

**PROGRAMA INSTITUCIONAL DE BOLSAS DE  
INICIAÇÃO CIENTÍFICA E TECNOLÓGICA  
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
DARCY RIBEIRO**

CENTRO CCT  
LABORATÓRIO LCMAT

Relatório do período: 06/2018 a 05/2019

**RELATÓRIO ANUAL**

**Título do Projeto: Desenvolvimento e controle de  
dispositivos para Internet das Coisas**

**Título do Plano de Trabalho: Estudo de padrões de Internet  
das Coisas e desenvolvimento de aplicações web para  
Arduino**

Nome da Bolsista: Isabela Correia Pereira  
Curso e N° Matrícula: Ciência da Computação / 00115110215  
Orientador: Fermín Alfredo Tang Montané

Fonte financiadora da Bolsa: PIBIC / CNPq

## 1.- Etapas propostas no plano de trabalho

O plano de trabalho proposto compreende as seguintes etapas:

- a) Estudos gerais sobre redes de computadores e em particular padrões específicos de conexão para Internet das Coisas.
- b) Familiarização com as componentes *Ethernet Shield*, *Wifi Shield* e ESP8266 para conexão com a internet, e realização de testes com estes componentes.
- c) Estudo de linguagens de programação necessárias para o desenvolvimento de aplicações web, tais como *javascript*, *ruby on rails*, entre outros que sejam compatíveis com sistemas micro-controlados.
- d) Extensão do projeto do braço robótico mediante a implementação do controle através da internet.
- e) Pesquisa sobre aplicativos web e dispositivos arduino para IoT.
- f) Escolha e desenvolvimento do segundo projeto de dispositivo para o micro-controlador Arduino. Incluindo as seguintes etapas: Projeto do protótipo eletrônico, estrutura física de suporte e desenvolvimento do aplicativo web.
- g) Realização de experimentos de avaliação e desempenho do dispositivo.
- h) Elaboração de relatório técnico.

A etapas do plano de trabalho foram em sua grande maioria cumpridas de maneira satisfatória. A bolsista deu prioridade ao desenvolvimento do item f) e através dos conhecimentos adquiridos ao longo da pesquisa, foi utilizada a placa NodeMCU na montagem de um protótipo, circuito e código para o dispositivo Planta IoT, fazendo com que o dispositivo se comunique com a internet através de um servidor online e enviando dados para o mesmo. Melhorias nesse projeto mostraram que é possível fazer a irrigação da planta sem interferência humana, através do uso da válvula solenoide. Em relação ao item d) extensão do braço robótico, tínhamos a expectativa de mudar o controle de seus motores para uma via remota, o que foi possível mediante a placa NodeMCU, mas para apenas um motor. A conexão dos quatro motores foi inviável para esta placa de maneira direta, precisando de um controlador independente.

## 2.- Introdução

O número de pessoas conectadas à internet tem crescido rapidamente nos últimos anos e com elas também uma diversidade de serviços prestados através da rede. Dentre os principais serviços devemos destacar o surgimento do comércio eletrônico, a mídia eletrônica, principalmente os serviços de notícias, a produção de conteúdo áudio visual, o streaming de vídeos e músicas, entre outras.

A Internet das Coisas é considerada a terceira revolução ligada à internet, após o surgimento dos dispositivos móveis e a disseminação da rede a nível global. Trata-se da ideia de que diversos dispositivos, muitos deles de uso cotidiano ou não, podem também se conectar à internet e produzir conteúdo e/ou prestar serviços. Além disso, tais dispositivos podem se comunicar entre si. Como exemplo de dispositivos pode-se destacar o uso de sensores de temperatura, humidade, que coletam informações de forma automática.

**A Internet das Coisas pode ser definida como uma infraestrutura de rede global, dinâmica e com capacidades de autoconfiguração, onde as “coisas” são dispositivos eletrônicos que possuem uma identidade e capacidade de interagir e se comunicar entre si através da rede. Estes dispositivos percebem o seu ambiente, “mundo real/físico” através de sensores e são capazes de produzir dados sobre esse ambiente. Os dados alimentam a rede e ativam objetos virtuais que executam processos e serviços. Tais processos podem envolver a intervenção humana ou não. Quando solicitadas as “coisas” são capazes de reagir de maneira autônoma através de atuadores produzindo mudanças no seu ambiente. Espera-se que as “coisas” se tornem participantes ativas nas mais diversas atividades humanas.**

## 3.- Objetivos

O presente trabalho visa dar continuidade ao primeiro ano de pesquisa de iniciação científica que tratou de aspectos gerais referentes ao desenvolvimento de protótipos para Internet das Coisas (IoT). Uma vez que existem diferentes tecnologias para conseguir a conexão de dispositivos à internet, propõe-se para o segundo ano de pesquisa, o estudo dos diferentes Padrões de Conexão de IoT voltado especificamente para aplicações web. Além disso, propõe-se também trabalhar em um segundo protótipo de dispositivo para Internet das coisas. Da mesma forma que no plano anterior, se deverá trabalhar todas as etapas de construção do dispositivo micro-controlado, desde o desenvolvimento do circuito eletrônico, construção da estrutura física de suporte e programação do micro-controlador e do aplicativo via web.

## 4.- Metodologia

A metodologia utilizada no projeto gira em torno da plataforma Arduino, do NodeMCU e da construção de dispositivos eletrônicos aplicando conceitos de eletrônica básica assim como princípios de programação em uma linguagem baseada em C/C++ específica para a plataforma Arduino. Além do *Javascript* e do HTML que são linguagens web, utilizadas para construção das páginas para manipulação dos dispositivos.

A fim de cumprir com os objetivos propostos, a pesquisa começou com consultas em livros e artigos disponibilizados pelo orientador, além de cursos online e seguiu com execução de testes e familiarização com os componentes *WiFi Shield* e *Esp8266*. Além testes com o projeto do braço, foi construído um protótipo para medir a umidade do solo e enviar dados para uma plataforma de IOT para decidir quando uma planta deve ser regada.

Nesta seção apresenta-se uma breve introdução aos componentes utilizados na realização dos projetos.

### 4.1.- ESP8266

De acordo com [1], A Internet das Coisas representa a próxima evolução da Internet, dando um grande salto na capacidade de coletar, analisar e distribuir dados que nós podemos transformar em informações, conhecimento e, por fim, sabedoria. Nesse contexto, ela tem como objetivo conectar itens à uma rede mundial de computadores. Ainda, pode-se afirmar, segundo [2], que ela se estende ao mundo real, representando a visão de que itens físicos não estão mais desconectados no mundo virtual, ao contrário disso, podem ser controlados remotamente e atuam como pontos de acesso aos serviços de internet. Dessa forma, a Internet das Coisas torna a Computação algo onipresente, gerando grande desenvolvimento tanto para economia, quanto para os indivíduos.

Baseado nisso, o ESP8266, traz a proposta de conectar objetos à internet de forma simples e barata e vem sendo cada vez mais utilizado. São diversos os módulos existentes, e os utilizados nesse projeto serão apresentados com mais detalhes a seguir.

#### 4.1.1-ESP8266-01

O módulo WiFi ESP8266 ESP-01 foi desenvolvido para que um microcontrolador como o Arduino possa ser conectado a uma rede WiFi de forma fácil, eficaz e a um baixo preço. Este módulo WiFi suporta as redes 802.11 b/g/n, muito usadas atualmente, podendo trabalhar como um Ponto de Acesso (*Access Point*) ou como uma Estação (*Station*) enviando e recebendo dados. A comunicação do módulo com o Arduino pode ser feita via serial utilizando os pinos RX e TX, podendo ser configurada através de comandos AT, que consistem em uma série de cadeias curtas de texto que podem ser combinadas para produzir comandos para operações tais como ligar, desligar ou mudar os parâmetros da conexão. A figura 1 ilustra o módulo ESP-01.

### **Especificações:**

- Chip: ESP8266
- Modelo: ESP-01
- Tensão de operação: 3,3V
- Suporte a redes: 802.11 b/g/n
- Alcance: 90m aprox.
- Comunicação: Serial (TX/RX)
- Suporta comunicação TCP e UDP
- Conectores: GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de Temperatura interno.
- Modo de segurança: OPEN/WEP/WPA\_PSK/ WPA2\_PSK/ WPA\_WPA2\_PSK
- Dimensões: 25 x 14 x 1mm
- Peso: 7g



**Figura 1** – ESP8266-01 – Fonte [3]

#### **4.1.2-NodeMCU**

É uma placa que utiliza o módulo ESP8266 ESP-12 e foi criada para facilitar o desenvolvimento de aplicações para este módulo, que vem soldado nela. Junto com o módulo, o NodeMCU já traz com ele todos os circuitos necessários para que o ESP-12 funcione. Sua alimentação pode ser feita através do próprio conector USB, ou externamente, utilizando o pino VIN, disponível na placa.

A grande vantagem dessa placa é que ela funciona como se fosse um Arduino. Pode ser programada na própria IDE do Arduino, utilizando a linguagem de programação C++, ou em sua própria IDE, através da linguagem LUA. Os modelos mais comuns são o NodeMCU ESP-12 N e o NodeMCU ESP12-E. Neste projeto foi utilizado o segundo modelo. A figura 2 ilustra um NodeMCU, além de mostrar sua distribuição de pinos.

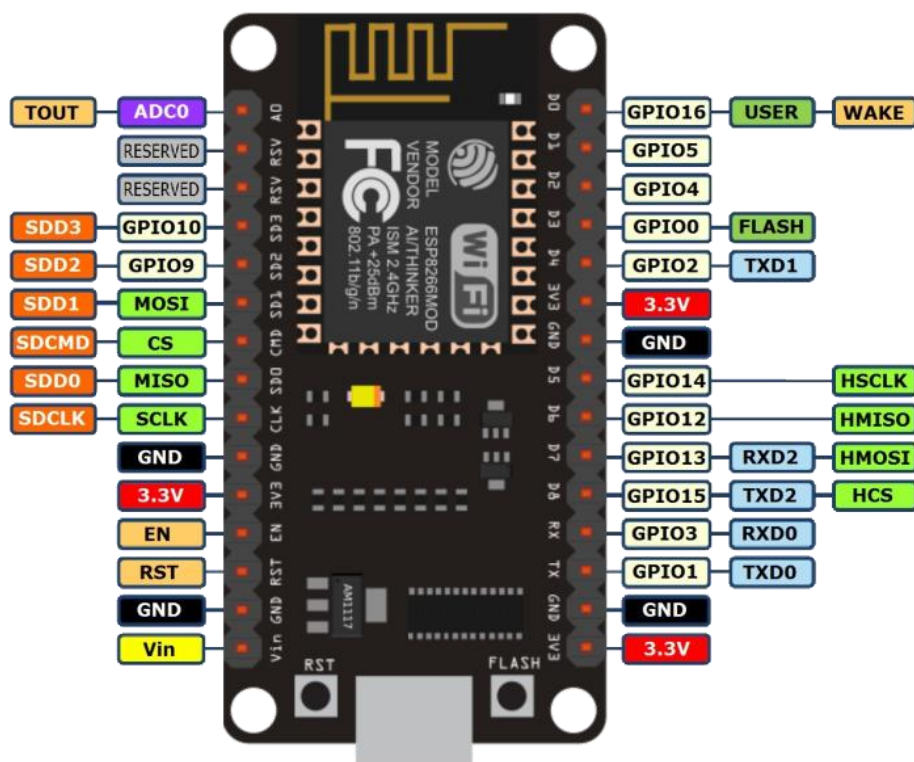


Figura 2 – NodeMCU – Fonte [4]

## 4.2 Componentes Básicos

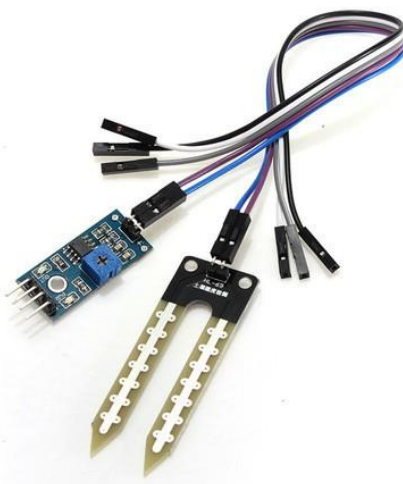
Nesta seção são apresentados componentes básicos utilizados na construção dos projetos detalhados nesse relatório. O conhecimento das características dos componentes permite que eles possam ser utilizados de maneira adequada, evitando possíveis danos a outros dispositivos.

### Sensor de Umidade do Solo – Higrômetro

Esse sensor é composto por duas hastes que ficam presas ao solo que desejamos monitorar. Um circuito irá nos retornar o nível de condutividade do solo. Essas hastes consistem em dois eletrodos que conduzirão uma corrente passando pelo solo. Dessa forma é possível ler o nível de umidade por comparação com a resistência do potenciômetro existente no módulo do sensor.

Assim, quando o solo estiver seco, a resistência entre os eletrodos do sensor de umidade de solo irá aumentar dificultando a passagem de corrente. Quando o solo absorve água, a resistência diminuirá permitindo que a corrente passe. A partir disso, conseguimos descobrir se o solo está molhado ou seco.

O Sensor de Umidade do Solo conta com uma saída digital (D0) e com uma saída analógica (A0). O sinal digital é ajustado através do potenciômetro presente em sua placa comparadora. A figura 4 ilustra o sensor.



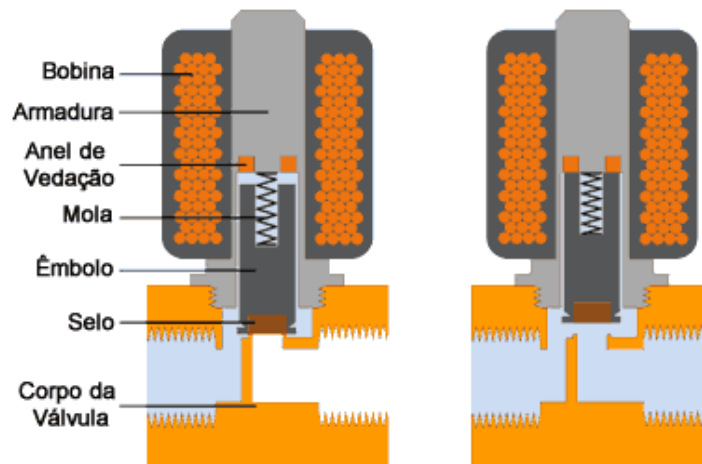
**Figura 4** – Sensor de Umidade do Solo - Higrômetro – Fonte [6]

### Válvula de Vazão Solenóide Água 12VDC

A **válvula solenóide** é uma válvula eletromecânica. Ela recebe o nome de solenóide devido ao seu componente principal ser uma bobina elétrica com um núcleo ferromagnético móvel no centro, sendo este núcleo chamado de êmbolo. Em posição de repouso, o êmbolo tampa um pequeno orifício por onde é capaz de circular um fluido. Quando uma corrente elétrica circula através da bobina, esta corrente cria um campo magnético que por sua vez exerce uma força no êmbolo. Como resultado, o êmbolo é puxado em direção ao centro da bobina de modo que o orifício se abre, permitindo passagem do líquido. Este é o princípio básico usado para abrir e fechar uma válvula solenóide. A figura 5 ilustra o funcionamento dessa válvula.

### **Especificações:**

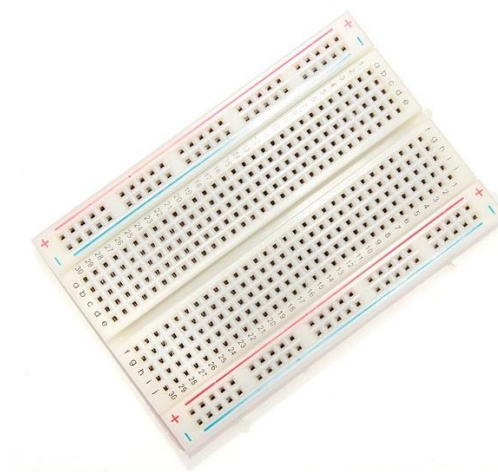
- Válvula de Vazão Solenóide
- Uso: Água e fluidos de baixa viscosidade
- Tensão de operação: 12VDC
- Corrente: 250mA
- Pressão – 0.02 à 0.8Mpa
- Modo de operação: Normal fechado
- Temperatura máxima do fluido: 100°C
- Diâmetro da rosca: 19,5mm
- Comprimento da rosca: 17mm
- Tipo de válvula: Diafragma (operada por servo)
- Dimensões: 85 x 53 x 42mm



**Figura 5** – Válvula de vazão solenóide – Fonte [7]

### Protoboard

A protoboard é uma placa de ensaio (matriz de contato) que serve como base para a construção de circuitos eletrônicos e teste de projetos. Ela permite que os componentes sejam retirados facilmente e reutilizados depois em uma outra montagem. Seus orifícios são interligados internamente através de contatos, o que elimina a necessidade do uso de solda. A Figura 6, ilustra uma protoboard.



**Figura 6** – Protoboard – Fonte [8]

### Jumper

Os jumpers são elementos condutores utilizados para conectar dois pontos em um circuito eletrônico. Quando um jumper está conectado à pelo menos dois pinos, ele está fechado. Sem conexão ou apenas com uma conexão, o circuito está aberto. A figura 7, ilustra jumpers do tipo macho x macho.





**Figura 7** – Jumper – Fonte [9]

### Resistores

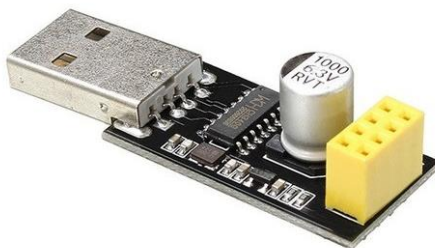
Um resistor é um componente eletrônico que é usado para limitar a voltagem em um determinado circuito de maneira a garantir níveis adequados de corrente para cada componente sem danificá-los. Cada resistor tem um valor diferente e é escolhido de acordo com a tensão do circuito que queremos diminuir. O grau de resistência dese componente é medido em ohms e codificado utilizando um código de cores. A figura 8 mostra resistores de 330 ohms.



**Figura 8** – Resistores 330 ohms – Fonte [10]

### Adaptador USB para Módulo ESP8266 ESP-01

O adaptador permite que o módulo ESP8266-01 seja facilmente programado, sem depender de outras placas ou cabos. Ele possui um chip UART-USB, que é reconhecido pelo computador como uma porta serial (COM). Com ele, é possível programar o ESP-01 usando comandos AT, a linguagem lua, ou a própria IDE do Arduino. A figura 9, ilustra o adaptador USB.



**Figura 9** – Adaptador USB para Módulo: ESP8166 ESP-01– Fonte [11]

## 5.- Resultados e Discussão

Como resultado desta pesquisa, além dos estudos e testes relacionados aos diversos componentes disponíveis para conexão com a internet e do estudo das linguagens de programação necessárias para o desenvolvimento de aplicações web, foi desenvolvido um projeto para Internet das Coisas. O projeto escolhido, denominado Plata IoT consiste em medir a umidade do solo de uma planta, a fim de saber quando é necessário regá-la, e é dividido em duas partes. Na primeira parte é possível saber quando o solo está seco demais, através dos dados enviados para uma plataforma online, que por sua vez gera um *tweet* avisando que a planta precisa ser regada. A segunda parte traz o desafio de fazer a planta ser regada sozinha sempre que necessário, utilizando uma válvula que permite ou não a passagem de água. Paralelamente ao desenvolvimento desse projeto, teste foram feitos para descobrir qual era a melhor maneira de controlar o braço robótico, desenvolvido no primeiro ano de pesquisa. Conclui-se que é utilizando o NodeMCU, assim como foi feito no projeto Planta IoT. No entanto, esse dispositivo traz uma limitação, por só ter um único pino analógico, tornou inviável o controle dos quatro servo motores presentes no braço, já que cada um deles exige um pino analógico individual. Devido ao pouco tempo hábil para contornar essa situação, foi dada preferência ao projeto principal, que, por sua vez, foi concluído com sucesso, permitindo com que o objetivo desse trabalho, de controlar dispositivos via internet, fosse concluído.

### 5.1.- Protótipo de monitoramento de uma Planta IoT

Nesta seção serão apresentados os detalhes sobre desenvolvimento do protótipo para o monitoramento de uma planta através da internet.

Este projeto de monitoramento utiliza a plataforma NodeMCU baseada no microcontrolador ESP8266-12E. O módulo NodeMCU é conectado a um sensor de umidade do solo, que faz leituras periódicas, calcula a umidade em termos percentuais e envia essa informação via rede *WiFi*, a cada trinta segundos, para um servidor online, chamado *ThingSpeak*. Esse servidor é um tipo de plataforma IoT, que oferece serviços de forma gratuita, permitindo o armazenamento de dados numéricos mediante *upload*, assim como a execução de alguns serviços. Tais dados ficam armazenados na nuvem, onde então é permitida a visualização dos mesmos em gráficos em tempo real. Ele possui a limitação de que o tempo entre upload de dados deve ser, no mínimo, de 15 segundos. Caso isso for desobedecido, os dados que forem enviados em intervalo de tempo inferior serão ignorados e perdidos.

Para enviar dados ao *ThingSpeak* é necessária a criação de uma chave de escrita que será associada ao canal de comunicação utilizado na plataforma. O envio de dados ao servidor é realizado na forma de uma requisição HTTP. Após enviar os dados para a plataforma, o dispositivo avalia o grau de umidade do solo, de maneira que sempre que a planta estivesse precisando de água, isto é, sempre que a porcentagem de umidade do solo esteja abaixo de 60%, é gerado um *tweet* informando a necessidade de regar a planta. O envio do *tweet* é um serviço fornecido por uma aplicação, chamada *React*, disponível no servidor

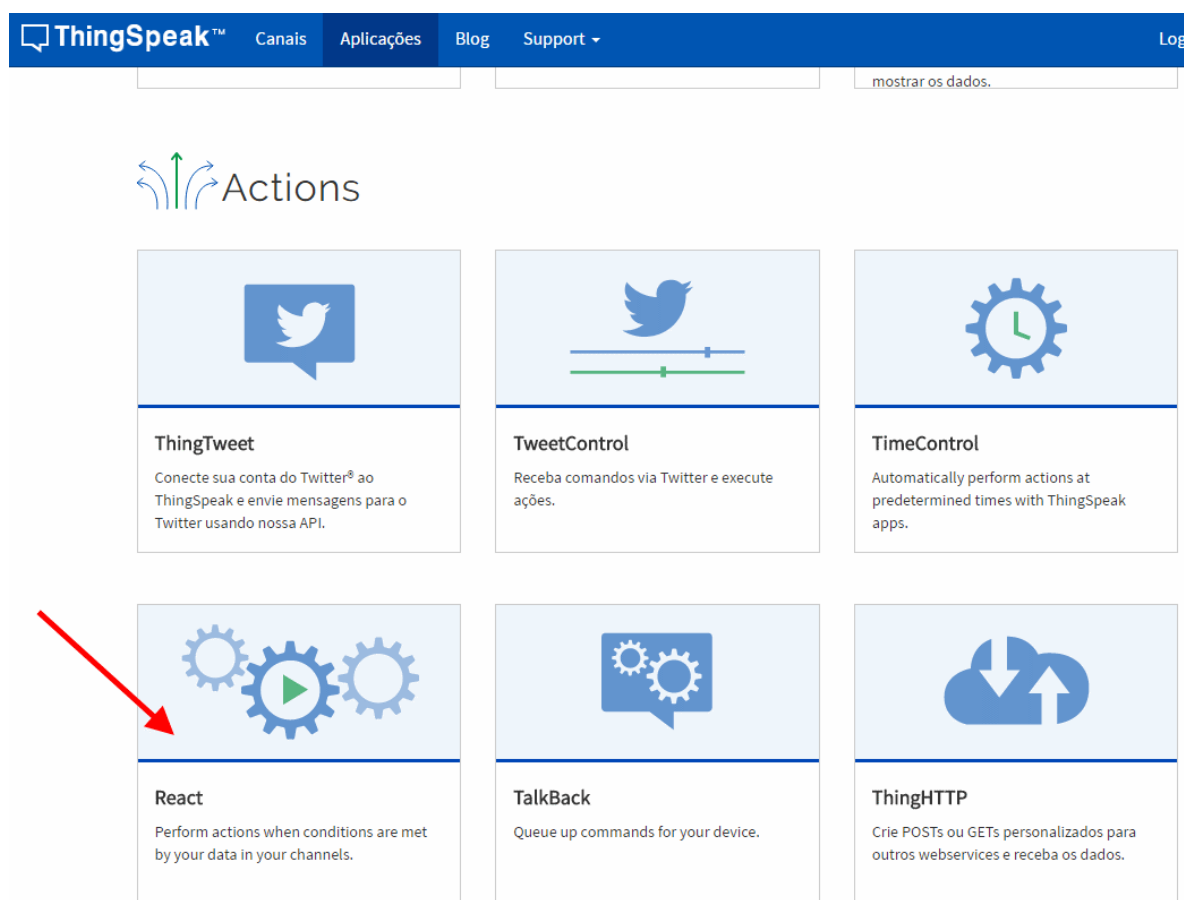
utilizado. Em geral, essa aplicação permite programar a ação que deverá ser tomada, desde que seja atendida uma determinada condição.

### Configurações Iniciais

Antes de começar o desenvolvimento do projeto, foi preciso configurar o NodeMCU para que ele pudesse ser programado na IDE do Arduino. Para isso, foi necessário instalar o pacote correspondente ao módulo ESP8266 e verificar se o dispositivo estava conectado na porta serial correta.

Após essa etapa, foi criada uma conta na plataforma *ThingSpeak*, para que uma chave de escrita pudesse ser gerada e um canal de comunicação definido. Assim, a chave criada serve para direcionar para qual canal os dados devem ser enviados. Por último, foi necessário criar uma conta no *Twitter* para representar a Planta IoT e configurar a aplicação *React* dentro do servidor *ThingSpeak*. A configuração do *React* segue as seguintes etapas:

1. Fazer login na plataforma *ThingSpeak* com a conta criada inicialmente;
2. Na página principal, clicar em Aplicações, e logo em seguida em *React*, conforme ilustrado na Figura 10;



**Figura 10** – Configurar tweets no ThingSpeak – Fonte: A autora

3. Clicar em “New React” e preencher os dados de acordo com a Figura 11;
4. Salvar a aplicação clicando em “Save React”.

Aplicações / React / Novo

**React Name**

**Condition Type**

**Test Frequency**

**Condition**   
  
 field

**Ação**   
 then tweet   
 using Twitter account

**Options** ☒ Run action only the first time the condition is met  
☐ Run action each time condition is met

**Save React**

**Figura 11** – Configurar tweets no ThingSpeak – Fonte: A autora

A apresentação do desenvolvimento deste protótipo é dividida nas seguintes seções: i) Componentes utilizadas, ii) Diagrama do dispositivo iii) Código de controle para o micro-controlador e iv) Protótipo do dispositivo.

### Componentes Utilizados

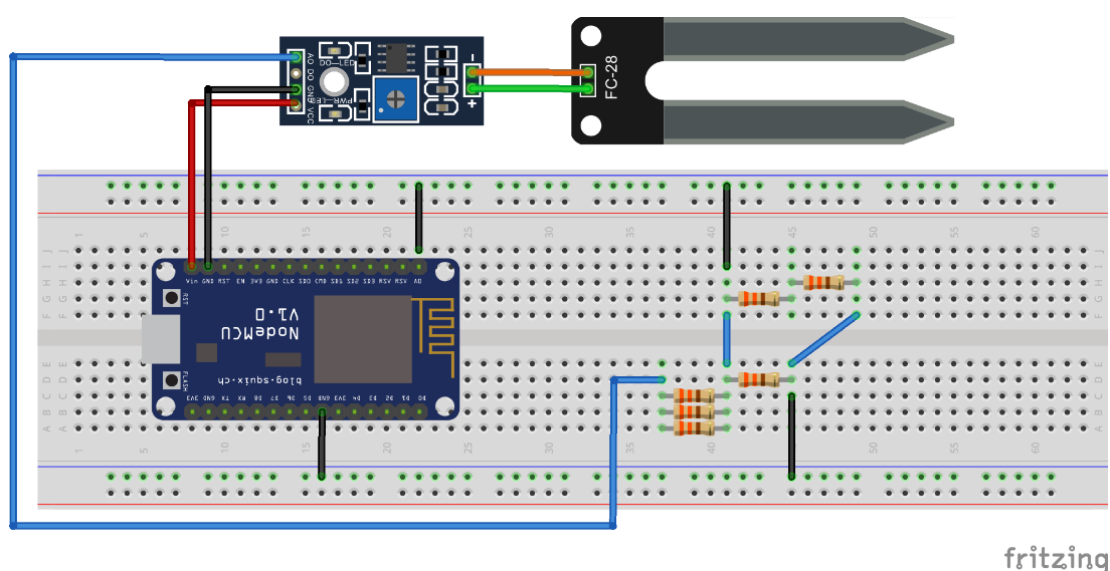
Foram utilizados os seguintes componentes, descritos na sessão anterior:

1. NodeMCU;
2. Jumpers;
3. Resistores de 330 ohms;
4. Protoboard;
5. Sensor de Umidade do Solo (Higrômetro).

### Diagrama de Conexão

O diagrama de conexão entre as componentes mencionadas, definem o circuito de controle. A figura 12 ilustra o diagrama correspondente. A alimentação de energia é feita pelo cabo USB. Os dados do sensor higrômetro são lidos através do único pino analógico do NodeMCU (A0), e posteriormente convertidos em unidades percentuais (0 – 100%). Para realizar esta conexão, foi necessário usar um divisor de tensão, pois a tensão de saída do higrômetro (linha azul) fornece 5V, enquanto a entrada analógica (A0) do NodeMCU aceita no máximo 3,3V. O diagrama original exigia um resistor de 100 ohms e um de 200 ohms, mas como não haviam esses resistores disponíveis, foi feita uma adaptação. Para substituir o resistor de 100 ohms, foram usados três de 330 ohms arranjados de forma paralela, de forma a obter um valor aproximado. Algo parecido foi feito para substituir o resistor de 200 ohms, mas dessa vez foi necessário dispor dois resistores de 330 ohms de forma serial e um de 330 paralelo a esses dois, obtendo um valor de resistência próximo a 220 ohms, o que foi suficiente nesse circuito.

Vale mencionar que o sensor higrômetro possui um potenciômetro (pequeno quadrado azul) que permite ajustar a sensibilidade do mesmo.



**Figura 12** – Diagrama de Conexão para Monitoramento da Planta IoT – Fonte: A autora

### Código de Controle

Como mencionado anteriormente, o circuito para o monitoramento da umidade do solo é controlado mediante uma placa NodeMCU que possui um módulo ESP8266 com capacidade de conexão a internet através da tecnologia *WiFi*.

O código de controle para este circuito realiza um conjunto de tarefas de configuração e controle. Na parte de configuração, define-se o nome da rede local que será utilizada assim como o endereço da plataforma de IoT, que funcionará como um servidor que publicará as informações sobre a umidade do solo. Na parte de controle, o dispositivo realiza medições periódicas sobre a

umidade do solo, em intervalos de 30 segundos, e envia essa informação à plataforma de IoT, para que de acordo com o nível de umidade, caso seja necessário se envie uma mensagem de alerta via *Twitter* informando-se a necessidade de regar a planta. O código utilizado é uma adaptação do código proposto pelo site da FilipeFlop [12]. Alguns comentários foram introduzidos no código para facilitar a compreensão.

Na parte inicial deste código, mostrada no Quadro 1, primeiramente inclui-se a biblioteca <ESP8266Wifi.h> que disponibiliza as funções necessárias para a conexão do módulo ESP8266 à rede *WiFi*. Define-se o nome da rede local que será utilizada assim como a respectiva senha. Além disso, especifica-se o intervalo de tempo para o envio de dados de umidade do solo à plataforma de IoT, definido em 30.000 milissegundos. Define-se também o endereço da plataforma de IoT, que corresponde a plataforma *ThingSpeak*, junto a sua chave de escrita gerada com antecedência. Além disso, define-se variáveis para medição do tempo de conexão e criação do cliente que solicita o serviço a plataforma de IoT. Finalmente, são declarados os protótipos das funções para envio de informações ao servidor, conexão *WiFi* e leitura de umidade do solo.

```
#include <ESP8266WiFi.h>

//defines
#define SSID_REDE "Pesquisa CC" //coloque aqui o nome da rede que se deseja conectar
#define SENHA_REDE "qr5dr1jw" //coloque aqui a senha da rede que se deseja conectar
#define INTERVALO_ENVIO_THINGSPEAK 30000//intervalo entre envios de dados ao ThingSpeak (em ms)

//constantes e variáveis globais
char EnderecoAPIThingSpeak[] = "api.thingSpeak.com";
String ChaveEscritaThingSpeak = "UC5MNQ5X5G3OIYU3";
long lastConnectionTime;
WiFiClient client;

//prototypes
void EnviaInformacoesThingspeak(String StringDados);
void FazConexaoWiFi(void);
float FazLeituraUmidade(void);
```

**Quadro 1.-** Código de Controle: Definição da Rede e Plataforma IOT

No Quadro 2, apresenta-se a função **EnviaInformacoesThingspeak()** que inicia uma nova conexão do cliente ao servidor, mediante o comando *client.connect()* e verifica o estado dessa conexão, e em caso de verificação positiva, envia à plataforma *ThingSpeak* a informação sobre umidade de solo, recebida na forma de uma *string*, como uma requisição HTTP. Além disso, a função registra o tempo transcorrido até o momento, desde o início da execução, mediante o comando *millis()* como forma de registrar o tempo da nova conexão.

```

void EnviaInformacoesThingspeak(String StringDados)
{
    if (client.connect(EnderecoAPIThingSpeak, 80))
    {
        //faz a requisição HTTP ao ThingSpeak
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY:
"+ChaveEscritaThingSpeak+"\n");
        client.print("Content-Type: application/x-www-form-
urlencoded\n");
        client.print("Content-Length: ");
        client.print(StringDados.length());
        client.print("\n\n");
        client.print(StringDados);

        lastConnectionTime = millis();
        Serial.println("- Informações enviadas ao ThingSpeak!");
    }
}

```

**Quadro 2.-** Código de Controle: Função de Envio de Dados ao *ThingSpeak*

A função **FazConexãoWiFi()** é mostrada no Quadro 3. Esta função finaliza a conexão do cliente que estiver usando o servidor e inicia uma nova conexão após um segundo de espera. Para isso, inicia uma nova conexão *Wifi*, de acordo com os parâmetros de identificação de rede e senha definidos previamente. Enquanto o estado da conexão não for verificado, aguardará por mais meio segundo e imprimirá uma mensagem no monitor serial. Após verificada a conexão, imprime no monitor serial o endereço de IP local do servidor.

```

void FazConexaoWiFi(void)
{
    client.stop();
    Serial.println("Conectando-se à rede WiFi...");
    Serial.println();
    delay(1000);
    WiFi.begin(SSID_REDE, SENHA_REDE);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi conectado com sucesso!");
    Serial.println("IP obtido: ");
    Serial.println(WiFi.localIP());

    delay(1000);
}

```

**Quadro 3.-** Código de Controle: Função de conexão *WiFi*

O Quadro 4 apresenta o código correspondente a função que faz a leitura da umidade do solo a partir do sensor higrômetro. Esse sensor está conectado a uma porta analógica, que no caso do NODEMCU é capaz de registrar tensões DC entre 0 até 3,3V, convertendo essa tensão em um valor numérico na escala de 0 a 978. Vale observar que quanto maior a leitura do sensor menor será a umidade do solo e vice-versa. Com base na leitura do sensor, a função calcula uma medida de umidade em termos percentuais (que pode variar de 0 a 100%), definido pela seguinte expressão:

$$Umidade \% = 100 \times \frac{(978 - ValorSensor)}{978}$$

A função retorna essa medida percentual.

```
float FazLeituraUmidade(void)
{
    int ValorADC;
    float UmidadePercentual;

    ValorADC = analogRead(0); //978 -> 3,3V
    Serial.print("[Leitura ADC] ");
    Serial.println(ValorADC);

    //Quanto maior o numero lido do ADC, menor a umidade.
    //Sendo assim, calcula-se a porcentagem de umidade por:
    //
    //      Valor lido                Umidade percentual
    //      -      0                    -      100
    //      |
    //      |
    //      -      ValorADC            -      UmidadePercentual
    //      |
    //      |
    //      -      978                -      0
    //
    //      (UmidadePercentual-0) / (100-0) = (ValorADC - 978) / (-
978)
    //      Logo:
    //      UmidadePercentual = 100 * ((978-ValorADC) / 978)

    UmidadePercentual = 100 * ((978-(float)ValorADC) / 978);
    Serial.print("[Umidade Percentual] ");
    Serial.print(UmidadePercentual);
    Serial.println("%");

    return UmidadePercentual;
}
```

**Quadro 4.-** Código de Controle: Função de Leitura de Umidade

Uma vez descritas todas as funções do código de controle, resta apresentar o programa principal de controle. Este programa é composto basicamente por duas funções: A função **setup()** que é executada uma única vez e serve para configuração e inicialização dos dispositivos e a função **loop()**, que é de fato o programa principal de controle, e é executado de forma contínua e ininterrupta. Ambas as funções são mostradas no Quadro 5.



Na função **setup()**, primeiro inicializa-se o monitor serial da IDE Arduino, que poderá ser utilizado para mostrar mensagens referentes ao estado corrente da aplicação. Depois chama-se a função **FazConexãoWiFi()** que permite que o NodeMCU inicie sua conexão com a internet.

Na função **loop()**, em cada iteração, primeiro verifica-se que o cliente esteja desconectado. Caso contrário, a conexão é concluída. Após essa verificação realiza-se a leitura do sensor de umidade. Finalmente, verifica-se se o intervalo de tempo transcorrido desde a última conexão é superior a 30 segundos. Em caso afirmativo os dados de umidade são enviados ao *ThingSpeak*. É possível observar essas operações no monitor serial da IDE do Arduino.

```
void setup()
{
    Serial.begin(9600);
    lastConnectionTime = 0;
    FazConexaoWiFi();
    Serial.println("Planta IoT com ESP8266 NodeMCU");
}

//loop principal
void loop()
{
    float UmidadePercentualLida;
    int UmidadePercentualTruncada;
    char FieldUmidade[11];

    //Força desconexão ao ThingSpeak (se ainda estiver conectado)
    if (client.connected())
    {
        client.stop();
        Serial.println("- Desconectado do ThingSpeak");
        Serial.println();
    }

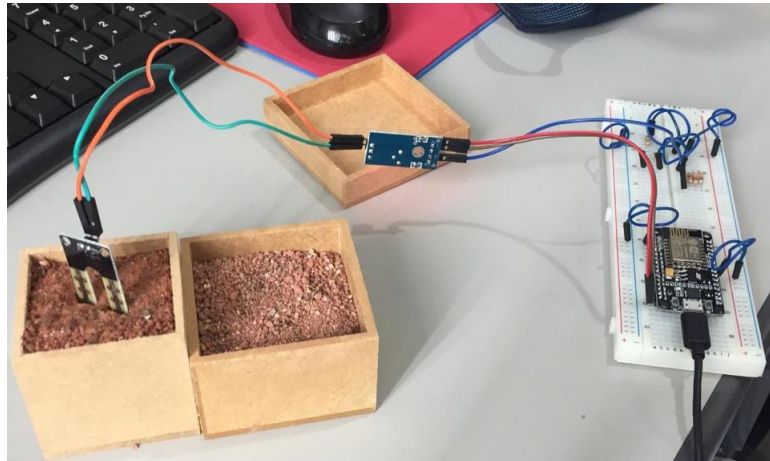
    UmidadePercentualLida = FazLeituraUmidade();
    UmidadePercentualTruncada = (int)UmidadePercentualLida; //trunca
    umidade como número inteiro

    //verifica se está conectado no WiFi e se é o momento de enviar
    dados ao ThingSpeak
    if(!client.connected() &&
        (millis() - lastConnectionTime > INTERVALO_ENVIO_THINGSPEAK))
    {
        sprintf(FieldUmidade, "field1=%d", UmidadePercentualTruncada);
        EnviaInformacoesThingspeak(FieldUmidade);
    }
    delay(1000);
}
```

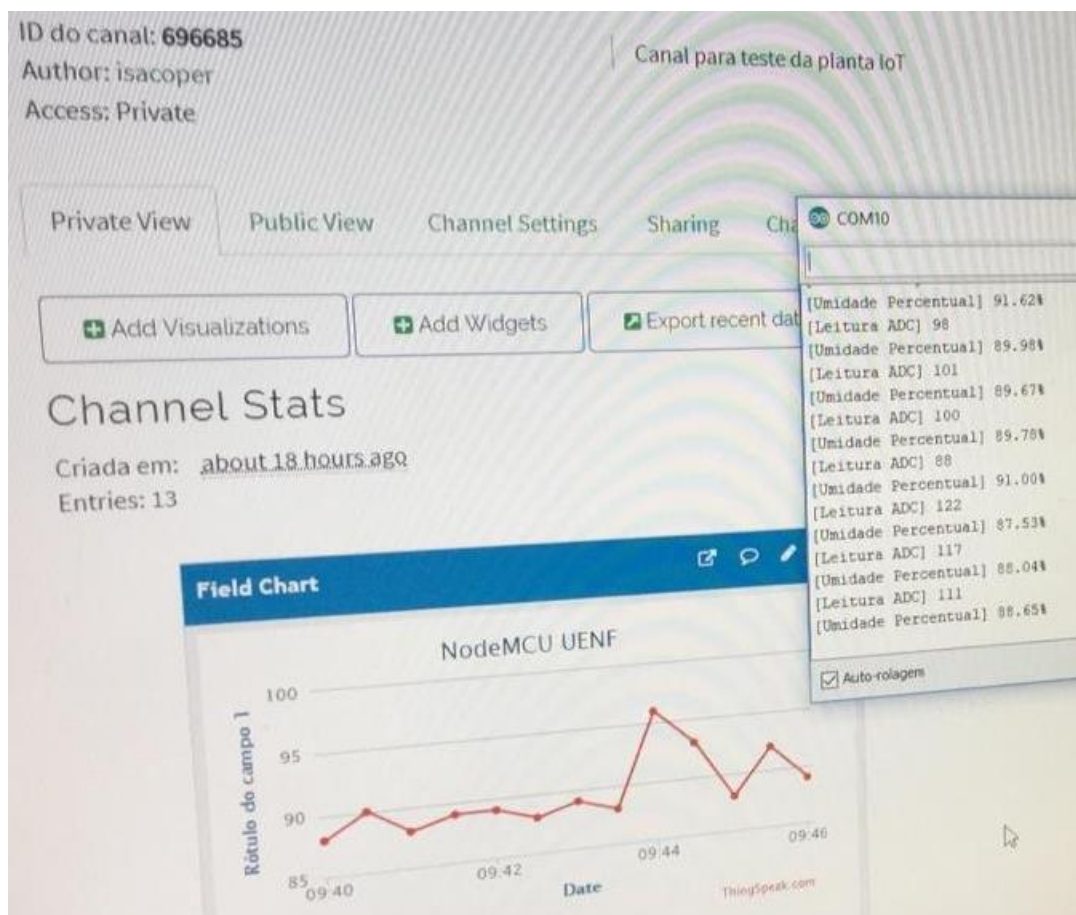
**Quadro 5.-** Código de Controle: Funções de Inicialização e Controle

### Protótipo do Dispositivo

Um primeiro protótipo para o monitoramento da umidade do solo foi desenvolvido e devidamente testado. Para isso foram utilizadas duas pequenas caixas em material mdf, que foram preenchidas com terra coletada na própria instituição. Na primeira caixa a terra foi umedecida enquanto na segunda ela foi mantida seca. A figura 13 ilustra o protótipo de monitoramento de umidade do solo devidamente montado.



**Figura 13 – Simulação de Planta IoT:**  
Protótipo do Dispositivo – Fonte: A autora



**Figura 14 – Dados na plataforma ThingSpeak** – Fonte: A autora

Já a Figura 14 mostra os dados de umidade do solo recebidos pela plataforma de IoT *ThingSpeak* e apresentados de forma gráfica ao longo do tempo, em termos percentuais e em intervalos de 30 segundos. Na imagem, observam-se as leituras de umidade na terra molhada, variando de 88,6% até 97%. A imagem mostra também os dados numéricos das leituras no monitor serial do IDE do Arduino.

Finalmente, a Figura 15 mostra o perfil da planta IoT no Twitter, ilustrando como seria a interação da mesma ao informar quando ela deveria ser regada.



**Figura 15** – Perfil no Twitter da Planta IoT – Fonte: A autora

## 5.2.- Protótipo para irrigação autônoma da Planta IoT

Nesta etapa apresenta-se uma modificação do dispositivo anterior, que é uma tentativa de melhoria que visa garantir a toma de decisão automática por parte do dispositivo. No caso anterior, temos a planta nos informando, através de *tweets* enviados pelo servidor, quando precisa ser regada. Porém, essa situação pode não ser considerada totalmente adequada para o usuário. Uma alternativa possível seria fazer com que a planta pudesse se regar “sozinha” toda vez que o sensor identificasse um nível de umidade considerado baixo. Na verdade, propõe-se desenvolver um dispositivo que tome essa decisão de maneira autônoma. Este tipo de dispositivo permitiria, por exemplo, que uma pessoa que precisa viajar constantemente, pudesse cultivar plantas em sua residência, sem se preocupar nos períodos de ausência.

Assim, em vez do servidor *ThingSpeak* enviar um *tweet* informativo, agora, ao receber uma medida de umidade abaixo de 60%, utilizaria um sistema de irrigação que acionaria uma válvula de vazão solenoide para permitir a passagem de água no sistema, que conectaria uma fonte de água (ou torneira) a uma mangueira (ou tubulação) até chegar na planta.

A apresentação do desenvolvimento deste protótipo é dividida nas seguintes seções: i) Componentes utilizadas, ii) Diagrama do dispositivo iii) Código de controle e iv) Protótipo do dispositivo.

### Componentes Utilizados

Para simular um sistema de irrigação, foram usados os componentes descritos abaixo:

- NodeMCU;
- Válvula de Vazão Solenóide 12V;
- Fonte 12V;
- Sensor de Umidade do solo Higrômetro;
- Resistor 1K ohms;
- Resistor 2K ohms;
- Protoboard;
- Transistor BC337;
- Diodo 1N4007;
- Jumpers;
- Mangueira;
- Galão de água 20L;

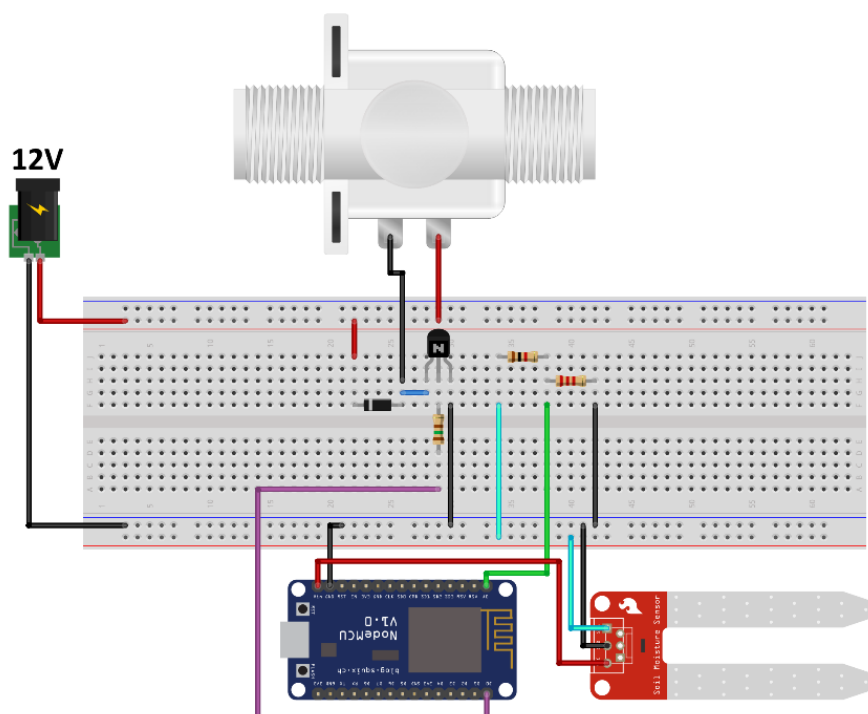
### Diagrama de Conexão

O diagrama de conexão entre as componentes mencionadas, definem o circuito de controle. A Figura 16, ilustra o diagrama para este circuito. A válvula solenóide está conectada a uma fonte de 12V, no entanto, a sua ativação é controlada utilizando-se um transistor BC337, que funciona como um interruptor eletrônico que permite a passagem de corrente sempre que um pequeno sinal é enviado pelo controlador do NodeMCU ao pino central do transistor (linha roxa). Vale observar que o controle da válvula é semelhante ao de um motor DC. De

maneira semelhante, quando essa válvula é fechada é necessário um mecanismo de proteção para a corrente inversa. Ou seja, uma vez desativada, devido à inércia, ela vai continuar funcionando por um período de tempo, e nesse curto espaço, ela pode gerar uma corrente elétrica contrária, que pode voltar para a placa NodeMCU e danificá-la. Devido a isso, o circuito utiliza um diodo como proteção.

O transistor tem três pernas: base, coletor e emissor. Juntas elas funcionam como um interruptor, ou seja, aplicando tensão à base, a corrente flui entre o emissor e o coletor, permitindo que uma corrente maior do que a inicialmente permitida, possa circular. Já o diodo é uma válvula de sentido único, que vai impedir que a corrente volte e garantir que o circuito funcione corretamente.

Como a tensão de saída do higrômetro, que registra a umidade do solo (linha azul clara) fornece 5V, mais uma vez, estão sendo usados resistores para regular a tensão, já que o NodeMCU aceita no máximo 3.3V, regulando assim a tensão de entrada na porta analógica (A0) do NodeMCU.(linha verde).



**Figura 16** – Diagrama de Irrigação autônoma de Planta IoT – Fonte [12]

### Código de Controle

O código de controle é muito semelhante ao código apresentado na seção anterior, apenas será necessário substituir a geração do *tweet* como resposta ao evento de ter o solo com baixa umidade pela ativação da válvula solenoide. Neste caso, o servidor deverá enviar uma mensagem de ativação ao NodeMCU. Por motivos de espaço essa parte do projeto não será apresentada.

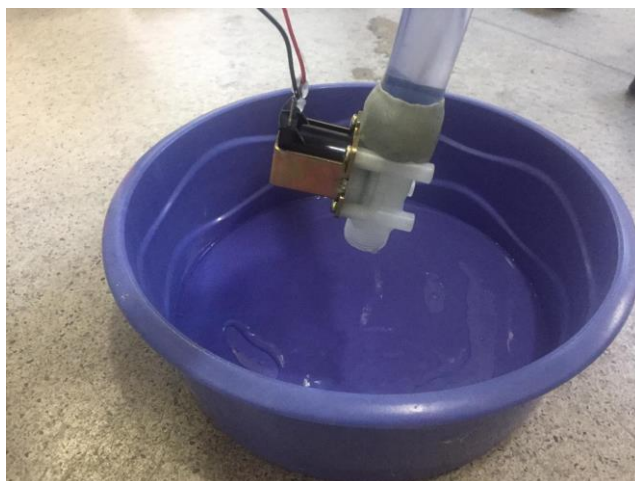
### Protótipo do Dispositivo

Para poder viabilizar o teste do dispositivo de irrigação, foi montado um sistema para simular uma fonte de água, representado por um galão de 20 litros, conectado a uma mangueira, que na sua outra extremidade foi conectada na válvula solenoide, conforme ilustrado na Figura 17. Para a construção deste protótipo o galão de água foi perfurado e a mangueira fixada ao galão e à válvula utilizando como material de vedação massa durepoxi, que funcionou perfeitamente com quase nenhum vazamento de água.



**Figura 17** – Protótipo do Sistema de Irrigação – Fonte: A autora

Com o galão cheio pela metade, o protótipo mostrou funcionamento adequado. Sempre que energizada, a válvula solenoide liberou a passagem de água, como mostra a imagem da Figura 18. Embora o volume de água liberado fosse adequado para regar a planta, observa-se que outras aplicações podem demandar uma bomba de água para aumentar o fluxo.



**Figura 18** – Protótipo do Sistema de Irrigação – Fonte: A autora



### 5.3.- Testes de comunicação com o Braço Robótico

Paralelamente aos trabalhos descritos anteriormente, foram realizados alguns testes de comunicação remota com o dispositivo do braço robótico, desenvolvido no primeiro ano da Iniciação Científica, e composto por quatro servo motores controlados através de módulos *joysticks*. Como proposta de melhoria, pretende-se mudar esse controle para que seja feito via internet, através de algum dispositivo como computador ou celular. Para isso, foram feitos alguns testes a fim de descobrir qual a placa ideal nesse caso.

Inicialmente foi estudada e testada a placa ESP8266 ESP-01 e, logo no início do processo foi observado que para uso dessa placa, é preciso que nela seja gravado um firmware para que seja possível a conexão à rede WiFi e que ela seja utilizada junto ao Arduino.

Nesse momento foi necessário o uso do Adaptador USB para Módulo ESP8266, descrito na sessão 4.2. Após alguns testes, verificou-se que o adaptador, na forma como comercializado, somente pode ser utilizado para conectar a placa ao computador. No entanto, para poder programá-la e necessária a conexão dos pinos GND e GPIO0, através de um fio com uma ponta soldada em cada um deles, conforme a figura 16. Somente com essa alteração no hardware é possível ativar o modo programação da placa ESP-01.



**Figura 19** – Adaptador USB com modo de programação ativado – Fonte: A autora

Uma vez ativado o modo programação, basta conectar o adaptador ao computador e fazer a instalação do firmware desejado, de acordo com [13] e [14]. São diversos os comandos AT disponíveis, mas os necessários para esse caso foram os que permitem fazer a configuração WiFi através do monitor serial da IDE do Arduino. São eles: AT+GMR para verificar a versão do firmware instalada, AT+CWMODE? Para verificar em que modo a placa está trabalhando, AT+CWLAP para fazer a conexão à rede, e por último, AT+CIFSR para verificar o número de IP fornecido pela rede.

Essa etapa de configuração foi bem sucedida, mas toda vez que o ESP-01 era desconectada do computador, todas as informações gravadas na placa eram apagadas, impedindo continuar com sua utilização.

Por isso, em consenso com o orientador, foi decidido que, apesar da placa ESP-01 ser uma opção mais econômica, ela não parece ser uma opção prática e confiável. Além disso, por já ter sido usado no projeto principal da Planta IoT e, por ter funcionado corretamente, o NodeMCU se traduz na melhor opção para controle dos servos, visto que ficaríamos muito tempo presos à uma única placa, sem sucesso.

Na continuação das pesquisas, foi visto que só seria possível controlar um único servo utilizando a placa NodeMCU, uma vez que ele possui um único pino analógico e não é capaz de alimentar os quatro servos dispostos no braço robótico. Por fim, chegamos à conclusão de que embora seja possível fazer esse controle remotamente, usando a placa NodeMCU, seria necessária uma pesquisa mais aprofundada para saber como fazer a alimentação de energia dos servomotores de forma independente a essa placa, para evitar a sua sobrecarga.

## **6.- Conclusões**

Ao começar a pesquisa, o orientador disponibilizou livros sobre Internet das Coisas, além de compartilhar com a orientada artigos e sites sobre esses temas. A primeira etapa visou focar no estudo das placas utilizadas nessa pesquisa, além das linguagens de programação necessárias para desenvolvimento web. Também houve uma familiarização com a área de redes, incluindo o estudo e conhecimento dos protocolos exigidos para estabelecer uma comunicação entre um dispositivo e a internet.

Terminada essa parte, foi decidido em acordo entre orientador e orientada que o projeto a ser desenvolvido seria o da Planta IoT, que consiste em um sistema de monitoramento e irrigação autônomo. O desenvolvimento do projeto foi concluído com sucesso, de acordo com as etapas especificadas no plano de trabalho. A orientada conseguiu, através dos conhecimentos adquiridos ao longo da pesquisa, além de montar todo o protótipo e circuito da planta, fazer com que ela se comunique com a internet através de um servidor online, enviando dados para o mesmo. Além disso, foi estudada uma maneira de melhorar o projeto e concluímos que é possível fazer com que a planta consiga ser regada sem interferência humana, através do uso de uma válvula solenoide. As etapas do plano de trabalho referentes à construção do dispositivo, foram em sua grande maioria cumpridas de maneira satisfatória.

Em relação ao braço robótico, tínhamos a expectativa de mudar o controle de seus motores para uma via remota, podendo ser usado um dispositivo móvel, por exemplo. Devido à problemas técnicos de alimentação e falta de recursos no laboratório, foi descoberta a maneira ideal de implementação desse controle via web, porém ainda não foi possível conectar os quatro servos instalados no braço. Ainda assim, o estudo de qual placa seria utilizada e de como fazer essa conexão, também foi concluído.



## **7.- Referências**

- [1] CISCO. A Internet das Coisas - Como a próxima evolução da Internet está mudando tudo**
- [2] MATTERN, Friedman. From the Internet of Computers to the Internet of Things**
- [3] <https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01/>**
- [4] <http://blog.eletrogate.com/nodemcu-esp12-introducao-1/>**
- [5] <http://blogmasterwalkershop.com.br/embarcados/wemos/conhecendo-a-wemos-d1/>**
- [6] <https://www.vidadesilicio.com.br/sensor-umidade-solo-higrometro>**
- [7] <https://www.filipeflop.com/produto/valvula-de-vazao-solenoides-agua-12vdc/>**
- [8] <https://www.filipeflop.com/produto/protoboard-400-pontos/>**
- [9] <https://www.creatroninc.com/product/8-pin-arduino-jumper-wire/>**
- [10] <https://www.robotics.org.za/RES-330E-50>**
- [11] <https://www.filipeflop.com/produto/adaptador-usb-para-modulo-wifi-esp8266-esp-01/>**
- [12] <https://www.filipeflop.com/blog/planta-iot-com-esp8266-nodemcu-parte-4/>**
- [13] <http://binarycodec.blogspot.com/2016/03/atualizar-o-firmware-do-esp8266.html>**
- [14] <https://www.youtube.com/watch?v=Na7CSbq467M>**

## **8.- Perspectivas de continuidade ou desdobramento do trabalho**

Na continuação da pesquisa, conforme descrito no plano de trabalho proposto na solicitação de renovação, propõe-se o aprimoramento de dois dispositivos, desenvolvidos em anos anteriores: Braço Robótico e Sistema de monitoramento da Planta IoT. Deverão ser estudadas possíveis melhorias na interface de controle dos dispositivos, na comunicação remota, no sistema de alimentação de energia, assim como no controle otimizado dos motores e válvulas. Com esta finalidade, propõe-se introduzir o uso de uma placa Raspberry Pi3 B+, recentemente adquirida, que conta com maior poder de processamento e propõe-se estudar a integração entre as plataformas disponíveis: Arduino, NodeMCU e Raspberry Pi, de modo a tirar melhor proveito das capacidades de cada uma delas.

## **9.- Participação em congressos e trabalhos publicados ou submetidos e outras atividades acadêmicas e de pesquisa**

Como parte da disciplina de inteligência artificial ministrada no semestre 2017-2, a aluna bolsista fez modificações em um veículo robótico incorporando um sensor de ultrassom de maneira a detectar a presença de obstáculos e corrigir a sua direção mudando de trajetória, sendo assim capaz de introduzir inteligência ao dispositivo.

## **10.- Data e assinatura do bolsista**

30 de Abril de 2019,



## **11.- Data e assinatura do orientador**

30 de Abril de 2019,

