**Integrantes**

550201 - Julia Leite Galvão

99403 - Giovanna Saori Arata

99068 - Jessica Witzler Costacurta

99840 - Carlos Augusto Campos Ganzerli

552511 - Marcio Hitoshi Tahyra

**Projeto Acqua Flora**

**IoT e Sua Importância**

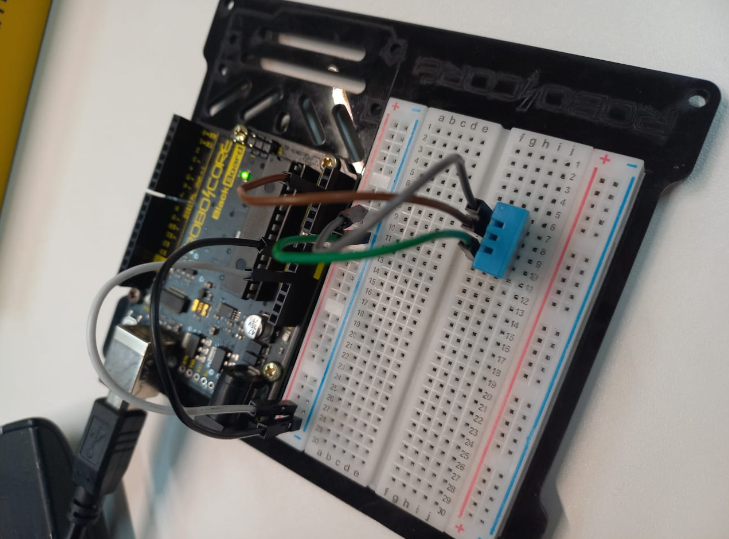
A Internet das Coisas (IoT) é uma tecnologia que conecta dispositivos físicos à internet, permitindo que eles coletem e compartilhem dados. Componentes-chave incluem dispositivos IoT, sensores, conectividade, plataforma IoT, nuvem, análise de dados e segurança. A arquitetura IoT envolve dispositivos IoT conectados a uma plataforma que se comunica com a nuvem, onde os dados são armazenados e processados. A importância da IoT reside em sua capacidade de melhorar a eficiência operacional, qualidade de vida, segurança, sustentabilidade e inovação em várias áreas. Em resumo, a IoT está transformando a maneira como interagimos com o mundo físico, trazendo benefícios significativos para a sociedade e a indústria.

A arquitetura adotada na IoT pode variar dependendo da aplicação, mas geralmente envolve os seguintes componentes principais:

* **Dispositivos IoT:** esses dispositivos coletam dados do ambiente ou executam ações com base em comandos.
* **Rede de Conectividade:** esta camada permite a comunicação dos dispositivos IoT com a plataforma IoT e a nuvem. Pode incluir redes sem fio (Wi-Fi, Bluetooth, 4G/5G) ou redes de baixa potência, como LoRa e Zigbee.
* Plataforma IoT: ela atua como um intermediário entre os dispositivos IoT e a nuvem, gerenciando a conectividade, autenticação, segurança e coleta de dados.
* **Nuvem:** é onde os dados são armazenados e processados em larga escala. Os dados são geralmente processados em tempo real ou em lote, dependendo dos requisitos da aplicação.
* **Aplicações e Serviços:** são as interfaces ou aplicativos que os usuários finais utilizam para interagir com os dados e controlar os dispositivos IoT. Isso pode incluir aplicativos móveis, painéis de controle da web e sistemas de automação residencial.

**Arduino**

Para o nosso trabalho, nós utilizamos uma Arduino Uno e um sensor Dht11 para a leitura dos valores de umidade.



Já no comando do arduino, nós montamos um código em json para que o DHT11 lesse os valores de umidade.

#include <ArduinoJson.h>

#include <DHT.h>

const int DHTPIN = A3;

const int DHTTYPE = DHT11;

const int JSON\_SIZE = 50;

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

dht.begin();

}

void loop() {

// Medir temperatura e umidade

float humidity = dht.readHumidity();

// Criar um objeto JSON

StaticJsonDocument<JSON\_SIZE> json;

json["umidade"] = humidity;

// Serializar o JSON e imprimir

serializeJson(json, Serial);

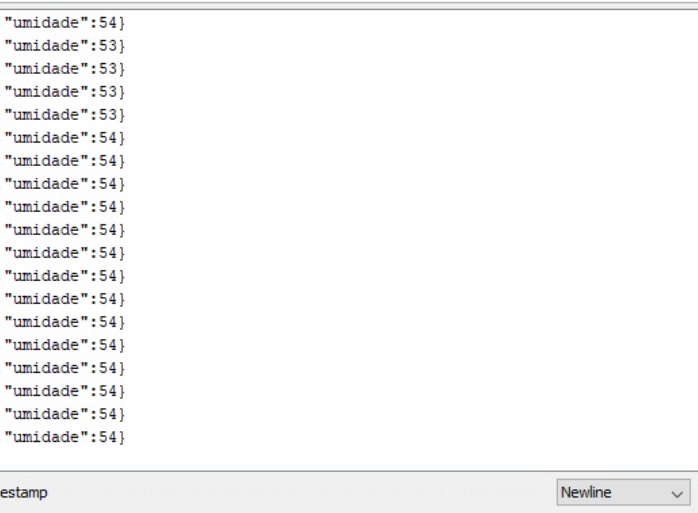
Serial.println();

// Aguardar 2 segundos antes da próxima leitura

delay(2000);

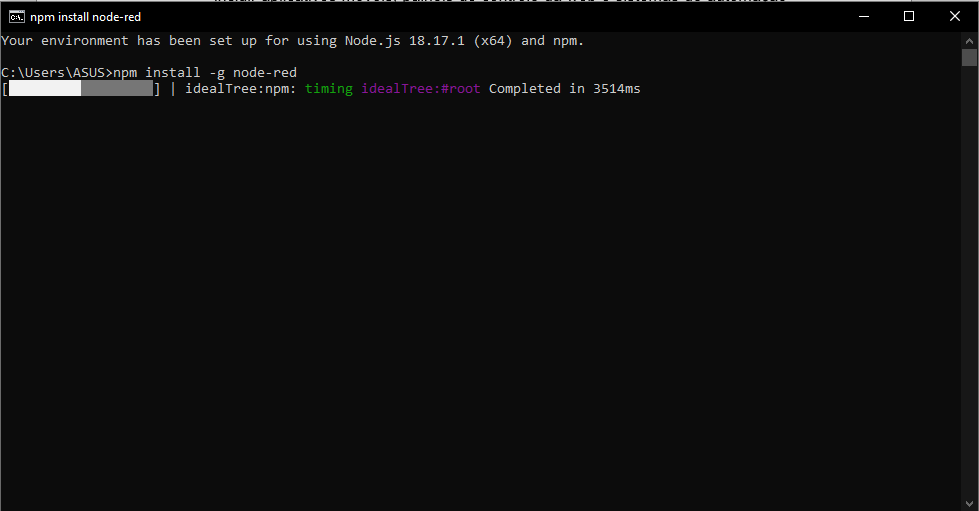
}

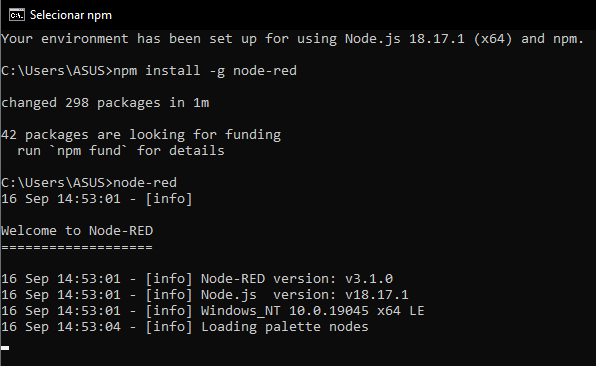
Após isso, os valores começaram a aparecer dentro do serial da seguinte maneira:



**Instalação do Node.js**

Para começar, devemos instalar o node.js. Dentro do prompt de comando dele, vamos pedir para que o node faça a instalação do node-red, e o acessamos através do comando “node-red”.

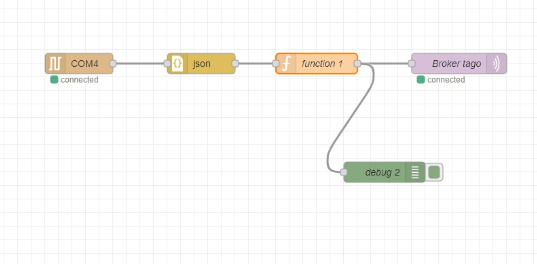




Depois disso, já dentro do site, acessamos o node-red através do http://localhost:1880 para começarmos a fazer o sistema que precisamos.

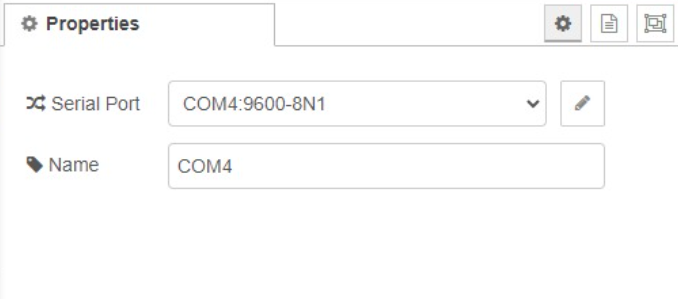
**Node-red**

Já dentro do Node-red, vamos começar a fazer o nosso fluxo. Para o nosso trabalho, criamos um fluxo com uma porta de entrada, json, uma function, uma saída para o Tago.io e um debug.

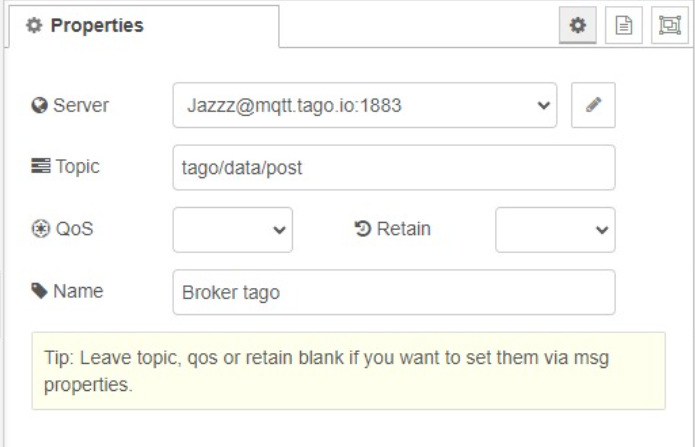


Dentro da porta de entrada, nós colocamos a COM4, que era a porta em que estava conectado o nosso arduino.

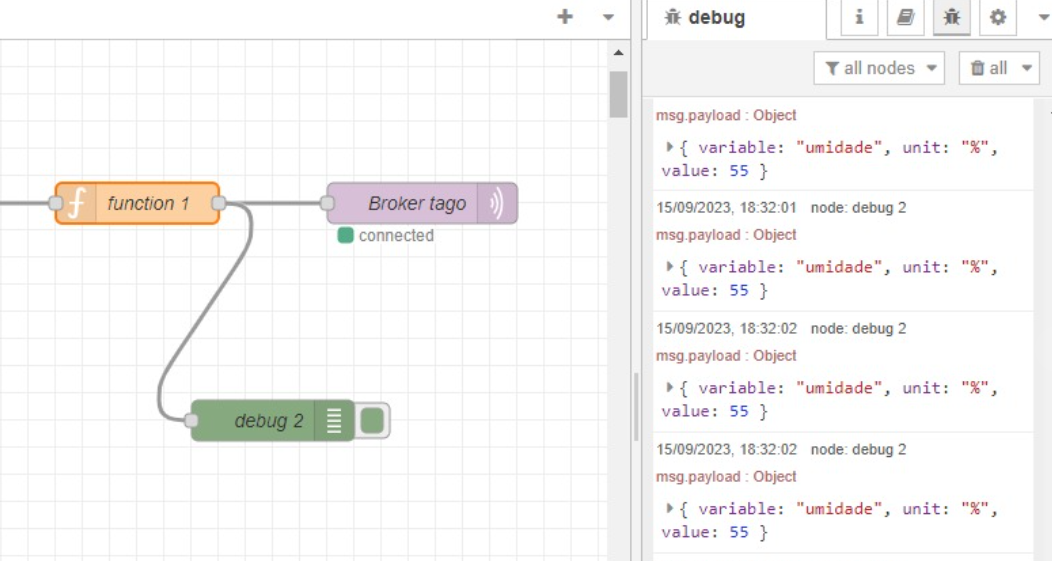
Já na function, nós criamos uma variável para umidade, que era o valor que nós precisávamos ler para o nosso projeto. O “msg.payload.umidade” foi importante dentro da função para acessar os valores lidos pelo arduino.

Após isso, conectamos dentro da saída o Tago.io, para que os valores que estavam sendo lidos através do arduino, fossem passado para o site. Ele foi conectado através do mqtt e do token que criamos.

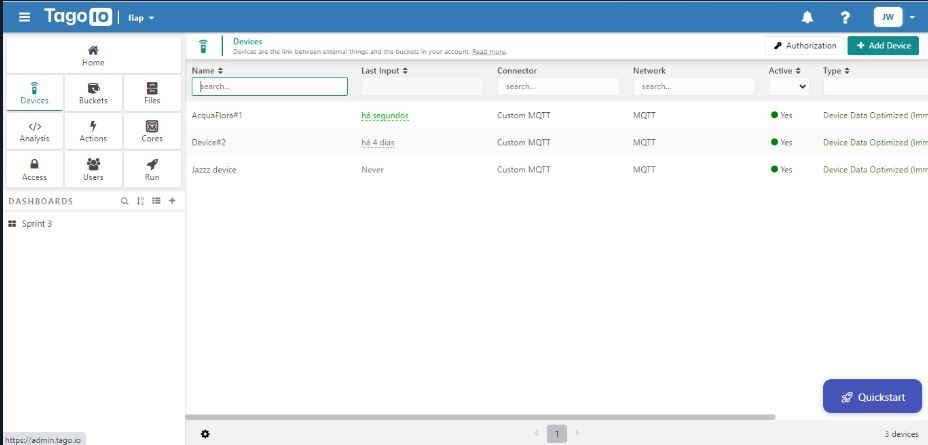


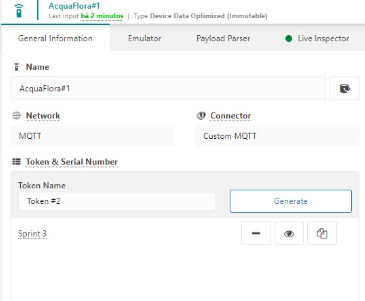
Após isso, as informações apareceram dentro do debug da seguinte forma:



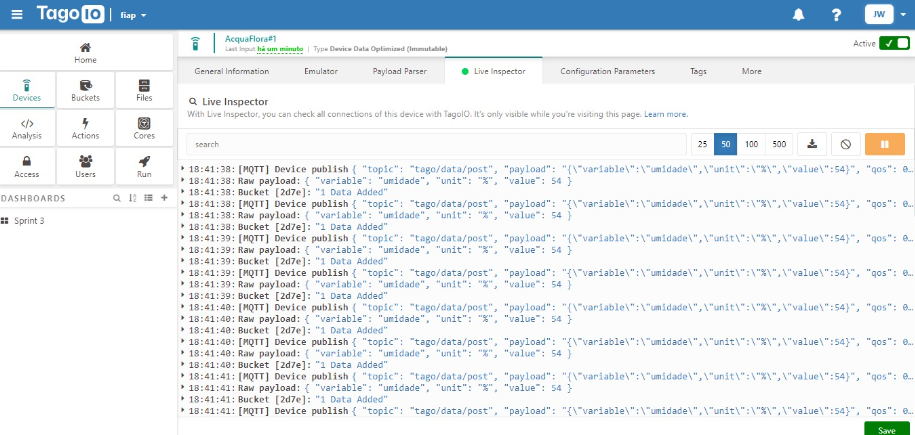
**Tago.io**

Já dentro do Tago.io, a primeira coisa que fizemos foi criar um device com o nome “AcquaFlora#1”. Esse device nos deu um token para que pudéssemos conectá-lo ao nosso node-red.

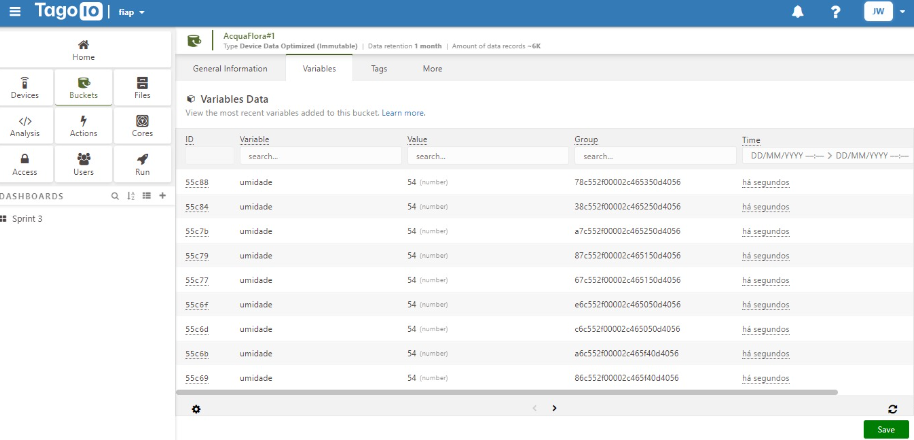




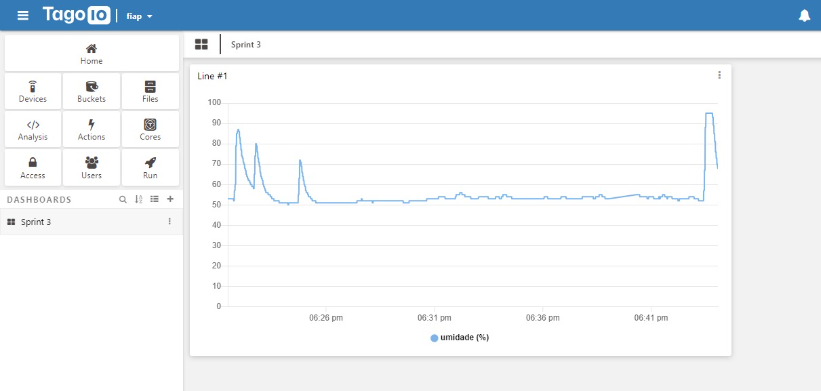
Depois de colarmos o token dentro do node-red, nós acessamos o live inspector para saber se os valores que estavam sendo lidos pelo arduino, estavam sendo passados para o Tago.



Acessamos então as variáveis no buckets para saber se estava também sendo lidas, e as salvamos.



Após isso, criamos um Dashboard com o nome Sprint3, e dentro do gráfico escolhido, colocamos o device e a variável que queríamos que fosse mostrada. O gráfico então, ficou da seguinte forma:



**Instrução de uso**

O nosso projeto foi pensado como um aspersor inteligente capaz de ler a quantidade de água que há no solo, sendo ligado apenas em situações onde o solo está seco.

Para o desenvolvimento do projeto, não tínhamos um sensor de umidade de solo disponível, por isso, fizemos a simulação utilizando o sensor DHT11 para ler os valores de umidade.

No caso, o gráfico nos mostra quando a quantidade de umidade está alta ou baixa. Quando ela estiver baixa, o arduino automaticamente irá iniciar, irrigando o solo. Já quando as quantidades de água estiverem altas, o aspersor não irá iniciar, fazendo com que haja controle da quantidade de água que está sendo utilizada e evitando desperdícios.

Também chegamos a pensar na possibilidade de medirmos a temperatura através do DHT11, mas achamos que não seria necessário. Por isso, incluímos dentro do nosso projeto, apenas os valores de umidade.