

# *Sistema de Irrigação Modular - SIM*

Leonardo de Holanda Bonifácio\*, Ana Caroline Alves Amaro†

Programa de Graduação em Engenharia Eletrônica, Faculdade Gama

Universidade de Brasília

Gama, DF, Brasil

\* e-mail: leohb2@gmail.com

† e-mail: acarolineamaro@gmail.com

**Abstract**—Raspberry Pi is a board that facilitates the implementation of intelligent systems, objects, things connected to the network. The project aims to use a Raspberry Pi 3 to control a modular irrigation system, where a real-time survey is made to identify according to soil moisture, the need for irrigation to be performed or not. Some electronical components will be used to achieve the goal of modular automated irrigation.

**Resumo**—Raspberry Pi é uma placa que facilita a implementação de sistemas inteligentes, objetos ligados à rede. O projeto visa utilizar uma Raspberry Pi 3 para controlar um sistema de irrigação modular, onde é feito um levantamento em tempo real para identificar, de acordo com a umidade do solo, a necessidade da irrigação ser realizada ou não. Alguns componentes eletrônicos serão utilizadas para alcançar o objetivo da irrigação automatizada modular.

**Index Terms**—Raspberry Pi, Soil Moisture, Automation.

## I. INTRUÇÃO

A agricultura irrigada é o ramo da atividade humana que mais consome água, sendo a ela atribuída o uso de 70 % da água disponível nas diversas atividades [1]. Durante muitos anos, a população vem usando os recursos naturais de maneira desgovernada e sem se preocupar com as consequências que podem ocorrer ao meio ambiente devido a esse mau uso. Nos últimos anos, o Brasil enfrenta uma crise hídrica que requer comprometimento social no que diz respeito ao uso racional da água [2].

O manejo de irrigação representa os procedimentos utilizados para se irrigar as plantas com uma quantidade correta de água antes que o teor de água no solo diminua a ponto de causar dificuldades para as raízes absorverem as quantidades de água que a planta necessita para manter seu desenvolvimento sem restrições. Em outras palavras, para manejá-la adequadamente a irrigação, deve-se utilizar parâmetros que possam auxiliar na determinação de quando irrigar e das definições de parâmetros para o quanto de água deve ser aplicada na irrigação [3].

A agricultura tem crescido bastante e devido aos longos períodos de seca durante o ano, a tentativa de suprir a falta de chuva e garantir um produto de melhor qualidade tem feito com que o uso de métodos de irrigação sejam cada vez mais difundidos nos meios rurais, necessitando assim de um maior monitoramento no gasto da quantidade de água que pode ser desperdiçada se não houver um controle mais adequado.

A automação se faz necessária não somente pela possibi-

lidade de diminuição dos custos com a mão de obra, mas principalmente por necessidades operacionais. [4]O manejo da irrigação realizado sem o controle do conteúdo de água do solo pode resultar em insucesso da atividade agrícola, reduzindo a produtividade das culturas, aumentando os custos de produção pelo maior uso de energia, desperdício de nutrientes, doenças e demais gastos acarretados por irrigações excessivas ou deficitárias. Dessa forma, o monitoramento das condições de solo e clima durante o desenvolvimento da planta, aliado ao conhecimento sobre a cultura, permite aplicar com mais precisão a quantidade requerida, bem como o momento correto [5].

Com o intuito de melhorar a eficiência dos sistemas de irrigação e diminuir o gasto de água, tanto em aplicações residenciais quanto rurais, esse trabalho apresenta um sistema modular de irrigação automatizada sem fio com a plataforma *Raspberry Pi 3*. Com o auxílio de diversos sensores como por exemplo sensor de umidade do solo e o sensor de nível de água, o módulo adapta-se a componentes externos, possui baixo custo de produção comparado a sistemas existentes no mercado e é mais sustentável do que sistemas convencionais. Uma das vantagens da tecnologia é que ela pode ser programada e controlada remotamente e reduz a necessidade de mão-de-obra.

## II. OBJETIVOS

O objetivo geral do trabalho é fornecer uma solução tecnológica de automação agrícola de baixo custo direcionada para atender o pequeno e o médio produtor que possua alta eficiência e baixo consumo de água, utilizando uma plataforma *Raspberry Pi 3* conectada a módulos de monitoramento de umidade, temperatura, presença de água, e ativação de carga, com sistema de controle via conexão *WiFi*, pelo aplicativo *Blynk*.

Para atingir o objetivo geral, têm-se como objetivos específicos o controle dos módulos separadamente, integração com aplicativo *Blynk* e refinamento do projeto para funcionamento pleno.

### III. JUSTIFICATIVA

Segundo estudo realizado em 2015 [12], diversas regiões do país apontaram que o deficit hídrico está entre os principais causadores de quebra na produtividade. Contudo, o resultado das lavouras também sofre influência de outros fatores como: solo, ambiente, qualidade da semente, nutrição, além de pragas e doenças.

Na cultura da soja, uma pesquisa mostrou que a quebra de produtividade – avaliada em diferentes regiões do país – variaram consideravelmente e que grande parte delas é oriunda do deficit hídrico (73,8%), enquanto o manejo inadequado representou 26,8% da quebra geral da produtividade. Na cana-de-açúcar, retratou que a ausência de água na quantidade ideal representou 75,6% da quebra na média nacional. Já o manejo agrícola foi responsável por 24,4%. No entanto, foram observadas diferenças entre as regiões, sendo que no Nordeste o deficit hídrico respondeu por aproximadamente 86% das quebras, ao passo que no Sul do país as maiores quebras foram causadas por fatores relacionados ao manejo agrícola e à ocorrência de geadas [13]. Esse resultados evidenciam um ponto fundamental para o alcance da máxima produtividade nos grãos: a importância de manter um fluxo de água constante durante todo o ciclo dos cultivos. Esse fator deve ser fortemente considerado quando o objetivo é extrair todo potencial das plantas.

O deficit hídrico é responsável por grande parte das quebras de produtividade nas principais culturas do país, e pode ser solucionado por meio da adoção de tecnologias como o SIM. Neste sistema, o produtor conta com a melhor gestão do recurso natural e a garantia de que a planta está recebendo a quantidade ideal de água para seu desenvolvimento.

### IV. REQUISITOS

Para realização do projeto, será necessário o desenvolvimento de uma interação hardware-software que atenda às tarefas propostas. Alguns requisitos a considerar:

- Relé para ativação de carga da bomba d'água;
- Sensor de umidade do solo para controlar necessidade de irrigação;
- Sensor de umidade e temperatura do ar para uso em estufa;
- Sensor para ciência do nível de água no reservatório;
- Canos distribuidores de água;
- Reservatório de água e da planta;
- Autonomia de alimentação: visto que o projeto envia dados de sensores com frequência alta, o sistema necessita de fonte constante de tensão, tanto para alimentar a *Raspberry* quanto os módulos utilizados;
- Distância da *Raspberry* ao *modem*: para que a conexão com a internet seja estabelecida e o envio de dados seja concretizado, a placa deve estar no alcance do roteador. Caso contrário, será necessária uma outra forma de conexão, como um módulo GSM.

### V. DESCRIÇÃO GERAL

Como o foco do projeto é o processamento de dados, sua realização visa o aprendizado das funções de processamento. Dessa forma, a parte do hardware é relativamente simples. Será usada para a elaboração do projeto a *Raspberry Pi 3* e alguns módulos externos. Até o presente momento são usados 4 módulos: para ativação da bomba de água, um módulo relé; para saber o nível de água do reservatório, um transistor que é polarizado ao entrar em contato com o líquido; um sensor de umidade de solo que será usado como forma de decisão para irrigação e, por fim, um sensor de umidade e temperatura do ar para controle de estufa. A Figura (6) mostra a montagem do circuito completo, feita pelo *software Fritzing*. A Tabela (I) exibe os componentes eletrônicos utilizados para realização do projeto.

Tais módulos e sensores são controlados pelo aplicativo *Blynk* de qualquer aparelho celular com conexão com a Internet.

Na Figura (1), o diagrama de blocos simplificado auxilia na melhor compreensão do projeto.

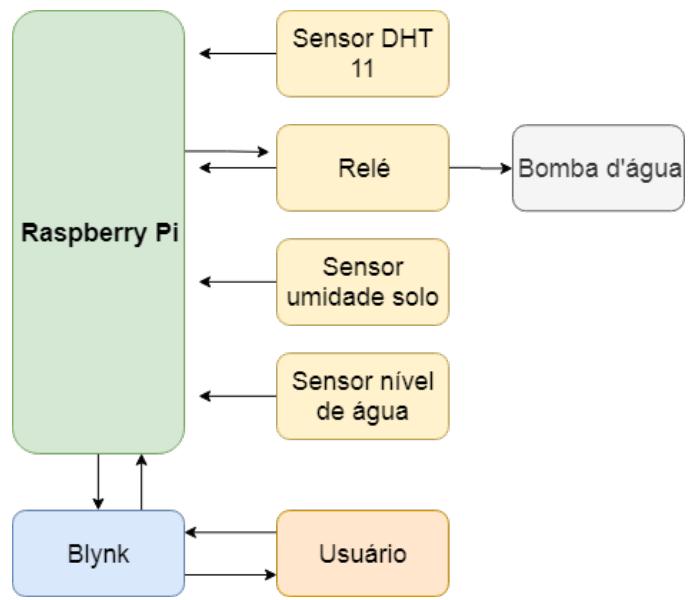


Figura 1. Diagrama de blocos.

### VI. DESCRIÇÃO INDIVIDUAL

Nessa seção, é explicado o funcionamento de cada bloco do projeto e como ele foi integrado com as outras partes.

#### A. Interface e app *Blynk*

*Blynk* é uma plataforma voltada para automação, com o conceito de Internet das Coisas. Através desse aplicativo, o usuário pode controlar módulos, sensores através de botões, *widgets* de sua escolha.

Para estabelecer comunicação entre a *Raspberry* e o aplicativo, é necessário estar conectado a Internet e de uma chave

Componente	Quantidade
Raspberry Pi 3B	1
Sensor DHT11	1
Sensor Umidade do Solo	1
Resistor 1k	1
Resistor 10k	1
Transistor BC548	1
Módulo Relé 2 canais	1
Jumper	15
LED 3mm	1
Fonte 5V,1A	1
Cabo mini-USB	1

Tabela I  
COMPONENTES UTILIZADOS.

individual de autenticação. Devido a certas incompatibilidades entre os dois, o projeto foi programado na linguagem JavaScript.

Na interface criada, primeiramente o usuário escolhe o modo de controle desejado: automático ou manual. Nas próximas subseções, os outros módulos e o processo de integração com o app são explicados.

#### B. Relé

O relé é um ativador de cargas que pode ser microcontrolado. No projeto, ele ativa a bomba d'água que regará a planta. O critério de decisão para o atendimento é, primeiramente, o modo de funcionamento decidido pelo usuário e, depois, a resposta do sensor de umidade de solo.

No modo automático, o relé é ligado quando o sensor informar que o solo está seco e desligado quando está molhado. No modo manual, o controle é feito inteiramente pelo usuário.

No app *Blynk*, o estado atual do relé, desligado ou ligado, é exibido em um botão, que serve tanto para ativação quanto para controle.

#### C. Umidade do Solo

O sensor é usado para detectar as variações de umidade no solo, tendo seu funcionamento da seguinte forma: quando o solo está seco, a saída do sensor fica em estado alto, e quando úmido em estado baixo.

No aplicativo, um *widget* do tipo LED mostra o estado atual do solo. Quando seco, o LED fica desligado e quando molhado, ligado. Esse estado é usado como critério para ativação da bomba d'água pelo relé.

Deve-se ter cuidado com a calibração de sensibilidade do sensor, visto que, devido a falta de portas ADC na *Raspberry*, desfruta-se de apenas uma saída digital.

#### D. Sensor de Nível de Água

A princípio, seria usado um sensor ultrassônico para determinar o nível de água no reservatório. Entretanto, a dupla encontrou muita dificuldade em programá-lo em *JavaScript*. Para solucionar esse problema, foi criado um sensor que apenas verifica se há água ou não em um certo nível do reservatório. A Figura (2) mostra a montagem desse circuito. É utilizado um transistor, em que fios são ligados à base e ao Vcc do mesmo. Quando o fio está fora da água, a tensão lida

na GPIO é 1 (leitura digital), quando o fio está submerso, o circuito fecha com o Vcc e o transistor é polarizado e a leitura é igual a 0.

Dessa forma, com os fios fixados no fundo do reservatório, um *widget* LED mostra o estado do reservatório, cheio ou vazio. Quando estiver vazio, o usuário receberá uma notificação através do *Blynk*, solicitando reencher o mesmo (Fig.3).

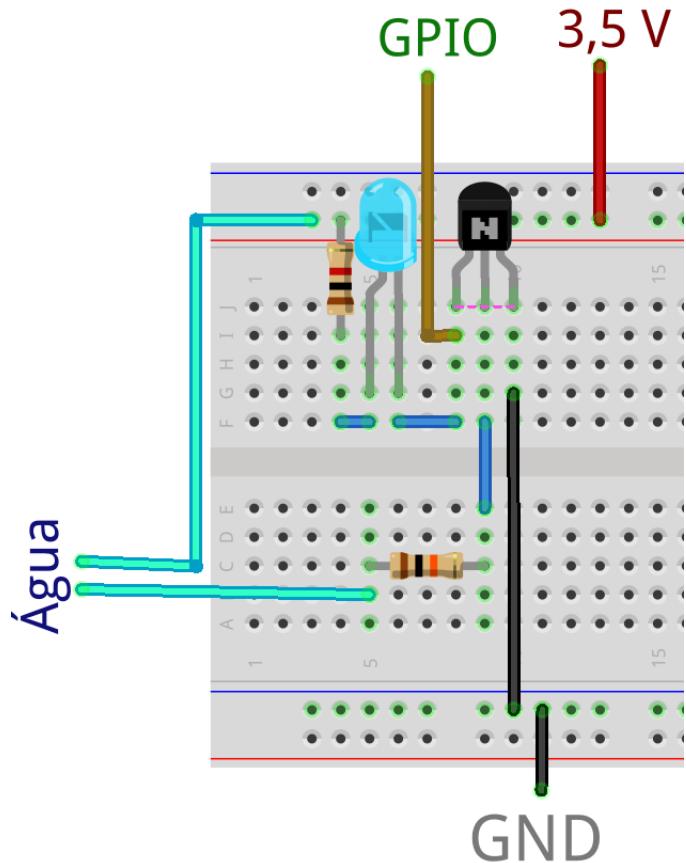


Figura 2. Sensor de Nível de Água.

#### E. Sensor de Temperatura e Umidade do Ar

Como ferramenta opcional, foi implementada, usando o sensor DHT11, uma função de leitura de temperatura e umidade do ar. Usando bibliotecas, a leitura é feita a cada 10 segundos e enviada a um pino virtual, que exibe o valor lido.

Esse sensor será usado como método de controle para estufa com a planta.

## VII. EXPLICAÇÃO DO CÓDIGO

Nessa seção, será explicado o código que está no apêndice. Ele foi programado em *JavaScript* devido à melhor compatibilidade com o app *Blynk*, fundamental no projeto.

Nas linhas 1-4, são instanciadas as bibliotecas usadas no projeto: a do *Blynk*, para funções de pinos virtuais, leitura, etc; Gpio, que dá controle de ativar e desativar as portas, e a biblioteca de leitura do sensor DHT11.

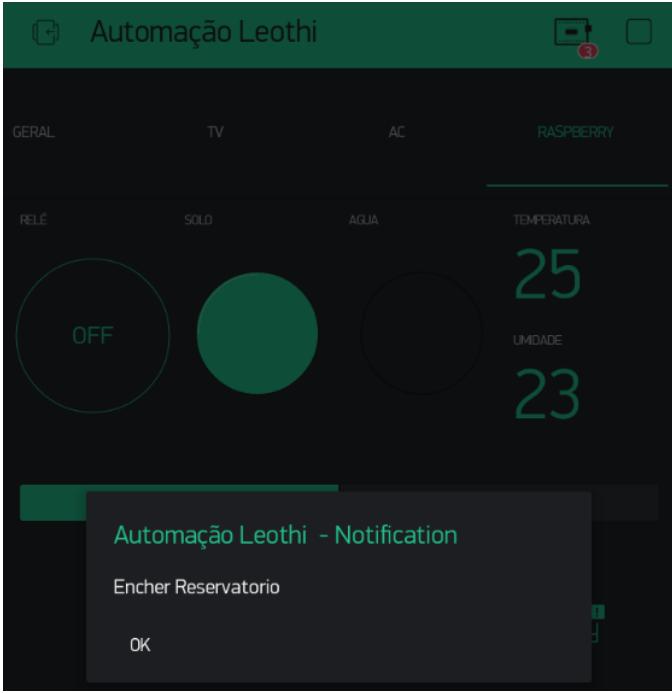


Figura 3. Notificação no aplicativo.

A seguir, os pinos usados são declarados para a biblioteca Gpio.

Nas linhas 11-14, as variáveis globais são instanciadas. São elas: tipo de sensor de temperatura (DHT11), o pino onde ele está ligado e o modo de operação, que é utilizado em duas funções.

As linhas 21-41 são para o funcionamento do *Blynk*. Primeiramente, a conexão com o servidor do aplicativo é feita com o código de autenticação da linha 17 e, depois, as variáveis de botões virtuais, mostrados na interface (Fig. 5) são declaradas. Em seguida, se a conexão foi executada, as variáveis de pino virtual 2 e 0, que estão relacionados com o relé e com o seletor de modo, são sincronizados com o estado do botão no aplicativo, para não haver problema de sincronicidade na inicialização do projeto.

A linha 44 inicializa o sensor DHT 11 com a função de sua biblioteca.

Nas linhas 46-49, as funções usadas no projeto tem seus *timers* setados. Dependendo do segundo argumento da função, ela é executada a cada X milissegundos. O critério de escolha do tempo foi só um número razoável para ter um controle satisfatório sem necessidade de muito envio de dados.

A função *v0.on* verifica se há mudança no pino virtual e escreve, como 1 ou 2, esse valor em param2, que é atribuído à variável modo. Nesse caso, v0 é o botão de seleção de modo. A variável global é usada na função de regar, de acordo com o modo selecionado.

A função de leitura do sensor é feita pela biblioteca do mesmo. A cada 1 s (linha 47), a leitura é feita e enviada aos pinos virtuais 3 e 4 através da função *virtualWrite* para visualização no aplicativo.

A função regar é a mais importante do projeto. Primeiramente, a variável leitura lê o sensor de umidade do solo. Se a variável global definida na função anterior for igual a 1 (modo automático), o código verifica duas condições: a primeira é se o solo está seco (leitura == 1). Caso isso seja verdadeiro, o LED na interface é desligado e o relé responsável pelo ativamente da bomba d'água é ligado. O *virtualWrite* sincroniza o botão do relé no aplicativo para ligado. A segunda condição é quando o solo está molhado (leitura == 0). Caso seja verdadeira essa condição, o LED é ligado, o relé desligado e o botão virtual desligado. Ou seja, a cada 1000 ms (definidos na linha 48), é verificada condição do solo, o relé com é alterado de acordo com essa condição e o *feedback* é dado ao usuário pelo aplicativo e pelo terminal da *Raspberry*, tudo automático.

A segunda condição da função é se está no modo manual (*modo==2*). Caso seja verdadeiro, primeiramente o relé é sincronizado com o botão virtual. Nas linhas 98-105, o LED do solo é alterado de acordo com a sua leitura, de forma automática. Porém, nas linhas 106-115, a função *v2.on* faz o controle manual do relé de acordo com a entrada do usuário no aplicativo.

A função *LeituraAgua()* é responsável pelo nível de água do reservatório. A cada 1000 ms (linha 49), é feita uma leitura digital do pino do sensor. Se o reservatório estiver cheio (*leitura == 0*), o LED no *Blynk* é ligado. Se estiver vazio, o LED é desligado.

Caso o reservatório estiver vazio, o usuário recebe uma notificação no celular, solicitando encher-lo. Isso é feito através dos widgets *Eventor* e *Notification* do aplicativo, que funcionam de forma parecida com uma condicional.

## VIII. RESULTADOS

Até o presente momento, os resultados obtidos foram:

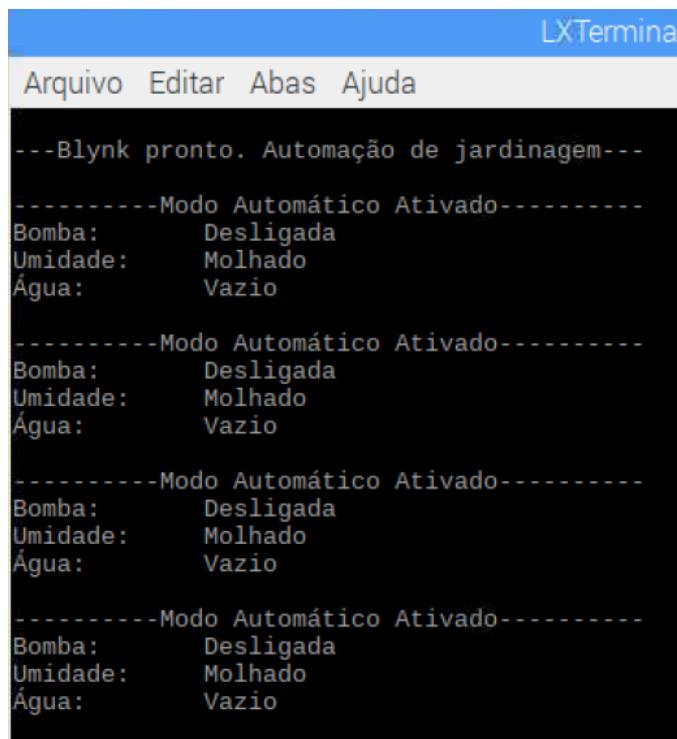
- Definição dos módulos a serem usados;
- Aquisição das características dos mesmos;
- Elaboração dos códigos para funcionamento individual dos módulos;
- Montagem do circuito e comunicação entre os módulos e a *Raspberry Pi 3*;
- Integração dos módulos com o app *Blynk* utilizando *JavaScript*;
- Montagem completa do projeto em sua estrutura final.

A interface final no aplicativo *Blynk* pode ser vista na Fig. (5). Ela é feita de um botão normal para controle e ativação do relé; dois LEDs que mostram o estado do solo e do nível de água; um botão de seleção de modo de operação; dois visores que mostram as leituras do DHT11 e dois widgets para notificação do celular.

A Figura (7) mostra a parte física do projeto sem o resto da estrutura. As Figuras (8) e (9) exibem o mesmo em sua estrutura final. Trata-se de uma caixa de plástico simples com uma garrafa de água como reservatório. Uma outra caixa está fora dessa estrutura, contendo a *Raspberry*, relé e a base de alguns sensores. É possível identificar todos os componentes

utilizados no diagrama de blocos (Fig. 1).

Caso seja de seu desejo, o usuário pode acessar o terminal da *Raspberry Pi* e obter alguns dados em tempo real (Fig. 4).



The screenshot shows a terminal window titled 'LXTerminal' with the following text output:

```
--Blynk pronto. Automação de jardinagem--  
-----Modo Automático Ativado-----  
Bomba: Desligada  
Umidade: Molhado  
Água: Vazio  
-----Modo Automático Ativado-----  
Bomba: Desligada  
Umidade: Molhado  
Água: Vazio  
-----Modo Automático Ativado-----  
Bomba: Desligada  
Umidade: Molhado  
Água: Vazio  
-----Modo Automático Ativado-----  
Bomba: Desligada  
Umidade: Molhado  
Água: Vazio
```

Figura 4. Terminal da placa.

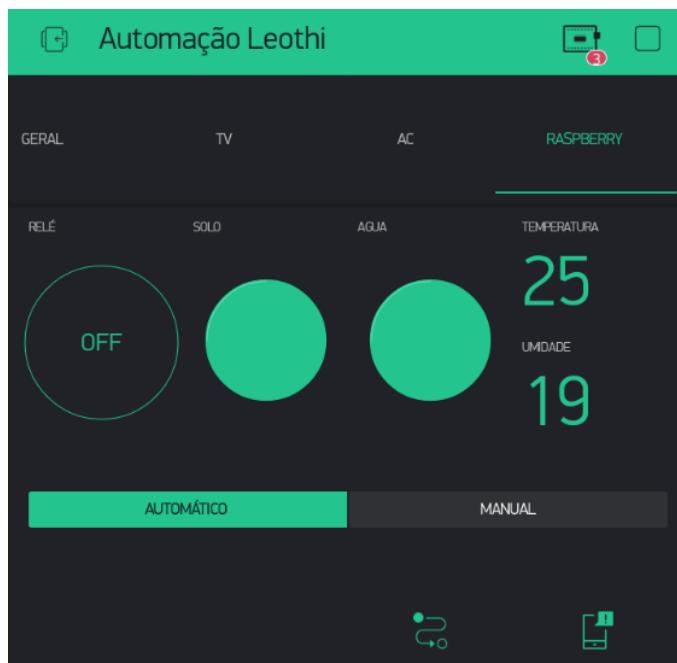


Figura 5. Interface do projeto.

Os resultados obtidos até agora são satisfatórios para a aplicação e requisitos do projeto. O usuário consegue ter

controle de sua plantação de onde estiver, apenas com conexão à Internet. A parte física funciona perfeitamente e a interface no aplicativo também.

## IX. CONCLUSÃO

O manejo de irrigação representa os procedimentos de controle para irrigar as plantas com a quantidade correta antes que o teor de água no solo diminua a ponto de causar dificuldades para absorção das raízes e desenvolvimento na planta.

O deficit hídrico está entre os principais causadores de quebra na produtividade. O estudo e otimização do uso da água pode reduzir de grande forma os custos de um produtor e melhor o crescimento de uma plantação. Por esse motivo, o uso de métodos eficientes de irrigação têm sido cada vez mais difundidos.

Levando esses fatos em consideração, o projeto visou criar um Sistema de Irrigação Modular, em que o controle e visualização das condições de uma planta são feitos remotamente, diminuindo custos e aumentando a eficiência. Para isso, diversos módulos eletrônicos foram utilizados e integrados com o aplicativo *Blynk*, uma plataforma de automação com conceito de Internet das Coisas, juntamente com a placa *Raspberry Pi 3B*.

A maior dificuldade para realização do projeto foi a incompatibilidade do *Blynk* com a *Raspberry*. A dupla começou a programar na linguagem C os módulos, mas na hora de integrar com o aplicativo, foi visto que, devido à arquitetura da placa, não funcionava bem nessa linguagem. Esse problema foi resolvido com o uso do *JavaScript*. No início, foi complicado compreender as funções e sintaxe da linguagem, mas ao final, o objetivo foi alcançado.

Como resultado, o usuário pode, através de uma interface simples, visualizar as condições de sua aplicação e decidir pelo controle automático ou manual da mesma. A bomba d'água pode ser ativada, os estados do sensor de umidade do solo e o nível do reservatório de água são exibidos em LEDs, e a leitura de temperatura e umidade do ar, que pode ser usada em uma estufa, também são exibidos na tela do projeto e no terminal do sistema *Raspbian*.

## X. BIBLIOGRAFIA

- 1 Medici, L. O. Rocha, H. S.; Carvalho, D. F.; Pimentel, C.; Azevedo, R. A. Automatic controller to water plants. Scientia Agricola. Piracicaba. V. 67. N. 6. 2010.
- 2 Desenvolvimento de um sistema modular automatizado para controle da irrigação residencial/rural - Disponível em: <http://laica.ifrn.edu.br/projetos/desenvolvimento-de-um-sistema-modular-automatizado-para-controle-da-irrigacao-residencialrural>. Acesso em 15 de outubro de 2018.
- 3 Calbo, A. G.; Silva, W. L. C. Sistema Irrigas para manejo de irrigação: fundamentos, aplicações e desenvolvimen-

- tos. Adonai Gimenez Calbo, 2005.
- 4 Suzuki, M. A.; Hernandez, F. B. T. Automação de Sistemas de Irrigação. Curso De Capacitação Em Agricultura Irrigada, v. 1, 1999.
- 5 Batista, S. C. O. Estudo técnico e econômico de um dispositivo visando à automação de sistemas de irrigação a partir do monitoramento do conteúdo de água no solo. 44p. Dissertação. Universidade Federal Rural do Rio de Janeiro. Seropédica. 2012.
- 6 NETAFIM. Sistema de irrigação inteligente. Disponível: <https://www.netafim.com.br>. Acesso em 15 de outubro de 2018.
- 7 InfoEscola. Relé. Disponível: <https://www.infoescola.com/electronica/rele/>. Acesso em 15 de outubro de 2018.
- 8 Irrigação: como escolher o melhor sistema para irrigar lavouras de grãos. Disponível: <https://www.agrolink.com.br/noticias/irrigacao-como-escolher-o-melhor-sistema-para-irrigar-lavouras-de-graos-405055.html>. Acesso em 15 de outubro de 2018.
- 9 Carvalho, Matheus Souza. Sensor para monitoramento de umidade do solo utilizando energia solar. Disponível: <https://www.quixada.ufc.br/wp-content/uploads/2017/03/2016-1-es-matheus-souza-de-carvalho.pdf>. Acesso em 15 de outubro de 2018.
- 10 Jannuzzi, Cristiano. Irrigação inteligente economiza água e não desperdiça. Disponível: <https://www.grupocultivar.com.br/artigos/irrigacao-inteligente-economiza-agua-e-nao-desperdica>. Acesso em 28 de outubro de 2018.
- 11 BATTISTI, R.; SENTELHAS, P. C. Drought tolerance of brazilian soybean cultivars simulated by a simple agrometeorological yield model. Experimental Agriculture, Cambridge, v. 51, p. 285-298, 2015.
- 12 Revista Cafeicultura. Produtividades recordes e a importância do manejo eficiente da irrigação. Disponível: <https://www.irrigacao.net/irrigacao/produtividades-recordes-e-a-importancia-do-manejo-eficiente-da-irrigacao/>. Acesso em 28 de outubro de 2018.
- 13 Souza Rezende, Ronaldo . Irrigação total. Disponível: <http://www.agencia.cnptia.embrapa.br/gestor/cana-de-acucar/arvore/CONTAG0156711200516718.html>. Acesso em 28 de outubro de 2018.
- 14 Ricardo Marin, Fábio . Fenologia . Disponível: <http://www.agencia.cnptia.embrapa.br/gestor/cana-de-acucar/arvore/CONTAG016822122006154840.html>. Acesso em 28 de outubro de 2018.

## XI. APÊNDICES

### A. Imagens

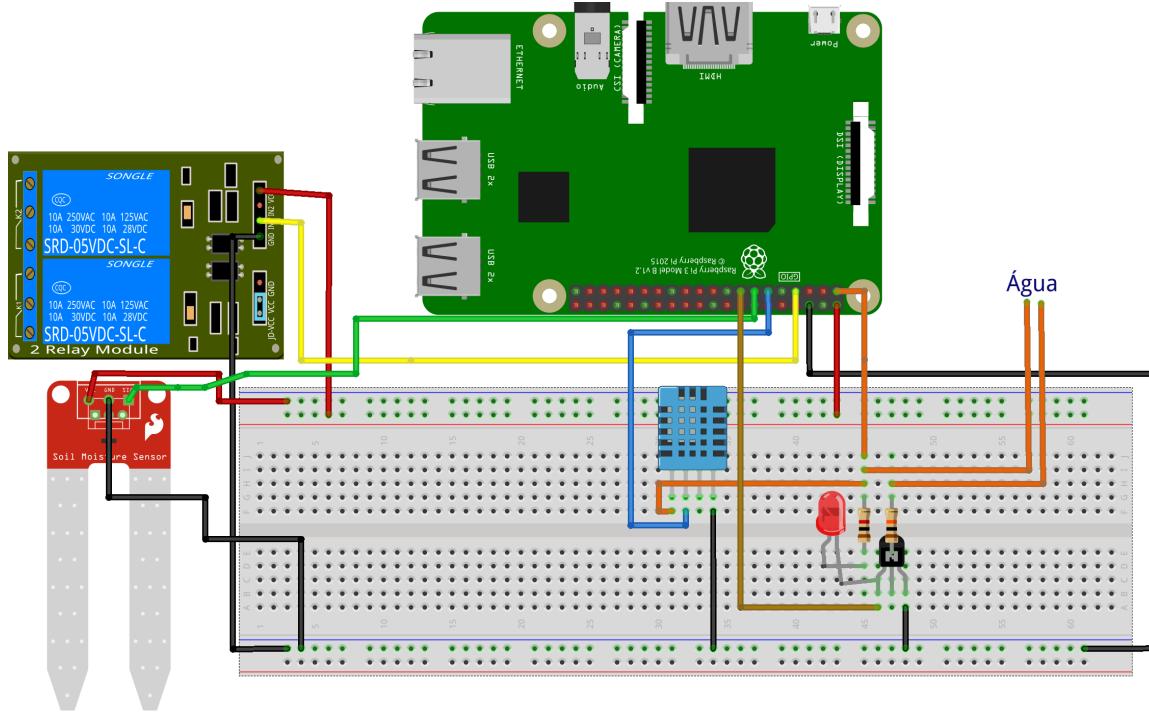


Figura 6. Montagem completa.

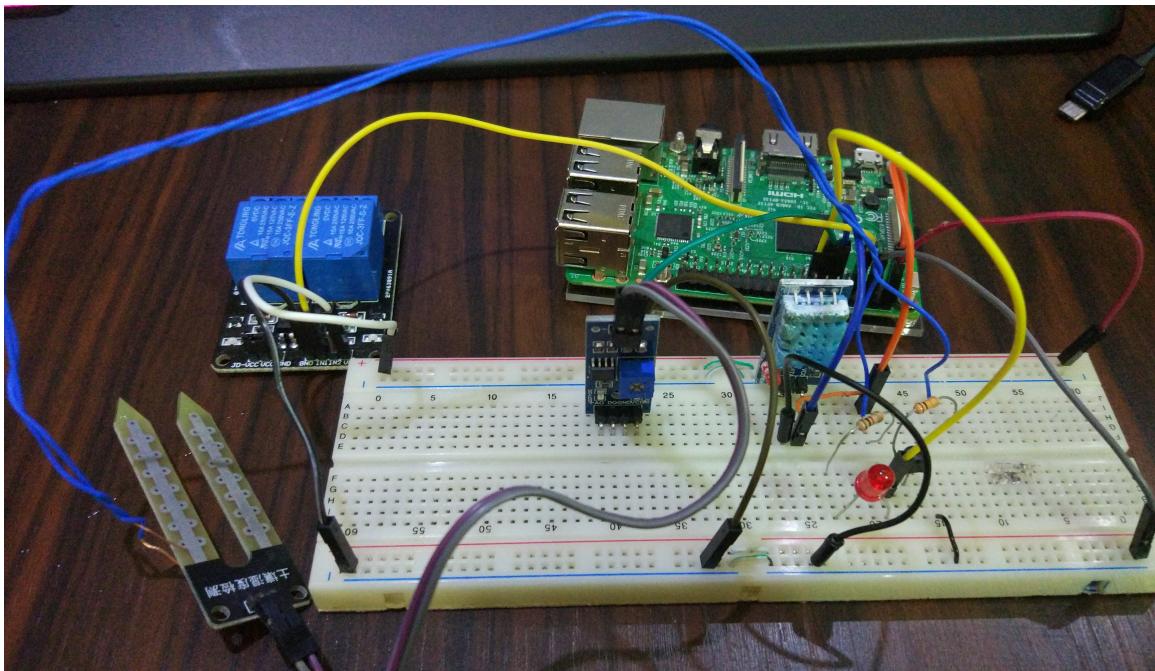


Figura 7. Projeto completo, sem estrutura física.



Figura 8. Projeto completo.

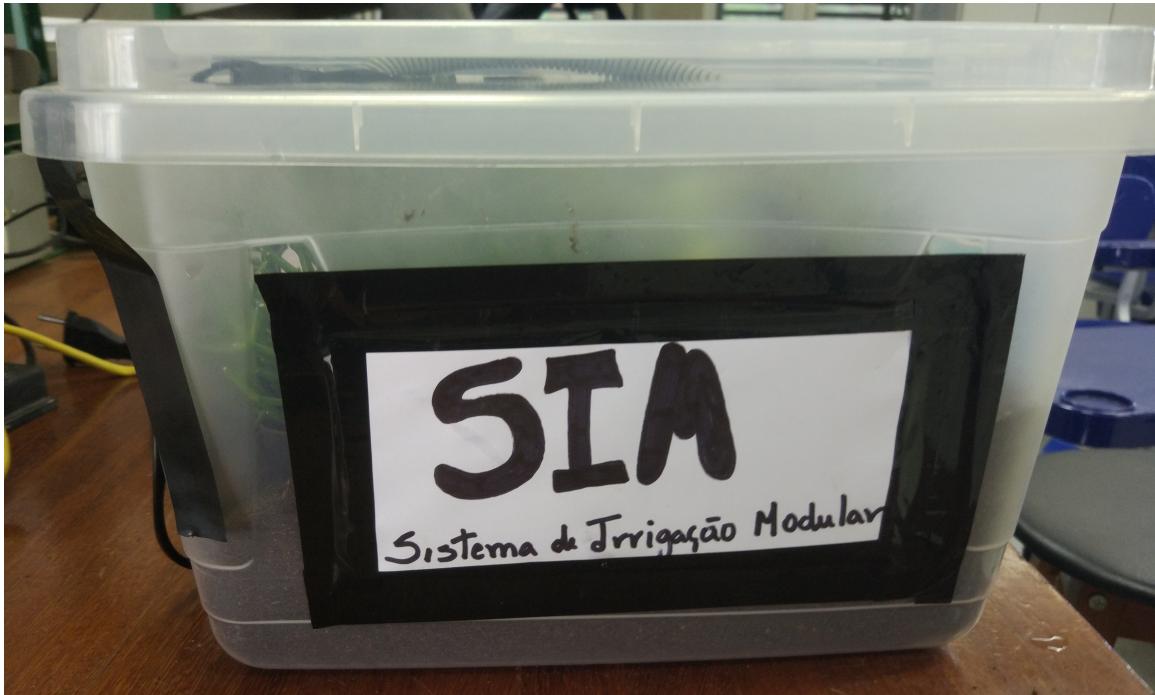


Figura 9. Projeto completo fechado.

## B. Códigos

```
1 // Bibliotecas
2 var Blynk = require('blynk-library'); // blynk
3 var Gpio = require('onoff').Gpio; // gpio
4 var sensorLib = require('/usr/lib/node_modules/node-dht-sensor'); //DHT11
5
6 // Pinos
7 var rele = new Gpio (4, 'out'); // saída para relé
8 var soil = new Gpio (27, 'in'); // entrada umidade do solo
9 var agua = new Gpio (22, 'in'); // entrada sensor água
10
11 // Variaveis Globais
12 var sensorType = 11; // 11 for DHT11, 22 for DHT22 and AM2302
13 var sensorPin = 17; // GPIO para DHT
14 var modo;
15
16 // Auth code e setup Blynk
17 var AUTH = '2e761xxxxxxea989ec73940d680850';
18
19 /////////////////////////////////
20
21 // Conectando Blynk
22 var blynk = new Blynk.Blynk(AUTH, options = {
23   connector : new Blynk.TcpClient() // primeira opção UnB
24 });
25 //var blynk = new Blynk.Blynk(AUTH); // segunda opção
26
27 // Pinos app Blynk (necessário estar depois da conexão com blynk)
28 var v1 = new blynk.VirtualPin(1); // LED água
29 var v2 = new blynk.VirtualPin(2); // botao rele
30 var v6 = new blynk.VirtualPin(6); // LED soil
31 var v0 = new blynk.VirtualPin(0); // Seletor de Modo
32
33 // Se conectado, enviar mensagem e sincronizar relé e modo com botão virtual
34 blynk.on('connect', function() {
35   console.log('');
36   console.log('');
37   console.log('');
38   console.log("");
39   blynk.syncVirtual(2);
40   blynk.syncVirtual(0);
41 });
42
43 // Inicialização
44 sensorLib.initialize(sensorType, sensorPin);
45
46 // Timers para funções
47 setInterval(LeituraDHT, 10000); // Leitura DHT a cada 10s
48 setInterval(regar, 1000); // Verificar regar a cada 1 s
49 setInterval(LeituraAgua,1000); // Leitura água a cada 1 s
50
51 /////////////////////////////////
52
53 // Definição de Modo
54 v0.on('write',function(param2) {
```

```

55     modo = param2;
56 }
57
58 // Função Leitura DHT
59 function LeituraDHT() {
60     var readout = sensorLib.read();
61     blynk.virtualWrite(3, readout.temperature.toFixed(1)); // V3 para temp
62     blynk.virtualWrite(4, readout.humidity.toFixed(1)); // V4 para humi
63
64     console.log('Temperatura:', readout.temperature.toFixed(1) + 'C');
65     console.log('Humidade: ', readout.humidity.toFixed(1) + '%');
66 }
67
68 // Função para Regar
69 function regar() {
70     var leitura = soil.readSync();
71     // Modo Automático
72     if(modo == 1){
73         console.log(' ');
74         console.log('-----Modo Automático Ativado-----');
75         // Solo Seco
76         if(leitura > 0){
77             blynk.virtualWrite(6, 0); // DESLIGA LED
78             rele.writeSync(0);
79             blynk.virtualWrite(2,1);
80             console.log('Bomba: Ligada');
81             console.log('Umidade: Seco');
82         }
83         // Solo Molhado
84         if (leitura == 0){
85             blynk.virtualWrite(6, 1023); // LIGA LED
86             rele.writeSync(1);
87             blynk.virtualWrite(2,0);
88             console.log('Bomba: Desligada');
89             console.log('Umidade: Molhado');
90         }
91     }
92     // Modo Manual
93     if(modo ==2) {
94         console.log(' ');
95         console.log('-----Modo Manual Ativado-----');
96         blynk.syncVirtual(0);
97         // Leitura do sensor e ativação do LED
98         if(leitura > 0){
99             blynk.virtualWrite(6, 0); // DESLIGA LED
100            console.log('Umidade: Seco');
101        }
102        if (leitura == 0){
103            blynk.virtualWrite(6, 1023); // LIGA LED
104            console.log('Umidade: Molhado');
105        }
106        v2.on('write', function(param1) {
107            // Bomba ligada
108            if(param1 == 1){
109                rele.writeSync(0);
110            }

```

```
111     // Bomba desligada
112     else{
113         rele.writeSync(1);
114     }
115 }
116 }
117 // Modo não-definido
118 if(modo ==0) {
119     console.log('Selecione o modo de operação');
120 }
121 }
122
123 // Função Nível Água
124 function LeituraAgua() {
125     var leituraagua = agua.readSync();
126     if (leituraagua ==0) {
127         console.log('Água:          Cheio');
128         blynk.virtualWrite(1, 1023); // LIGA LED
129     }
130     if (leituraagua == 1) {
131         console.log('Água:          Vazio');
132         blynk.virtualWrite(1, 0); // DESLIGA LED
133     }
134 }
```