

# Mini Estufa automatizada

Lourrany Gomes De Mesquita  
Universidade de Brasília (UnB)  
Faculdade do Gama (FGA)  
Brasília, Brasil  
lgomesdemesquita@gmail.com

Maria Tereza Dourado Melo  
Universidade de Brasília (UnB)  
Faculdade do Gama (FGA)  
Brasília, Brasil  
mariaterezadourado.melo@gmail.com

**Resumo**—Este documento apresenta a proposta de um sistema para monitorar cultivos de forma automatizada. O sistema deve ser capaz de acionar automaticamente o modo de irrigação utilizando como ferramenta principal a medição da umidade do solo. O valor medido será convertido em níveis de tensão que serão processadas pela plataforma Raspberry Pi. Logo após ocorrerá o acionamento ou desligamento do módulo que controla o processo de irrigação.

**Palavras-chave** — *Umidade, Irrigação, Controle.*

## I. INTRODUÇÃO

Para garantir um jardim com plantas vivas e saudáveis é necessário o cuidado constante com aplicação de adubos, nutrientes e água nas proporções corretas. Esse processo exige atenção já que a aplicação de doses excessivas pode ocasionar problemas às raízes das plantas, causando o apodrecimento com a lavagem de nutrientes essenciais. Também, pode haver a aplicação de doses insuficientes que pode prejudicar o desenvolvimento da planta [1].

A utilização de recursos de automação vem se mostrando uma ferramenta bem conceituada em setores produtivos, uma vez que aperfeiçoa a utilização de recursos e minimiza custos de produção. Em sistemas de irrigação tem mostrado-se como uma alternativa para poder programar o horário, a frequência e a duração das regas [2][3]. Este processo ajuda a estabelecer a quantidade de água fornecida às plantas, o que é essencial para o seu desenvolvimento saudável.

## II. JUSTIFICATIVA

A irrigação é uma atividade utilizada não só na agricultura, mas também em viveiros, jardins, pomares e demais formas existentes para de cultivos de plantas. Essa técnica tem por objetivo o fornecimento de água em quantidade suficiente e no momento certo para cada cultura, garantindo assim a sobrevivência e também enriquecendo o aspecto físico.

Além desse cuidado com a irrigação, também se faz necessário, para o sucesso da colheita, o monitoramento do crescimento da planta, bem como a análise dos aspectos físicos do solo. Diante disso, este trabalho descreve o desenvolvimento de um sistema automatizado capaz de controlar a irrigação de plantações de pequeno porte. A proposta visa evitar riscos de perdas de culturas de plantas por

falta ou excesso de água, mas também, pode ser uma fator primordial à economia, evitando o desperdício das fontes hídricas e reduzindo o tempo de cuidado e tutela empregado nas plantações.

## III. OBJETIVO

Elaborar um sistema eletrônico que automatiza o monitoramento e irrigação de campos de cultivo. O Sistema será constituído pela plataforma Raspberry Pi, desenvolvida no Reino Unido pela Fundação Raspberry Pi. Esta atuará sobre os sensores na aferição de parâmetros referentes às condições do ambiente onde estão inseridos e na construção de uma base de dados, bem como no processamento de imagem para a análise do crescimento da planta. Dessa forma, é possível definir o momento apropriado e ajustar a quantidade de água que deve ser utilizada na irrigação.

## IV. REQUISITOS

A ideia proposta sugere o monitoramento constante de um determinado tipo de cultura de planta. Para a montagem do sistema de irrigação proposto os principais componentes a serem utilizados serão: Raspberry Pi, os sensores e o sistema de alimentação.

O sensor de umidade detecta dados referentes às condições do solo e enviam as informações captadas ao Raspberry Pi. Nesta, ficam armazenado o banco de dados, com as informações das necessidades pré-programadas para o tipo de cultura onde os sensores estão inseridos.

A válvula Solenóide é usada para permitir o fluxo de água. Quando energizada, ela abre e permite a vazão de água, caso contrário, ela fecha e corta o fluxo. Para ativar ou não a válvula, será utilizado um módulo relé, que será acionado de acordo com a situação do solo: se estiver seco, aciona o relé, caso contrário, este será desligado.

O sistema de alimentação será utilizado para energizar o módulo relé e consequentemente, a válvula solenóide. Dessa forma, permite a oferta de água.

Será utilizado um sensor de nível a fim de verificar a quantidade de água presente no reservatório. Se a quantidade de água for baixa, o sistema não realiza a irrigação do cultivo e envia um sinal de alerta indicando a necessidade de reabastecer o reservatório. Caso contrário, o sistema é irrigado.

Será utilizado uma webcam a fim de capturar imagens das plantas, de forma a manter um controle de seu desenvolvimento, que poderá ser ajustado em um intervalo de tempo configurado pelo usuário.

O processamento principal do sistema será realizado com a Raspberry Pi, responsável por receber e processar instruções da interface com o usuário. Além disso, será responsável por enviar os dados lidos dos sensores aos usuários e armazená-los em um Banco de Dados para uma pesquisa de histórico de dados.

Tabela 1. Lista de Componentes.

LISTA DE COMPONENTES	
COMPONENTE	QUANTIDADE
Raspberry Pi	1
Sensor de umidade	1
Sensor de nível	1
Válvula solenóide	1
Módulo relé	1
Webcam	1
Reservatório de água	1

#### A. Sensor de umidade do solo

O sensor é formado por duas hastes (sondas) que formam uma resistência sensível às variações de umidade do solo. Quanto maior a quantidade de água presente no solo maior será o nível de condutividade elétrica do sensor. Isto significa que o solo está úmido e consequentemente a resistência nos terminais do sensor é baixa e a tensão é alta. No entanto, quanto menor a quantidade de água no solo menor será o nível de condutividade elétrica do sensor. Dessa forma a resistência do sensor é alta e a tensão é baixa.

Este sensor tem duas opções de ser ligado podendo ser digital ou analógica. A diferença é que o modo digital trabalha com os valores 0 ou 1 e por isso apresenta menor precisão. Já o modo analógico trabalha com um intervalo de valores, de 0 a 1023, apresentando maior precisão. No entanto dependerá de um circuito Conversor Analógico Digital (ADC).

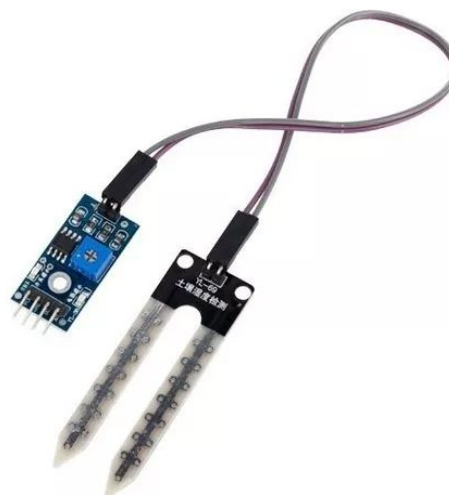


Figura. 1. Sensor de umidade do solo – Higrômetro.

#### B. Sensor de nível tipo boia.

O sensor de nível do tipo boia é formado por um corpo oco flutuador (bóia) que realiza o movimento de subir ou descer de acordo com o aumento ou redução do nível do líquido.



Figura. 2. Sensor de nível tipo boia.

#### C. Módulo relé.

O relé é um dispositivo que possui uma bobina, e um contato preso a uma mola de rearme, que conecta com os terminais nas posições Normalmente Aberto (NA) e Normalmente Fechado (NF). Em estado de repouso, o contato está na posição NF (normalmente fechado).

Quando a bobina é energizada, ela cria um campo eletromagnético, que funciona como um ímã e portanto atrai e desloca o contato. Então o relé passa a desconectar do NF do contato central que passa a estar conectado no NA. O módulo selecionado para uso na proposta apresentada é ativada com nível lógico baixo.

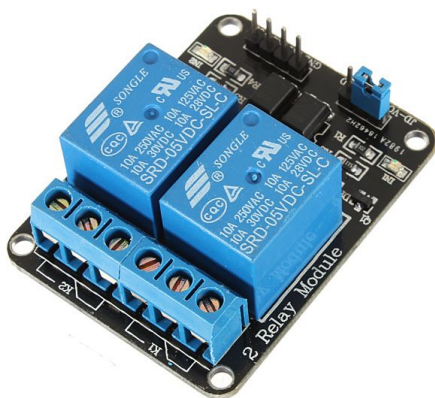


Figura. 3. Módulo relé.

## V. DESCRIÇÃO DE HARDWARE

O diagrama de blocos do sistema se dá pela Fig. 4, onde é possível observar os blocos individuais conectados a plataforma Raspberry Pi: sensor de umidade, sensor de nível, módulo relé e Webcam.

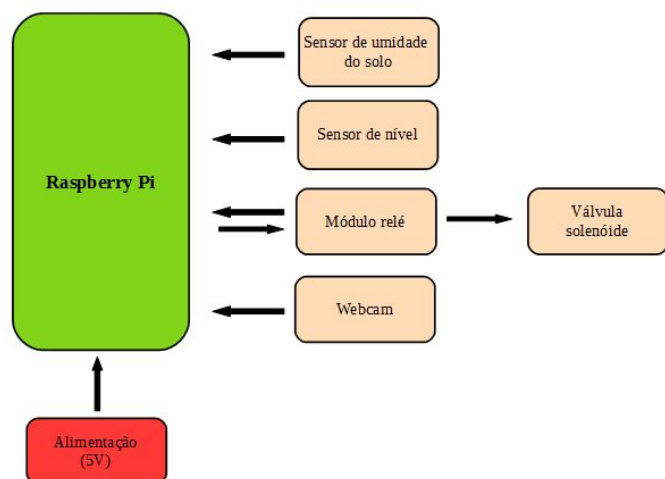


Figura. 4. Diagrama de blocos.

## VI. DESCRIÇÃO DE SOFTWARE

Inicialmente cada bloco foi testado de forma individual a fim de verificar seu comportamento junto a plataforma Raspberry Pi.

### A. Sensor de umidade.

O nível de umidade do solo é detectado pelo sensor de umidade que apresenta o seguinte comportamento: quando o ambiente onde o sensor estiver inserido detectar baixa umidade ele enviará um sinal a Raspberry Pi que apresenta a mensagem de baixa ou alta umidade. O sensor se comunica com a Raspberry pelo modo digital e por isso, este foi calibrado em seu próprio módulo através do potenciômetro,

que permite definir qual é o nível que divide o ligado e desligado.

A rotina de funcionamento deste sensor ilustra o seguinte comportamento: a Raspberry irá ler os dados do sensor através do pino 0 indicando que quando este permanece em contato com uma região úmida, maior será o nível de condutividade elétrica do sensor, o que significa que a resistência do sensor é baixa e a tensão é alta. Isto implica em uma saída com nível lógico alto.

No entanto, quanto menor a quantidade de água no solo menor será o nível de condutividade elétrica do sensor, a resistência do sensor é alta o que implica em um nível baixo de tensão.

### B. Sensor de nível.

A rotina escrita para o sensor indica que o pino da Raspberry irá detectar o sinal indicado por este. Se o valor lido for 0, indica que o sensor desceu devido a ausência de líquido e a mensagem a ser impressa deve informar que o reservatório está vazio. Caso contrário, a boia do sensor sobre e a mensagem informa que o reservatório está cheio.

O sensor de nível ficou imerso em uma região simulando um reservatório de água. Quando o nível de água foi reduzido o flutuador desceu e o valor lido indicava nível lógico 0. Dessa forma foi impressa a mensagem de reservatório vazio. O comportamento contrário implicou na impressão da mensagem reservatório cheio.

### C. Módulo relé.

O código para verificar o comportamento do relé forçava as entradas 0 ou 1 no pino de entrada do dispositivo a fim de verificar a mudança no estado do contato deste. Ao receber 0 no valor de entrada o relé deve ser acionado e emitir a mensagem informando que está ligado. Caso contrário, será desligado.

### D. Módulo relé e sensor de nível.

Considerando que o módulo relé será acionado somente se o reservatório conter líquido suficiente, foi realizado um teste para verificar o comportamento conjunto destes.

O código indica que quando o pino 1 lê o valor 0, indica que o sensor desceu devido a ausência de líquido e então o relé não será acionado. Ao lê o valor 1, implica que o flutuador do sensor está no nível alto devido a presença de líquido. Por isso, o relé é acionado e o sistema poderá ser irrigado.

### E. Webcam

Para a captura das imagens foram realizados testes utilizando o aplicativo *fswebcam*, de forma a captar imagens pela Webcam. Esse aplicativo é capaz de capturar imagens de várias fontes diferentes, além de realizar pequenas manipulações nas imagens. Ele pode salvar os arquivos em vários formatos, como por exemplo em BAYER, o qual

satisfaz a necessidade do projeto, no requisito de processamento de imagens.

#### F. Envio de dados por email

Já o envio dos dados por email do sistema de irrigação foi testado através do protocolo de comunicação SMTP ( Simple Mail Transfer Protocol), um protocolo padrão de fornecimento de serviços de email em uma rede TCP/IP. Ele é capaz de receber e enviar mensagens de email. Para utilizá-lo no decorrer dos testes, foi necessário baixar a biblioteca ssmtp e configurar o usuário local, o servidor, o agente de transferência e o agente de entrega de email.

## VII. RESULTADOS

### A. Sensor de umidade.

A fig. 5 demonstra o resultado obtido pelo terminal o qual indica os níveis de umidade detectado pelo sensor: seco ou úmido.

```

pi@raspberrypi: ~
Arquivo Editar Abas Ajuda
pi@raspberrypi:~$ sudo ./umidadevai
bash: sudo. umidadevai: comando não encontrado
pi@raspberrypi:~$ sudo ./umidadevai
Solo seco _verificação 0
Solo seco _verificação 1
Solo seco _verificação 2
Solo seco _verificação 3
Solo seco _verificação 4
Solo seco _verificação 5
Solo seco _verificação 6
Solo seco _verificação 7
Solo seco _verificação 8
Solo seco _verificação 9
Solo seco _verificação 10
Solo seco _verificação 11
Solo seco _verificação 12
Solo seco _verificação 13

```

Figura. 5. Resultado no terminal \_Sensor de umidade.

### B. Sensor de nível.

A fig. 6 demonstra o resultado obtido pelo terminal o qual indica a presença ou não de líquido no reservatório e imprime as mensagens: Reservatório cheio ou Reservatório vazio.

```

pi@raspberrypi: ~
Arquivo Editar Abas Ajuda
pi@raspberrypi:~$ cd teste1/
pi@raspberrypi:~/teste1$ gcc -Wall -o teste1 teste1.c -lwiringPi
pi@raspberrypi:~/teste1$ ls
teste1.c
pi@raspberrypi:~/teste1$ sudo ./teste1
Reservatório cheio: 0° Verificação
Reservatório cheio: 1° Verificação
Reservatório cheio: 2° Verificação
Reservatório vazio: 3° Verificação
Reservatório vazio: 4° Verificação
Reservatório vazio: 5° Verificação
Reservatório vazio: 6° Verificação
Reservatório vazio: 7° Verificação
Reservatório vazio: 8° Verificação
Reservatório vazio: 9° Verificação
Reservatório vazio: 10° Verificação
Reservatório vazio: 11° Verificação
Reservatório cheio: 12° Verificação
Reservatório cheio: 13° Verificação
Reservatório cheio: 14° Verificação
Reservatório vazio: 15° Verificação
Reservatório vazio: 16° Verificação

```

Figura. 6. Resultado no terminal \_Sensor de nível.

### C. Módulo relé.

A fig. 7 demonstra o resultado obtido pelo terminal o qual indica os estados do relé: acionado ou desligado.

```

pi@raspberrypi: ~/rele
Arquivo Editar Abas Ajuda
pi@raspberrypi:~$ cd rele/
pi@raspberrypi:~/rele$ gcc -Wall -o rele rele.c -lwiringPi
pi@raspberrypi:~/rele$ ls
rele rele.c rele.c.save
pi@raspberrypi:~/rele$ sudo ./rele
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado
Relé desligado
Relé acionado

```

Figura. 7. Resultado no terminal - relé.

Já a fig. 8 e fig. 9 indicam os valores registrados na entrada no módulo relé: nível lógico 0 (0,79V medido no multímetro) para quando está acionado e nível lógico 1 (3,20V medido no multímetro) quando está desligado.



Figura. 8. Resultado 1 no multímetro \_Relé acionado.



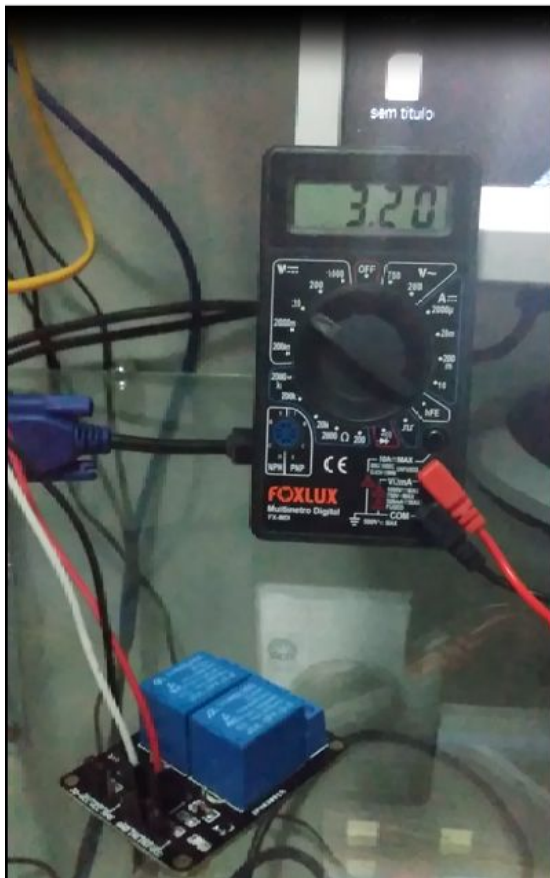


Figura. . Resultado 2 no multímetro \_Relé desligado.

#### D. Módulo relé e sensor de nível.

Resultado apresentado na fig. 10 indica que o relé recebe nível lógico 0, indicado por 0,06V no multímetro, quando o sensor de nível detecta presença de líquido. Neste caso, o sistema permite o processo de irrigação porque o relé será acionado. O que pode ser confirmado pelo led vermelho aceso.

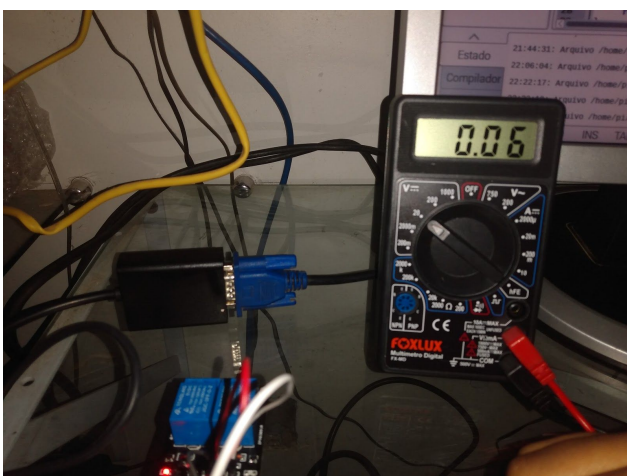


Figura. 10. Resultado 1 no multímetro - relé e sensor.

Já o resultado apresentado na fig. 11 indica que o relé recebe nível lógico 1, indicado por 3,27V no multímetro, em sua entrada e por isso está desligado. Este fato ocorreu porque o sensor de nível identificou que o reservatório estava vazio. O que pode ser confirmado pelo led que, neste caso, está apagado.

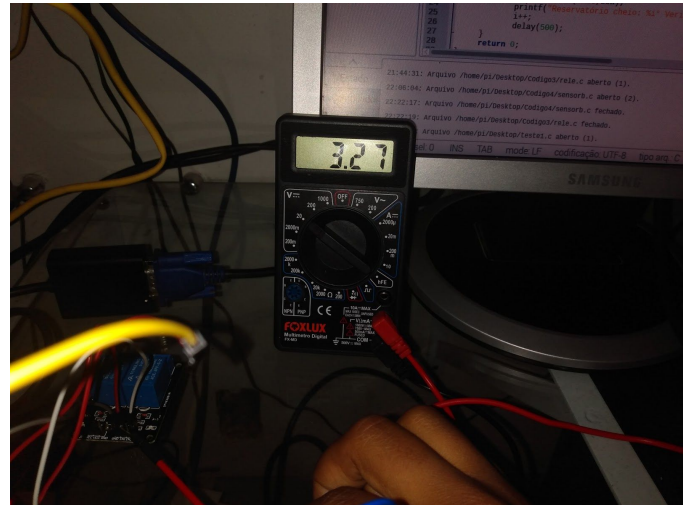


Figura. 11. Resultado 2 no multímetro - relé e sensor.

#### E. Webcam

O resultado apresentado na fig. 12 indica que a plataforma Raspberry Pi reconhece a webcam conectada e realiza a captura de fotos.

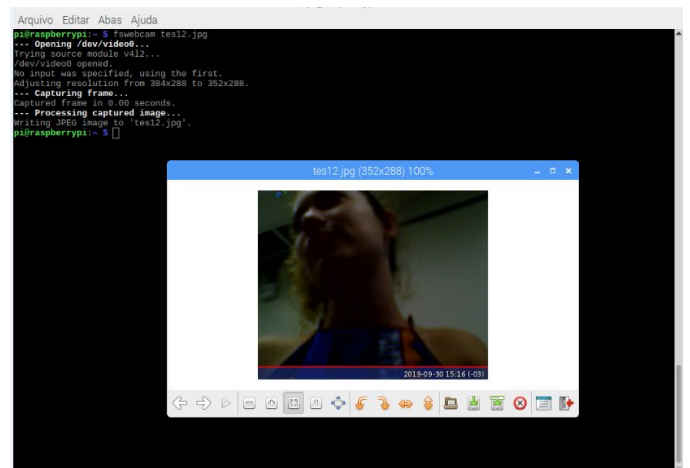
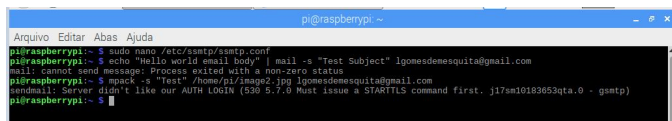


Figura. 12. Resultado Webcam.

#### F. Envio de dados por email.

O envio de dados por email foi executado conforme fig. 13, no entanto, foram encontrados problemas e não foi possível realizar o envio de dados.



```
pi@raspberrypi ~  
Arquivo Editor Abas Ajuda  
pi@raspberrypi:~$ sudo nano /etc/smtplib/smtplib.conf  
pi@raspberrypi:~$ echo "Hello world email body" | mail -s "Test Subject" lgomesdonesquita@gmail.com  
mail: Cannot send message: Process exited with a non-zero status  
pi@raspberrypi:~$ nano /etc/smtplib/smtplib.conf  
sendmail: Server didn't like our AUTH LOGIN (530 5.7.0 Must issue a STARTTLS command first. j17em10183653qta.0 - smtp)  
pi@raspberrypi:~$
```

Figura. 13. Resultado envio de dados.

## VIII. CONCLUSÃO

Os códigos foram desenvolvidos e testados separadamente, uma vez que, o objetivo desta etapa consistiu em verificar como cada componente deve atuar no sistema maior bem como, analisar e encontrar erros de lógica e estruturação dos códigos.

A próxima etapa terá foco na união dos códigos de cada bloco apresentado neste relatório, e também, na verificação de como será realizado o tratamento das imagens capturadas pela webcam. Além disso, serão testadas e adaptadas ao projeto formas de envio de dados via email

## REFERENCIAS

- [1] Fernandes, D. G. Sistema automatizado de controle de estufas para cultivo de hortaliças, 2017.
- [2] PanteShsolutions. Disponível em: <https://www.pantechsolutions.net/smart-irrigation-system-using-iot-and-cloud>. Acesso em 25/08/2019.
- [3] GardenBot. Disponível em: <http://gardenbot.org/howTo/>. Acesso em 26/08/2019.
- [4] WiringPi. Disponível em: <https://learn.sparkfun.com/tutorials/raspberry-gpio/c-wiringpi-api>. Acesso em 24/09/2019.
- [5] Webcam. Disponível em: <https://www.raspberrypi.org/documentation/usage/webcams/>. Acesso em 26/09/2019.
- [6] SSMTP. Disponível em: [https://raspberrypi-projects.com/pi/software\\_utilities/email/ssmtp-to-send-emails](https://raspberrypi-projects.com/pi/software_utilities/email/ssmtp-to-send-emails). Acesso em 28/09/2019.

CÓDIGO: SENSOR DE UMIDADE:

```

#include <stdio.h>
#include <wiringPi.h>
#define umidade 0

int main()
{
    wiringPiSetup();
    pinMode(umidade, INPUT);
    int i = 0;

    for(;;) //teste para verificar a umidade
    {
        if(digitalRead(umidade)<1 //Para ler o estado digital de um pino
        {
            printf("Solo úmido_Verificação %i\n", i);
        }
        else
            printf("Solo seco_Verificação %i\n", i);
        i++;
        delay(500);
    }
    return 0;
}

```

CÓDIGO: MÓDULO RELÉ:

```

#include <stdio.h>
#include <wiringPi.h>

int main()
{
    wiringPiSetup();
    pinMode(1,OUTPUT);

    while(1)
    {
        digitalWrite(1, HIGH); // liga o pino 1
        printf("Relé desligado \n");
        delay(1000);           // espera 1 segundo
        digitalWrite(1, LOW);  // desliga o pino 1
        printf("Relé acionado \n");
        delay(1000);           // espera 1 segundo
    }
    return 0;
}

```

### CÓDIGO: SENSOR DE NÍVEL:

```
#include <stdio.h>
#include <wiringPi.h>

#define boia 1

int main()
{
    wiringPiSetup();
    pinMode(boia,OUTPUT);

    int i = 0;

    for(;;) //teste para verificar o nível de água
    {
        if(digitalRead(boia)<1) //Para ler o estado digital de um pino
        {
            printf("Reservatório vazio: %iº Verificação\n", i);
        }
        else
            printf("Reservatório cheio: %iº Verificação\n", i);
        i++;
        delay(500);
    }
    return 0;
}
```

### CÓDIGO: MÓDULO RELÉ E SENSOR DE NÍVEL:

```
#include <stdio.h>
#include <wiringPi.h>

#define boia 1

int main()
{
    wiringPiSetup();
    pinMode(boia,OUTPUT);

    int i = 0;

    for(;;) //teste para verificar o nível de água
    {
        if(digitalRead(boia)<1) //Para ler o estado digital de um pino
        {
            digitalWrite(rele,LOW);
            printf("Reservatório vazio: %iº Verificação\n", i);
        }
        else
        {
            digitalWrite(rele,HIGH);
            printf("Reservatório cheio: %iº Verificação\n", i);
        }
        i++;
        delay(500);
    }
    return 0;
}
```