

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL (SENAI)
CURSO TÉCNICO DESENS. DE SISTEMAS

KAIO GOMES DO NASCIMENTO MAZZA
RAFAEL DE ALMEIDA DE MAGALHÃES

CASA AUTOMATIZADA

Joinville

2025

**KAIO GOMES DO NASCIMENTO MAZZA
RAFAEL DE ALMEIDA DE MAGALHÃES**

CASA AUTOMATIZADA: Sistema Integrado de Automação Residencial

Trabalho Situação Aprendizagem
para apresentação ao Curso Técnico
Desenvolvimento de Sistemas do
Serviço Nacional de Aprendizagem
Industrial (SENAI) – Joinville – Santa
Catarina, como parte das exigências
para obtenção do grau de Técnico em
Desenvolvimento de Sistemas.

Orientador: Sergio Luiz da Silveira

JOINVILLE

2025

SUMARIO

Sumário

1. IDENTIFICAÇÃO DO PROJETO.....	4
2. DESCRIÇÃO GERAL DO PROJETO	5
3. OBJETIVOS DO PROJETO	5
3.1. OBJETIVO GERAL.....	5
3.2. OBJETIVOS ESPECÍFICOS	5
4. COMPONENTES UTILIZADOS.....	6
5. DIAGRAMA OU CROQUI DO PROJETO	7
6. CÓDIGO-FONTE (SKETCH ARDUINO)	9
7. FUNCIONAMENTO E TESTES	19
8. RESULTADOS ESPERADOS E BENEFÍCIOS	21
9. LINK DO GITHUB.....	21
10. CONCLUSÃO	22
11. REFERÊNCIAS	22

1. IDENTIFICAÇÃO DO PROJETO

Preencha a tabela abaixo com suas informações e dados.

Item	Descrição
Título do Projeto:	Casa Automatizada: Sistema Integrado de Automação Residencial
Área de Aplicação:	Residencial
Integrantes da Dupla:	Kaio Gomes do Nascimento Mazza Rafael de Almeida de Magalhães
Turma:	Técnico em Desenvolvimento de Sistemas – UC Internet das Coisas Turma: T DESN 2024/2 N1
Docente Responsável:	Sergio Luiz da Silveira
Período de Execução:	Início: 17/11/2025 Final:02/10/2025

2. DESCRIÇÃO GERAL DO PROJETO

O projeto consiste no desenvolvimento de um sistema IoT (Internet das Coisas) aplicado ao ambiente residencial, com o objetivo de automatizar tarefas, aumentar a segurança e melhorar o conforto dos moradores. A solução integra diversos componentes conectados, permitindo que diferentes funções da casa sejam executadas de forma automática e inteligente.

A proposta busca resolver problemas comuns em residências modernas, como falta de segurança, desperdício de energia, pouca praticidade em tarefas diárias e necessidade de monitoramento ambiental. Sensores de presença, chuvas, chuva e proximidade permitem que o sistema tome decisões de forma autônoma, enquanto servomotores e LEDs executam ações físicas no ambiente, como destrancar portas, abrir a garagem e acionar iluminação.

O impacto prático esperado inclui maior segurança, redução de consumo energético, automação de processos domésticos, conforto operacional e uma experiência residencial mais moderna e tecnológica, simulando funcionalidades encontradas em casas inteligentes reais.

3. OBJETIVOS DO PROJETO

Este projeto tem como finalidade demonstrar, na prática, a aplicação de conceitos de IoT na automação residencial, utilizando sensores, atuadores e lógica embarcada para proporcionar maior segurança, eficiência e praticidade. O sistema integra diferentes recursos tecnológicos que interagem entre si para automatizar rotinas domésticas e responder a eventos do ambiente.

3.1. OBJETIVO GERAL

Desenvolver um sistema de automação residencial utilizando princípios de IoT, capaz de monitorar o ambiente, reagir automaticamente a diferentes situações e facilitar atividades cotidianas dentro de uma residência.

3.2. OBJETIVOS ESPECÍFICOS

- Automatizar o destrancamento da porta utilizando tecnologia de leituras de tags.
- Controlar a abertura da garagem por meio de botões físico.
- Implementar um varal automatizado com sensor de chuva.
- Desenvolver uma lixeira automática acionada por aproximação.
- Criar um sistema de alarme para detecção de intrusos.
- Automatizar o acionamento de luzes a partir da presença.

- Monitorar riscos ambientais com a integração de um sensor de chamas.
- Demonstrar o uso integrado de sensores e atuadores aplicados em IoT para fins educacionais e residenciais.

4. COMPONENTES UTILIZADOS

Para a implementação do sistema de automação residencial, foram utilizados diversos componentes eletrônicos, módulos e sensores responsáveis por monitorar o ambiente e executar ações por meio de atuadores. Cada elemento desempenha uma função específica dentro do conjunto, permitindo que o projeto responda automaticamente a diferentes situações do cotidiano. A combinação desses dispositivos possibilita a integração de múltiplas funcionalidades.

A seguir, estão listados todos os componentes utilizados e suas respectivas funções.

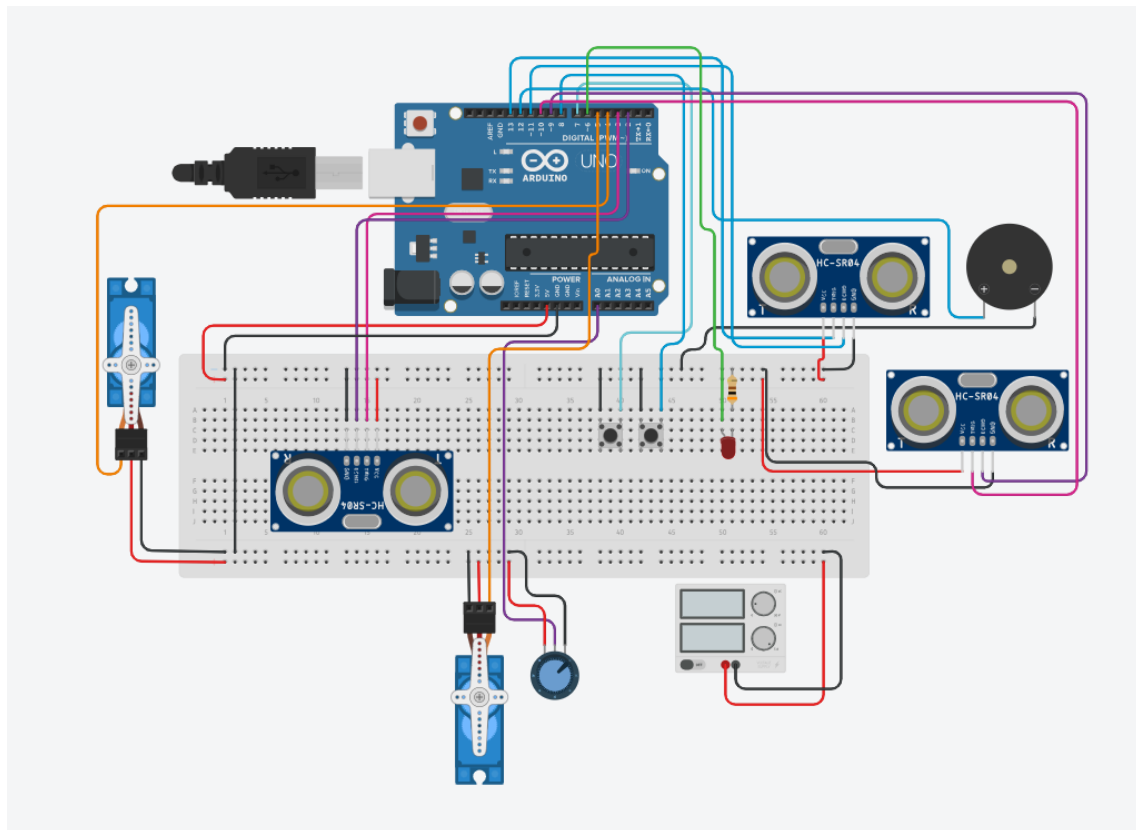
Componente	Quantidade	Função no sistema
Arduino UNO	2	Controlador principal
Sensor de Chuva	1	Detecta presença de água
Sensor de Chamas	1	Detecta fogo ou calor anormal
Sensor Ultrassônico	3	Mede distância / detecta proximidade
Servomotor	5	Realiza movimentos
Buzzer	3	Emite sons
Botão	4	Entrada de comando manual
Potenciômetro	1	Entrada de comando manual
RFID	1	Módulo de identificação de cartão/tag
Protoboard, fios jumper	—	Montagem do circuito

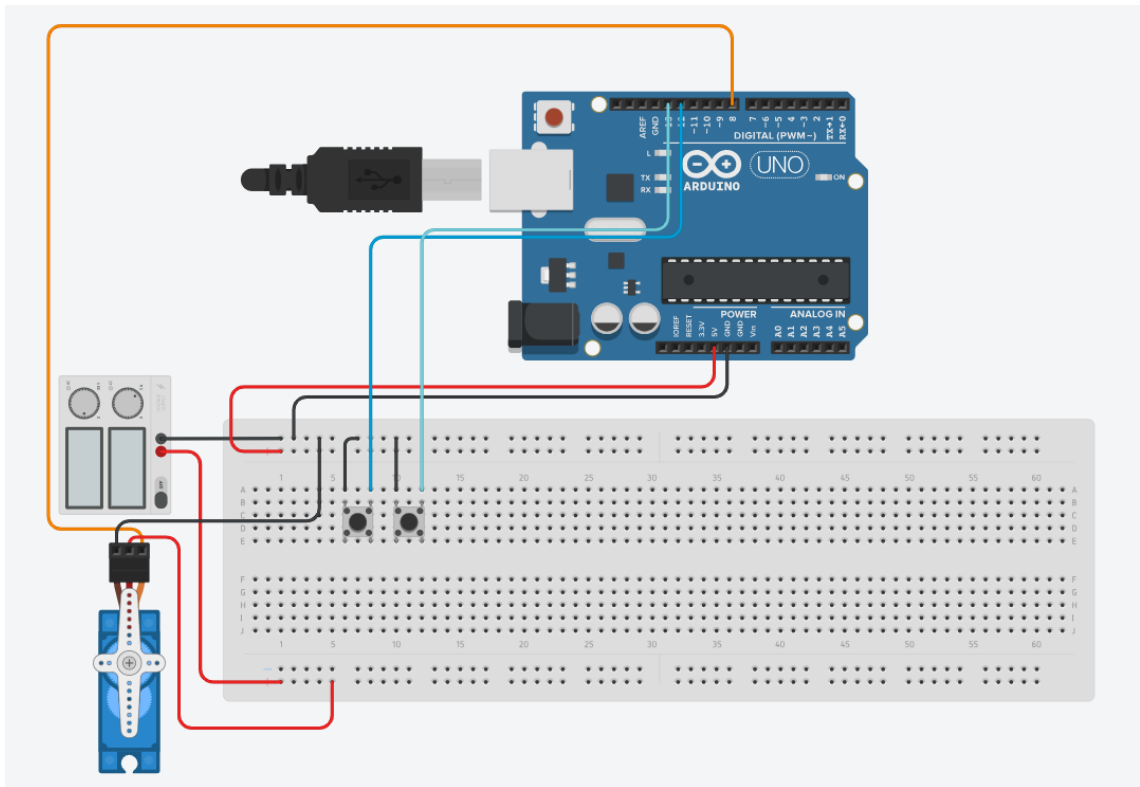
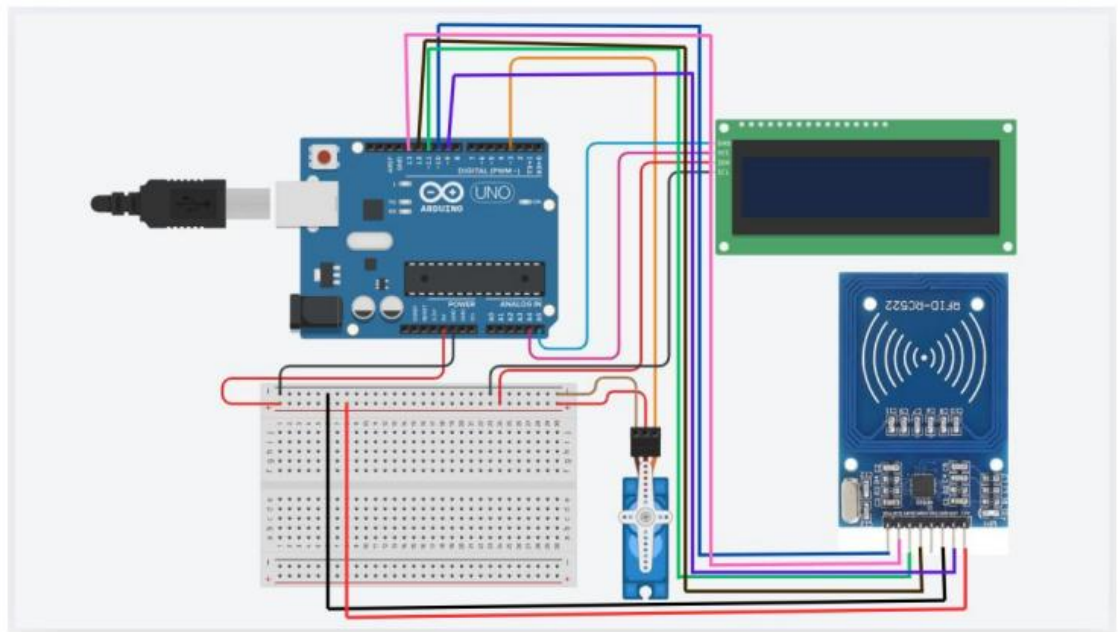
5. DIAGRAMA OU CROQUI DO PROJETO

Para representar a estrutura física e lógica da casa automatizada, foram elaborados projetos digitais contendo a ligação dos componentes eletrônicos, sensores, atuadores e unidades de controle. As imagens anexadas ilustram a

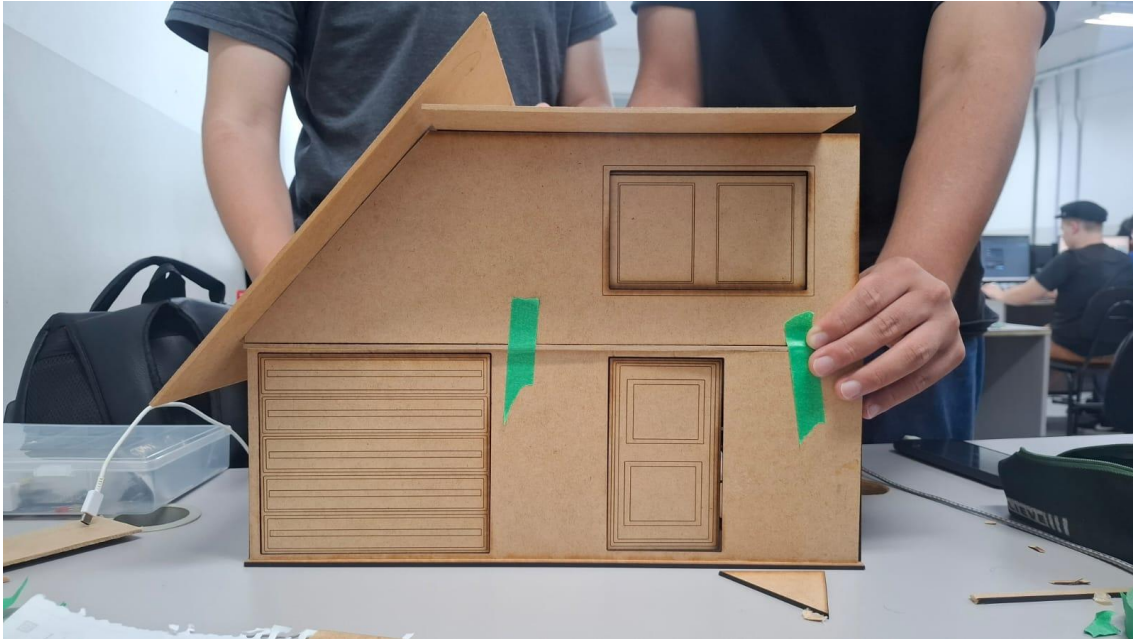
disposição dos módulos nos protoboards, a conexão com os Arduinos e a forma como cada subsistema interage, como o controle por RFID, os sensores ultrassônicos para presença e proximidade, o sistema de chuva, o módulo de chamas e os servomotores responsáveis pelos movimentos mecânicos.

Os diagramas também permitem visualizar claramente como a alimentação, sinais digitais e analógicos circulam entre os dispositivos, oferecendo uma visão completa da arquitetura IoT desenvolvida. Cada conjunto representa uma parte funcional da residência simulada, facilitando a compreensão do funcionamento geral e auxiliando no processo de montagem, manutenção e testes.





MAQUETE:



6. CÓDIGO-FONTE (SKETCH ARDUINO)

ARDUINO1

```
// =====  
// PROJETO LIXO  
// =====  
#include <Servo.h>  
Servo servoLixo;  
  
int lixo_ECHO = 2;  
int lixo_TRIG = 3;  
  
long lixo_duracao = 0;  
long lixo_distancia = 0;  
  
// =====  
// PROJETO VARAL  
// =====  
#define potenciometro A0  
Servo servoVaral;  
  
int leitura = 0;  
int angulo = 0;  
  
// =====
```

```

// PROJETO ALARME
// =====
#define alarme_TRIG 13
#define alarme_ECHO 11
#define buzzer 12
#define botaoLiga 7
#define botaoDesliga 8

bool alarmeLigado = false;
bool disparado = false;

long alarme_duracao = 0;
long alarme_distancia = 0;

// =====
// PROJETO LAMPADA
// =====
#define luz_TRIG 10    // pino TRIG do ultrassônico
#define luz_ECHO 9     // pino ECHO do ultrassônico
#define luz_led 6      // led

int luz_tempoInicio = 0;    // Guarda o momento em que o movimento foi
detectado
bool luz_luzAcesa = false; // Variável de controle

// =====
// SETUP
// =====
void setup() {
    Serial.begin(9600);

    // LIXO
    pinMode(lixo_ECHO, INPUT);
    pinMode(lixo_TRIG, OUTPUT);
    servoLixo.attach(4);
    servoLixo.write(0);

    // VARAL
    servoVaral.attach(5);
    servoVaral.write(0);

    // ALARME
    pinMode(alarme_TRIG, OUTPUT);
    pinMode(alarme_ECHO, INPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(botaoLiga, INPUT_PULLUP);

```

```

pinMode(botaoDesliga, INPUT_PULLUP);

noTone(buzzer); // garante buzzer desligado no início

//LAMPADA
pinMode(luz_TRIG, OUTPUT);
pinMode(luz_ECHO, INPUT);
pinMode(luz_led, OUTPUT);

}

// =====
// LOOP PRINCIPAL
// =====
void loop() {

// =====
// PROJETO LIXO
// =====
digitalWrite(lixo_TRIG, LOW);
delayMicroseconds(5);
digitalWrite(lixo_TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(lixo_TRIG, LOW);

lixo_duracao = pulseIn(lixo_ECHO, HIGH, 25000);
lixo_distancia = lixo_duracao / 58;

Serial.print("Lixeira: ");
Serial.print(lixo_distancia);
Serial.println(" cm");

if (lixo_distancia > 0 && lixo_distancia <= 50) {
    servoLixo.write(90);
    delay(3000);
} else {
    servoLixo.write(0);
}

delay(50);

// =====
// PROJETO VARAL
// =====
leitura = analogRead(potenciometro);
angulo = map(leitura, 0, 1023, 0, 90);

```

```
servoVaral.write(angulo);
```

```
Serial.print("Varal | Leitura: ");  
Serial.print(leitura);  
Serial.print(" Angulo: ");  
Serial.println(angulo);
```

```
delay(5);
```

```
// =====  
// PROJETO ALARME  
// =====
```

```
// BOTÃO LIGAR
```

```
if (digitalRead(botaoLiga) == LOW) {  
    alarmeLigado = true;  
    Serial.println("Alarme LIGADO");
```

```
    tone(buzzer, 1200); delay(150);  
    tone(buzzer, 2000); delay(150);  
    noTone(buzzer);  
    delay(200);  
}
```

```
// BOTÃO DESLIGAR
```

```
if (digitalRead(botaoDesliga) == LOW) {  
    alarmeLigado = false;  
    disparado = false;  
    noTone(buzzer);
```

```
    Serial.println("Alarme DESLIGADO");
```

```
    tone(buzzer, 800); delay(150);  
    tone(buzzer, 1000); delay(150);  
    noTone(buzzer);  
    delay(200);  
}
```

```
// LEITURA DO ULTRASSÔNICO DO ALARME
```

```
if (alarmeLigado) {
```

```
    digitalWrite(alarme_TRIG, LOW);  
    delayMicroseconds(5);
```

```
    digitalWrite(alarme_TRIG, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(alarme_TRIG, LOW);
```

```

    alarme_duracao = pulseIn(alarme_ECHO, HIGH, 25000);
    alarme_distancia = alarme_duracao / 58;

    Serial.print("Alarme: ");
    Serial.print(alarme_distancia);
    Serial.println(" cm");

    if (alarme_distancia > 0 && alarme_distancia <= 50) {
        disparado = true;
        Serial.println("INTRUSO DETECTADO!");
    }
}

// ALARME DISPARADO
if (disparado && alarmeLigado) {
    tone(buzzer, 1200);
    delay(200);
    tone(buzzer, 1600);
    delay(200);
} else {
    noTone(buzzer);
}

// =====
// PROJETO LAMPADA
// =====
// ----- LEITURA ULTRASSÔNICO (transformando em "movimento") -----
digitalWrite(luz_TRIG, LOW);
delayMicroseconds(5);

digitalWrite(luz_TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(luz_TRIG, LOW);

long luz_duracao = pulseIn(luz_ECHO, HIGH, 25000); // timeout 25ms
long luz_distancia = luz_duracao / 58;           // converte p/ cm

bool luz_movimento = false;

// Se a distância for válida e menor que 50 cm, consideramos "movimento"
if (luz_distancia > 0 && luz_distancia <= 50) {
    luz_movimento = true;
}

// -----

// Verificando "movimento" (como se fosse o PIR)

```

```

if (luz_movimento) {
  Serial.println("DETECTADO");

  digitalWrite(luz_led, HIGH);
  luz_luzAcesa = true;

  luz_tempoInicio = millis(); // Guarda o horário que detectou movimento
} else {
  Serial.println("n/a");
}

// Mantém a Luz acesa por 4 segundos (ou 150000 ms se quiser 2,5 minutos)
if (luz_luzAcesa) {
  if (millis() - luz_tempoInicio >= 4000) {
    digitalWrite(luz_led, LOW); // Apaga a luz após o tempo
    luz_luzAcesa = false;
    Serial.println("Luz Apagada");
  }
}
}
}

```

ARDUINO2:

```

//
=====
=
// PROJETO COMPLETO: GARAGEM AUTOMÁTICA
// RFID + BOTÕES + SERVO LENTO + SENSOR DE FOGO + SENSOR DE
CHUVA
//
=====
=

#include <Servo.h>
#include <MFRC522.h>
#include <SPI.h>

//
=====
=
// SERVOMOTOR DA GARAGEM
//
=====
=
Servo servoMotor;

const int botaoAbrir = 12;
const int botaoFechar = 13;

int leituraAbrirAtual, leituraAbrirAnterior = HIGH;

```

```

int leituraFecharAtual, leituraFecharAnterior = HIGH;

int anguloAtual  = 0;
int anguloDestino = 0;

//
=====
=
// RFID
//
=====
=
#define PINO_RST 9
#define PINO_SDA 10

#define LED_RED    2
#define LED_GREEN  3
#define BUZZER_DOOR 4

MFRC522 rfid(PINO_SDA, PINO_RST);

String cartao_rf  = " 93 d3 75 e4";
String chaveiro_rf = " 07 e3 41 66";

int FREQUENCIA = 3300;

//
=====
=
// SENSOR DE FOGO
//
=====
=
#define sensor_fogo 8
#define buzzer_fogo 7
#define freq_fogo 960

//
=====
=
// SENSOR DE CHUVA + SERVO DA JANELA
//
=====
=
#define pino_sensor_chuva A1 // Porta analógica
#define servo_janela 6      // Mudado de 0 → 6 (pino válido)

Servo servoJanela;
bool is_raining = false;

```

```

//
=====
=
// SETUP
//
=====
=
void setup() {

    Serial.begin(9600);

    // ---- SERVO E BOTÕES DA GARAGEM ----
    servoMotor.attach(5);
    servoMotor.write(0);

    pinMode(botaoAbrir, INPUT_PULLUP);
    pinMode(botaoFechar, INPUT_PULLUP);

    // ---- RFID ----
    SPI.begin();
    rfid.PCD_Init();

    pinMode(LED_GREEN, OUTPUT);
    pinMode(LED_RED, OUTPUT);
    pinMode(BUZZER_DOOR, OUTPUT);

    digitalWrite(LED_RED, HIGH);

    // ---- SENSOR DE FOGO ----
    pinMode(sensor_fogo, INPUT);
    pinMode(buzzer_fogo, OUTPUT);

    // ---- SENSOR DE CHUVA E SERVO DA JANELA ----
    pinMode(pino_sensor_chuva, INPUT);
    servoJanela.attach(servo_janela);
    servoJanela.write(0);

    Serial.println("Sistema iniciado. Aproxime a TAG.");
}

//
=====
=
// LOOP PRINCIPAL
//
=====
=
void loop() {

    //

```

```

=====
// 1) SENSOR DE FOGO (emergência)
//
=====
if (digitalRead(sensor_fogo) == HIGH) {
    alarmeIncendio();
}

//
=====
// 2) SENSOR DE CHUVA → ABRE/FECHA JANELA
//
=====
is_raining = digitalRead(pino_sensor_chuva);

if (is_raining) {
    servoJanela.write(180); // Fecha janela quando chove
} else {
    servoJanela.write(0);   // Abre janela quando não chove
}

//
=====
// 3) BOTÕES DA GARAGEM
//
=====
leituraAbrirAtual = digitalRead(botaoAbrir);
leituraFecharAtual = digitalRead(botaoFechar);

if (leituraAbrirAtual == LOW && leituraAbrirAnterior == HIGH) {
    anguloDestino = 90;
    delay(200);
}

if (leituraFecharAtual == LOW && leituraFecharAnterior == HIGH) {
    anguloDestino = 0;
    delay(200);
}

leituraAbrirAnterior = leituraAbrirAtual;
leituraFecharAnterior = leituraFecharAtual;

//
=====
// 4) RFID
//
=====
String conteudo = "";

if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {

```

```

for (byte i = 0; i < rfid.uid.size; i++) {
    conteudo.concat(String(rfid.uid.uidByte[i] < 16 ? " 0" : " "));
    conteudo.concat(String(rfid.uid.uidByte[i], HEX));
}

if (conteudo == chaveiro_rf || conteudo == cartao_rf) {
    Serial.println("Acesso Liberado!");
    efeitoAcesso();
    abrirEFecharGarage();
}
else {
    Serial.println("TAG desconhecida:");
    Serial.println(conteudo);
    efeitoErro();
}
}

//
=====
// 5) MOVIMENTO LENTO DO SERVO DA GARAGEM
//
=====
moverServoLento();
}

//
=====
=
// FUNÇÕES DO SISTEMA
//
=====
=

void moverServoLento() {
    if (anguloAtual < anguloDestino) {
        anguloAtual++;
        servoMotor.write(anguloAtual);
        delay(15);
    }
    else if (anguloAtual > anguloDestino) {
        anguloAtual--;
        servoMotor.write(anguloAtual);
        delay(15);
    }
}

void abrirEFecharGarage() {
    anguloDestino = 90;
    delay(15000);
}

```

```

    anguloDestino = 0;
}

void efeitoAcesso() {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, HIGH);

    tone(BUZZER_DOOR, FREQUENCIA); delay(150);
    noTone(BUZZER_DOOR);          delay(150);
    tone(BUZZER_DOOR, FREQUENCIA); delay(150);
    noTone(BUZZER_DOOR);          delay(500);

    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_RED, HIGH);
}

void efeitoErro() {
    for (int i = 0; i < 3; i++) {
        digitalWrite(LED_RED, LOW);
        tone(BUZZER_DOOR, FREQUENCIA);
        delay(100);
        digitalWrite(LED_RED, HIGH);
        noTone(BUZZER_DOOR);
        delay(100);
    }
}

void alarmeIncendio() {
    tone(buzzer_fogo, freq_fogo);
    delay(1500);
    noTone(buzzer_fogo);
    delay(1500);
}

```

7. FUNCIONAMENTO E TESTES

O sistema de automação residencial foi desenvolvido para operar de forma integrada, utilizando sensores para detectar eventos e atuadores para executar ações automáticas. Cada módulo funciona de maneira independente, mas todos se comunicam por meio dos dois Arduinos, garantindo respostas rápidas e precisas às condições do ambiente. A seguir, é descrito o funcionamento de cada parte do sistema e os testes realizados para validar seu desempenho.

Funcionamento:

Controle de acesso com RFID: O leitor RFID identifica uma tag autorizada e, ao validá-la, libera o servomotor responsável por destrancar a porta. Esse processo substitui chaves físicas, aumentando a praticidade e segurança do acesso.

Lixeira automática por aproximação: Um sensor ultrassônico detecta objetos próximos (como a mão do usuário) e aciona um servomotor que abre ou fecha a tampa da lixeira automaticamente, evitando contato manual.

Iluminação automática: Outro sensor ultrassônico monitora a presença. Quando detecta movimento, a luz é acesa automaticamente, contribuindo para economia de energia e comodidade ao usuário.

Varal e janela automatizados com sensor de chuva: Ao identificar gotas de água, o sensor de chuva envia o sinal para dois servomotores:

- um responsável por recolher o varal,
- outro responsável por fechar a janela.

Isso protege as roupas e evita a entrada de água na residência.

Controle manual do varal com potenciômetro: Além do modo automático por chuva, o varal pode ser ajustado manualmente utilizando um potenciômetro. O usuário consegue abrir ou fechar o varal gradualmente, conforme necessidade.

Alarme por presença: Um sensor ultrassônico dedicado detecta quando há aproximação indesejada. Quando isso ocorre, um buzzer é acionado, funcionando como alarme sonoro. O sistema conta com **dois botões**, um para ativar e outro para desativar o modo de alarme.

Controle da garagem: Foram implementados **dois botões** adicionais que permitem abrir e fechar a garagem por meio de um servomotor. O acionamento é rápido e funciona como um controle simplificado.

Alarme por detecção de chamas: O sensor de chamas monitora sinais de fogo e, ao detectar algo, aciona um alerta sonoro através do buzzer. Isso aumenta a segurança e permite reação rápida em situações de risco.

Testes Realizados:

Todos os testes realizados foram feitos de forma separada, e no final todos os códigos foram juntados e testados em conjunto.

Teste do RFID: verificação da leitura da tag e do acionamento correto do servomotor da porta.

Teste dos sensores ultrassônicos: medição de distância, tempo de resposta, ativação da lixeira, luz automática e alarme.

Teste do sensor de chuva: simulação de gotas para confirmar o recolhimento do varal e fechamento da janela.

Teste do potenciômetro: controle manual do varal, avaliando suavidade e precisão do movimento.

Teste dos botões: abertura e fechamento da garagem, ativação e desativação do alarme.

Teste do sensor de chamas: simulação de fonte de temperatura para verificar o acionamento do alerta.

Teste integrado: validação de todos os módulos funcionando simultaneamente, garantindo que não houvesse conflitos entre sensores no sistema.

8. RESULTADOS ESPERADOS E BENEFÍCIOS

A implementação da Casa Automatizada proporciona uma série de impactos práticos e benefícios diretos ao ambiente residencial. Os resultados esperados incluem maior segurança, eficiência energética e conforto no dia a dia dos usuários. Com a automação de processos rotineiros, como acender luzes, abrir portas, acionar alarmes e proteger o varal da chuva, o sistema reduz a necessidade de intervenção manual, tornando a residência mais prática e moderna.

Além disso, o uso de sensores inteligentes evita desperdício de energia, pois a iluminação só permanece acesa quando há presença detectada. A segurança também é reforçada por meio de alarmes automáticos e controle de acesso por RFID, diminuindo riscos de invasões e incidentes domésticos. O projeto também oferece maior proteção do ambiente e dos pertences da casa, por exemplo, com o fechamento automático da janela ao detectar chuva e o alerta imediato em caso de chamas.

9. GITHUB

[Clique aqui para abrir o link do GitHub](#)

10. CONCLUSÃO

O desenvolvimento da Casa Automatizada demonstrou, na prática, a viabilidade e eficiência da aplicação de tecnologias IoT em soluções residenciais de pequeno porte. A integração entre sensores, atuadores e lógica embarcada no

Arduino permitiu criar um sistema funcional, capaz de monitorar o ambiente e reagir automaticamente a diferentes situações do cotidiano.

O projeto evidenciou que, mesmo com componentes considerados simples, é possível construir um sistema completo de automação, trazendo segurança, economia de energia e praticidade ao usuário. A experiência também reforçou a importância da prototipagem, testes e integração de múltiplos módulos, contribuindo para o aprendizado técnico dos alunos.

Assim, conclui-se que o projeto cumpre seus objetivos educacionais e funcionais, mostrando o potencial da automação residencial como uma área em expansão e acessível, com soluções que podem ser aprimoradas e escaladas para aplicações mais complexas no futuro.

11. REFERÊNCIAS

Links e materiais usados como base (vídeos, artigos, manuais etc.):

- [Casa automatizada Arduino Uno - Amazon](#)
- [Tutorial de como utilizar o sensor de chuva - YouTube](#)
- [Tutorial de como utilizar o sensor de chamas - YouTube](#)
- [Tutorial de como utilizar o sensor RFID - YouTube](#)