

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

**EBERTH DA SILVA RODRIGUES**  
**RUAN AUGUSTO ALVES**

**SISTEMA INTELIGENTE DE ILUMINAÇÃO COM DETECÇÃO, CONTROLE  
MANUAL E TEMPORIZAÇÃO AJUSTÁVEL**

**JOINVILLE**

**2025**

**EBERTH DA SILVA RODRIGUES  
RUAN ALVES**

**SISTEMA DE ILUMINAÇÃO INTELIGENTE: PROTÓTIPO DE ILUMINAÇÃO  
COM DETECÇÃO DE PRESENÇA, CONTROLE MANUAL E  
TEMPORIZAÇÃO AJUSTÁVEL**

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: Prof. SERGIO LUIZ DA  
SILVEIRA

**JOINVILLE**  
**2025**  
**SUMÁRIO**

**Sumário**

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

## 1. IDENTIFICAÇÃO DO PROJETO

Preencha a tabela abaixo com suas informações e dados.

Item	Descrição
Título do Projeto:	Sistema Inteligente de Iluminação com Detecção de Presença, Controle Manual e Temporização Ajustável
Área de Aplicação:	RESIDENCIAL
Integrantes da Dupla:	EBERTH DA SILVA RODRIGUES E RUAN AUGUSTO ALVES
Turma:	Técnico em Desenvolvimento de Sistemas – UC Internet das Coisas
Docente Responsável:	SERGIO LUIZ DA SILVEIRA
Período de Execução:	17/11/2025
Link Repositório GitHub	<a href="https://github.com/RuanAlves07/SensorMovimentoArduino.git">https://github.com/RuanAlves07/SensorMovimentoArduino.git</a>
Link TinkerCAD projeto teórico	<a href="https://www.tinkercad.com/things/eUiTs6TYXNQ-projeto-sa-final?sharecode=OOjMCHPE3mKT0294_3w_3k_LU1UMcTJeF6OtwR89ozw">https://www.tinkercad.com/things/eUiTs6TYXNQ-projeto-sa-final?sharecode=OOjMCHPE3mKT0294_3w_3k_LU1UMcTJeF6OtwR89ozw</a>

## 2. DESCRIÇÃO GERAL DO PROJETO

O projeto realiza a **detecção de presença**, permitindo que a iluminação seja acionada automaticamente quando alguém entra no ambiente, com opções de controle manual ou temporização ajustável (30 segundos, 1 minuto ou 5 minutos). Isso oferece flexibilidade ao usuário, que pode escolher entre modos totalmente automáticos, manuais ou desligados, conforme a necessidade.

**Os principais problemas que ele resolve incluem:**

- A falta de praticidade de acionar manualmente as luzes em ambientes de passagem ou uso esporádico;
- O desperdício de energia causado por luzes deixadas acesas acidentalmente;
- A necessidade constante de interação humana em tarefas repetitivas que podem ser automatizadas com segurança e eficiência.

O impacto prático esperado é a **melhoria do conforto**, a economia de energia elétrica e a promoção de hábitos sustentáveis em ambientes residenciais ou comerciais, por meio de uma solução simples, acessível e inteligente que integra sensores, interface de usuário e lógica de controle automatizado.

### 3. OBJETIVOS DO PROJETO

#### **Objetivo geral:**

Desenvolver um projeto IOT com foco em um detector de presença com sensor PIR e conseguir controlar o tempo de funcionalidade do mesmo.

#### **Objetivos específicos:**

Aplicar funcionalidades com o painel de controle e o sensor

Automatizar processo com o Arduino sobre o tempo de iluminação

#### 3.1. OBJETIVO GERAL

Desenvolver um protótipo automatizado de iluminação inteligente capaz de operar em diferentes modos (desligado, manual e automático com temporização), utilizando detecção de presença e interface interativa com o usuário.

#### 3.2. OBJETIVOS ESPECÍFICOS

- Implementar um sistema de detecção de presença com sensor PIR para acionamento automático da iluminação;
- Desenvolver uma interface de usuário com display LCD 16×2 para exibição do modo de operação, tempo restante e mensagens de status;
- Criar três modos de funcionamento selecionáveis: Desligado, Ligado (manual contínuo) e Temporizador (automático com ajuste de tempo);
- Permitir ao usuário configurar o tempo de desligamento automático em três opções: 30 segundos, 1 minuto ou 5 minutos;
- Garantir que a interação do usuário com os botões de controle não cause falsos acionamentos do sensor de presença, por meio de posicionamento adequado dos componentes;
- Integrar todos os componentes em uma maquete funcional compatível com os recursos disponíveis no laboratório de IoT.

#### 4. COMPONENTES UTILIZADOS

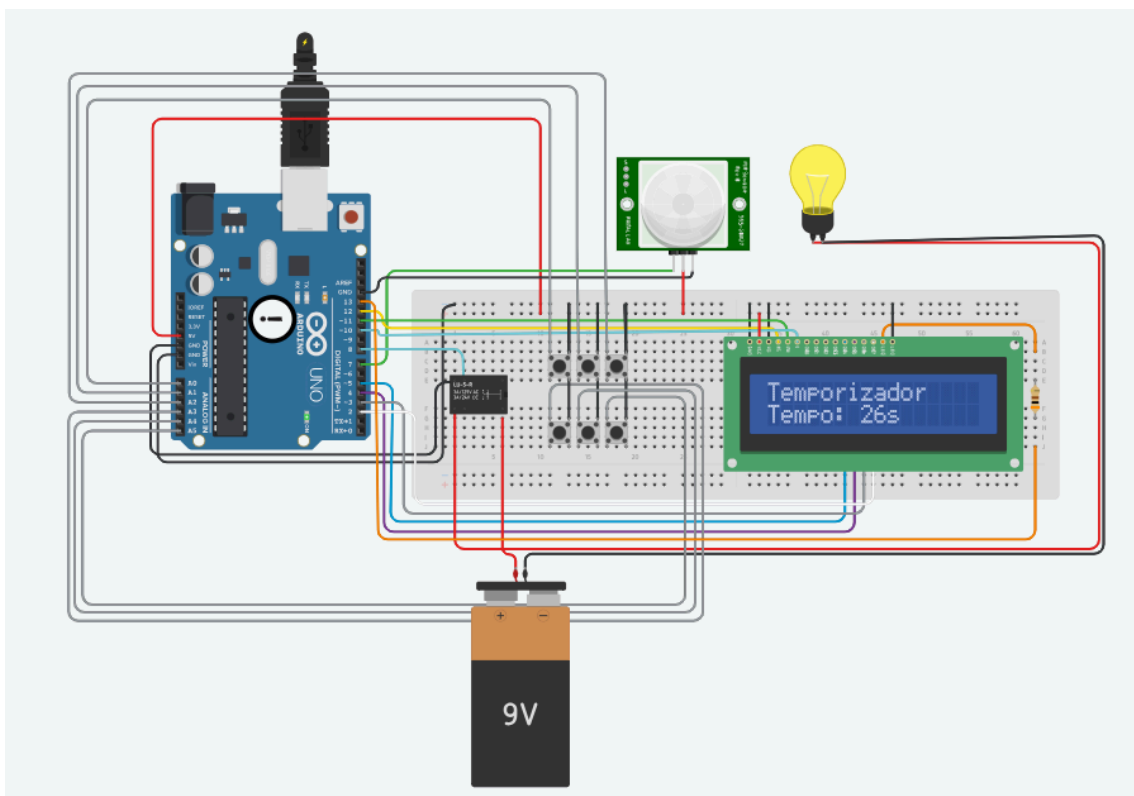
Componentes utilizados

Componente	Quantidade	Função no sistema
Sensor PIR (HC-SR501)	1	Detecção de presença
Módulo Relé	1	Ligar/desligar Lâmpada
Botão	6	Escolher: 30s, 1min e 5 min
Display LCD 16x2	1	Feedback visual do usuário
Microcontrolador (arduino)	1	Cérebro do sistema
Fonte de alimentação (Bateria 9v)	1	Alimenta todo o circuito
Lâmpada	1	iluminar ambiente local

## 5. DIAGRAMA OU CROQUI DO PROJETO

A imagem abaixo tem o intuito de **informar as funcionalidades do projeto**, como detecção de proximidade, opções de ligar/desligar e um seletor de botões com tempo pré-determinado (30s, 1m e 5m). Os componentes também serão explicados com base na sua funcionalidade no projeto em si.

### 5.1. CONEXÕES DOS COMPONENTES NO ARDUINO:

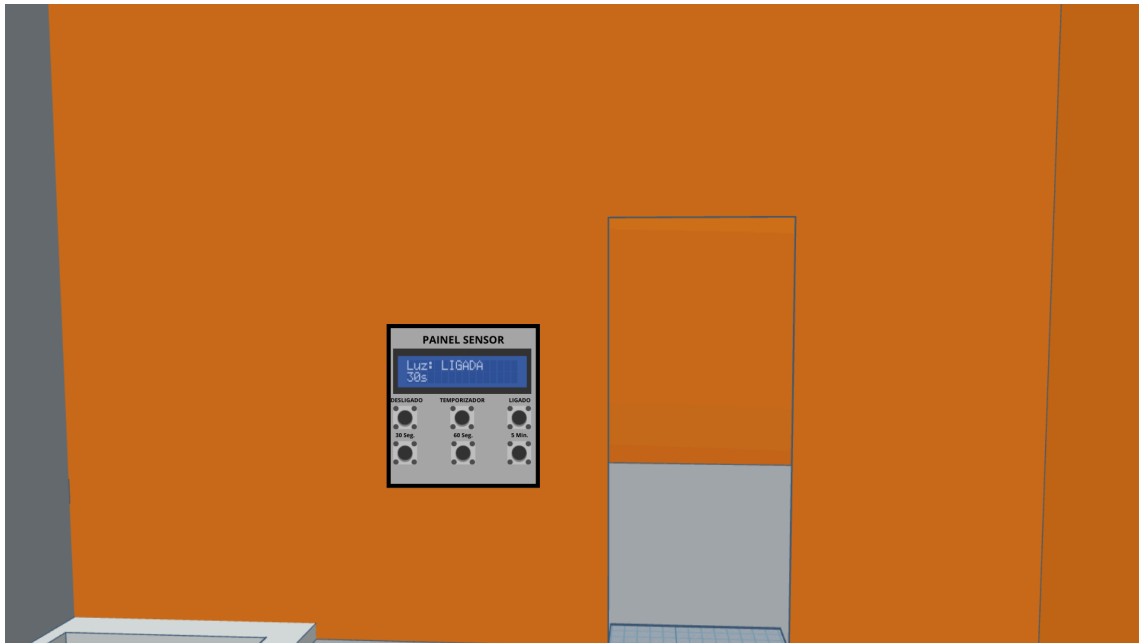






Na foto há:

- Sensor Pir (para detectar movimento)
- Lâmpada (recebe a informação para realizar o acionamento ou desligamento)



Na foto há:

- Painel LCD para informar usuário
- Botões com as seguintes opções:
  - Desligar
  - Temporizador
  - Ligado
  - 30 Seg.
  - 60 Seg.
  - 5 Min.

## 6. CÓDIGO-FONTE (SKETCH ARDUINO)

```
#include <LiquidCrystal.h>

// Configuração do LCD
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

// Pinos
int rele = 8;
int PIR = 7;
int backLight = 13;

// Botões (usando pinos analógicos como digitais)
// Botões do painel que será usado como central para realizar os ajustes

const int BTN_DESLIGAR = A0; // Botão 1, Esquerdo superior
const int BTN_TEMPORIZADOR = A1; // Botão 2, Centro superior
const int BTN_LIGAR = A2; // Botão 3, Direita superior
const int BTN_30S = A3; // Botão 4, Esquerdo inferior
const int BTN_60S = A4; // Botão 5, Centro inferior
const int BTN_300S = A5; // Botão 6, Direita inferior

// Variáveis de controle
unsigned long lastUpdate = 0;
int defaultTime = 30;
int remainingTime = 30;
bool motionDetected = false;
bool timerMode = true;

// Debounce do PIR
unsigned long lastPIRChange = 0;
const unsigned long DEBOUNCE_DELAY = 500;
int lastPIRState = LOW;

void setup() {
  pinMode(rele, OUTPUT);
  pinMode(PIR, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(backLight, HIGH);

  // Configuração dos botões com pull-up interno

  pinMode(BTN_LIGAR, INPUT_PULLUP);
  pinMode(BTN_TEMPORIZADOR, INPUT_PULLUP);
  pinMode(BTN_30S, INPUT_PULLUP);
  pinMode(BTN_60S, INPUT_PULLUP);
  pinMode(BTN_300S, INPUT_PULLUP);
  pinMode(BTN_DESLIGAR, INPUT_PULLUP);

  lcd.begin(16, 2);
```

```

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temporizador");
    lcd.setCursor(0, 1);
    lcd.print("30s");

    Serial.begin(9600);
}

void loop() {
    // Leitura do PIR

    int currentPIRState = digitalRead(PIR);
    if (millis() - lastPIRChange >= DEBOUNCE_DELAY) {
        if (currentPIRState != lastPIRState) {
            lastPIRState = currentPIRState;
            lastPIRChange = millis();
        }
    }
    motionDetected = (lastPIRState == HIGH);

    // Leitura dos botões
    bool btnLigar = digitalRead(BTN_LIGAR) == LOW;
    bool btnTemp = digitalRead(BTN_TEMPORIZADOR) == LOW;
    bool btn30s = digitalRead(BTN_30S) == LOW;
    bool btn60s = digitalRead(BTN_60S) == LOW;
    bool btn300s = digitalRead(BTN_300S) == LOW;
    bool btnDesligar = digitalRead(BTN_DESLIGAR) == LOW;

    // Debug no Serial Monitor
    // Existência ótima pra ver se tá tudo funcionando internamente
    Serial.print("BTN_LIGAR=");
    Serial.print(btnLigar ? "1" : "0");
    Serial.print(" | BTN_TEMP=");
    Serial.print(btnTemp ? "1" : "0");
    Serial.print(" | BTN_DESL=");
    Serial.print(btnDesligar ? "1" : "0");
    Serial.print(" | PIR=");
    Serial.println(motionDetected ? "1" : "0");

    // Botões de tempo
    if (btn30s) {
        defaultTime = 30;
        if (timerMode) {
            remainingTime = 30;
        }
    }
    lcd.setCursor(0, 1);
    lcd.print("Tempo: 30s    ");
    delay(300);
}

```

```

if (btn60s) {
    defaultTime = 60;
    if (timerMode) {
        remainingTime = 60;
    }
    lcd.setCursor(0, 1);
    lcd.print("Tempo: 60s  ");
    delay(300);
}
if (btn300s) {
    defaultTime = 300;
    if (timerMode) {
        remainingTime = 300;
    }
    lcd.setCursor(0, 1);
    lcd.print("Tempo: 300s  ");
    delay(300);
}

// Botões de controle
if (btnLigar) {
    timerMode = false;
    digitalWrite(rele, HIGH);
    lcd.setCursor(0, 0);
    lcd.print("Luz: LIGADA  ");
    delay(300);
}
if (btnTemp) {
    timerMode = true;
    remainingTime = defaultTime;
    lcd.setCursor(0, 0);
    lcd.print("Temporizador  ");
    delay(300);
}
if (btnDesligar) {
    timerMode = false;
    digitalWrite(rele, LOW);
    lcd.setCursor(0, 0);
    lcd.print("Luz: DESLIGADA ");
    delay(300);
}

// Modo Temporizador
if (timerMode) {
    if (millis() - lastUpdate >= 1000) {
        lastUpdate = millis();

        if (motionDetected) {
            remainingTime = defaultTime;
        } else {

```

```

    if (remainingTime > 0) {
        remainingTime--;
    }
}

lcd.setCursor(0, 1);
lcd.print("      ");
lcd.setCursor(0, 1);
lcd.print("Tempo: ");
lcd.print(remainingTime);
lcd.print("s");

if (remainingTime <= 0) {
    digitalWrite(rele, LOW);
} else {
    digitalWrite(rele, HIGH);
}

    // M significa "Movimento", então quando o objeto se move no sensor,
    aparece na tela LCD um M para indicar isso
    lcd.setCursor(14, 0);
    lcd.print(motionDetected ? "M" : " ");
}
}

digitalWrite(LED_BUILTIN, digitalRead(rele));
delay(50);
}

```

## **7. FUNCIONAMENTO E TESTES**

Ao realizar a montagem, o sistema teve testes passando pelos seus 3 módulos de temporizador, sendo: 30s, 60s e 5 minutos. Todos com eles ao finalizarem seu tempo e não acharem nenhuma atividade de movimento na região do sensor PIR, o mesmo desligou após a finalização do tempo. Realizado testes também que, enquanto o tempo estava decaindo gradativamente, o mesmo entrou em movimento e resetou o tempo, voltando ao tempo selecionado no painel. testado os botões de processos manuais, no qual não envolvem o sensor PIR (liga e desliga).

## **8. RESULTADOS ESPERADOS E BENEFÍCIOS**

As principais motivações e razões que tivemos ao realizar este projeto, temos como ideia ter esses seguintes objetivos para a razão do uso:

- Economia de energia
- Conforto e praticidade
- Solução de baixo custo e fácil replicação
- Promoção de hábitos sustentáveis

## **9. CONCLUSÃO**

O projeto comprovou a eficácia de uma solução simples e de baixo custo para automação de iluminação residencial, integrando sensor de presença, controle manual e temporização ajustável por meio do Arduino. A implementação atingiu os objetivos propostos, demonstrando ganhos em conforto, economia de energia e usabilidade, além de reforçar o potencial da IoT em aplicações práticas do cotidiano.

## **10. REFERÊNCIAS**

- [Minha 1ª Automação: Automática com Sensor de luz e Arduino](#)