

```
/*
```

Este é um projeto de um sensor de luminosidade com LEDs e LCD.

O sensor é responsável por fazer a leitura da iluminação ambiente. Essa informação será transmitida para o display para ser exposta ao usuário.

Os LEDs se comportam de acordo com o valor do LDR, convertido para porcentagem.

```
*/
```

```
// Definição dos pinos
```

```
const int pinoLDR = A0; // LDR está conectado ao pino A0
```

```
// Definição de Variáveis
```

```
int valorLDR; // Representa a leitura do sensor
```

```
int intensidadeLuz; // Variável que representa o valor do LDR convertido para porcentagem.
```

```
int mediaMovel[10]; // Array para média.
```

```
int contagemLeituras = 0; // Contador de leituras
```

```
int redLight = 6; // Variável para o LED vermelho (pino 6)
```

```
int yellowLight = 7; // Variável para o LED amarelo (pino 7)
```

```
int greenLight = 8; // Variável para o LED verde (pino 8)
```

```
int buzzerPin = 9; // Variável para o buzzer (pino 9)
```

```
// Para o funcionamento do LCD, é incluída a biblioteca LiquidCrystal.
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
```

```
// Definição dos caracteres especiais, que estarão presentes na indicação dos valores no display
```

```
byte sad[8] = {
```

```
    B00000,
```

```
    B10001,
```

```
    B00000,
```

```
    B00000,
```

```
    B01110,
```

```
    B10001,
```

```
    B00000,
```

```
};
```

```
byte happy[8] = {
```

```
    B00000,
```

```
    B10001,
```

```
    B00000,
```

```
    B00000,
```

```
    B10001,
```

```
    B01110,
```

```
B00000,  
};
```

```
byte alert[8] = {  
    B00000,  
    B10001,  
    B00000,  
    B00000,  
    B11111,  
    B00000,  
    B00000,  
};
```

```
// Implementação da função setup  
void setup() {  
    Serial.begin(9600); // Indica o início do processo  
    pinMode(pinoLDR, INPUT); // Definição do LDR como entrada  
    lcd.begin(16,2); // Indica que o LCD deve ser ligado  
    lcd.clear(); // Limpa o display  
    lcd.setCursor(0,0); // Indicador da posição do caractere no LCD  
    lcd.print("Sensor de Luz"); // Mensagem de apresentação  
    lcd.setCursor(0,1); // Indicador da posição do caractere no LCD  
    lcd.print("Ligado!"); // Mensagem de apresentação  
    // Criação dos caracteres especiais, usando os bytes definidor anteriormente  
    lcd.createChar(0, sad);  
    lcd.createChar(1, happy);  
    lcd.createChar(2, alert);  
    // Definição dos LEDs e buzzer como saída  
    pinMode(redLight, OUTPUT);  
    pinMode(yellowLight, OUTPUT);  
    pinMode(greenLight, OUTPUT);  
    pinMode(buzzerPin, OUTPUT);  
    delay(2000); // A primeira mensagem do LCD ficará na tela por 2 segundos  
    lcd.clear(); // Limpa o display  
}
```

```
void loop() {  
    // Leitura do sensor  
    valorLDR = analogRead(pinoLDR);  
  
    // Adiciona a nova leitura à média móvel  
    for (int i = 9; i > 0; i--) {  
        mediaMovel[i] = mediaMovel[i - 1];
```

```

}
mediaMovel[0] = valorLDR;
contagemLeituras++;

// Verifica se já foram feitas 10 leituras
if (contagemLeituras >= 10) {
    // Cálculo da média dos valores lidos, para maior precisão
    int soma = 0;
    for (int i = 0; i < 10; i++) {
        soma += mediaMovel[i];
    }
    valorLDR = soma / 10;
}

// Conversão do valorLDR para porcentagem, definindo uma escala de 0 a 100
intensidadeLuz = map(valorLDR, 0, 957, 0, 100);
if (intensidadeLuz >= 0 && intensidadeLuz <= 33)
{
    // Quando o valor do LDR estiver baixo, apenas o LED verde acenderá e o buzzer
    permanecerá silencioso, indicando um ambiente favorável
    digitalWrite(greenLight, HIGH);
    digitalWrite(yellowLight, LOW);
    digitalWrite(redLight, LOW);
    digitalWrite(buzzerPin, LOW);
    // Configuração das informações que devem ser apresentadas no display
    lcd.print("Luminosidade:");
    lcd.write(byte(0));
    lcd.setCursor(0,1);
    lcd.print((intensidadeLuz));
    lcd.print("%");
}
else if (intensidadeLuz >= 34 && intensidadeLuz <= 66)
{
    // Quando o valor do LDR estiver intermediário, apenas o LED acenderá e o buzzer é ativado
    em intervalos de 3 segundos, indicando um ambiente desfavorável
    digitalWrite(yellowLight, HIGH);
    digitalWrite(redLight, LOW);
    digitalWrite(greenLight, LOW);
    digitalWrite(buzzerPin, HIGH);
    // Configuração do tom e do loop do buzzer
    tone(buzzerPin, 440);
    delay(3000);
    digitalWrite(buzzerPin, LOW);
    noTone(buzzerPin);
}

```

```

    delay(5000);
    digitalWrite(buzzerPin, HIGH);
    // Configuração das informações que devem ser apresentadas no display
    lcd.print("Luminosidade:");
    lcd.write(byte(2));
    lcd.setCursor(0,1);
    lcd.print((intensidadeLuz));
    lcd.print("%");
}
else if (intensidadeLuz >= 67 && intensidadeLuz <= 100)
{
    // Quando o valor do LDR estiver alto, apenas o LED vermelho acenderá e o buzzer
    permanece silencioso, indicando possíveis danos ao produto
    digitalWrite(redLight, HIGH);
    digitalWrite(greenLight, LOW);
    digitalWrite(yellowLight, LOW);
    digitalWrite(buzzerPin, LOW);
    // Configuração das informações que devem ser apresentadas no display
    lcd.print("Luminosidade:");
    lcd.setCursor(0,1);
    lcd.write(byte(1));
    lcd.setCursor(0,1);
    lcd.print((intensidadeLuz));
    lcd.print("%");
}
else
{
    // Neste caso o projeto não apresenta resposta, pois os valores podem estar incorretos
    digitalWrite(redLight, LOW);
    digitalWrite(greenLight, LOW);
    digitalWrite(yellowLight, LOW);
    digitalWrite(buzzerPin, LOW);
}

// Imprime a porcentagem do valor medido pelo sensor
Serial.print("Intensidade de Luz: ");
Serial.println(intensidadeLuz);

delay(10); // Indica uma leitura a cada 10 milissegundos
delay(100);
lcd.setCursor(15, 1);

```

}