

AUTOMAÇÃO RESIDENCIAL



ALUNOS RESPONSÁVEIS

EMILAINÉ DO PRADO CORREIA

JOÃO VITOR DE SOUZA RIBEIRO

MATHEUS QUINAUD BEDESCHI

MURYLO HENRIQUE ALVES FAGUNDES

VINICIUS FERREIRA COUTO

2º ANO | CCOMP

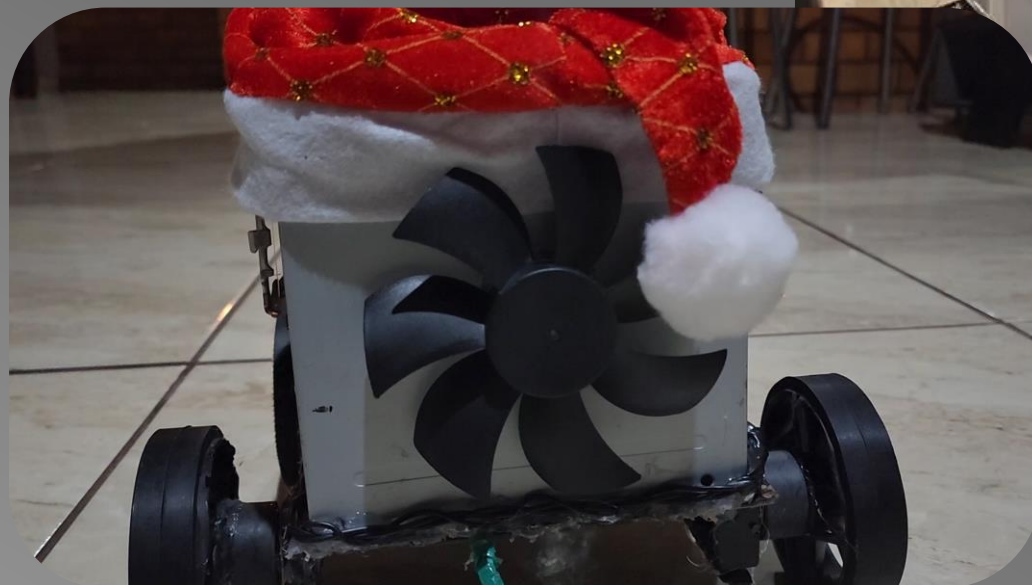


PRÁTICAS EXTENSIONISTAS

→ Extensão: ato ou efeito de estender-se

OBJETIVOS

- ❖ aliar a teoria da sala de aula à prática na comunidade;
- ❖ elaborar ações extensionistas;
- ❖ estreitar a distância entre discentes e sociedade;
- ❖ participar de mostras itinerantes;
- ❖ apresentar áreas temáticas da computação.

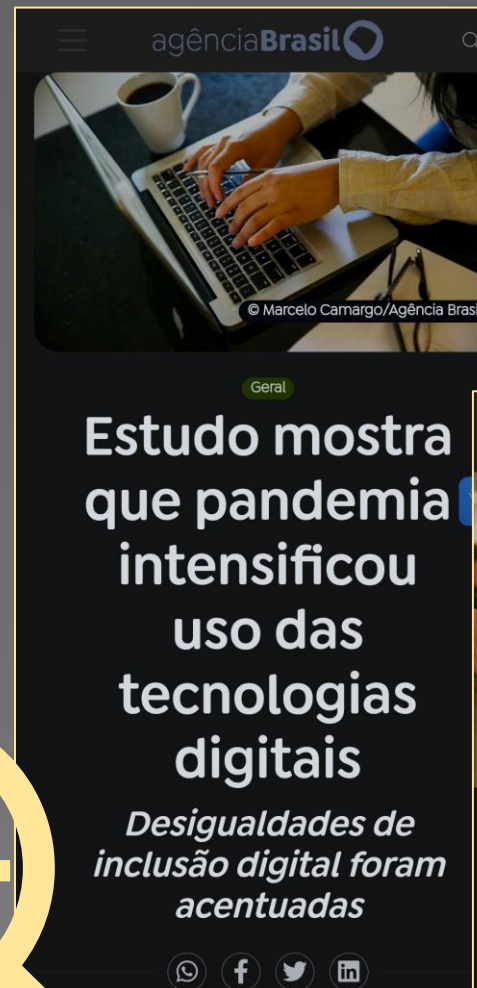
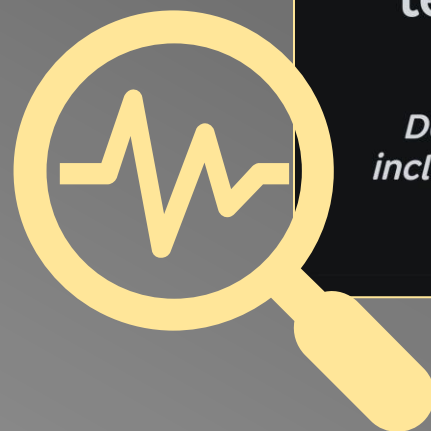




ESCOLHA DO TEMA

PRINCIPAIS PONTOS

- ❖ a pandemia e o aumento do uso de tecnologia digital;
- ❖ facilitar tarefas cotidianas;
- ❖ aproximar a tecnologia da comunidade;
- ❖ projeções futuras da área;
- ❖ apresentação visual;
- ❖ qualidade de vida.





ESCOLHA DO TEMA

12,3% de aumento anual
até 2028

291 milhões de dólares
em 2021

Automação Residencial

Mercado em expansão

70% da população BR
vê um problema no preço

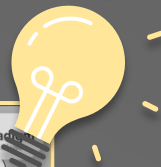
Itens de segurança com aumento
de 65% em 2020.



DESENVOLVIMENTO DO PROJETO

Made with
VisualParadigm
For non-commercial use

← 1º Andar | Térreo

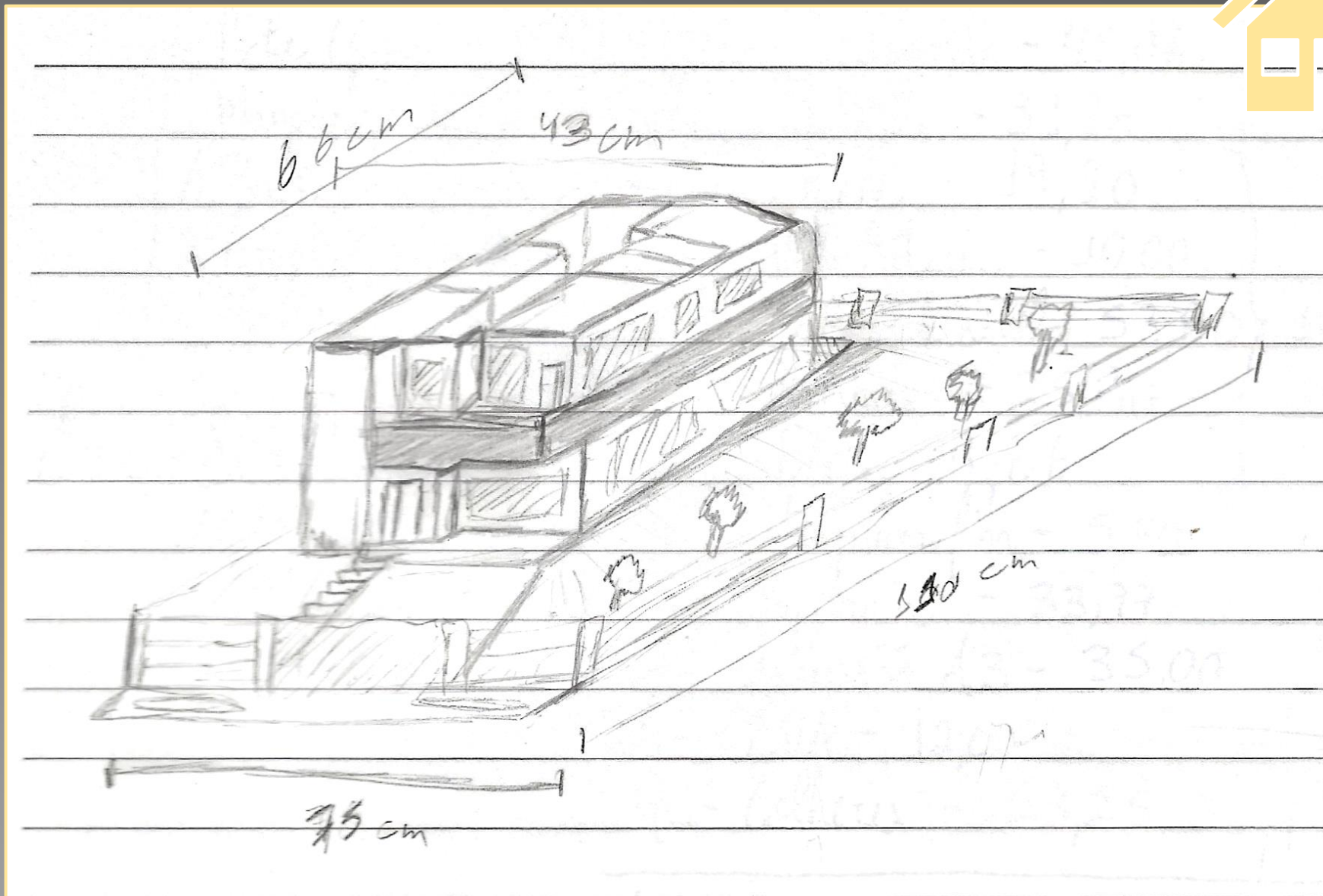


Made with
VisualParadigm
For non-commercial use

Planta Baixa

OBJETIVO

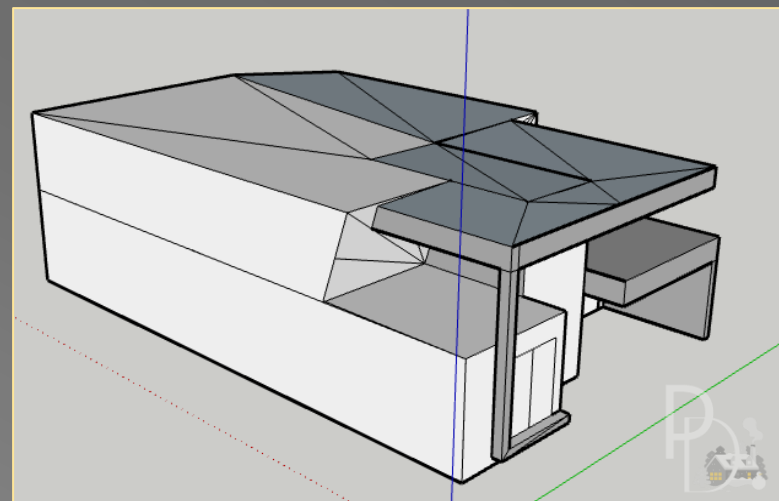
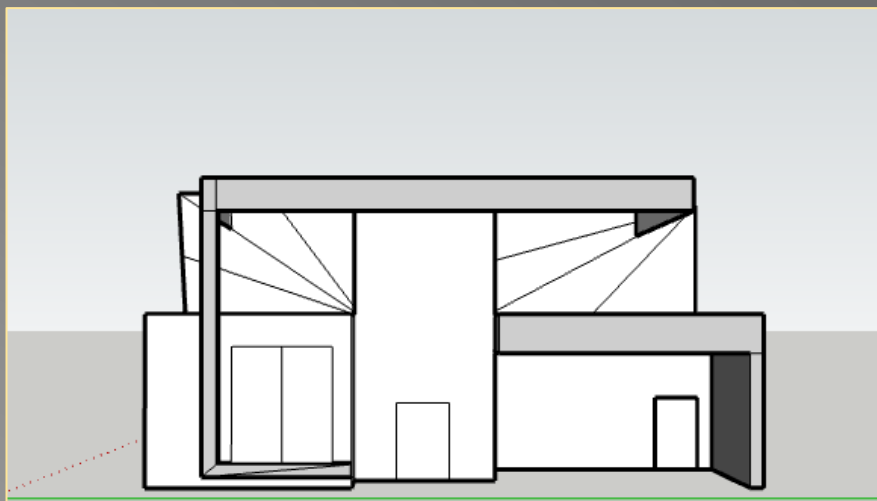
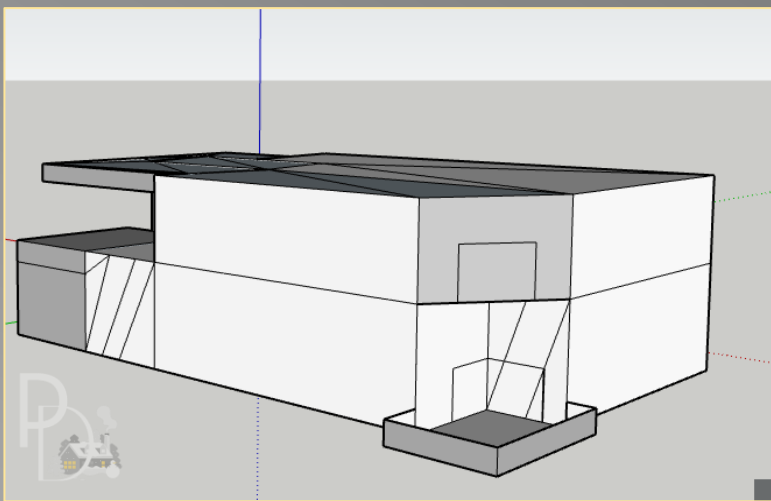
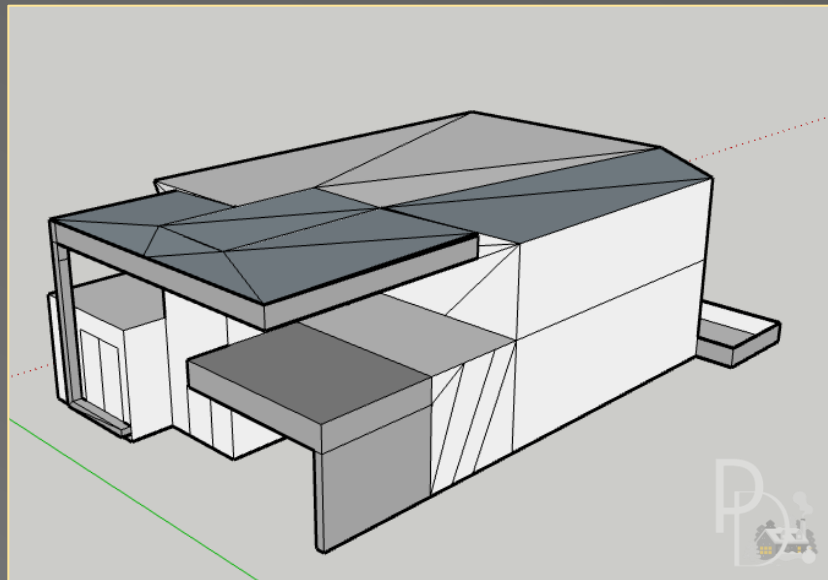
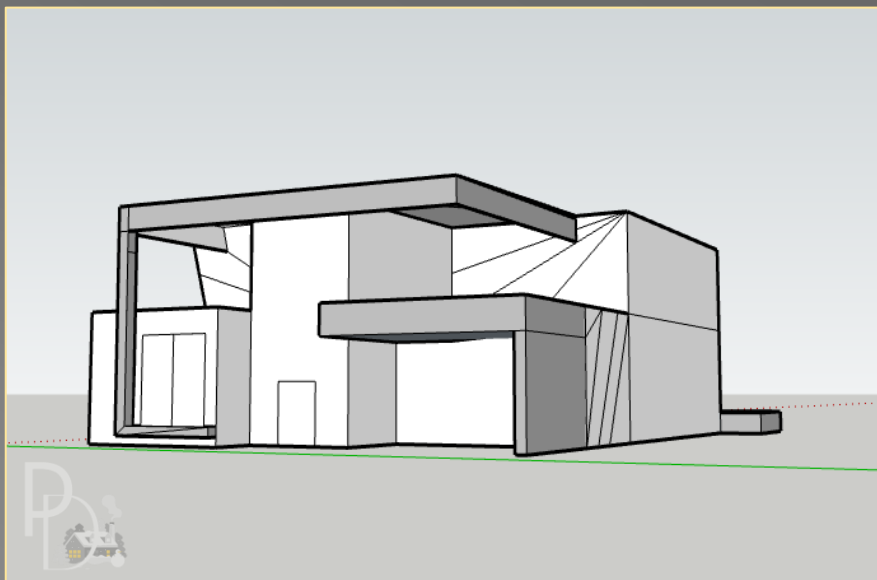
- ❖ base para a automatização;
- ❖ estabelecer o modelo a ser seguido;
- ❖ estimar os ambientes da casa;
- ❖ servir de fundamento para a construção.





ESTRUTURA

Visão externa simulada





PROJETOS DIDÁTICOS



CONSTRUÇÃO





CONSTRUÇÃO





FUNCIONALIDADE

Projeto de Automação

Leds internos e externos representando:

- pendentes e luminárias;
- spots;
- abajures;
- iluminação do jardim.



Acionamento de alarme por presença.



Portão de entrada/saída.

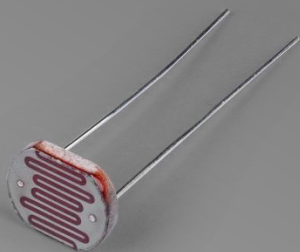




FUNCIONALIDADE



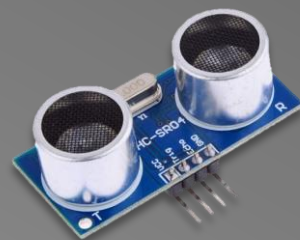
- ❖ acendimento por meio de um sensor de luminosidade;



LDR



- ❖ acionamento por meio de um sensor de distância ultrassônico;



HC-SR04



- ❖ abertura/fechamento por meio de um motor de passo.



Motor De Passo +
Driver Uln2003



FUNCIONALIDADE - ARDUINO

```
1  #include <Stepper.h>
2
3  // define identificadores para as portas do microcontrolador
4  #define photoresistor A0
5  #define buzzer 2
6  #define rele 3
7  #define trigger 4
8  #define echo 5
9  #define button 7
10 #define smInput1 8
11 #define smInput2 9
12 #define smInput3 10
13 #define smInput4 11
14
15 // define uma constante para quantidade de passos de 360°
16 const int steps = 64;
17 // cria um objeto do tipo Stepper passando como parâmetros a quantidade de passos do motor e suas portas de ligação
18 Stepper stepMotor (steps, smInput1, smInput3, smInput2, smInput4);
19 // cria uma variável para o sensor LDR (Light Dependent Resistor) e para o sensor ultrassônico (HC-SR04)
20 int luminosity = 0;
21 double distance;
```



FUNCIONALIDADE - ARDUINO

```
void setup () {
24  // habilita portas do microcontrolador associadas a cada identificador, em modo de entrada ou saída
25  pinMode (photoresistor, INPUT);
26  pinMode (buzzer, OUTPUT);
27  pinMode (rele, OUTPUT);
28  pinMode (button, INPUT);
29  // utiliza o método setSpeed () da classe Stepper para determinar a velocidade de rotação do motor em RPM
30  stepMotor.setSpeed (350);
31  // inicializa o monitor serial a taxa de transmissão de 9600 bps
32  Serial.begin (9600);
33 }
34
35 void loop () {
36  // define aspectos operacionais para o motor de passos
37  static int previousButtonState = 0;
38  int pushButton = digitalRead (button);
39  if (pushButton == 1 && previousButtonState == 0) {
40  |   rotateClockwise ();
41  | }
42  else if (pushButton == 0 && previousButtonState == 1) {
43  |   rotateCounterClockwise ();
44  | }
45  previousButtonState = pushButton;
```



FUNCIONALIDADE - ARDUINO

```
48 // define aspectos operacionais para o sensor ultrassônico
49 distance = readUltrasonicSensor (); // verifica a distância do objeto, aciona ou desativa o buzzer
50 luminosity = analogRead (photoresistor);
51 Serial.print ("Distância: ");
52 Serial.print (distance);
53 Serial.println (" cm");
54 Serial.println ("-----");
55 delay (0);
56 if (distance < 17) {
57     digitalWrite (buzzer, HIGH);
58     tone (buzzer, 261); // aciona o buzzer na frequência Dó (261,63 Hz)
59     delay (200); // permanece ativo na frequência Dó por 0.2 segundos
60     noTone (buzzer); // desativa a frequência Dó
61     tone (buzzer, 293); // aciona o buzzer na frequência Ré (293,66 Hz)
62     delay (200); // permanece ativo na frequência Ré por 0.2 segundos
63     noTone (buzzer); // desativa a frequência Ré
64 }
65 else {
66     digitalWrite (buzzer, LOW);
67 }
68 // define aspectos operacionais para o sensor LDR
69 Serial.print ("Luminosidade: ");
70 Serial.print (luminosity);
71 Serial.println (" lux");
72 Serial.println ("-----");
73 if (luminosity < 300) { // verifica o valor de entrada de luminosidade do ambiente, aciona ou desativa o relé conectado as luzes
74     digitalWrite (rele, LOW);
75 }
76 else {
77     digitalWrite (rele, HIGH);
78 }
79 delay (500);
}
```



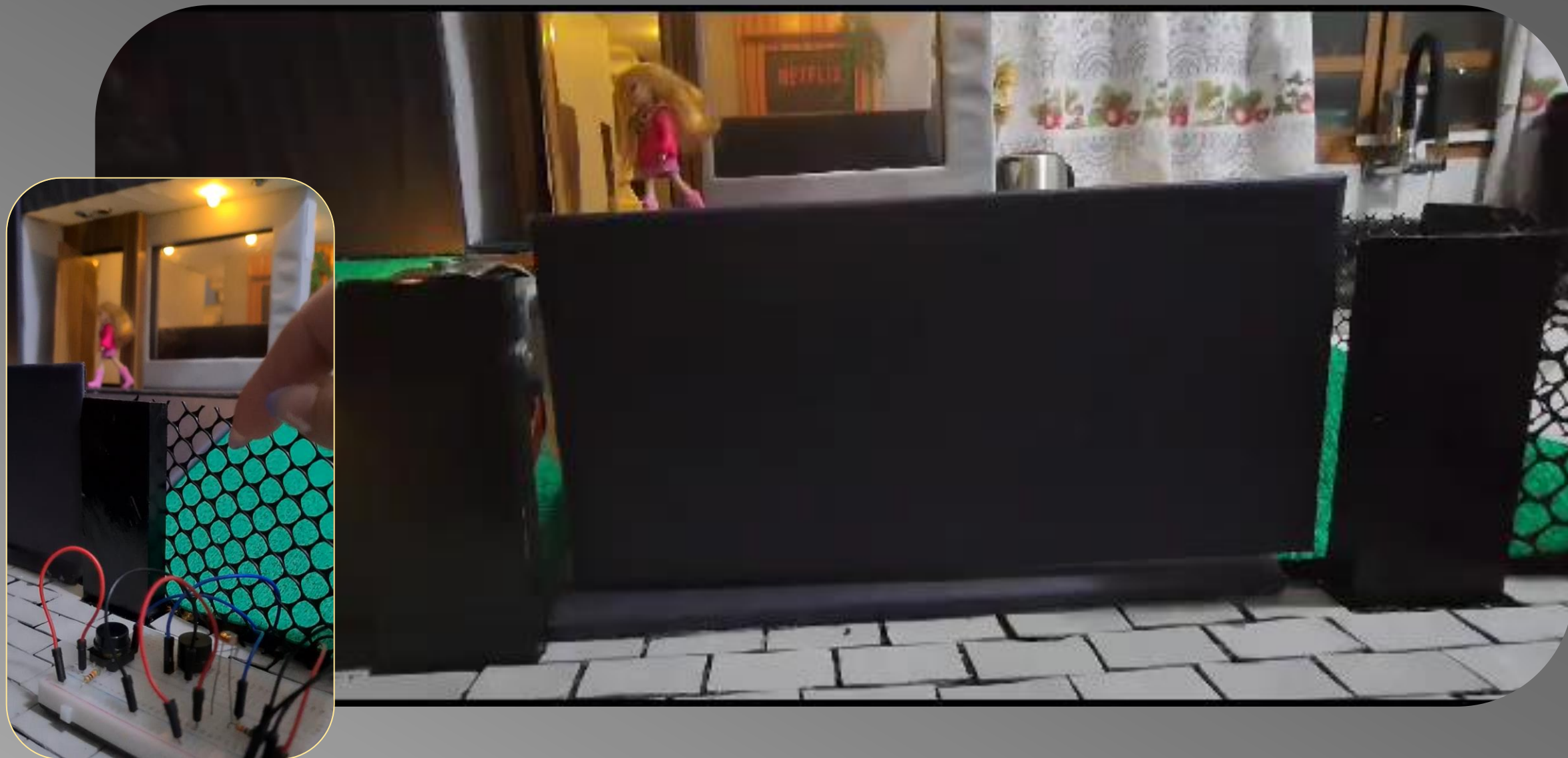

FUNCIONALIDADE - ARDUINO

```
82 // função para emitir um pulso em microssegundos, mensurar sua duração, e retornar um valor aproximado em centímetros
83 long readUltrasonicSensor () {
84     digitalWrite (trigger, HIGH); // aciona o pino trigger e emite um pulso de onda em microssegundos
85     delayMicroseconds (10); // permanece ativo por 10 microssegundos
86     digitalWrite (trigger, LOW); // desativa o pino trigger e interrompe o pulso de onda
87     long duration = pulseIn (echo, HIGH); // aciona o pino echo e mede a duração do pulso em microssegundos
88     Serial.println (duration);
89     return duration / 58; // retorna o valor em microssegundos dividido por 58 para estimar a distância em centímetros
90 }
91
92 // função para rotacionar o motor de passo em 90° no sentido horário
93 void rotateClockwise () {
94     for (int i = 0; i < 8; i++) {
95         stepMotor.step (steps); // utiliza o método step () da classe Stepper para rotacionar o motor de passo
96     }
97 }
98
99 // função para rotacionar o motor de passo em 90° no sentido anti horário
100 void rotateCounterClockwise () {
101     for (int i = 0; i < 8; i++) {
102         stepMotor.step (-steps); // utiliza o método step () da classe Stepper para rotacionar o motor de passo
103     }
104 }
```





DEMONSTRAÇÃO PRÁTICA - PORTÃO





DEMONSTRAÇÃO PRÁTICA - LUZES

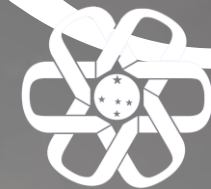




DEMONSTRAÇÃO PRÁTICA - ALARME



AUTOMAÇÃO RESIDENCIAL



UNESPAR
Universidade Estadual do Paraná



INFOS



Plano de
Apresentação.

Projetos
didáticos 2023.





OBRIGADO!
