

IOT NA AGRICULTURA FAMILIAR - SISTEMA DE IRRIGAÇÃO

Samara Accioly Lins

Descrição do projeto

O projeto tem funcionalidade para o sistema de irrigação, num contexto de agricultura familiar, ou até mesmo para a rega de plantas a nível residencial. O projeto funciona a partir de uma placa com arduino e NodeMCU (WiFi), um motor de engrenagem e 3 leds que funcionam como atuadores, além de um sensor de umidade do solo. Assim, as informações coletadas pela placa são mandadas para o ThingSpeak que, por sua vez, manda os dados para o aplicativo. No aplicativo, o usuário poderá supervisionar qual o nível da umidade do solo (úmido, médio ou seco). Além do aplicativo, o objeto acenderá um LED que corresponde ao nível de umidade. É válido ressaltar que esse projeto está aliado com políticas públicas para ampliar o acesso à internet no meio rural e no centro do país, uma vez que, nessas regiões, há uma baixíssima oferta de internet.

Lista de materiais (Hardware)

- Placas de controle e comunicação:

Arduino e NodeMCU

- Sensores:

Sensor de umidade do solo

- Atuadores:

1 motor de engrenagem

3 LEDS (vermelho, amarelo e verde)

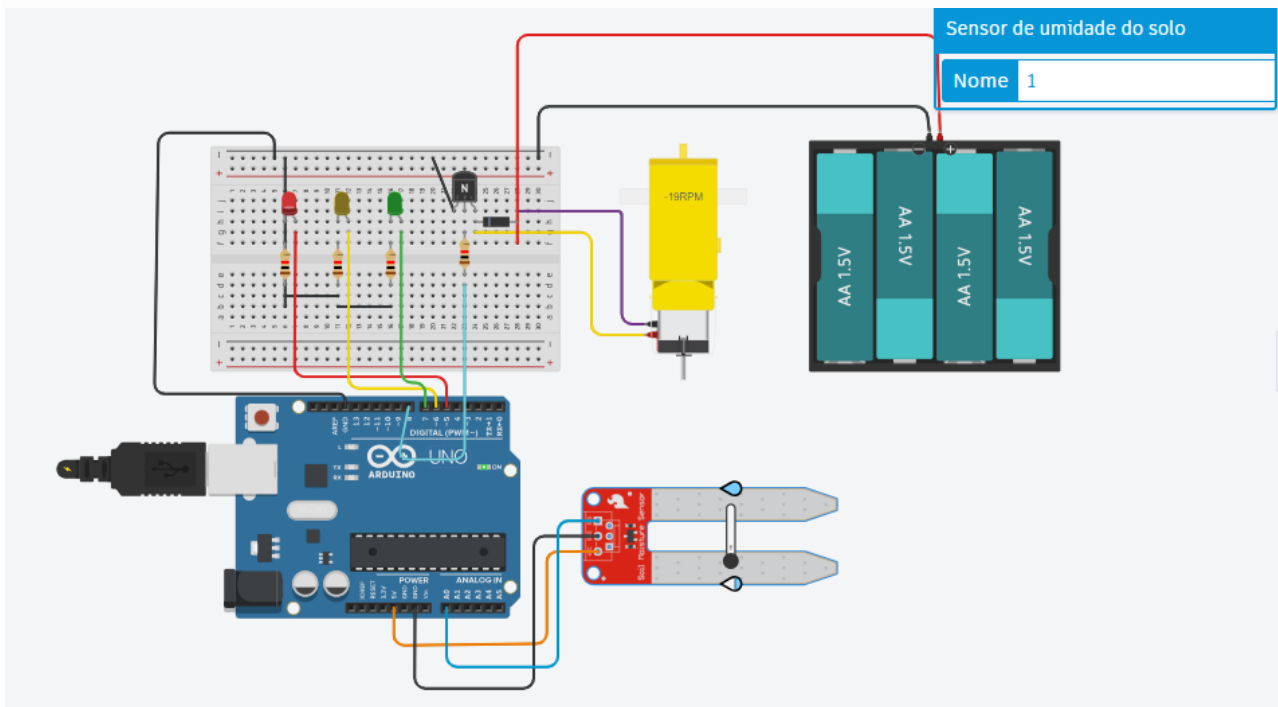
- Componentes auxiliares

Resistores e jumpers

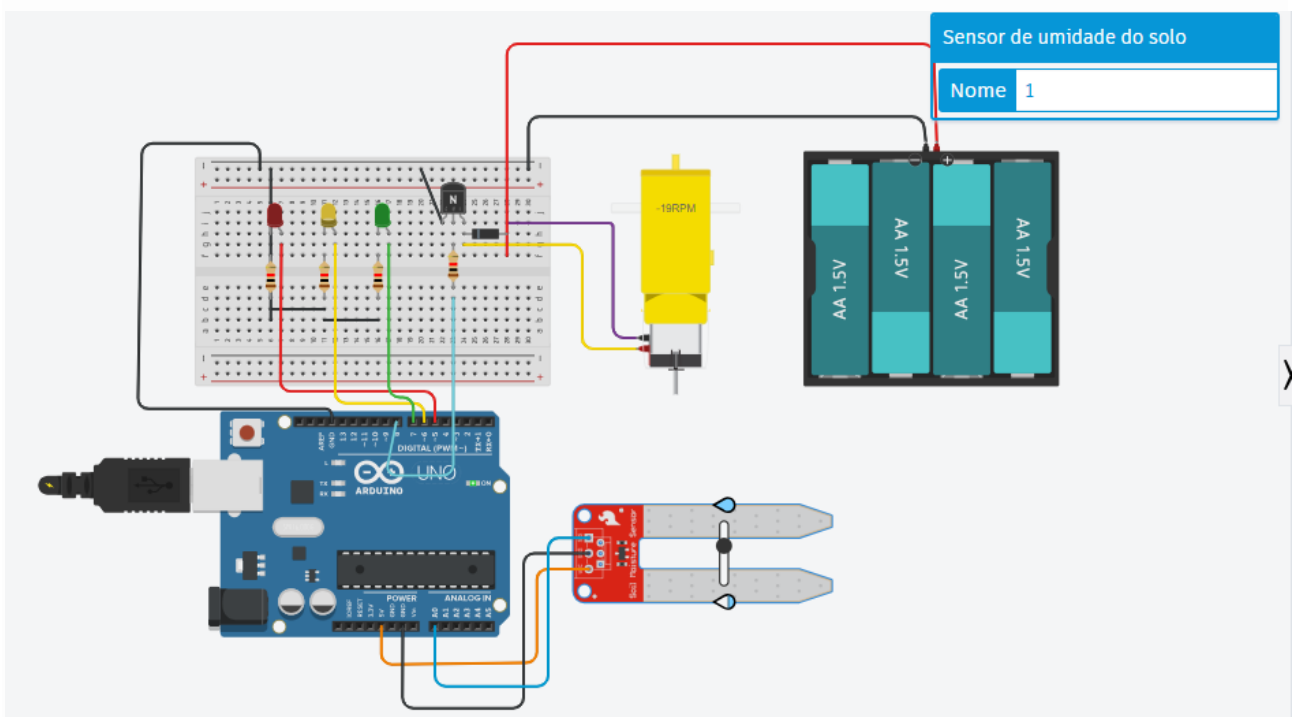
Circuito

Como o Tinkercad não possui o NodeMCU e eu não consigo custear o Fritzing ou as próprias peças, fiz o circuito sem o NodeMCU (que no esquema corresponde ao arduino)

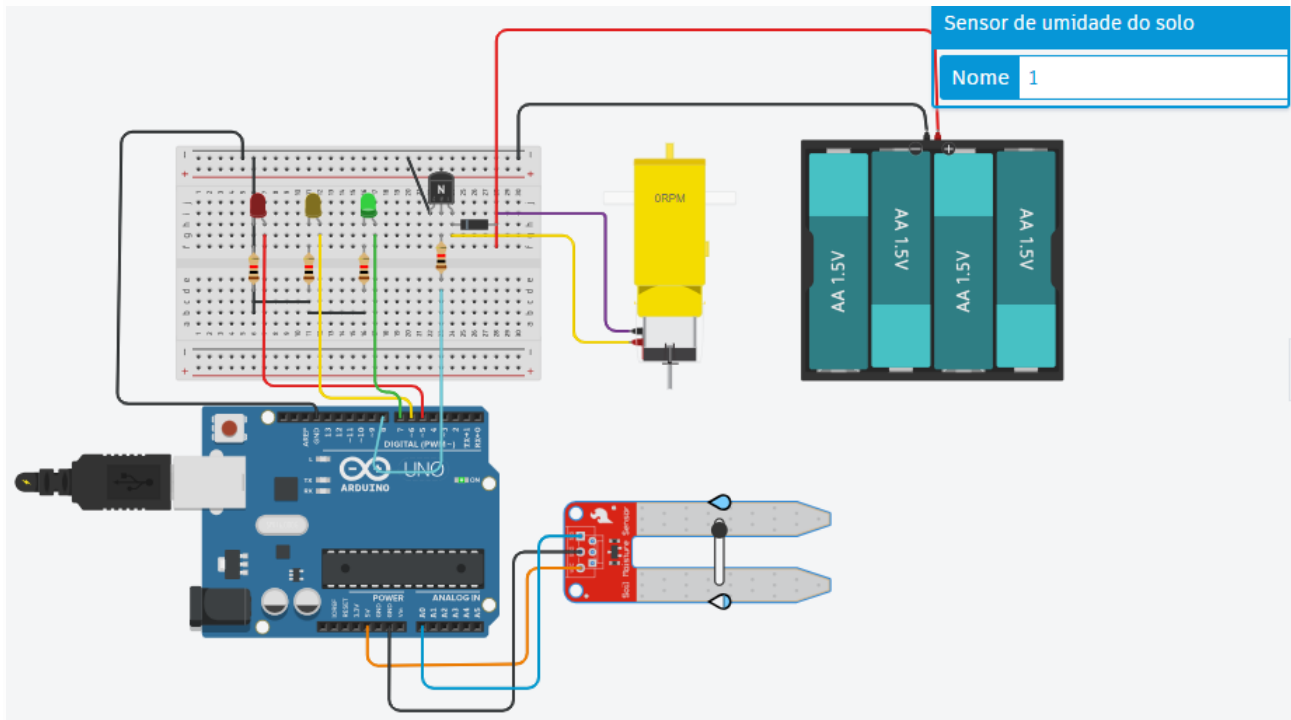
led vermelho aceso e motor ligado (solo seco)



Led amarelo aceso (solo com média umidade)



Led verde aceso e motor desligado (solo úmido)



Código no Tinkercad

```
#define pino_sinal_analogico A0

#define pino_led_vermelho 5

#define pino_led_amarelo 6

#define pino_led_verde 7

#define pinoMotor 8

int valor_analogico = 0;

void setup() {

  Serial.begin(9600);

  pinMode(pino_sinal_analogico, INPUT);

  pinMode(pino_led_vermelho, OUTPUT);
```

```
pinMode(pino_led_amarelo, OUTPUT);

pinMode(pino_led_verde, OUTPUT);

pinMode(pinoMotor, OUTPUT);

digitalWrite(pinoMotor, LOW); }

void loop() {

valor_analogico = analogRead(pino_sinal_analogico); //Le o valor do pino A0 do sensor Serial.print("Porta analogica: "); //Mostra
o valor da porta analogica no serial monitor

Serial.print(valor_analogico);

    //Solo umido, acende o led vermelho

    if (valor_analogico > 0 && valor_analogico < 400) {

Serial.println(" Status: Solo seco");

apagaleds();

        digitalWrite(pino_led_vermelho, HIGH);

        digitalWrite(pinoMotor, HIGH); }

    //Solo com umidade moderada, acende led amarelo

    if (valor_analogico > 400 && valor_analogico < 800) {

        Serial.println(" Status: Umidade moderada");

        apagaleds();

        digitalWrite(pino_led_amarelo, HIGH); }

    //Solo seco, acende led verde

    if (valor_analogico > 800 && valor_analogico < 1024) {

        Serial.println(" Status: Solo úmido");

        apagaleds();

        digitalWrite(pino_led_verde, HIGH);

        digitalWrite(pinoMotor, LOW); }

    delay(100); }
```

```
void apagaLEDs() {  
  
    digitalWrite(pino_led_vermelho, LOW);  
  
    digitalWrite(pino_led_amarelo, LOW);  
  
    digitalWrite(pino_led_verde, LOW); }  

```

Código com NodeMCU

```
#include "ThingSpeak.h"  
#include <ESP8266WiFi.h>  
const char* ssid = "Rose";  
const char* password = "não vou informar aqui";  
using long channel = 2350398;  
int pino_led_vermelho = 5;  
int pino_led_amarelo = 6;  
int pino_led_verde = 7;  
int pinoMotor = 8;  
WiFiClient client;  
using long channel = 2350402;  
int pino_sinal_analogico = A0;  
int valor_analogico = 0;  
WiFiClient client;  
void setup () {  
    Serial.begin (115200);  
    delay (100);  
    pinMode(pino_led_vermelho, OUTPUT);  
    pinMode(pino_led_amarelo, OUTPUT);  
    pinMode(pino_led_verde, OUTPUT);  
    pinMode(pinoMotor, OUTPUT);  
    digitalWrite(pinoMotor, LOW);  
    pinMode(pinoMotor, OUTPUT);  
    digitalWrite(pinoMotor, LOW); }  
    Serial.println ("conectando-se a ");  
    Serial.println (ssid);  
    WiFi.begin (ssid, password);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay (500);  
        Serial.print("."); }  
    Serial.println ("");  
    Serial.println ("WiFi conectado");  
    Serial.println ("Endereço IP: ");  
    Serial.println (WiFi.localIP() );  
    Serial.println (Máscara: " ");  
    Serial.println (WiFi.subnetMask() );  
    Serial.println ("Gateway: ");  
    Serial.println (WiFi.gatewayIP() );  
    ThingSpeak.begin (client);  
}  
void loop() {
```

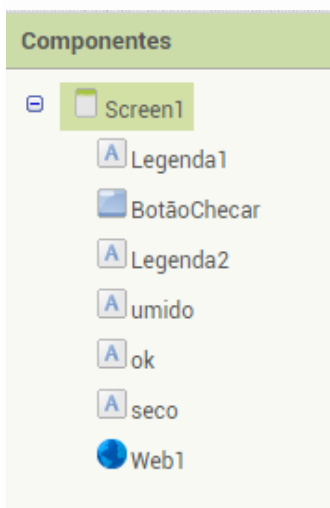
```

valor_analogico = analogRead(pino_sinal_analogico); //Le o valor do pino A0 do sensor Serial.print("Porta analogica: "); //Mostra
o valor da porta analogica no serial monitor
Serial.print(valor_analogico);
//Solo umido, acende o led vermelho
if (valor_analogico > 0 && valor_analogico < 400) {
Serial.println(" Status: Solo seco");
apagaleds();
digitalWrite(pino_led_vermelho, HIGH);
digitalWrite(pinoMotor, HIGH); }
//Solo com umidade moderada, acende led amarelo
if (valor_analogico > 400 && valor_analogico < 800) {
Serial.println(" Status: Umidade moderada");
apagaleds();
digitalWrite(pino_led_amarelo, HIGH); }
//Solo seco, acende led verde
if (valor_analogico > 800 && valor_analogico < 1024) {
Serial.println(" Status: Solo úmido");
apagaleds();
digitalWrite(pino_led_verde, HIGH);
digitalWrite(pinoMotor, LOW); }
delay(100); }
void loop() {
valor_analogico = analogRead(pino_sinal_analogico); //Le o valor do pino A0 do sensor Serial.print("Porta analogica: "); //Mostra
o valor da porta analogica no serial monitor
Serial.print(valor_analogico);
//Solo umido, acende o led vermelho
if (valor_analogico > 0 && valor_analogico < 400) {
Serial.println(" Status: Solo seco");
apagaleds();
digitalWrite(pino_led_vermelho, HIGH);
digitalWrite(pinoMotor, HIGH); }
//Solo com umidade moderada, acende led amarelo
if (valor_analogico > 400 && valor_analogico < 800) {
Serial.println(" Status: Umidade moderada");
apagaleds();
digitalWrite(pino_led_amarelo, HIGH); }
//Solo seco, acende led verde
if (valor_analogico > 800 && valor_analogico < 1024) {
Serial.println(" Status: Solo úmido");
apagaleds();
digitalWrite(pino_led_verde, HIGH);
digitalWrite(pinoMotor, LOW); }
delay(100); }
void apagaleds() {
digitalWrite(pino_led_vermelho, LOW);
digitalWrite(pino_led_amarelo, LOW);
digitalWrite(pino_led_verde, LOW); }

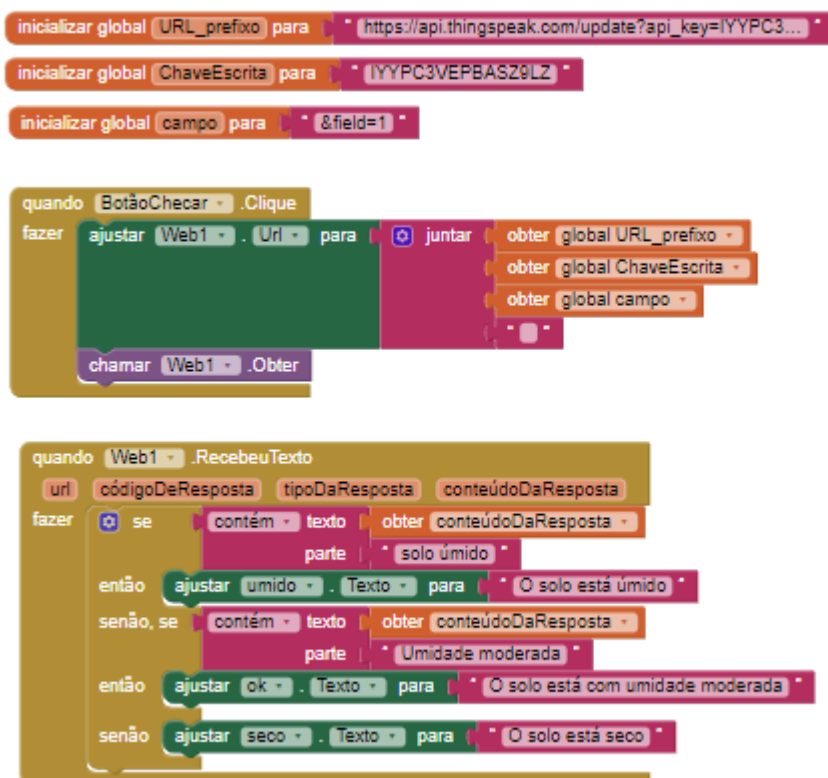
```

Descrição e imagem do aplicativo

Design e componentes do aplicativo.



Parte dos blocks



O usuário irá checar as informações sobre a umidade do solo. Dependendo do valor da umidade, ele obterá a mensagem se o solo está úmido, com umidade moderada ou seco.

Links

Circuito no Tinkercad:

<https://www.tinkercad.com/things/93M9bTZcSzQ-fantastic-gaaris>

Aplicativo app inventor:

<https://gallery.appinventor.mit.edu/?galleryid=4477ab7e-b0e2-4e98-b48a-146e26bc0c24>

ThingSpeak: <https://thingspeak.com/channels>

Em razão de não conseguir ter o NodeMCU, não será possível ver o aplicativo funcionando. No entanto, neste arquivo tem todas as etapas bem descritivas e, se aplicadas com o esp8266 será possível obter a funcionalidade, ao menos em teoria (já que não foi possível o teste).

