

## SISTEMA DE MONITORAMENTO IOT PARA GERENCIAMENTO DO FLUXO DE LÍQUIDOS NA INDÚSTRIA 4.0

**Lucas Gabriel Bastos Marques<sup>(1)</sup>**

Concluinte do Curso de Engenharia Civil do Centro Universitário do Estado do Pará.

**Ozias dos Santos Cardoso Neto<sup>(2)</sup>**

Concluinte do Curso de Engenharia Civil do Centro Universitário do Estado do Pará.

**MSc Luciana de Azevedo Vieira<sup>(3)</sup>**

Graduada em Engenharia Elétrica pela Universidade Federal do Pará.

Mestre em Engenharia de Controle e Automação pela Universidade Federal do Pará.

### RESUMO

A eficiência e o controle são fundamentais na indústria, logo, o gerenciamento preciso do fluxo de líquidos desempenha um papel crítico em diversos fatores, garantindo a qualidade do produto, a segurança dos processos e o uso eficiente dos recursos, e assim, reduzindo custos e minimizando os impactos ambientais. Logo, a ferramenta apresentada nesse trabalho possibilita a medição precisa do desempenho de sistemas e equipamentos, fornecendo dados que podem ser usados para otimização e manutenção preventiva. Neste momento do trabalho, são apresentados as tecnologias e dispositivos de software, descrevendo as 6 etapas para a fundamentação do trabalho e os cálculos desenvolvidas para a execução do projeto, sendo elas: Conexão dos dispositivos; Criação do Canal no ThingSpeak; Desenvolvimento da programação para o ESP32; Desenvolvimento do Aplicativo no Thunkable; Armazenamento de Dados no Airtable; Utilização do Aplicativo. Uma melhor gestão dos fluxos de líquidos, com um monitoramento totalmente portátil, rápido e prático, objetivando uma maior eficiência em todos os âmbitos. Assim como, identificar problemas de forma mais rápida e prática, diminuindo gastos e prejuízos.

**PALAVRAS-CHAVE:** Aplicação móvel; IOT; Multiplataformas.

### ABSTRACT

Efficiency and control are paramount in the industry; therefore, precise management of liquid flow plays a critical role in various factors, ensuring product quality, process safety, and efficient resource utilization, thereby reducing costs and minimizing environmental impacts. The tool presented in this work enables accurate measurement of the performance of systems and equipment, providing data that can be used for optimization and preventive maintenance. In this work, technologies and software devices are presented, describing the 6 steps for the project foundation and the calculations developed for project execution, namely: Device Connection; Creation of the Channel on ThingSpeak; Programming Development for ESP32; Application Development on Thunkable; Data Storage on Airtable; Application Usage. A better management of liquid flows, with completely portable, fast, and practical monitoring, aiming for greater efficiency in all aspects. Also, identifying problems more quickly and practically, reducing expenses and losses. The main objective of the work is to develop a real-time IoT monitoring system for fluid flows.

**KEYWORD:** Mobile Application; IoT; Multi-platforms.

### INTRODUÇÃO

A eficiência e o controle são fundamentais na indústria, logo, o gerenciamento preciso do fluxo de líquidos desempenha um papel crítico em diversos processos. A Indústria 4.0 representa uma revolução na forma como as indústrias operam, integrando tecnologias avançadas de automação, digitalização e conectividade. Dessa forma, o monitoramento do fluxo de líquidos desempenha um papel crucial em uma variedade de aplicações industriais, ambientais e de engenharia (PROPEQ, 2019).

Nesse sentido, refere que o monitoramento do fluxo de líquidos se torna imprescindível para o: Controle de Processos Industriais e Economia dos Recursos; Detecção de Vazamentos e Segurança; Medição de Desempenho; Gerenciamento de Recursos Hídricos; Monitoramento de Sistemas de Distribuição e desenvolvimento científico,

pois, por meio desse monitoramento, os recursos usados pelas empresas se tornam bem empregados (OLIVEIRA, 2019).

Isso, segundo Zanuzzo (2017) ocorre devido ao seu uso ser essencial para indústrias relacionadas a petroquímica, farmacêutica, de alimentos e bebidas, e manufatura, pois permite o controle preciso do fluxo de líquidos, garantindo a qualidade do produto, a segurança dos processos e o uso eficiente dos recursos, assim, reduzindo custos e minimizando os impactos ambientais.

Além disso, essa ferramenta possibilita a medição precisa do desempenho de sistemas e equipamentos, fornecendo dados que podem ser usados para otimização e manutenção preventiva, garantindo um suprimento adequado e confiável. Desse modo, torna-se necessário que haja a atualização frequente do campo científico, para atender as demandas, planejar novas formulações e teorias, e tornar o processo mais preciso e validado (DINGMAN, 2002).

Nessa perspectiva, o monitoramento do fluxo de líquidos por meio de tecnologia de sistema, desempenha um papel fundamental em uma ampla gama de aplicações e setores, pois contribui para o controle de processos, eficiência operacional, segurança e conformidade regulatória, além de fornecer dados essenciais para a melhora da qualidade, economia de recursos e proteção do meio ambiente (OLIVEIRA; SANTOS; RODRIGUES, 2014)

Em suma, o objetivo do estudo é: Desenvolver um sistema IoT de monitoramento em tempo real dos fluxos de fluidos. Para isso, se utilizará do ESP32 e o ThingSpeak para criar um sistema de monitoramento de fluxo de líquidos com aplicação ampla na Indústria 4.0, por meio da linguagem Arduino para o desenvolvimento do projeto.

## REVISÃO BIBLIOGRÁFICA

- Internet das Coisas (IoT)

De acordo com Galeale *et al.*, (2016), muitos estudiosos referem que a origem do termo “Internet da Coisas (IoT)” pode ser evidenciada na publicação do artigo "When Things Start to Think" de Neil Gershenfeld (1999), pois o autor relata um ambiente onde os objetos processam informações, haja vista o exemplo que ela aborda, em que o primeiro eletrodoméstico inteligente foi uma geladeira, lançada pela LG no ano de 2002, a qual permitia a conexão com a internet, podendo ser utilizada para refrigerar os alimentos e ao mesmo tempo, navegar na internet.

Com o avanço dos serviços tecnológicos, Aliger (2019), confere que a comunicação sem fio proporcionada pelos dispositivos móveis, é um fator que aumenta o interesse das grandes empresas, uma vez que isso agrega maior capacidade de monitoramento, agilidade operacional e possui maior custo-benefício para o emprego de tecnologias no mercado de trabalho. De fato, com o estabelecimento da Internet das Coisas (IoT), houve o estreitamento da relação entre o mundo virtual e físico, por meio da transmissão de dados, processos e redes de pessoas.

Nesse sentido, Atzori (2010) aborda que a Iot é um local central onde objetos, sites, dispositivos podem interagir e cooperar entre si para atingir objetivos comuns. Logo, essa ferramenta pode ser visualizada por meio de sensores, transmissores ou receptores que facilitam a comunicação entre objetos de forma virtual e os indivíduos, constituindo um novo paradigma de comunicação.

- ThingSpeak

Em seguida, Dornelas e Oliveira (2017), referem que a utilização de multiplataformas é uma ferramenta facilitadora para a formulação de aplicativos de baixo custo, mas que configuram os processos de forma efetiva. Assim, o aplicativo criado no estudo captura os dados da plataforma IoT, denominada de ThingSpeak, a qual recebe e armazena os dados enviados em tempo real pelo ESP-32 que realiza a leitura do fluxo por meio do sensor de fluxo YF-S201.

Nesse sentido, essa plataforma permite a coleta e armazenamento de dados em nuvem, o que possibilita uma análise e visualização desses dados em tempo real. Além disso, o ThingSpeak faz com que os elementos coletados possam ser analisados por meio de softwares matemáticos, contemplados por gráficos, além de permitir a visualização personalizada da programação HTML, sendo as ações próprias mais utilizadas dessa ferramenta, o

envio de tweets, reações a eventos específicos, comandos via HTTP e o utilizado no estudo em questão, que se refere a controles temporizados (MATHWORKS, 2018).

- Sensor de fluxo YF-S201

Por sua vez, reconhece-se que os fluxostatos ou sensores de fluxo do tipo YF-S201, são capazes de mensurar o fluxo de líquidos, ar ou gases dentro de uma tubulação. Além disso, sua variabilidade, permite atender amplamente a demanda do usuário, sendo este medidor geralmente do tipo magnético, constituído de pás rotativas que têm sua velocidade de giro firmada a partir da quantidade de fluxo que passa por elas, baseado na utilização do método de Hall, o qual tem função de detectar campos magnéticos que são externos às cargas em movimento de um determinado material semiconductor (MUNDIM, 1999).

Nessa perspectiva, esse sensor se constitui como um medidor de vazão, que permite leitura de 1 até 30 L/min. O seu emprego no estudo se deve ao fato da disponibilidade no mercado e o baixo poder aquisitivo, que faz com que as empresas que o utilizem tenham uma eficácia no desenvolvimento de seus projetos, sem que haja relativa despesa (SANTOS; LIMA, 2017).

- ESP-32

Outrossim, a utilização do ESP-32, permite facilidade de programação, além de ser um dispositivo de baixo consumo de energia, alto desempenho e confiabilidade de dados coletados, haja vista que este consiste em um microprocessador de baixa potência dual core Tensilica Xtensa 32-bit LX6 que possui suporte embutido às redes Wi-Fi, Bluetooth, e memória flash integrada (AMERICO e IZIDORO, 2022).

A partir disso, reconhece-se que este microcontrolador é um dos controladores mais utilizados no mercado, devido suas notórias características, como velocidade de processamento, conectividade e acessibilidade. Por conseguinte, o ESP-32 possui grande capacidade de armazenamento com até 4MB de memória flash integrada, quando em comparação ao microcontrolador Arduino, onde possui o dobro de memória flash em relação ao modelo ATmega 2560 (KOLBAN, 2018).

Assim, utilizou-se esta ferramenta para automatização do sistema de monitoramento de fluxo de líquidos, haja vista a usabilidade do ESP-32 para execução do projeto, que objetiva o baixo custo e alta conectividade. Além disso, a fácil programação e compatibilidade com diversos softwares, principalmente o Arduino IDE, faz com que este seja um dos mais utilizados no mercado de trabalho (IBRAHIM, 2017).

- Thinkable

De acordo com Prnewswire (2018), muitas empresas ainda têm dificuldade em desenvolver aplicativos, pois reconhecem o processo de programação deles como dificultoso. Sendo assim, companhias começaram a criar projetos de sites e aplicativos sem o uso de programação, como por exemplo o Thinkable, criado pelo Instituto de Tecnologia de Massachusetts (MIT) em conjunto com o Google e fundada por Arun Saigal e WeiHua Li no ano de 2015.

Essa plataforma permite a criação de aplicativos por meio do uso de blocos lógicos que substituem a programação. Assim, ela permite que não-programadores possam desenvolver projetos por meio de dois componentes principais: O Designer, onde é possível criar a interface do usuário adicionando botões, blocos de texto ou outros elementos interativos; e a Linguagem de Blocos, que permite criar toda a inteligência da aplicação (THATCHER et al., 2006).

Dessa forma, a Thinkable possibilita programar projetos de forma Cross Platform, fazendo com que cada aplicativo criado possa ser gerado para as plataformas Android e iOS, facilitando a utilização, principalmente para programadores iniciantes no desenvolvimento de aplicativos móveis, pois, conforme o CEO e fundador da Thinkable “a tecnologia móvel só irá perceber seu total potencial para ajudar pessoas quando qualquer um puder criar um aplicativo para solucionar seus problemas” (WINN, 2019).

- **Airtable**

Na última década, a gestão de dados ganhou destaque, impulsionando o desenvolvimento de ferramentas inovadoras. O Airtable, surgido em 2012, destaca-se como uma plataforma de gerenciamento de banco de dados que combina a flexibilidade de uma planilha com a robustez de um banco de dados (LIU *et al.*, 2012).

Sua arquitetura se baseia em células, tabelas e blocos, proporcionando uma interface intuitiva (Alda, 2019). Oferece recursos como criação de relações entre tabelas, anexação de arquivos e integração com outras ferramentas. O Airtable encontrou aplicação em diversos setores, desde desenvolvimento até marketing, adaptando-se para rastreamento de projetos, gerenciamento de inventário e acompanhamento de leads.

Entre suas vantagens estão a flexibilidade, facilidade de uso e colaboração em tempo real. Contudo, desafios incluem a gestão de grandes conjuntos de dados e a curva de aprendizado para usuários menos familiarizados com conceitos de banco de dados. Comparado a ferramentas tradicionais como Excel e Google Sheets, o Airtable destaca-se por sua abordagem inovadora (FERREIRA, SOARES e BARBOSA, 2023).

No horizonte futuro, espera-se a contínua evolução do Airtable, mantendo-se como uma ferramenta adaptável às demandas cambiantes do ambiente corporativo (DIRK e MADDOX, 2018). Em resumo, o Airtable oferece uma abordagem revolucionária para o gerenciamento de dados, sendo uma escolha valiosa para empresas e profissionais que buscam eficiência e adaptabilidade em suas práticas de gerenciamento de informações.

- **Dashboard**

Os Dashboards desempenham um papel crucial na visualização eficiente de dados, consolidando informações de maneira atrativa para monitoramento e análise. Por conseguinte, eles podem ser categorizados em: operacionais, estratégicos e analíticos, logo, esses dashboards atendem a diversas necessidades organizacionais, proporcionando desde informações em tempo real até suporte à tomada de decisões a longo prazo (FEW, 2006; WANG *ET AL.*, 2017).

Por conseguinte, o design é fundamental, pois enfatizam elementos visuais claros, layout intuitivo e escolha criteriosa de cores para garantir rápida compreensão. Assim, a integração de dados, conforme discutido por Shneiderman e Plaisant (2006), é essencial para fornecer uma visão abrangente do desempenho organizacional, conectando informações de várias fontes.

Quanto às ferramentas, autores como Few (2006) exploram opções como planilhas avançadas e plataformas específicas de business intelligence, como Tableau, Power BI e Qlik. Além disso, Bajwa e Kaur (2017), destaca a importância da usabilidade e da experiência do usuário, enquanto desafios éticos relacionados à precisão dos dados discutidos pelos autores supracitados.

Outrossim, Sarikaya *et al.*, (2018) exploram tendências futuras ao destacarem o papel em evolução dos dashboards, incorporando tecnologias como inteligência artificial e realidade aumentada, pois estes podem aprimorar a interatividade e a capacidade analítica. Essa revisão bibliográfica oferece uma compreensão abrangente do papel vital dos dashboards na interpretação visual de dados.

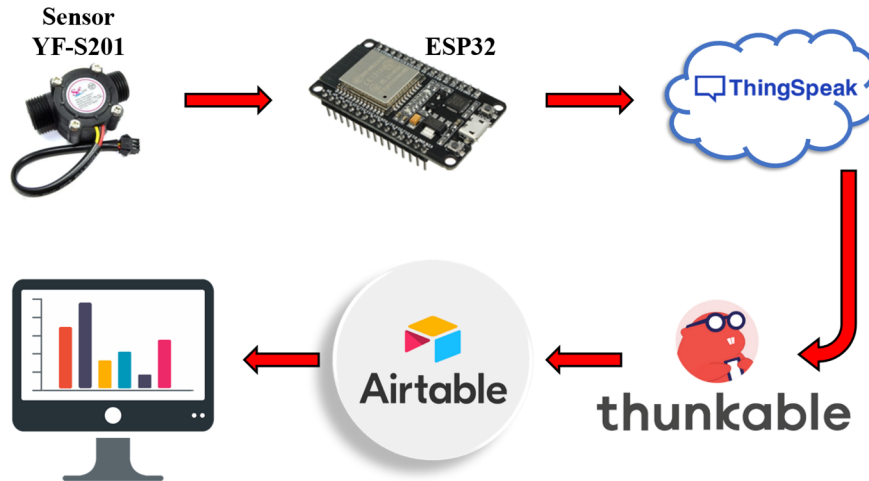
## **METODOLOGIA**

O projeto do estudo é apresentado por meio do Fluxograma 1, e consiste em um sistema de monitoramento de fluxo de água opera de maneira integrada, coletando e disponibilizando informações de maneira eficaz. Inicialmente, um sensor de fluxo de água coleta dados que são transmitidos para um ESP32, um dispositivo central. O ESP32, por sua vez, envia esses dados para o ThingSpeak, um serviço de nuvem que os armazena.

Para que os usuários acessem e visualizem os dados, um aplicativo móvel é desenvolvido no Thunkable. Este aplicativo se conecta ao ThingSpeak, coleta as informações e as armazena com segurança em uma base de dados hospedada no Airtable. Essa integração entre hardware, serviços em nuvem e aplicativo móvel possibilita o

monitoramento em tempo real do fluxo de água, com aplicações diversas, desde a agricultura até o gerenciamento de recursos hídricos em residências e ambientes comerciais.

**Fluxograma 1: Sistema de Monitoramento do fluxo de líquidos.**



Fonte: Arquivo dos autores.

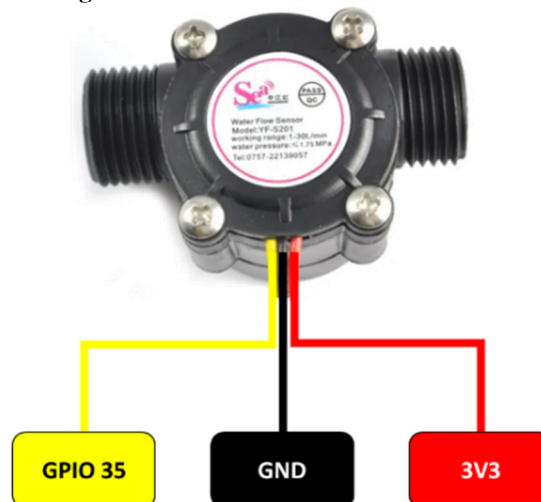
Logo, esse modelo de sistema possui relativa simplicidade na instalação e baixo custo para o usuário final, em comparação a outros processos, pois aplica o conceito Internet das coisas (IoT) e de sustentabilidade ambiental, algo em foco, visto que muitas empresas não executam completamente seus projetos devido alto custo e complexidade do sistema (TAVARES, 2019).

Neste momento, são apresentadas as tecnologias e dispositivos de software, descrevendo as 6 etapas para a fundamentação do trabalho e os cálculos desenvolvidas para a execução do projeto, sendo elas:

### 1. Conexão do sensor

Primeiramente, foi realizada a conexão do sensor de fluxo de água YF-S201 ao ESP32, o sensor é equipado com um conector de 3 pinos. Esses pinos incluem: Sinal, que é o pino pelo qual o sensor envia dados, GND sendo o terra e VCC, pino de alimentação, que estão conectados respectivamente aos pinos GPIO 35, GND e 3V3 do ESP32, assim como pode ser visualizado na Figura 1.

**Figura 1: Conexão do sensor ao ESP32.**



Fonte: Arquivo dos autores.

## 2. Criação do Canal no ThingSpeak

Antes de desenvolver o programa para o ESP32 e o aplicativo no Thunkable, foi necessário criar um canal no ThingSpeak para armazenar e visualizar os dados de fluxo de água através de gráficos ou mostradores digitais, além de servir como meio de comunicação entre o ESP32 e o aplicativo móvel a ser desenvolvido. Esta etapa envolveu:

### 2.1. Criação de um novo canal no ThingSpeak

A plataforma ThingSpeak foi acessada através do domínio “https://thingspeak.com” utilizando uma conta válida, assim sendo criado um canal no ThingSpeak, onde os dados de fluxo do líquido trabalhado seriam armazenados.

### 2.2. Aquisição da Chave de API

Obteve-se a chave de API gerada pelo ThingSpeak, necessária para a comunicação entre o ESP32 e o canal no ThingSpeak.

## 3. Desenvolvimento da Programação para o ESP32 no Arduino IDE

Com o canal ThingSpeak criado, a próxima fase do projeto envolveu o uso da plataforma Arduino IDE para desenvolvimento da programação a ser inserida no ESP32 (Figuras 2 e 3), que mede o fluxo do líquido e envia os dados para o canal ThingSpeak. O processo foi dividido em três etapas principais.

**Figura 2: Primeira parte do código principal.**

```
1 #include <WiFi.h>
2 #include <ThingSpeak.h>
3
4 const char* ssid = "Nome"; // Nome da sua rede Wi-Fi
5 const char* password = "Senha"; // Senha da sua rede Wi-Fi
6 const unsigned long channelID = ID; // ID do seu canal ThingSpeak
7 const char* apiKey = "Chave API"; // Sua chave API do ThingSpeak
8 const int sensorPin = 35;
9 volatile long pulse;
10 unsigned long lastTime;
11 float vazao;
12 WiFiClient client;
13
14 void setup() {
15     pinMode(sensorPin, INPUT);
16     Serial.begin(115200);
17     attachInterrupt(digitalPinToInterrupt(sensorPin), increase, RISING);
18     WiFi.begin(ssid, password);
19     while (WiFi.status() != WL_CONNECTED) {
20         delay(1000);
21         Serial.println("Conectando ao WiFi...");
22     }
23     Serial.println("Conectado ao WiFi.");
24     ThingSpeak.begin(client);
25 }
```

Fonte: Arquivo dos autores.

Figura 3: Segunda parte do código principal.

```
26 void loop() {
27     vazao = 2.663 * pulse / 1000 * 30;
28     if (millis() - lastTime > 2000) {
29         pulse = 0;
30         lastTime = millis();
31     }
32     Serial.print(vazao);
33     Serial.println(" L/min");
34
35     ThingSpeak.setField(1, vazao);
36     int response = ThingSpeak.writeFields(channelID, apiKey);
37     if (response == 200) {
38         Serial.println("Envio para ThingSpeak bem-sucedido.");
39     } else {
40         Serial.println("Erro no envio para ThingSpeak. Código de erro: " + String(response));
41     }
42
43     delay(2000);
44 }
45 ICACHE_RAM_ATTR void increase() {
46     pulse++;
47 }
```

Fonte: Arquivo dos autores.

A segunda etapa consiste em realizar as definições de constantes, logo as bibliotecas necessárias para o programa foram incluídas (Linhas 1 e 2). Wi-Fi.h é usada para a comunicação Wi-Fi, e ThingSpeak.h é usada para interagir com o serviço ThingSpeak; As informações de rede Wi-Fi (nome da rede e senha), foram definidas como constantes conforme as Linhas 4 e 5. É importante ressaltar que esses valores foram ajustados para corresponder à rede Wi-Fi utilizada; O ID do canal e a chave de API do ThingSpeak (Linhas 6 e 7), foram definidos como constantes. É importante ressaltar que esses valores foram ajustados para corresponder aos dados obtidos no subtópico 1.3; Nas linhas de 8 a 11, o acesso GPIO35 foi definido como o pino do sensor e ariáveis foram declaradas para contar pulsos, rastrear o tempo e armazenar o volume calculado; E por fim um objeto WiFiClient chamado client (Linha 12), foi criado para lidar com a conexão Wi-Fi.

A segunda etapa consiste em iniciar a função setup() (Linha 14), que foi executada apenas uma vez no início para definição de comandos de configuração do código que devem ser realizados apenas na inicialização do programa sem repetições; Definiu-se o pino do sensor como uma entrada (Linha 15), indicando seu uso para leitura dos pulsos do sensor; Estabeleceu-se a comunicação serial com uma taxa de 115200 bauds (Linha 16), permitindo a impressão de informações no monitor serial; Uma interrupção foi anexada ao pino do sensor para contar pulsos quando ocorre uma borda de subida (Linha 17); Nas linhas de 18 a 25 iniciou-se a conexão Wi-Fi com as credenciais fornecidas, aguardando até que o dispositivo esteja conectado à rede. Ao obter sucesso na conexão de rede, o dispositivo se conectou ao serviço ThingSpeak.

A terceira etapa consiste em iniciar a função loop() (Linha 26) onde os comandos acontecem repetidamente; O volume foi calculado com base no número de pulsos armazenado em pulse. A fórmula utilizada para volume pode variar de acordo com o líquido utilizado, no caso do projeto foi trabalhado o uso da água, e a calibragem do sensor para conseguir os valores da fórmula é mostrada no tópico 4, assim achando a equação conforme mostrado nas Linhas 27; Realizou-se uma verificação para confirmar se passaram mais de 2 segundos desde a última leitura (lastTime). Se isso for verdade, o contador pulse é redefinido para zero. Isso é feito para permitir o cálculo do volume em intervalos regulares, assim como exposto nas Linhas de 28 a 31; O valor da vazão calculado foi impresso em litros por minuto (L/min) no monitor serial para fins de exibição e depuração como exibido nos comandos da Figura 3; O valor do campo 1 no canal ThingSpeak foi definido com o valor do volume calculado (Linha 35); Os dados foram enviados para o canal ThingSpeak especificado usando a função writeFields(). O resultado foi armazenado na variável response (Linha 36); Por fim, verificou-se se o envio para o ThingSpeak foi bem-sucedido com base no código de resposta. Se o código de resposta for 200 (código HTTP "OK"), é exibida uma mensagem de sucesso. Caso contrário, é exibida uma mensagem de erro com o código de erro retornado. Foi aplicado um Delay, que introduz um atraso de 2 segundos antes de iniciar a próxima iteração do loop, para controlar a taxa de envio dos dados, conforme exibido nas linhas 37 a 45.

#### 4. Calibração do sensor de fluxo de água

Para verificação da funcionalidade do sensor YF-S201, foi necessário fazer a aquisição de um novo sensor (ICS006-DLS) já existente no mercado e calibrado, assim como mostrado na Figura 4, para isso precisou ser criado um dispositivo que permitisse que ambos os sensores fossem ligados a uma mesma alimentação de água, dispositivo esse que foi construído com os seguintes materiais: um kit de engate rápido de mangueira; uma luva PVC roscável 3/4", dois niple PVC roscável 3/4", duas luvas de redução PVC 3/4" x 1/2", um registro esfera PVC rosca 3/4" e uma garrafa PET de 2 (dois) litros, como pode ser observado no modelo real do Sensor (Figura 5).

**Figura 4: Dispositivo para testes.**



Fonte: Arquivo dos autores.

**Figura 5: Dispositivo para testes.**



Fonte: Arquivo dos autores.

Após a construção do dispositivo para testes, passou-se para a etapa calibração do sensor YF-S201 em relação ao sensor ICS006-DLS, ao conectar o dispositivo para testes em uma torneira residencial, foi possível fazer uma calibração do sensor através de 4 etapas, sendo elas:

##### 4.1. Contagem de pulsos do sensor

**Figura 6: Código para contagem de pulsos do sensor.**

```
int sensorPin = 35;
volatile long pulse;
void setup() {
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(sensorPin), increase, RISING);}
void loop() {
  Serial.print(pulse);
  delay(500);}
void increase() {
  pulse++;}
```

Fonte: Arquivo dos autores.



Foi aplicado um código recomendado pelo fabricante do sensor YF-S201 para achar a quantidade de pulsos que o sensor envia para cada litro do líquido trabalhado (Figura 6), no caso desse projeto, a água foi o líquido trabalhado, assim tendo como resultado o coeficiente multiplicador de pulsos de 2,663 para transformar pulsos em volume de água em litros.

#### 4.2. Cálculo do volume de água

**Figura 7: Código para cálculo do volume de água.**

```
int sensorPin = 35;
volatile long pulse;
float volume;
void setup() {
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(sensorPin), increase, RISING);}
void loop() {
  volume = 2.663 * pulse;
  Serial.println(volume);
  delay(500);}
void increase() {
  pulse++;}
```

**Fonte: Arquivo dos autores.**

Foi aplicado um código recomendado pelo fabricante do sensor YF-S201 para achar o volume de água que passa pelo sensor, utilizando o multiplicador de 2,663 de acordo com o valor achado no subtópico 4.1.

#### 4.3. Cálculo de conversão do volume para vazão

Levando em consideração o tempo de atualização de 2 segundos, foi criada a equação responsável por calcular a vazão de acordo com o volume de água passado pelo sensor durante determinado período. A equação considera o volume dividido por trinta mil milissegundos, assim como pode ser visto na Linha 27 do código principal.

#### 4.4. Comparação de vazão entre sensores

Após o cálculo para que o sensor YF-S201 consiga imprimir a vazão, foi realizada uma validação de dados ao comparar resultados com o sensor ICS006-DLS que já está operante no mercado. Esta comparação ocorreu ao conectar o dispositivo de testes a uma torneira residencial, sendo possível acompanhar em tempo real a vazão apontada por cada sensor, assim concluído a calibração do sensor YF-S201.

### 5. Desenvolvimento do Aplicativo no Thunkable

Para permitir que os usuários tenham um gerenciamento mais flexível que facilmente acessem e visualizem os dados de fluxo de água em tempo real, tendo possibilidade de filtrar esses dados de maneira rápida e agilizar processos, um aplicativo móvel foi desenvolvido no Thunkable. O processo foi desenvolvido em 5 principais etapas, sendo elas:

### 5.1. Configuração do Projeto no Thunkable

Criação de um novo projeto no Thunkable para o desenvolvimento do aplicativo, incluindo a definição e programação das interfaces de exibição e filtro de dados.

### 5.2. Telas do Aplicativo

Ao criar aplicativos em Thunkable, é necessário criar primeiro o design das telas, pois após a definição dos elementos visuais pode ser feita a programação para definição das funções de cada item. O aplicativo inclui cinco telas, sendo elas:

- Home (Figura 8): Tela inicial do aplicativo, a qual permite que os usuários acessem o aplicativo, tendo acesso aos dados apenas fazendo login com credenciais válidas;
- Cadastro (Figura 9): Permite que novos usuários se registrem no aplicativo;
- Sistema (Figura 10): Exibe as leituras em tempo real do sistema de fluxo de água medido pelo ESP32.
- Filtro de dados (Figura 11): Exibe o fluxo médio dentro do período escolhido.
- Dashboard (Figura 12): Exibe os dashboards interativos do Airtable.

#### 5.2.1. Design das telas

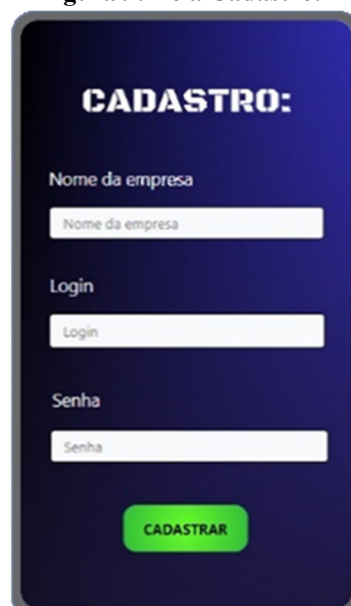
Foram desenvolvidos os designs das telas do aplicativo, adicionando todos os elementos visuais que o compõem.

**Figura 8: Tela Home.**



Fonte: Arquivo dos autores.

**Figura 9: Tela Cadastro.**



Fonte: Arquivo dos autores.

Figura 10: Tela Sistema.



Fonte: Arquivo dos autores.

Figura 11: Tela Filtro de dados.



Fonte: Arquivo dos autores.

Figura 12: Tela Dashboard.

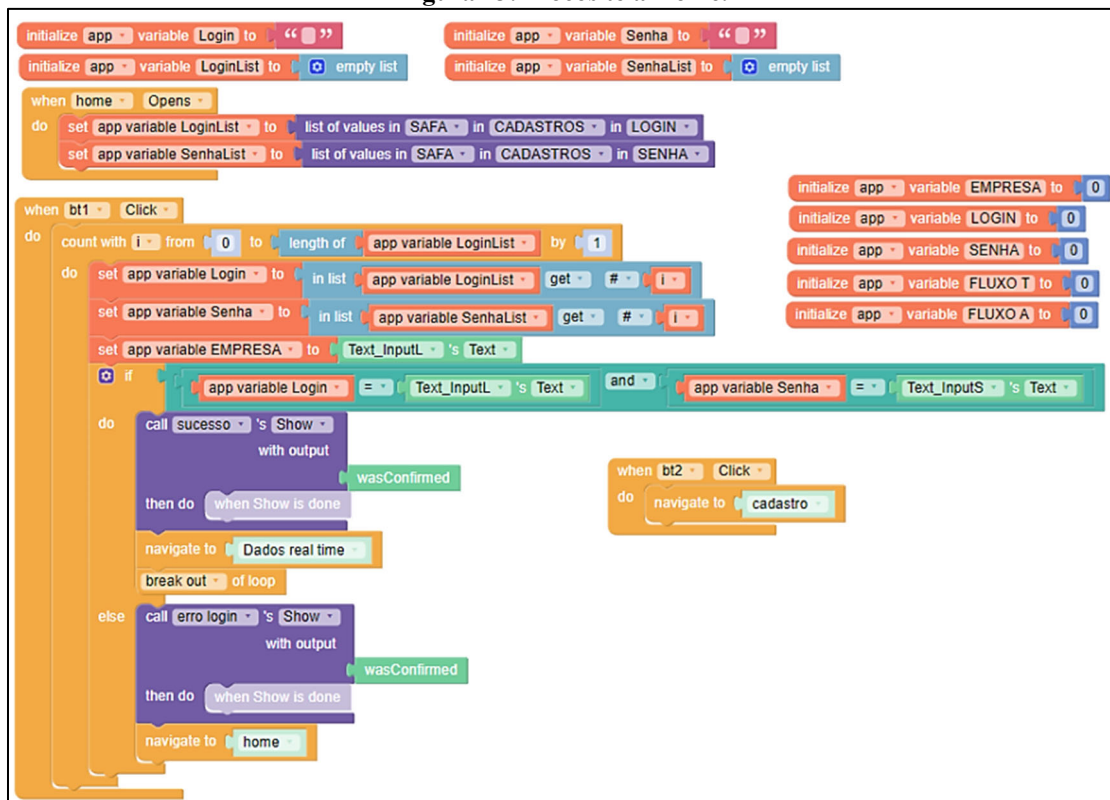


Fonte: Arquivo dos autores.

### 5.2.2. Blocos de programação das telas

Dentro do Thunkable, após montar o design das telas, deve ser feita a programação para que tudo funcione, nesse caso, por se tratar de uma plataforma low code, essa programação é realizada através de blocos.

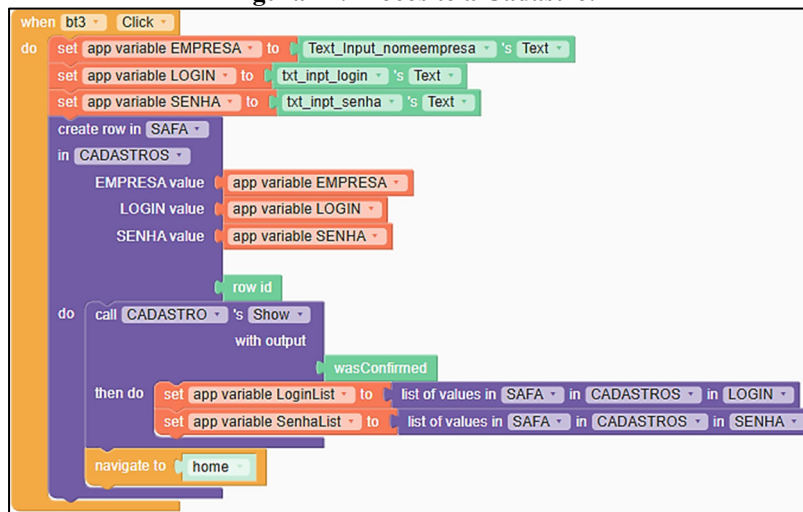
Figura 13: Blocos tela Home.



Fonte: Arquivo dos autores.

A tela Home recebe programação responsável por fazer a verificação das credenciais inseridas para login e permitir acesso apenas para usuário e senha válidos (Figura 13).

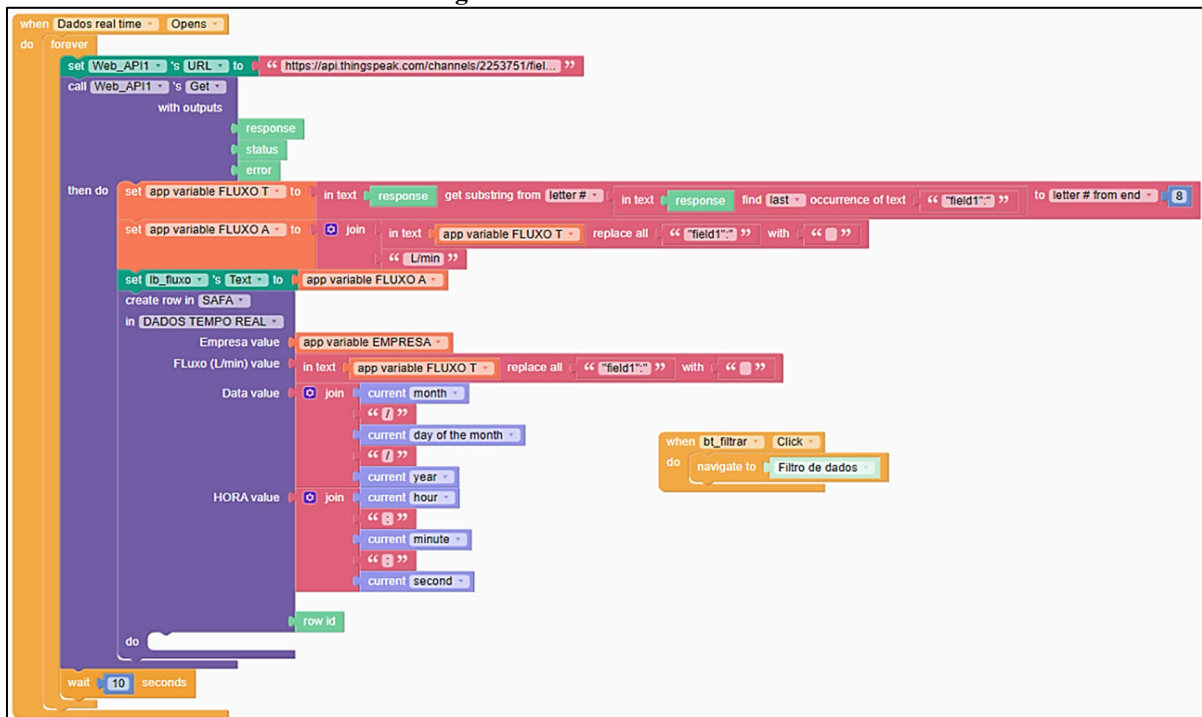
Figura 14: Blocos tela Cadastro.



Fonte: Arquivo dos autores.

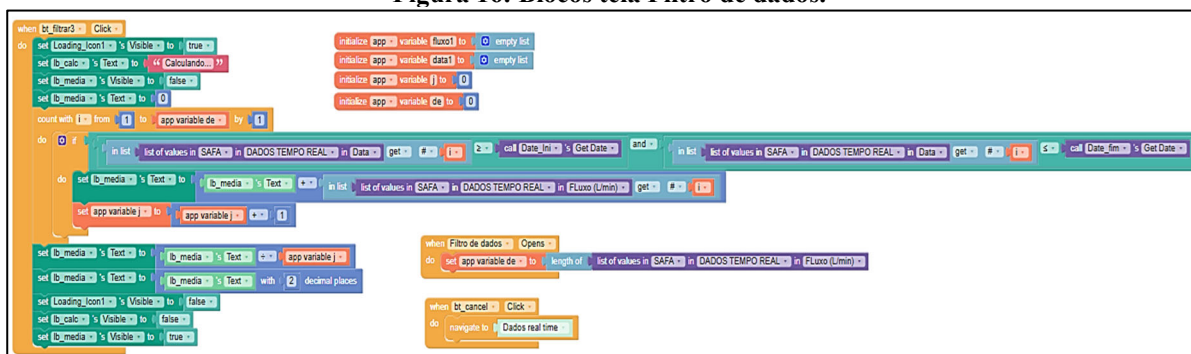
A tela Cadastro recebe programação responsável por receber os dados de nome da empresa, login e senha para armazená-los no banco de dados de cadastros (Figura 14).

Figura 15: Blocos tela Sistema.



Fonte: Arquivo dos autores.

A tela Sistema recebe programação responsável por capturar os dados do ThingSpeak e mostrá-los na tela em tempo real, além de armazená-los no banco de dados, incluindo os parâmetros de nome da empresa, fluxo em litros por minuto, data e hora de gravação dos dados (Figura 15).

**Figura 16: Blocos tela Filtro de dados.**


Fonte: Arquivo dos autores.

A tela Filtro de dados recebe programação responsável por analisar as datas inicial e final escolhidas, assim sendo possível imprimir o fluxo médio do período selecionado (Figura 16).

**Figura 17: Blocos tela Dashboard.**


Fonte: Arquivo dos autores.

A tela Dashboard recebe programação responsável por atualizar a visualização do web viewer que mostra os dashboards, além de configurar um botão como opção de retorno para a tela Sistema (Figura 17).

### 5.3. Integração com ThingSpeak

Realizou-se a configuração da integração do aplicativo com o ThingSpeak usando a API (Linha 7 do código principal), para recuperar os dados de fluxo de água do canal específico.

### 5.4. Desenvolvimento das Funcionalidades

Desenvolveram-se funcionalidades para fazer login (Figuras 8 e 13), cadastros (Figuras 9 e 14), e exibir os dados de fluxo de maneira clara e compreensível (Figuras 10 e 15) ou como forma de dashboards interativos (Figuras 12 e 17), permitindo que os usuários acessem as leituras mais recentes, assim como podem selecionar uma data de início e uma data final para filtrar e exibir uma média do fluxo do período selecionado (Figuras 11 e 16).

### 5.5. Testes e Depuração

**Figura 18: APP Thinkable Live.**


Fonte: Arquivo dos autores.

Foram realizados testes e depuração do aplicativo através do APP Thunkable Live (Figura 18), disponível na App Store pra dispositivos IOS e Google Play Store para Android, para garantir seu correto funcionamento, incluindo a autenticação de usuários e a exibição em tempo real das leituras do sensor.

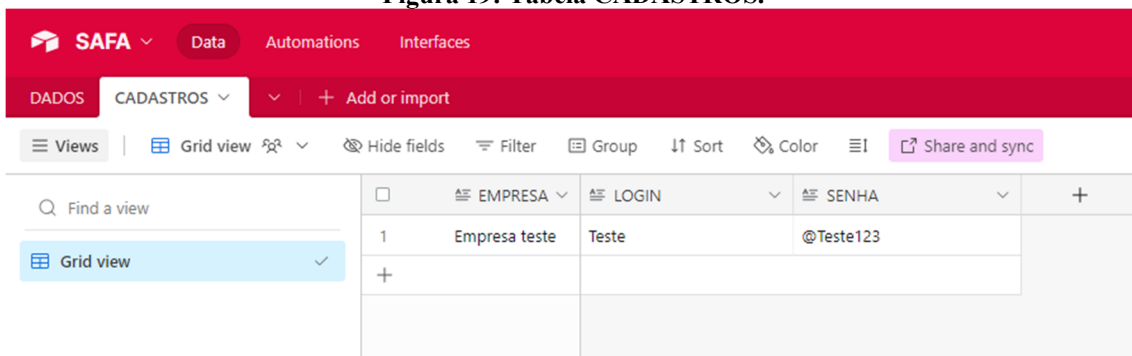
## 6. Criação de Um Banco de Dados no Airtable

Para o armazenamento de informações de login dos usuários e as leituras do sensor, foi utilizada uma base de dados no Airtable. O processo envolveu:

### 6.1. Configuração da Base de Dados

Criou-se uma base de dados no Airtable com tabelas específicas para armazenar informações de login dos usuários (Figura 19) e dados do sensor (Figura 20).

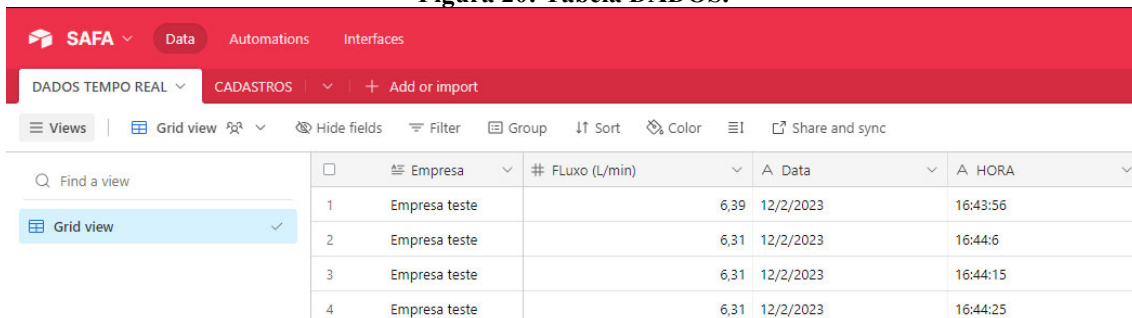
**Figura 19: Tabela CADASTROS.**



	EMPRESA	LOGIN	SENHA
1	Empresa teste	Teste	@Teste123

Fonte: Arquivo dos autores.

**Figura 20: Tabela DADOS.**



	Empresa	Fluxo (L/min)	Data	HORA
1	Empresa teste	6,39	12/2/2023	16:43:56
2	Empresa teste	6,31	12/2/2023	16:44:6
3	Empresa teste	6,31	12/2/2023	16:44:15
4	Empresa teste	6,31	12/2/2023	16:44:25

Fonte: Arquivo dos autores.

### 6.2. Integração com o Aplicativo

Foi realizada a integração do aplicativo desenvolvido no Thunkable com a base de dados do Airtable para permitir a gravação e recuperação de informações.

### 6.3. Criação de dashboards interativos

Para um gerenciamento de dados mais imersivo e melhor visualização, foram criados dashboards interativos, conforme mostrado nas Figuras 27, 28 e 29.

Em síntese, a criação do sistema consistiu-se no processo demonstrado no Fluxograma 1, um canal no ThingSpeak foi estabelecido para coletar dados de fluxo de fluidos em tempo real. O ESP32 foi programado para adquirir e enviar esses dados, e um aplicativo móvel Thunkable foi desenvolvido para fornecer acesso. Os dados foram armazenados no Airtable, permitindo um monitoramento eficiente do fluxo de fluidos em ambientes industriais através de dashboards interativos.

## RESULTADOS E DISCUSSÃO

Logo após a criação do canal, a programação no Arduino IDE foi desenvolvida e carregada em um ESP32, um componente versátil e poderoso. Essa etapa possibilitou a transmissão eficiente dos dados do sensor para o canal no ThingSpeak. A escolha estratégica do ESP32 aprimorou a comunicação, contribuindo para a coleta contínua e automática de informações cruciais.

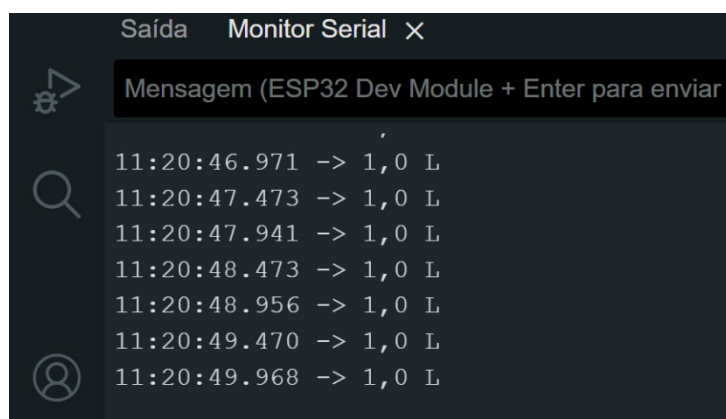
Em paralelo à programação no Arduino, ocorreu a validação do sensor YF-S201 em relação ao sensor ICS006-DLS. Essa comparação, crucial para garantir a confiabilidade dos dados, ocorreu durante o desenvolvimento do código em Arduino e antes do avanço para o próximo estágio, o desenvolvimento do aplicativo no Thunkable. Essa validação foi necessária devido à particularidade do sensor YF-S201, que demanda uma calibração específica através de cálculos incorporados à programação Arduino carregada no ESP32. Em contraste, o sensor ICS006-DLS já estava programado e calibrado de acordo com as configurações de mercado, como visualizado nas imagens abaixo (Figuras 21 e 22), que demonstram valor de volume e água durante testagem, assim sendo possível definir a constante necessária para os cálculos de calibração do sensor.

**Figura 21: Volume de água visualizado no Sensor ICS006DLS.**



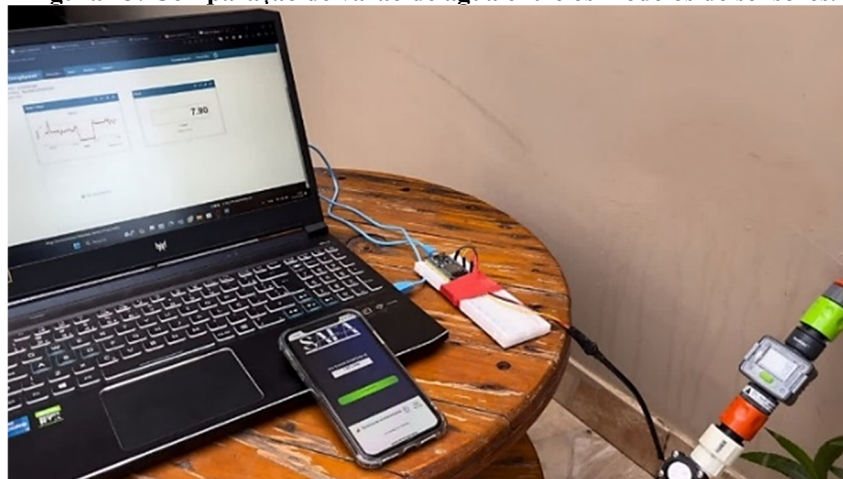
Fonte: Arquivo dos autores.

**Figura 22: Volume de água visualizado no Sensor YF-S201.**



Fonte: Arquivo dos autores.

**Figura 23: Comparação de vazão de água entre os modelos de sensores.**



Fonte: Arquivo dos autores.

**Figura 24: Vazão de água no ThingSpeak.**



Fonte: Arquivo dos autores.

O ThingSpeak além de realizar a ponte de conexão entre o ESP32 e o aplicativo móvel, oferece uma visualização em tempo real dos valores de vazão, como pode ser visto na Figura 24, a plataforma é equipada com gráficos e mostradores que permitem a visualização da vazão de água em tempo real.

**Figura 25: Vazão de água no APP móvel.**



Fonte: Arquivo dos autores.

**Figura 26: Vazão de água no sensor ICS006-DLS.**



Fonte: Arquivo dos autores.

Foi obtido sucesso na validação da vazão da água com o sensor ICS006-DLS, como evidenciado nas Figuras 23, 24, 25 e 26, foi validada a atualização de dados do aplicativo, sendo obtido no teste o resultado de vazão de água igual a 7,90 L/min entre as três comparações. A validação do sensor YF-S201 em comparação com o sensor ICS006-DLS foi uma etapa crucial, garantindo a confiabilidade dos dados coletados. Essa validação não apenas assegura a precisão das medições, mas também destaca nosso compromisso com a qualidade e a confiabilidade dos dados gerados pelo sistema.



O aplicativo móvel desenvolvido nesta plataforma, foi criado e programado para receber os dados do ThingSpeak e oferecer praticidade e mobilidade ao acompanhamento de dados, o que proporcionou uma interface intuitiva e acessível para visualização dos dados, assim como pode ser visto na Figura 25. Essa escolha veio como um passo subsequente, tornando a experiência do usuário mais amigável e ampliando o alcance do monitoramento para diversos usuários.

A mobilidade proporcionada pelo aplicativo móvel Thinkable é crucial para uma gestão dinâmica. Permite o monitoramento em qualquer lugar e a qualquer momento, agilizando a tomada de decisões e contribuindo para uma gestão mais eficiente em resposta às condições em constante mudança.

Em adição, a integração dos dados coletados com o Airtable possibilitou a criação de dashboards interativos personalizados (Figuras 27, 28 e 29), fornecendo uma visualização intuitiva dos dados. Esses dashboards não apenas simplificam a análise dos dados do sensor, mas também oferecem uma plataforma interativa para extrair insights significativos.

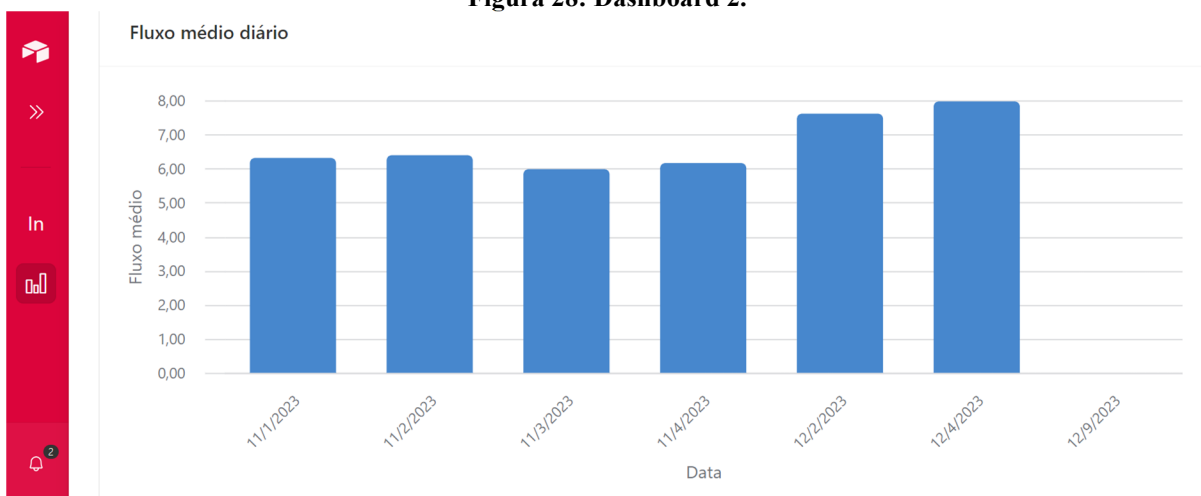
Consoante a isso, os dashboards, apresentam de forma clara os valores do fluxo, sendo possível visualizar nas Figuras 27 e 28: a média, que serve com um indicador de “meio termo” da quantidade de fluidos que passam pelo sensor, além de auxiliar a encontrar outros dados, como o desvio padrão; desvio padrão, que serve para medir a dispersão dos dados, identificando quando um dado se afasta da média; valor mínimo e valor máximo de vazão de água, que é útil para descobrir possíveis irregularidades na tubulação, durante determinado período, demonstrando um gráfico em colunas do fluxo médio diário, sendo organizado pelos Destaques da Semana (Figura 29) que evidencia dados como: Nome da empresa, Fluxo de água, Data e Hora de criação dos dados, fator esse extremamente necessário para identificação de prováveis erros e manutenção do serviço ofertado.

Figura 27: Dashboard 1.



Fonte: Arquivo dos autores.

Figura 28: Dashboard 2.



Fonte: Arquivo dos autores.

Figura 29: Dashboard 3.



Fonte: Arquivo dos autores.

A abordagem metodológica adotada, desde a criação do canal no ThingSpeak até a validação do sensor, programação no Arduino IDE, desenvolvimento do aplicativo no Thunkable e integração com o Airtable, culminou em um sistema coeso e confiável para o monitoramento de fluxo de água. Este projeto não apenas atendeu, mas superou as expectativas, pavimentando o caminho para futuras inovações na área de sensoriamento e gestão de dados hídricos.

## CONCLUSÃO

Este projeto representa um avanço significativo na gestão do fluxo de fluidos, pois demonstra que um aplicativo móvel com tamanha complexidade e grau de importância, pode ser produzido de maneira mais simples e intuitiva, principalmente em ambientes industriais, onde a eficiência e o controle desempenham papéis críticos, pois através do estudo desenvolvido, determinou-se que a implementação de um sistema de monitoramento de fluxo de líquidos na indústria civil é de extrema importância por vários motivos. Esse tipo de sistema é projetado para controlar e supervisionar o movimento de líquidos em tubulações e processos industriais. Aqui estão algumas razões que destacam a importância desse sistema: Controle de Qualidade; Manutenção Preditiva (manutenção baseada na condição); Detecção de Vazamentos; Economia de Recursos e Diminuição do Impacto Ambiental (uma das virtudes da Indústria 4.0). Desde a criação do canal no ThingSpeak até a implementação prática da aplicação móvel no Thunkable, cada etapa foi meticulosamente projetada para atender ao nosso objetivo principal: otimizar a gestão do fluxo de fluidos em tempo real.

Ao atingir este objetivo, concluiu-se que os benefícios substanciais que esta aplicação pode oferecer à indústria. A capacidade de monitorar o fluxo de fluidos de maneira móvel e eficiente não apenas aprimora a eficiência operacional, mas também tem implicações positivas na redução de custos e na minimização do impacto ambiental. A mobilidade proporcionada pela aplicação móvel Thunkable oferece a flexibilidade necessária para um monitoramento dinâmico, facilitando tomadas de decisão rápidas e intervenções oportunas.

A integração bem-sucedida dos dados com o Airtable permitiu a criação de dashboards interativos personalizados, proporcionando uma visualização intuitiva e simplificada dos dados. Esses dashboards não apenas simplificam a análise, mas também oferecem uma plataforma interativa para extrair insights significativos. Essa abordagem não apenas atende às necessidades operacionais, mas também fornece uma base sólida para a análise contínua e aprimoramento do sistema.

Este estudo não é apenas um teste de viabilidade; é uma complementação que destaca a importância prática da tecnologia na otimização de processos industriais. À medida que as indústrias buscam continuamente melhorar eficiência e sustentabilidade, soluções como a aplicação desenvolvida tornam-se instrumentais para alcançar operações eficientes e responsáveis. Este projeto é um primeiro passo significativo para um gerenciamento mais preciso e eficiente do fluxo de fluidos na indústria. Admite-se que, à medida que novas tecnologias emergem, é imperativo nos mantermos atualizados e adaptados a mudanças constantes. Esperamos que este estudo não apenas forneça uma solução eficaz para a gestão do fluxo de fluidos, mas também inspire pesquisas futuras e aplicações práticas que continuem a construir uma indústria mais eficiente e sustentável.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. Atzori, Luigi; IERA, Antonio; MORABITO, Giacomo. **The Internet of Things: a survey** Computer Networks, 2010.
2. ALIGER. A Importância da comunicação sem fio e sua relação com IoT. [S. l.], 2019. Disponível em: <https://www.aliger.com.br/blog/importancia-da-comunicacao-sem-fio-e-sua-relacao-com-iot>.
3. AMERICO, G.M; IZIDORO, C.L. Sistema de baixo custo para monitoramento do consumo de Água. **Revista Vincci-Periódico Científico do UniSATC**, v. 7, n. 1, p. 292-317, 2022.
4. BAJWA, J. K.; KAUR, J. Comparative study of apm tools. **International Journal of Engineering Science**, v. 24, p. 26–35, 2017.
5. DIRK, K.; MADDOX, J. Arquivos e Airtable: Usando ferramentas baseadas em nuvem para pesquisa de arquivamento e gerenciamento de fluxo de trabalho. 2018.
6. DORNELAS, E.; OLIVEIRA, C. S. Monitoramento de consumo doméstico de Água utilizando uma meta-plataforma de iot. **Revista de Engenharia e Pesquisa Aplicada**, v. 2, n. 2, 2017.
7. FEW, Stephen. Information dashboard design: The effective visual communication of data. **O'Reilly Media, Inc.**, 2006.
8. GALEGALE, Gustavo Perri et al. Internet das Coisas aplicada a negócios-um estudo bibliométrico. **JISTEM-Journal of Information Systems and Technology Management**, v. 13, p. 423-438, 2016.
9. IBRAHIM, Dragan. The Complete ESP32 Projects Guide. 1a. ed. [S.l.]: **Elektor Digital**, 2017.
10. KOLBAN, Neil. Kolban's book on ESP32. [S.l.]: **Leanpub**, 2018.
11. MATHWORKS. MathWorks. 2018. Disponível em: <<https://www.mathworks.com/help/thingspeak/>> Acesso em: 17 out. 2023.
12. LIU, S. *et al.* Rumo a uma melhor análise de modelos de aprendizado de máquina: uma perspectiva de análise visual. **Informática Visual**, v. 1, n. 1, pág. 48-56, 2017.
13. MICROSOFT. Compartilhe os painéis e os relatórios do Power BI com colegas e outras pessoas. 2019d. Disponível em: <https://docs.microsoft.com/pt-br/power-bi/collaborate-share/service-share-dashboards>> Acesso em: 16 out. 2023.
14. MUNDIM, K. C. O Efeito Hall. 1999. Disponível em: <<http://ensinoadistancia.pro.br/EaD/Eletromagnetismo/EfeitoHall/EfeitoHall.html>>.
15. OLIVEIRA, G. P. N. Os Diferentes Tipos De Desenvolvimento Mobile, Uma Visão Prática Do Multiplataforma. **Monografia**. UNICESUMAR – Centro Universitário de Maringá. Maringá (PR). 2019.
16. OLIVEIRA, H. R. I.; SANTOS, R. B. C.; RODRIGUES, L. A. M. Desenvolvimento de um aplicativo android para monitoramento microcontrolado do nível de um reservatório de Água residencial em tempo real. **Conferência de Estudo em Engenharia Elétrica**, 2014.
17. PROPEQ (BRASIL). Conheça os medidores de vazão e sua importância no controle de processos!. [S. l.], 2019.
18. PRNEWSWIRE. Após levantar US\$ 3,3 milhões de financiamento, a Thinkable lança o “Thinkable X”, um criador de aplicativos móveis do tipo faça você mesmo para não programadores. **Exame**, 05 jun. 2018.
19. SANTOS, R.D.C. Power BI: a experiência de implantação em um escritório de contabilidade. 2018.
20. SANTOS, E. A. B.; LIMA, O. L. T. Sistema Supervisório de Medição e Controle do Nível de Água em Reservatórios Residenciais. **Anais da Mostra de Extensão, Inovação e Pesquisa**, v. 4, 2017.
21. SARIKAYA, Alper et al. Do que falamos quando falamos de dashboards? **Transações IEEE sobre visualização e computação gráfica**, v. 1, pág. 682-692, 2018.
22. SHNEIDERMAN, Ben; PLAISANT, Catarina. Estratégias para avaliar ferramentas de visualização de informações: estudos de caso multidimensionais e aprofundados de longo prazo. In: **Anais do workshop AVI de 2006 sobre Além do tempo e erros: novos métodos de avaliação para visualização de informações**. 2006. pág. 1-7.
23. TAVARES, Lucas. O que é um Servidor Web (Web Server). [S. l.], 2019.
24. THATCHER, J. et al. Web Accessibility: Web Standards and Regulatory Compliance. New Work: Friends of – Springer-Verlag. 2006.
25. ZANUZZO, Lucas Tavares. Sistema de monitoramento do consumo de água controlado por um aplicativo Android. 2017. Trabalho de conclusão de curso (Bacharelado em Engenharia da Computação) - **Escola de Engenharia de São Carlos da Universidade de São Paulo**, [S. l.], 2017.
26. WANG, Hui et al. Algoritmo Firefly com atração de vizinhança. **Ciências da Informação**, v. 382, p. 374-387, 2017.
27. WINN, Z. Democratizing app development. MIT News Office. 2019. Disponível em: <http://news.mit.edu/2019/thinkable-app-inventor-0116>. Acesso em: 16 out. 2023.