



# TINY HOUSE

## IOT

APLICAÇÃO EM INTERNET DAS COISAS



ENGENHARIA  
DA COMPLEXIDADE



GUIA DO ESTUDANTE

# SUMÁRIO



## INTRODUÇÃO

**01**

## ARRANJO EXPERIMENTAL

**02**

## OBJETIVO

**03**

## FUNÇÃO 1

Sistema de Alarme

**04**

## FUNÇÃO 2

Sistema de Luminosidade

**07**

## FUNÇÃO 3

Sistema de Refrigeração

**09**

## FUNÇÃO 4

Sistema de acionamento de Porta e Janela

**12**

## EXTRA

**14**

## DESAFIO

**15**

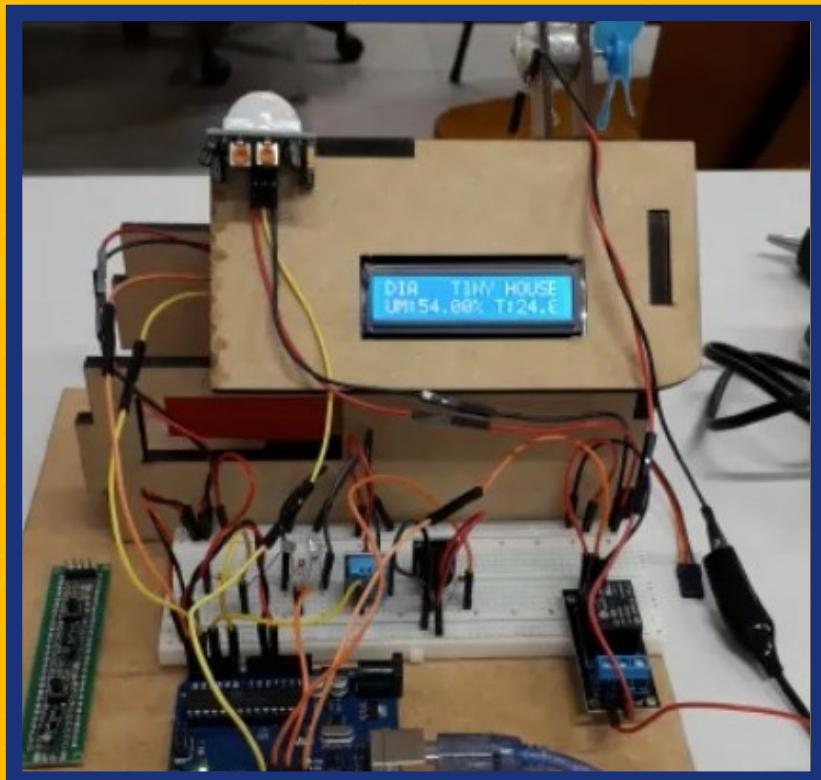
## KAHOOT

**16**

# INTRODUÇÃO



Olá a todos, durante este Tempo Prático (TP) vocês vão ajudar meu amigo Jordan que quer transformar sua casa em uma casa automatizada inteligente. Como é um excelente faz-tudo, já instalou sensores e atuadores dentro de sua casa. Mas ele não tem nenhuma habilidade de programação, então ele precisa de sua ajuda para esta etapa. Abaixo uma foto da casa de Jordan com todos os sensores e atuadores instalados:



Para gerenciar sua casa inteligente, Jordan quer usar uma placa Arduino Uno e já realizou as conexões de todos os sensores/atuadores para a placa Arduino. Abaixo você encontra a lista de todos os sensores/atuadores instalados e quais são suas respectivas conexões com a placa Arduino.

# ARRANJO EXPERIMENTAL



A lista de material utilizado neste TP pode ser vista na [APLICAÇÃO EM INTERNET DAS COISAS](#) tabela a seguir:

Componentes	Conexão no Arduino
Sensor PIR	Digital 9
Buzzer passivo	Digital 10
Push button 1	Digital 2
LED vermelho	Digital 12
Ventilador	Digital 13
Push button 2	Digital 3
Servo_1 (controle da porta)	Digital 5 (PWM)
Servo_2 (controle da janela)	Digital 6 (PWM)
LCD (16x2 caracteres)	SDA (A4 - i2c)
LCD (16x2 caracteres)	SDL (A5 - i2c)
Sensor LDR	Analógico A0
Sensor de temperatura	Analógico A2

Jordan deu a lista das diferentes funções que deseja implementar. Você os encontrará descritos nas próximas sessões. Vamos começar!



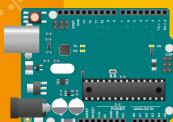
SE VOCÊ NÃO ESTÁ MUITO FAMILIARIZADO COM O AMBIENTE ARDUINO, VERIFIQUE A DOCUMENTAÇÃO DO ARDUINO [AQUI](#). VOCÊ ENCONTRARÁ UMA DESCRIÇÃO DE TODAS AS FUNÇÕES, VARIÁVEIS E ESTRUTURAS BÁSICAS DO ARDUINO.

# OBJETIVO



TINY HOUSE

IOT  
APLICAÇÃO EM INTERNET DAS COISAS



CONHECER E COMPREENDER O FUNCIONAMENTO DA PLACA ARDUINO;



EXECUTAR E DEPURAR UM PROGRAMA C++ FEITO NO IDE DO ARDUINO;



ENTENDER OS DIAGRAMAS ELÉTRICOS DE CIRCUITOS ELETRÔNICOS;



IDENTIFICAR E COMPREENDER A FUNÇÃO DOS COMPONENTES DO SISTEMA E RESPECTIVAS SIMBOLOGIAS;



REALIZAR O ACIONAMENTO DE BOTÕES, BUZZER, LED, MOTOR DC E SERVO MOTOR; E FAZER A LEITURA DE SENsoRES PIR, LDR, UMIDADE E TEMPERATURA.

# FUNÇÃO 1

## Sistema de Alarme



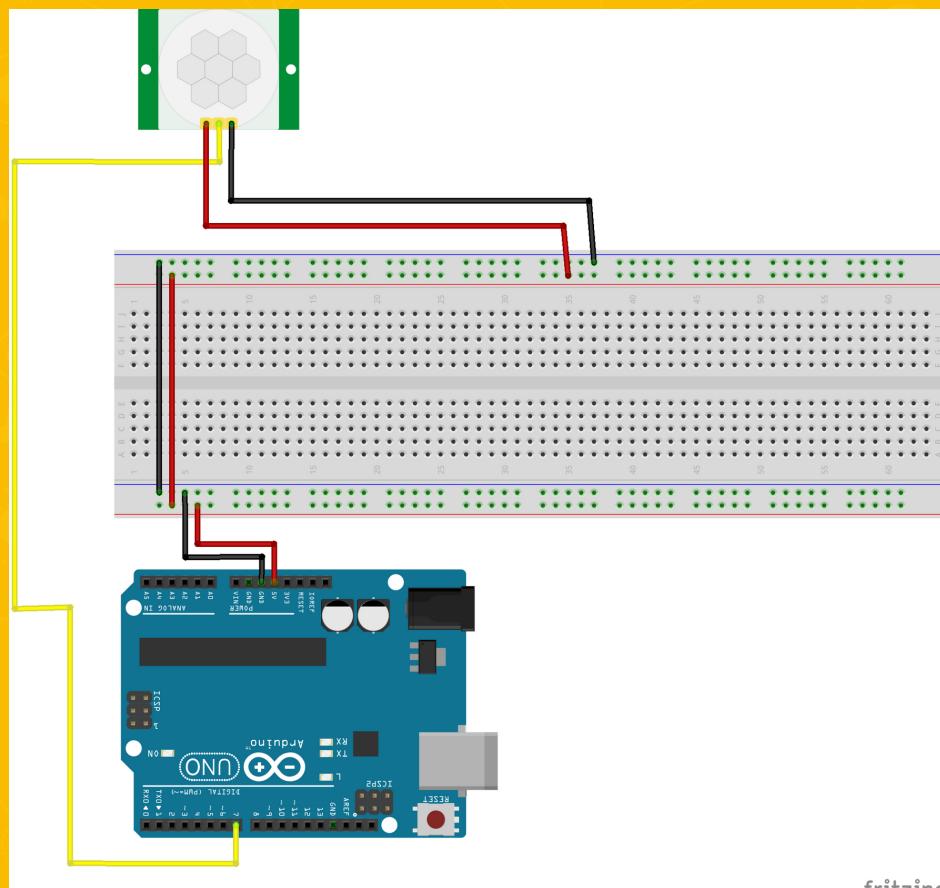
TINY HOUSE  
IOT  
DISPOSITIVO DA INTERNET DOS COISAS

IMPLEMENTAR UM ALARME SONORO ACIONADO QUANDO ALGUÉM É DETECTADO PELO SENSOR DE MOVIMENTO.



JORDAN QUER AJUDA PARA REALIZAR UM PROGRAMA QUE LIGA UM ALARME QUANDO O SENSOR PIR DETECTA UM MOVIMENTO E QUE DEMORE 5 S PARA DESLIGAR O ALARME QUANDO O SENSOR PIR NÃO DETECTA MAIS NADA.

### ESQUEMÁTICO SENSOR PIR



fritzing

# FUNÇÃO 1

## Sistema de Alarme



TINY HOUSE  
IOT  
APRENDA NA INTERNET DOS COISAS

### CÓDIGO DO SENSOR PIR

```
PIR §

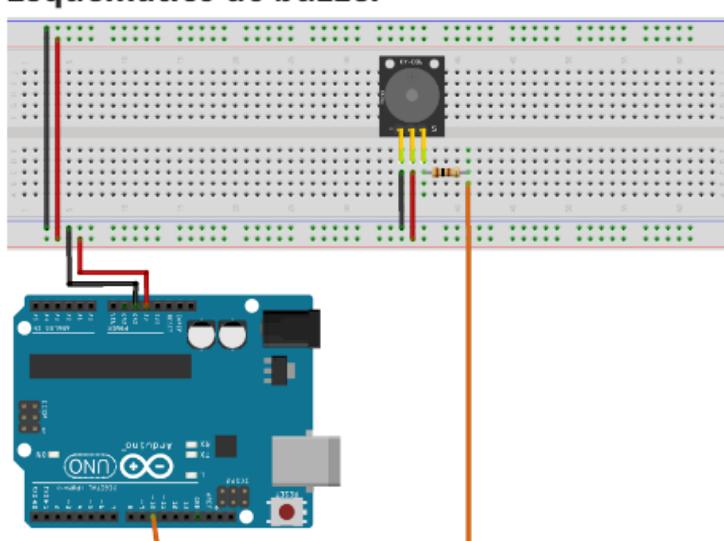
const int sensor = 9; // Guarda o valor 8 na variável int (inteiro) chamado sensor
int leitura = 0; // Define uma variável chamada leitura
void setup() {
    pinMode(sensor, INPUT); // Define a variável sensor como um pino e como um "INPUT" ou entrada
    Serial.begin(9600); // Começa o serial com uma taxa de transferência de 9600
}

void loop() {
    leitura = digitalRead(sensor); // Armazena o valor do sensor na variável leitura
    // A função "digitalRead" faz a leitura da variável como HIGH (ativo) e LOW (desativo)

    // O "if" é uma função condicional onde só irá passar ele se essa condição for atendida
    if (leitura == HIGH) {
        Serial.println("Movimento detectado"); // Escreve no serial o valor da leitura
    }
    // A função else é nada mais que a consequência de se o "if" acima não for atendido
    else {
        Serial.println("Nenhum movimento detectado"); // Escreve no serial o valor do sensor
    }

    delay(200); // Tempo (0.2 segundos) para diminuir a velocidade do código
}
```

### ESQUEMÁTICO BUZZER



# FUNÇÃO 1

## Sistema de Alarme



TINY HOUSE  
IOT  
Inteligência em Internet das Coisas

### CÓDIGO DO BUZZER

```
buzzer§  
const int buzz = 10; // Guarda o valor 10 na variável buzz  
void setup() {  
    pinMode(buzz, OUTPUT); // Define buzz (10) como um pino e define como "OUTPUT" ou saída  
}  
  
void loop() {  
    tone(buzz, 200); // Ativa o som do buzzer com uma frequência de 200  
    delay(500); // Tempo de 0.5 segundos  
    noTone(buzz); // Desativa o som do buzzer  
    delay(500);  
    tone(buzz, 1000); // Ativa o som do buzzer com uma frequência de 1000  
    delay(2000);  
}
```



AGORA É A SUA VEZ

JUNTE OS DOIS CIRCUITOS E AJUDE A JORDAN  
A MONTAR O SISTEMA DE ALARME!

# FUNÇÃO 2

## Sistema de Luminosidade

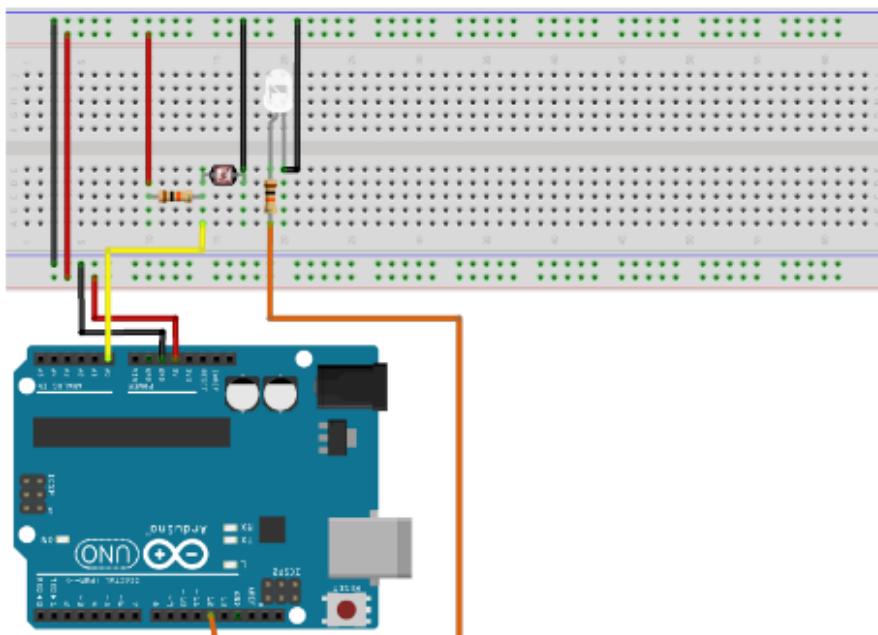


IMPLEMENTAR UMA FOTOCÉLULA COM SENSOR LDR PARA CONTROLAR A LUMINOSIDADE EXTERNA E REDUZIR O CONSUMO DE ENERGIA



JORDAN QUER AJUDA PARA REALIZAR UM PROGRAMA QUE LIGA A LUZ DO LED QUANDO O SENSOR LDR DETECTA QUE A LUMINOSIDADE ESTÁ BAIXA.

### ESQUEMÁTICO LDR COM LED



# FUNÇÃO 2

## Sistema de Luminosidade



### CÓDIGO DO SENSOR LDR

```
LDR $  
#include <Wire.h>  
const int LDR = A0; // Pino analógico que o sensor está conectado  
int valorsensor; // valor que sera armazenado o valor do LDR  
void setup() {  
  
    pinMode(LDR, INPUT); // Coloca a variavel LDR como entrada  
    Serial.begin(9600); // inicializa a comunicação serial com a taxa de 9600 bps  
  
}  
  
void loop() {  
    valorsensor = analogRead(LDR); // Faz a leitura do pino analógico LDR e armazena o valor na variavel valorsensor  
    Serial.print("LDR : "); // Mostra o valor no monitor serial  
    Serial.println(valorsensor);  
    if (valorsensor < 600) { // Se o valor de VALORENSOR for menos que 600  
  
        Serial.println(" TURNO: DIA ");  
    } else { // Se não  
  
        Serial.println("TURNO: NOITE");  
    }  
    delay(1000); // Aguarda 100 milisegundos  
}
```

### CÓDIGO DO LED

```
led $  
const int luz = 12; // Pino Referente ao led  
void setup() {  
    pinMode(luz, OUTPUT); // Coloca variável luz é um sinal de saída  
}  
  
void loop() {  
    digitalWrite(luz, HIGH); // Ligar led  
    delay(400); // Aguarda 0.4 segundo  
    digitalWrite(luz, LOW); // Desligar led  
    delay(1000); // Aguarda 1 segundo  
}
```



AGORA É A SUA VEZ

JUNTE OS DOIS CIRCUITOS E AJUDE A JORDAN  
A MONTAR O SISTEMA DE LUMINOSIDADE!

# FUNÇÃO 3

## Sistema de Refrigeração

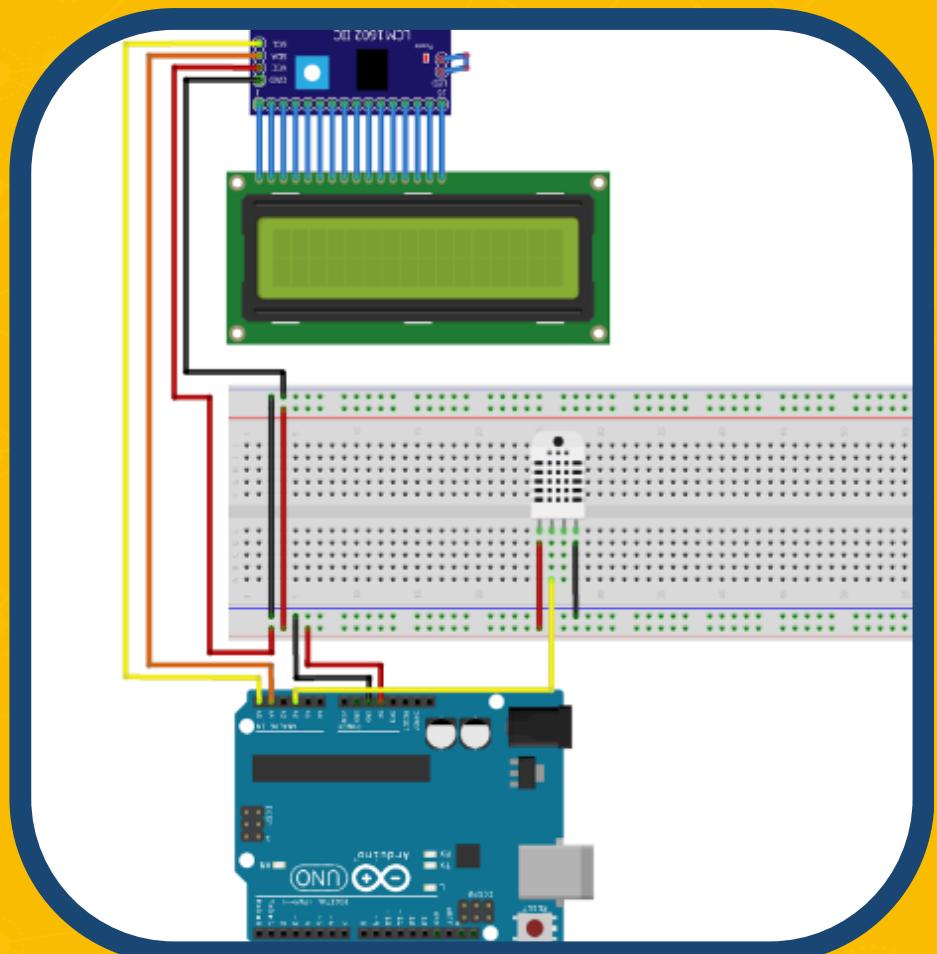


IMPLEMENTAR UM SISTEMA DE REFRIGERAÇÃO AUTOMÁTICA COM VENTILADOR E SENSOR DE TEMPERATURA.



JORDAN QUER AJUDA PARA REALIZAR UM PROGRAMA QUE LIGA UM VENTILADOR QUANDO O SENSOR DHT11 DETECTA UMA TEMPERATURA ACIMA DE 32°C E APRESENTAR ESSES VALORES NA LCD.

### ESQUEMÁTICO SENSOR DHT11 COM A LCD



# FUNÇÃO 3

## Sistema de Refrigeração



### CÓDIGO DO SENSOR DHT11 - SENSOR DE TEMPERATURA

```
Sensor_de_temperatura
#include "dht.h" //Inclusão da biblioteca

const int pinoDHT11 = A2; //Pino analogico pelo DHT11

dht DHT; //Variável do tipo dht

void setup() {
  Serial.begin(9600); //Inicializa o serial
  delay(1000); //Intervalo de 1 segundo
}

void loop() {
  DHT.read11(pinoDHT11); //Lê as informações do sensor
  Serial.print("Umidade: "); //Imprime a umidade do sensor
  Serial.print(DHT.humidity); //Imprime a umidade no serial
  Serial.print("%"); //Escreve o texto "%"
  Serial.print(" / Temperatura: ");
  Serial.print(DHT.temperature, 0); //Imprime a temperatura e tira a parte decimal
  Serial.println("*C");
  delay(2000); //Intervalo de 2 segundos
}
```

### CÓDIGO DO LCD

```
LCD
#include <Wire.h> // Inicializa a biblioteca "Wire" para a comunicação do arduino com os outros dispositivos
#include <LiquidCrystal_I2C.h> // Inicializa a biblioteca "LiquidCrystal_I2C" para o manuseio do LCD através do módulo I2C

//Inicializa o display no endereço 0x27 contendo 16 colunas e 2 linhas
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init(); // Serve para iniciar a comunicação com o display já conectado
  lcd.backlight(); // Serve para ligar a luz do display
}

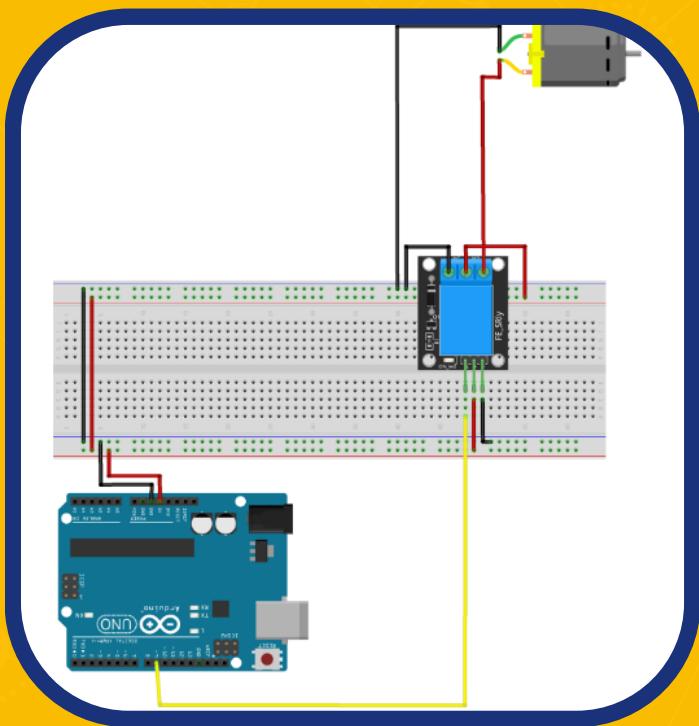
void loop() {
  lcd.setCursor(0, 0); // Coloca o cursor do display na coluna 1 e linha 1
  lcd.print(" WELCOME! "); // Escreve "TINY HOUSE" no LCD
  delay(5000); // Aguarda 5 segundos
  lcd.clear(); // Apaga o texto do lcd
  lcd.setCursor(0, 1); // Coloca o cursor do display na coluna 4 e linha 1
  lcd.print(" TINY HOUSE");
  delay(3000); // Aguarda 3 segundo
}
```

# FUNÇÃO 3

Sistema de Refrigeração TINY HOUSE



## ESQUEMÁTICO MOTOR DC



## CÓDIGO DO MOTOR DC

```
motor_com_rel_
const int motor = 13; // Define o pino do motor

void setup() {
    pinMode(motor, OUTPUT); // Define o motor como saída
}

void loop() {
    digitalWrite(motor, HIGH); // Ativa o motor
    delay(5000); // Aguarda 5 segundos
    digitalWrite(motor, LOW); // Desativa o motor
    delay(2000);
}
```



AGORA É A SUA VEZ  
JUNTE OS TRÊS CIRCUITOS E AJUDE A JORDAN  
A MONTAR O SISTEMA DE REFRIGERAÇÃO!

# FUNÇÃO 4

Sistema de acionamento de Porta e Janela

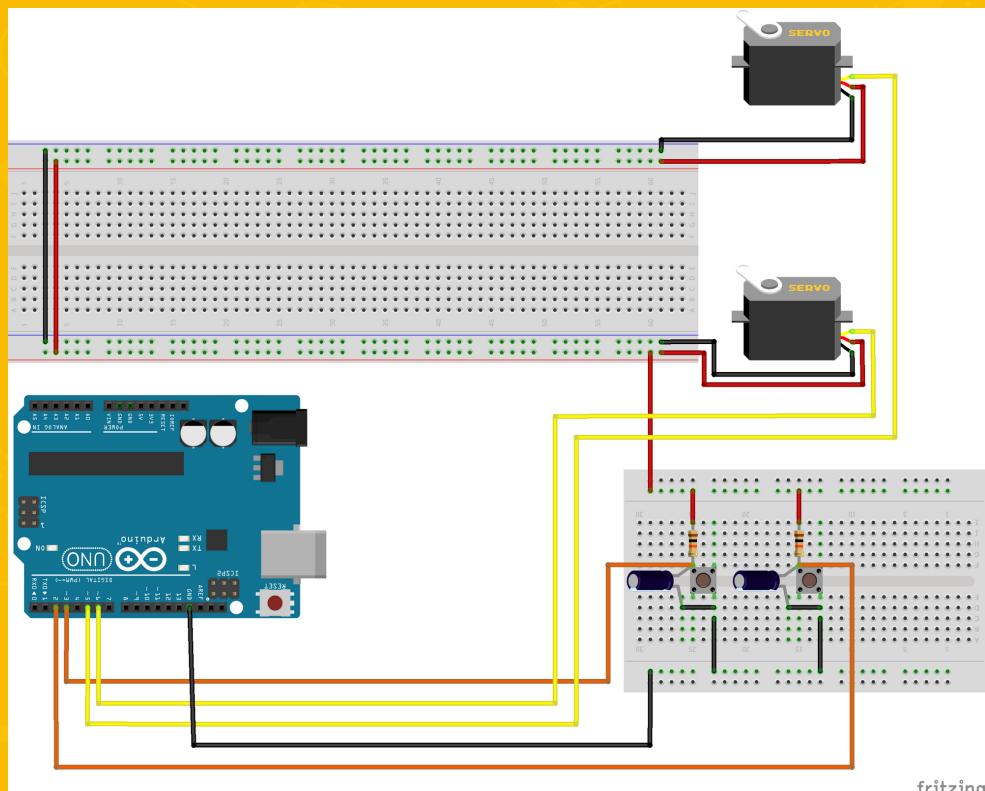


IMPLEMENTAR UM SISTEMA DE CONTROLE PARA ACIONAMENTO DA PORTA E JANELA DA CASA VIA TECLADO DIGITAL.



JORDAN QUER AJUDA PARA REALIZAR UM PROGRAMA QUE ABRE E FECHA A PORTA QUANDO APERTA-SE UM BOTÃO E OUTRO SISTEMA QUE ABRE E FECHA A JANELA QUANDO APERTA-SE O OUTRO BOTÃO.

## ESQUEMÁTICO SERVOS COM OS BOTÕES



# FUNÇÃO 4

Sistema de acionamento de Porta e Janela



## CÓDIGO DO SERVO

```
Servo $  
#include <Servo.h> // Inicializa a biblioteca do servo motor  
Servo servo; // Define uma variavel do tipo servo  
void setup() {  
    servo.attach(6); // Define o servo na porta 6  
}  
  
void loop() {  
    servo.write(180); // Move o servo para a direção 180 que fecha a porta. Para abrir a porta mova o servo para a direção 90 ou menor  
    delay(2000); // Aguarda 2 segundos  
    servo.write(90); // Move o servo para a direção 90 abrindo a porta  
    delay(2000); // Aguarda 2 segundos  
}
```

## CÓDIGO DO BOTÃO

```
bot_o $  
const int botao_1 = 2; // Define o botão 1 no pino 2  
const int botao_2 = 3; // --  
  
int porta, janela = 0;  
  
void setup() {  
    pinMode(botao_1, INPUT_PULLUP); // Define o botão 1 como INPUT ou entrada  
    pinMode(botao_2, INPUT_PULLUP); // --  
    Serial.begin(9600);  
}  
  
void loop() {  
    porta = digitalRead(botao_1);  
    if (porta == LOW) {  
        Serial.println("O botão 1 foi apertado!"); // Escreve a mensagem no serial que o botão foi apertado  
    }  
  
    janela = digitalRead(botao_2);  
    if (janela == LOW) {  
        Serial.println("O botão 2 foi apertado!"); // --  
    }  
}
```



AGORA É A SUA VEZ  
JUNTE OS DOIS CÓDIGOS E IMPLEMENTE O CÓDIGO DO  
SEGUNDO SERVO E AJUDE A JORDAN A MONTAR O  
SISTEMA DE ACIONAMENTO DA PORTA E JANELA!

# EXTRA

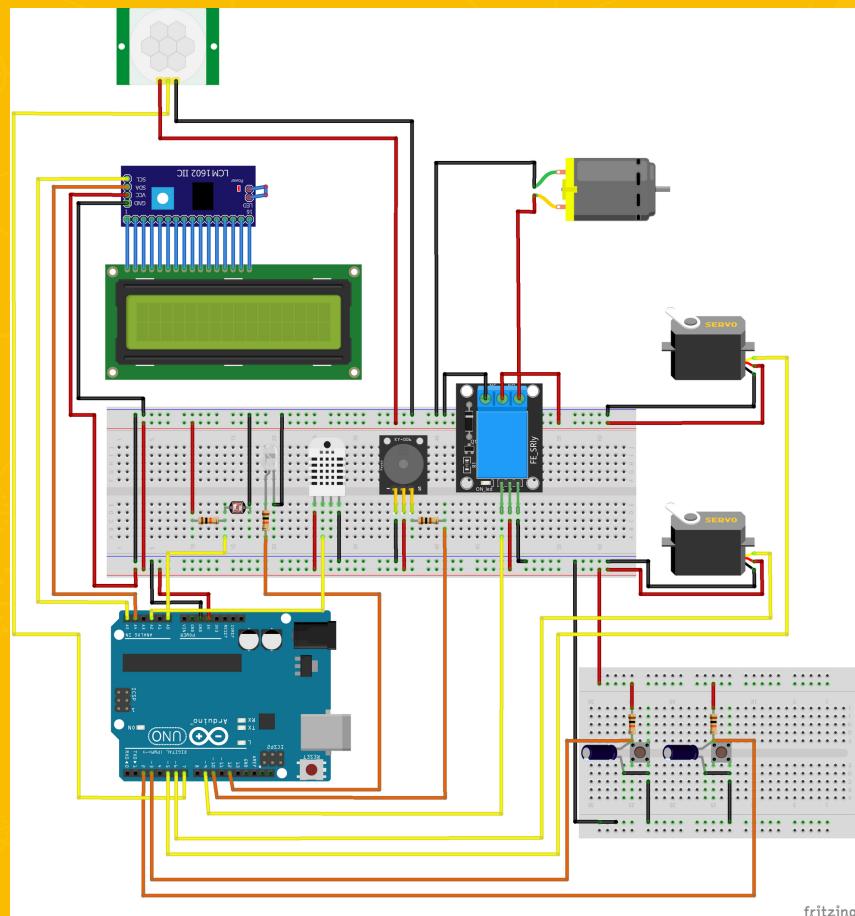


CRIAR UMA FUNÇÃO QUE FECHE A PORTA E JANELA DA CASA AUTOMATICAMENTE APÓS UM DETERMINADO TEMPO QUE ESTÃO ABERTOS (4S).



AJUDE JORDAN A NÃO ESQUECER AS PORTAS E JANELAS ABERTAS, REALIZANDO UM CÓDIGO QUE FECHE A PORTA E JANELA DEPOIS DE 4S SEM RESPOSTA DO BOTÃO.

## ESQUEMÁTICO DO CIRCUITO COMPLETO



# DESAFIO

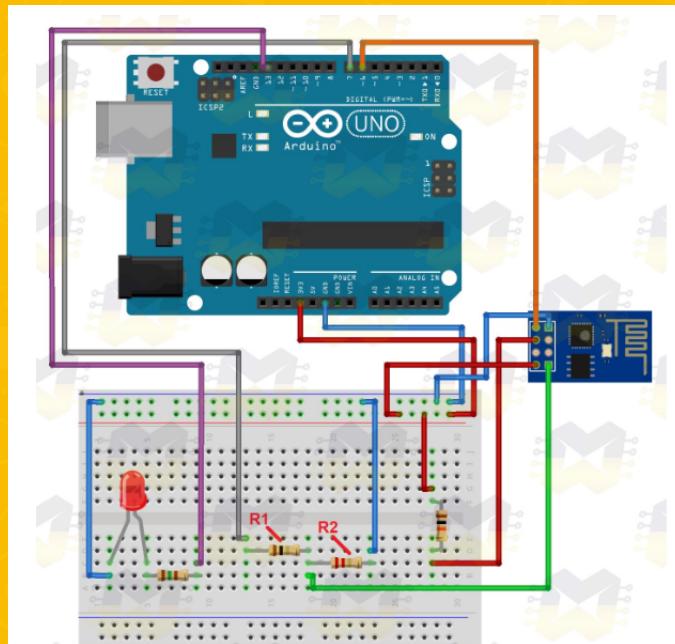


PARA INICIAR NO MUNDO DO IOT, AGORA QUE VOCÊ JÁ TEM A BASE DA ELETRÔNICA, QUE TAL ASCENDER E DESLIGAR UMA FONTE LUMINOSA VIA WIFI?



AJUDE JORDAN A ASCENDER A LAMPADA DA ENTRADA DA SUA CASA, VIA WIFI, QUANDO ESTIVER SE APROXIMANDO DE CASA.

## ESQUEMÁTICO DO CIRCUITO :



## DICA

PARA AUXILIAR A FAZER ESTA ATIVIDADE EM CASA, ACESSE OS DOIS LINKS ABAIXO E SIGA O PASSO A PASSO:

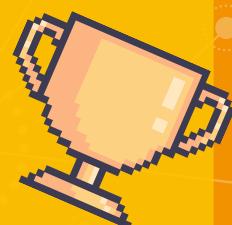
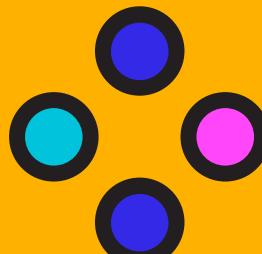
- DESENVOLVER DE FORMA FÍSICA - CASO VOCÊ POSSUA OS COMPONENTES EM CASA.
- DESENVOLVER DE FORMA VIRTUAL - CASO VOCÊ NÃO POSSUA OS COMPONENTES EM CASA.



TINY HOUSE  
IOT

APLICAÇÃO EM INTERNET DAS COISAS

RAHOOOT



VAMOS JOGAR?!

