



**INSTITUTO FEDERAL**

São Paulo  
Campus Catanduva

# Comando e Monitoramento de Unidades de Escoamento ou Alimentação de Contêineres

## **Alunos**

---

Daniel Gonçalves CT3016943

Murilo dos Reis Tavares CT3016986

## **Orientador**

---

Prof. Dr. Marcos Rodrigues Costa

## **Coorientador**

---

Prof. Me. Edivaldo Pastori Valentini

# Introdução

IoT: um paradigma tecnológico que forma uma rede de máquinas e dispositivos capazes de interagirem entre si de forma automática e sem nenhuma ou praticamente nenhuma interação humana. (LI; QIU, 2006) (LEE; LEE, 2015)

Este trabalho propõe a criação de uma rede de unidades que possam mensurar e monitorar níveis de fluídos em contêineres, com capacidade de comandar o escoamento ou alimentação destes fluídos, de forma automática, com ou sem interação humana.

Linhas de guia:

- Sistemas independentes
- Responsabilidades limitadas bem definidas <sup>1</sup>
- Comunicação remota entre dispositivos
- Automação e monitoramento
- Infraestrutura distribuída, potencialmente escalável

<sup>1</sup> Premissas de alta coesão e baixo acoplamento em sistemas de software; Couple and cohesion metrics; Stevens; Meyers; Constantine; Structured Design, IBM Systems Journal, 1974



# Justificativa e Objetivos

- Experimentar um projeto com base em uma demanda comum da indústria; (GONÇALVES, 2019)
- Utilizar componentes comuns e amplamente disponíveis, e integrá-los a uma infraestrutura moderna e potencialmente escalável, que possibilite às unidades independentes cumprirem as funções para as quais foram projetadas.

# Trabalhos Correlatos

**Taíra e Siqueira, 2018**

**Departamento de Engenharia Química e Petróleo  
Universidade Federal Fluminense - RJ**



O projeto é desenvolvido de modo que os componentes (sensores) responsáveis pela medição, os componentes responsáveis pela movimentação do fluido (atuadores) e os módulos de controle, estão todos ligados em um sistema monolítico, capaz de fazer a movimentação do fluido de um tanque inferior para um tanque superior em um aparato onde todos os componentes, incluindo os tanques, sejam parte uma única peça.



Foto: Taíra e Siqueira

# Trabalhos Correlatos

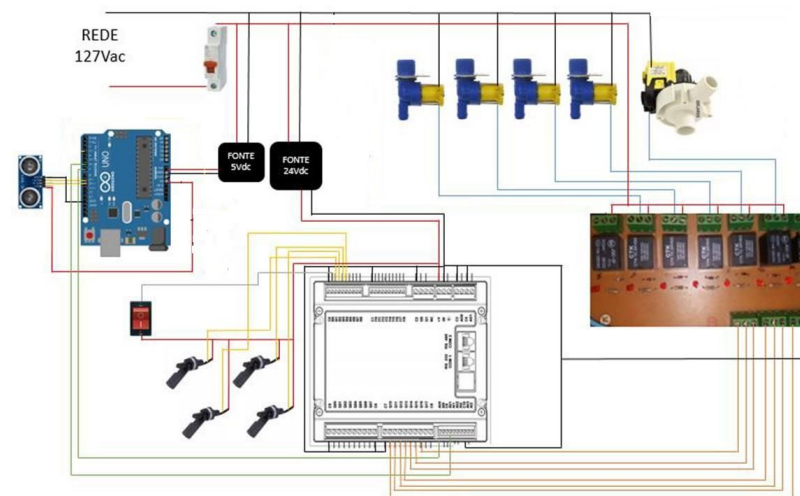
**Matheus Gonçalves, 2019**  
**Engenharia de Controle e Automação**  
**Universidade Federal de Ouro Preto - MG**



**UFOP**

Universidade Federal  
de Ouro Preto

O projeto emprega o uso de um CLP didático em oposição à plataformas mais acessíveis como Arduino ou ESP32. O propósito do projeto é controlar os níveis de fluídos entre três tanques, sendo que dois deles servem como reservatório de fluído para o tanque principal, que é o tanque monitorado medindo-se o nível através de um sensor ultrassônico HC-SR04. Os tanques reservatórios são mantidos dentro do mínimo e máximo através de sensores do tipo chave-bóia.



Autor: Matheus Gonçalves

# Trabalhos Correlatos

Thalys Gadelha, 2020

**Centro de Engenharia Elétrica e Informática**  
**Universidade Federal de Campina Grande - PB**



Universidade Federal  
de Campina Grande

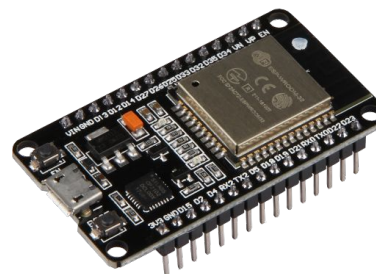
Trata-se de um projeto voltado para a manutenção de vasilhames para alimentação de animais domésticos. Utiliza ESP32 e sensores para determinar se o nível do conteúdo medido está abaixo de um nível considerado crítico, para então utilizar uma rede GSM para enviar notificações de SMS, além de um sinal sonoro (buzzer). Neste trabalho não há implementação de alimentação automática do conteúdo.



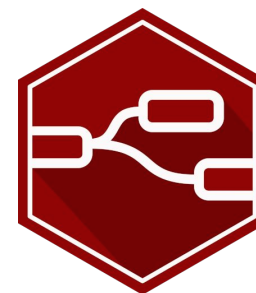
Foto: Thalys Gadelha

# Fundamentação Teórica

- Placa para prototipação com ESP32 (Node MCU), WiFi e Bluetooth LE integrados
  - ESP32 WROOM 32 DevKit V1
  - WiFi



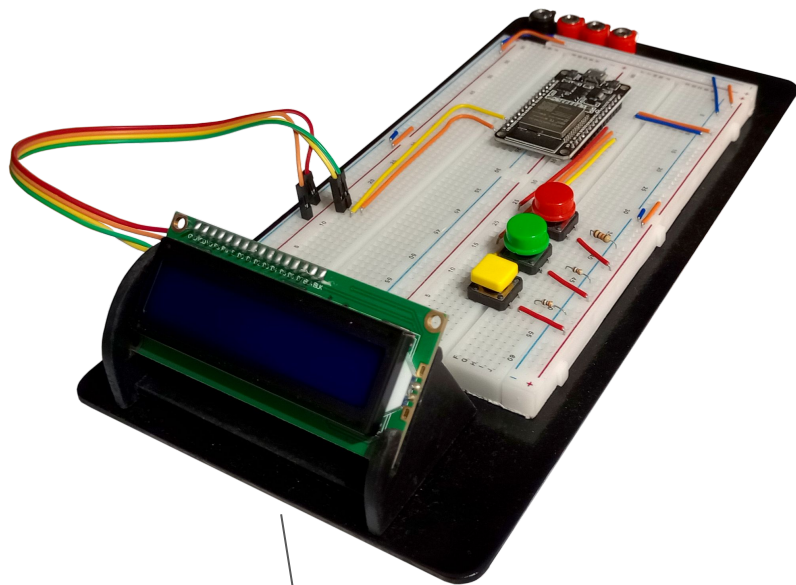
- Comunicação, integração e rastreabilidade
  - MQTT Eclipse Mosquitto
  - RDBMS PostgreSQL
  - Node-RED



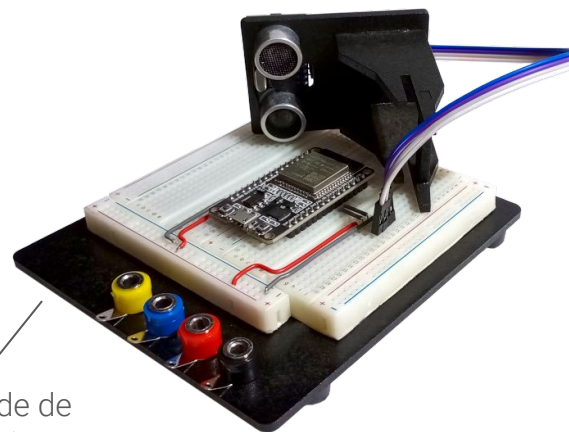
Imagens: Internet; As logomarcas são propriedades dos seus respectivos proprietários

# Desenvolvimento | Hardware

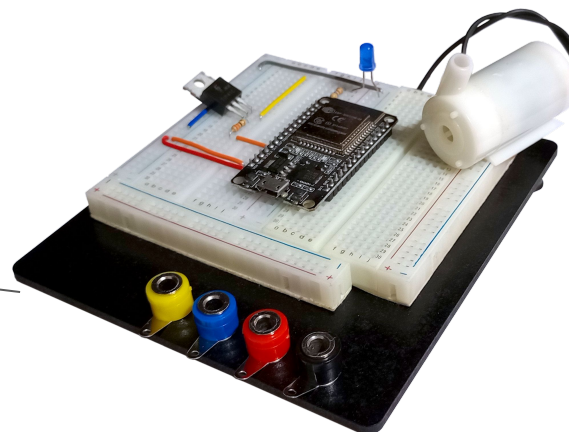
- Protótipo construído usando componentes simples e acessíveis, com protoboards montadas em placas de MDF 3mm cortadas em CNC laser, fornece uma base onde as unidades independentes foram desenvolvidas



Unidade de  
Monitoramento e  
Comando



Unidade de  
Medição



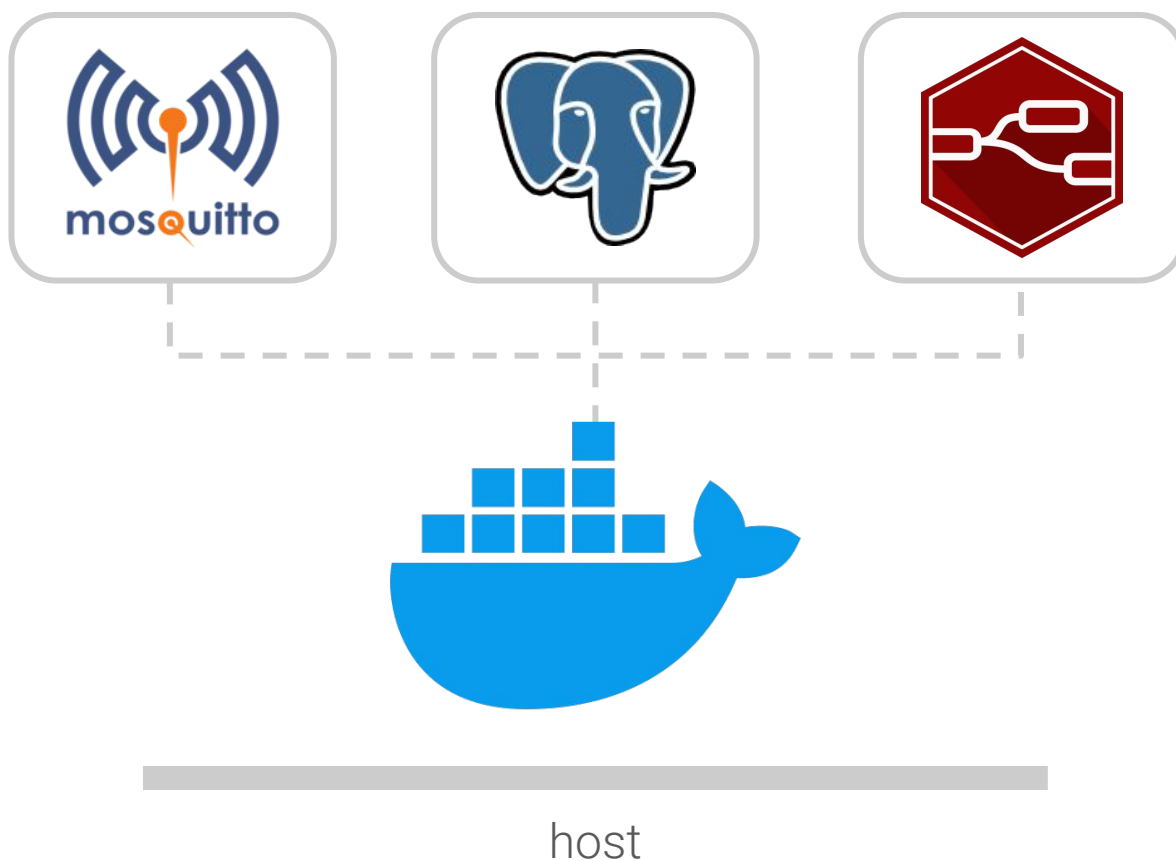
Unidade de  
Escoamento ou  
Alimentação

Fotos: Próprios autores



# Desenvolvimento | Software

- Infraestrutura baseada em virtualização para isolamento dos serviços e da aplicação Node-RED



As logomarcas são propriedades dos seus respectivos proprietários; Composição da Ilustração: Próprios autores

# Desenvolvimento | Visão Geral

- Uma foto das unidades posicionadas próximas. Embora o objetivo seja de que as unidades possam estar a qualquer distância, umas das outras, visualizar as unidades assim, dá uma ideia melhor de como funcionam.

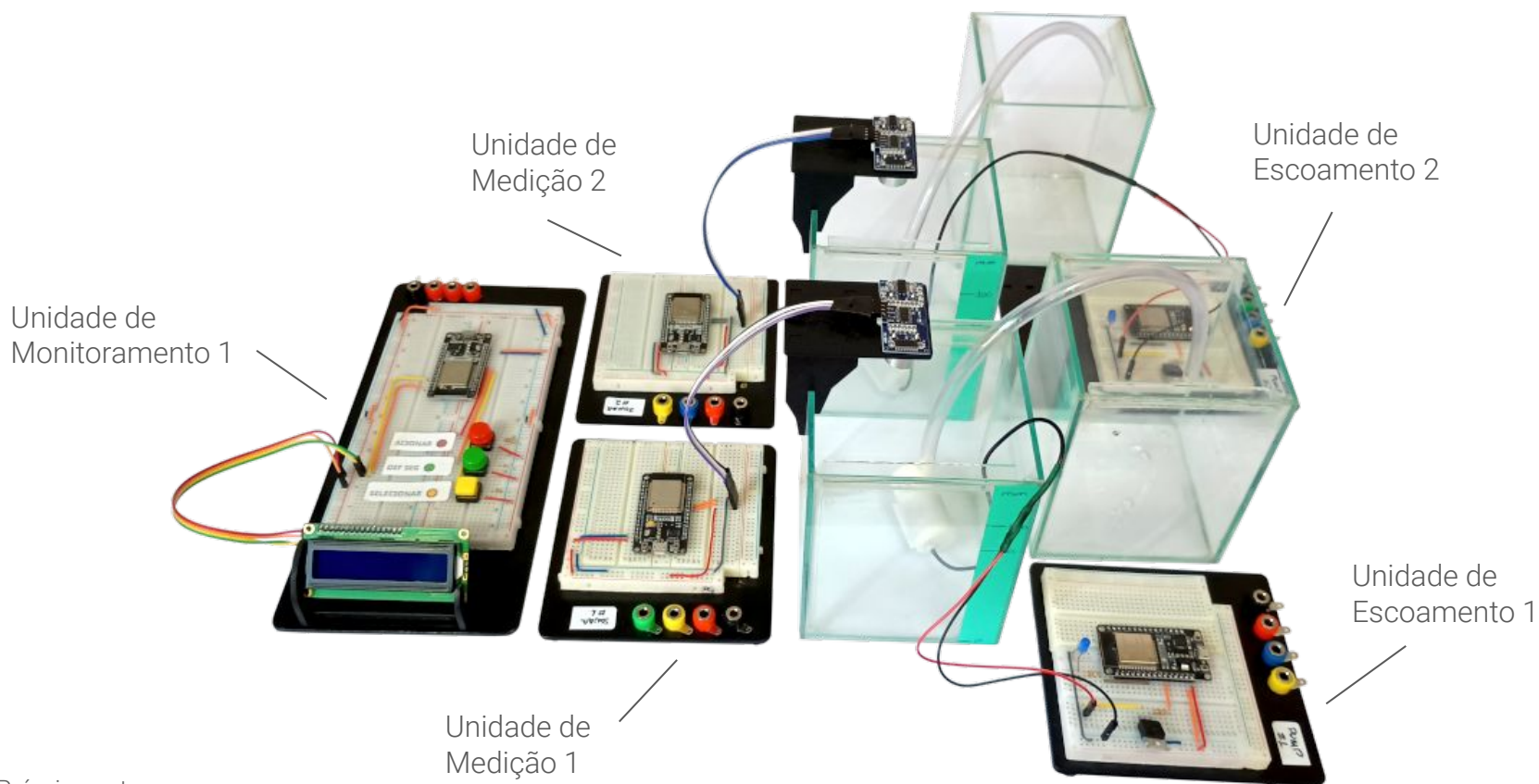


Foto: Próprios autores

# Desenvolvimento | Visão Geral - Contêineres

- Visão das unidades de escoamento e dos contêineres. Este protótipo conta com dois jogos de unidades de medição e escoamento com 1 par de contêineres cada.

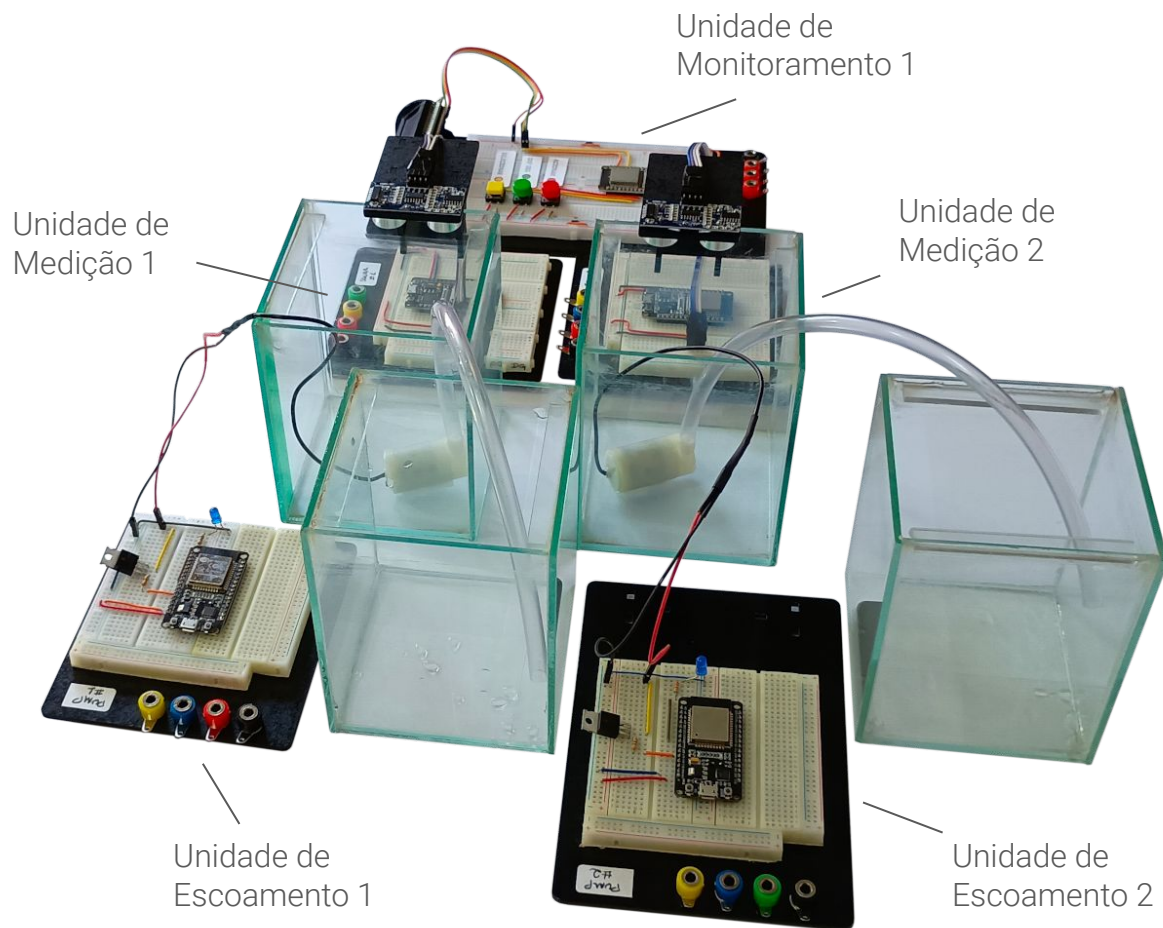
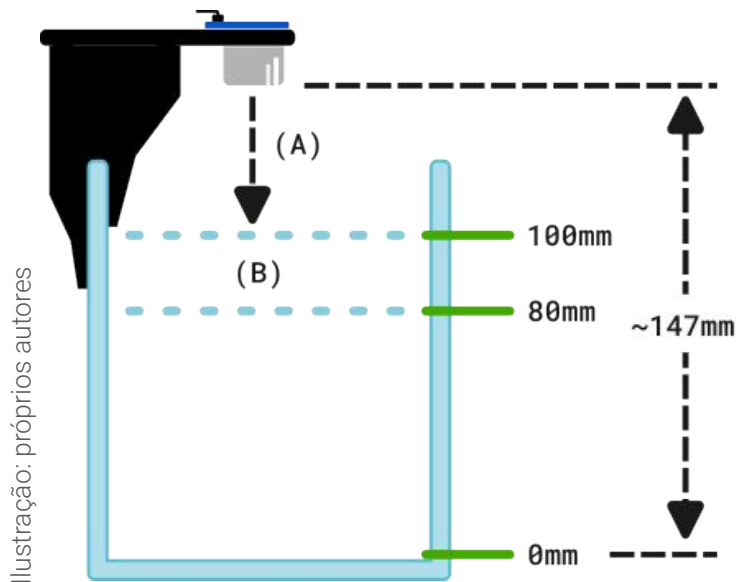


Foto: Próprios autores

# Desenvolvimento | Visão Geral - Parâmetros

- Comportamento esperado das unidades de medição e escoamento neste protótipo. Embora o objetivo não seja a precisão da medição ou escoamento, a parametrização no código-fonte deve considerar as limitações do protótipo



(A)  
 $\sim 278\text{ms} = \sim 4\text{cm}$

(B)  
Container 94x94mm  
1mm água = 8,85ml  
20mm água = 177ml

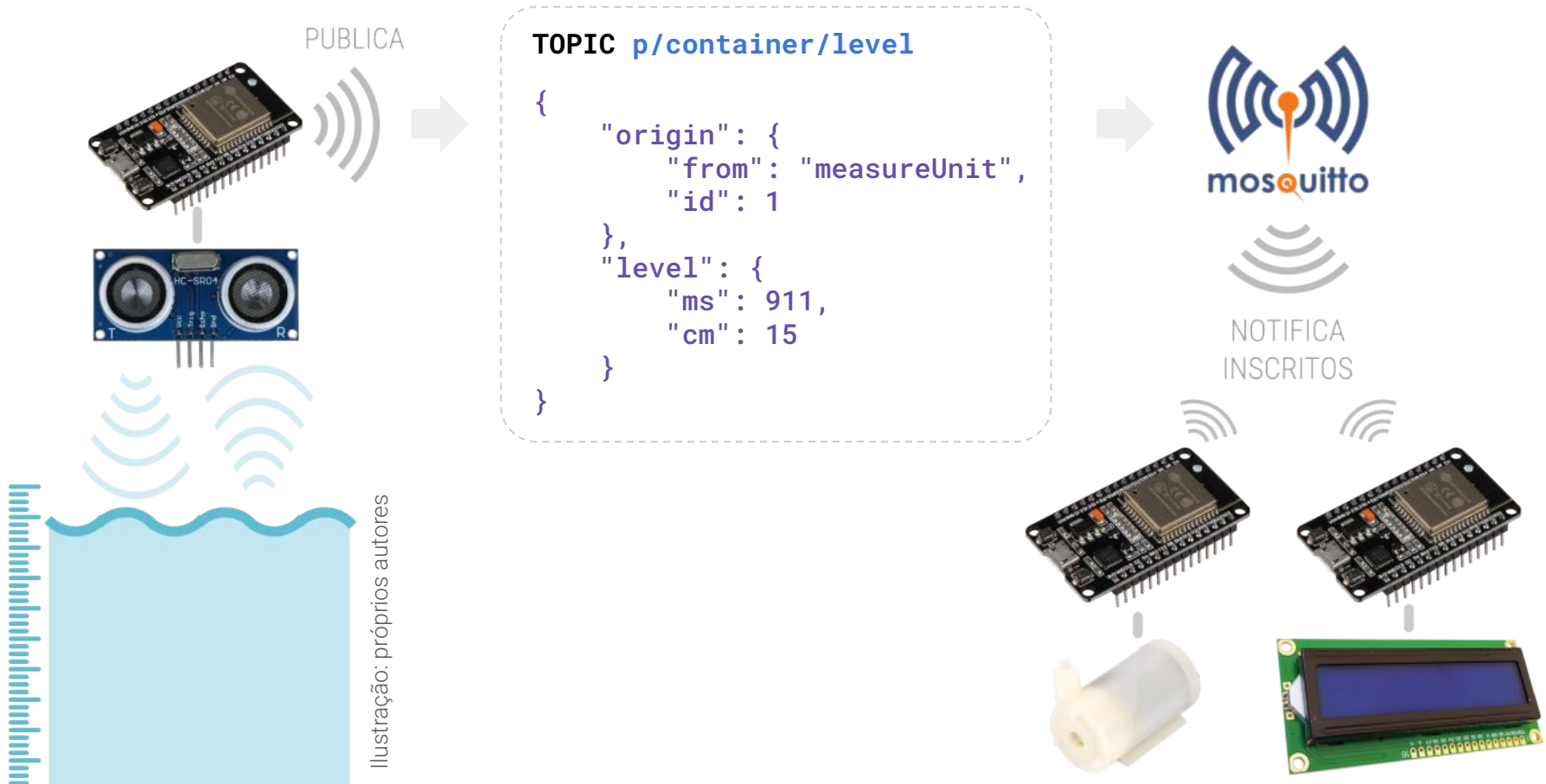
Bomba movimenta 29,6 ml/s com 4V~5V

$177\text{ml} / 29,6 \text{ ml/s} = 5,97\text{s} = \sim 6\text{s}$

Aciona a bomba por ~6s para escoar o equivalente a 177ml água

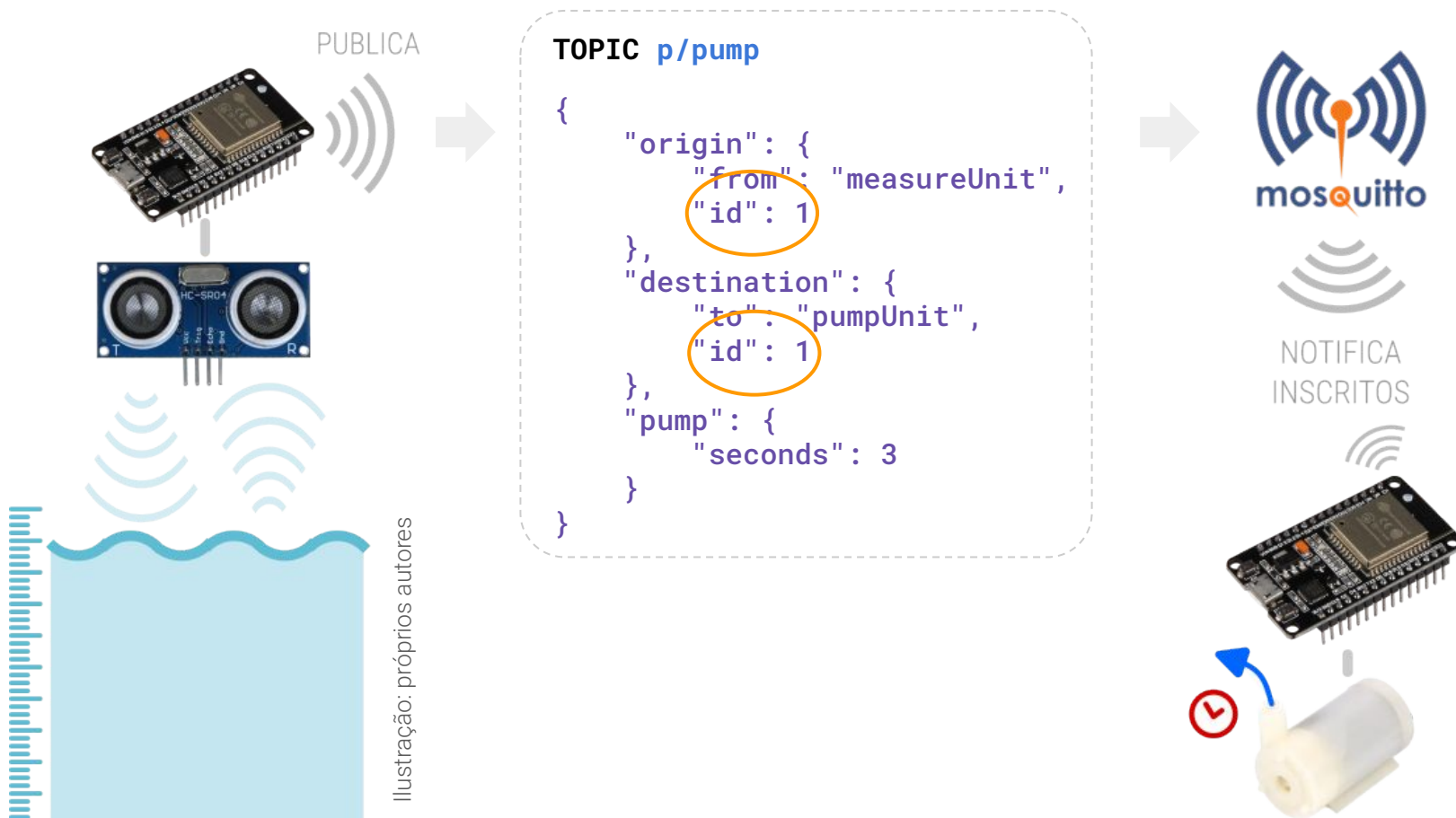
# Desenvolvimento | Publicação da Medição

- Interação entre as unidades de medição, monitoramento e escoamento, via *publish/subscribe* através de um *broker* MQTT Eclipse Mosquitto



# Desenvolvimento | Comando de Escoamento

- Acionamento do escoamento é feito pela unidade de medição, que detém o parâmetro para comandar o acionamento da bomba de mesmo identificador



# Desenvolvimento | Comando Manual

- É possível comandar o escoamento de qualquer container por um período de tempo arbitrário, usando a unidade de monitoramento

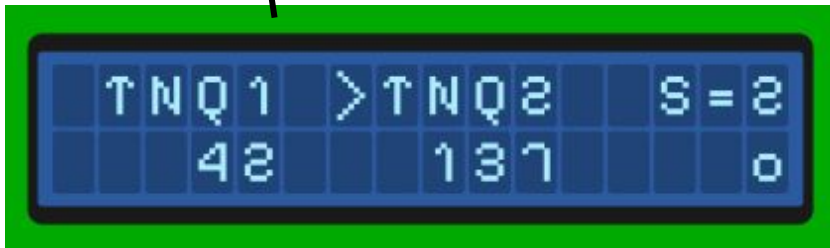
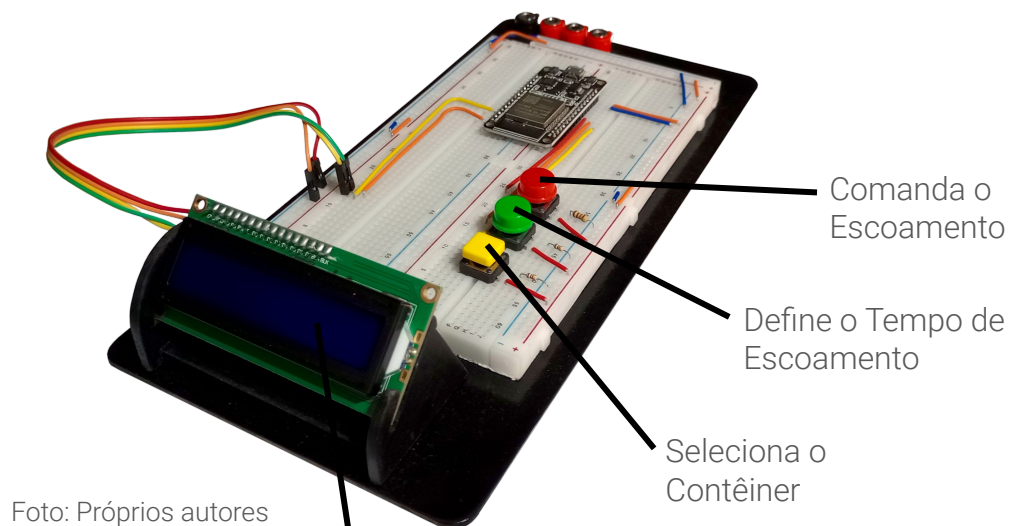


Ilustração: Próprios autores

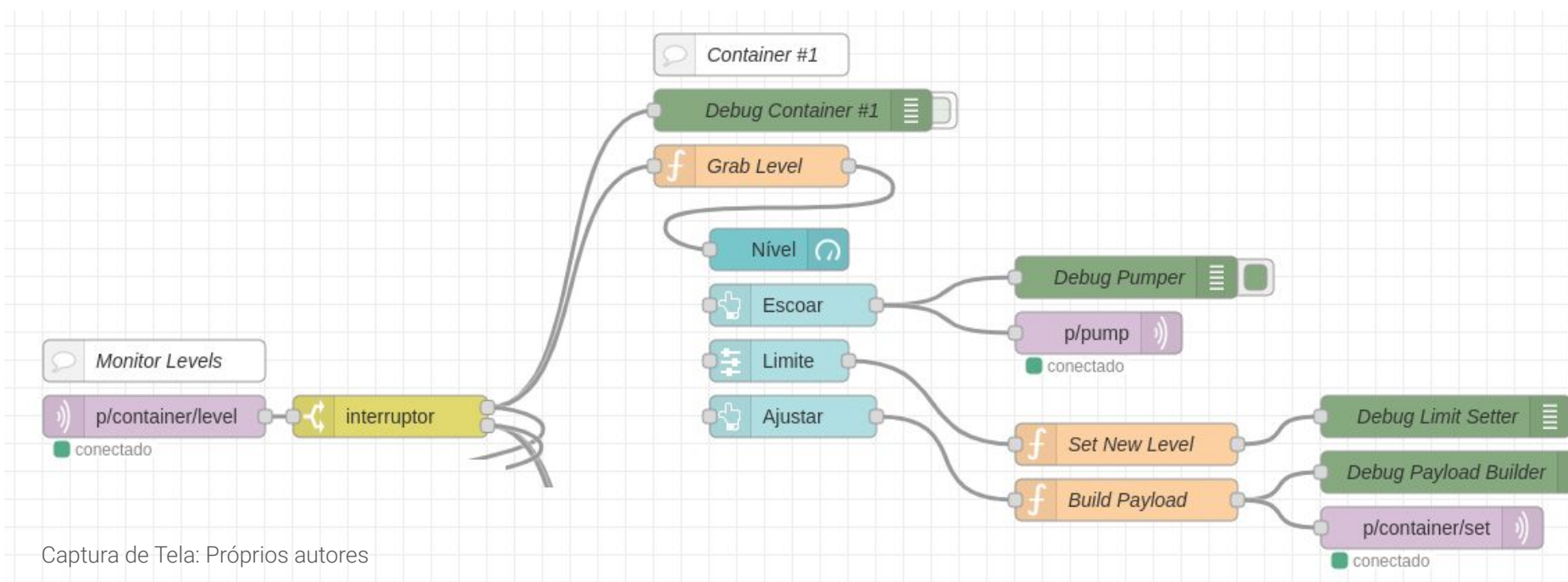
## TOPIC p/pump

```
{  
  "origin": {  
    "from": "monitorUnit",  
    "id": 1  
  },  
  "destination": {  
    "to": "pumpUnit",  
    "id": 2  
  },  
  "pump": {  
    "seconds": 2  
  }  
}
```



# Desenvolvimento | Integração via Low-Code

- Através da plataforma Node-RED foi possível adicionar monitoramento via navegador, registrar os níveis de leitura em um banco de dados PostgreSQL e ainda comandar o escoamento de qualquer um dos contêineres monitorados
- Além dos nodes padrão, foram usadas outras dependências para fornecer recursos extras, como a interface do painel, bancos de dados e outras funcionalidades

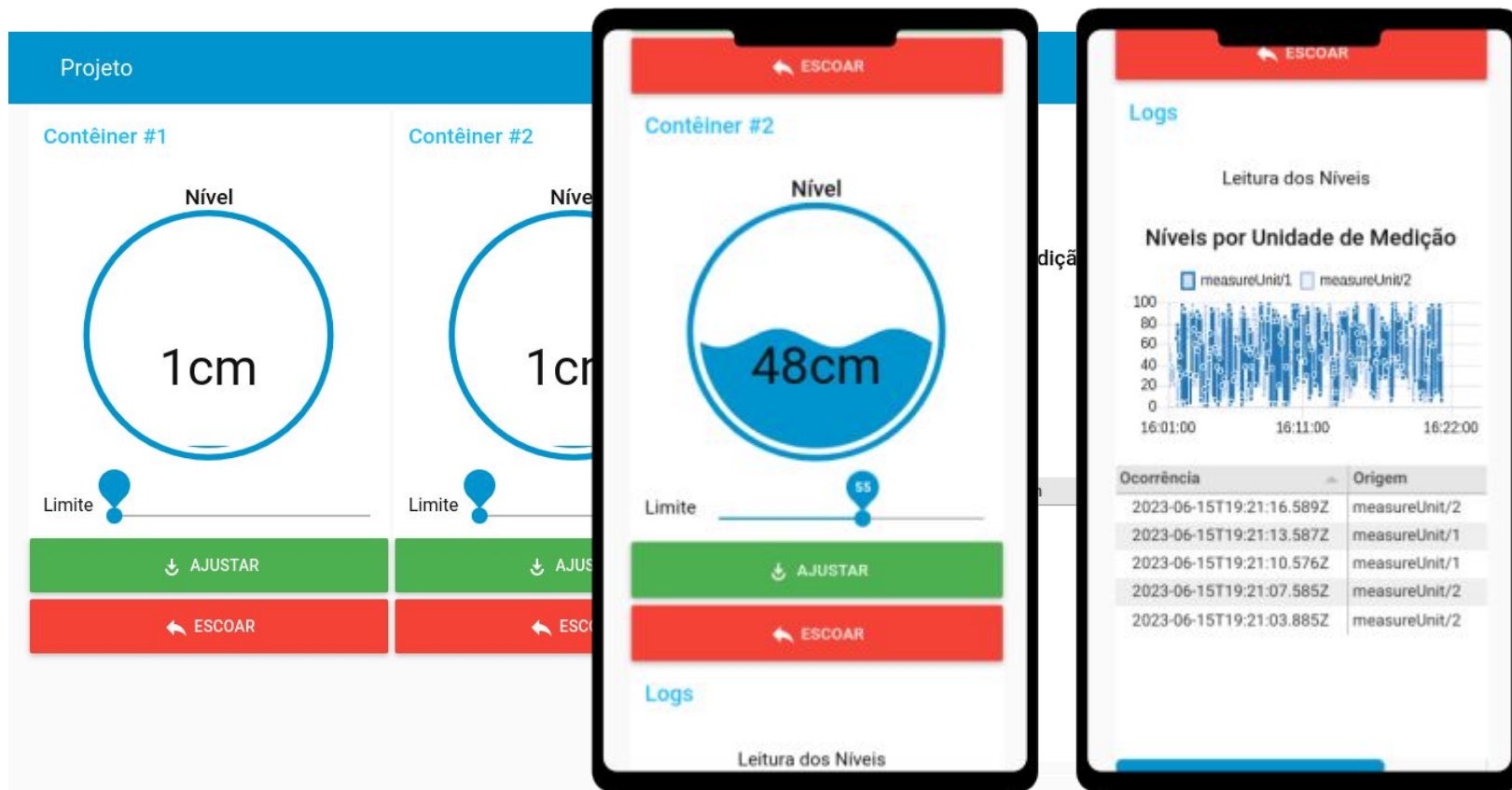


Captura de Tela: Próprios autores



# Desenvolvimento | Monitoramento via Navegador

- O painel de controle (*dashboard*). Fornece monitoramento para duas unidades de medição e controles para comandar as unidades de escoamento, além de manter registros (*logs*) das medições ao longo do tempo



Captura de Tela: Próprios autores

# Conclusão | Problemas e Possíveis Soluções

- **Fragilidade do sensor ultrassônico HC-SR04**

- Dependendo da movimentação da água no contêiner, o sensor resulta em leituras erráticas;
- Algumas vezes, com pouco conteúdo no contêiner, o sensor resulta em leituras com diferenças sensíveis entre uma leitura e a seguinte. Suspeitamos do "click" ultrassônico reverberando nas paredes do contêiner, pois aumentando a distância do sensor em relação à borda superior, o problema ocorre com menor frequência;
- Uma possível solução seria a implementação de um "ciclo de aferição", quando a diferença entre uma leitura e outra atingir um certo limite, antes de sinalizar uma "leitura crítica";
- Outra solução que chegamos a discutir é a utilização de um algoritmo PID (Proporcional, Integral, Derivativo) digital, que é comumente empregado na indústria.

# Conclusão | Problemas e Possíveis Soluções

- **Ausência de sinal WiFi**

- Esse talvez seja o maior problema: manter as unidades comunicando-se com a infraestrutura. Uma provável solução seria adicionar meios de comunicação alternativos, talvez LoRa ou LoRaWan, e integrá-los à infraestrutura. Porém, para este protótipo, optamos por manter apenas a comunicação via WiFi.

- **Indisponibilidade do serviço MQTT**

- Um ponto crítico de falha na infraestrutura pois impossibilitaria a comunicação entre as unidades. Uma solução viável seria a implementação de redundância, mantendo mais de um serviço MQTT disponível e implementando balanceamento de carga.

# Conclusão | Possíveis Melhorias

- **Melhorar a precisão da leitura**

- Como citado anteriormente, os sensores HC-SR04 são baratos e úteis mas relativamente frágeis e suscetíveis a perturbações que podem ser atenuadas através de algum algoritmo que faça uma aferição das leituras antes de sinalizar o escoamento ou a implementação de alguma variação de um algoritmo de PID digital;

- **Atualização de software nos microcontroladores**

- Placas de prototipação baseadas no ESP32, como as utilizadas neste protótipo, requerem que os dispositivos sejam conectados fisicamente através de um cabo USB. Uma maneira de facilitar esse processo é a implementação do OTA (*Over-the-Air updates*) permitindo a atualização do software remotamente;

# Conclusão | Possíveis Melhorias

- **Credenciais de acesso ao WiFi**

- Manter as credenciais de acesso à rede via WiFi escritas diretamente no código (*hardcoded*) é uma prática perigosa, pois algum descuido poderia expor essas credenciais. Uma solução é a utilização da biblioteca WiFiManager<sup>2</sup> que permite configurar o acesso à rede WiFi de forma remota.

<sup>2</sup> WiFiManager library. Arduino documentation. Disponível em <https://docs.arduino.cc/libraries/wifimanager/> acesso em 5 dez. 2024;



# Disponibilidade do Projeto



Todos os códigos-fonte, listas de componentes, diagramas esquemáticos, instruções detalhadas para implantação da infraestrutura de servidores e aplicações, os códigos-fonte LaTeX da monografia, inclusive esta apresentação, estão disponíveis em um repositório de acesso público na plataforma GitHub no endereço:

`https://github.com/danielgoncalves/iot-ifsp-ctd-tcc`

# Obrigado



**INSTITUTO  
FEDERAL**

São Paulo

---

Campus  
Catanduva