



Controle e gestão do consumo de água das residências: usando a IoT para economizar e gerir os recursos hídricos

Raul Vitorino Silva¹, Andre Luis de Oliveira¹

¹ Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil

{10408472}@mackenzista.com.br

Abstract. *This article aims to develop an automation project using IoT to provide control and management of water consumption in Brazilian homes by processing data generated by sensors, as well as controlling water waste due to malfunctioning traditional mechanical equipment. The project will involve the use of Arduino, a water flow sensor, a water level sensor and a water pump. This project does not aim to generate exorbitant costs for its implementation in homes, as it aims to be compatible with traditional equipment already used in Brazilian homes.*

Resumo. *Esse artigo tem como propósito o desenvolvimento de um projeto para automação por meio da IoT, a fim de proporcionar o controle e a gestão do consumo de água das residências brasileiras por meio do processamento dos dados gerados pelos sensores, bem como, o controle de desperdícios de água devido ao mal funcionamento de equipamentos mecânicos tradicionais. O projeto vai envolver o uso de arduino, um sensor de fluxo de água, um sensor de nível de água e uma bomba de água. Esse projeto não visa gerar custos exorbitantes para sua implementação nas residências, pois tem como meta ser compatível com os equipamentos tradicionais já utilizados nas casas dos brasileiros.*

1. Introdução

O Brasil se destaca negativamente quando se trata do desperdício de água potável, havendo perda de 39,2% de toda a água potável captada, segundo dados do estudo do Instituto Trata Brasil. A quantidade desperdiçada seria o suficiente para abastecer aproximadamente 63 milhões de brasileiros por 1 ano. Boa parte desse desperdício se dá a más condições da estrutura de saneamento, ou seja, envolve o poder público.

Entretanto, não é incomum encontrar uma casa ou edifício derramando água na rua porque a caixa encheu, ou então contas de água excessivamente altas por uso indiscriminado da água, e desse ponto de vista, não envolve o poder público, depende somente da conscientização das pessoas em buscar melhorias para suas casas a fim de reduzir custos, evitar desperdícios, gerir e controlar melhor o gasto de água.

O projeto visa desenvolver um sistema de automação para controlar e monitorar constante e periodicamente o consumo de água de uma residência, sendo capaz de gerar dados e processá-los a fim de poder identificar oportunidades de melhorias no consumo

de água e momentos de desperdício, tudo isso por meio da relação harmônica entre dispositivos modernos de geração de dados/processamento (como o arduino, seus sensores e atuadores) com os instrumentos tradicionais hidráulicos que as casas e edifícios já possuem, para não gerar custos muito altos para as famílias a ponto de tornar a ideia do projeto inviável financeiramente.

2. Materiais e métodos

Os componentes de hardware usados no projeto serão o esp32, esp01, um sensor de nível de água, um sensor de fluxo de água e uma mini bomba de água,

O esp32 é um micro controlador usado na IoT para desenvolver protótipos de automação. É um dispositivo capaz de se comunicar com vários sensores e atuadores simultaneamente, podendo enviar e receber dados via internet. Nesse projeto, o esp32 vai enviar dados via internet por meio do protocolo MQTT (Message Queuing Telemetry Transport). Esse protocolo permite a troca de mensagens entre publicadores e subscritores com intermédio de um broker, o qual faz a gestão e o controle dessas mensagens. O componente de software usado no projeto será o software da Arduino IDE para a programação e para a visualização das mensagens enviadas via mqtt, será utilizado o mqtt explorer.

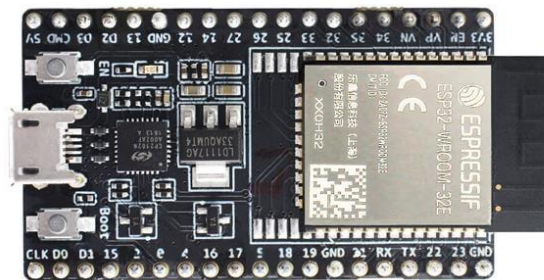


Imagem 1: esquema do esp 32. Fonte: https://br.mouser.com/images/espressifsystems/lrg/ESP32-DevKitC-32E_SPL.jpg

O esp01 é um micro controlador de dimensões bem reduzida, ideal para projetos que necessitam de poucos pinos ou que sejam acoplados a um rele.

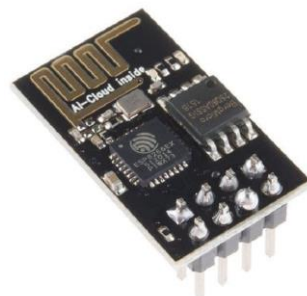


Imagem 2: esquema do esp 01. Fonte: <https://www.makerhero.com/wp-content/uploads/2017/07/6WL01-M%C3%B3dulo-WiFi-ESP8266-ESP-01-1-min.jpg>

O sensor do nível de água é do tipo boia, será usado um sensor no nível alto para indicar quando o recipiente com água está cheio e quando o nível de água baixar e o reservatório não estiver mais cheio, acionando a bomba d'água. O funcionamento desse sensor é do tipo chave liga-desliga. Quando a água chega no nível da boia, ele empurra a boia, fechando contato e transmitindo a corrente elétrica.



Imagem 2: sensor de nível de água. Fonte: https://http2.mlstatic.com/D_NQ_NP_2X_676387-MLB43496907769_092020-F-sensor-de-nivel-de-agua-interruptor-de-boia-e-mini-boia.webp

Pressão suportada	0.2 MPa;
Temperatura de funcionamento	-10 a 85°C;
Comprimento Do cabo	36cm (aprox)

Tabela 1: parâmetros do sensor de nível de água. Fonte: https://www.casadarobotica.com/sensores-e-modulos/sensores/outros/sensor-nivel-lateral-agua-arduino-microcontrolador-aquario?srsId=AfmBOorH_TTVhjz-9QVbJEE458jRIhqAyr9P6jNYxbb5bZHat4L6dk4m.

O sensor de fluxo de água é um dispositivo capaz de medir a quantidade de água que está passando por unidade de tempo. A função dele no projeto é para determinar quanto de água está sendo consumida periodicamente e armazenar esses dados para consulta por parte dos moradores a fim de monitoramento.



Imagem 3: Sensor de fluxo de água. Fonte: https://www.eletrogate.com/sensor-de-fluxo-de-agua-de-1-2?srsId=AfmBOoo2G_Ext0Dh6l_41RDirGVjO5E2fSH_iuGCep80lpRAR7sB6ZhT.

Fluxo de Água	1 a 30 litros / minuto
Temperatura de operação	-25 a + 80 °C
Precisão	± 10%
Tensão de operação	5V a 18V
Tipo de sensor	Efeito Hall
Modelo	YF-S201

Tabela 2: parâmetros e especificações do sensor de fluxo de água. Fonte: https://www.casadarobotica.com/sensores-e-modulos/sensores/outros/sensor-nivel-lateral-agua-arduino-microcontrolador-aquario?srsId=AfmBOorH_TTVhjz-9QVbJEE458jRIhqAyr9P6jNYxbb5bZHat4L6dk4m.

A mini bomba de água vai ser o atuador responsável por encher o reservatório quando receber o sinal do sensor de nível de água baixo, bem como, parar de encher o reservatório quando receber o sinal do sensor do nível de água alto.



Imagem 4: mini bomba de água. Fonte: <https://www.eletrogate.com/mini-bomba-de-agua-motor-dc-rs-385?srsId=AfmBOor8xhf7Jwe75bqnlaczfYKlqrV5G-upJRPp79Wlfnf-sEpcuLK>

Precisão	$\pm 10\%$
Vazão por Hora	Mínimo 90L; Máximo 120L por hora
Tensão de funcionamento	DC12V

Tabela 3: parâmetros e especificações da mini bomba de água. Fonte: https://www.casadarobotica.com/sensores-e-modulos/sensores/outros/sensor-nivel-lateral-agua-arduino-microcontrolador-aquario?srsId=AfmBOorH_TTVhjz-9QVbJEE458jRIhqAyr9P6jNYxbb5bZHat4L6dk4m

Para programação do micro controlador, será utiliza IDE do arduino e a linguagem C++. O servidor para comunicação via MQTT será o <https://broker.hivemq.com/>

Abaixo, segue um modelo de montagem do projeto.

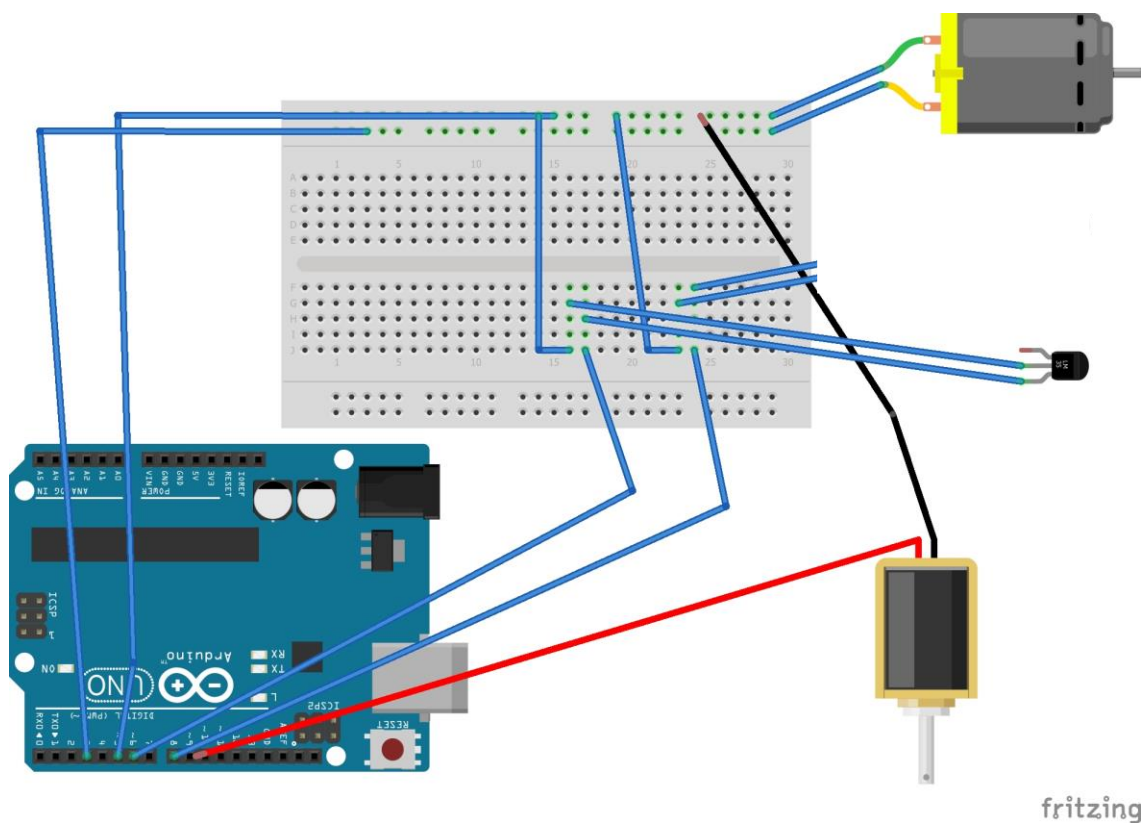
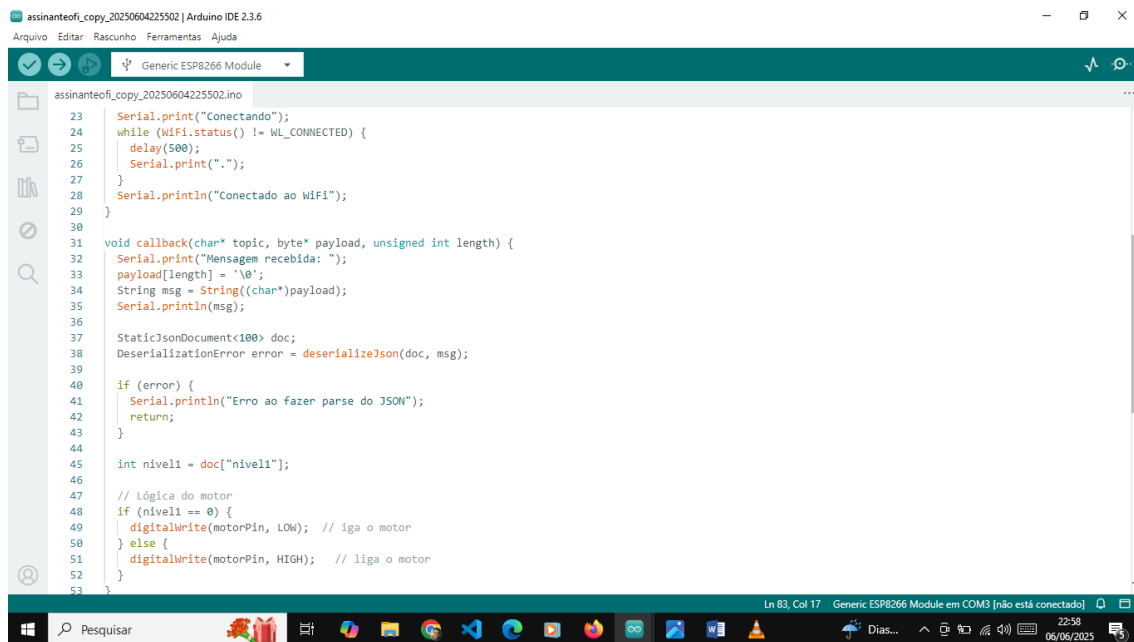


Imagem 5: modelo do projeto com adaptações.

As adaptações são referentes aos componentes, que não estão disponíveis no software fritzing. O sensor de nível de água está representado na forma de sensor de temperatura, o sensor de fluxo de água está representado na forma de uma válvula solenoide e a bomba de água está representada como um motor e o esp32 está representado pelo arduino uno.

Imagem 6: os códigos do publicador e assinante abertos lado a lado com seus respectivos monitores serial sendo exibidos.

A imagem 6 mostra os códigos do assinante e publicador, bem como, a transmissão do sensor de nível pelo esp 32 (IDE da direita) e o recebimento dos dados pelo esp 01 (IDE da esquerda), Nesse print também é possível ver as mensagens que são exibidas no monitor serial do esp 32 e esp 01 (respectivamente, publicador e assinante) quando estão ligados nas suas portas usb desde o início, com a conexão do WIFI sendo feita com sucesso, a conexão ao broker também feita com sucesso e as mensagens que são exibidas no monitor serial quando ocorrem essas conexões.



```
assinanteof copy_20250604225502.ino
23 Serial.print("Conectando");
24 while (WiFi.status() != WL_CONNECTED) {
25   delay(500);
26   Serial.print(".");
27 }
28 Serial.println("Conectado ao WiFi");
29 }
30
31 void callback(char* topic, byte* payload, unsigned int length) {
32   Serial.print("Mensagem recebida: ");
33   payload[length] = '\0';
34   String msg = String((char*)payload);
35   Serial.println(msg);
36
37   StaticJsonDocument<100> doc;
38   DeserializationError error = deserializeJson(doc, msg);
39
40   if (error) {
41     Serial.println("Erro ao fazer parse do JSON");
42     return;
43   }
44
45   int nivel1 = doc["nivel1"];
46
47   // Lógica do motor
48   if (nivel1 == 0) {
49     digitalWrite(motorPin, LOW); // liga o motor
50   } else {
51     digitalWrite(motorPin, HIGH); // liga o motor
52   }
53 }
```

Imagem 7: trecho de código do assinante no qual consta as funções de callback para receber os dados enviados por mqtt do esp 32 (publicador)

A imagem 7 mostra o código do assinante, no qual constam as funções de funcionamento do motor mediante a mensagem que recebe via mqtt, se a mensagem foi um, a bomba d'água é ligada, se a mensagem for zero, a bomba d'água é desligada. Outras funções importantes também estão sendo mostradas, como a função callback, ela recebe o payload do publicador via mqtt e exibe a mensagem enviada no monitor serial.


```

1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 // Config WiFi
6 const char* ssid = "cavalo de troia";
7 const char* password = "francisca49";
8
9 // Config MQTT
10 const char* mqtt_server = "broker.hivemq.com";
11 const int mqtt_port = 1883;
12 const char* mqtt_topic = "nivel/sensores";
13
14 WiFiClient espClient;
15 PubSubClient client(espClient);
16
17 // Pino do motor (relé ou driver)
18 const int motorPin = 2; // GPIO2 (D4 no ESP01)
19
20 void setup_wifi() {
21   delay(10);
22   WiFi.begin(ssid, password);
23   Serial.print("Conectando");
24   while (WiFi.status() != WL_CONNECTED) {
25     delay(500);
26     Serial.print(".");
27   }
28   Serial.println("Conectado ao WiFi");
29 }
30
31 void callback(char* topic, byte* payload, unsigned int length) {

```

Imagem 8: trecho de código do assinante no qual consta as funções de conexão do WIFI, do broker e as mensagens que são exibidas quando as conexões têm sucesso.

A imagem 8 mostra o código do assinante, no qual constam as declarações do mqtt utilizada, a porta utilizada para comunicação e também o tópico utilizado, que é igual ao tópico utilizado pelo publicador. Há também as bibliotecas utilizadas, nome e senha da rede WIFI para realizar conexão, bem como, a função que faz a conexão do WIFI, que enquanto a conexão não tem sucesso, exibe um ponto no monitor serial, até que a conexão é feita, quando então é exibido a mensagem de “conectado” e “conectado ao WIFI”.

Sensor/atuador	Tempo de resposta
Sensor de fluxo	3 segundos
Sensor de fluxo	3 segundos
Sensor de fluxo	3 segundos
Sensor de fluxo	3 segundos – Tempo médio de 3 segundos
Sensor de nível	3 segundos
Sensor de nível	3 segundos
Sensor de nível	3 segundos
Sensor de nível	3 segundos – Tempo médio de 3 segundos
Bomba d’água	2 segundos
Bomba d’água	3 segundos

Bomba d'água	3 segundos
Bomba d'água	2 segundos – Tempo médio de 2,5 segundos
Comunicação via mqtt do esp 32 com o esp 01	2 segundos
Comunicação via mqtt do esp 32 com o esp 01	3 segundos
Comunicação via mqtt do esp 32 com o esp 01	3 segundos
Comunicação via mqtt do esp 32 com o esp 01	2 segundos – Tempo médio de 2,5 segundos

Tabela 1: tabela mostrando os tempos para cada sensor responder as mudanças e para o atuador funcionar conforme a mudança do sensor

Em seguida, segue o gráfico contendo os dados dos tempos para cada sensor e para a comunicação via mqtt entre o esp 32 e o esp 01 medidos em segundos.

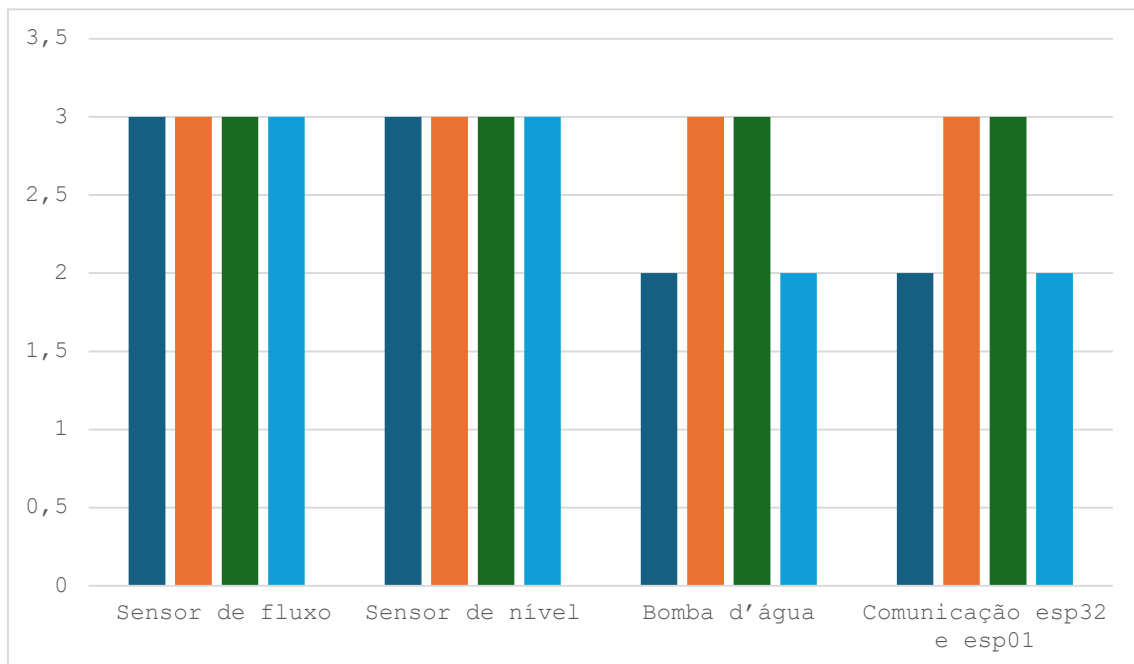
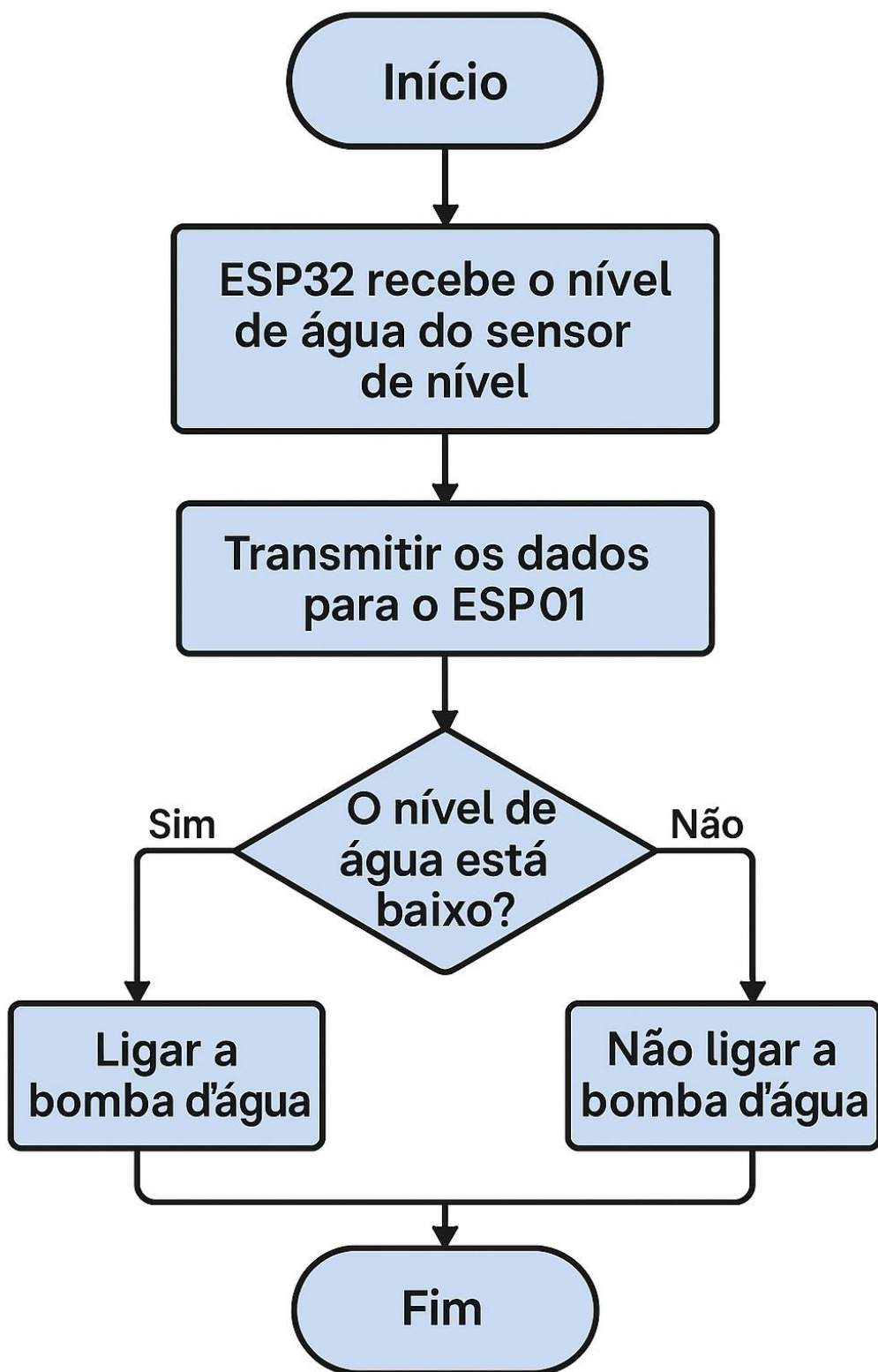


Gráfico 1: compilação dos dados da tabela 1



Fluxograma 1: funcionamento básico da comunicação via mqtt do projeto

4. Conclusões

Os resultados apresentados representam que foi possível desenvolver um projeto com dois sensores e um atuador. No início foi desafiante imaginar como faria a conexão via protocolo mqtt, mas com ajuda de amigos, muitos testes e ajustes, foi possível realizar a conexão e entender quanto simples e poderoso esse protocolo de comunicação pode ser.

Os objetivos do projeto foram alcançados quase na sua totalidade, pois o sensor de fluxo não ficou calibrado e monitorando quantidades exatas e sem ruído. Mas como foi falado no vídeo do youtube, a provável causa desses ruídos e monitoramento não exato do sensor de fluxo se deve a parte elétrica, por não ser um circuito elétrico adequado para usar um sensor de fluxo com precisão e com baixo ruído. Os principais problemas enfrentados foram a conexão dos dois esp com um mqtt, tentei vários mqtt diferentes até encontrar esse, acredito que com os outros mqtt's havia a necessidade de fazer uma codificação diferente, com especificidades para cada mqtt, não precisando usar apenas o url do broker, porta e tópico. Quando um mqtt que usasse apenas esses dados foi encontrado, a conexão entre os dois esp teve sucesso. Outro problema encontrado foi, como já mencionado, a calibração e redução de ruído do sensor de fluxo. Tentei fazer melhorias, mas ainda ficou com muito ruído e pouco exato.

Outra dificuldade para a realização do projeto foi a tentativa de uso do arduino uno, não obtendo resultados satisfatórios para conexão do WIFI e nem do broker, passo iniciais e primordiais para o projeto poder se desenvolver. A partir do momento que houve a troca do arduino uno para os esp 32 e esp 01, facilitou muito o processo de conexão do WIFI e do broker.

As vantagens do projeto são: simples, barato, fonte de muito aprendizado sobre a comunicação entre dois dispositivos via protocolo mqtt, o funcionamento exato e com total sincronia do sensor de nível e da bomba d'água. As desvantagens do projeto: a dificuldade em fazer um esquema elétrico adequado para haver a leitura correta pelo sensor de fluxo, bem como, a dificuldade para calibrar o mesmo sensor.

Para melhorar o projeto, seriam necessários mais estudos a fim de entender e poder desenvolver um melhor esquema elétrico para o senso de fluxo para poder haver a redução do ruído e a possibilidade de calibração.

5. Referências

Brasil chega aos 40% de perdas de água potável. 4 de abr. de 2022. Disponível em:
<https://tratabrasil.org.br/brasil-chega-aos-40-de-perdas-de-agua-potavel/>
Acesso em: 07 de abril de 2025

Esp primeiros passos. Disponível em:
<https://www.youtube.com/playlist?list=PL7CjOZ3q8fMe6DxoJEFuDx4BP0qbbpKtP>

Acesso em: 02 de junho de 2025

Como o sistema de IoT pode ajudar na redução do desperdício de água? Disponível em:
<https://egesolucoes.com.br/como-o-sistema-de-iot-pode-ajudar-na-reducao-do-desperdicio-de-agua/#:~:text=Os%20sistemas%20da%20Internet%20das,da%20madrugada%20em%20um%20cliente.>

Acesso em: 02 de junho de 2025