



UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

DISCIPLINA DE TÓPICOS ESPECIAIS EM ENGENHARIA DE TELEINFORMÁTICA I



FRANCISCO CARLOS FREIRE NUNES JUNIOR - 371802

LAILSON AZEVEDO DO REGO - 371828

SAYONARA SANTOS ARAÚJO - 371869

**FORTALEZA**

**2016**

Controle de versão		v3.3
Data	Modificação	Autor
25/10	Criação do documento	Carlos, Lailson e Sayonara
17/11	Adição do tópico de Comunicação em Resultados	Carlos
08/12	Adição da seção Sensor de Temperatura e Umidade	Sayonara
08/12	Adição do seção Aplicação	Lailson
27/12	Adição da seção Prova de Conceito	Carlos
28/12	Adição das seções Sensor de Fluxo de Água e Sensoriamento	Sayonara
28/12	Adição de sessões Válvula Solenóide e Atuação e Monitoramento de Vazão	Lailson e Sayonara

## SUMÁRIO

<b>1. Introdução</b>	<b>3</b>
<b>2. Material</b>	<b>4</b>
<b>3. Objetivo</b>	<b>5</b>
<b>4. Fundamentação Teórica</b>	<b>6</b>
4.1. Arduino	6
4.2. Módulo Sensor de Chuva	7
4.2.1. Especificações	7
4.2.2. Funcionamento	7
4.3. Módulo Sensor de Umidade	8
4.3.1. Especificações	8
4.3.2. Funcionamento	8
4.4. Módulo Relé	9
4.4.1. Especificações	9
4.4.2. Funcionamento	10
4.5. Sensor de Temperatura e Umidade	10
4.5.1. Especificações	11
4.5.2. Funcionamento	11
4.6. Sensor de Fluxo de Água	11
4.6.1. Especificações	12
4.6.2. Funcionamento	12
4.7. ESP8266	13
4.7.1. Introdução	13
4.7.2. Especificações	13
4.7.3. Tecnologia de baixo consumo	14
4.7.4. MCU	14
4.7.5. Programação	14
4.7.6. Funcionamento	15
4.8. Válvula Solenóide	16
4.8.1 Especificações	16
4.8.2 Funcionamento	16
<b>5. Resultados</b>	<b>18</b>
5.1. Sensoriamento	19
5.2. Atuação e monitoramento da vazão	20
5.3. Comunicação	21
5.3.1. Servidor	21
5.3.2. Módulo cliente	22
5.4 Aplicação	22
5.5 Prova de conceito	24
5.5.1 Módulos de teste do sistema	24
5.5.1.1 Módulo de sensoriamento	24
5.5.1.2 Módulo de atuação	26
5.5.2 Database e gráficos de monitoramento	27
<b>6. Bibliografia</b>	<b>33</b>

## **1. INTRODUÇÃO**

O manejo de água é um tema preocupante. Em grandes plantações, sistemas de pivôs abastecem áreas de cerca de 250 mil hectares. Logo, grandes investimentos estariam correndo um risco de desabastecimento e desperdício caso não exista um bom controle de fornecimento de água nessas regiões.

O problema de desperdício de água persiste em ambientes caseiros como em jardins e hortas. Dessa forma, o uso de água de irrigação em ambientes caseiros foi escolhido como ponto de partida desta pesquisa devido a facilidade de abstração e por ser um dos segmentos que mais utiliza recursos hídricos.

Diante disso, For Garden's Sake tem como objetivo montar um sistema de monitoramento de jardim ou plantação, focando no controle do fornecimento da água de irrigação, um momento e em quantidade adequados. Portanto, o sistema fornecerá mais água em períodos de seca, e evitará desperdício em períodos chuvosos. Além disso, a gestão será feita através de uma plataforma IoT que possibilitará a consulta de informações coletadas em tempo real.

## **2. MATERIAL**

- 2 x Arduino Uno;
- Computador;
- Módulo sensor de chuva;
- Módulo sensor de fluxo de água;
- Módulo sensor de temperatura e umidade;
- Módulo sensor de umidade do solo;
- Módulo relé;
- 2 x Módulo WiFi ESP8266 ESP-01;
- 2 x Protoboard;
- Resistores diversos;
- Válvula Solenóide 220V.

### **3. OBJETIVO**

O sistema funcionará em quatro etapas:

1. A abertura da torneira de irrigação se dará por meio de uma válvula solenóide controlada por um Arduino, que será responsável por obter dados do consumo a partir de um sensor de vazão acoplado à torneira. Os comandos de ligar ou desligar serão recebidos do servidor, e as informações de consumo serão enviadas ao servidor.
2. O monitoramento da plantação será realizado por um conjunto de sensores formado por sensor de chuva, sensor de umidade e temperatura (ambiente), além de sensor de umidade do solo. Estes serão controlados por um Arduino responsável por comutar as informações e enviá-las ao servidor.
3. O servidor será composto por um computador que receberá as informações coletadas pelos sensores, e as armazenará em um banco de dados. Através dessas informações, o servidor decidirá quando irrigar a plantação enviando uma tomada de decisão ao Arduino responsável por controlar a torneira.
4. O cliente é composto por uma aplicação móvel que mostrará ao usuário as informações trocadas com o servidor sobre consumo e meio ambiente (temperatura e umidade da plantação). Além disso, a aplicação possibilitará o desligamento do sistema de irrigação.

## 4. FUNDAMENTAÇÃO TEÓRICA

### 4.1. Arduino

O Arduino é uma placa fabricada na Itália com um microcontrolador Atmel AVR, suporte de entrada/saída embutido e uma linguagem de programação essencialmente C/C++, bastante utilizado como plataforma de prototipagem eletrônica de Hardware livre cujo objetivo é tornar a robótica mais acessível.

A fonte de alimentação recebe energia externa por uma tensão de, no mínimo, 7 volts e máximo de 35 volts com corrente mínima de 300mA. A placa e demais circuitos funcionam com tensões entre 5 e 3,3 volts.

As unidades são constituídas por um microcontrolador Atmel AVR de 8 bits, pinos digitais e analógicos de entrada e saída, entrada USB que permite conexão com computadores, entrada serial e código aberto. Por possuir código aberto, há várias modificações que deram origem a derivados “ino”, tais como Marminino, Produino, Garagino, etc.

O Arduino pode ser combinado com outras extensões chamadas shields que são responsáveis por gerar recursos de rede à placa. Embutido na placa há ainda um firmware carregado na memória da placa controladora que aceita Windows, Linux e Mac OS X.

As aplicações com Arduino abrangem diversas áreas tais como designer, arquitetura, engenharia, agronomia, etc; com a placa é possível enviar ou receber informações de basicamente qualquer sistema eletrônico, como desligar uma válvula solenóide ao detectar a presença de chuva, e abrir as janelas de um escritório de acordo com a intensidade da luz do sol e temperatura ambiente.



Figura 4.1.1: Arduino Uno.

## **4.2. Módulo Sensor de Chuva**

O módulo sensor de chuva YL-83 é composto por um sensor de chuva e uma placa de controle. Este módulo pode ajudar no monitoramento condições climáticas, detectando gotas de chuva ou neve.

### **4.2.1. Especificações**

- Tensão de Operação: 3,3-5v
- Corrente de Saída: 100mA
- Potenciômetro (trimpot)
- Saída Digital e Analógica
- Led indicador para tensão
- Led indicador para saída digital
- Comparador LM393
- Dimensões Sensor de Chuva: 5x4 cm
- Dimensões Placa de Controle: 2,1x1,4 cm

### **4.2.2. Funcionamento**

O sensor de chuva é revestido de uma malha condutora, desenhada no formato de linhas paralelas e de modo que linhas adjacente estão isoladas umas das outras. Então, quando o sensor está energizado, as linhas apresentam sinal positivo e negativo de forma alternada. Assim, quando cai um líquido condutor no sensor, fechando um circuito entre uma linha positiva e uma negativa, a corrente passa pela malha.

A placa de controle é constituída por um comparador de tensão, um potenciômetro para ajuste de sensibilidade, pinos de alimentação, uma saída digital e uma saída analógica, com valores de 0 (zero) a 1024. Basicamente, a placa amplifica o sinal lido pelo sensor e converte o sinal analógico em digital.

Com isso, quando está chovendo, o módulo apresenta o sinal digital em estado baixo e o sinal analógico mais próximo de zero; no clima seco, o módulo apresenta o sinal digital em estado alto e o sinal analógico mais próximo de 1024.



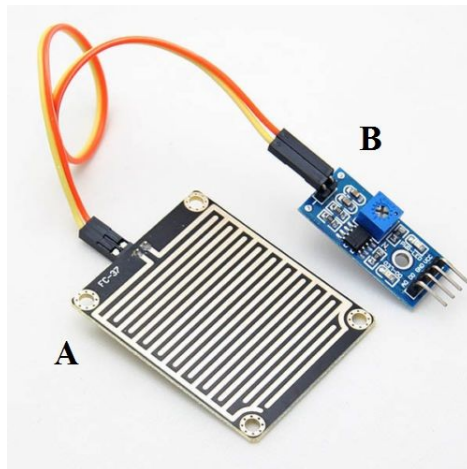


Figura 4.2.1: Módulo sensor de chuva. A - Sensor de chuva. B - Placa de controle.

### 4.3. Módulo Sensor de Umidade

O sensor de umidade do solo higrômetro é usado para detectar variações de umidade no solo, de modo que sua saída fica em estado alto e baixo quando o solo está seco e úmido, respectivamente. Há dois tipos de saída no sensor de umidade, a analógica A0 que permite melhor resolução possível e a saída digital D0 que é regulada por um potenciômetro interno ao sensor; este potenciômetro permite que o limite entre seco e úmido seja ajustado.

#### 4.3.1. Especificações

- Tensão de Operação: 3,3-5v
- Potenciômetro
- Saída Digital e Analógica
- Led indicador para tensão (vermelho)
- Led indicador para saída digital (verde)
- Comparador LM393
- Dimensões PCB: 3x1,5 cm
- Dimensões Sonda: 6x2 cm

#### 4.3.2. Funcionamento

O sensor de umidade do solo consiste em duas partes: uma sonda que entra em contato com o solo e um pequeno módulo contendo um chip comparador LM393.

Os dois terminais da sonda que entram em contato com o solo, quando energizados, conduzem uma tensão conforme a umidade da terra.

A placa de controle, semelhante a do sensor de chuva, é constituída por um comparador de tensão, um potenciômetro para ajuste de sensibilidade, pinos de alimentação, uma saída digital e uma saída analógica, com valores de 0 (zero) a 1024. Basicamente, a placa amplifica o sinal lido pelo sensor e converte o sinal analógico em digital.

Dessa forma, quando o solo está úmido, o módulo apresenta o sinal digital em estado baixo e o sinal analógico mais próximo de zero; já quando o solo apresenta umidade baixa, o módulo apresenta o sinal digital em estado alto e o sinal analógico mais próximo de 1024.

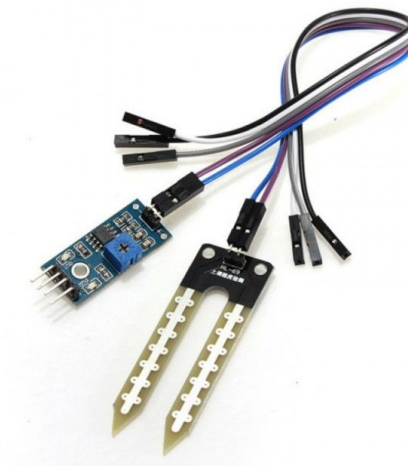


Figura 4.3.1: Módulo sensor de umidade do solo higrômetro.

#### **4.4. Módulo Relé**

O módulo relé, utilizado no projeto, permite uma integração com uma variedade de microcontroladores como Arduino, AVR, PIC, ARM. A partir das suas saídas digitais pode-se controlar cargas altas e dispositivos como motores AC ou DC, eletroímãs, solenóides e lâmpadas incandescentes.

##### **4.4.1. Especificações**

- Tensão de operação: 5 DC (VCC e GND)
- Tensão de sinal: TTL - 5 DC (IN)
- Corrente típica de operação: 15~20mA
- Terminais: NA, NF e o Comum
- Capacidade: 30V DC a 10A ou 250V AC a 10A
- Indicador LED de funcionamento
- Dimensões: 43mm (L) x 17mm (C) x 19mm (H)

#### 4.4.2. Funcionamento

Os relés são chaves eletromecânicas usadas para o acionamento de cargas de alta tensão ou corrente a partir de um circuito de baixa tensão. Um relé é, basicamente, constituído por um eletroímã (bobina), armadura de ferro móvel, conjunto de contatos, mola de rearme e terminais. Quando a bobina é percorrida por uma corrente, é criado um campo magnético, que irá atuar sobre a armadura, atraindo-a e fechando os contatos. Ao cessar a corrente da bobina, o campo magnético deixa de existir, e os contatos voltam à posição original.

O módulo utilizado apresenta pinos de alimentação (VCC e GND); o pino IN, que recebe o sinal para comutar a chave; o contato NC, que é normalmente fechado, ou seja, abre quando a bobina recebe corrente, e o contato NO, que é normalmente aberto, fechando-se quando a bobina recebe corrente.



Figura 4.4.1: Módulo relé.

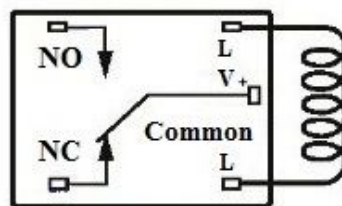


Figura 4.4.2: Esquemático do relé.

#### 4.5. Sensor de Temperatura e Umidade

O DHT11 é um sensor composto de temperatura e umidade relativa do ar, capaz de medir temperaturas de 0 a 50 Celsius e umidade na faixa de 20 a 90% UR. O sensor possui 4 pino, um apenas para a leitura tanto da temperatura e quanto da umidade. O componente também apresenta baixo consumo de potência e capacidade de transmissão de sinal de até 20

metros. Com isso, o DHT11 é comumente empregado junto à placa de desenvolvimento Arduino para aplicações como desumidificador, estações meteorológicas, regulador de umidade, etc.

#### **4.5.1. Especificações**

- Modelo: DHT11
- Faixa de medição de umidade: 20 a 90% UR
- Faixa de medição de temperatura: 0° a 50°C
- Precisão de umidade:  $\pm 5\%RH$
- Precisão de temperatura:  $\pm 2^{\circ}C$
- Tensão de operação: 3-5VDC
- Corrente: 200uA a 500mA em medição e 100uA a 150 uA em espera.
- Tempo de espera: 1s
- Dimensões: 23 x 12 x 5mm

#### **4.5.2. Funcionamento**

O sensor é constituído por um termistor do tipo NTC e o sensor de umidade HR202 conectados a um microcontrolador de 8 bits. O circuito interno faz a leitura dos sensores e transmite essa leitura ao microcontrolador. Os valores da temperatura e da umidade podem ser obtidos por uma única saída digital, que garante uma alta confiabilidade e estabilidade a longo prazo.

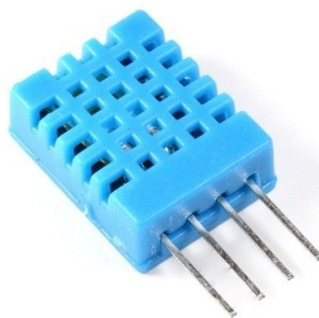


Figura 4.5.1: Sensor DHT11.

#### **4.6. Sensor de Fluxo de Água**

O sensor YF-S201 é indicado para realizar verificação de o consumo de água em casas, prédios. Este sensor faz uso do Efeito Hall, que consiste na criação de diferença de

potencial em um semicondutor situado em um campo magnético perpendicular ao fluxo de corrente.

#### 4.6.1. Especificações

- Modelo: YF-S201
- Tipo de sensor: Efeito Hall
- Tensão de operação: 5-24V
- Corrente máxima: 15mA (5V)
- Faixa de fluxo: 1-30L/min
- Pressão máxima: 2,0 MPa
- Pulsos por litro: 450
- Frequência (Hz) = Fluxo(L/H)/7,5
- Temperatura de trabalho: -25 a 80°C
- Exatidão: 10%
- Comprimento do cabo: 15cm
- Dimensão conexão: 1/2"
- Dimensão diâmetro interno: 0,78"
- Dimensão externa: 2,5" x 1,4" x 1,4"

#### 4.6.2. Funcionamento

O sensor de fluxo de água contém, em seu interior, uma turbina com um ímã acoplado, paralelo ao sensor Hall. Quando a água passa pela turbina, a hélice com o ímã passa pelo sensor Hall, detectando cada volta da hélice. A cada volta detectada, é transmitido um pulso pelo fio (amarelo) de sinal ao Arduino.



Figura 4.6.1: Sensor de fluxo de água.

## 4.7. ESP8266

### 4.7.1. Introdução

O módulo ESP8266-01 é um conjunto SOC (System On Chip) de alta performance e alta integração sem fio, de baixo custo, projetado para plataformas móveis, ocupando pouco espaço e capaz de operar com restrições de energia.

O módulo possui memória ROM e SRAM on-chip, nos quais é armazenado e executado o firmware, que implementa TCP/IP e todo protocolo 802.11 b/g/n/e/i WLAN MAC e WiFi Direct. Podendo funcionar como provedor de conexão sem fio para outro controlador ou microprocessador, ou operando como aplicação através do seu MCU.

### 4.7.2. Especificações

- MCU integrado 32-bit de baixo consumo
- Conversor integrado 10-bit ADC
- Integrado com pilha de protocolo TCP/IP
- Tensão de operação: 3,3V
- Suporte à redes: 802.11 b/g/n
- Alcance: 90m aproximadamente
- 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

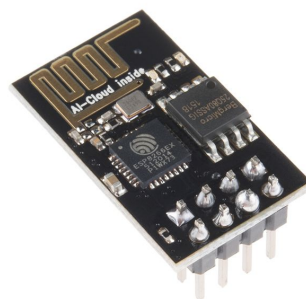


Figura 4.7.1: Módulo ESP8266-01.

#### 4.7.3. Tecnologia de baixo consumo

Projetado para aplicações móveis, vestíveis e de internet das coisas, e com o objetivo de alcançar o menor consumo, o módulo ESP8266 combina várias técnicas e propriedades. A arquitetura de economia de energia opera principalmente em 3 modos: active mode, sleep mode e deep sleep mode.

Paramêtros	Valor	Unidade
Tx802.11b, CCK 11Mbps, P OUT=+17dBm	170	mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm	140	mA
Tx 802.11n, MCS7, P OUT =+13dBm	120	mA
Rx 802.11b, 1024 bytes packet length , -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Modem-Sleep	15	mA
Light-Sleep	0.9	mA
Deep-Sleep	10	uA
Power Off	0.5	uA

Figura 4.6.2: Consumo de acordo com datasheet.

#### 4.7.4. MCU

Embarcado com o microcontrolador Tensilica L106 32-bit e 16-bit RSIC, com clock speed de 80MHz, podendo alcançar o máximo de 160MHz. Com RTOS (Real Time Operation System) habilitado, e com apenas 20% do MIPS ocupados pela pilha WiFi, com possibilidade de utilizar o restante para programação e desenvolvimento.

#### 4.7.5. Programação

É possível embarcar o firmware no ESP8266 através da IDE do Arduino ou com programa de gravação de memória flash por comunicação serial. Para isto, inicializa-se o módulo com o GPIO0 em GND e faz upload do firmware. Com o firmware gravado, o módulo passará executar e manterá o programa em memória não volátil. Após desligar e ligar módulo, é necessário manter o pino GPIO0 em alto ou desconectado para manter o firmware anteriormente gravado no dispositivo.

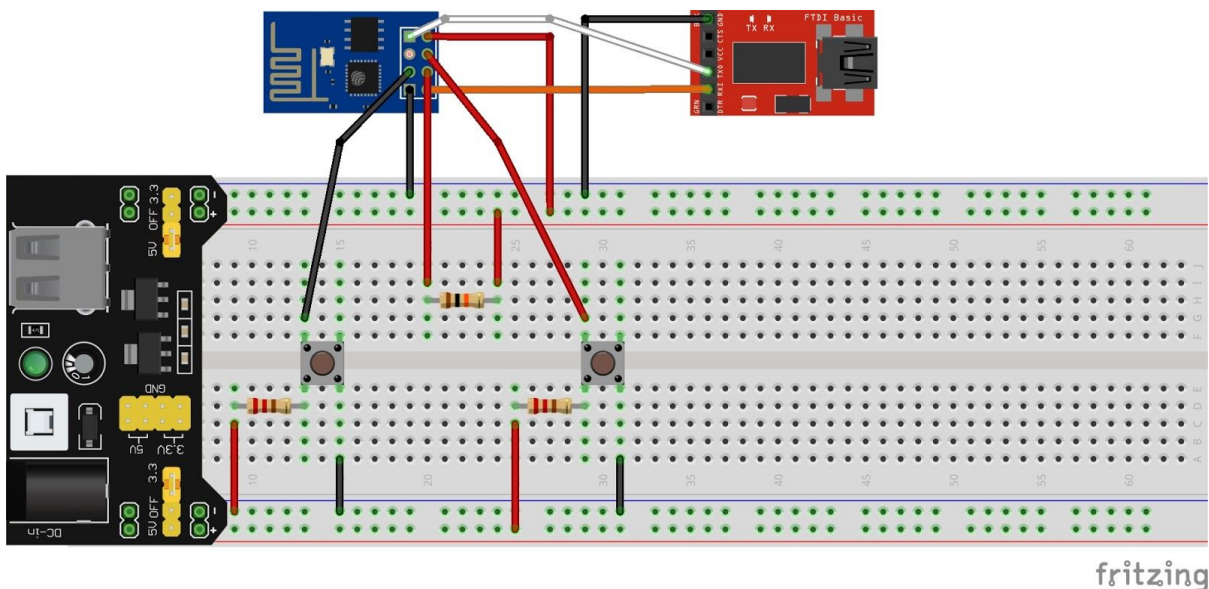
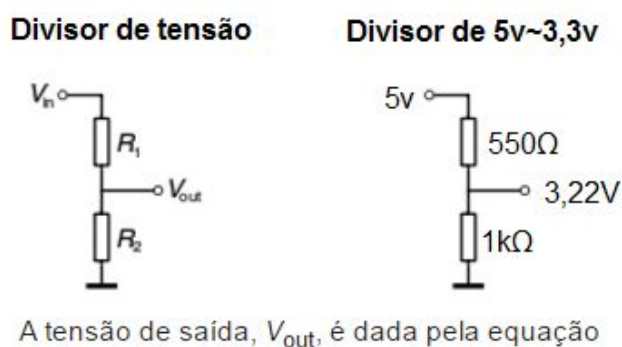


Figura 4.7.3: Ligação para programação com conversor serial.

#### 4.7.6. Funcionamento

Após gravação do firmware, para utilizar o módulo em conjunto com Arduino, é necessário utilizar uma fonte de alimentação externa, pois as portas do ATmega328 suportam no máximo 50mA e o ESP8266 pode consumir 200mA. Para comunicar com o módulo, utiliza-se as portas RX (Read) e TX (Transmitte) através de comunicação serial. Como o módulo funciona em 3.3v e os pinos digitais da plataforma Arduino trabalham a 5v, é possível fazer uma conversão através de um divisor de tensão.



$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Figura 4.7.4: Divisor de tensão de 5v para ~3,3v.



## **4.8 Válvula Solenóide**

A válvula solenóide é um equipamento que tem utilizações em diversas aplicações envolvendo água, ar, gases, GNV, GLP, óleo, filtros de manga e outros fluídos.

### **4.8.1 Especificações**

- Corpo termoplástico
- Filtro plástico removível
- Partes metálicas constituídas de aço zincado
- Membrana borracha (padrão) e silicone (atóxica)
- Terminais: latão
- Pressão de operação: de 0,2 Kgf/cm<sup>2</sup> à 8 Kgf/cm<sup>2</sup>
- Temperatura máxima do líquido 60°C
- Rigidez dielétrica: 1.500Vca - 1 min
- Saída de água: conexão para mangueira
- Disposição geométrica em relação à entrada: 90°
- Número de entradas/Saídas: 1 entrada / 1 saída
- Vida útil: 50.000 operações
- Faston 6,3mm x 0,8mm
- Faston 4,8 x 0,8mm
- Tensão Bobina 220Vca / 50-60Hz

### **4.8.2 Funcionamento**

A válvula solenóide é formada por duas partes funcionais (Figura 4.8.1): o corpo, contendo o orifício(2); e uma bobina solenóide com seu núcleo(1), constituída de um fio enrolado através de um cilindro. Quando a corrente passa por este fio, ela gera uma força no centro da bobina solenóide fazendo com que o êmbolo da válvula seja acionado, criando o sistema de abertura e fechamento, ou seja, esse sistema de abertura e fechamento é proporcionado pela movimentação do núcleo que é atraído pela solenóide quando a bobina é energizada.

O corpo da válvula possui um disco que é posicionado para permitir, ou não, a passagem de fluido quando sua haste é acionada pela força da bobina. Esta força é que faz o pino ser puxado para o centro da bobina, permitindo a passagem do fluído.

O processo de fechamento da válvula solenóide ocorre quando a bobina perde energia, pois o pino exerce uma força através de seu peso e da mola que tem instalado. Em geral, o funcionamento da válvula solenóide não depende da pressão da linha ou da vazão, e a válvula opera de zero ao valor máximo de pressão (Kgf/cm<sup>2</sup>) específico de cada uma.

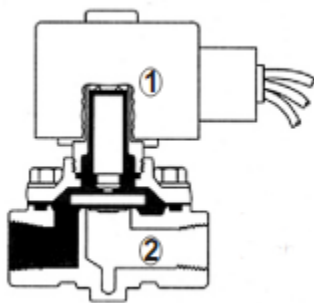


Figura 4.8.1: Válvula de ação direta normalmente fechada.

Para o projeto, um dos cabos da solenóide é ligado diretamente à energia e outro é ligado em comum ao relé responsável por fazer o chaveamento do circuito, permitindo a passagem, ou não, de corrente, bem como a passagem, ou não, de fluidos.

## 5. RESULTADOS

### 5.1. Sensoriamento

O sensoriamento da plantação é realizada pelos sensores de chuva, umidade do solo e temperatura conectados a uma placa Arduino. Então, todos os dados coletados desses sensores são repassados ao servidor através do módulo WiFi ESP8266 também conectado à placa.

A seguir, são apresentadas partes do script que determina o funcionamento dos sensores e a comunicação com o módulo WiFi.

```
#include <DHT.h>

#define pinoUmidade A0
#define pinoChuva A1
#define pinoTemperatura A3
#define dhttipo DHT11

int valorUmidade = 0; //Variavel para armazena o valor do sensor de umidade
int valorChuva = 0; //Variavel para armazena o valor do sensor de chuva
DHT dht(pinoTemperatura, dhttipo);
float valorTemperatura = 0; //Variavel para armazena o valor do sensor de temperatura
String t1 = ";umidade=";
String t2 = ";chuva=";
String t3 = ";temperatura=";
String mensagem;

void setup()
{
    // Define os pinos dos sensores como entrada
    pinMode(pinoUmidade, INPUT);
    pinMode(pinoChuva, INPUT);

    Serial.begin(115200);
    dht.begin();
}
```

Figura 5.1.1: Código do Arduino do sensoriamento - setup.

Inicialmente no código, foi incluída biblioteca DHT[12], necessária para o funcionamento do sensor de temperatura. Com o uso dessa biblioteca, cria-se uma variável do tipo DHT, indicando o modelo do sensor e a pino do Arduino que recebe os valores do dispositivo DHT. Além disso, são feitas as declarações dos pinos dos sensores e das variáveis que armazenam os valores lidos e compõem a mensagem passada ao ESP8266.

```

void loop()
{
    valorChuva = analogRead(pinoChuva);

    valorUmidade = analogRead(pinoUmidade);

    valorTemperatura = dht.readTemperature();

    mensagem = t1 + valorUmidade + t2 + valorChuva + t3 + valorTemperatura;

    Serial.println(mensagem);

    delay(10000);
}

```

Figura 5.1.2: Código do Arduino do sensoriamento - loop.

Na rotina do código, adquirem-se os valores dos pinos dos sensores de chuva, umidade de solo e temperatura. Depois, concatenam-se todos os valores e as Strings na variável “mensagem”, formando uma sentença com a seguinte estrutura:

- “;chuva:<valor obtido pelo sensor de chuva>;umidade:<valor obtido pelo sensor de umidade>;temperatura:<valor obtido pelo sensor de temperatura>”.

Essa mensagem é enviada ao ESP8266 pela porta serial do Arduino composta pelos pinos digitais 0 (RX) e 1 (TX), onde os pinos TX e RX do módulo Wifi estão conectados respectivamente. Após o envio da mensagem, a rotina se repete a cada 10 segundos.

## 5.2. Atuação e monitoramento da vazão

A seguir, são apresentadas partes do script que determina o monitoramento da vazão bem como a atuação através do módulo relé.

```

#define pinoVazao 2
#define pinoRele 4

int pulsoAtual = 0; //Variavel para armazenar o sinal do sensor de vazao
int pulsoAnterior = 0; //Variavel para armazenar o sinal do sensor de vazao anterior
float vazao = 0; //Vazao apresentada a cada 1s (L/min)
float mediaVazao = 0; //Vazao apresentada a cada 10s (L/min)
volatile int contaPulsos; //Frequencia do sensor de vazao
int qtdSegundos = 0; //Contador para o envio da mensagem
int tempoMensagem = 10; //Tempo para enviar a mensagem
int inicioIntervalo = 0;
int intervalo = 0; //contador de 1 segundo
String mensagem;

void setup()
{
    pinMode(pinoRele, OUTPUT);
    digitalWrite(pinoRele, HIGH);

    Serial.begin(115200);

    pinMode(pinoVazao, INPUT);
    inicioIntervalo = millis();
}

```

Figura 5.2.1: Código do Arduino do atuador - setup.

Inicialmente, são definidos os pinos da placa Arduino conectados ao sensor de vazão e módulo relé, e são declaradas algumas variáveis importantes para o cálculo da vazão e a comunicação entre a placa e o módulo Wi-fi, como se observa na Figura 5.2.1.

---

```

void loop ()
{
    pulsoAtual = digitalRead(pinoVazao);

    if(pulsoAtual != pulsoAnterior){
        contaPulsos++;
    }

    pulsoAnterior = pulsoAtual;
    intervalo = millis() - inicioIntervalo;

    //Calcula a vazao a cada 1s
    if(intervalo == 1000){
        vazao = contaPulsos * 7.5 / 60; //vazao em litros por minuto
        mediaVazao = mediaVazao + vazao;
        qtdSegundos++;
        inicioIntervalo = millis();
        contaPulsos = 0;
    }

    //Calcula a vazao a cada 10 segundos e envia a vazao pela porta serial
    if(qtdSegundos == tempoMensagem){
        mediaVazao = mediaVazao/tempoMensagem;
        Serial.print(mediaVazao);
        qtdSegundos = 0;
        mediaVazao = 0;
        Serial.flush();
    }
}

```

---

Figura 5.2.2: Código do Arduino do atuador - loop (1).

Em parte da rotina do código, calcula-se a vazão como mostra a figura acima. A cada segundo, é feita uma contagem da quantidade de pulsos, obtendo-se a frequência e vazão de tal momento. Isso é feito com base na razão de 7,5 entre a vazão (em litros por hora) e a frequência, uma característica do sensor de fluxo de 1/2 polegada. Então, todas as vazões obtidas são somadas durante dez segundos.

Desse modo, passados os dez segundos, uma média da vazão é calculada e enviada para o ESP8266 pela porta serial.

```

//Verifica se há algum dado no buffer
if (Serial.available() > 0){
    mensagem = Serial.readString();

    if(mensagem == "IniciaIrrigacao"){
        digitalWrite(pinoRele, LOW);
    }
    if(mensagem == "CancelaIrrigacao"){
        digitalWrite(pinoRele, HIGH);
    }
}

```

Figura 5.2.3: Código do Arduino do atuador - loop (2).

Na Figura 5.2.3, é mostrado a parte do script responsável pela atuação do sistema. Assim, é feita uma simples rotina onde verifica-se a existência de algum dado/comando no buffer que é recebido do servidor, através do ESP8266. O monitoramento é feito através do Monitor Serial; caso a mensagem seja “IniciarIrrigacao” o pino do relé é acionado com um digitalWrite LOW, caso a mensagem seja “CancelaIrrigacao” o pino do relé é acionado com um digitalWrite HIGH pois a bobina fecha quando passa uma corrente, ou seja, o cabo de tensão é ligado ao comando NO (normalmente aberto) do relé. Dessa forma, quando a mensagem “IniciarIrrigacao” aparecer no Monitor Serial, o normalmente comum do relé entra em contato com o comando NO, fechando o circuito o que ocasiona o acionamento da válvula solenóide, dando início a irrigação.

### 5.3. Comunicação

#### 5.3.1. Servidor

O servidor é implementado em Java a partir do código disponibilizado por Andsel no GitHub, intitulado de Moquette.

Construído para responder ao protocolo MQTT, o servidor funciona como Broker recebendo mensagens de Publish e Subscribe, e respondendo aos tópicos em que haja inscritos.

O servidor também é responsável pelo tratamento dos dados recebidos via mensagem, contendo os valores de leitura dos sensores. Com os dados recebidos, é feita uma análise e uma tomada de decisão sobre o ligamento ou desligamento da válvula. A decisão será publicada para o módulo de comunicação conectado ao controlador responsável pela manutenção da válvula.

### **5.3.2. Módulo cliente**

Para comunicação com o servidor é utilizado o módulo ESP8266-01 conectados às placas de Arduino, ficando a cargo do módulo a transmissão e recebimento de mensagens com o servidor. Após obter uma conexão com a rede local, a partir das bibliotecas padrão do ESP8266, é possível trocar mensagens com servidor com a utilização do protocolo de comunicação MQTT, implementado pela biblioteca PubSubClient de Knolleary, disponibilizado pelo GitHub.

Para transmitir mensagens da placa Arduino para o ESP8266, é utilizado uma comunicação serial implementada a partir dos pinos digitais do Arduino com a utilização da biblioteca SoftwareSerial, já disponibilizado por padrão pela IDE, que simula uma comunicação serial a partir dos pinos digitais. Também é possível criar uma comunicação serial diretamente com o ESP8266 que estará em laço esperando ler dados. Ao enviar os dados, o ESP8266, a partir da programação do firmware, fará uma leitura dos dados identificando o conteúdo e transmitirá por WiFi para o servidor. O recebimento de dados pelo Arduino acontece de forma inversa, ao receber dados o ESP8266 escreverá na serial de comunicação e o Arduino, após identificar dados a serem lidos, receberá a mensagem.

Para transmitir dados ao servidor com protocolo MQTT é necessário indicar o tópico que identifica o conjunto do valor lido, no caso o sensor, e a mensagem a ser enviada que será o valor lido. Após a leitura, via serial, da mensagem com o ESP8266 e identificação da mensagem indicando o tópico e valor lido, é feito um Publish ao servidor.

Para receber mensagens do servidor, realiza-se um Subscribe indicando o tópico que se deseja receber mensagens. O módulo responsável por receber mensagens do servidor será o que está ligado ao Arduino conectado à válvula solenóide que, a partir das mensagens publicadas no tópico da válvula de ‘Ligado’ ou ‘Desligado’, como o módulo estará inscrito, se dará um SelfPublish no servidor no tópico indicado e o módulo receberá a mensagem e enviará ao Arduino para realizar a determinada ação.

## **5.4 Aplicação**

O sistema usa o aplicativo IoT MQTT Dashboard desenvolvido em Android pela Nghia TH. O aplicativo tem como objetivo controlar projetos de Internet das coisas usando o protocolo MQTT.



Características do aplicativo:

- Permite múltiplas ligações entre dispositivos;
- Conexão feita por SSL;
- Permite que o administrador/usuário especifique a unidade dos dados;
- Apresenta valores numéricos em um gráfico em tempo real;
- Em sua interface, há disponível vários componentes customizáveis como texto, botão, interruptor, barra de procura, caixa de combinação e seletor de cores.

A seguir é mostrado a visão inicial e final do aplicativo após toda configuração, bem como a descrição de cada imagem:

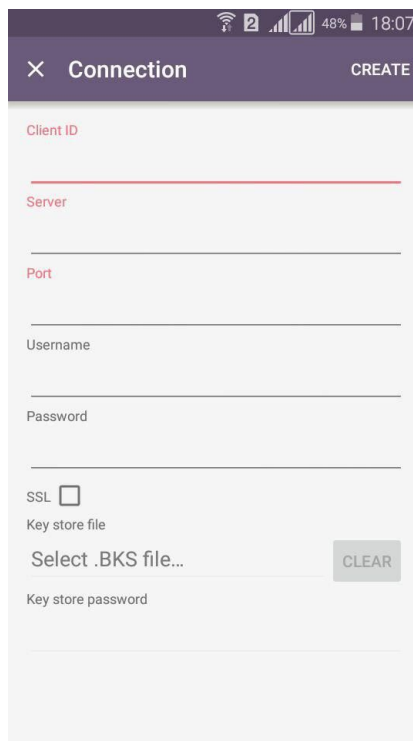


Figura 5.4.1: Visão inicial de conexão do aplicativo.

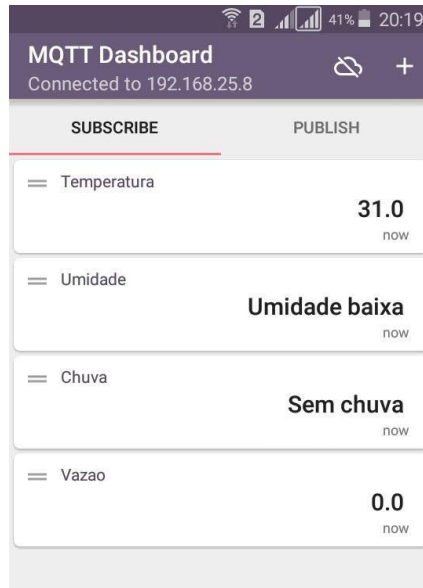


Figura 5.4.2: Visão Subscriber. Os dados do sistema são visualizados.

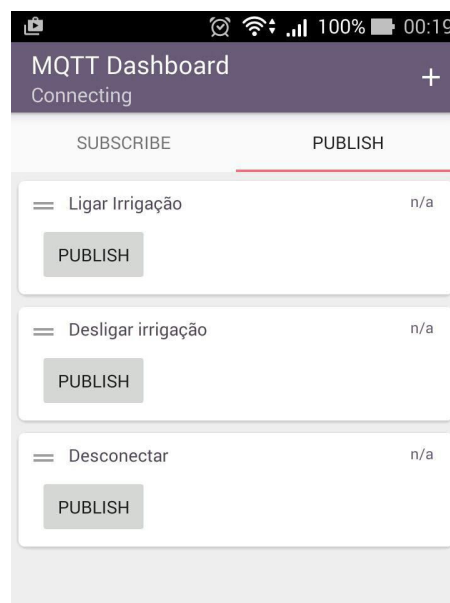


Figura 5.4.3: Visão Publisher. O usuário como atuador.

Na Figura 5.4.1 é mostrada a visão inicial do aplicativo para configuração com o servidor, onde é solicitado o IP do servidor, a porta, usuário e senha. Em seguida, a Figura 5.4.2 representa a visão do usuário inscrito em quatro tópicos: temperatura, umidade, chuva e vazão. E finalmente, na Figura 5.4.3 é mostrado três botões, inscritos no tópico ‘solenóide’. Os botões ‘Ligar’, ‘Desligar’ são responsáveis por ligar e desligar a válvula solenóide, ou seja, ao clicar nestes botões a torneira é acionada. Quanto ao botão ‘Desconectar’, ao ser

clicado, o usuário muda a preferência do sistema quanto ao monitoramento, deixando à cargo de todo o sensoriamento do sistema. Isso acontece pois o sistema é desenvolvido em cima de duas preferências: uma onde o usuário pode comandar a irrigação através do dispositivo móvel e outra onde o sistema é gerenciado através dos sensores instalados.

## **5.5 Prova de conceito**

### **5.5.1 Módulos de teste do sistema**

#### **5.5.1.1 Módulo de sensoriamento**

O módulo de sensoriamento deve ficar próximo da plantação e é constituído por um Arduino com o módulo ESP8266, sensor de temperatura DHT11, sensor de umidade do solo e sensor de chuva YL-83. Para facilitar a mobilidade, o módulo foi montado em uma caixa feita sob medida com máquina de corte a laser e é alimentado por duas baterias recarregáveis 18650 de 3.7v em série. Foi verificado em testes que bateria toda carregada mantém o módulo em funcionamento contínuo por três horas, um tempo pequeno, porém, justificado pela natureza genérica das baterias e o uso de placas de testes para o projeto, além da alta taxa de transmissão de mensagens, sendo a comunicação um dos principais consumidores e a falta de otimização no código para economizar energia.



Figura 5.5.1.1.1: Módulo de sensoriamento visão frontal.



Figura 5.5.1.1.2: Módulo de sensoriamento visão dorsal.

#### **5.5.1.2 Módulo de atuação**

O módulo atuador é o responsável pelo controle da válvula, e nele está ligado um Arduino junto com um ESP8266, um relé que é usado para ligar e desligar a válvula e o sensor de vazão. Este módulo é alimentado por cabo a rede elétrica devido a válvula solenóide ser de 220V, e é conectado a torneira onde há o transiente de água para irrigação.



Figura 5.5.1.2.1: Módulo de atuação visão superior.

### 5.5.2 Database e gráficos de monitoramento

A partir da POC (Proof of Concept), foram realizados testes de funcionamento com o intuito de coletar dados dos sensores e comandos processados pelo servidor, salvando automaticamente em um arquivo de texto o horário e valores representados em decimal. Para ler os dados salvos pelo servidor foi programado um script em Matlab para gerar gráficos do funcionamento automaticamente.

Os dados do servidor são salvos automaticamente a partir de um tempo programado, iniciando com a criação de um arquivo com título que contém a data de geração e salvando os dados a cada término de um segundo tempo programado para cálculo das médias dos valores coletados naquele tempo, salvando em um objeto database e gravando os dados no arquivo gerado, ou caso o arquivo já exista, anexando os dados no arquivo existente.



Nome	Data de modificaç...	Tipo	Tamanho
2016-12-04	04/12/2016 18:21	Arquivo TXT	14 KB
2016-12-05	06/12/2016 01:27	Arquivo TXT	140 KB
2016-12-06	06/12/2016 11:05	Arquivo TXT	21 KB
2016-12-06-1	06/12/2016 04:46	Arquivo TXT	3 KB
2016-12-06-2	06/12/2016 05:10	Arquivo TXT	5 KB
2016-12-06-3	06/12/2016 07:38	Arquivo TXT	36 KB
2016-12-21	21/12/2016 22:00	Arquivo TXT	60 KB
2016-12-22	22/12/2016 17:21	Arquivo TXT	99 KB

Figura 5.5.2.1: Pasta database com os arquivos gerados do monitoramento.

```
00:30:09 umidade=1023.0 chuva=678.0 temperatura=31.0 vazao=0.0 acao=ligar
00:30:19 umidade=245.0 chuva=678.0 temperatura=31.0 vazao=0.0 acao=ligar
00:30:29 umidade=258.0 chuva=678.0 temperatura=31.0 vazao=0.0 acao=desligar
00:30:39 umidade=283.0 chuva=678.0 temperatura=31.0 vazao=0.0 acao=desligar
```

Figura 5.5.2.2: Exemplo de arquivo texto contendo os dados dos sensores.

Os valores dos sensores manipulados são recebidos em decimal via MQTT. A conversão e tratamento dos dados é feito no Arduino e enviados pelo ESP8266 e no servidor é utilizado uma heurística tratando os dados de acordo com as seguintes tabelas.

Sensor	Umidade alta	Umidade moderado	Umidade baixa
Umidade	1 á 400	401 á 800	801 á 1024

Tabela 5.5.2.1: Valores de tradução da umidade.

Sensor	Chuva forte	Chuva fraca	Sem chuva
Chuva	1 á 400	401 á 600	601 á 1024

Tabela 5.5.2.2: Valores de tradução da chuva.

O sensor de temperatura já apresenta seu valor em grau Celsius não necessitando de tradução. O sensor de vazão apresenta valores zerados pois estava em fase de programação quando os dados foram coletados.

Alguns gráficos do monitoramento são apresentados a seguir para comprovar o funcionamento do sistema.

A partir do gráfico 1, no qual foi realizado um experimento de teste manipulando os sensores, é possível perceber que quando umidade se encontra próximo de 1024, ou seja, indicando uma umidade baixa e o sensor de chuva próximo de 700, indicando que não há presença de chuva, a válvula passa para o estado de ligado até que ocorra uma queda no valor de chuva. Com acréscimo de gotas de água ao sensor, indicando que foi detectado chuva, a válvula desliga e o seu valor diminui. Após não ser detectado mais chuva e a umidade se encontrar baixa, a válvula volta a ser ligada até que haja uma queda no valor de umidade.

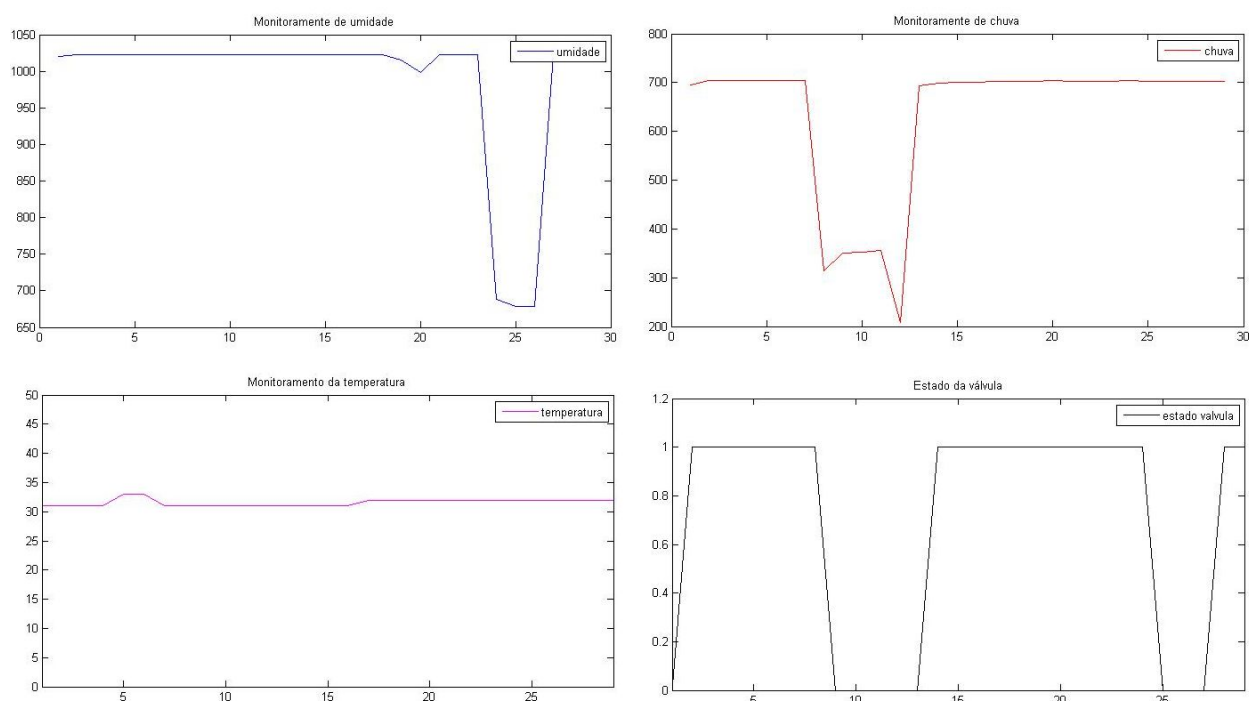


Figura 5.5.2.3 : Gráfico 1 de monitoramento.

O gráfico 2 foi realizado com uma manipulação maior dos sensores e apresentaram resultados dentro do esperado.

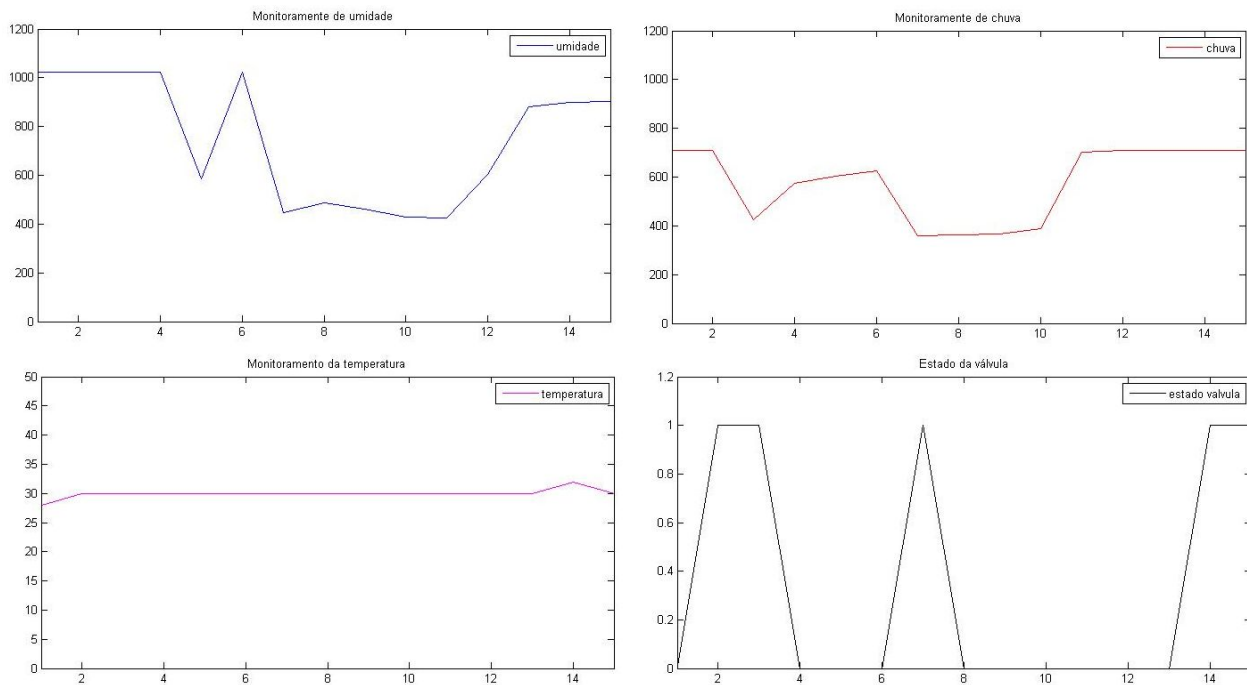


Figura 5.5.2.4: Gráfico 2 de monitoramento.

O gráfico 3 foi realizado em teste prático com uma planta de jardim. Neste gráfico, percebe-se que no início a válvula se encontra fechada e não há alteração nos valores pois o sistema se encontra fechado. Ao ligar o sistema, os valores de umidade começam a variar. Quando umidade está baixa, ou seja, com valores próximos de 1000, a válvula é ligada. Quando os valores de umidade caem para menos que 800 a válvula desliga e ocorre uma alternância entre os valores de umidade e estado da válvula. Por último o sensor de chuva é acionado após a detecção de chuva junto com umidade alta ocasionando o desligamento da válvula.



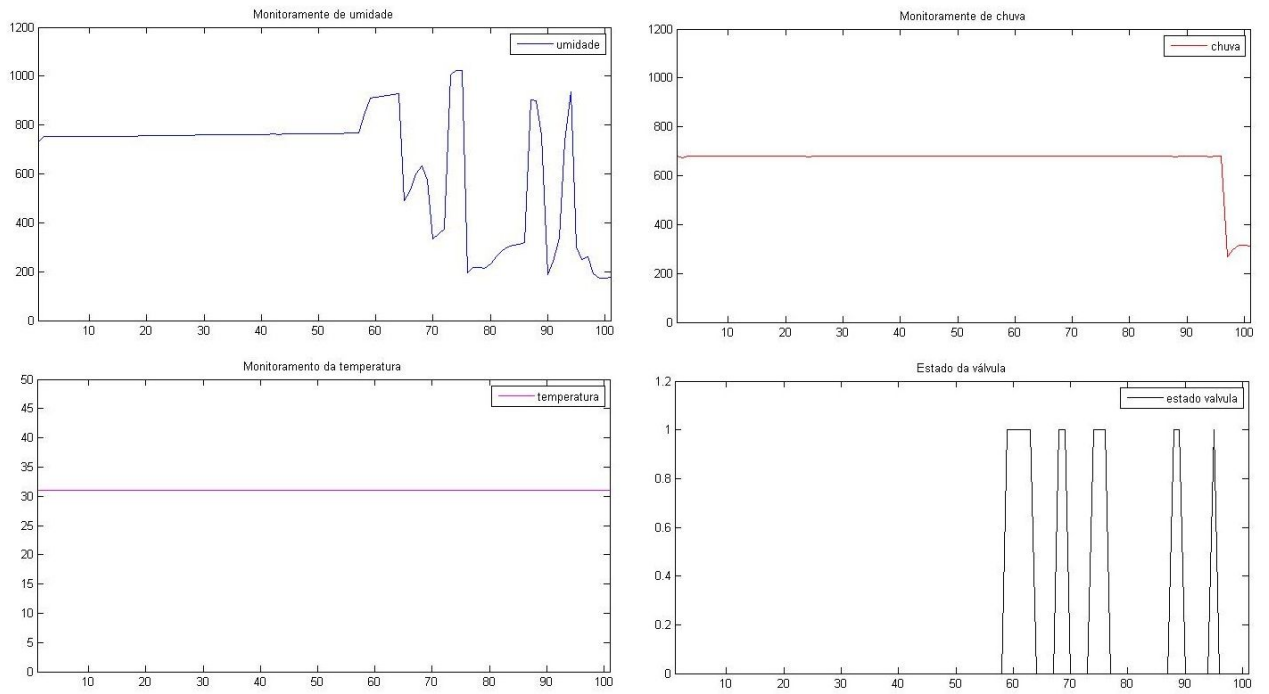


Figura 5.5.2.5: Gráfico 3 de monitoramento.

## 6. BIBLIOGRAFIA

1. ARDUINOBR. **Usando um relé para controlar dispositivos elétricos.** Conteúdo disponível em:  
<<http://www.arduino.br/arduino/acionadores/usando-um-rele-para-controlar-dispositivos-eletricos/>>. Acesso em: 24 de outubro, 2016.
2. BLOG FILIPEFLOP. **Monitoramento de planta usando Arduino.** Conteúdo disponível em:  
<<http://blog.filipeflop.com/sensores/monitore-sua-planta-usando-arduino.html>>. Acesso em: 24 de setembro, 2016.
3. BLOG FILIPEFLOP. **Sensor de chuva YL-83.** Conteúdo disponível em:  
<<http://blog.filipeflop.com/sensores/sensor-de-chuva-yl-83.html>>. Acesso em: 16 de outubro, 2016.
4. BLOG FILIPEFLOP. **Sensor de umidade de solo Higrômetro.** Conteúdo disponível em:  
<<http://www.filipeflop.com/pd-aa99a-sensor-de-umidade-do-solo-higrometro.html>>. Acesso em: 16 de outubro, 2016.
5. BLOG FILIPEFLOP. **Tutorial módulo wireless ESP8266 com Arduino.** Conteúdo disponível em: <<http://blog.filipeflop.com/wireless/esp8266-arduino-tutorial.html>>. Acesso em: 12 de outubro, 2016.
6. BLOG FILIPEFLOP. **Upgrade de firmware do módulo ESP8266.** Conteúdo disponível em:  
<<http://blog.filipeflop.com/wireless/upgrade-de-firmware-do-modulo-esp8266.html>>. Acesso em: 12 de outubro, 2016.
7. BLOG FILIPEFLOP. Datasheet YF-S201. Conteúdo disponível em:  
<[http://img.filipeflop.com/files/download/Datasheet\\_YF-S201.pdf](http://img.filipeflop.com/files/download/Datasheet_YF-S201.pdf)>. Acesso em: 28 de dezembro, 2016.
8. EMBARCADOS. **ESP8266 com Arduino.** Conteúdo disponível em:  
<<http://www.embarcados.com.br/esp8266-com-arduino/>>. Acesso em: 19 de outubro, 2016.
9. ESPRESSIF SYSTEMS. **Datasheet ESP8266EX v4.3.** Conteúdo disponível em:  
<<http://download.arduino.org/products/UNOWIFI/0A-ESP8266-Datasheet-EN-v4.3.pdf>>. Acesso em: 25 de outubro, 2016.

10. GITHUB. **Moquette, Java MQTT lightweight broker**. Conteúdo disponível em:  
<<https://github.com/andsel/moquette>>. Acessado em: 17 de novembro, 2016.
11. GITHUB. **Pubsubclient, A client library for the Arduino Ethernet Shield that provides support for MQTT**. Conteúdo disponível em:  
<<https://github.com/knolleary/pubsubclient>>. Acessado em: 17 de novembro, 2016.
12. GITHUB. **DHT-sensor-library, Arduino library for DHT11DHT22, etc Temp & Humidity Sensors**. Conteúdo disponível em:  
<<https://github.com/adafruit/DHT-sensor-library>>. Acessado em: 28 de dezembro, 2016.
13. NEWTON C. BRAGA. **Tudo Sobre Relés**. Conteúdo disponível em:  
<<http://www.newtoncbraga.com.br/index.php/como-funciona/597-como-funcionam-os-reles?showall=1&limitstart=>>. Acesso em: 24 de outubro, 2016.