



Universidade Federal da Paraíba  
Centro de Informática - CI  
Projeto de Introdução a  
Engenharia  
Alunas: Anabelle Carreiro de  
Sousa, Laura de Faria M. Ayres e  
Tassany Onofre de Oliveira

# Introdução ao Projeto

---

## Casa Inteligente (Smart Home)

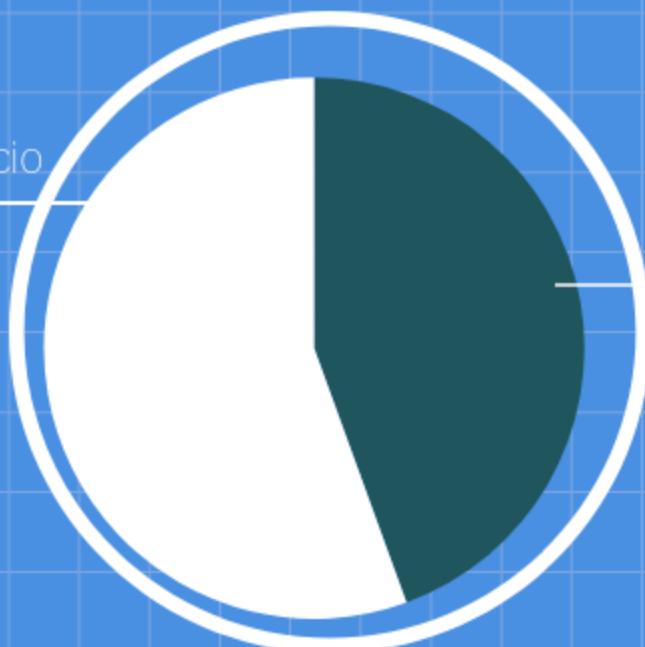
O projeto se baseia na ideia do uso da tecnologia com o objetivo de simplificar e tornar automáticas algumas tarefas habituais. A proposta também está voltada para a área de acessibilidade física.

### Questão Custo - Benefício

---

O grande dilema da questão de automação residencial está relacionada ao alto custo de instalação.

Todavia, a longo prazo, a economia mensal de custos e de energia torna tal opção mais viável.



### Ambiente Inclusivo

---

Um dos pontos mais atrativos do projeto é a seu desenvolvimento de acordo com a necessidade de cada cliente, introduzindo o projeto a esfera de acessibilidade social.

## Orçamento

---



A estimativa do valor gasto para a produção \$R 483.00

---

Período de Produção

Já a duração de fabricação foi entre aproximadamente 14 dias (2 semanas).

# Visão Geral da Casa

## Divisão de Cômodos:

### Área Externa:

Caixa de Correspondência  
Alarme  
Portão  
Sistema de Irrigação

### Sala de Estar:

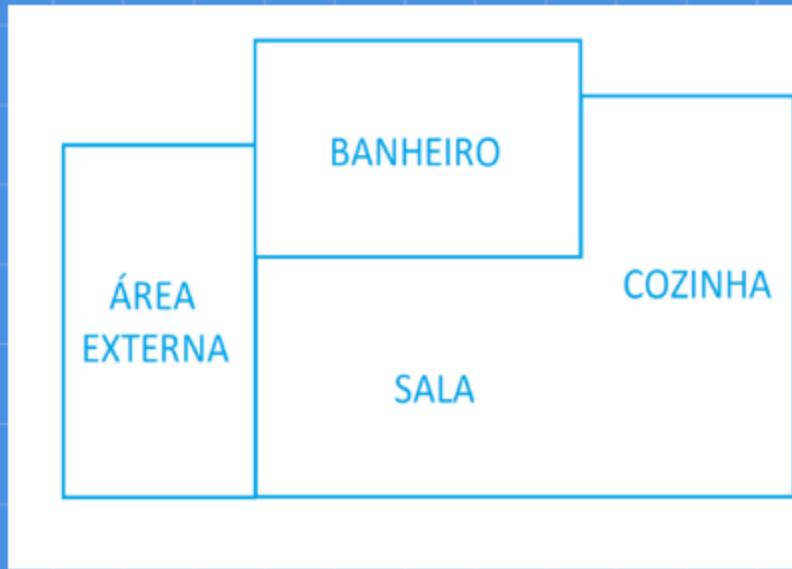
Sistema de Luzes  
Visor

### Cozinha:

Caixa de Remédios  
Lixeira Automática

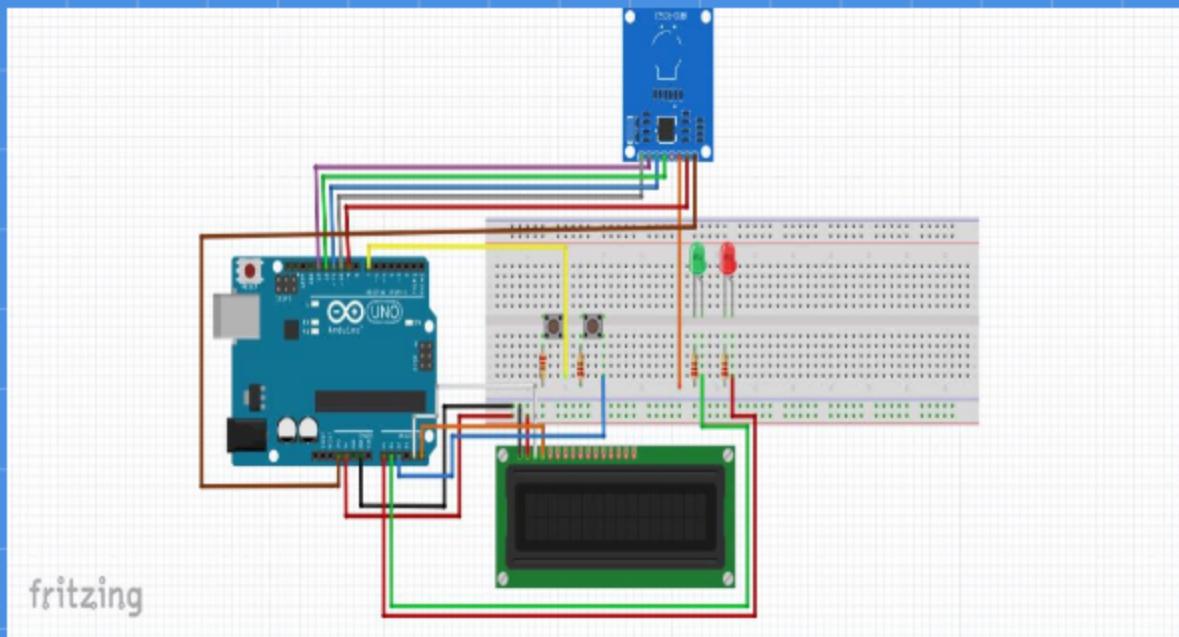
### Banheiro:

Vaso Automático  
Chuveiro



# Área Externa

## 1.Caixa de Correspondência/visor





# Área Externa

Arquivo Editor Sketch Ferramentas Ajuda

sketch\_sep22a | Arduino 1.8.5

```
sketch_sep22a.cpp

1 #include <Wire.h> //inclui as bibliotecas
2 #include <LiquidCrystal_I2C.h>
3 #include <SPI.h>
4 #include <MFRC522.h>
5 #define ledVerde 2 //define o pino do led verde
6 #define ledVermelho 6 //define o pino do led vermelho
7 #define SS_PIN 10
8 #define RST_PIN 9
9 #define botao 8 //define o botão (para quando deixa a correspondência
10 #define botaoG 4 //define o botão (para gravar outro cartão)
11 String senhaAtual=""; //string que contém o código para abrir o correio
12 int flag=0; //variável que altera o seu valor quando um cartão é gravado
13 int caixa=0; //variável que informa se há correspondência ou não
14 int valorGravar=0; //status do botão para gravar outro código
15 int valorBotao1=0; //status do botão para gravar outro código
16 int valorBotao2=1; //status do botão anteriormente
17 LiquidCrystal_I2C lcd(0x3F, 16, 2); //declara o display LCD
18 MFRC522 mfrc522(SS_PIN, RST_PIN);
19 MFRC522::HIFARE_Key key;
```

```
19 MFRC522::HIFARE_Key key;
20
21 String modo_gravacao(String senha){ //função para gravar outro código de cartão, substituindo o
22     while(1){
23         if (! mfrc522.PICC_IsNewCardPresent()) {
24             continue;
25         }
26
27         // Select one of the cards
28         if (! mfrc522.PICC_ReadCardSerial()) {
29             continue;
30         }
31         senha = "";
32         for (byte i = 0; i < mfrc522.uid.size; i++) {
33             Serial.print(mfrc522.uid.uidByte[i], HEX);
34             senha.concat(String(mfrc522.uid.uidByte[i], HEX));
35         }
36         flag=1; //iguala flag a 1, simbolizando que a s
37     }
38 }
```

```
Arquivo Editar Sketch Ferramentas Ajuda sketch_sep22a5 sketch_sep22a5
37     flag=1;
38     break;
39 }
40 return senha;
41 }
42
43 void setup()
44 {
45     Serial.begin(9600);
46     SPI.begin();
47     mfrc522.PCD_Init();
48     Serial.println("Aproxime o seu cartão do leitor...");
49     Serial.println();
50     lcd.begin(16, 2);
51     lcd.init();
52     lcd.setCursor(0,0);
53     lcd.print("");
54     lcd.backlight();
55     pinMode(ledVerde, OUTPUT);
```

Arduino: Win7 (Mega 2560, Atmega2560 (Mega 2560) com USBtinyAVR)

```
53     lcd.backlight();
54     pinMode(ledVerde, OUTPUT);
55     pinMode(ledVermelho, OUTPUT);
56     pinMode(botaoG, INPUT);
57     pinMode(botao, INPUT);
58     senhaAtual="20be2419"; //define a senha padrão (o cartão original)
59
60 }
61
62 void loop()
63 {
64     digitalWrite(ledVerde, LOW); //ambos os leds ficam apagados
65     digitalWrite(ledVermelho, LOW);
66
67     valorGravar = digitalRead(botaoG);
68
69     if(valorGravar==HIGH){ //se o valor recebido pelo botão(gravar) for 1,
70         senhaAtual = modo_gravacao(senhaAtual); //        chama a função "modo_gravação"
71         delay(2000);}
```

Arduino: Win7 (Mega 2560, Atmega2560 (Mega 2560) com USBtinyAVR)

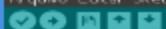
```
71     delay(2000);
72 }
73
74 valorBotao1= digitalRead(botao);
75
76 if(valorBotao1==HIGH){           //se o valor do botão atual for diferente do anterior
77     caixa=1;                      // ele iguala a variável caixa a 1
78 }
79
80 valorBotao2=valorBotao1;
81
82 if(caixa==1){                  //se o valor de caixa for diferente de zero
83     lcd.clear();
84     lcd.setCursor(0, 0);
85     lcd.print("CORRESPONDENCIA");
86 }
87 else{                          //se o valor de caixa for diferente de zero
88     lcd.setCursor(6, 0);
89     lcd.print("SEM");
```

```
145
89     lcd.print("SEM");
90     lcd.setCursor(0, 1);
91     lcd.println("CORRESPONDENCIA");          //mostra no display que não há correspondência
92 }
93 if ( ! mfrc522.PICC_IsNewCardPresent())    //verifica a presença de algum cartão lido
94 {
95     return;
96 }
97 if ( ! mfrc522.PICC_ReadCardSerial())
98 {
99     return;
100 }
101
102 String chave = "";                     //declara a string 'chave' para armazenar a UDI do
103 for (byte i = 0; i < mfrc522.uid.size; i++)
104 {
105     Serial.print(mfrc522.uid.uidByte[i], HEX);
106     chave.concat(String(mfrc522.uid.uidByte[i], HEX));    //aramzena cada termo na string chave
107 }
```

```
107 }
108
109 if(flag==0){
110     chave.toUpperCase();
111 }
112
113 if (chave.substring(0) == senhaAtual)           //se todos os termos da string chave forem igual
114 {                                               // ao cartão gravado
115     digitalWrite(ledVerde, HIGH);
116     digitalWrite(ledVermelho,LOW);
117     Serial.println();
118     lcd.clear();
119     lcd.setCursor(0,0);
120     lcd.print("Acesso permitido");
121     caixa=0;
122     delay(2000);
123     mensageminicial();                         //chama a função
124 }
125
```

```
116     digitalWrite(ledVermelho,LOW);
117     Serial.println();
118     lcd.clear();
119     lcd.setCursor(0,0);
120     lcd.print("Acesso permitido");
121     caixa=0;
122     delay(2000);
123     mensageminicial();                         //chama a função
124 }
125
126 if(chave == senhaAtual && flag==0){           //se a string chave for igual a senha do cartão
127     digitalWrite(ledVerde,HIGH);
128     digitalWrite(ledVermelho,LOW);
129     Serial.println();
130     lcd.clear();
131     lcd.setCursor(0,0);
132     lcd.print("Acesso permitido");
133     caixa=0;
134     delay(2000);                                //mostra no LCD a mensagem de "acesso permitido
135                                              // e abre a caixa de correio
136                                              //a variável fica = 0, mostrando como se não h
```

Arquivo Editar Sketch Ferramentas Ajuda



```

134   delay(2000);                                //chama a função
135   mensageminicial();
136 }
137
138 else if (chave.substring(0) != senhaAtual && chave.substring(0) !=0) //caso o código do cartão detect
139 {                                              //da senha atual e não seja igual
140   digitalWrite(ledVerde,LOW);
141   digitalWrite(ledVermelho,HIGH);               //acende o led vermelho
142   Serial.println();
143   lcd.clear();
144   lcd.setCursor(2,0);
145   lcd.print("Acesso Negado");
146   delay(3000);
147   mensageminicial();                          //mostra no LCD a mensagem de "aces
148 }                                              //e não abre a caixa de correio
149 }                                              //chama a função
150
151 void mensageminicial()                      //mensagem para aparecer no monitor
152 {

```

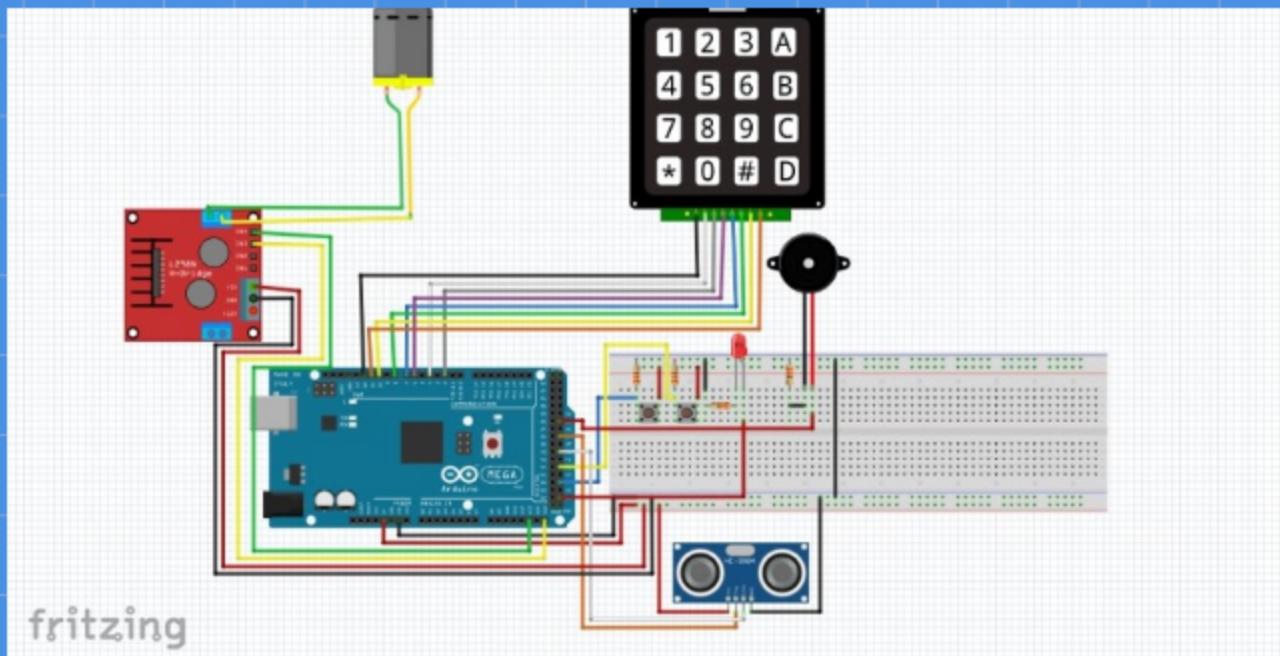
```

142
143
144
145
146
147
148
149
150
151 void mensageminicial()                      //mensagem para aparecer no monitor
152 {
153   lcd.clear();
154   Serial.println("Aproxime o seu cartão");
155   Serial.println();
156 }

```

# Área Externa

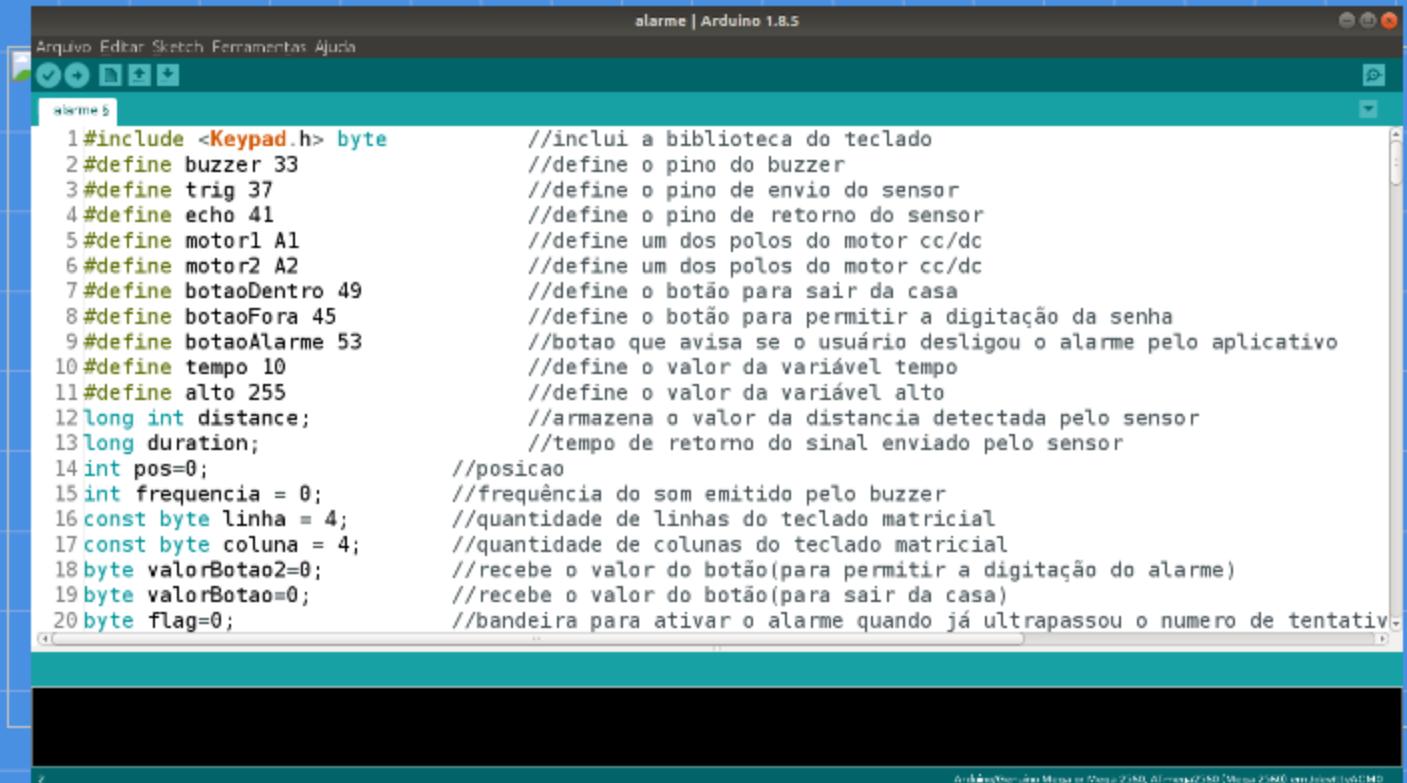
## 2. Portão/Alarme





# Área Externa

## 2. Portão/Alarme



The screenshot shows the Arduino IDE interface with a sketch titled "alarme". The code is written in C++ and defines various pins and variables for a keypad-based alarm system. The comments explain the purpose of each line of code.

```
alarme | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

alarme.ino

1 #include <Keypad.h> byte           //inclui a biblioteca do teclado
2 #define buzzer 33                  //define o pino do buzzer
3 #define trig 37                   //define o pino de envio do sensor
4 #define echo 41                   //define o pino de retorno do sensor
5 #define motor1L A1                //define um dos polos do motor cc/dc
6 #define motor2 A2                //define um dos polos do motor cc/dc
7 #define botaoDentro 49             //define o botão para sair da casa
8 #define botaoFora 45              //define o botão para permitir a digitação da senha
9 #define botaoAlarme 53             //botao que avisa se o usuário desligou o alarme pelo aplicativo
10 #define tempo 10                 //define o valor da variável tempo
11 #define alto 255                 //define o valor da variável alto
12 long int distance;            //armazena o valor da distancia detectada pelo sensor
13 long duration;               //tempo de retorno do sinal enviado pelo sensor
14 int pos=0;                   //posicao
15 int frequencia = 0;          //frequência do som emitido pelo buzzer
16 const byte linha = 4;         //quantidade de linhas do teclado matricial
17 const byte coluna = 4;        //quantidade de colunas do teclado matricial
18 byte valorBotao2=0;          //recebe o valor do botão(para permitir a digitação do alarme)
19 byte valorBotao=0;            //recebe o valor do botão(para sair da casa)
20 byte flag=0;                 //bandeira para ativar o alarme quando já ultrapassou o numero de tentativas
```

alarme | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

```
alarme S
20 byte flag=0; //bandeira para ativar o alarme quando já ultrapassou o numero de tentativas
21 byte vezes=0; //numero de teclas do código(para destravar/ativar o alarme)
22 byte ativa; //quando todas as teclas sao digitadas corretamente ele ativa o portao e desativa o alarme
23 byte desativado=0; //quando igual a zero, significa que o alarme e o sensor de presença estão desativados
24 byte valorSensor; //pega o valor do sensor de presença (para alertar se alguém invadiu a casa)
25 char chave; //armazena o valor da tecla digitada no teclado
26 char hexaKeys[linha][coluna] = {
27     {'1','2','3','A'},
28     {'4','5','6','B'},
29     {'7','8','9','C'},
30     {'*','0','#','D'}
31 };
32 byte linPins[linha] = {24, 4, 2, 6}; // pinos das linhas do teclado
33 byte colPins[coluna] = {7, 8, 10,5}; //pinos das colunas do teclado
34
35 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), linPins, colPins, linha, coluna); //initialize an instance of keypad
36
37 int abrirDentro(int valorBotao){ //abre a porta pelo lado de dentro(pressionando o botão)
38     analogWrite(motor1,alto); //motor gira para abrir, para por um período de 5 segundos
39     analogWrite(motor2,0); //e gira pra fechar, depois fica parado

```

alarme | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

```
alarme S
45     analogWrite(motor2,alto);
46     delay(7000);
47     return 0;
48 }
49
50 int alarme(int valorSensor){
51     /*feito pelo sensor de presença
52     ativa a sirene e ela só vai ser desligada quando
53     o dono sinalizar no aplicativo*/
54     int i,valorAlarme;
55     if(0==0){ //verdadeiro
56         valorAlarme=digitalRead(botaoAlarme);
57         delay(250);
58         while(valorAlarme == LOW){ //enquanto o dono nao desliga o alarme pelo aplicativo
59             valorAlarme=digitalRead(botaoAlarme);
60             analogWrite(motor1,0); //motor não se movimenta
61             analogWrite(motor2,0);
62             for (frequencia = 150; frequencia < 1800; frequencia += 1){ //dispara o buzzer, alterando a sua
63                 tone(buzzer, frequencia, tempo);
64                 delay(1);
65             }
66         }
67     }
68 }
```

Arquivo Editar Sketch Ferramentas Ajuda

alarme | Arduino 1.8.5

```
alarme5
68     delay(1);
69 }
70 }
71 }
72 } //reinicia o valor do botão
73 }
74
75 void setup(){
76   Serial.begin(9600);
77   pinMode(buzzer, OUTPUT);
78   pinMode(trig, OUTPUT);
79   pinMode(echo, INPUT);
80   pinMode(botaoDentro, INPUT_PULLUP);
81   pinMode(botaoFora, INPUT_PULLUP);
82   pinMode(motor1, OUTPUT);
83   pinMode(motor2, OUTPUT);
84   pinMode(botaoAlarme, INPUT);
85 }
86
87 void loop(){
```

Arquivo Editar Sketch Ferramentas Ajuda

alarme | Arduino 1.8.5

```
alarme5
87 void loop(){
88   analogWrite(motor1,0); //motor sempre parado
89   analogWrite(motor2,0);
90   if(desativado==0){ //se o alarme estiver ativado o sensor fica captando a presença de
91     digitalWrite(trig, LOW);
92     delayMicroseconds(2);
93     digitalWrite(trig, HIGH);
94     delayMicroseconds(10);
95     digitalWrite(trig, LOW);
96     duration = pulseIn(echo, HIGH);
97     distance = duration*0.034/2;
98     if(distance<=15){ //se algo for detectado em uma distância de 15cm, ele dispara o alarme
99       alarme(distance);
100    }
101  }
102
103  if(desativado!=0){ //se o alarme estiver desativado
104    valorBotao= digitalRead(botaoDentro); //lê o valor do botão dentro da casa
105    if(valorBotao==HIGH){ // se for 1 ele chama a função para abrir o portão
106      
```

Arquivo Editar Sketch Ferramentas Ajuda

alarme | Arduino 1.8.5

```
alarme.sketchbook
106     if(valorBotao==HIGH){           // se for 1 ele chama a função para abrir o portão
107         valorBotao= abrirDentro(valorBotao);
108     }
109 }
110
111 valorBotao2= digitalRead(botaoFora);    //lê o valor do botão fora da casa para colocar a senha
112 if(valorBotao2==1){                      //se o botão foi pressionado ele recebe o que for digitado
113
114 while(flag<4 && ativa==0){            //enquanto não ultrapassou o número de tentativas
115     vezes=0;                           //toda vez reinicia a variável, para contar as tentativas de
116     if(desativado--0){                 //se o alarme estiver ativado o sensor fica captando a prese
117         digitalWrite(trig, LOW);
118         delayMicroseconds(2);
119         digitalWrite(trig, HIGH);
120         delayMicroseconds(10);
121         digitalWrite(trig, LOW);
122         duration = pulseIn(echo, HIGH);
123         distance = duration*0.034/2;
124         if(distance<=15){             //se algo for detectado em uma distância de 15cm, ele dispara
125             alarme(distance);
126         }
127     }
128
129     while(vezes<5){                   //quantidade de digitações
130         chave = customKeypad.waitForKey(); //espera o usuário digitar no teclado
131
132         if(chave=='2' && vezes==0){   //se o usuário digitar '2' na primeira tentativa
133             ativa++;                  //incrementa a variável que representa um dígito correto
134
135         } else{
136             if(chave=='4' && vezes==1){ //se o usuário digitar '4' na segunda tentativa
137                 ativa++;                  //incrementa a variável que representa um dígito correto
138             }
139
140             else{
141                 if(chave=='6' && vezes==2){ //se o usuário digitar '6' na terceira tentativa
142                     ativa++;                  //incrementa a variável que representa um dígito correto
143                 }
144             }
145         }
146     }
147 }
```

Arquivo Editar Sketch Ferramentas Ajuda

alarme | Arduino 1.8.5

```
alarme.sketchbook
125     alarme(distance);
126 }
127 }
128
129 while(vezes<5){                   //quantidade de digitações
130     chave = customKeypad.waitForKey(); //espera o usuário digitar no teclado
131
132     if(chave=='2' && vezes==0){   //se o usuário digitar '2' na primeira tentativa
133         ativa++;                  //incrementa a variável que representa um dígito correto
134
135     } else{
136         if(chave=='4' && vezes==1){ //se o usuário digitar '4' na segunda tentativa
137             ativa++;                  //incrementa a variável que representa um dígito correto
138         }
139
140         else{
141             if(chave=='6' && vezes==2){ //se o usuário digitar '6' na terceira tentativa
142                 ativa++;                  //incrementa a variável que representa um dígito correto
143             }
144         }
145     }
146 }
```

alarme | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

alarme

```
144 }
145
146     else{
147         if(chave=='8' && vezes==3){           //se o usuário digitar '8' na quarta tentativa
148             ativa++;                         //incrementa a variável que representa um dígito correto
149         }
150
151     else{
152         if(chave=='D' && vezes==4){           //se o usuário digitar 'D' na quinta tentativa
153             ativa++;                         //incrementa a variável que representa um dígito correto
154         }
155     else{
156         if(chave=='A' && vezes==4){           //se o usuário digitar 'A' na quinta tentativa
157             ativa=ativa+2;                     //adiciona 2 à variável que representa um dígito correto
158         }
159     else{
160         ativa=0;                           //se o usuário digitar qualquer outra coisa fora das situ
161         }
162     }
163 }
```

alarme | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

alarme

```
166 }
167     vezes++;                         //incrementa a quantidade de tentativas
168 }
169 if(ativa==5){                      /*código digitado pra destravar o alarme
170                                     porta abre*/
171     digitalWrite(motor1,alto);        //motor gira para abrir, para por um período de 5 segundos
172     digitalWrite(motor2,0);          // e gira pra fechar
173     delay(3000);
174     digitalWrite(motor1,0);
175     digitalWrite(motor2,0);
176     delay(5000);
177     digitalWrite(motor1,0);
178     digitalWrite(motor2,alto);
179     delay(3000);

180     flag=0;                          //zera a quantidade de erros de digitações
181     desativado=1;                   //desativa o alarme
182     ativa=0;
183     break;
184 }
```

alarme | Arduino 1.8.5

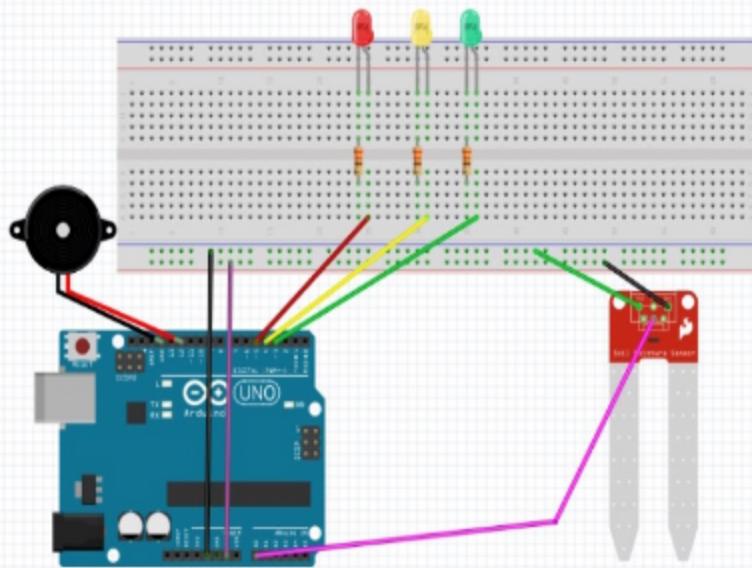
```
Arquivo Editar Sketch Ferramentas Ajuda
alarme.ino
180
181     flag=0;                      //zera a quantidade de erros de digitações
182     desativado=1;                  //desativa o alarme
183     ativa=0;
184     break;
185 }
186 else{
187     if(ativa==6){                //código digitado para ativar o alarme
188         flag=0;                  //zera a quantidade de erros de digitações
189         desativado=0;              //ativa o alarme
190     }
191     else{
192         digitalWrite(buzzer,HIGH); //buzzer toca em toda tentativa falha
193         delay(1000);
194         digitalWrite(buzzer,LOW);
195         flag++;                   //incrementa a variável que armazena a quantidade de erros de
196     }
197 }
```

alarme | Arduino 1.8.5

```
Arquivo Editar Sketch Ferramentas Ajuda
alarme.ino
190
191 else{
192     digitalWrite(buzzer,HIGH);      //buzzer toca em toda tentativa falha
193     delay(1000);
194     digitalWrite(buzzer,LOW);
195     flag++;                      //incrementa a variável que armazena a quantidade de erros de
196 }
197
198 if(flag==2){                    //só pode digitar novamente após 10 segundos
199     Serial.println("TENTE NOVAMENTE EM 10 SEGUNDOS");
200     delay(10000);
201 }
202 }
203 }
204
205 if(flag==4){                  //quando já ultrapassou o limite de erros
206     flag=alarme(flag);          //espera um comando pra desligar o alarme e iguala flag a 0 para uma próxima te
207 }
208
209 }
```

# Área Externa

## 3. Sistema de Irrigação





# Área Externa

## 3. Sistema de Irrigação

```
SensoreUmidade.h

#define pino_sinal_analogico A0
#define pino_led_vermelho 5 //led vermelho no pino 5
#define pino_led_amarelo 4 //led amarelo no pino 4
#define pino_led_verde 3 //led verde no pino 3
#define buzzer 12 //buzzer no pino 12
int valor_analogico; //variavel que recebe o valor analógico

void setup() {
    // inicio da comunicação serial
    Serial.begin(9600);
    pinMode(pino_sinal_analogico, INPUT); //define o pino conectado ao sinal analógico com entrada
    pinMode(pino_led_vermelho, OUTPUT); //define o pino conectado ao led vermelho com saída
    pinMode(pino_led_amarelo, OUTPUT); //define o pino conectado ao led amarelo com saída
    pinMode(pino_led_verde, OUTPUT); //define o pino conectado ao led verde com saída
    pinMode(buzzer, OUTPUT); //define o pino conectado ao buzzer com saída
}

void loop() {
```

# Área Externa

## 3. Sistema de Irrigação

```
SensordeUmidade
```

```
void loop() {  
  
    // Leitura do sensor no pino A0 e armazenamento em SensorValue  
  
    valor_analogico = analogRead(pino_sinal_analogico);  
  
    //Mostra o valor da porta analogica no serial monitor  
    Serial.print("Valor da porta analogica: ");  
    Serial.print(valor_analogico);  
  
    //Solo umido, acende o led verde  
    if (valor_analogico > 0 && valor_analogico < 500)  
    {  
        apagaleds();  
        digitalWrite(pino_led_verde, HIGH);  
    }  
  
    //solo com umidade moderada, acende led amarelo  
    if (valor_analogico > 500 && valor_analogico < 900)  
    {  
        apagaleds();  
    }  
}
```

# Área Externa

## 3. Sistema de Irrigação

Sensor de Umidade

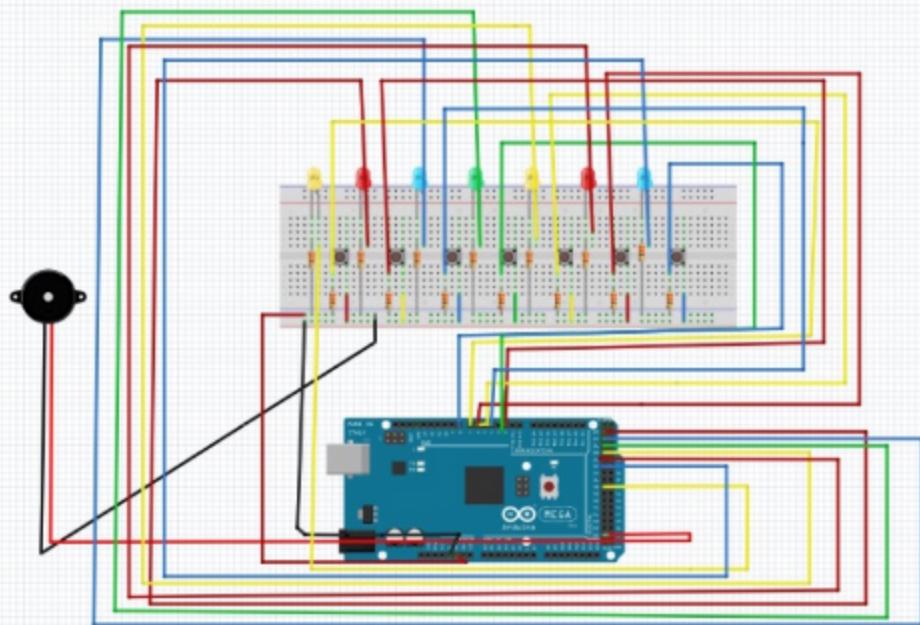
```
|
  apagaleds();
  digitalWrite(pino_led_amarelo, HIGH);
}

// Solo seco, acende led vermelho
if (valor_analogico > 900 && valor_analogico < 1024)
{
  apagaleds();
  digitalWrite(pino_led_vermelho, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(1500);
  digitalWrite(buzzer, LOW);
}
delay(100);
}

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

# Cozinha

## 1.Caixa de Remédios





# Cozinha

## 1.Caixa de Remédios

```
Ressana

#define pinoLedAmarelo1 30 // pino digital em que o led amarelo 1 está conectado
#define pinoLedVermelho1 24 // pino digital em que o led vermelho 1 está conectado
#define pinoLedAzul1 26 // pino digital em que o led azul 1 está conectado
#define pinoLedVerde1 28 // pino digital em que o led verde 1 está conectado
#define pinoLedAmarelo2 30 // pino digital em que o led amarelo 2 está conectado
#define pinoLedVermelho2 32 // pino digital em que o led vermelho 2 está conectado
#define pinoLedAzul2 34 // pino digital em que o led azul 2 está conectado
#define buzzer 53 // DEFINE O PINO DIGITAL EM QUE O BUZZER ESTÁ LIGADO
#define botaoAmarelo1 7 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoVermelho1 6 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoAzul1 5 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoVerde1 4 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoAmarelo2 3 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoVermelho2 2 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO
#define botaoAzul2 8 // DEFINE O PINO DIGITAL EM QUE O BOTÃO ESTÁ LIGADO

int statusLedAmarelo1 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED AMARELO1 É DESLIGADO
int statusLedVermelho1 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED VERMELHO1 É DESLIGADO
int statusLedAzul1 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED VERDE1 É DESLIGADO
int statusLedVerde1 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED AZUL1 É DESLIGADO
int statusLedAmarelo2 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED AMARELO2 É DESLIGADO
int statusLedVermelho2 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED VERMELHO2 É DESLIGADO
int statusLedAzul2 = LOW; //DEFINE QUE O ESTADO INICIAL DO LED AZUL2 É DESLIGADO
```

# Cozinha

## 1. Caixa de Remédios

```
Resenha

int statusBotaoVermelho1 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)
int statusBotaoAzul1 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)
int statusBotaoVerde1 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)
int statusBotaoAmarelo2 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)
int statusBotaoVermelho2 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)
int statusBotaoAzul2 = 0; //DEFINE QUE O ESTADO INICIAL DO BOTAO( HIGH = 1, LOW = 0)

long MillisAmarelo1Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED AMARELO1
long MillisVermelho1Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED VERMELHO1
long MillisAzul1Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED AZUL1
long MillisVerde1Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED VERDE1
long MillisAmarelo2Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED AMARELO2
long MillisVermelho2Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED VERMELHO2
long MillisAzul2Anterior = 0; //VARIÁVEL QUE CONTROLA O TEMPO DO LED AZUL2

long amareloLedInterval = 500; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AMARELO1 (86400000ms = 24 HORAS)
long vermelho1LedInterval = 2000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED VERMELHO1 (43200000ms = 12 HORAS)
long azul1LedInterval = 4000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AZUL1 (21600000ms = 6 HORAS)
long verde1LedInterval = 6000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED VERDE1 (12960000ms = 3 HORAS)
long amarelo2LedInterval = 8000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AMARELO2 (60000ms = 1 MINUTO)
long vermelho2LedInterval = 10000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED VERMELHO2 (15000ms = 30 SEGUNDOS)
```

# Cozinha

## 1.Caixa de Remédios

```
Resenha

long vermelhoLedInterval = 2000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED VERMELHO1 (43200000ms = 12 HORAS)
long azulLedInterval = 4000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AZUL (28800000ms = 8 HORAS)
long verdeLedInterval = 6000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AZUL (21600000ms = 6 HORAS)
long amareloLedInterval = 8000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AMARELO2 (60000ms = 1 MINUTO)
long vermelho2LedInterval = 10000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED VERMELHO2 (15000ms = 30 SEGUNDOS)
long azul2LedInterval = 1000; //VARIÁVEL QUE REGISTRA O INTERVALO DO LED AZUL2 (15000ms = 15 SEGUNDOS)

void desligaBuzzer()
{
    digitalWrite(buzzer, LOW);
}

void setup(){
    Serial.begin(9600);
    Serial.println("ENTROU 1");
    pinMode(pinLedAmarelo1, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedVermelho1, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedAzul1, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedVerde1, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedAmarelo2, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedVermelho2, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinLedAzul2, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(botaoAmarelo1, INPUT); //DEFINE O PINO COMO ENTRADA
    pinMode(botaoVermelho1, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
```

# Cozinha

## 1.Caixa de Remédios

```
Ressenha
pinMode(pinoLedAmarelo2, OUTPUT); //DEFINE O PINO COMO SAÍDA
pinMode(pinoLedVermelho2, OUTPUT); //DEFINE O PINO COMO SAÍDA
pinMode(pinoLedAzul2, OUTPUT); //DEFINE O PINO COMO SAÍDA
pinMode(botacAmarelo1, INPUT); //DEFINE O PINO COMO ENTRADA
pinMode(botacVermelh1, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(botacAzul1, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(botacVerde1, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(botacAmarelo2, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(botacVermelh2, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(botacAzul2, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(buzzer, OUTPUT); //DEFINE O PINO COMO SAÍDA
}
void loop()
{
    Serial.println("ENTROU 2");
    unsigned long MillisAmarelo1Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVermelh1Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAzul1Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVerde1Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAmarelo2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVermelh2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAzul2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    Serial.println("ENTROU 3");
}
```

# Cozinha

## 1.Caixa de Remédios

```
Reserva 6
pinMode(botaoAzul2, INPUT_PULLUP); //DEFINE O PINO COMO ENTRADA
pinMode(buzzer, OUTPUT); //DEFINE O PINO COMO SAÍDA
}
void loop()
{
    Serial.println("ENTROU 2");
    unsigned long MillisAmareloAtual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVermelhoAtual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAzulAtual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVerdeAtual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAmarelo2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisVermelho2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    unsigned long MillisAzul2Atual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
    Serial.println("ENTROU VOID LOOP");
    Serial.println(MillisAmareloAtual);
    delay(1000);

    if (MillisAmareloAtual - MillisAmareloAnterior >= amareloLedInterval) { //SE O TEMPO ATUAL MENOS O TEMPO ANTERIOR FOR MENOR QUE O INTERVALO, SIGNIFICA QUE JÁ PASSOU
        MillisAmareloAnterior = MillisAmareloAtual ; //TEMPO ANTERIOR GUARDA O TEMPO ATUAL
        Serial.println("ENTROU AMARELO");
        delay(1000);
    }
}
```

# Cozinha

## 1.Caixa de Remédios

```
Reserva§
Serial.println("ENTROU AMARELO1");
delay(1000);
if (statusLedAmarelo1 == LOW) | //VERIFICA O ESTADO ATUAL DO LED E SE ESTIVER EM DESLIGADO MUDA PARA LIGADO
    Serial.println("ENTROU §");
    while(statusBotaoAmarelo1 == 0)
    {
        statusBotaoAmarelo1 = digitalRead(botaoAmarelo1);
        digitalWrite(pinoLedAmarelo1, HIGH); //VARIÁVEL RECEBE ESTADO HIGH(LIGA O LED)
        digitalWrite(buzzer, HIGH);
    }
    desligaBuzzer();
} else { //SE NÃO, FAZ
    statusLedAmarelo1 = LOW; //VARIÁVEL RECEBE ESTADO LOW(DESLIGA O LED)
}
digitalWrite(pinoLedAmarelo1, statusLedAmarelo1); //ESCREVE NO PINO DIGITAL O ESTADO ATUAL (LIGADO OU DESLIGADO)
}
//FIM DO CONTROLE DO LED AMARELO1

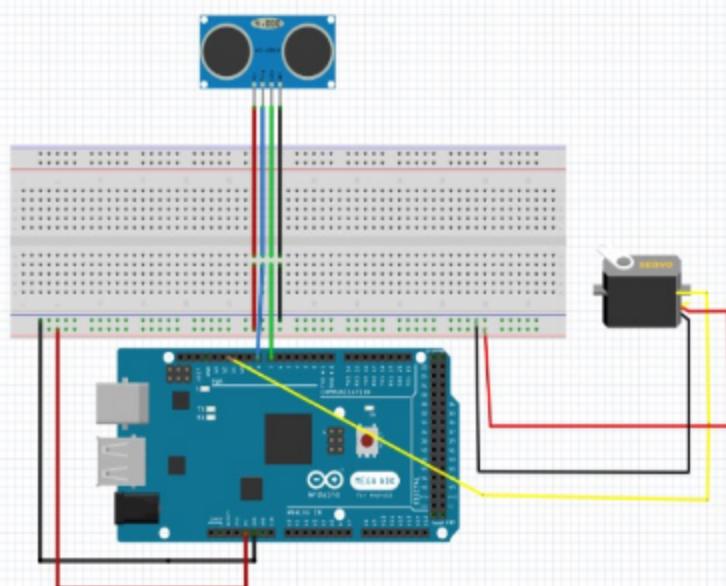
//-----
//INICIO DO CONTROLE DO LED VERMELHO1 unsigned long MillisVermelhoAtual = millis(); //VARIÁVEL RECEBE O TEMPO ATUAL EM MILISSEGUNDOS
if (MillisVermelhoAtual - MillisVermelhoAnterior >= vermelhoLedIntervalo) | //SE O TEMPO ATUAL MENOS O TEMPO ANTERIOR FOR MENOR QUE O INTERVALO, FAZ
{
```



# Cozinha

---

## 2. Lixeira Automática



# Cozinha

---

## 2. Lixeira Automática

```
LIXERA
#include <Servo.h>
const int trigPin = 8; //pino de saida ultrassonico
const int echoPin = 7; //pino de entrada ultrassonico

Servo servo;

long duracao;
int distancia;//distancia em centimetros
int angulo= 0; //o servo inicia com posicao zero(no sentido de angulacao)

void setup() {
  pinMode(trigPin, OUTPUT); // define a saida
  pinMode(echoPin, INPUT); // define a entrada
  Serial.begin(9600);
  servo.attach(9); // pino do servo motor

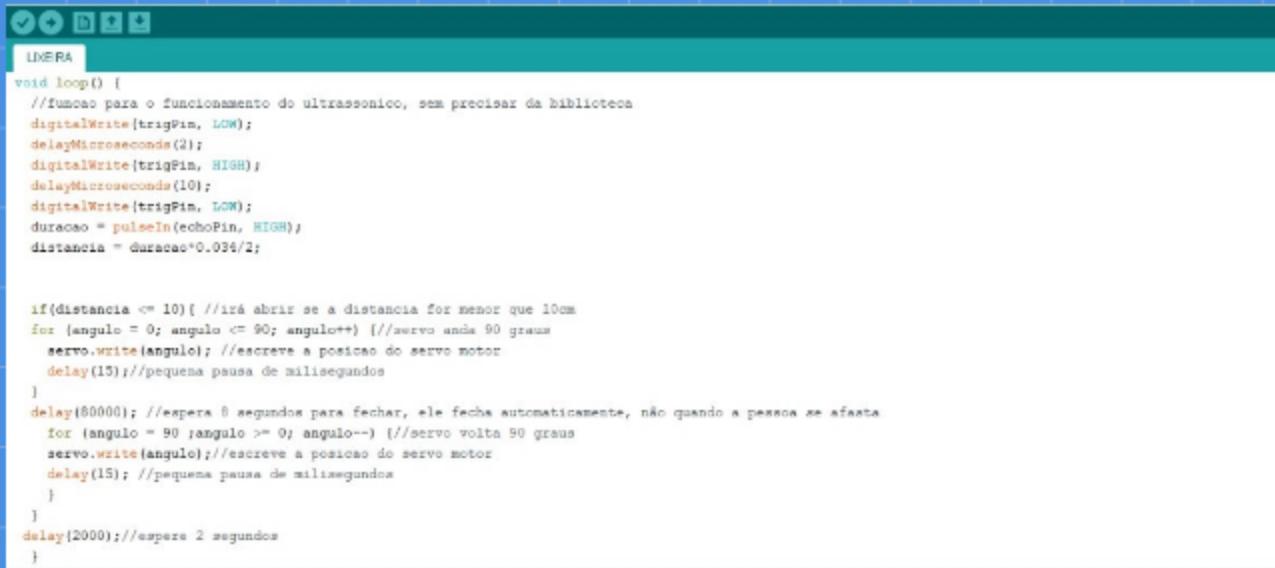
}

void loop() {
  //funcao para o funcionamento do ultrassonico, sem precisar da biblioteca
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

# Cozinha

---

## 2. Lixeira Automática



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Shows the title "LIXEIRA" and the standard Arduino IDE icons.
- Code Area:** Displays the C++ code for the "LIXEIRA" project. The code includes comments explaining the ultrasonic sensor setup and the servo motor's range of motion (0 to 90 degrees). It also includes a delay of 2 seconds at the end of the loop.

```
void loop() {
    //funcao para o funcionamento do ultrassonico, sem precisar da biblioteca
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duracao = pulseIn(echoPin, HIGH);
    distancia = duracao*0.034/2;

    if(distancia <= 10){ //irá abrir se a distancia for menor que 10cm
        for (angulo = 0; angulo <= 90; angulo++) //servo anda 90 graus
            servo.write(angulo); //escreve a posicao do servo motor
            delay(15); //pequena pausa de milisegundos
    }
    delay(80000); //espera 8 segundos para fechar, ele fecha automaticamente, n&gt;o quando a pessoa se afasta
    for (angulo = 90; angulo >= 0; angulo--) //servo volta 90 graus
        servo.write(angulo); //escreve a posicao do servo motor
        delay(15); //pequena pausa de milisegundos
    }
}
delay(2000); //espera 2 segundos
}
```

# Banheiro

---

## 1. Vaso Automático

. O que é?

O dispositivo funciona a partir de um sensor ultrassônico acoplado a base do protótipo, recebendo o sinal da distância da pessoa em relação a ele.

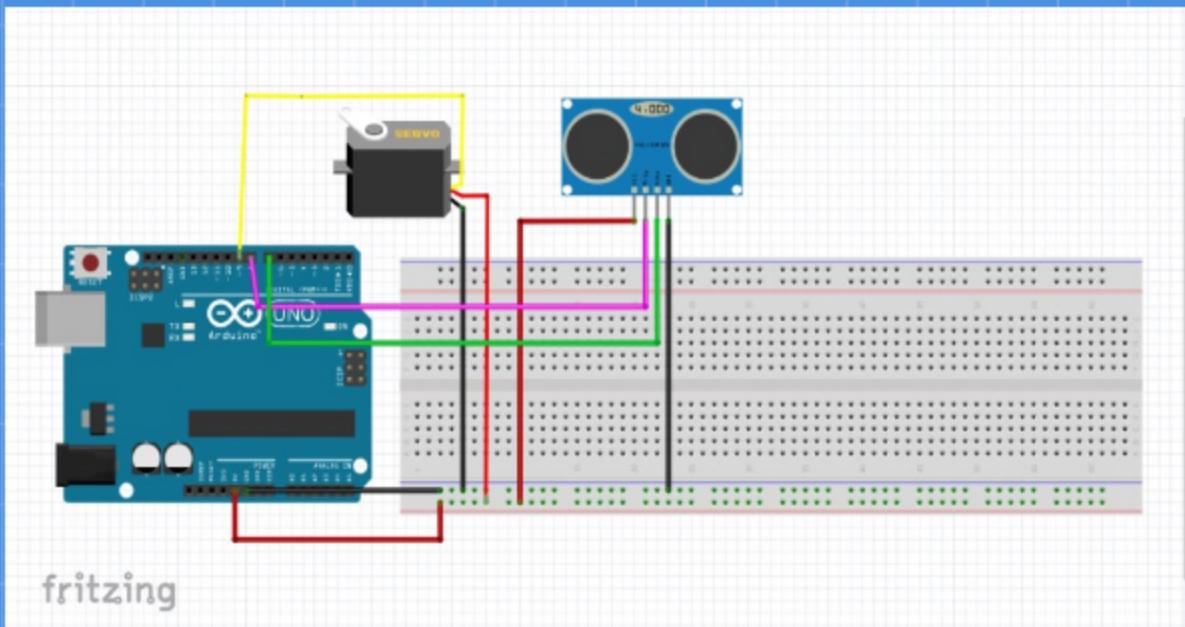
A lógica consiste em determinar se o usuário está próximo suficiente para utilizar o vaso, caso contrário a tampa fechará.

. Vantagens

- Acessibilidade para pessoas com limitações físicas
  - Higiene
-

# Banheiro

## 1. Vaso Automático



# Banheiro

---

## 1. Vaso Automático

```
||#include <Servo.h>
const int trigPin = 8; //pino de saída ultrassônico no pino 8
const int echoPin = 7; //pino de entrada ultrassônico no pino 7
int flag = 0;

Servo servo;

long duracao;// duração da movimentação
int distancia;//distância do objeto
int pos=0; //o servo inicia com posição zero

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  serial.begin(9600);
  servo.attach(9); // pino do servo motor no 9
}

void loop() {
  //Função para o funcionamento do ultrassônico
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duracao = pulseIn(echoPin, HIGH);
  if (duracao > 1000) {
    pos = 180;
  } else {
    pos = 0;
  }
  servo.write(pos);
}
```

# Banheiro

---

## 1. Vaso Automático

```
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);  
duracao = pulseIn(echoPin, HIGH);  
distancia = duracao*0.034/2;  
  
//abertura  
if(distancia <= 10 && flag==0){ //se a distancia for menor que 10cm e entrar com flag 0 (fechamento)  
//entra no for para abrir  
for (pos = 0; pos <= 90; pos += 1) //servo anda 90 graus  
servo.write(pos); //escriva a posicao do servo motor  
delay(1); //pequena pausa de milisegundos  
}  
flag=1; //muda flag para 1 quando abrir  
}  
//fechamento  
else if(flag==1 && distancia > 10){ //se a distancia for maior que 10cm e houver flag 1 (abertura)  
flag=0; //muda flag para 0 quando fechar  
for (pos = 90; pos >= 0; pos -= 1) //servo volta 90 graus  
servo.write(pos); //escriva a posicao do servo motor  
delay(1); //pequena pausa de milisegundos  
}  
delay(2000); //espere 2 segundos  
}
```

# Luzes

---

.O que é?

O sistema de iluminação da casa inteligente consiste em uma plataforma desenvolvida em um aplicativo celular, no qual permite interagir com as luzes de todos os cômodos apenas em um toque. Além da fácil utilização, o sistema também envia sinais sonoros e vibratórios de identificação, voltadas para as pessoas que possuem deficiência sonoro e visual.

.Vantagens

- Facilidade
  - Acessibilidade
-

# Luzes

---

```
#define BUZZER 13 //buzzer no pino 13
#define SALA 12 //led da sala no pino 12
#define BANHEIRO 11 //led do banheiro no pino 11
#define COZINHA 10 //led da cozinha no pino 10
#define ALARME 6 // pino do alarme no pino 6

void setup() {
  pinMode(BUZZER ,OUTPUT);
  pinMode(SALA,OUTPUT);
  pinMode(BANHEIRO,OUTPUT);
  pinMode(COZINHA,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  char leitura;
  leitura = Serial.read(); // leitura do char enviado pelo aplicativo

  if(leitura == 'A')digitalWrite(SALA,HIGH); //ativar led da sala
  if(leitura == 'a')digitalWrite(SALA,LOW); //desativar led da sala

  if(leitura == 'B')digitalWrite(BANHEIRO,HIGH); //ativar led do banheiro
  if(leitura == 'b')digitalWrite(BANHEIRO,LOW); //desativar led do banheiro

  if(leitura == 'C')digitalWrite(COZINHA,HIGH); //ativar led da cozinha
  if(leitura == 'c')digitalWrite(COZINHA,LOW); //desativar led da cozinha
```

# Luzes

---

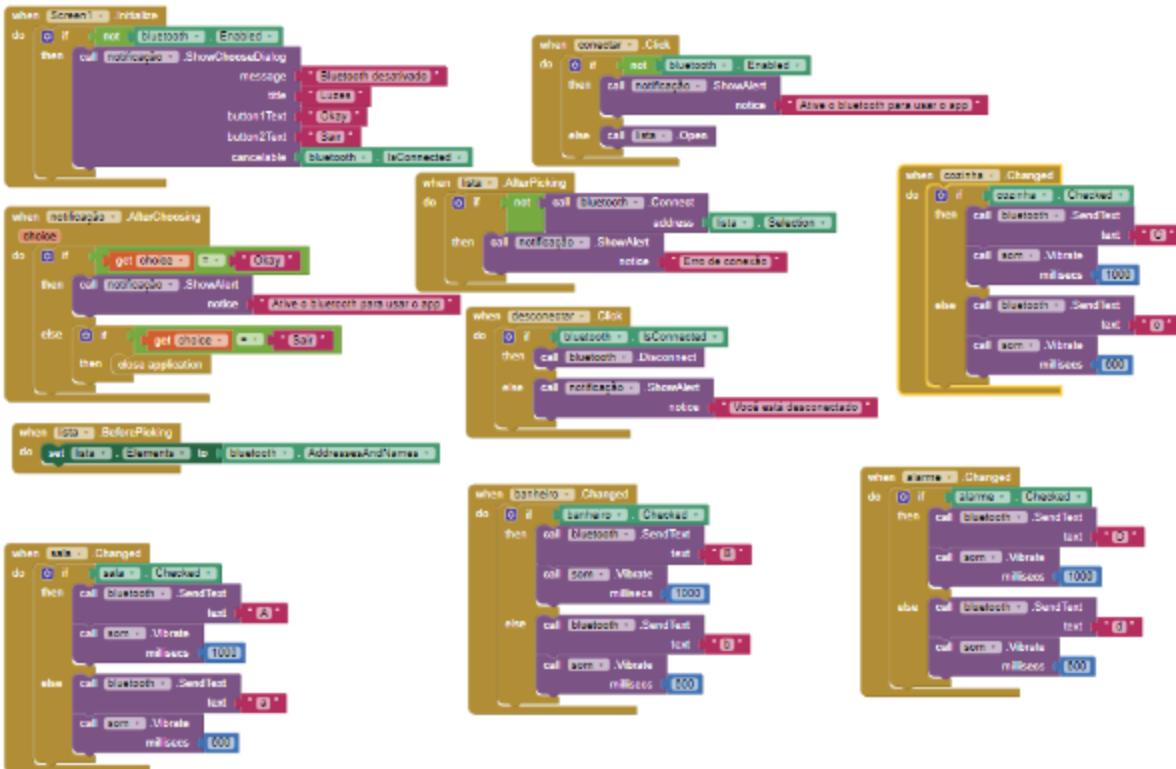
```
if(leitura == '0')digitalWrite[ALARME ,HIGH];//ativar alarme
if(leitura == '1')digitalWrite[ALARME ,LOW];//desativar alarme

if(leitura == 'A'){
  tone(BUZZER,262,200); //DO
}
if(leitura == 'a'){
  tone(BUZZER,262,200); //DO
  delay(500);
  tone(BUZZER,262,200); //DO
}

if(leitura == 'B'){
  tone(BUZZER,294,300); //RE
}
if(leitura == 'b'){
  tone(BUZZER,294,300); //RE
  delay(500);
  tone(BUZZER,294,300); //RE
}

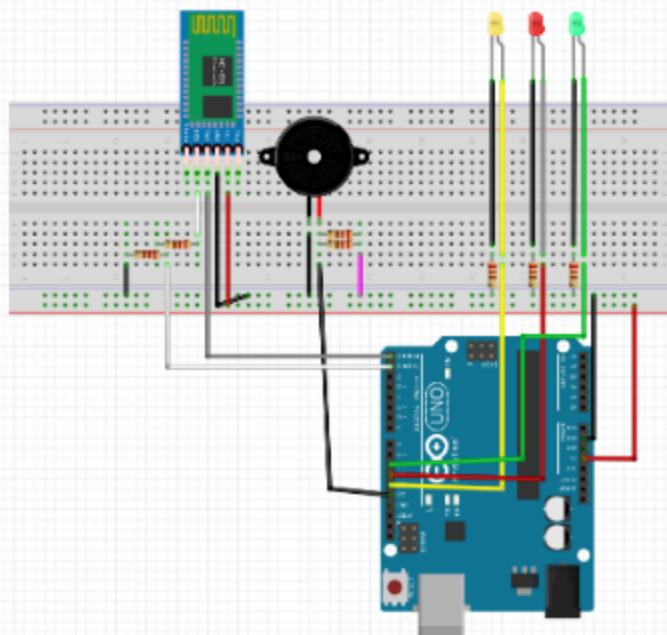
if(leitura == 'C'){
  tone(BUZZER,330,300); //MI
}
if(leitura == 'c'){
  tone(BUZZER,330,300); //MI
  delay(500);
  tone(BUZZER,330,300); //MI
```

# Luzes



# Luzes

---



fritzing

