

MONITORAMENTO DE SENSORES DE UMIDADE DO SOLO COM INTERNET DAS COISAS (IOT) APLICADO NA AGRICULTURA DE PRECISÃO

Bruno Guilherme Lima Dias¹; Fernando de Almeida Martins³; Rogério Cassares Pires³;
Wânderson de Oliveira Assis²; Alessandra Dutra Coelho²

¹ Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professores da Escola de Engenharia Mauá (EEM/CEUN-IMT);

³ Engenheiros do Centro de Pesquisa da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo.

Este trabalho visa desenvolver um mecanismo para que agricultores e donos de grandes plantações possam monitorar a plantação nos parâmetros de quantidade de água no solo, temperatura e umidade ambiente, com a finalidade de uma melhor safra com menor desperdício de água tornando a plantação mais sustentável.

Para realizar esse monitoramento foi utilizado um microcontrolador, no qual foi conectado dois sensores, um para medir a temperatura e umidade ambiente e outro para medir a umidade do solo. Como a visualização desses dados deve ser de fácil entendimento, para isso foi necessário o desenvolvimento de um aplicativo que contém todas as informações coletadas pelos sensores. Esses dados devem ser manipulados e publicados de forma que sua leitura seja simples.

Esse monitoramento será diário e síncrono, condições do solo para que ocorra uma otimização baseada na condição de umidade daquela área no decorrer do dia. E a acessibilidade dos dados para o usuário será feita pensando no conceito de internet das coisas (Internet of Things (IoT)).

Introdução

De acordo com o relatório de Conjuntura dos Recursos Hídricos no Brasil (CRHB) da Agência Nacional de Águas (ANA) a atividade agrícola é responsável pelo consumo de 72% dos recursos hídricos, pois um dos aspectos mais importantes na agricultura para o bom desenvolvimento de plantas é a quantidade de água no solo, já que este não pode estar nem muito úmido e nem muito seco, disponível em (EOS, 2019).

A aplicação tem como objetivo monitorar a umidade do solo de uma forma periódica e diária, visando saber a condição atual do solo e se há a necessidade de se ligar o sistema de irrigação local.

A acessibilidade dessa informação se tornará algo prático com sua inserção no conceito de IoT, com isso a informação poderá ser adquirida e analisada uma vez que estará disponível tanto em uma página na web quanto por meio de um aplicativo de celular.

O projeto se pautou em dois estágios. O primeiro referente à transmissão dos dados, para o qual foi utilizado o LoRa que é um chip criado pela LoRa Alliance que utiliza a tecnologia de radiofrequência sem fio, semelhante ao Wifi e ao Bluetooth. O LoRa é um chipset sem fio de longo alcance e baixo consumo de energia usado em muitas aplicações envolvendo IoT. Porém é importante ressaltar que com o dispositivo LoRa não é possível transmitir voz ou vídeo e nem navegar na internet pois a finalidade de se usar o LoRa é transmitir dados.

Para se realizar essa tarefa o chip LoRa usa uma modulação complexa chamada de Chirp Spread Spectrum (CSS). Essa Modulação é uma técnica de aumento ou diminuição de frequência ao longo do tempo e extremamente resistente à interferência e a recuperação do sinal é muito eficiente ao ponto de se recuperar um sinal de 20dB abaixo do nível de ruído, detalhado em (ALMEIDA, 2019).

O segundo estágio refere-se à utilização de um Arduino UNO para fazer a leitura dos dados dos sensores visando uma calibração mais simples. O Arduino tem uma função que faz regra de três

entre os valores analógicos coletados e os valores em porcentagem, sendo assim esses valores serão comparados com os valores do sensor capacitivo com a finalidade de ajustá-lo. O sensor capacitivo é o que melhor se adequou ao nosso projeto.

Material e Métodos

Inicialmente, após adequações para a proposta do trabalho, foi gravado o programa desenvolvido pelo Centro de Pesquisa do Instituto Mauá de Tecnologia, foi utilizada a linguagem C no MPLab para o microcontrolador PIC24 ilustrado na figura 1. A programação faz com que após um ciclo de inércia da placa, gerado pelo processo conhecido como deep sleep, ela comece a transmitir os dados via LoRaWAN, que é quando o chip LoRa envia os dados via padrão *Low Power Wide Area Network* (LPWAN). Os esquemas elétricos da placa utilizada são retratados na figura 2, 3 e 4 retiradas de (IMT, 2019). Os dados que estão sendo transmitidos são provenientes do sensor DHT22 que mede a temperatura e umidade do ambiente, e do sensor capacitivo desenvolvido pelo centro de pesquisa da Mauá que nos fornece a umidade do solo a partir da transcondutância. Ambos os sensores são mostrados nas figuras 5 e 6.

Vale ressaltar que o LoRa com a eficiência energética decorrente do *deep sleep* faz com que a placa permaneça por oito minutos consumindo $4\mu\text{A}$, depois entra em funcionamento por cinco segundos consumindo 14mA e depois volta para a o *deep sleep* consumindo os valores descritos anteriormente. Esse processo funciona devido ao *software* e *firmware* na placa desenvolvido no Smart Campus junto com o *hardware*. Mesmo com esse baixo consumo de energia o alcance em área rural chega a até 15 quilômetro, isso usando uma potência muito baixa, da ordem de 20dbm ou 100Mw , sendo as informações sobre o *deep sleep* coletadas em (MIRKAI, 2018).

Figura 1 – PIC 24FJ128GA306.

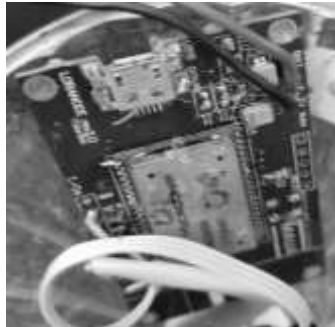


Figura 2 – Esquema elétrico da CPU do circuito.

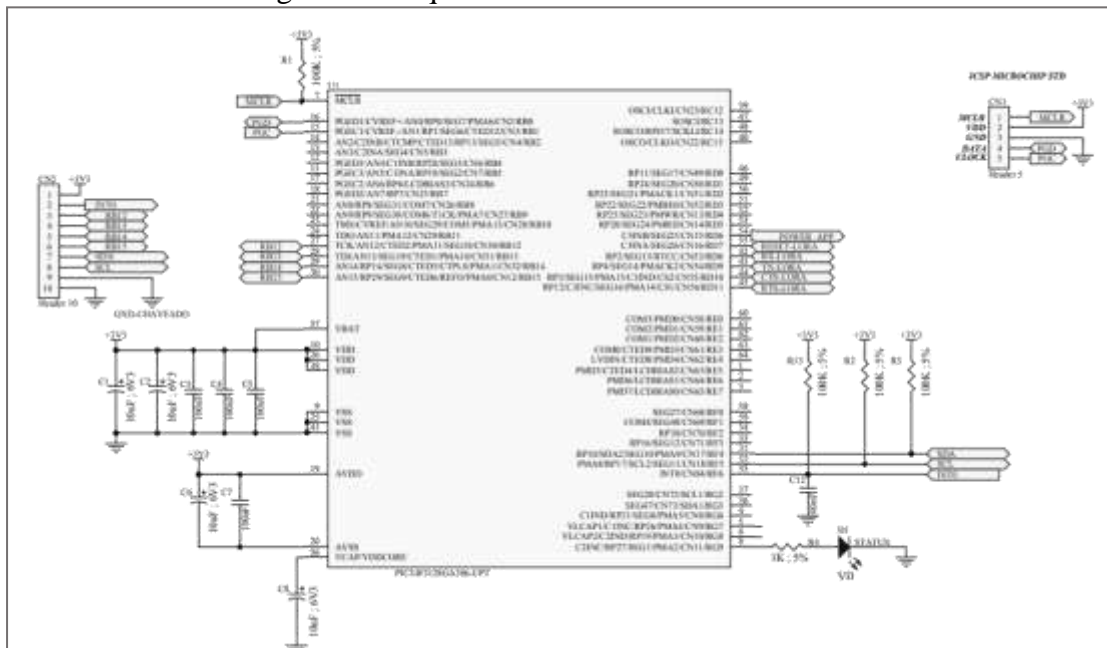


Figura 3 – Esquema elétrico da fonte de alimentação.

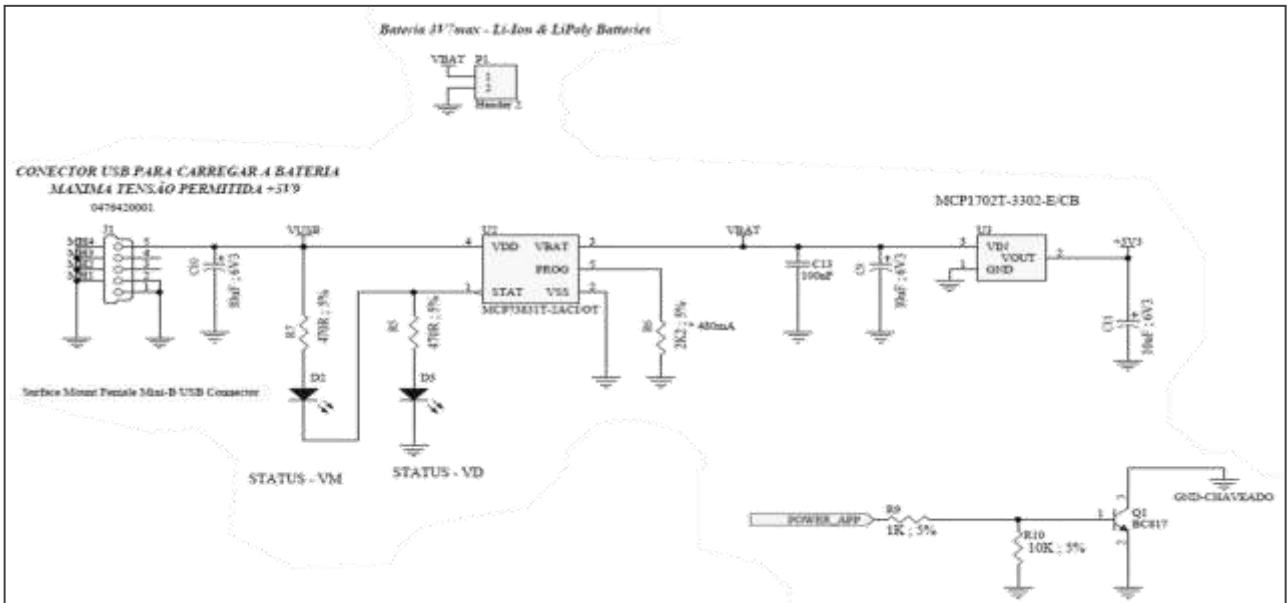


Figura 4 – Esquema elétrico da conexão do módulo LoRa.

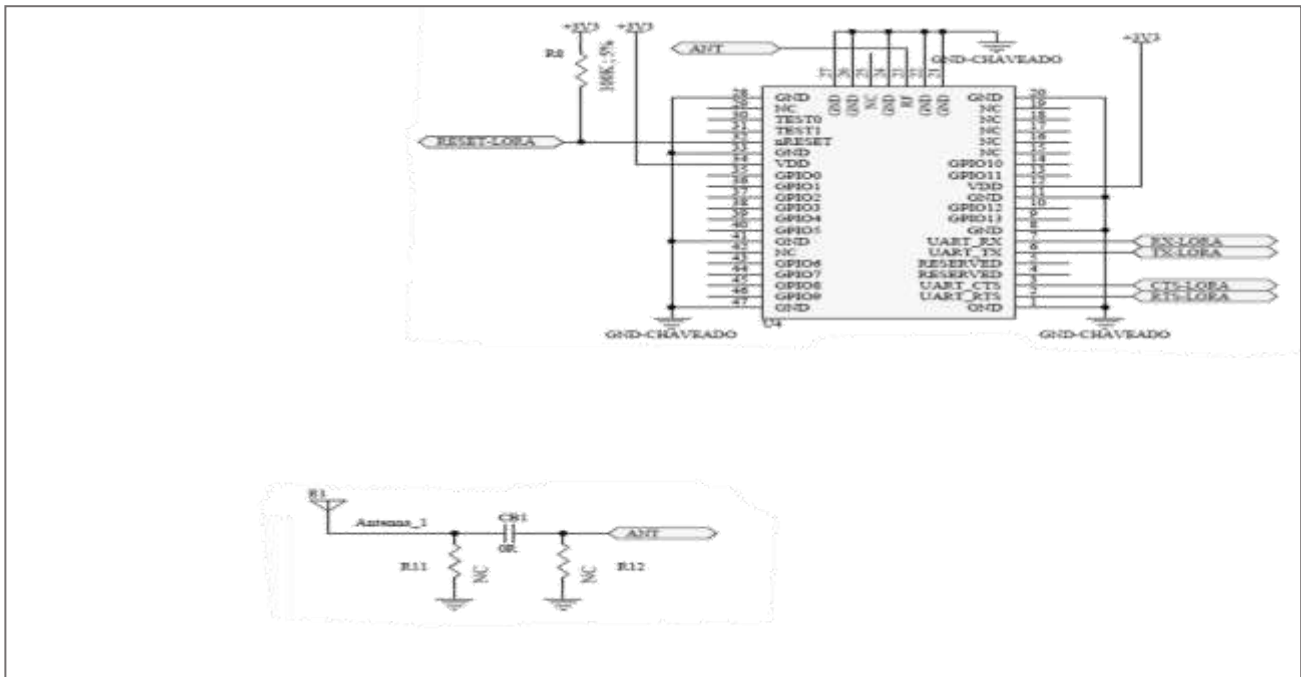
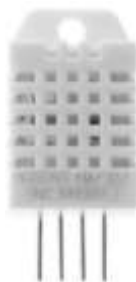


Figura 5 – Sensor Capacitivo.



Figura 6 – Sensor de umidade e temperatura ambiente DHT22.



Os dados coletados pelos sensores e enviados pela placa LoRa são recebidos pelo *gateway*, sendo este um tipo de antena que recebe e coleta dados e está localizado nas dependências da Mauá, e posteriormente são salvos em um servidor. Estes dados foram transmitidos no formato *Message Queuing Telemetry Transport* (MQTT), que é um protocolo para transmitir mensagens leves para sensores e/ou para dispositivos móveis utilizando o sistema de publicador e subscritor.

As informações enviadas ao *Network Server*, o qual contém um MQTT *broker* responsável por gerenciar as publicações e as subscrições do protocolo, podem ser visualizadas no *dashboard* do site do Smart Campus Mauá que possui disponível orientações para tal procedimento na janela Documentação e Aplicação. A *dashboard* do site do Smart Campus serve como um painel de informações para mostrar todos os dados que chegam no *gateway* da Mauá e que estão sendo armazenados no *Network Server*.

A partir deste ponto, os dados são disponibilizados tanto para o site do ThingSpeak quanto para um servidor público, mqtt.eclipse.org, utilizando um sistema de tópicos para organizar essas informações. Essa parte do processo foi feita com o Node-RED, que é uma ferramenta voltada para aplicações relacionadas ao IoT que utiliza abordagem gráfica de ambiente de código aberto, ou seja, o Node-RED para elaborar uma aplicação ele faz uma conexão entre blocos que possuem código predefinidos para a realização de tarefas. Foram seguidas as orientações de instalação do Node-RED disponível em (Madeira, 2019) e o método de como utilizar tal ferramenta disponível em (IMT, 2019). O objetivo deste trabalho é distribuir os dados tanto para a página na *web* do ThingSpeak quanto para o servidor público além de criar uma janela de apresentação no *dashboard* do site do Smart Campus Mauá.

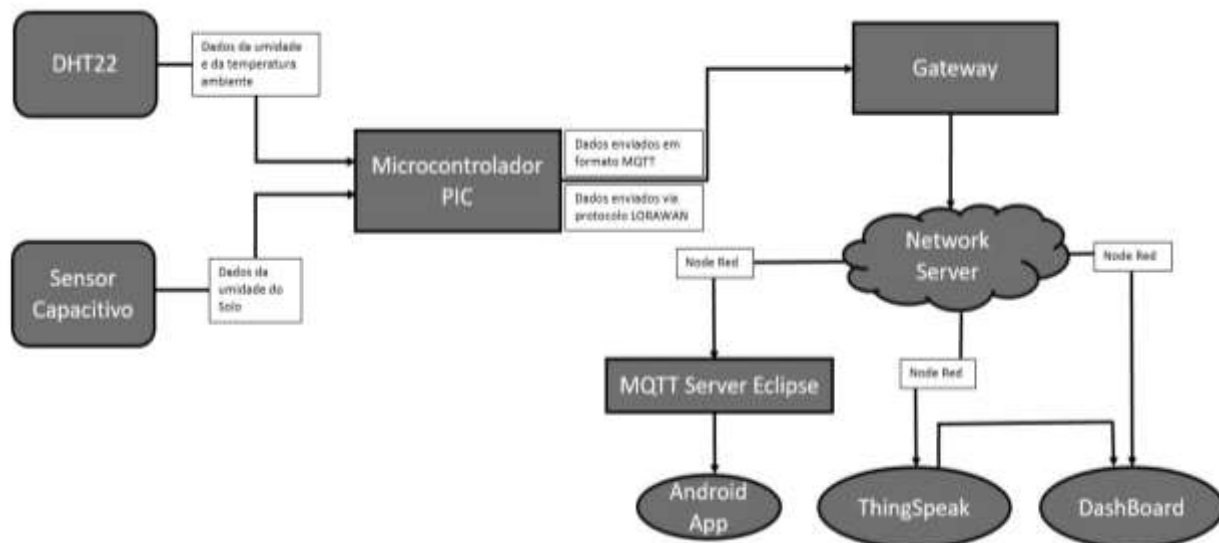
No ThingSpeak os dados são plotados simultaneamente em uma área pública de visualização. Nela são criados quatro gráficos, sendo um para a umidade do solo, um para a umidade ambiente, um para a temperatura ambiente e um para o status da bateria. Esses pontos são plotados e salvos para possibilitar qualquer análise posterior.

Em seguida no Android MIT App Inventor os gráficos gerados pelo ThingSpeak são copiados via *HyperText Markup Language* (HTML), porém essa visualização não mostra claramente o status do solo pois para isso seria necessário a leitura via MQTT, sendo que essa ferramenta no Android MIT App Inventor não está bem desenvolvida. Por conta dessa dificuldade ele foi descartado e um outro mecanismo foi estudado para se ter a visualização da informação no celular.

A solução encontrada foi a utilização de um aplicativo que tivesse um leitor de dados MQTT e assim pudesse gerar uma espécie de *dashboard* própria, personalizada e de fácil utilização para o usuário. Os aplicativos investigados foram: MQTT Dash (IoT Smart Home); MQTT Dashboard – IoT and Node-RED controller; IoT MQTT Dashboard; IoT MQTT Panel.

Esses aplicativos funcionam da seguinte forma: os dados que estão no servidor público, mqtt.eclipse.org, são acessados pelo aplicativo por meio de uma organização pelo sistema de tópico, assim é selecionar e coletar somente os dados desejados dentro do servidor público. Desta forma é possível se separar as informações e criar a sua própria *dashboard* em forma de aplicativo. O caminho dos dados pode ser compreendido na figura 7.

Figura 7 – Fluxograma da rota dos dados.



O próximo passo foi a calibração do sensor capacitivo. Nesta etapa utilizou-se um Arduino UNO junto com um kit do módulo de sensor higrômetro de umidade de solo, demonstrados nas figuras 8 e 9.

O sensor funciona da seguinte forma, quanto mais seco está a terra que o sensor está medindo, maior é a resistência medida entre os dois terminais de referência e assim será dado um valor decimal maior. Por exemplo, se deixar o sensor totalmente seco, a umidade igual a 0%. O valor lido pelo Arduino mostrado no monitor serial é 1024, sendo que essa é a maior resolução possível em um Arduino, em contra partida, para 100% de umidade o valor é de 442. O programa para se obter as medições é apresentado na figura 10.

Figura 8 – Arduino UNO.



Figura 9 – Sensor Hidrômetro.

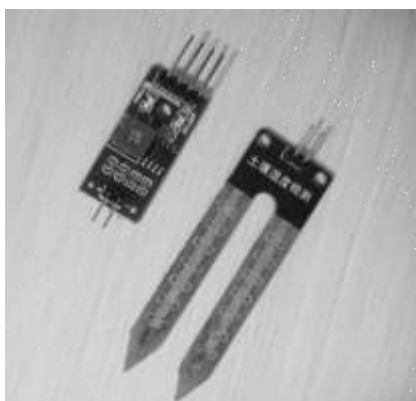


Figura 10 – Programa do sensor de umidade no Arduino.

```

sketch_jun26SU $
#define entradaAnalogica 0

int aSensor;
int sensorValue;

void setup() {
  Serial.begin(9600);
}

void loop() {
  aSensor = analogRead(entradaAnalogica);
  Serial.print("Valor da umidade: ");
  sensorValue = map(aSensor, 442, 1020, 100, 0);
  Serial.println(sensorValue);
  Serial.print(aSensor);
  Serial.println();
  delay(1000);
}

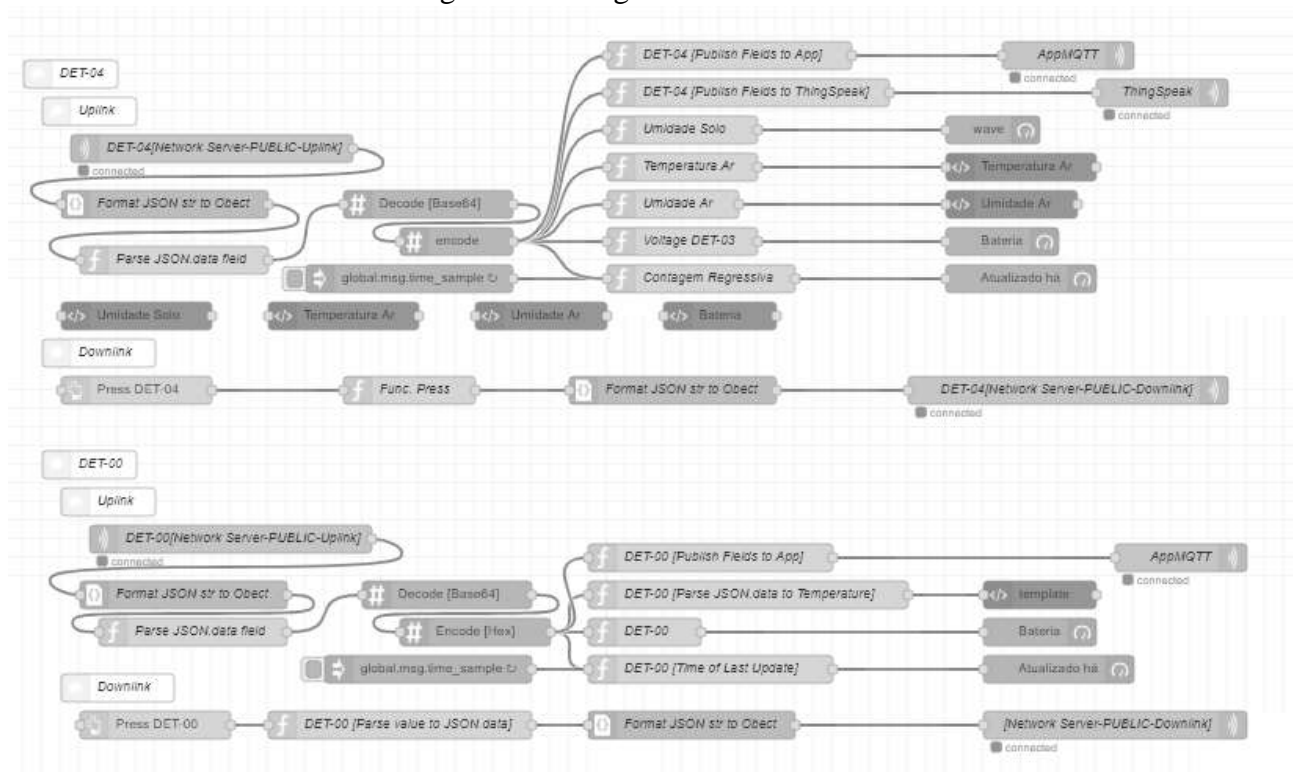
```

No sensor capacitivo, após os dados coletados passarem por um processo de codificação e decodificação no Node-RED, representado na figura 11, eles são expressos em hexadecimal na *dashboard*. Para se fazer a calibração é necessário fazer uma interpolação dos valores dos sensores, para isso foi utilizado o Excel.

Os sensores foram expostos ao ar e mergulhados em água para se obter a leitura de 0% a 100%. Em seguida ambos são colocados próximos no solo a fim de se obter a mesma leitura, o capacitivo gerando valores em hexadecimal e o sensor hidrômetro gerando valores em porcentagem.

Esses valores são plotados no Excel e interpolados, com o objetivo de se obter um gráfico. Os valores em hexadecimal foram convertidos para decimal e a partir disso é criada uma reta média dos pontos coletados.

Figura 11 – Programa no Node-RED



A equação da reta é inserida no programa do Node-RED e os valores coletados pelo sensor capacitivo são convertidos de hexadecimal para porcentagem em decimal, publicando assim a umidade nas *dashboards*.

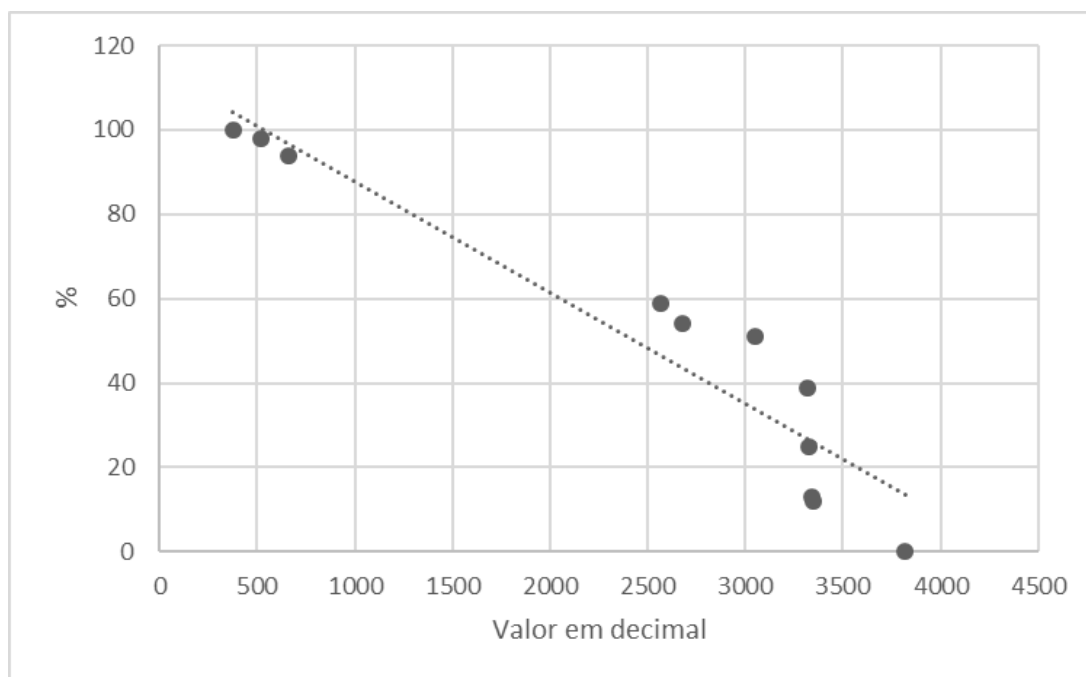
Resultados e Discussão

A partir dos valores coletados com os dois sensores de umidade do solo foi possível construir a Tabela 1 e assim fazer o levantamento da variação da umidade ao longo de algumas medições, obtendo-se o gráfico da figura 12.

Tabela 1 – Valores experimentais da leitura dos sensores.

Valor Sensor Hexadecimal	Valor Sensor Decimal	Umidade (%)
0eec	3820	0
0a05	2565	59
0a76	2678	54
0d12	3346	12
0d0e	3342	13
0cfe	3326	25
0cfa	3322	39
0bed	3053	51
207	519	98
293	659	94
179	377	100

Figura 12 – Valores do sensor capacitivo em função do sensor no Arduino.



Com essas informações foi possível se obter a reta de carga do sensor capacitivo e assim calibrá-lo. A reta linear que passa pela média dos pontos é $-0.0264x + 114.31$, ela foi inserida no Node-RED para que os dados disponibilizados nas *dashboard* já apareçam em porcentagem.

Como mencionado anteriormente a interface obtida no Android MIT App Inventor, figuras 13 e 14, não foi satisfatória, desta forma houve a necessidade de uma nova abordagem para a visualização das informações no celular.

Após a análise dos aplicativos listados no tópico anterior determinou-se que o que melhor se adequa com o projeto é o IoT MQTT Panel, ele possui mais ferramentas e é melhor estruturado em comparação com outros como o MQTT Dash (IoT Smart Home) que ainda está em fase de desenvolvimento. Assim o resultado obtido pode ser visualizado na figura 15.

Figura 13 – Interface de programação do Android MIT App Inventor.



Figura 14 – Interface visível ao usuário do Android MIT App Inventor.

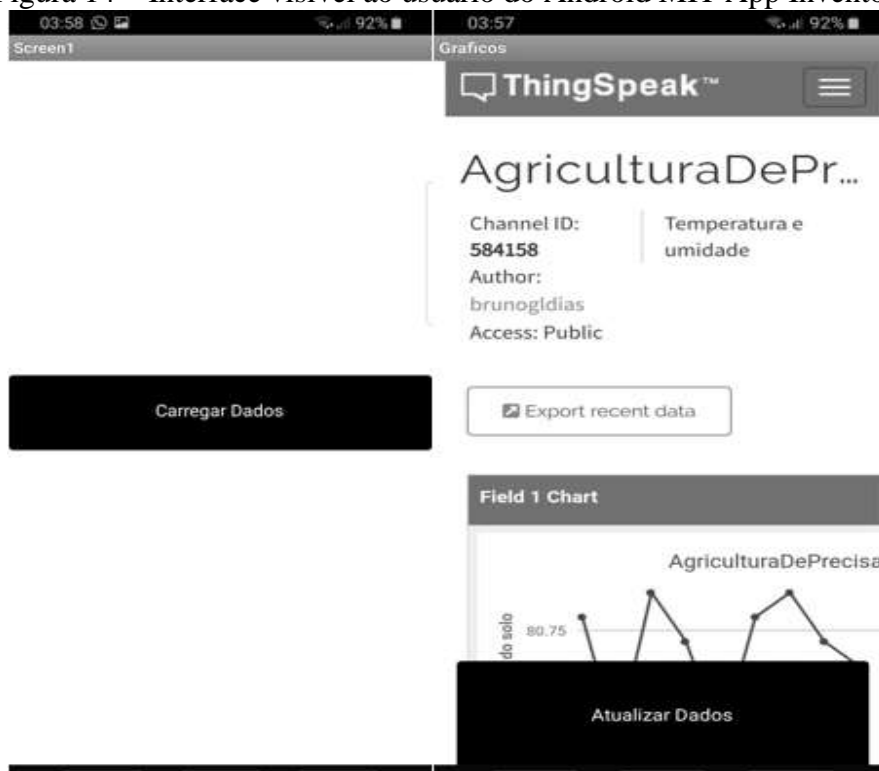
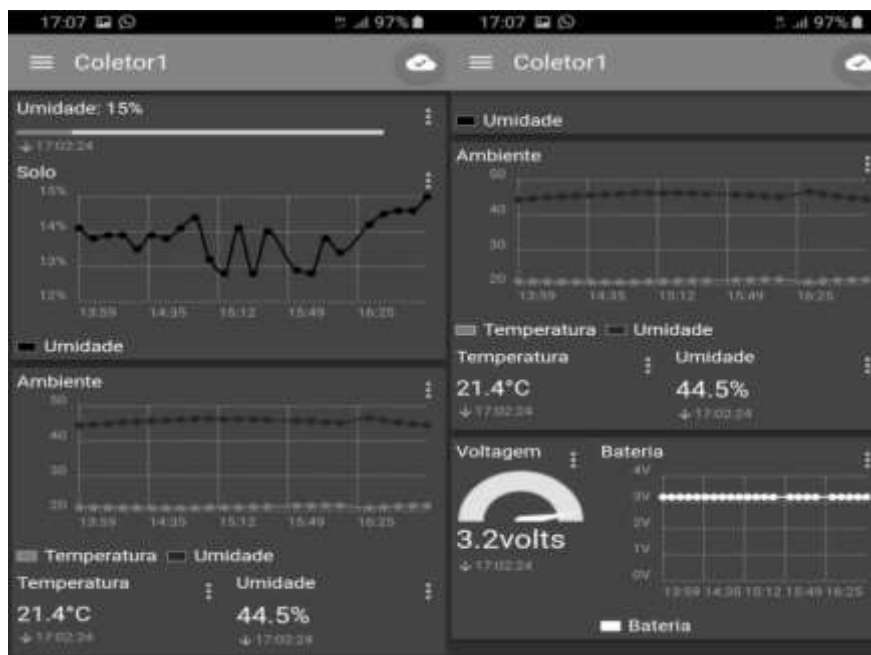


Figura 15 – Interface para o usuário do IoT MQTT Panel



A seguir temos o layout do *dashboard* gerado pelo Node-RED e os gráficos gerados pelo ThingSpeak, exibidos nas figuras 16 e 18, sendo que os gráficos presentes na *dashboard* são os gerados pelo ThingSpeak. A figura 17 mostra a instalação das placas LoRa e dos sensores no solo.

Figura 16 – Dashboard

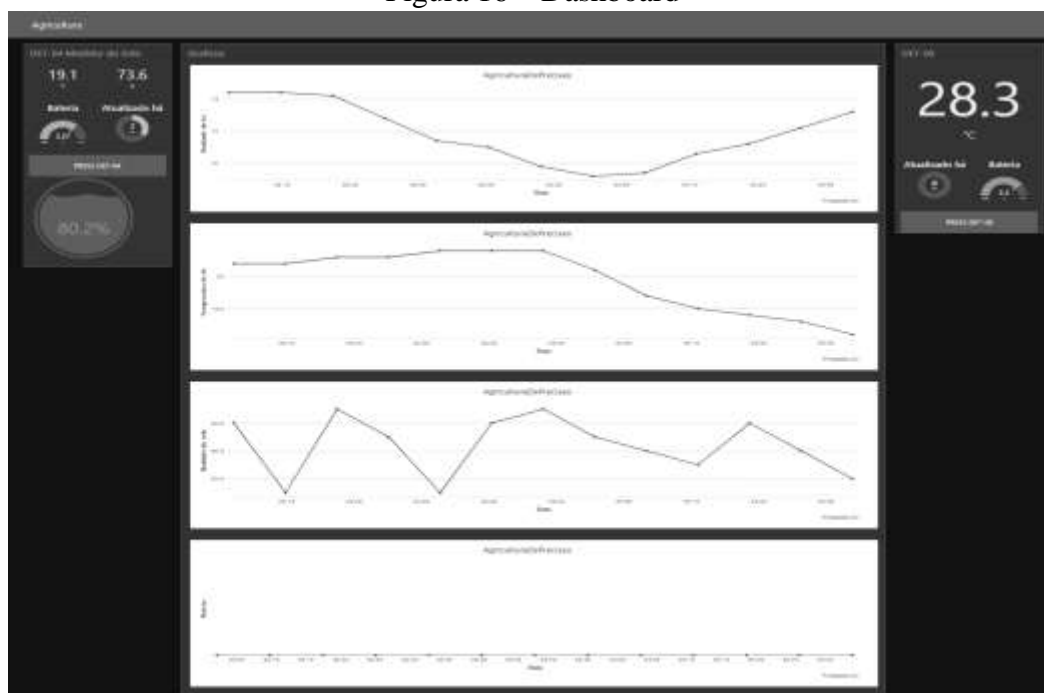


Figura 17 – Sensores no solo.



Figura 18 – Gráficos gerados pelo ThingSpeak



Conclusões

A premissa deste trabalho é se obter um ou mais mecanismos para que se possa visualizar os dados que estão sendo coletados pelos sensores descritos neste artigo de tal forma que o usuário possa retirar as informações necessárias para se compreender as condições da área do plantio.

A partir da substituição da forma de se criar a interface para apresentar os dados coletados para o usuário, troca do App Inventor pelo IoT MQTT Panel, a aplicação se tornou viável pois ficou muito mais fácil de se entender a atual condição do solo.

Para um próximo estágio de desenvolvimento seria interessante a construção de um sensor de três níveis para realizar a detecção da umidade de três camadas diferentes do solo para assim conseguir um melhor espectro daquela área e com isso poder regular não só quando irrigar mais sim o método de irrigação. Desta forma, para cada camada se poderia ajustar um tipo de irrigação como exemplo na camada mais superficial um simples spray e na camada mais funda um jato concentrado possibilitando uma redução ainda maior no consumo de água para o agricultor.

Referências Bibliográficas

- ALMEIDA, Orlan. Tecnologia LoRa: O que é, distância e teste prático. **easyIOT**, 2019. Disponível em: <<https://www.easyiot.com.br/tecnologia-lora/>>. Acesso em: 20 de ago. de 2019.
- EOS. A situação do consumo e desperdício de água no Brasil. **EOS organização e sistemas**, 2017. Disponível em: <<https://www.eosconsultores.com.br/consumo-e-desperdicio-de-agua/>>. Acesso em: 20 de ago. de 2019.
- IMT. Documentação de Aplicação. **Smart Campus Maua**, 2019. Disponível em: <https://smartcampus.maua.br/?page_id=55>. Acesso em: 20 de ago. de 2019.
- MADEIRA, Daniel. Primeiros passos com o Node-RED e Arduino UNO. **Filipeflop**, 2019. Disponível em: <<https://www.filipeflop.com/blog/primeiros-passos-node-red-arduino-uno/>>. Acesso em: 20 de ago. de 2019.
- MIRKAI, Beatriz. Aplicações de Automação Voltadas para o Rastreamento de Animais Domésticos. **Instituto Mauá de Tecnologia**, 2018. Disponível em: <<https://maua.br/files/122018/aplicacoes-automacao-voltadas-para-rastreamento-animais-domesticos-281632.pdf>>. Acesso em: 20 de ago. de 2019.

Agradecimento

O projeto teve suporte da divisão de eletrônica do Centro de Pesquisas do Instituto Mauá de Tecnologia.