



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Alimentador Automático para Peixes

Caio Fillipi dos Santos

¹Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-
907 – Brazil

`caio.fillipi@outlook.com`

Abstract. *This article describes the entire process of developing an automatic fish feeder, from its creation to its final conception.*

Resumo. Este artigo descreve todo o processo de desenvolvimento de um alimentador automático para peixes, desde a sua elaboração até sua concepção final.

1 Introdução

O avanço tecnológico permitiu que diversas novas áreas anteriormente inexploradas, pudessem ser estudadas de uma forma ampla, como o IoT. Lâmpadas, tomadas e casas se tornaram inteligentes, chegando a identificar a presença de alguém em um determinado cômodo e com isso acender as luzes ou analisar a temperatura e ligar o ar-condicionado, mantendo assim a temperatura do ambiente agradável. Diversas tarefas do nosso cotidiano puderam ser automatizadas, graças aos grandes avanços tecnológicos. Ao analisar as possibilidades de implementar uma solução de automação, identifiquei uma tarefa muito comum em casas que possuem pets, alimentar os pets. É possível encontrar diversos projetos que automatizam essa tarefa, porém grande parte são para animais de médio porte, como cães, gatos, coelhos entre outros. Ao analisar outros cenários, encontrei pets que são totalmente dependentes de nós para sua alimentação, os peixes. Peixes apresentam uma maior fragilidade se comparada a outros animais, caso não sejam alimentados da maneira correta eles podem morrer e causar tanto danos emocionais como perdas financeiras, principalmente em aquários de água salgada, na qual os peixes têm preços muito mais elevados em relação aos de água doce e requerem uma maior atenção. Atualmente existem diversas soluções que são ofertadas no mercado, com as mais diversas especificações. Pensando nesse cenário decidi desenvolver meu próprio alimentador automático utilizando a plataforma Arduino, permitindo assim que eu me ausente por longos períodos sem me preocupar com a alimentação dos peixes.

2 Materiais e Métodos

Plataforma: Arduino Uno

Descrição: É uma placa de microcontrolador baseada no Atmega328P. Possui 14 pinos de entrada / saída digitais, 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, uma tomada de força, um cabeçalho ICSP e um botão de reinicialização.



*Figura: Placa Arduino. Fonte:
<https://store-usa.arduino.cc/products/arduino-uno-rev3>*



*Figura: Mini Motor DC. Fonte:
encurtador.com.br/nuGL9*

O mini motor DC será utilizado para fazer a dispersão de alimento dos peixes.

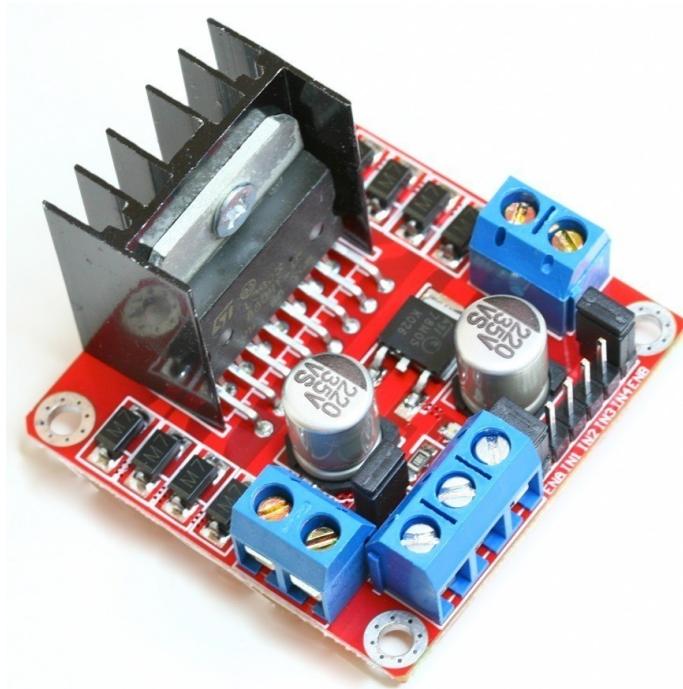


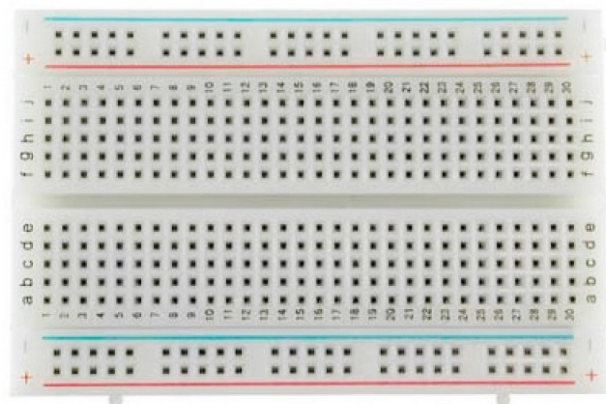
Figura: Figura : Driver Ponte H L298N Fonte:
<https://create.arduino.cc/projecthub/now/l298n-motor-module-service-ba0f56>

A ponte H, será utilizada para realizar o controle do motor e fornecer potência.



Figura: Modulo ESP32 Fonte:
<https://www.embarcados.com.br/wp-content/uploads/2018/01/projeto-iot-portatil-850x510.png>

O módulo ESP32 será responsável por realizar a comunicação via MQTT.



*Figura: Placa Protoboard de 400 Pontos. Fonte:
<https://www.systronyx.com/image/cache/catalog/mediaprotol-800x640.jpg>*

Placa Protoboard utilizada para realizar as ligações dos circuitos.



*Figura: Porta Pilhas para 4 Pilhas. Fonte:
https://http2.mlstatic.com/D_NQ_NP_987254-MLB46742289443_072021-W.jpg*

Porta pilhas utilizado para realizar a alimentação da Ponte H.



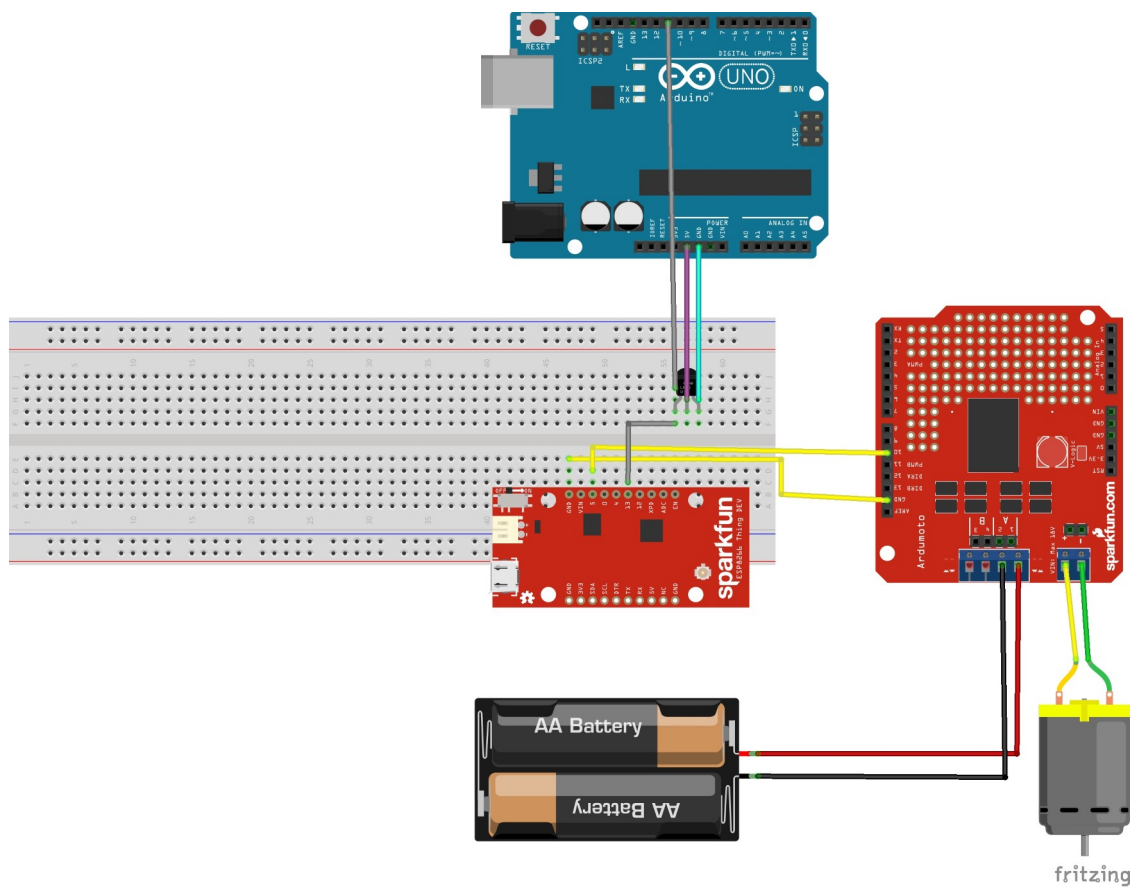
Figura: Sensor de Temperatura DS18B20.
Fonte: https://www.huinfinito.com.br/1490-large_default/sensor-de-temperatura-ds18b20-prova-d-agua.jpg

Sensor de temperatura DS18B20, escolhi ele por ter uma maior precisão entre os sensores.

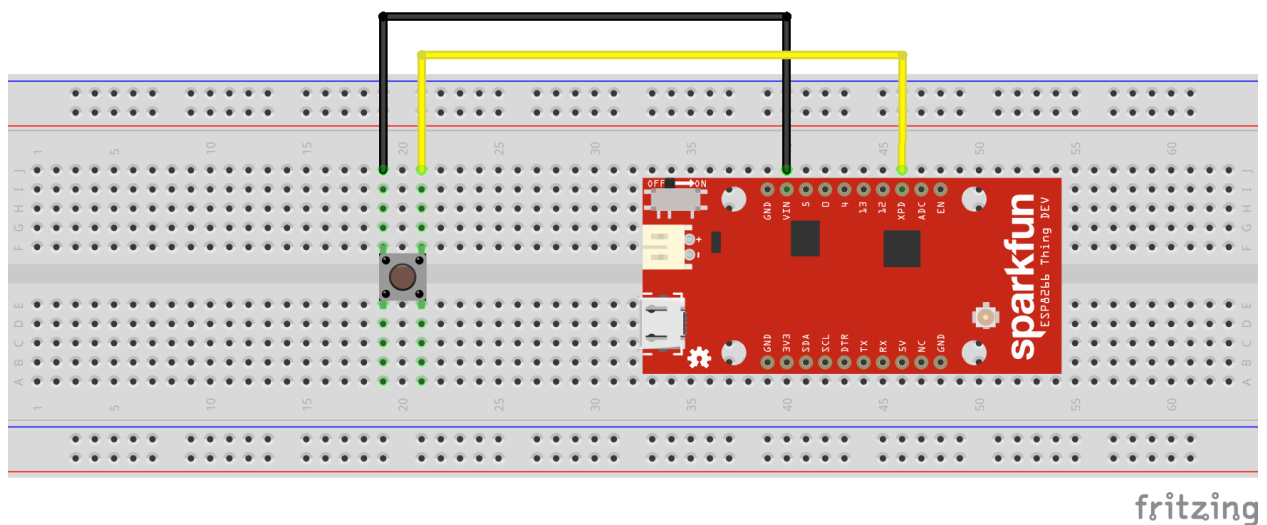


Figura: Push Button 4 pinos. Fonte: https://http2.mlstatic.com/D_NQ_NP_821808-MLB46512363443_062021-O.webp

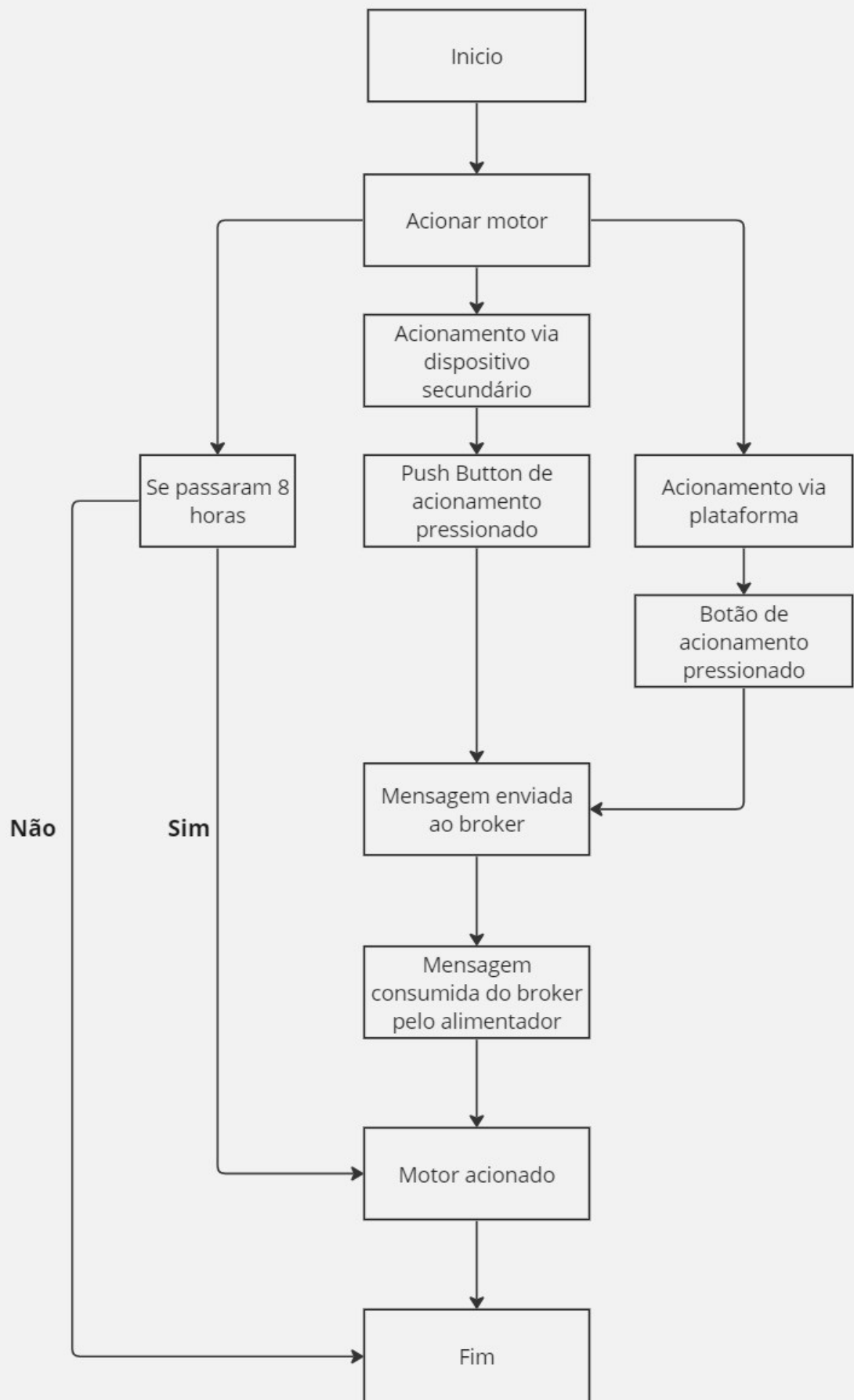
Push Button de 4 pinos, será o gatilho responsável pelo envio de mensagens para o Broker pelo dispositivo de acionamento secundário.



Projeto do alimentador automático de peixes. Esse projeto montado no Fritzing utilizei outro tipo de sensor de temperatura, pois não tinha disponível a versão tipo sonda.



Projeto do dispositivo secundário que permitirá o acionamento remoto do alimentador de peixes, via internet utilizando o protocolo MQTT.



Acima segue um fluxograma exemplificando os modos de acionamento do alimentador, que são o acionamento automático a cada 8 horas, acionamento com o dispositivo secundário e o acionamento via portal.

Funcionamento

O protótipo possui 3 modos de acionamento, um acionamento automático que ocorrerá a cada 8 horas, outro acionamento que será realizado por meio de um site utilizando MQTT e o terceiro por meio de um dispositivo secundário que quando conectado a internet realizará a comunicação via MQTT com o alimentador. O gerenciamento do sensor de temperatura é feito por meio de algumas bibliotecas para operar sensores.

No modo de ativação utilizando o MQTT, será aberto um site intuitivo para realizar o acionamento do motor. Nele existe um botão para realizar a ativação do motor e uma label exibindo a temperatura da água em C°. Para que ocorra a comunicação do portal com o broker, utilizei a biblioteca MQTT.js(<https://github.com/mqttjs/MQTT.js>).

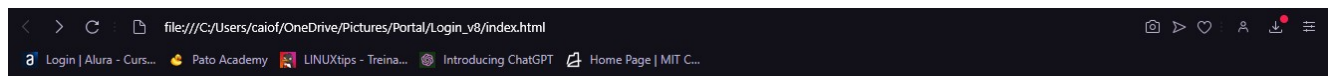
No modo de ativação remota com a utilização do dispositivo secundário, será feito via internet com a utilização do protocolo MQTT e um Broker gratuito. O Broker a ser utilizado será o Mosquitto(<https://test.mosquitto.org>), que permitirá a comunicação entre o dispositivo secundário e o alimentador.

No modo de ativação automático o motor está programado para rodar a cada 8 horas, esse tempo pode ser editado. Esse tempo pode ser editado no método **startEngineAutomatic()**, dentro do método basta alterar o valor dentro da estrutura condicional. Um ponto de atenção é que o tipo da variável utilizada para determinar o valor do intervalo é **int**, então o valor máximo que ela suporta é 2147483647 que corresponde a 592.52 horas, não é possível colocar um valor superior a este.

Para realizar o controle do sensor de temperatura utilizei as bibliotecas **OneWire** e **DallasTemperature**, existem diversas outras. A utilização dessas bibliotecas é simples, antes de utiliza-lás é necessário realizar a importação delas na IDE do Arduino, caso não encontre em sua IDE é possível realizar o download do arquivo zip pelo site <https://www.arduino.cc/reference/en/libraries/>, após baixar basta importar o arquivo zip em sua IDE.

Abaixo estão algumas capturas de tela do portal, na qual será possível realizar o acionamento do motor via MQTT e também monitorar a temperatura por meio do sensor DS18B20.

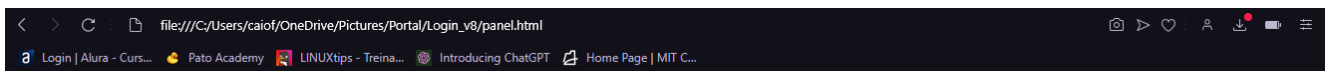
Tela de login do portal.



Bem-Vindo

ACESSAR

Tela com os controles do alimentador, o botão para acionamento do motor e o sensor de temperatura.



Pressione para acioar o motor:

Acionar Motor

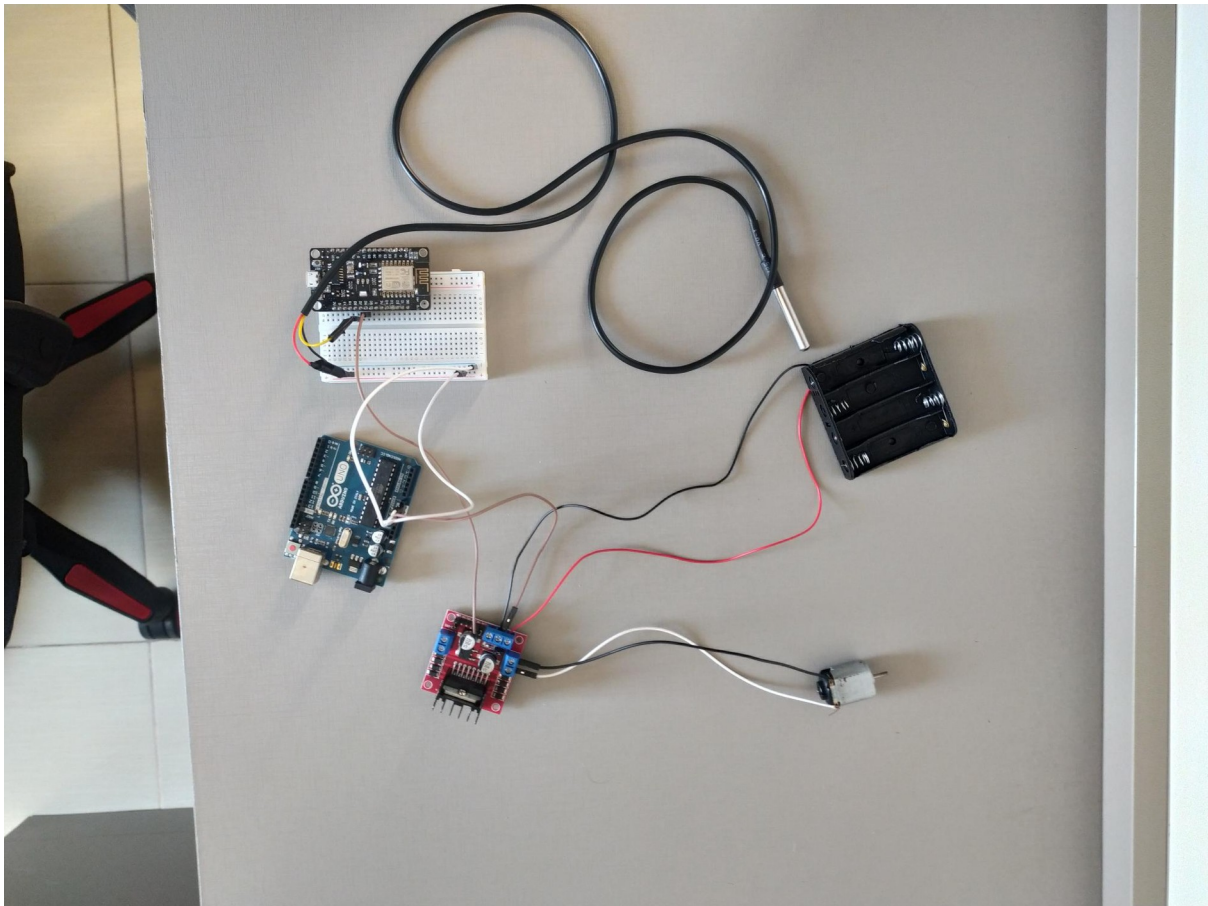
Temperatura: 22° C

3 Resultados

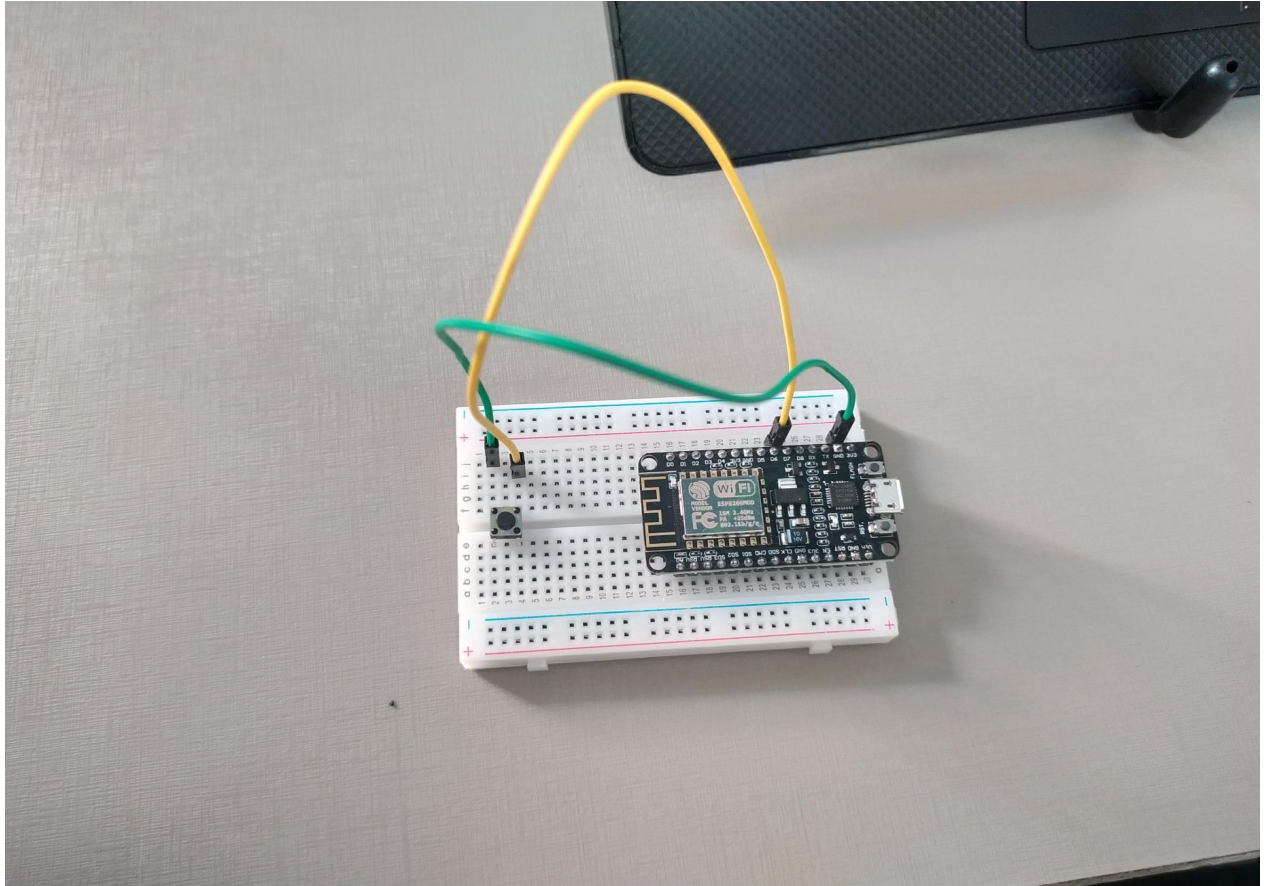
Abaixo está a tabela com as medições feitas para o sensor e atuador presente no protótipo. Essa medição tem o intuito de exibir o tempo de comunicação entre o protótipo e o broker MQTT.

Núm. Medida	Sensor/Atuador	Tempo Resposta
1	Atuador - Motor DC	00:00.89
2	Atuador - Motor DC	00:00.86
3	Atuador - Motor DC	00:00.91
4	Atuador - Motor DC	00:00.80
5	Sensor DS18B20	00:01:35
6	Sensor DS18B20	00:01:48
7	Sensor DS18B20	00:01:31
8	Sensor DS18B20	00:01:35
Tempo Médio Motor DC		00:00.86
Tempo Médio Sensor		00:01:37

Abaixo uma imagem do alimentador de peixes montado. É possível visualizar todos os componentes utilizados e as ligações realizadas no projeto



Abaixo está o dispositivo secundário desenvolvido, que será responsável pelo acionamento remoto via internet utilizando o protocolo MQTT.



No trecho de código abaixo estão alguns passos para a realização da conexão do broker e da internet. Passamos o nome da rede e senha do Wifi para a conexão do Node MCU, também o endereço do broker e porta. As variáveis ID_MQTT e QUEE_TOPIC são responsáveis pelo envio para o broker, nesse caso o ID_MQTT é a identificação único do dispositivo e QUEE_TOPIC é para qual fila a mensagem será enviada.

```
const char* SSID = "My ASUS_6362";
const char* PASSWORD = "41025199";

const char* BROKER_MQTT = "test.mosquitto.org";
int BROKER_PORT = 1883;

#define ID_MQTT "RTX42"
#define QUEE_TOPIC "Fish_Feeder_Project_Mackenzie"
```

No método **setup()** passamos algumas configurações referentes aos pinos a serem utilizados, iniciamos a conexão com o Wifi, por meio do método **WifiConnection()**, por fim informamos para a biblioteca do MQTT as informações de conexão, que foram definidas nas variáveis acima.

```
void setup() {  
    pinMode(btnPushBroker, INPUT_PULLUP);  
  
    Serial.begin(115200);  
  
    WifiConnection();  
    MQTT.setServer(BROKER_MQTT, BROKER_PORT);  
}
```

O método **MQTTConnection()**, é responsável por gerenciar toda a parte de conexão com o MQTT, nele verificamos se estamos conectados ao broker, caso não seja possível realizar a conexão com o broker, ele faz uma nova tentativa a cada 10 segundos.

```
void MQTTConnection() {  
    while (!MQTT.connected()) {  
        Serial.print("Conectando ao Broker MQTT: ");  
        Serial.println(BROKER_MQTT);  
        if (MQTT.connect(ID_MQTT)) {  
            Serial.println("Conexão com Broker realizada com sucesso!");  
        }  
        else {  
            Serial.println("Conexão com o Broker ");  
            Serial.println(BROKER_MQTT);  
            Serial.println(" falhou!");  
            Serial.println("Nova tentativa de conexão em 10 segundos");  
            delay(10000);  
        }  
    }  
}
```

Abaixo estão os links de acessos para o repositório contendo todo o código desenvolvido para a concepção do projeto e o link de acesso para o vídeo no Youtube com a demonstração do projeto em funcionamento.

Link para repositório Github: <https://github.com/CaioFillipi/objetos-inteligentes-conectados>

Link apresentação Youtube: <https://youtu.be/TGeEVUa8AbI>

4 Conclusão

Com o projeto concluído, foi possível verificar que todos os objetivos propostos foram alcançados. Todos os pontos pensados durante a fase de planejamento foram atingidos com sucesso.

Durante o desenvolvimento enfrentei um problema com o Broker que escolhi utilizar, havia escolhido utilizar o Eclipse MQTT, porém após diversas tentativas sem sucesso de conexão, alterei para o broker Mosquitto. Outro problema enfrentado foi em relação a Ponte H, não estava passando corrente para as saídas do motor, após diversos testes foi necessário realizar a substituição da Ponte H.

O projeto tem a grande vantagem de permitir o controle do alimentador de forma remota, eliminando assim uma grande preocupação das pessoas que possuem aquário, que é a questão de alimentar os peixes quando eles estão longe. Com o alimentar de peixes integrado ao protocolo MQTT foi possível eliminar esse problema. Uma desvantagem presente no projeto é a questão da dependência da internet do local onde o alimentador está, caso ocorra uma falha na provedora de internet, não será possível realizar o acionamento remoto, apenas o acionamento automático programado estará disponível até que o serviço de internet seja reestabelecido.

Uma opção que apresentaria uma melhora significativa para o projeto seria a adição de um módulo de SIM Card, caso ocorresse uma falha na internet local, seria acionado o módulo de SIM Card para realizar a conexão com a internet, assim evitando que o acionamento remoto ficasse indisponível.

5 Referências

BARBOSA, Vítor. Pet Center - Feed and Entertain Your Pet. **Project Hub**, 2018. Disponível em: <https://create.arduino.cc/projecthub/vitor-barbosa/pet-center-feed-and-entertain-your-pet-f6a1c1?ref=tag&ref_id=mqtt&offset=6/>. Acesso em: 5 de mar. de 2023.

Brincando com Ideias. Como Programar o ESP8266 para ser um WEB SERVER, Youtube, 18 mar de 2018. Disponível em: <<https://www.youtube.com/watch?v=kCjWYQEMvwc&list=PL7CjOZ3q8fMe6DxoJEFuDx4BP0qbbpKtP&index=3>> Acesso em: 5 de mar. de 2023.

Brincando com Ideias. Como Fazer um WebServer com ESP8266 sendo Controlado pela Web, **Youtube**, 25 mar de 2018. Disponível em: <<https://www.youtube.com/watch?v=woPFuwvHPoA&list=PL7CjOZ3q8fMe6DxoJEFuDx4BP0qbbpKtP&index=4>> Acesso em: 5 de mar. de 2023.

COSTA, Daniel. Como controlar um motor com o Arduino. **PPLWARE**, 2014. Disponível em: <<https://pplware.sapo.pt/tutoriais/como-controlar-um-motor-com-o-arduino/>> Acesso em: 10 de mar. de 2023.

OLIVEIRA, Euler. Como usar com Arduino – Sensor de Temperatura DS18B20 Prova D'água do Tipo Sonda. **MASTERWALK BLOG**, 2021. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-temperatura-ds18b20-prova-dagua-do-tipo-sonda>> Acesso em: 10 de mar. De 2023.

STRAUB, Matheus Gebert. Comedouro automático com Arduino — Tratador para pets. **USINAINFO**, 2020. Disponível em: <<https://www.usinainfo.com.br/blog/comedouro-automatico-com-arduino-tratador-para-pets/>>. Acesso em: 10 de mar. De 2023.

MQTT - Como Comunicar 2 Dispositivos via Internet (parte 3), **Brincando com Ideias**. Disponível em: <<https://www.youtube.com/watch?v=Ukl1wvq-Tk&t=69s>>. Acesso em: 10 de Maio de 2023.

COPE, Steve. Using The JavaScript MQTT Client With Websockets. **STEVE'S INTERNET GUIDE**, 2023. Disponível em: <<http://www.steves-internet-guide.com/using-javascript-mqtt-client-websockets/>>. Acesso em: 6 de maio. De 2023

LOCATELLI, Caroline. Como utilizar o DS18B20. **CURTO CIRCUITO**, 2021. Disponível em: <<https://curtocircuito.com.br/blog/Categoria%20Arduino/como-utilizar-o-ds18b20>> Acesso em: 10 de maio de 2023