



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



SISTEMA DE IRRIGAÇÃO INTELIGENTE COM SENSOR DE UMIDADE.

Meuriam Silva de Assis, Thiago Siqueira Garbuio, Wilian França Costa

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

31856411@mackenzista.com.br, 31808451@mackenzista.com.br,
wilian.costa@mackenzie.br

Abstract. *This project presents a proposal for the development of an intelligent low-cost acquisition prototype (developed on the Arduino platform) for monitoring and automatic control of soil irrigation, with remote activation via application. The work consisted in the construction of a physical device (hardware), software using the MQTT protocol. Automation equipment and remote communication instruments were used. The system also has a set of algorithms that will allow the best use of water resources based on information collected from sensors inserted in the environment that will be subject to irrigation.*

Resumo. *Este projeto apresenta uma proposta de desenvolvimento de um protótipo inteligente de baixo custo de aquisição (desenvolvido na plataforma Arduino) para monitoramento e controle automático da irrigação do solo, com acionamento remoto via aplicativo. O trabalho consistiu na construção de dispositivo físico (hardware), software utilizando o protocolo MQTT. Utilizou-se de código aberto (open source), de equipamentos de automação e de instrumentos de comunicação remota. O sistema também conta com um conjunto de algoritmos que permitirá o melhor uso dos recursos hídricos a partir de informações coletadas de sensores inseridos no ambiente que estará sujeito à irrigação.*

1. Introdução

Por conta do isolamento social e das medidas de quarentena na pandemia, de acordo com a ferramenta Google Trends (TRENDS, G. 2021)[1], a procura por plantas para ambientes internos apresentou aumento de mais de 100% no número de pesquisas na plataforma. Com a nova rotina da quarentena, muitos tem procurado se dedicar à alimentação orgânica e saudável ou apenas cultivo de certas espécies, e para tanto, sem a disponibilidade de sempre comparecer ao mercado, a onda das hortas em casa e cultivo de plantas internas em geral só aumentou. Contudo, o cuidado necessário demandado pelas plantas é algo que nem sempre pode ser provido, seja pela falta de falta de tempo, seja conhecimento das condições essenciais para a sobrevivência das plantas, como a irrigação. Portanto, esse projeto apresenta um protótipo de como a manutenção da irrigação dessas plantas pode ser feita através de um monitoramento

remoto ou não, para que a umidade do solo esteja em condições favoráveis para a saúde da flora desejada. Tendo como objetivo principal indicar, através do CloudMQTT (Cloudmqtt.com, 2021)[4], os resultados obtidos acerca da umidade captados por sensores no solo. Em outras palavras, o foco é mostrar ao usuário quando a planta essa devidamente regada ou não, para que assim o sistema automaticamente a regue quando necessário.

2. Materiais e Métodos

- NodeMCU ESP8266

Figura 1 – Módulo Wifi NodeMCU ESP8266

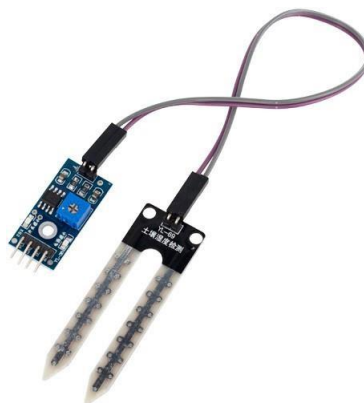


Fonte: <https://produto.mercadolivre.com.br/MLB-1215585235-modulo-esp8266-nodemcu-v3-wifi-80211-arduino-lua>

- a. Módulo WI-FI NodeMCU ESP8266, opção escolhida dentre várias, como a placa de Arduino convencional. Isso porque, com seu chip ESP8266 em conjunto com a interface usb-serial e regulador de tensão de 3.3V é programado na interface do Software “Arduino IDE”, onde recebe, por sua vez, as instruções necessárias de comando, além de permitir conexão com a internet para monitoramento da irrigação das plantas.

- Sensor de Umidade

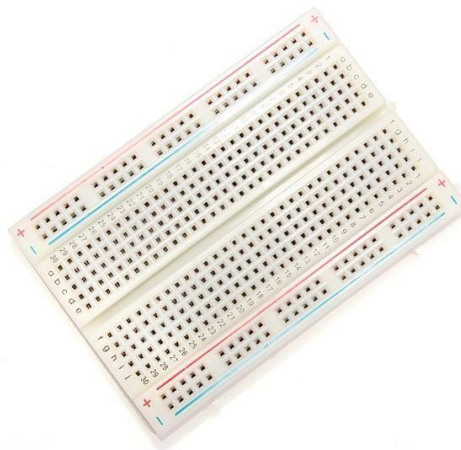
Figura 2 - Sensor de Umidade do Solo Higrômetro



Fonte: <https://produto.mercadolivre.com.br/MLB-766348495-modulo-e-sensor-de-umidade-do-solo-para-arduino- JM>

- b. O Sensor de Umidade de Solo para Arduino, utilizado no protótipo com o objetivo de captar as condições do solo e suas variações. Sem esse elemento não seria possível enviar o parâmetro ao módulo WiFi, que por sua vez, ao receber a informação do estado do solo através do sensor, envia a resposta de que a condição da terra está ruim e executa a irrigação, ou que a sua condição está aceitável e não executar a irrigação. O Sensor de Umidade do Solo Higrômetro é utilizado para identificar a condição do solo, utilizamos ela em nosso projeto para detectar se o solo está ou não na condição recomendada para o tipo de planta. O sensor é formado por duas hastes de eletrodos que conduz uma corrente pelo solo, sendo possível identificar a qualidade. A outra parte do sensor é onde escolhemos se queremos utilizar a saída digital (D0) ou a saída analógica (A0) para identificar qual o tipo de solo ideal para a planta utilizada.
- Protoboard 400 Pontos

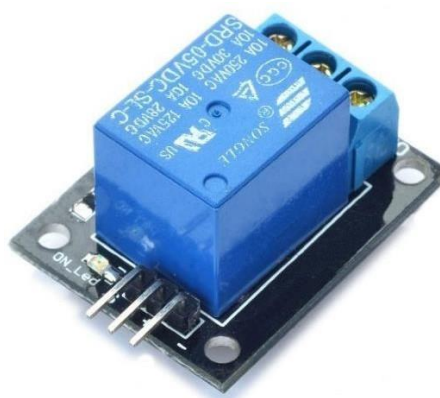
Figura 3 – Protoboard para montagem de circuitos eletrônicos



Fonte: <https://produto.mercadolivre.com.br/MLB-933371332-protoboard-breadboard-400-pontos-furos-pinos- JM>

- c. Protoboard 400 pontos, utilizado para ligar os componentes dos circuitos eletrônicos com a definição de entradas de cada ferramenta para o seu devido e adequado funcionamento.
- Módulo Relé

Figura 4 – Módulo Relé 5V 1 Canal



Fonte: <https://produto.mercadolivre.com.br/MLB-807436295-modulo-rele-1-canal-5v10a- JM>

- d. O Módulo Relé 5V de 1 Canal, funciona como um interruptor, sendo usado como controlador de dispositivos, em nosso caso, ele irá ligar e desligar a Mini Bomba de Água.

- Jumpers

Figura 5 – Kit de Jumpers



Fonte: <https://produto.mercadolivre.com.br/MLB-807436295-modulo-rele-1-canal-5v10a- JM>

- e. Jumpers, para realizar a ligação eletrônica de cada um dos componentes, do protoboard ao relé, à bomba, ao Módulo WiFi NodeMCU, e que define as portas e entradas que serão determinadas em cada uma das ligações.

- USB

Figura 6 – Cabo Micro USB



Fonte: <https://produto.mercadolivre.com.br/MLB-1178626841-cabo-usb-para-micro-usb-15-m-preto- JM>

- f. Cabos USB com conectores AM para Micro do tipo macho, utilizados para fazer a ligação do Módulo WiFi NodeMCU e a bomba à energia com a voltagem de 5V.

- Mini Bomba de Água

Figura 7 – Mini Bomba de água submersível vertical



Fonte: https://produto.mercadolivre.com.br/MLB-1179542863-mini-bomba-agua-sapo-submersa-jt100-para-5v-dc-_JM

- g. A Mini Bomba de água é utilizada para enviar o fluxo de água para uma mangueira fazendo a irrigação do solo quando for detectado que a condição do solo não é a ideal no momento. Os seus fios são ligados no Relé que controla o acionamento da bomba e outro é alimentado na energia 5V.

- Mangueira Aquário

Figura 8 – Mangueira



Fonte: <https://www.americanas.com.br/produto/96262939?OPN=>

- h. 1,5 Metros de mangueira para Aquário, utilizada como meio de transporte para a irrigação com água.

- LEDs

- **Figura 9 – LED**

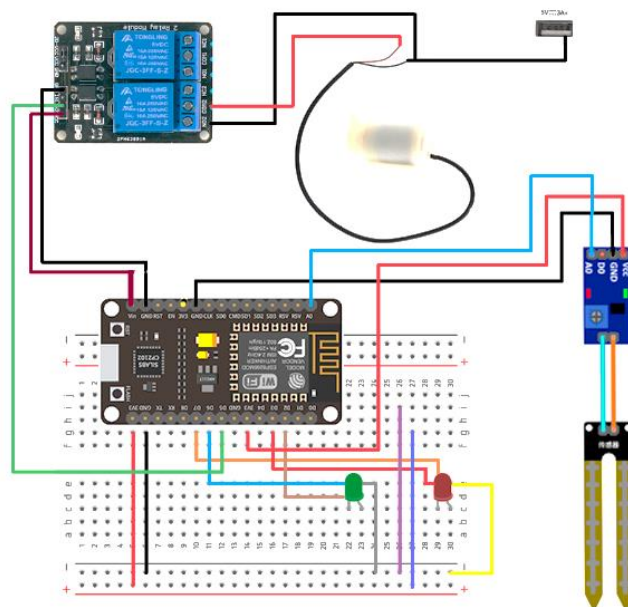


Fonte: <https://produto.mercadolivre.com.br/MLB-1483951901-100-leds-de-3mm-diversas-cores-difuso-20un-cada-para-arduino- JM>

- i. 2 LEDs, utilizados para demonstrar a condição do solo, sendo: vermelho (condição ruim) e verde (condição boa).

Na montagem utilizamos: Protoboard de 400 Pontos, NodeMCU ESP8266, Sensor de Umidade de Solo, Módulo Relé, Bomba de Água ligado a mangueira e ao cabo USB e Jumpers.

Figura 10 – Protótipo



Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/Imagens/ProjetoCircuito.png>

Figura 11 – Configurações No App MQTTDash

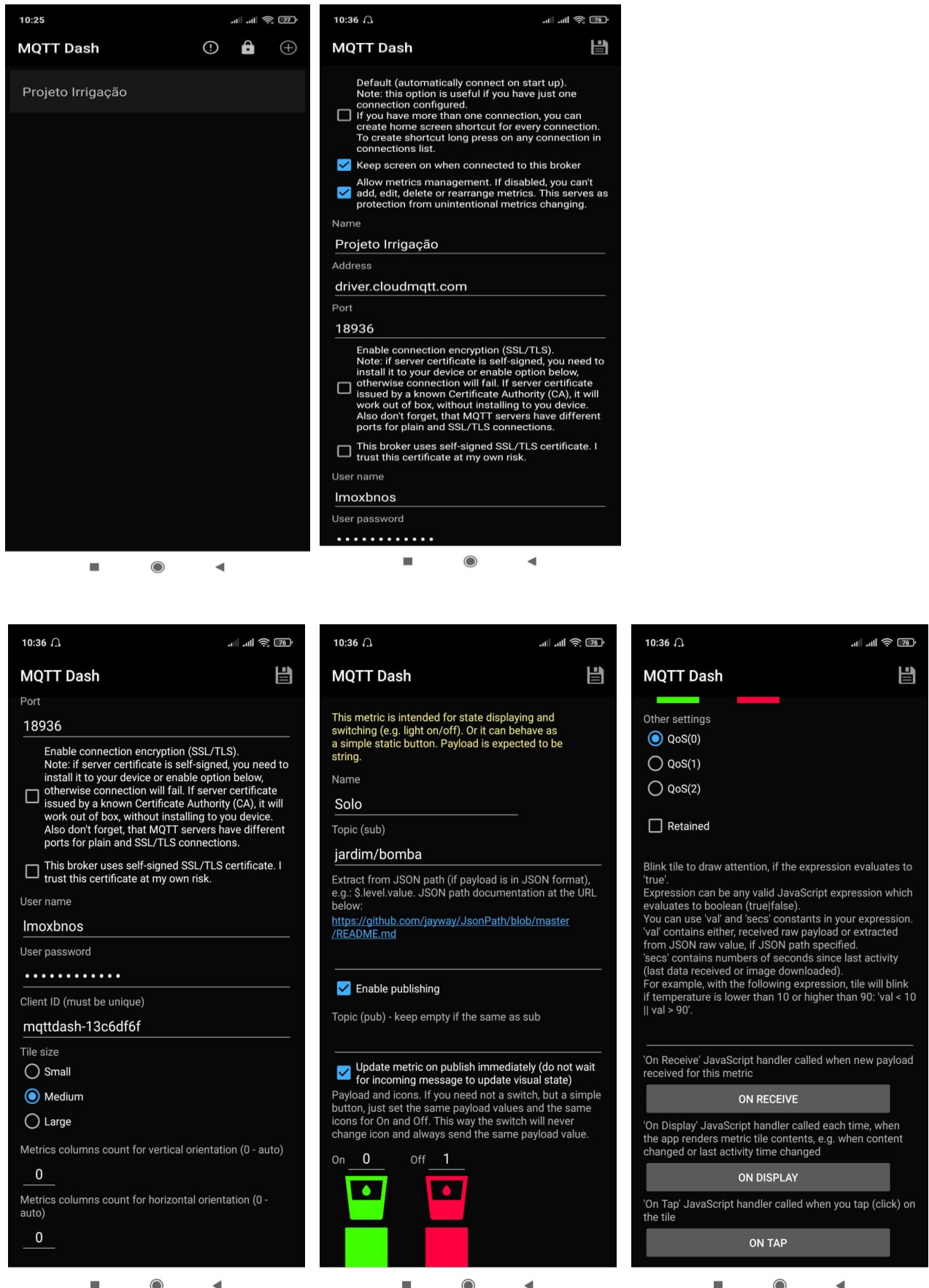
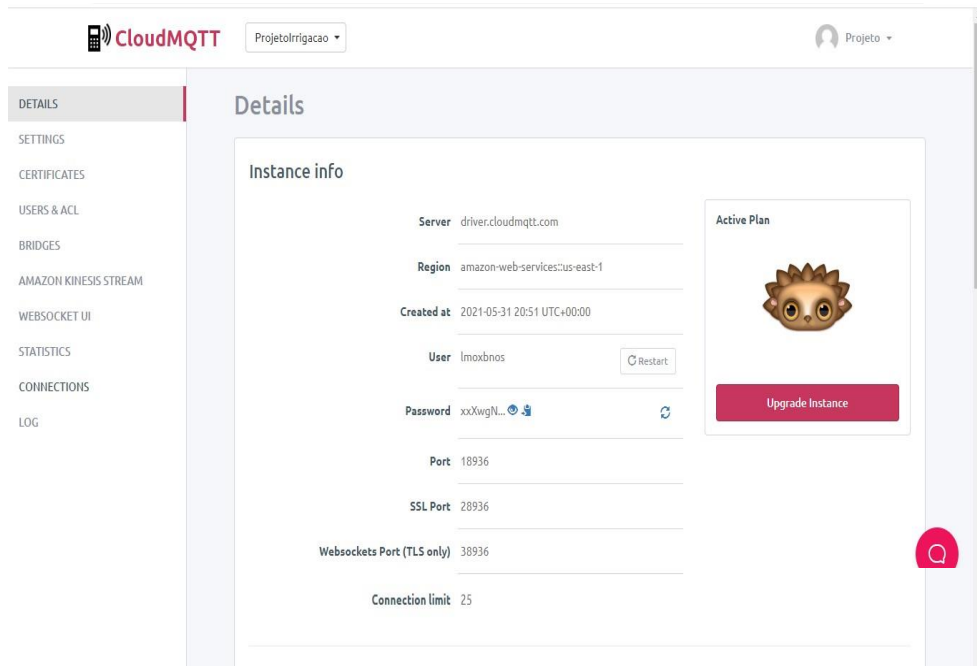
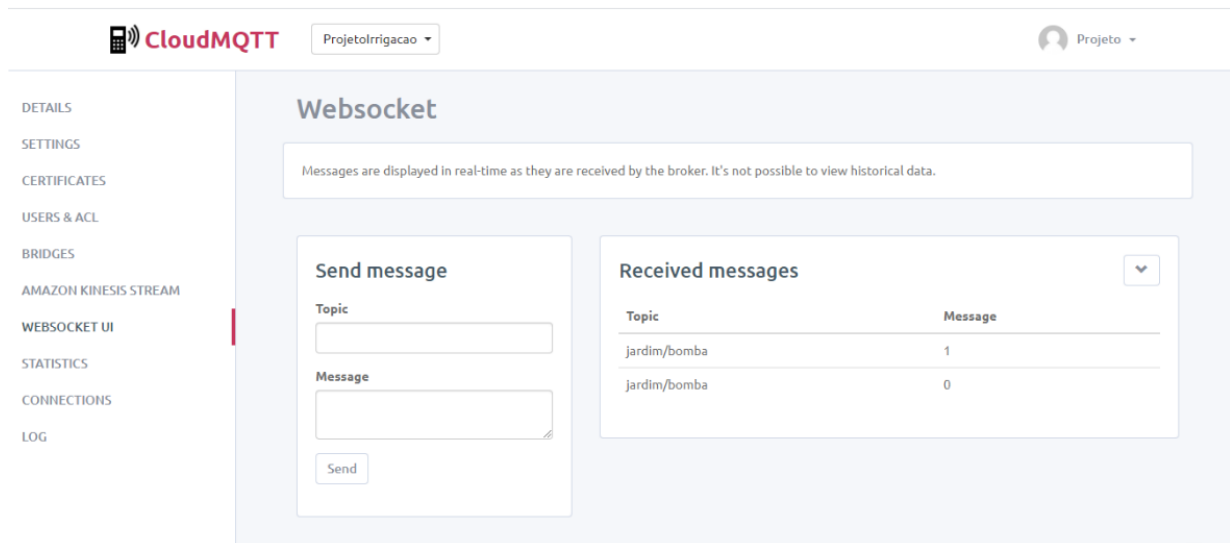


Figura 12 – Configurações de chamada tópico MQTT



Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/Imagens/ConfigChamadaMQTT.jpg>

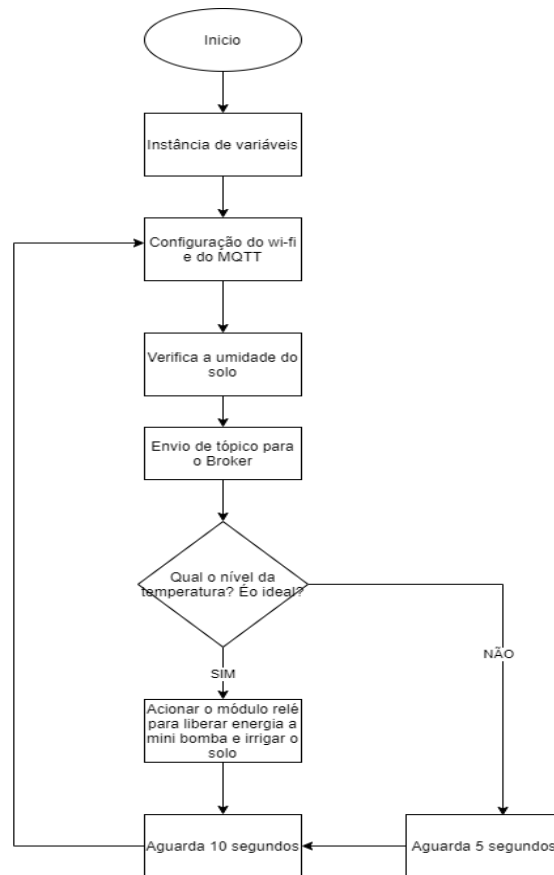
Figura 13 – Retorno do valor da umidade



Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/Imagens/RetornoTopicoSensor.PNG>

O fluxograma abaixo mostra visualmente o fluxo do projeto, onde após a inicialização de variáveis e configurações de conexão, o sistema checa e responde de acordo com o resultado coletado no solo e devolvendo via protocolo MQTT para o sub tópico jardim/bomba.

Figura 14 – Fluxograma do sistema



Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/Imagens/Fluxograma.png>

No início do código começamos importando as bibliotecas necessárias para o funcionamento correto do sistema. Essas bibliotecas, são respectivamente usadas para configurar o WiFi de conexão do ESP8266 e para configurar o broker MQTT.

Figura 15 – Importação das bibliotecas

```
//Incluindo as Bibliotecas da Placa NodeMCU ESP8266 para estabelecer conexão via WiFi
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Logo após configuramos os dados de configuração para conectar com o MQTT, onde primeiramente definimos um ID e o nome do tópico onde iremos dar Subscribe. Depois configuramos a url de conexão com broker público e a porta que usaremos, em nosso caso escolhemos a porta descriptografada.

Figura 16 – Declaração das variáveis do tópico MQTT e broker

```
//Informações da Instância do broker MQTT
const char* mqttServer = "driver.cloudmqtt.com"; //server
const char* mqttUser = "lmoxbnos"; //user
const char* mqttPassword = "xxXwgNY7ACec"; //password
const int mqttPort = 18936; //port
const char* mqttTopicSub = "jardim/bomba"; //tópico que será assinado no Broker

WiFiClient espClient;
PubSubClient client(espClient);
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Em seguida definimos os pinos que usaremos os nossos componentes do circuito, sendo eles:

- Módulo Relé (pino 2) que será usado para controlar o acionamento da bomba de água
- Sensor de Solo Higrômetro (pino 17) que será usado para identificar e retornar a umidade do solo no momento.

Figura 17 – Definição dos pinos

```
#define pino_led_vermelho 13 //5
#define pino_led_verde 12 //7
#define pino_bomba 14 //8 14
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Nesta etapa você irá declarar o nome e senha de sua rede WiFi para conexão com a internet e em baixo definimos o client para conexão com o MQTT.

Figura 18 – Declaração do nome e senha do WiFi

```
//Informações da Rede WIFI para conexão com o CloudMQTT, que est
const char* ssid = "Bravo 007"; //SSID da rede WIFI
const char* password = "mercado088"; //senha da rede wifi
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Por fim, instanciamos as últimas variáveis globais necessárias na lógica do loop(), um valor inteiro para a umidade e uma coleção de char para tráfego de mensagens.

Figura 19 – Declaração das variáveis globais

```
//Variáveis
int valor_analogico;
String strMSG = "0";
char mensagem[30];
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrigacao.ino>

Criamos métodos para fazer e verificar as conexões, o primeiro é o ReconnectMQTT que garante a conexão do client MQTT para a ESP822 com o broker definido. Ela é verificada no início do loop.

Figura 20 – Método que reconecta com o MQTT

```
//Função pra reconectar ao servidor MQTT
void reconnect() {
  //Enquanto estiver desconectado
  while (!client.connected()) {
#ifdef DEBUG
    Serial.print("Tentando conectar ao servidor MQTT");
#endif

    bool conectado = strlen(mqttUser) > 0 ?
      client.connect("ESP8266Client", mqttUser, mqttPassword) :
      client.connect("ESP8266Client");

    if (conectado) {
#ifdef DEBUG
      Serial.println("Conectado!");
#endif
      //Subscreve no tópico
      client.subscribe(mqttTopicSub, 1); //nível de qualidade: QoS 1
    } else {
#ifdef DEBUG
      Serial.println("Falha durante a conexão.Code: ");
      Serial.println( String(client.state()).c_str());
      Serial.println("Tentando novamente em 10 s");
#endif
      //Aguarda 10 segundos
      delay(5000);
    }
  }
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrigacao.ino>

Método que inicia o MQTT dentro do setup

Figura 21 – Método que inicia o MQTT

```
// Iniciando o MQTT
void StartMQTT(void) {
  MQTT.setServer(BROKER, PORTA);
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrigacao.ino>

O método ReconnectWIFI garante a conexão com WiFi, ela é chamada no início do loop, juntamente com a função do ReconnectMQTT.

Figura 22 – Método que reconecta com o WiFi

```
// Função que reconecta o serviço de WiFi
void ReconnectWIFI(void) {
    if (WiFi.status() == WL_CONNECTED)
        return;
    WiFi.begin(SSID, PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.print("Conectado na rede ");
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Na execução do setup, chamamos a função para iniciar o MQTT, além de definir o pinMode de nossos componentes do circuito.

Figura 23 – Função setup

```
void setup() {

    Serial.begin(115200);

    pinMode(sensor_umidade, INPUT);
    pinMode(rele, OUTPUT);

    StartMQTT();
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

A função ConsultaConexoes é chamada no início do loop para verificar as conexões com o WiFi e MQTT, para garantir o funcionamento do sistema de forma integra.

Figura 24 – Função que verifica se o WiFi e MQTT estão conectados

```
void ConsultaConexoes(void) {
    ReconnectWIFI();
    ReconnectMQTT();
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

Por fim temos o loop do sistema, começamos por verificar as conexões na função ConsultaConexoes, logo após fizemos um analogRead no nosso sensor de umidade, que retorna qual o valor do solo no momento, este valor pode ser de 0 a 1024, quando ele

estiver abaixo de 499 significa que o solo está seco e deve ser irrigado. Logo após coletas a umidade do solo, enviamos para o broker do MQTT, que mostrará em sua dashboard qual o valor do solo no momento, podendo saber se está irrigado ou não. Fazemos um IF ELSE simples para detectar se que relé deve ativar a bomba, caso a umidade esteja entre 0 e 499 a bomba será ativada, caso ela esteja entre 500 e 1024 a qualidade do solo está boa e a bomba não será acionada.

Figura 25 – Fluxo principal do sistema

```
//Definindo o parâmetro para um Solo devidamente úmido, acendendo o LED VERDE
if (valor_analogico >= 0 && valor_analogico < 500)
{
    Serial.println("Status: Solo umido");

    digitalWrite (pino_led_vermelho, LOW);
    digitalWrite (pino_led_verde, HIGH);
    digitalWrite (pino_bomba, LOW);

    //Enviando Mensagem ao Broker
    sprintf(mensagem, "0"); //Definindo o valor zero como parâmetro para o Broker mostrar uma imagem "vermelha"
    Serial.print("Mensagem enviada: ");
    Serial.println(mensagem);
    client.publish("jardim/bomba", mensagem);
    Serial.println("Mensagem enviada com sucesso...");

}

//Definindo o parâmetro para um Solo Seco, acendendo o LED VERMELHO
if (valor_analogico >= 500 && valor_analogico <= 1024)
{
    Serial.println(" Status: Solo seco");

    digitalWrite (pino_led_verde, LOW);
    digitalWrite (pino_led_vermelho, HIGH);
    digitalWrite (pino_bomba, HIGH);

    //Enviando Mensagem ao Broker
    sprintf(mensagem, "1"); //Definindo o valor UM como parâmetro para o Broker mostrar uma imagem "verde"
    Serial.print("Mensagem enviada: ");
    Serial.println(mensagem);
    client.publish("jardim/bomba", mensagem);
    Serial.println("Mensagem enviada com sucesso...");

}
delay(10000);

client.loop();
}
```

Fonte: <https://github.com/Meuriam/IOT---ProjetoIrriga-o/blob/main/CodigoProjetoIrrigacao.ino>

3. Resultados

Para que o sistema funcione conforme o planejado, é preciso entender o seu desenho e o processo de demonstração de acordo com os objetivos iniciais. Nesse sentido, deve-se entender que o projeto de irrigação e monitoramento de solo não é apenas uma tecnologia utilizada no contexto da Internet das Coisas (IOT). Também pode obter resultados quantitativos e qualitativos a partir do próprio cuidado com as plantas, pois o projeto auxilia na solução de problemas da planta e solo relacionados ao cuidado inadequado atrelados à irrigação. Melhora e ajuda os cuidados devidos à falta de conhecimento nesta área, e reduz os custos de manutenção de plantas e solo. Neste caso, entendemos que qualquer tipo de pessoa pode utilizar este método e adotar este método de manutenção automática do solo: O primeiro passo é incluir a arquitetura montada e pronta para uso no local desejado a ser utilizado, colocar o sensor de umidade no solo. Em seguida, colocar a bomba em um recipiente com água, para a irrigação quando necessária. Feito isso, é necessário fornecer componentes que necessitem de energia para funcionar, ou seja, conectar o módulo ao USB e a bomba. Desta forma, o sistema iniciará e executará automaticamente suas funções no local por ele indicado, podendo ser monitorado a qualquer momento uma vez que estiver conectado à Internet.

Com os componentes necessários você pode replicar facilmente esse projeto, a montagem do projeto deve ser de acordo com o protótipo de circuito e o vídeo de demonstração do projeto que pode ser acessado em <https://youtu.be/58mIqt8DQOE> [5], para a alimentação do NodeMCU ESP8266 é necessário conectá-lo a um USB e para a Bomba de Água também é necessário conecta-la a uma energia 5V, em nosso caso usamos outro cabo USB. Para melhor entendimento também disponibilizamos um repositório com toda documentação e código disponíveis e necessárias para a execução deste projeto, que pode ser acessado em <https://github.com/Meuriam/IOT---ProjetoIrriga-o>[6].

Figura 26 – Projeto montado



Fonte: Meuriam Assis, 2021

4. Conclusões

Desde a criação da roda, o objetivo da tecnologia sempre foi facilitar e melhorar o cotidiano da vida como um todo. Com o avançar do tempo, essas tecnologias foram evoluindo de forma que também ficassem melhores. É evidente que tais tecnologias, como a internet e automação, são responsáveis por esse grande avanço. Pensando nisso, como minúscula parcela dessa mudança, esse projeto teve o objetivo apresentar como a automação no contexto de Internet das Coisas oferecem uma alternativa que pode auxiliar não somente no cuidado das plantas, mas também em otimização de tempo - o que no cenário atual se fazem ainda mais necessárias. Além disso, oferece um cuidado à flora desejada muitas vezes negligenciado, seja por falta, do próprio tempo, como mencionado anteriormente, seja por falta de conhecimento de quando as plantas devem ou não ser irrigadas. Portanto, conclui-se que o objetivo inicial foi satisfatoriamente alcançado, podendo ser usado e implementado de acordo com a preferência do eventual usuário, atendendo as especificações estabelecidas nesse projeto.

5. Referências

- [1] TRENDS, G. (2020) “plantas para ambientes internos”. Google Trends. Disponível em: <<https://trends.google.com.br/trends/explore?q=plantas%20para%20ambientes%20internos&geo=BR>>. Acesso em: 03 mar. 2021.
- [2] GUIMARÃES, F. (2019) “Faça um Controle de Irrigação Inteligente com Arduino”. Brincando com Ideias. Disponível em: < <https://www.youtube.com/watch?v=BDwgDO3MgDs> >. Acesso em: 15 mar. 2021.
- [3] STRAUB, M. G. (2019) “Projeto Arduino de Irrigação Automática – Sua Planta Sempre Bem Cuidada”. UsinaInfo. Disponível em: <<https://www.usinainfo.com.br/blog/projetoarduino-de-irrigacao-automatica-sua-planta-sempre-bem-cuidada>>. Acesso em: 03 mar. 2021.
- [4] “API Documentation Cloud MQTT”. Disponível em: <https://www.cloudmqtt.com/docs/index.html>. Acesso em: 31 maio 2021.
- [5] Irrigação Automática - Projeto Objetos Inteligentes – Youtube. Direção e Produção: Meuriam Assis e Thiago Garbuió. São Paulo, 11 jun. 2021. Disponível em <https://youtu.be/58mIqt8DQQE>
- [6] Microsoft, GitHub, 2021. Código fonte do projeto Irrigação Automática. Disponível em: <https://github.com/Meuriam/IOT---ProjetoIrriga-o>. Acesso em 11 jun. 2021.