etapa consiste na construção definitiva do circuito eletrônico, aplicação do sistema e coleta de dados.

Na etapa de planejamento do circuito eletrônico, a função de cada pino do microcontrolador será definida através da elaboração do código inicial, que também permitirá determinar os componentes complementares necessários para o funcionamento do protótipo.

Na segunda etapa serão estudadas as possíveis plataformas de comunicação com o microcontrolador, visando escolher opções que permitam o envio de dados para a assistente virtual *Alexa* (onde o sistema especialista será aplicado) e para o smartphone. O código do ESP8266 será completado de acordo com os requisitos de comunicação dessas plataformas.

O código do sistema especialista será desenvolvido na própria plataforma escolhida para comunicação entre o microcontrolador e a assistente virtual, focando em executar tarefas simples, sendo a principal delas a informação do consumo de água da residência na semana em questão.

Na etapa final, o circuito será soldado em uma placa de fenolite perfurada, e alocado em um recipiente de proteção, tendo em vista a necessidade de posicionamento do circuito próximo ao hidrômetro. Serão coletados os dados de consumo de água informados pelo sistema de monitoramento e também a medição da concessionária de água, para análise posterior.

### **1.5. Delimitações**

Este trabalho delimitou-se no desenvolvimento de um sistema *iot* para o monitoramento do consumo de água somente em residências que possuem acesso ao *wi-fi*. Tendo em vista o foco do trabalho na informação ao usuário, a medição do consumo será realizada por um hidrômetro ultrassônico, evitando problemas de imprecisão que poderiam ser ocasionados por medidores de vazão populares no mercado, que são conectados diretamente ao microcontrolador.

### **1.6. Organização**

No Capítulo 2 a revisão bibliográfica é vista. O desenvolvimento do sistema *iot* é exposto no Capítulo 3. No Capítulo 4 a aplicação do sistema e os resultados são apresentados. No Capítulo 5 as conclusões obtidas são descritas.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. Internet das coisas

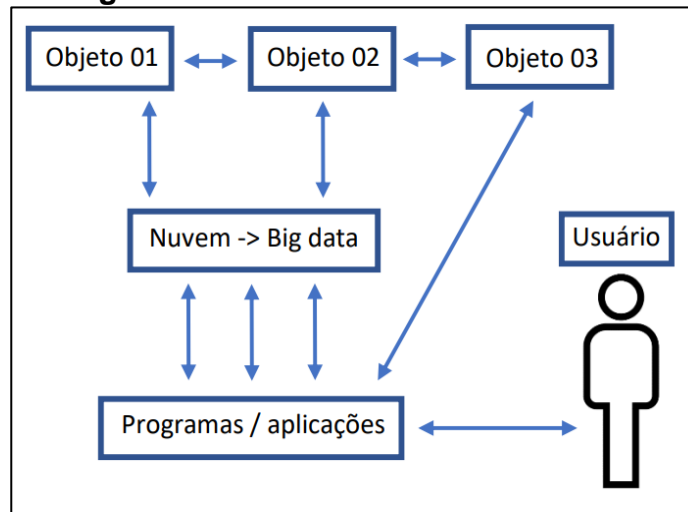
Segundo Magrani (2018), existem várias definições diferentes para o conceito de *IoT*, entretanto, todas chegam a um ponto em comum, na interação e processamento de dados realizados por dispositivos, em um contexto de hiperconectividade. Um dispositivo *IoT* está conectado em uma rede, se comunicando com indivíduos, consumindo ou contribuindo com o fluxo de dados, sendo que estes dados podem ser tratados e transformados em informação útil.

Em algumas aplicações de sistemas *IoT*, surgirá a necessidade de armazenar e transferir grandes volumes de dados, que precisam estar disponíveis para os dispositivos conectados. Estes volumes são chamados de *Big Data*. “*Big data* é um termo em evolução que descreve qualquer quantidade volumosa de dados estruturados, semi-estruturados ou não estruturados que têm o potencial de ser explorados para obter informações.” (MAGRANI, 2018, p.20).

Com diferentes dispositivos acessando simultaneamente um volume de dados, torna-se necessário a aplicação do *Armazenamento em nuvem*. “Com o armazenamento em Nuvem, os dados são armazenados em vários servidores de terceiros, em vez de nos servidores dedicados usados no armazenamento de dados em rede tradicional.” (SIBUYA, 2019, p.32).

A estrutura de um sistema *IoT* pode ser representada através de um diagrama de blocos, conforme Figura 1, onde os objetos se comunicam, buscam e fornecem dados na nuvem e alguns trocam informações diretamente com as aplicações, que por sua vez, podem promover a interação com o usuário ou cumprir um propósito específico, como analisar os dados coletados e chegar em alguma conclusão útil para a atuação de dispositivos.

**Figura 1 – Diagrama de blocos da estrutura de um sistema iot**



Fonte: Elaborado pelo autor

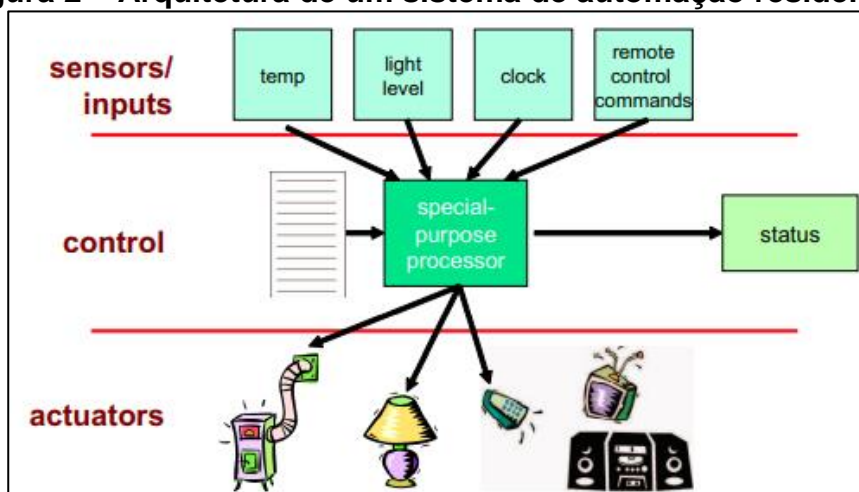
## 2.2. Automação residencial

A automação residencial é um ramo da automação que tem evoluído nos últimos anos. Apesar do desenvolvimento tecnológico de dispositivos eletrônicos e inteligência artificial, ainda existem desafios a serem vencidos para tornar a automação residencial acessível e plena. Segundo Silva Filho (2018), apesar de existirem vários sistemas de automação, muitos cumprem um propósito específico e não se integram aos demais dispositivos da residência, tornando-se sistemas isolados. Visando a comunicação entre os dispositivos, outros modelos são adquiridos e situações de redundância surgem, duplicando os equipamentos do local.

A automação residencial pode usar sensores, dados da internet e informações do usuário como suprimento para a criação de estruturas lógicas que controlam atuadores presentes na residência, abrindo ou fechando janelas, acendendo luzes, ligando equipamentos e mais, fornecendo uma boa experiência ao morador.

Castro e Psota (2009) representaram a estrutura de um sistema de automação residencial em 3 camadas, apresentadas na Figura 2.

**Figura 2 – Arquitetura de um sistema de automação residencial**



Fonte: (CASTRO e PSOTA, 2009, p. 5)

A primeira camada é a de entradas e sensores, ela é a responsável pela coleta de dados. A segunda camada se trata da unidade controladora, que ao obter as informações fornecidas pela primeira camada, pode emitir um sinal para a terceira camada, dependendo do seu código de controle. Na última camada ficam os dispositivos atuadores, como relês, motores de portão eletrônico e fechaduras eletrônicas. Após o recebimento do sinal emitido pela unidade controladora, estes dispositivos atuam no mundo real, ligando equipamentos, fechando portões, cumprindo o seu propósito de atuação (CASTRO e PSOTA. 2009).

Um sistema de automação residencial pode utilizar a inteligência artificial para tornar a experiência do usuário mais personalizada, através da técnica *Machine Learning*. Arthur Samuel (1959) utilizou a técnica de *Machine Learning* pela primeira vez, para a criação de um jogo de damas. Quando criado, o programa sempre perdia para o seu criador, mas depois de várias partidas, o programa aprendeu as melhores e piores posições do tabuleiro, através do acúmulo de experiências, e venceu. Conforme Aaron *et al.* (2017), foi atribuída ao Arthur Samuel a seguinte definição de *Machine Learning*: “Campo de estudo que dá aos computadores a capacidade de aprenderem sem serem explicitamente programados”.

### **2.3. Sistema especialista**

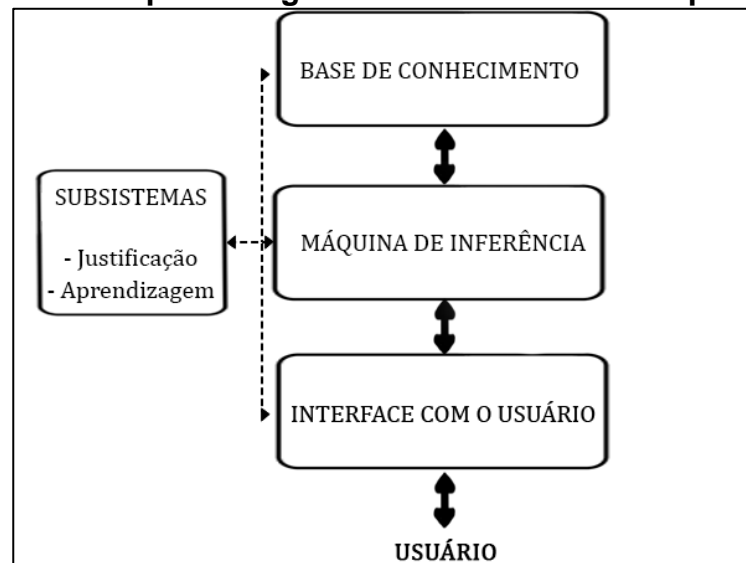
Um sistema especialista busca emular um especialista humano, tomando decisões a respeito de um assunto específico usando as informações que possui. Segundo Heinzle (1995) eles são sistemas computacionais projetados e



desenvolvidos para solucionarem problemas dentro de uma específica área de atuação, utilizando uma profunda e organizada base de conhecimentos. O sistema especialista também possui um método de raciocínio ou inferência, permitindo que ele responda questionamentos, e justifique suas conclusões.

A Figura 3 apresenta uma arquitetura básica e genérica de um sistema especialista, composta por três blocos principais: A base de conhecimento, a máquina de inferência e a interface com o usuário.

**Figura 3 – Arquitetura genérica de um sistema especialista**



Fonte: Elaborado pelo autor com dados extraídos de CÉSAR DA SILVA (1994, p. 26)

Na interface com o usuário a linguagem humana é traduzida para a linguagem de máquina, e vice-versa. A máquina de inferência é o bloco responsável por todo o raciocínio e inferência realizadas pelo sistema (CÉSAR DA SILVA, 1994). O método de inferência pode ocorrer de maneiras diferentes. Segundo Durkin (1994) esse método pode ser o *Forward-chaining* iniciando na coleta de informações e depois partindo para a hipótese, ou *Backward-chaining*, iniciando na hipótese e depois partindo para a coleta de informações. Por último, conforme Klotz (2002), a base de conhecimento é onde são armazenados os fatos e regras da área de especialidade do sistema. Este bloco é um dos mais importantes, pois guarda o conhecimento do especialista em si.

O sistema especialista tem ainda, dois subsistemas. O primeiro é de justificação, responsável por explicar como determinada resposta foi encontrada. O outro é de aprendizagem, responsável por aperfeiçoar a base de conhecimento. Este

procedimento pode ser feito manualmente por um especialista humano ou de forma autônoma (CÉSAR DA SILVA, 1994).

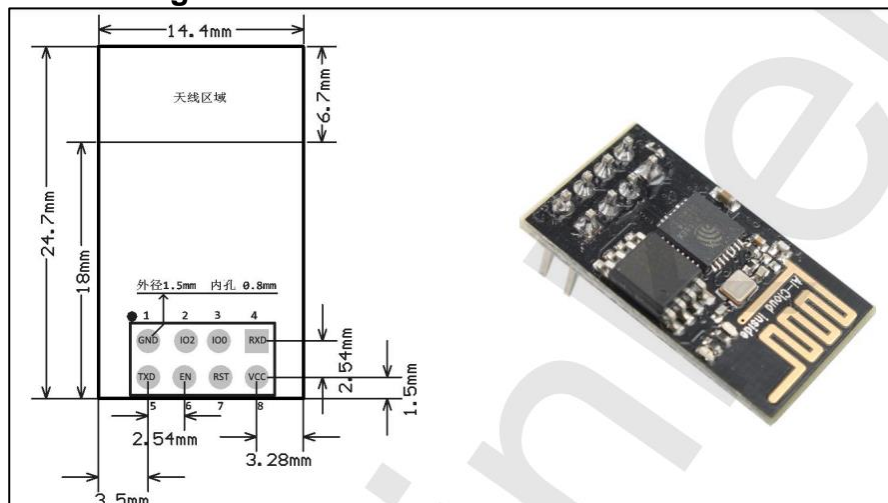
#### 2.4. Microcontrolador ESP8266

O microcontrolador ESP8266 é fabricado pela Espressif systems. “Um microcontrolador é um circuito integrado capaz de executar programas” Kolban (2016).

Hoje existem vários microcontroladores no mercado, com preços e poder de processamento diferentes. Um diferencial do ESP8266 em comparação com outros chips da mesma faixa de preço, é a sua compatibilidade com a comunicação *wi-fi*. Segundo Kolban (2016) esse microcontrolador pode ser utilizado como módulo *wi-fi* para complementar outros microcontroladores que são focados em aspectos técnicos diferentes, mas além disto, o ESP8266 também pode ser usado de forma independente.

O ESP8266 foi projetado visando um consumo de energia eficiente e ocupação de pouco espaço físico. A empresa Ai-thinker desenvolveu um módulo para o ESP8266, com o objetivo de facilitar o seu uso. Este módulo concentra o Soc (System on Chip) em uma pequena placa de circuito, junto com os componentes principais para o seu funcionamento. Esse módulo é difundido comercialmente e denominado ESP-01. A Figura 4 apresenta o ESP-01 e suas dimensões.

**Figura 4 – Dimensões do módulo ESP-01**



Fonte: (AI-THINKER, 2017)

Segundo a Ai-Thinker (2017) o ESP-01 pode ser amplamente utilizado em uma variedade de redes, para automação residencial, industrial, controle sem fio, variados produtos eletrônicos, sensores de localização sem fio, e outros.

Uma vantagem do ESP8266 é a compatibilidade com a linguagem e IDE do Arduino. Essa linguagem é comumente ensinada em cursos técnicos e universidades, no desenvolvimento de tarefas didáticas que envolvem criação de protótipos.

Existem outros microcontroladores parecidos com o ESP8266 fabricados pela Espressif. Esses possuem o mesmo processador, e quantidades diferentes de *GPIO's* (General Purpose Input/Output). As *GPIO's* são portas do microcontrolador disponíveis para o uso de diferentes formas. Elas são usadas para a conexão de dispositivos ou sensores, podendo ser configuradas como pino de entrada ou saída.

A tabela 1 mostra algumas especificações técnicas do ESP-01.

**Quadro 1 – Algumas especificações técnicas do ESP-01**

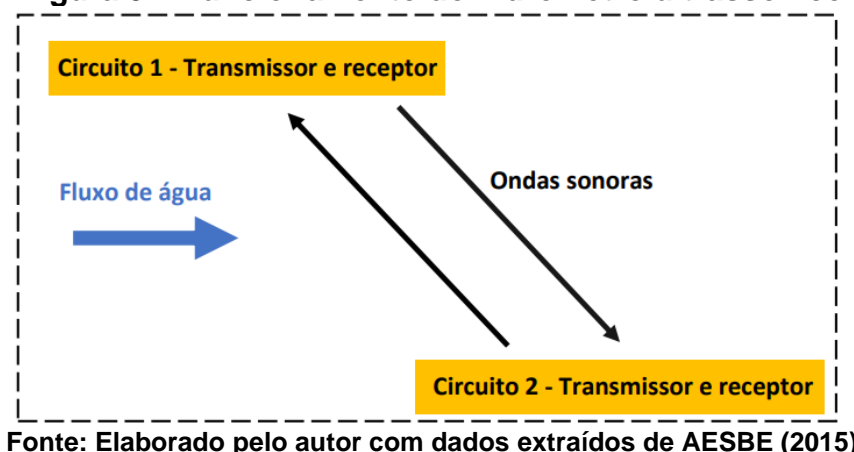
Alimentação	Voltage 3.0V ~ 3.6V Current >300Ma
Temperatura de operação	-20 °C ~ 85 °C
Faixa de frequência	2412 ~ 2484MHz
Peso	0.65g

Fonte: Elaborado pelo autor com dados extraídos de AI-THINKER (2017)

## 2.5. Hidrômetro ultrassônico

Existem diversos tipos de hidrômetros, que efetuam a medição do volume de água através de diferentes princípios físicos. O hidrômetro ultrassônico efetua a medição utilizando dois circuitos eletrônicos que emitem e recebem ondas de som ultrassônicas. O tempo em que a onda sonora leva para sair do emissor e chegar ao receptor é medido pelo hidrômetro, e com base nesse tempo é possível saber se o meio físico percorrido pela onda é água ou ar. A Figura 5 ilustra o funcionamento descrito acima (OYAMADA TAMAKI, 2003).

**Figura 5 – Funcionamento do hidrômetro ultrassônico**



Através do tempo de propagação da onda, é possível calcular, além do consumo de água, a vazão, velocidade e a direção do fluxo de água. No processo de medição as ondas sonoras vão tanto do Circuito 1 para o 2, quanto do Circuito 2 para o 1 (OYAMADA TAMAKI, 2003).

Os hidrômetros ultrassônicos possuem a vantagem de atingir elevadas classes de exatidão, pois o fluxo de ar não é contabilizado na medição. Além disso, possuem uma vida útil maior quando comparada com outros tipos de hidrômetros, que tem partes móveis mecânicas em contato com o fluido medido (PEREIRA E SILVA, 2022).

## 2.6. API e JSON

Durante o desenvolvimento de programas nas mais diversas áreas, surge a necessidade de transferir dados entre serviços, e para isso existem as *API's* (Interface de programação de Aplicação). Segundo Rakhmawati *et al.* (2019), através do exemplo da Indonésia, um dos países com maior número de usuários de internet do mundo, observa-se o uso de *API's* públicas como um dos fatores facilitadores que contribuem para o crescimento de dados e tecnologia. “A *API* pública ou aberta é uma interface de programação de aplicativos que é publicamente disponível a todos, como desenvolvedores com acesso programático à interface.” (Rakhmawati *et al.*, 2019).

Uma *API* pode ser consumida através de uma *JSON* (JavaScript Object Notation). Segundo Izquierdo e Cabot (2016), no ano de pesquisa, o número de *API's* estava crescendo, abrindo portas para vários novos aplicativos que utilizam o serviço. Dentre essas *API's*, a maioria usa *JSON*. Segundo Tiwary, Stroulia e Srivastava (2021), *JSON* é um tipo de documento estruturado, utilizado para transferências de

dados, sendo um formato baseado em texto que contém chaves específicas, nas quais o dado que está sendo trocado está. Essas chaves são repetidas várias vezes tanto na solicitação quanto na resposta dos documentos.

## **2.7. Trabalhos relacionados**

Gonçalves Junior (2019) desenvolveu um medidor eletrônico de consumo de água. O dispositivo é composto pelo microcontrolador ESP8266 e fornece os dados de consumo remotamente e autonomamente via *wi-fi*. O medidor foi construído para a aplicação no Polo de Inovação Campos dos Goytacazes, com foco no baixo custo. Ao final do trabalho o custo de R\$98,75 por medidor foi alcançado, estando dentro do objetivo proposto de R\$100,00 a unidade.

Machado Junior (2020) estudou o potencial da colheita de energia através da micro hidrogenação. Ele desenvolveu uma plataforma de testes composta pelo ESP8266, sensor de fluxo YF-S201, uma bomba d'água, uma microturbina e um circuito alimentador. Posteriormente o ESP8266 foi substituído pelo ESP32. Após medir as potências envolvidas no processo, ele concluiu que a autossustentabilidade no universo *iot* pode se tornar realidade no médio prazo.

Utilizando o microcomputador Raspberry Pi e o sistema operacional Android, Silva Filho (2018) desenvolveu um sistema de automação residencial microcontrolado para residências. O sistema inclui a aquisição de dados via *wi-fi*. A interação entre o usuário e o sistema é feita remotamente pelo celular, e o usuário é informado através de um monitor conectado ao Raspberry Pi. Silva Filho concluiu que é possível associar diferentes sensores em um único sistema, reduzindo a quantidade de centrais de controle.

Caíque Vendemiatti (2020) desenvolveu um sistema de monitoramento do consumo de água em tempo real. Através do ESP8266 e de um sensor de fluxo por *efeito Hall*, o sistema envia dados de utilização de água que podem ser analisados para a identificação de padrões de consumo ou vazamentos. Os dados coletados são enviados para um servidor web. O sistema foi aplicado em diferentes residências e foi constatada a sua eficácia na detecção de vazamentos.

No trabalho de Scroccaro Costa (2020) um dispositivo de telemetria para hidrômetros foi desenvolvido juntamente de estudos de geração e consumo energético. O dispositivo utiliza a colheita de energia solar, aliada a um

supercapacitor. Visando o baixo consumo energético, o envio de dados ocorre por meio de *Bluetooth Low Energy* (BLE), e a medição da vazão é feita por um sensor óptico que identifica as rotações da aleta metálica de um hidrômetro mecânico.

Silva Júnior (2017) propôs uma arquitetura de rede de sensores de fluxo com comunicação sem fio. No trabalho um protótipo foi desenvolvido, utilizando o microcontrolador Arduino Mega e um módulo de comunicação *Zigbee*. O protótipo mostrou-se eficaz nas simulações, sendo capaz de identificar perdas no sistema de abastecimento de água em até 10 minutos. Também foi desenvolvida uma aplicação para o monitoramento dos dados, capaz de mostrar o histórico de consumo em diferentes gráficos.

Apesar dos trabalhos acima estarem relacionados à internet das coisas, automação residencial e monitoramento do consumo de água, nenhum deles tem como foco de estudo a informação ao usuário aliada ao uso de uma assistente virtual, como proposto neste trabalho.

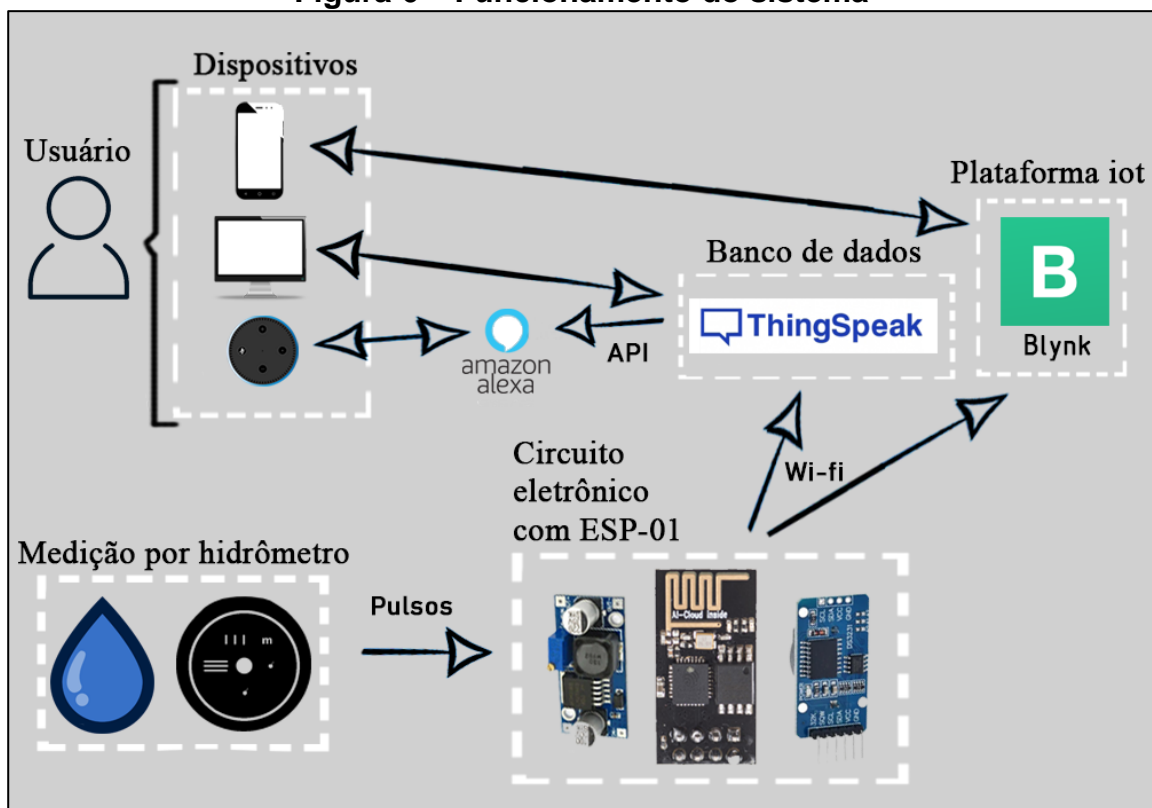
### 3. DESENVOLVIMENTO DO SISTEMA

#### 3.1. Funcionamento

Nesta seção o funcionamento do sistema será abordado, mostrando seus componentes, para facilitar a compreensão das etapas de desenvolvimento apresentadas nos tópicos seguintes.

A Figura 6 ilustra o funcionamento do sistema.

**Figura 6 – Funcionamento do sistema**



Fonte: Elaborado pelo autor

A medição do consumo de água é realizada pelo hidrômetro ultrassônico, que envia pulsos para o ESP-01 de acordo com o fluxo de água. A partir do circuito eletrônico, o ESP-01 envia os dados medidos à plataforma Blynk IoT e ao banco de dados gratuito ThingSpeak, via *wi-fi*. A plataforma Blynk torna possível a leitura do consumo diário pelo smartphone. Através de uma *skill* feita para a assistente virtual Alexa, o consumo semanal pode ser informado pelo dispositivo da Amazon, que lê os dados diretamente do banco de dados. A *skill* desenvolvida também fornece outras informações. Os dados de consumo também podem ser vistos diretamente no site da plataforma ThingSpeak, em um computador ou smartphone.

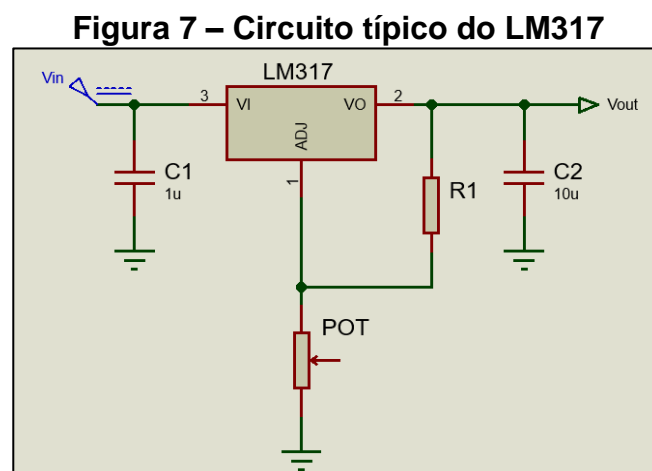
### 3.2. Circuito eletrônico

Dentro do sistema de monitoramento proposto, o circuito eletrônico é uma peça fundamental para que a comunicação entre o ESP-01 e o hidrômetro ultrassônico ocorra. Seu desenvolvimento foi realizado levando em consideração os requisitos de alimentação estabelecidos no *datasheet* do microcontrolador, e os componentes complementares que são necessários para a sua correta operação conforme o código embarcado.

#### 3.2.1. Alimentação

Para determinar a alimentação do circuito, foram analisadas duas possibilidades, o uso de bateria ou de retificador (fonte DC conectada à rede elétrica). Tendo em vista que a tensão de alimentação do ESP-01 pode variar entre 3,0 ~ 3,6V, foi necessário usar um componente para adequar a tensão de entrada a esses valores. Para isto, existem dois tipos de componentes populares no mercado, o regulador linear e o Conversor DC-DC (também conhecido como “Buck”).

O regulador linear funciona, simplificada, como um resistor variável, adequando o seu valor ôhmico de acordo com a necessidade de tensão da carga. Existem reguladores lineares que permitem o ajuste da tensão de saída, como o modelo LM317. Outros são fabricados especificamente para uma tensão de saída fixa, como a série 78xx. O modelo 7805, por exemplo, é capaz de fornecer uma tensão de saída de 5V, desde que se tenha uma tensão de entrada entre 7V ~ 35V. Cogitou-se utilizar o regulador linear LM317 para auxiliar na alimentação do circuito, conforme circuito mostrado na Figura 7.



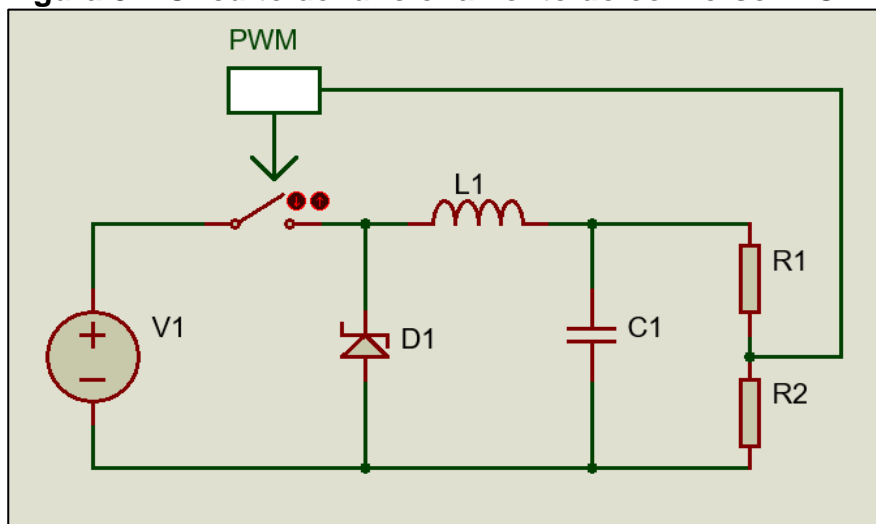
Fonte: Elaborado pelo autor



Optou-se por não usar reguladores lineares para a alimentação, devido à preocupação com a eficiência do circuito.

O conversor DC-DC foi escolhido para adequar a tensão de entrada no circuito eletrônico proposto. Ele funciona pelo princípio de modulação da largura de pulso (PWM). O circuito apresentado na Figura 8 demonstra o funcionamento do conversor.

**Figura 8 – Circuito de funcionamento do conversor DC-DC**



Fonte: Elaborado pelo autor

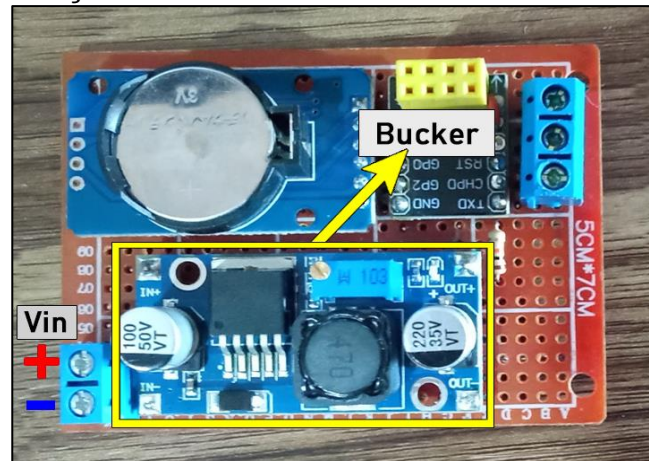
A fonte V1 representa a tensão de entrada, que precisa estar cerca de 2V a 3V acima da tensão de saída, variando este valor de acordo com o fabricante e modelo do conversor. Este valor é chamado de *Dropout*. O interruptor posicionado em série com a fonte representa um transistor ou mosfet que irá abrir e fechar em altas frequências. No caso do LM2596, escolhido para o protótipo, a frequência de oscilação típica é 150kHz. Por meio deste chaveamento a tensão de saída é ajustada. A tensão após o chaveamento terá a forma de onda quadrada, e por este motivo existe um filtro passa baixa em seguida, formado pelo indutor L1 e capacitor C1. Com o filtro, a tensão de saída terá uma forma de onda mais próxima da tensão contínua ideal, mas ainda assim apresentará pequenas variações, que podem ser prejudiciais para circuitos sensíveis. No caso do circuito proposto, isto não será um problema.

O diodo D1 é do tipo schottky, caracterizado por ter um tempo de recuperação menor do que os diodos de silício tradicionais, como o IN4007. É importante que este tempo seja pequeno para que a forma de onda da tensão de entrada do filtro passa baixa não fique distorcida, se distanciando de uma onda quadrada ideal. O diodo schottky é responsável por dissipar a energia armazenada no indutor toda vez que a

chave for aberta, impedindo a queima do mosfet ou transistor de chaveamento. Por último, o circuito possui ainda um feedback do sinal de saída, que servirá para que a largura dos pulsos possa ser ajustada de acordo com a variação da carga.

A Figura 9 mostra a posição do conversor DC-DC no circuito desenvolvido.

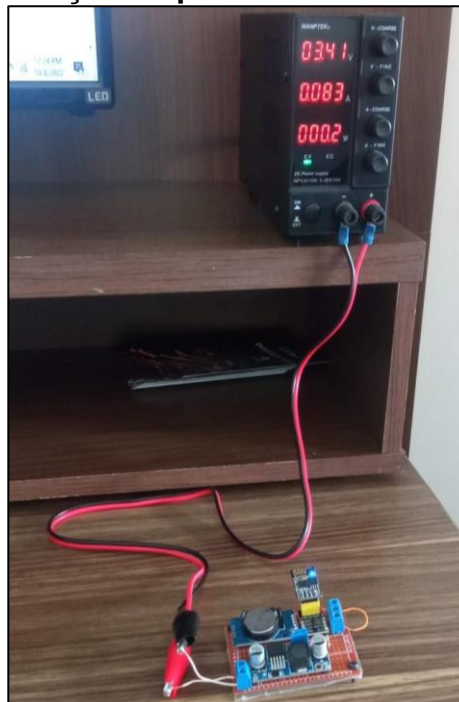
**Figura 9 – Posição do conversor DC-DC no circuito desenvolvido**



Fonte: Elaborado pelo autor

Após todo o circuito ter sido montado, uma fonte de bancada foi utilizada para medir a potência elétrica consumida, visando dimensionar a bateria necessária para que o protótipo funcione por no mínimo 7 dias consecutivos. A Figura 10 mostra a medição.

**Figura 10 – Medição de potência com fonte de bancada.**



Fonte: Elaborado pelo autor

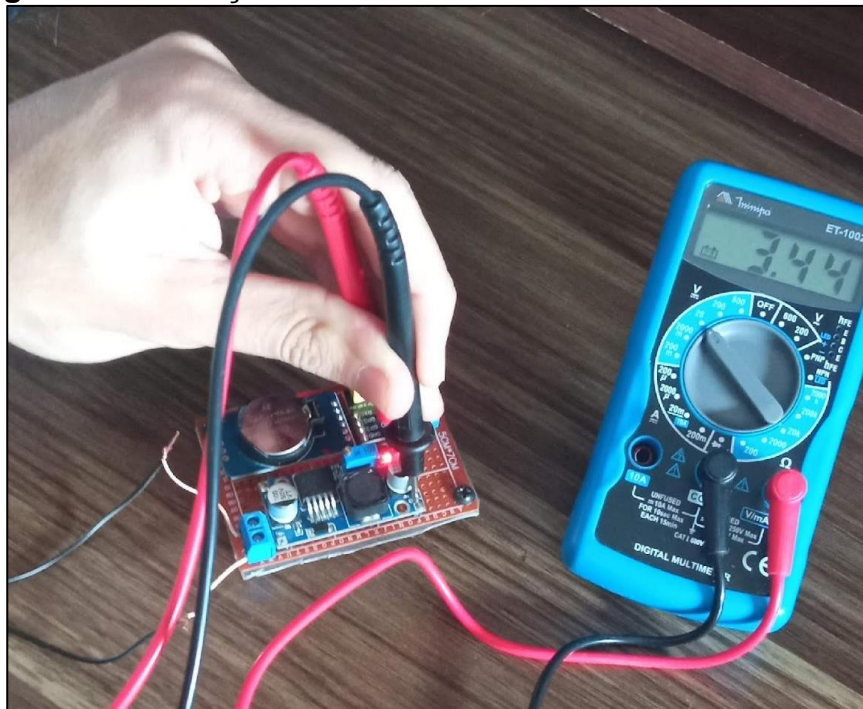
Nota-se que com uma tensão de 3,41V a corrente fornecida foi de 0,083A. Logo, a capacidade da bateria para alimentar o circuito por 7 dias é dada por:

$$\text{Capacidade} = 0,083 \times 24 \text{ horas} \times 7 \text{ dias} = 13944\text{mAh} \quad (1)$$

Ao analisar as baterias disponíveis no mercado, percebe-se que a capacidade requisitada pelo circuito é alta, e para atendê-la seriam necessárias várias baterias do encapsulamento convencional, de células cilíndricas. Como o baixo consumo de energia não é o foco de estudo deste trabalho, optou-se pelo uso de um retificador comum, uma fonte de tensão de 7,5VDC para alimentar o circuito.

Dois conectores tipo KRE foram soldados na placa para permitir a conexão e desconexão da fonte ao conversor buck. A fonte de 7,5VDC foi conectada à rede elétrica e a tensão de saída do conversor foi medida com um multímetro, conforme mostrado na Figura 11.

**Figura 11 – Medição da tensão de saída do conversor buck**



Fonte: Elaborado pelo autor

Nota-se que a tensão de saída igual a 3,44V está dentro da faixa requisitada para o correto funcionamento do ESP01.

### 3.2.2. RTC3232

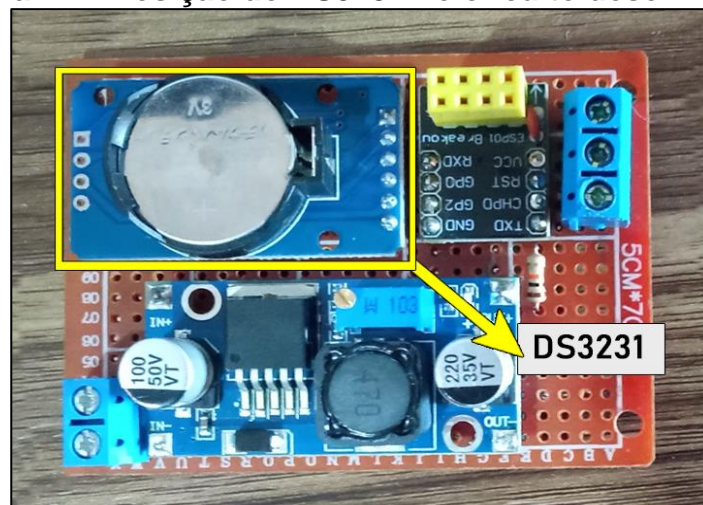
Durante o desenvolvimento do primeiro código para o ESP-01, percebeu-se que o microcontrolador necessitaria da informação de tempo, visto que de 15 em 15 segundos o consumo de água precisaria ser enviado para o banco de dados, e no fim do dia o consumo precisaria ser zerado.

Para a contagem do tempo, o microcontrolador possui uma função interna denominada millis. A função millis retorna o tempo contado desde quando o ESP8266 foi iniciado, em milissegundos. A função é útil para diversas aplicações, e pode ser utilizada para a contagem de dias. Entretanto, a contagem de tempo pode apresentar imprecisões quando comparada com circuitos Real Time Clock (RTC), pois não é feita por um componente físico. “Este não é um relógio real (como em um relógio de CPU), mas é um período de temporizador lógico [...]” (KOLBAN, 2017, p.231).

Para o circuito desenvolvido, optou-se pelo uso do componente eletrônico DS3231, um circuito RTC que fornece informações de tempo como ano, mês, dia, hora, minuto e segundo. Em situações onde a energia elétrica é cessada, o DS3231 utiliza uma bateria tipo moeda para continuar contando o tempo, não perdendo os dados inseridos.

A informação de tempo é passada para o ESP01 através do protocolo de comunicação I2C. A Figura 12 mostra a posição do DS3231 no circuito.

**Figura 12 – Posição do DS3231 no circuito desenvolvido**



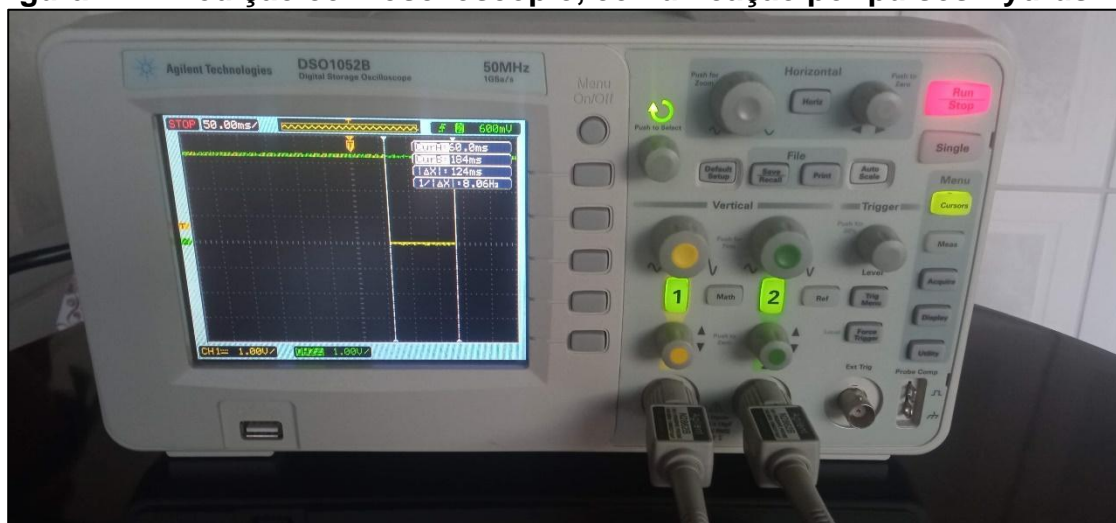
Fonte: Elaborado pelo autor





de instalação. A Figura 14 mostra a leitura do osciloscópio.

**Figura 14 – Medição com osciloscópio, comunicação por pulsos Hydrus 1.3**



Fonte: Elaborado pelo autor

A função STOP do osciloscópio foi utilizada para analisar o pulso do transistor no momento da passagem do fluxo. Foi identificado que o pulso ocorre a cada 10 litros de água que passam pelo hidrômetro. Usando os cursores, a largura do pulso foi medida, seu valor é de 124ms.

O ideal para o protótipo em desenvolvimento seria um pulso para cada um litro, pois assim, pequenas variações do consumo já seriam mostradas ao usuário. Pelo fato do ambiente ser residencial, em diversos momentos uma pequena quantidade de água é consumida. De acordo com o datasheet do hidrômetro, esta correspondência de um pulso para cada um litro pode ser configurada, entretanto, ao conversar com o fabricante, descobriu-se que seria necessário utilizar um software específico. Tendo em vista as dificuldades de acesso ao software, foi decidido continuar com a configuração de um pulso a cada 10 litros.

Na medição realizada com o osciloscópio, percebeu-se ainda que o sinal do canal dois permaneceu em nível lógico alto durante a passagem do fluxo de água, indo para o nível lógico baixo somente quando um fluxo de direção contrária à inicial foi forçado a percorrer o hidrômetro. Logo, como esperado, o sinal denominado pelo fabricante como “pulso dois” é um recurso para identificação de fluxos reversos.

Um gerador de sinais foi configurado para gerar pulsos com as mesmas características vistas na Figura 14, eliminando a necessidade de montar um circuito de recirculação de água para testar a comunicação entre o hidrômetro e o microcontrolador em bancada. O modelo do gerador de pulsos utilizado é genérico, e

funciona pelo princípio de PWM, através de dois mosfets. Para configurá-lo, foi necessário entrar com a frequência e o duty cycle desejados. Pensando em um pulso com duração de 125ms em estado Low, o duty cycle foi calculado.

$$\text{duty cycle} = 100 \times \frac{\text{TON}}{\text{TON} + \text{TOFF}} = \frac{100 \times 875\text{ms}}{875\text{ms} + 125\text{ms}} = 87,5\% \quad (1)$$

Então, o gerador de pulsos foi configurado para um duty cycle de 87,5% e uma frequência de 1Hz, resultando assim, em 1 pulso por segundo, com largura de 125ms no nível lógico baixo. Isto foi utilizado durante todo o período de desenvolvimento do circuito eletrônico.

#### 3.2.4. Código base em Arduino e compatibilização com o circuito

O funcionamento do código desenvolvido se resume em ler os dados do hidrômetro ultrassônico, enviar o consumo diário para a Blynk *IoT* a cada um segundo e enviar o consumo semanal para o ThingSpeak a cada 15 segundos.

Inicialmente, foi necessário mapear as conexões entre o microcontrolador e os componentes complementares necessários para o correto funcionamento do código. O ESP-01 possui quatro pinos que podem ser usados como GPIO's. Eles precisam ser referenciados no código embarcado com números específicos. As GPIO's foram utilizadas conforme o Quadro 2.

**Quadro 2 – Utilização das GPIO's do ESP-01**

GPIO	Número de referência na IDE do Arduino	Aplicação
0	0	Pino SCL para comunicação I2C com o DS3231
TXD	1	Recepção dos pulsos do sensor ultrassônico
2	2	Pino SDA para comunicação I2C com o DS3231
RXD	3	Não possui

**Fonte: Elaborado pelo autor**

Tendo a utilização de cada pino mapeada, o código pôde ser desenvolvido. Na parte inicial, as bibliotecas foram declaradas, as variáveis criadas e os pinos foram definidos. Quatro bibliotecas foram utilizadas, e são apresentadas no quadro 3.

**Quadro 3 – Bibliotecas utilizadas**

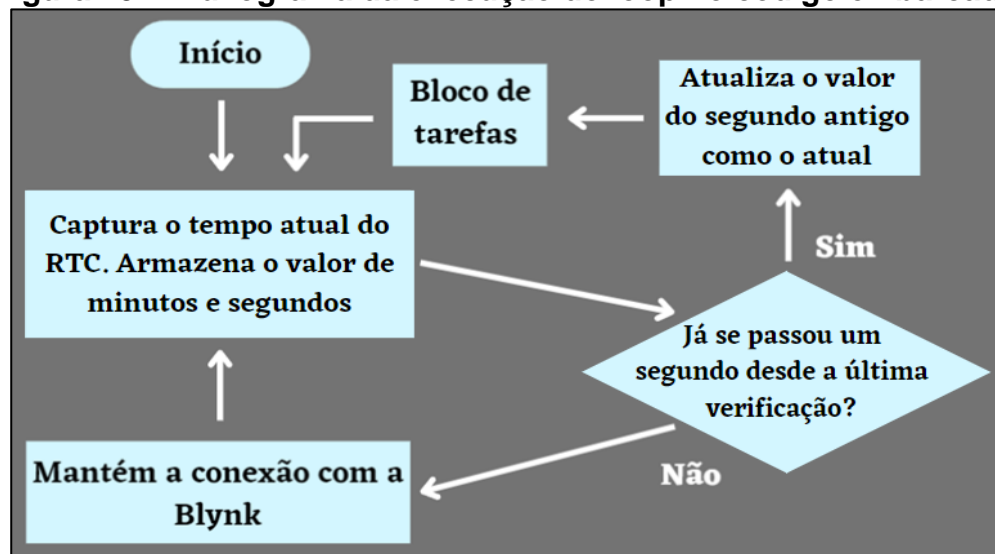
Biblioteca	Propósito de utilização
“ESP8266WiFi”	Facilitar a conexão entre o microcontrolador e a rede wi-fi
“BlynkSimpleESP8266”	Permitir a conexão entre o microcontrolador e a plataforma Blynk iot
“Wire”	Facilitar a comunicação por protocolos I2C
“RtcDS3231”	Facilitar a coleta de informação de tempo do RTC DS3231

Fonte: Elaborado pelo autor

Na declaração das variáveis, o SSID e a senha da rede *wi-fi* da residência foram inseridos. No setup do código, a interrupção foi habilitada para o pino TXD. Ela faz com que o microcontrolador pare qualquer tarefa que esteja executando, para atender a demanda do pulso enviado. A interrupção foi configurada para ocorrer na borda de descida do pulso, e cada vez que for acionada, guiará a execução do código para uma função que incrementa a variável “pulsos” com 0,1. Desta forma, os pulsos enviados pelo hidrômetro nunca serão ignorados pelo microcontrolador.

Ainda no setup, o ano, mês, dia, hora, minuto e segundos foram definidos para o RTC DS3231. Esse procedimento só precisou ser realizado uma vez, sendo assim, depois de inserir o microcontrolador no circuito, as linhas de código referentes a essa parte foram comentadas e o código foi carregado novamente.

No loop, o microcontrolador segue uma sequência lógica específica, que consiste apenas da estrutura condicional “IF”. Um fluxograma apresenta a sequência lógica desenvolvida e é mostrado na Figura 15.

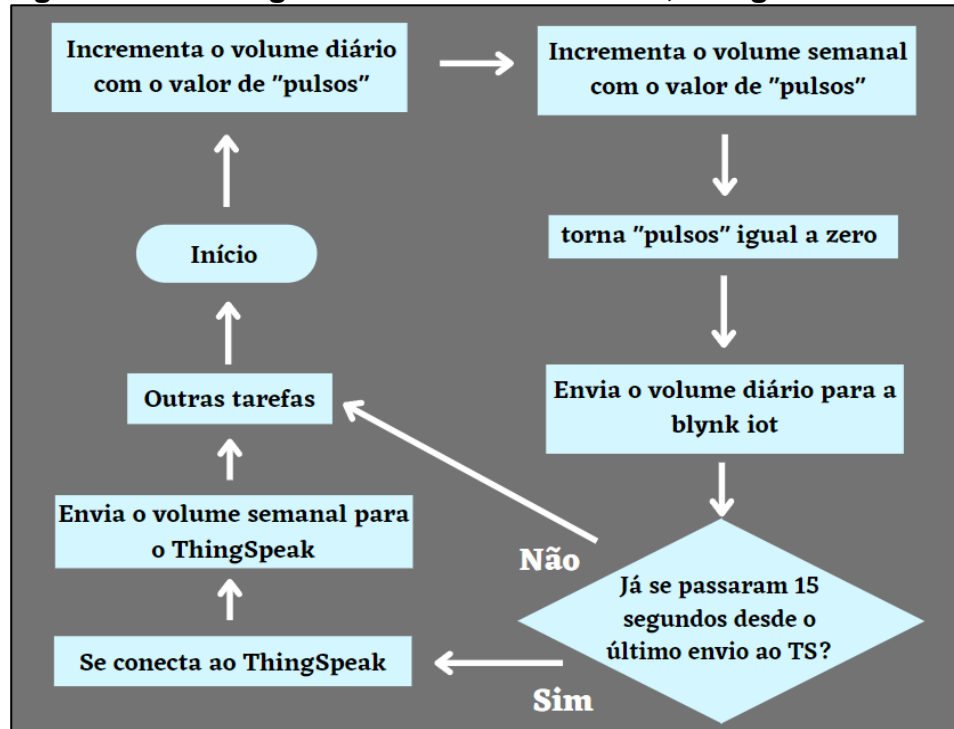
**Figura 15 – Fluxograma da execução do loop no código embarcado**

Fonte: Elaborado pelo autor



A cada loop o microcontrolador compara o valor do segundo atual com o valor do segundo anterior, e caso sejam diferentes, quer dizer que se passou um segundo desde a última atualização. A cada *loop* uma função chamada “*Blynk.run*” é chamada, para manter a conexão com a plataforma. O bloco de tarefas que é executado a cada segundo é apresentado na Figura 16.

**Figura 16 – Fluxograma do bloco de tarefas, código embarcado**



Fonte: Elaborado pelo autor

As tarefas executadas no bloco “outras tarefas”, são secundárias. Para ver o código completo, verifique o apêndice A.

### 3.3. Plataformas para comunicação com o ESP-01

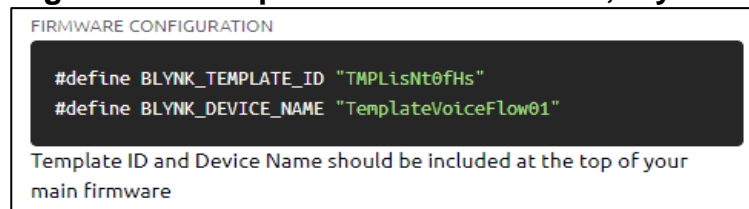
#### 3.3.1. Blynk IoT

A Blynk é uma plataforma de internet das coisas que oferece ferramentas para a integração entre hardware e software. A plataforma tem uma versão disponível para uso gratuito, com algumas limitações, que não atrapalham o desenvolvimento do sistema proposto. Ela foi escolhida para o projeto devido a facilidade de transferência dos dados entre o microcontrolador e o smartphone. Para fazer essa movimentação, é necessário criar um template e associar um pino virtual a ele.

A criação do template foi realizada no computador, pelo site da Blynk *iot*. Para

criá-lo, basta definir um nome, o hardware que se conectará a ele e o tipo de conexão. Com o template criado, a plataforma gera um ID que deve ser inserido na programação do microcontrolador, juntamente com o nome do dispositivo. A plataforma apresenta as linhas de código para inserção na IDE, conforme Figura 17.

**Figura 17 – Template ID e Device name, Blynk iot**

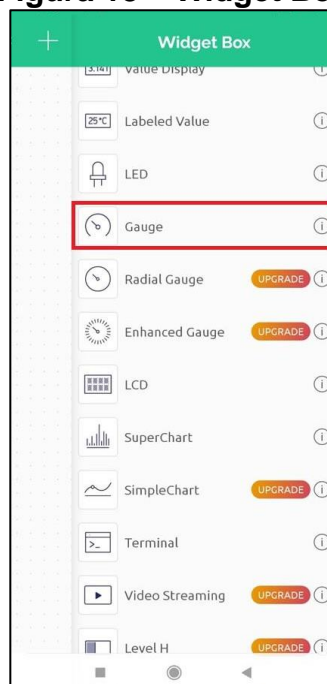


Fonte: (BLYNK IoT, 2022)

Após criar o template, o dispositivo a ser conectado precisa ser cadastrado no site da plataforma. Durante o cadastro, será gerado mais um código, chamado Auth Token, que é associado ao dispositivo, no caso, o ESP-01. O Auth Token também precisa estar presente no código do microcontrolador.

Por último, usando um celular é possível definir a forma de visualização dos dados. Ao entrar no template e selecionar as configurações, a plataforma apresenta uma widget Box com várias opções de visualização e ferramentas de interação com o usuário, como botões, joystick, display e outros. O item “gauge” foi selecionado na widget Box, mostrado na Figura 18.

**Figura 18 – Widget Box**



Fonte: (BLYNK IoT, 2022)

O gauge permite a visualização dos dados com base em um valor máximo configurável. Ele foi selecionado para o projeto, pois o usuário poderá estabelecer um consumo de água diário máximo, desta forma, ao entrar no aplicativo ele já irá saber se está dentro do consumo de referência ou não.

Para que o gauge receba o valor do consumo diário, é preciso associá-lo a um pino virtual. O pino virtual é como uma variável da plataforma, sendo ele responsável por armazenar o dado durante a comunicação entre ESP-01 e Blynk. Para o sistema em questão, o pino virtual “V0” foi escolhido, e configurado para receber dados do tipo *double*, ou seja, dados compridos, com vários dígitos. Os dados enviados terão somente uma casa decimal, entretanto, na plataforma só é possível escolher entre o tipo “*int*” e “*double*”.

Após definir o pino virtual no aplicativo, o envio de dados a partir do microcontrolador foi feito utilizando a função “Blynk.virtualWrite”, através da biblioteca “BlynkSimpleESP8266”.

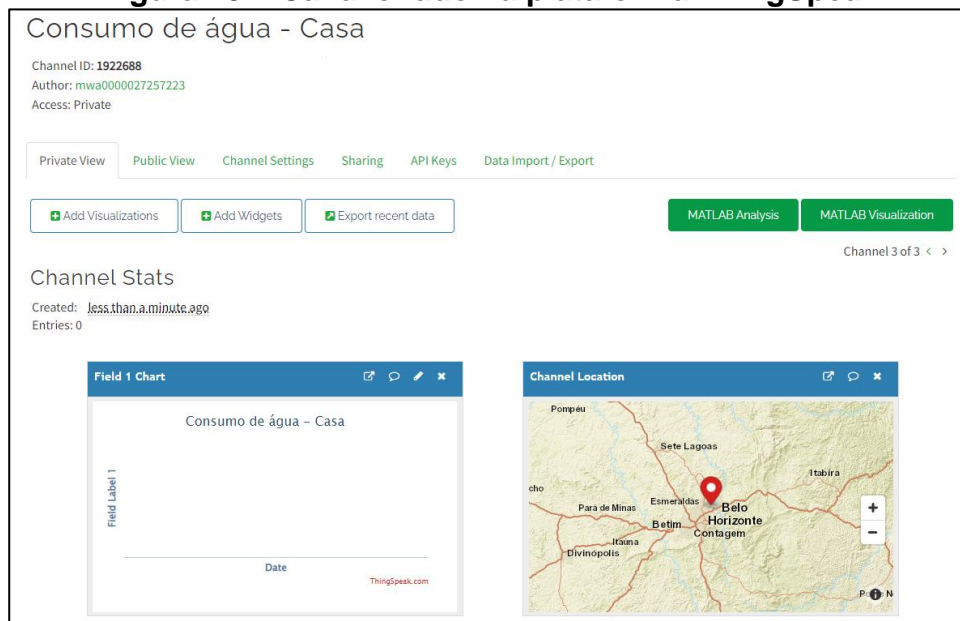
### 3.3.2. ThingSpeak

O ThingSpeak é uma plataforma para projetos de internet das coisas, focada em recursos de armazenamento e gerenciamento de dados na nuvem. A plataforma possui versões comerciais e uma versão gratuita com algumas limitações, que não são um problema para o projeto desta monografia. Dentro do sistema de monitoramento do consumo de água residencial, o ThingSpeak é responsável pelo banco de dados, que armazenará o consumo de água semanal, recebendo dados a cada 15 segundos, conforme mostrado na seção 3.2.4.

A plataforma funciona utilizando o conceito de canais e campos. Um mesmo canal pode conter vários campos, mostrando diferentes dados em cada um deles. O campo é o painel de visualização de dados propriamente dito.

Primeiramente, o canal “Consumo de água – Casa” foi criado, contendo somente um campo, que é mostrado na Figura 19.

**Figura 19 – Canal criado na plataforma ThingSpeak**



Fonte: (THINGSPEAK, 2022)

A comunicação entre o microcontrolador e o ThingSpeak é feita através de uma *API* que é gerada pela plataforma para cada canal criado. Para enviar os dados a partir do ESP-01, uma função chamada “*enviats*” foi adicionada no código do microcontrolador. A função se baseia no envio de *string*'s para a plataforma, compondo uma *Json*. Na composição da *Json*, o destinatário é identificado, especificando para qual canal a informação está indo e para qual campo deste canal, usando a *Write API Key* gerada pelo ThingSpeak.

Com os dados enviados, um gráfico é gerado automaticamente no ThingSpeak. É possível ainda, escolher preferências para o gráfico, como valores mínimos e máximos para cada eixo, permitindo um acompanhamento detalhado dos dados recebidos. A Figura 20 mostra o gráfico gerado para a plataforma para os dados de consumo de água semanal. Ao passar o cursor do mouse em cima dos pontos, o dia e horário de recebimento são mostrados.

**Figura 20 – Consumo de água semanal no ThingSpeak**



Fonte: (THINGSPEAK, 2022)

### 3.4. Sistema Especialista com VoiceFlow

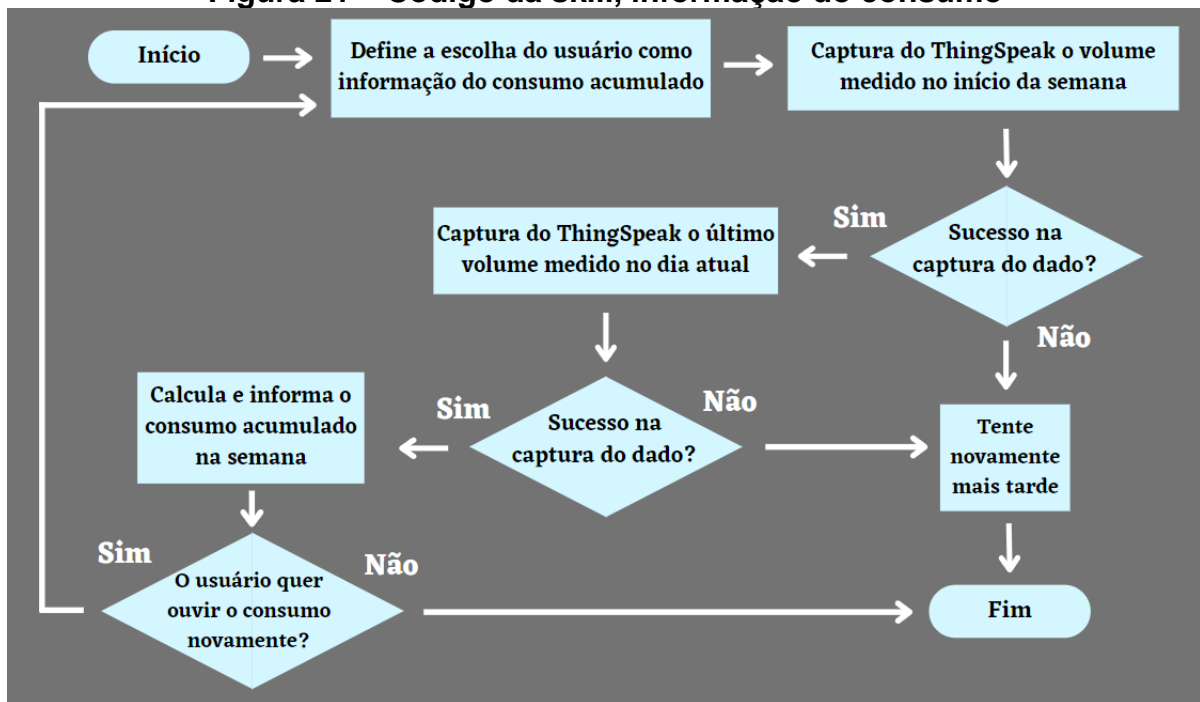
O sistema especialista desenvolvido funciona na assistente virtual Alexa, a partir de uma *skill*. As *skill's* são aplicativos com funções personalizadas, que podem ser desenvolvidos por qualquer pessoa com conta de desenvolvedor na Amazon. A criação de uma *skill* pode ser realizada diretamente na Console de desenvolvedor, em várias linguagens de programação, como Python e Node.js. Entretanto, visando facilitar o processo, a plataforma VoiceFlow foi escolhida para a criação da *skill*.

Com o VoiceFlow é possível criar *skill's* utilizando programação em blocos. A plataforma possui versões pagas e uma versão gratuita, com algumas limitações, que não são um problema para o sistema proposto nesta monografia.

#### 3.4.1. Primeira função - Informação do consumo

A principal e mais básica função da *skill* é informar o consumo de água acumulado desde o início da semana, permitindo que o usuário faça o monitoramento sem entrar no site do ThingSpeak. O processo de informação do consumo é representado no fluxograma da Figura 21.

**Figura 21 – Código da skill, informação do consumo**

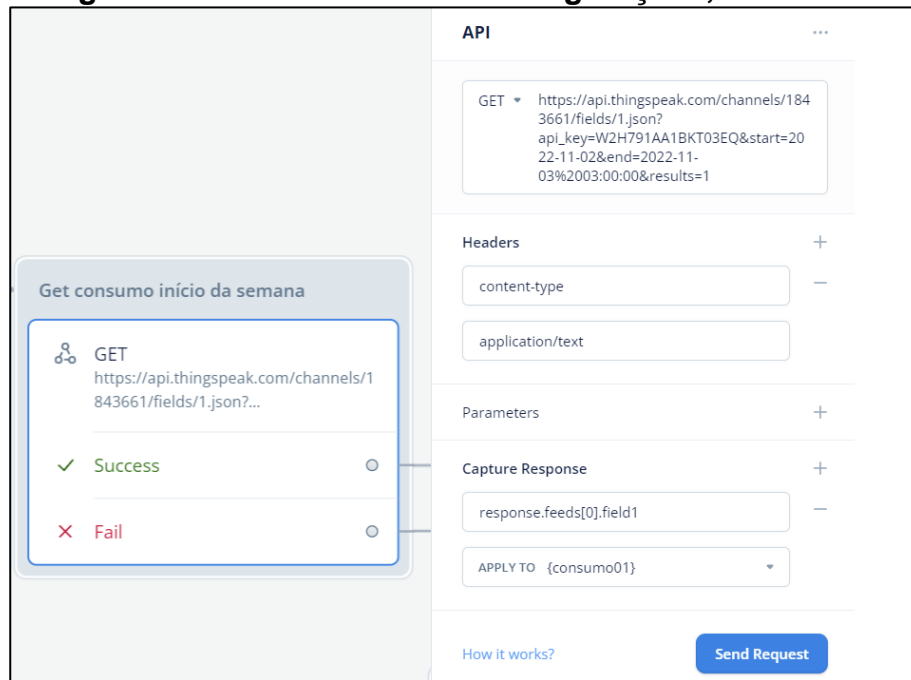


Fonte: Elaborado pelo autor

Após o usuário escolher que quer escutar o consumo de água acumulado, é atribuído valor a uma variável que é usada posteriormente na *skill*. Em seguida, o volume medido no início da semana é armazenado em uma variável, que será subtraída do último volume medido no dia em questão, para se obter o consumo acumulado total referente à semana. Finalmente, o consumo é informado ao usuário, e caso ele não queira repetir o processo, a Alexa encerrará a *skill*.

A captura de dados do ThingSpeak é realizada através de uma *API* gerada pela plataforma, de forma muito parecida com o que ocorre no envio de dados pelo ESP-01. A principal diferença é que no envio de dados do microcontrolador para a plataforma, é utilizado a chave *API* de escrita, enquanto que na captura de dados no Voiceflow, é utilizada a chave *API* de leitura. O VoiceFlow fornece um bloco específico para o consumo de *API*'s, mostrado na Figura 22.

**Figura 22 – Bloco de API's e configurações, VoiceFlow**



Fonte: (VOICEFLOW, 2022)

#### 3.4.2. Segunda função - Estimativa de gastos

A função de estimativa dos gastos serve para informar o usuário quanto ele irá pagar pelo consumo de água acumulado durante a semana. Para isto, a *skill* necessita aplicar o mesmo método de cálculo da concessionária de água, que é feito com base nas faixas de consumo e nas tarifas de água e esgoto. A fatura do consumo de água é calculada por:

$$Fatura = tf_a + tf_e + \sum_{i=1}^6 tai \times Ci + \sum_{i=1}^6 tei \times Ci \quad (2)$$

Sendo:  $tf_a$  é a tarifa fixa de água;

$tf_e$  é a tarifa fixa de esgoto;

$ta_i$  é a tarifa de água da faixa “i”;

$C_i$  é o consumo de água da faixa “i”;

$te_i$  é a tarifa de esgoto da faixa “i”.

A Figura 23 mostra as tarifas da COPASA para a categoria de consumo residencial.

**Figura 23 – Tarifas residenciais para água e esgoto, COPASA**

<b>Categorias</b>	<b>Faixas</b>	<b>Água</b>	<b>Esgoto</b>	<b>Unidade</b>
<b>Residencial</b>	Fixa	17,61	13,03	R\$/mês
	0 a 5 m <sup>3</sup>	1,82	1,35	R\$/m <sup>3</sup>
	> 5 a 10 m <sup>3</sup>	3,886	2,876	R\$/m <sup>3</sup>
	> 10 a 15 m <sup>3</sup>	6,023	4,457	R\$/m <sup>3</sup>
	> 15 a 20 m <sup>3</sup>	8,222	6,084	R\$/m <sup>3</sup>
	> 20 a 40 m <sup>3</sup>	10,458	7,739	R\$/m <sup>3</sup>
	> 40 m <sup>3</sup>	12,759	9,441	R\$/m <sup>3</sup>

Fonte: (ARSAE-MG, 2021)

No VoiceFlow foram criadas 14 variáveis para armazenar as tarifas. Após definir as tarifas, a skill passa por um processo de identificação da faixa de consumo formado por estruturas condicionais. Com a faixa de consumo identificada, o cálculo do gasto de água é realizado para cada faixa. O mesmo ocorre para o cálculo do gasto com tratamento de esgoto. Em seguida os gastos são somados e a estimativa da fatura é informada. Todo o processo desde a definição das tarifas até a informação ao usuário totaliza 24 blocos.

O consumo de água da semana usado no cálculo da fatura é capturado do ThingSpeak, da mesma forma como é feito na função 1, mostrado na seção 3.4.1.



## 4. APLICAÇÃO

O sistema de monitoramento foi aplicado na residência do próprio autor, durante o período de uma semana.

### 4.1. Instalação do hidrômetro e locação do circuito

A instalação do hidrômetro ultrassônico foi realizada no sótão da residência, em um percurso anterior à caixa d'água. O ideal seria instalá-lo logo após o hidrômetro da COPASA, mas na residência em questão existe uma grade de ferro que impede o acesso ao cano. A Figura 24 mostra o hidrômetro ultrassônico instalado, com o circuito eletrônico ao lado.

**Figura 24 – Hidrômetro ultrassônico e circuito instalado**



Fonte: Elaborado pelo autor

Apesar do sótão ser um local pouco acessível, possui a vantagem de ser coberto, eliminando o risco de molhar o circuito eletrônico em dias de chuva. Para armazenar o circuito, uma caixa plástica de 13x13 cm foi utilizada. A caixa já possui os furos para passagem dos cabos de alimentação e de comunicação com o hidrômetro, que foram conectados aos bornes conforme Figura 25.

**Figura 25 – Conexão dos cabos ao circuito eletrônico**

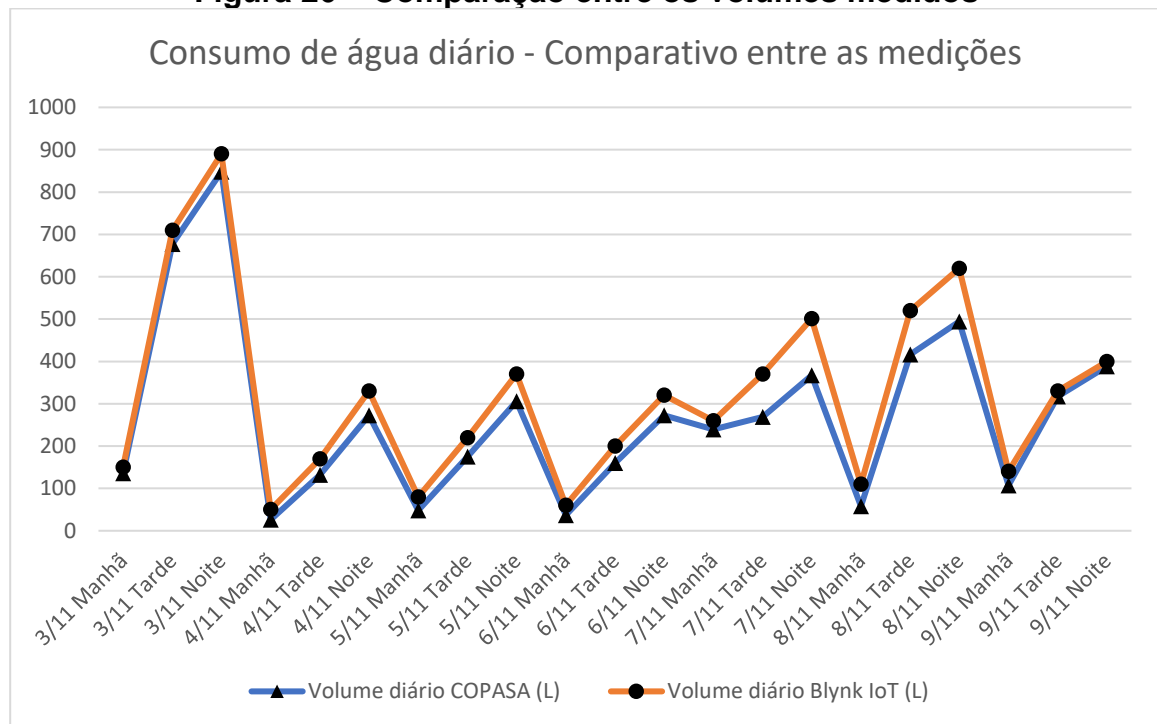


Fonte: Elaborado pelo autor

## **4.2. Coleta e análise dos dados**

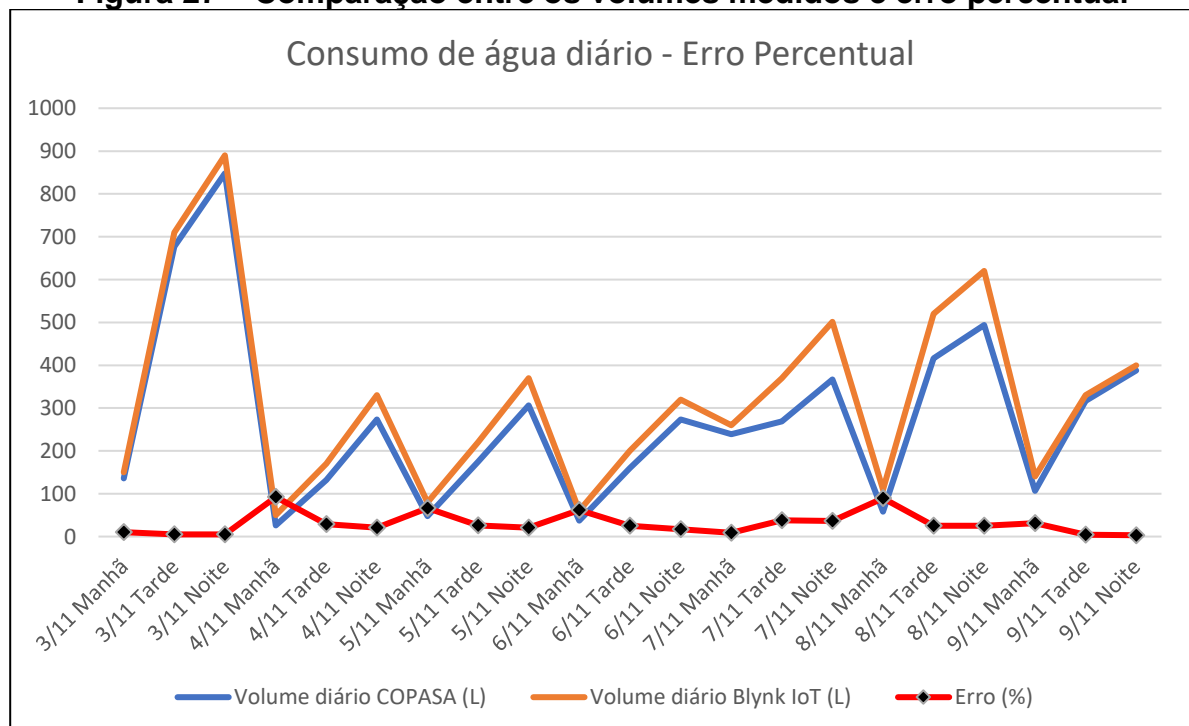
### *4.2.1. Resultados do consumo informado pela Blynk IoT*

Durante o período de testes os dados de consumo foram coletados 3 vezes ao dia, nos períodos da manhã, tarde e noite. A cada coleta foi tirada uma fotografia do hidrômetro da concessionária, e posteriormente os dados de consumo foram comparados com os volumes informados no aplicativo Blynk IoT. A Figura 26 mostra a comparação entre os valores medidos.

**Figura 26 – Comparação entre os volumes medidos**

Fonte: Elaborado pelo autor

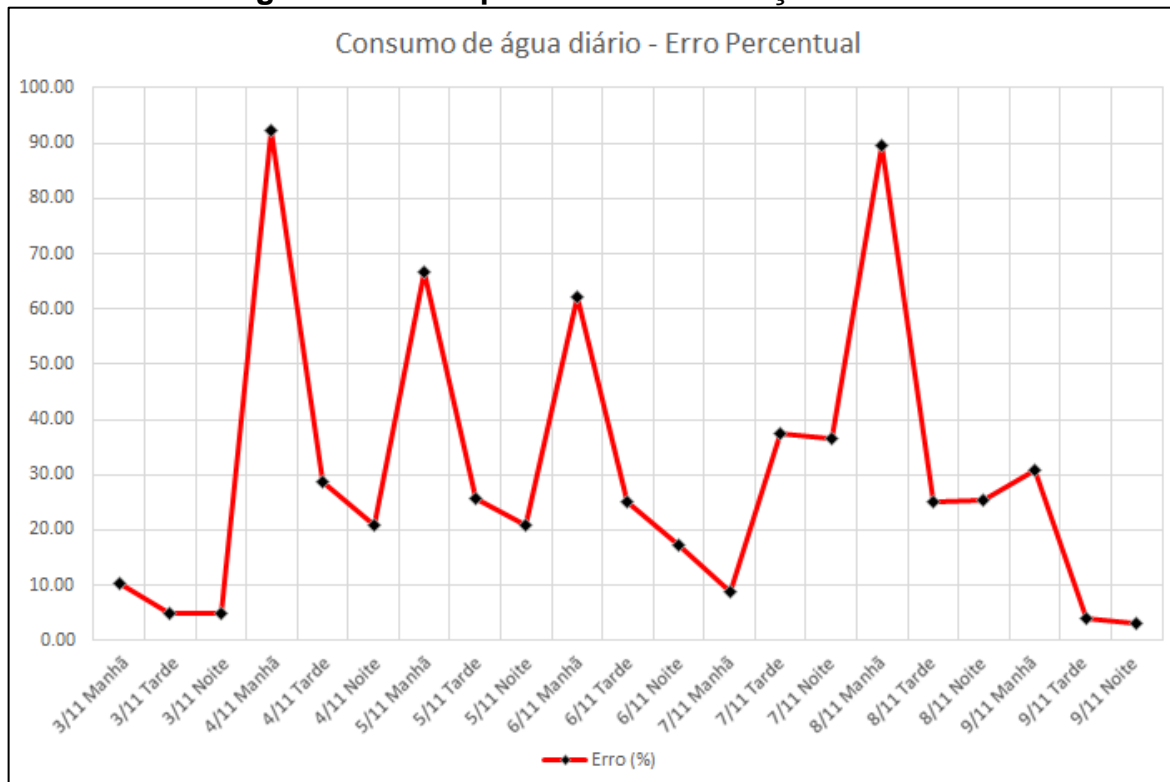
Observando a Figura 26, percebe-se que o sistema de monitoramento acompanhou as medições da concessionária de água, mas obteve sempre um valor superior. A Figura 27 mostra o erro percentual para os mesmos dados da Figura 26.

**Figura 27 – Comparação entre os volumes medidos e erro percentual**

Fonte: Elaborado pelo autor

A Figura 28 mostra o erro percentual separadamente, para que seus valores possam ser visualizados mais de perto.

**Figura 28 – Erro percentual da medição de volume**



Fonte: Elaborado pelo autor

Analisando o gráfico da Figura 28, percebe-se um erro percentual alto em alguns pontos, entre 60% e 90%. Esses erros seguem um padrão, ocorrendo no período da manhã. Eles se referem as medições realizadas entre 23:59 e a primeira coleta de dados do dia seguinte, aproximadamente às 10:00. Apesar de percentualmente o erro ser grande, o consumo em litros a qual ele se refere é baixo, entre 26 e 58 litros, valor pequeno quando comparado com a medição do consumo total da semana, que foi em torno de 3000 litros. Logo, o sistema se mostra mais confiável para medições de grandes consumos. O motivo da diferença entre as medições é atribuído a dois fatores. O primeiro deles é a vazão mínima de medição dos hidrômetros, que é diferente. O hidrômetro da COPASA é do tipo mecânico, e a vazão mínima na qual ele começa a contabilizar o consumo é de 15 L/h, enquanto o Hydrus 1.3 utilizado no sistema possui uma vazão mínima de trabalho igual a 6,25 L/h. Logo, é de se esperar que as medições sejam diferentes, pois em momentos de baixa vazão, há volume sendo contabilizado na medição do sistema de monitoramento, enquanto no hidrômetro mecânico não. Isto justifica a característica do erro percentual

de ter o valor máximo durante as manhãs, visto que este momento ocorre posteriormente a toda uma noite de vazões baixas, comumente encontradas em residências, provocadas por vazamentos na caixa d'água dos vasos sanitários. Na residência de aplicação existem três vasos sanitários, contribuindo para a existência deste fator. Este consumo que não é medido pelo hidrômetro mecânico compõe a submedição, uma das perdas da concessionária de água no faturamento.

O segundo fator responsável pelo erro de medição como um todo, está na possibilidade de existência de fluxo reverso. Ao realizar a leitura do hidrômetro da COPASA no período da manhã de um dos dias posteriores à semana de testes, foi constatado um volume inferior ao do dia anterior no período noturno, mostrando que houve fluxo reverso. O circuito eletrônico do sistema de monitoramento não foi desenvolvido levando em consideração a direção do fluxo. Logo, nos momentos em que houve fluxo reverso, o sistema de monitoramento considerou o volume medido como consumo natural da residência. O fluxo reverso não serve como justificativa para o comportamento de maior erro percentual durante as manhãs, pois pode ocorrer em qualquer horário.

O erro percentual referente ao volume total medido durante a semana também foi calculado. Para o cálculo, o consumo semanal considerado foi obtido somando os consumos diários informados na plataforma Blynk IoT. O hidrômetro da COPASA mediu 2949 litros consumidos durante a semana, enquanto o sistema de monitoramento mediu 3431 litros.

$$\text{Erro}_{\text{total}} = \frac{\text{Leitura}_{\text{COPASA}} - \text{Consumo}_{\text{Blynk}}}{\text{Leitura}_{\text{COPASA}}} \times 100 = \frac{2949 - 3431}{2949} \times 100 = 16,34\% \quad (3)$$

#### 4.2.2. Avaliação da skill baseada no sistema especialista

As duas funções da *skill* foram avaliadas separadamente. Como o sistema especialista captura a informação do consumo diretamente do ThingSpeak, a análise da primeira função foi feita comparando o valor disponível na plataforma ThingSpeak com o consumo informado pela Blynk IoT. Os dados foram exatamente os mesmos, como o esperado.

A segunda função da *skill*, de estimar a fatura, foi avaliada comparando o valor informado pela assistente virtual com o valor calculado pelo próprio autor, seguindo o

método da COPASA. O sistema especialista conseguiu calcular a fatura para todos os volumes testados, em faixas diferentes de consumo, como mostrado na tabela 1.

**Tabela 1 – Fatura calculada para diferentes faixas de consumo**

Volume (L)	Fatura calculada pelo autor	Fatura informada pela assistente virtual
10	R\$30,64	R\$31,00
2000	R\$36,98	R\$37,00
8500	R\$70,16	R\$70,00
15000	R\$132,70	R\$133,00
22000	R\$240,62	R\$241,00
30000	R\$386,20	R\$386,00
45000	R\$679,17	R\$679,00

**Fonte: Elaborado pelo autor**

Avaliou-se, ainda, a estimativa de fatura informada pela assistente virtual referente ao consumo total da semana de aplicação. A avaliação foi feita comparando a estimativa informada com a fatura calculada para o consumo de água medido pelo hidrômetro da COPASA durante a semana de aplicação. A Alexa estimou uma fatura de R\$42,00, enquanto que o valor calculado com o consumo medido pela COPASA foi R\$39,99. O erro percentual foi -5,03%.

## 5. CONCLUSÃO

O sistema de monitoramento proposto responde como esperado, sendo capaz de informar o consumo de água da residência de formas diferentes para o usuário, com dados que representam a realidade do consumo.

A diferença entre as medições do sistema de monitoramento e as medições da COPASA, são em parte, naturais da tecnologia dos hidrômetros. Uma futura adaptação do circuito eletrônico que permita a identificação da direção do fluxo seria um caminho adequado para tornar o sistema mais próximo do consumo real, diminuindo uma parcela da diferença existente.

O sistema de monitoramento se mostra como uma ferramenta flexível, visto que outros hidrômetros capazes de se comunicar por pulsos de onda quadrada, semelhantemente ao Hydrus 1.3, podem ser utilizados, necessitando de poucas modificações no código, não interferindo na distribuição de dados para as plataformas.

Quanto ao armazenamento de dados, a plataforma ThingSpeak atendeu perfeitamente as necessidades do sistema, visto que os dados podem ser consultados a qualquer momento, sendo excluídos da plataforma após o período de 1 ano, que é suficiente para uma aplicação residencial.

O sistema desenvolvido pode ainda ser alterado para monitorar o consumo mensal, possibilitando estimativas de fatura para comparativo com o valor real pago pelo morador. Entretanto, algumas situações podem atrapalhar as medições do sistema, como a queda de energia, que retorna o consumo para zero. Portanto, para tornar o sistema mais confiável para que seja utilizado continuamente sem problemas, é necessário incrementar o consumo diretamente na plataforma ThingSpeak, e não no microcontrolador, como é feito atualmente.

As funções da *skill* baseada em um sistema especialista funcionam como o esperado, e dependem somente de dados de consumo mais confiáveis para que melhorem o seu desempenho.

## REFERÊNCIAS

AESBE – Associação Brasileira das Empresas Estaduais de Saneamento. **Guia prático de procedimentos para estimativa de submedição no parque de Hidrômetros**. Brasília, 2015

AGÊNCIA REGULADORA DE SERVIÇOS DE ABASTECIMENTO DE ÁGUA E DE ESGOTAMENTO SANITÁRIO DO ESTADO DE MINAS GERAIS. **Resolução ARSAE-MG 154, de 28 de junho de 2021**. Autoriza a Companhia de Saneamento de Minas Gerais - Copasa a aplicar aos serviços públicos de abastecimento de água e de esgotamento sanitário [...]. Belo Horizonte: ARSAE-MG, 2021. Disponível em: [http://www.arsae.mg.gov.br/wp-content/uploads/2021/06/Resoluc%CC%A7a%CC%83o\\_Copasa\\_PosCP23.pdf](http://www.arsae.mg.gov.br/wp-content/uploads/2021/06/Resoluc%CC%A7a%CC%83o_Copasa_PosCP23.pdf). Acesso em: 06 nov. 2022.

AI-THINKER (China). **ESP-01**: product features/specifications. Product Specifications. 2017. Especificações do fabricante. Disponível em: [http://www.ai-thinker.com/pro\\_view-60.html](http://www.ai-thinker.com/pro_view-60.html). Acesso em: 07 maio 2022.

BLYNK IOT. **Blynk IoT platform, Templates**. 2022. Disponível em: <https://blynk.io/>. Acesso em: 05 nov. 2022.

BRASIL. SISTEMA NACIONAL DE INFORMAÇÕES SOBRE SANEAMENTO. **Painel de Saneamento**: painel de indicadores. Painel de Indicadores. 2020. Disponível em: [http://appsniis.mdr.gov.br/indicadores/web/agua\\_esgoto/mapa-agua](http://appsniis.mdr.gov.br/indicadores/web/agua_esgoto/mapa-agua). Acesso em: 07 maio 2022.

CASTRO, Javier; PSOTA, James. The Specification, Design, and Implementation of a Home Automation System. **Introductory Digital Systems Laboratory, MIT**, 2004.

CÉSAR DA SILVA, Júlio. **SISTEMA ESPECIALISTA CONJUGADO A UM SISTEMA CAD PARA AVALIAR E DIAGNOSTICAR OS CONHECIMENTOS DE UM ESTUDANTE SOBRE COTAGEM NO DESENHO TÉCNICO**. 1994. 90 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1994.

COSTA, Mathias Scroccaro. **Desenvolvimento de dispositivo de telemetria para hidrômetros sob a ótica IoT: Estudos de geração e consumo energético**. 2020. 68 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 2020.

DIEHL METERING (Ansbach). **Medidor de água ultrassônico Type 161A/B**: guia de instalação. Guia de instalação. 2021. Disponível em: <https://www.diehl.com/metering/en/products-services/water-metering/hydrus-13-fr/>. Acesso em: 30 out. 2022.

DURKIN, John. **Expert Systems: Design and Development**. 2. ed. Englewood Cliffs: Macmillan, 1994. 800 p.



GONÇALVES JUNIOR, Marcos José Rangel. **Desenvolvimento de um medidor eletrônico de vazão sob o paradigma da internet das coisas**. 2019. 43 f. Dissertação (Mestrado) - Curso de Sistemas Aplicados à Engenharia e Gestão, Instituto Federal de Educação Ciência e Tecnologia Fluminense, Campos dos Goytacazes, 2019.

HEINZLE, Roberto. **PROTÓTIPO DE UMA FERRAMENTA PARA CRIAÇÃO DE SISTEMAS ESPECIALISTAS BASEADOS EM REGRAS DE PRODUÇÃO**. 1995. 145 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Santa Catarina, 1995.

IZQUIERDO, Javier Luis Cánovas; CABOT, Jordi. JSONDiscoverer: Visualizing the schema lurking behind JSON documents. **Knowledge-Based Systems**. Barcelona, p. 52-55. 1 abr. 2016.

KOLBAN, Neil. Overview. In: KOLBAN, Neil. **O livro de Kolban sobre ESP32**. [S.L.]: [S.n.], 2017.

LEE, Aaron *et al.* Machine Learning Has Arrived! **Ophthalmology**, [S.L.], v. 124, n. 12, p. 1726-1728, dez. 2017. Elsevier BV.  
<http://dx.doi.org/10.1016/j.ophtha.2017.08.046>.

MACHADO JUNIOR, Ireno Gonçalves. **Estudo da Colheita de Energia para Medição de Vazão em Aplicações de Internet das Coisas**. 2020. 96 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2020.

MAGRANI, Eduardo. **A Internet das Coisas**. Rio de Janeiro: Fgv Editora, 2018. 190 p.

Organização das Nações Unidas. **Agenda 2030 para o Desenvolvimento Sustentável**: transformando nosso mundo: a agenda 2030 para o desenvolvimento sustentável. Transformando Nosso Mundo: A Agenda 2030 para o Desenvolvimento Sustentável. 2015. Disponível em: <https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustentavel>. Acesso em: 08 maio 2022.

OVADIA, Steven. Automate the Internet With “If This Then That” (IFTTT). **Behavioral & Social Sciences Librarian**. New York, p. 208-211. ago. 2014.

PEREIRA E SILVA, Lucas Damaceno. **Redução de perdas aparentes no sistema de distribuição de água de Buritirana/Palmas - TO**. 2022. 86 f. Dissertação (Mestrado) - Curso de Engenharia Ambiental, Universidade Federal do Tocantins, Palmas, 2022.

RAKHMAWATI, Nur Aini *et al.* INDONESIA'S PUBLIC APPLICATION PROGRAMMING INTERFACE (API). **Jurnal Penelitian Pos Dan Informatika**. Surabaya, p. 85-96. 2 dez. 2019.

SIBUYA, Celso Toshio. **Implementação e uso de atributos de escolha de armazenamento em ambiente de Nuvem Privada**. 2019. 132 f. Dissertação

(Mestrado) - Curso de Engenharia de Computação, Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2019.

SAMUEL, Arthur. Some Studies in Machine Learning Using the Game of Checkers. **Ibm Journal Of Research And Development**. [S.l.], p. 211-229. jul. 1959.

SILVA FILHO, Sandoval Santos. **Proposta de sistema microcontrolado para aquisição de dados e automação de ambientes residenciais**. 2018. 55 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade Estadual de Feira de Santana, Feira de Santana, 2018. Cap. 2.

SILVA JÚNIOR, João Ferreira da. **Detecção de perdas em sistemas de distribuição de água através de rede de sensores sem fio**. 2017. 166 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, 2017.

TEZA, Vanderlei Rabelo. **ALGUNS ASPECTOS SOBRE A AUTOMAÇÃO RESIDENCIAL - DOMÓTICA**. 2002. 106 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2002.

THINGSPEAK. **ThingSpeak for IoT Projects**. 2022. Disponível em: <https://thingspeak.com>. Acesso em: 06 nov. 2022.

TIWARY, Gyan P.; STROULIA, Eleni; SRIVASTAVA, Abhishek. Compression of XML and JSON API Responses. **IEEE Access**, Indore, v. 9, p. 57426-57439, 20 abr. 2021.

VENDEMIATTI, Caíque. **SISTEMA DE MONITORAMENTO DO CONSUMO DE ÁGUA EM TEMPO REAL**. 2020. 145 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual Paulista "Júlio de Mesquita Filho", Sorocaba, 2020.

VOICEFLOW. **Where teams build chat and voice assistants**. 2022. Disponível em: <https://www.voiceflow.com/>. Acesso em: 06 nov. 2022.

## APÊNDICE A – CÓDIGO DO MICROCONTROLADOR ESP-01

```
//Identificadores da aplicação Blynk
#define BLYNK_TEMPLATE_ID "TMPLisNt0fHs"
#define BLYNK_DEVICE_NAME "TemplateVoiceFlow01"
#define BLYNK_PRINT Serial
#define BLYNK_AUTH_TOKEN "0vUypNDzGeVr7ecKqV8G4Tv3kxhx6UzS"
//-----
#define sensor 1
#define scl_esp 0
#define sda_esp 2

//Inclusão de bibliotecas
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <RtcDS3231.h>

RtcDS3231<TwoWire> rtcObject(Wire); //declaração para permitir conexão I2C com o DS3231
WiFiClient client; //declaração para permitir envio de dados ao ThingSpeak

char ssid[] = "casa_ers"; //SSID da rede wi-fi
char pass[] = "casa_4576"; //senha da rede wi-fi
char auth[] = "0vUypNDzGeVr7ecKqV8G4Tv3kxhx6UzS"; //token de autenticação do dispositivo para conexão ao Blynk iot
char server[] = "api.thingspeak.com"; //endereço API ThingSpeak
String api_key = "DVL5G03WIJWEQ4DS"; //chave da API ThingSpeak

//Variáveis globais
double vol_d = 0; //receberá o volume diário
double vold_antes = 0; //receberá o volume diário anterior
double vol_s = 0; //receberá o volume acumulado durante a semana
double vols_antes = 0; //receberá o volume semanal antigo
int seg; //receberá a contagem de segundos do DS3231
int seg_antes; //receberá a contagem de segundos antiga do DS3231
double pulsos = 0; //receberá o número de pulsos dados pelo sensor de volume
int cont_d = 0; //conterá quantos dias da semana se passaram
int cont_seg = 0; //conterá quantos segundos se passaram
int dia; //receberá a contagem de minutos do DS3231
int dia_ant = 0; //receberá a contagem de minutos antiga do DS3231

//FUNÇÕES
void ICACHE_RAM_ATTR conta_pulsos(); //armazenamento da função na RAM
void ICACHE_RAM_ATTR conta_pulsos(){ //função propriamente dita, contará os pulsos recebidos pelo sensor de volume
  pulsos = pulsos + 0.1;
}

void ICACHE_RAM_ATTR enviats(); //armazenamento da função na RAM
void ICACHE_RAM_ATTR enviats(void){ //função propriamente dita, envia dados ao ThingSpeak
  if (client.connect(server, 80))
  {
    String postStr = api_key;
    postStr += "&field1=";
    postStr += String(vol_s, 1); //escreve o volume como uma string
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\r\n");
    client.print("Host: api.thingspeak.com\r\n");
    client.print("Connection: close\r\n");
    client.print("X-THINGSPEAKAPIKEY: " + api_key + "\r\n");
    client.print("Content-Type: application/x-www-form-urlencoded\r\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
    client.print("https://api.thingspeak.com/update?api_key=DVL5G03WIJWEQ4DS&field1=9");
  }
  client.stop();
}
```

```

void setup() {
  sei(); //habilita as interrupções
  pinMode(sensor, INPUT); //define o pino "sensor" como entrada
  attachInterrupt(digitalPinToInterrupt(sensor), conta_pulsos, RISING); //declara a interrupção no pino "sensor"
  Wire.begin(scl_esp,sda_esp); //define os pinos de SCL e SDA para o ESP01
  rtcObject.Begin(); //inicia comunicação com o DS3231
  //RtcDateTime currentTime = RtcDateTime(22, 11, 02, 17, 38, 00); //define a data e hora atuais
  //rtcObject.SetDateTime(currentTime); //configura o DS3231 com a data definida
  RtcDateTime currentTime = rtcObject.GetDateTime();
  Blynk.begin(auth, ssid, pass); //inicia conexão com a Blynk iot
  delay(5000);
  dia_ant = currentTime.Day(); //pega o dia atual do DS3231
  seg = currentTime.Second(); //pega o segundo atual do DS3231
  seg_antes = seg;
}

void loop() {
  Blynk.run(); //permite conexão com a Blynk iot
  if((seg >= (seg_antes + 1)) || (seg < seg_antes)){
    seg_antes = seg;
    vol_d = vold_antes + 100*pulsos;
    vold_antes = vol_d;
    vol_s = vols_antes + pulsos;
    vols_antes = vol_s;
    pulsos = 0;
    Blynk.virtualWrite(V0, vol_d); //escreve o volume diário no pino V0 da Blynk iot
    Blynk.virtualWrite(V1, "{\"status\":\"" + String(vol_d) + "\"}"); //envia json para o pino V1, o dado poderá ser lido pelo Voiceflow
    if((dia > dia_ant) || (dia < dia_ant)){
      dia_ant = dia;
      cont_d++;
      vol_d = 0;
      vold_antes = vol_d;
      if(cont_d >= 7){
        vol_s=0;
        vols_antes=vols_s;
      }
    }
    cont_seg++;
    if(cont_seg>=15){
      enviats();
      cont_seg=0;
    }
  }
  RtcDateTime currentTime = rtcObject.GetDateTime();
  seg = currentTime.Second();
  dia = currentTime.Day();
}

```