



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

Dimmer Digital

Equipe:

João Marcos

João Filipe

Rodrigo Moura

Paulo Roberto

Professor: Afonso

Dimmer Digital

- **Introdução:**

A automação residencial, a qual é uma tendência global na atualidade, encontra-se em constante evolução, na medida em que surgem, quase que instantaneamente, novos dispositivos, sistemas e técnicas de controle, gerando, com isso, melhorias e conforto na vida de diversas pessoas. O dimmer digital, por exemplo, é um dispositivo que altera a intensidade luminosa em ambientes de acordo com a incidência de luz local, sendo este controle, portanto, de suma importância na iluminação residencial tornando o ambiente mais versátil e agradável, além disso, é um bom aparelho para ter uma redução no desperdício de energia elétrica

Dessa forma, neste projeto iremos mostrar o passo a passo da como é feita a implementação deste dispositivo, a partir de componentes eletrônicos básicos, módulo bluetooth e um micro controlador.

Este projeto será desenvolvido com os seguintes equipamentos:

- Micro controlador arduino UNO – Controlador do sistema.
- Led's 5 mm de alto brilho – Atuadores do sistema
- Ldr - Este componente será o sensor do sistema.
- Módulo bluetooth HC05
- Demais componentes: placa protoboard, resistores de 1k e de 10k, fios e cabos jumpers diversos.

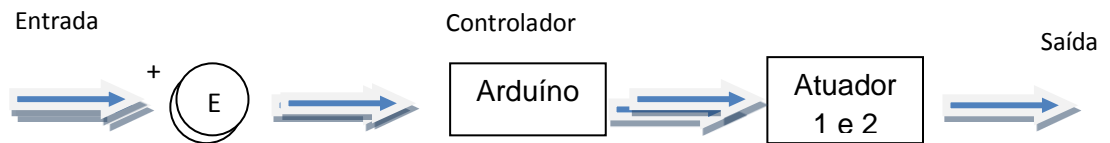
Fundamentação teórica:

O controle automático de sistemas, desde o século XVIII , quando James Watt realizou o primeiro feito significativo relativo a isto, construindo um regulador centrífugo para controlar a velocidade de uma máquina a vapor, revela-se essencial em qualquer campo da engenharia e da ciência. Com o passar dos anos foram sendo desenvolvidas ferramentas de análise e representação de sistemas, bem como equações matemáticas e funções de transferência para modelagem de sistemas.

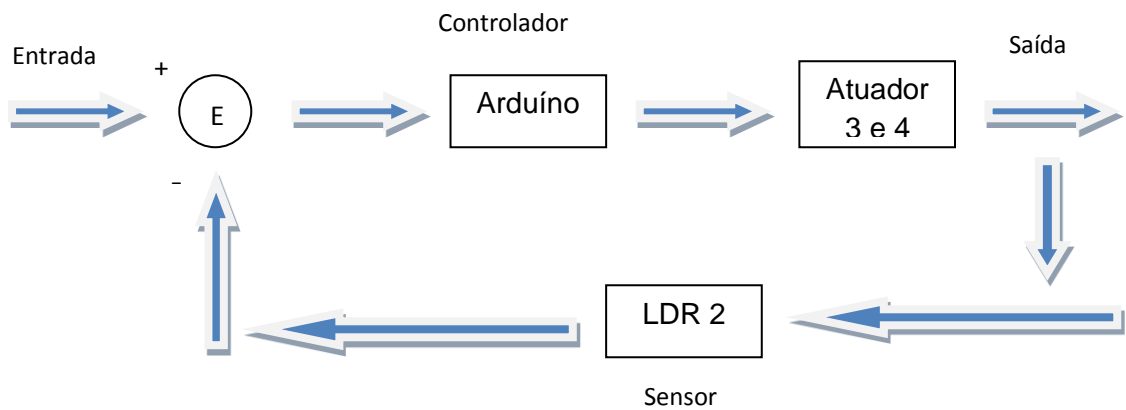
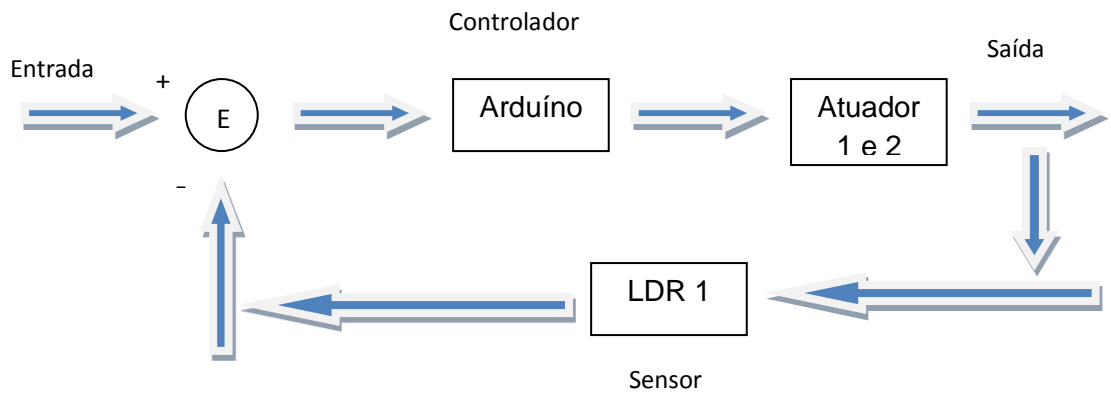
Seguem abaixo o fluxograma e os diagramas de blocos que representam sistemas de malha fechada e malha aberta referente a este projeto.

- Diagrama de Blocos

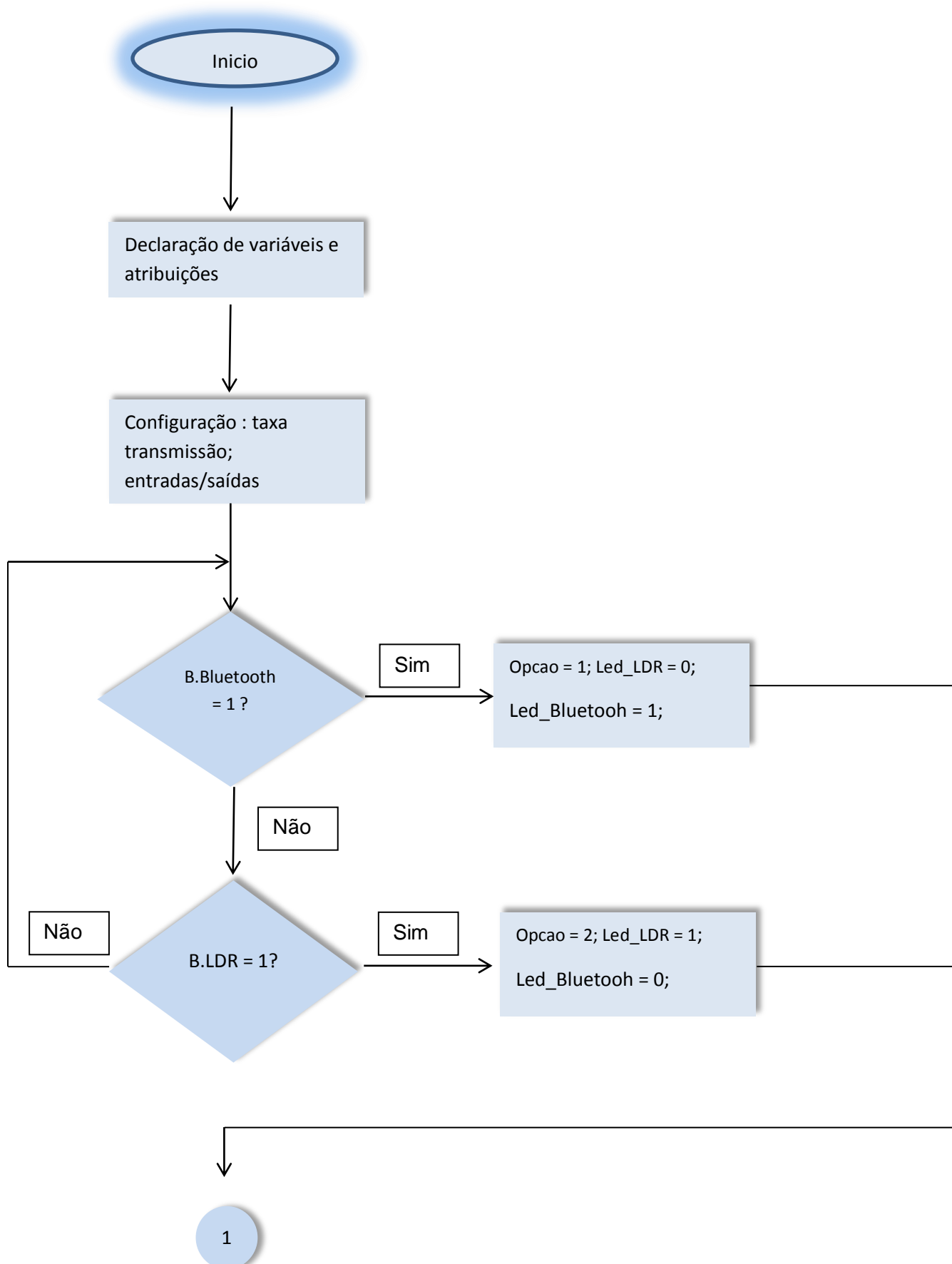
- Malha Aberta:

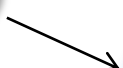
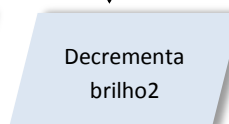
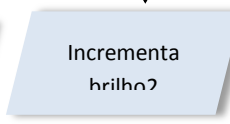
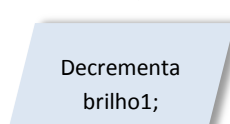
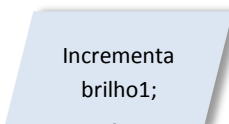
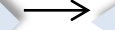
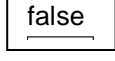
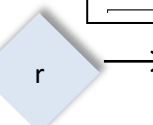
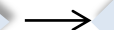
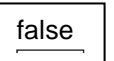
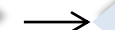
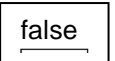
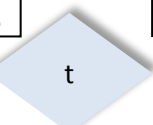
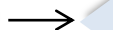
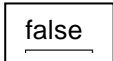
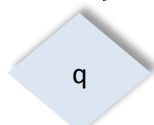
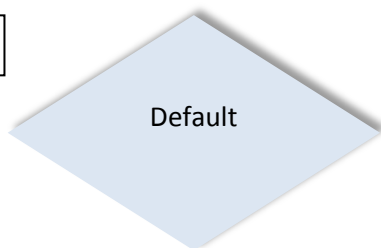
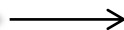
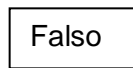
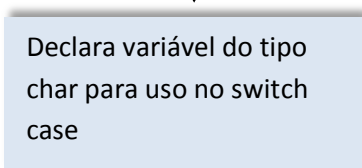
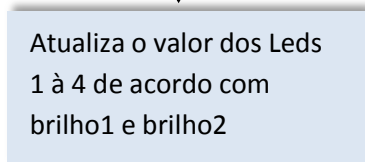
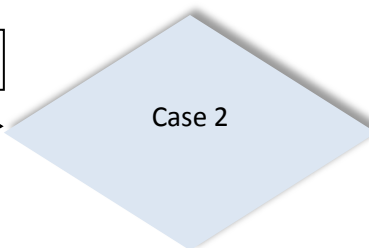
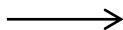
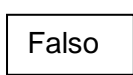
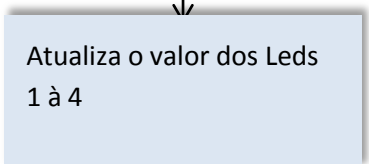
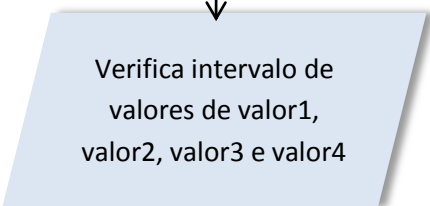
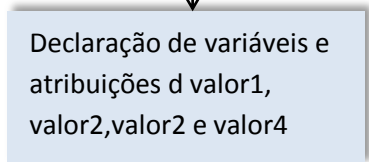
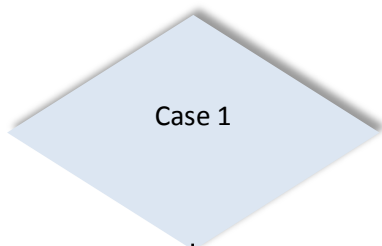


- Malha Fechada:



- Fluxograma





- **Implementação:**

Nesta tarefa foi feito primeiramente o estudo do que se propunha realizar, ou seja, entender como cada elemento do sistema funciona e a partir daí esboçar o seu funcionamento, Dessa forma, fizemos primeiramente simulações no *tinkercard circuits*, as quais foram realizadas de forma gradativa, ou seja, primeiro fizemos o esboço para um led e um ldr, depois para um ldr e dois leds e assim por diante até fazermos as simulações com o módulo bluetooth.

Observe abaixo a descrição de cada componente o exemplo de seu funcionamento:

- Modulo Bluetooth HC05



Modulo Bluetooth: neste projeto será possível fazer o controle de luminosidade via celular (app Mobile), com isso, esse modulo fará essa comunicação entre o arduino e o celular.

Figura 1 - Modulo Bluetooth HC-05

- Arduino UNO



Figura 2 - Arduino

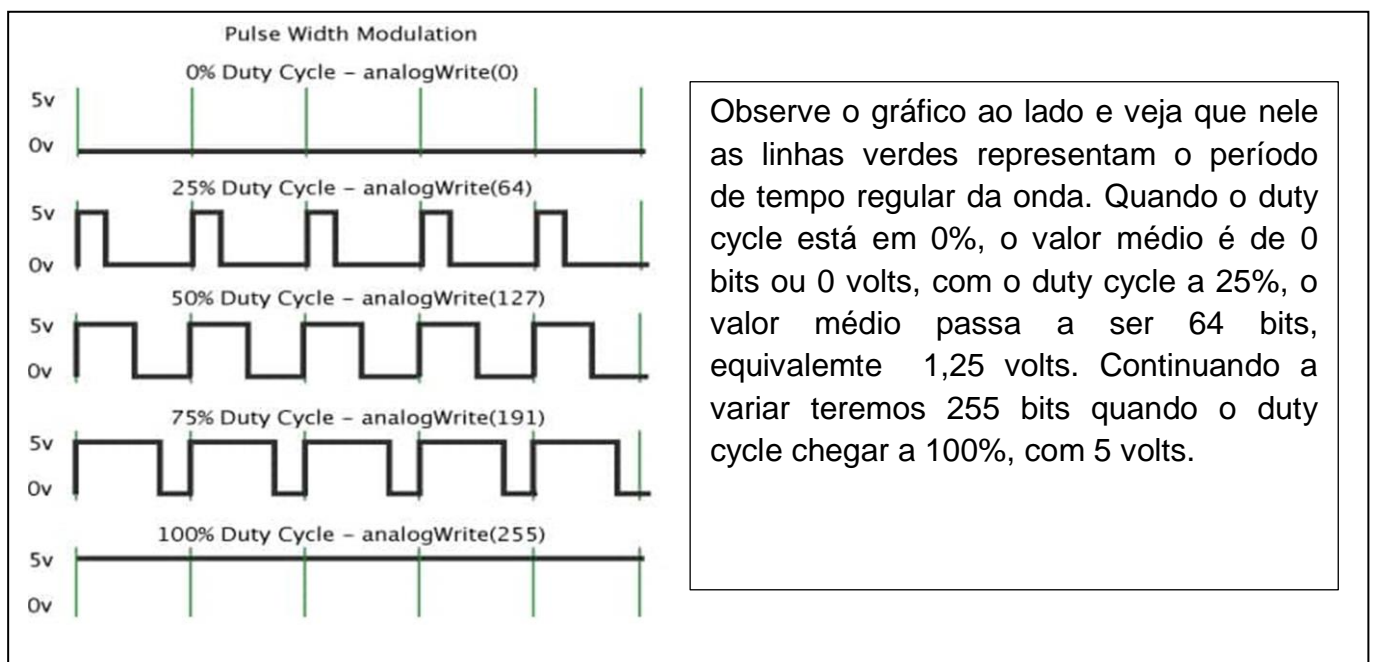
Arduino: é o controlador do projeto, ele é quem fornece a energia de 5 v alimentando os sensores e os leds. O Arduino possui entradas analógicas e saídas digitais com PWM e a programação do projeto é passado do computador para o arduino por meio de uma entrada USB.

-

- Portas pwm arduino Uno

Tais portas funcionam modulando a largura dos pulsos de uma onda quadrada fornecendo seu valor médio, através da variação do tempo em que se mantém a onda em nível lógico alto, mantendo sua frequência constante (técnica chamada de *duty cycle* - *ciclo ativo da forma de onda*).

Os pulsos da onda possuem valor variando de 0 a 255 em escala de bits que equivalem a variação de 0 a 5 em escala de volts. Observe que estas portas fornecem resultados analógicos através de meios digitais.



- Led 5mm de Alto Brilho



Figura 3 - Led

Um led consiste em um diodo emissor de luz que, para funcionar corretamente (emitir luz), deve ser polarizado de forma direta e ser associado a uma resistência para obter uma queda de tensão, a qual limite a corrente que passa por ele. Sendo assim, associado a uma porta pwm, este componente funciona recebendo pulsos com variações de valores entre 0 a 255 bits, que, em escala de volts vai de 0 a 5v. Com isso, podemos observar uma variação na intensidade do brilho deste componente.

- LDR



Figura 4 - LDR

LDR: é um resistor dependente de luz constituído a partir de materiais semicondutores que permitem a variação de sua resistência elétrica em milhares de Ohms. Ele representa o sensor do nosso projeto e quando submetido a luz a sua resistência aumenta e quanto mais escuro, sua resistência diminui, ou seja, ele é quem vai fazer o controle da tensão que vai para os leds, permitindo o controle da intensidade do brilho dos mesmos.

- Código (C/C++)

//Declaração de variáveis

//Declaração pinos LDRs

const int ldr1 = A0; //Entrada analógica 0

const int ldr2 = A1; //Entrada analógica 1

//Declaração pinos leds_alto brilho

const int led1 = 3;

const int led2 = 5;

const int led3 = 6;

const int led4 = 9;

//Declaração pinos LEDs de escolha do tipo de controle

const int ledTestLdr = 12;

const int ledTestBluetooth = 13;

//Declaração dos pinos dos botoes de escolha do tipo de controle

const int botaoLDR = 10;

const int botaoBLUETOOTH = 11;

//Declarando variáveis para lerem os estados dos botões

int estadoBotaoLDR = 0;

int estadoBotaoBLUETOOTH = 0;

**//Declarando variáveis para controle do brilho via
bluetooth**

int brilho1=0;

int brilho2=0;

//Variável para escolha via switch case

int opcao;

void setup() {

//Configurando taxa de transmissao serial

Serial.begin(9600);

//Configurando pinos de entrada e de saída

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

pinMode(ledTestBluetooth, OUTPUT);

pinMode(ledTestLdr, OUTPUT);

pinMode(botaoLDR, INPUT);

pinMode(botaoBLUETOOTH, INPUT);

//Definindo os Leds de alto brilho em estado baixo

analogWrite(led1,LOW);

analogWrite(led2,LOW);

analogWrite(led3,LOW);

analogWrite(led4,LOW);

}// Fim do setup

void loop() {

// Inicio do loop

// Leitura digital (0 ou 1) do estado dos botões

estadoBotaoLDR = digitalRead(botaoLDR);

```
estadoBotaoBLUETOOTH =  
digitalRead(botaoBLUETOOTH);
```

```
// Condição de escolha do controle via bluetooth (Malha  
aberta)  
if((estadoBotaoBLUETOOTH==HIGH)&&(estadoBotaoLDR  
==LOW)){  
    opcao= 2;  
    brilho1=0;  
    brilho2=0;  
    digitalWrite(ledTestLdr, LOW);  
    digitalWrite(ledTestBluetooth, HIGH);  
    }delay(500);// Tempo necessário para o sistema fazer a  
leitura do botão
```

```
// Condição de escolha do controle via LDR (Malha  
fechada)  
if((estadoBotaoLDR==HIGH)&&(estadoBotaoBLUETOOTH  
==LOW)){  
    opcao = 1;  
    digitalWrite(ledTestBluetooth, LOW);  
    digitalWrite(ledTestLdr, HIGH);  
    }delay(500);// Tempo necessário para o sistema fazer a  
leitura do botão
```

```
switch (opcao){  
    case 1: //comando via controle de LDR's (Malha fechada  
    {  
        int valor1 = analogRead(ldr1);  
        int valor2 = map(valor1, 800, 60, 0, 255);  
        int valor3 = analogRead(ldr2);  
        int valor4 = map(valor3, 650, 40, 0, 255);  
        if(valor2<0)  
        {  
            valor2=0;  
        }  
    }
```

```
if(valor2>255){
    valor2=255;
}
if(valor4<0){
    valor4=0;
}
if(valor4>255){
    valor4=255;
}
analogWrite(led1,valor2);
analogWrite(led2,valor2);
analogWrite(led3,valor4);
analogWrite(led4,valor4);
```

```
break;
```

```
}//fim do case 1
```

```
case 2: // Comando via controle bluetooth - Malha
```

```
Aberta
```

```
{
    analogWrite(led1,brilho1);
    analogWrite(led2,brilho1);
    analogWrite(led3,brilho2);
    analogWrite(led4,brilho2);
    char var = Serial.read();
    switch (var) {
        case 'q': // Incrementa o brilho dos leds 1 e 2
        {

            brilho1 = brilho1 + 10;
            if(brilho1>250){
                brilho1=250;
            }


```

```
break;
```

```
}
```

```
case 't': // Decrementa o brilho dos leds 1 e 2
```

```

{
    if(brilho1 >0)
    {
        brilho1 = brilho1 - 10;

    }else
    {
        brilho1 = 0;
    }

    break;
}
case 'w': // Incrementa o brilho dos leds 3 e 4
{

    brilho2 = brilho2 + 10;
    if(brilho2>250)
    {
        brilho2=250;
    }

    break;
}

case 'r': // Decrementa o brilho dos leds 3 e 4
{
    if(brilho2 >0)
    {
        brilho2 = brilho2 - 10;

    }else
    {
        brilho2 = 0;
    }
    break;
}

```

default:

break;

}// Fim swicth brilho

break;

}//Fim do case2

default:

break;

}//fim switch geral

}//fim loop

- **Conclusão:**

O Dimmer vem se tornando um dispositivo cada vez mais presente no nosso cotidiano, com este projeto pôde-se concluir que o mesmo já é bastante acessível, podendo gerar uma elevada economia em indústrias e residências; ainda permitindo que o ambiente possa ser mais confortável para as pessoas presentes nele. Além da relativa facilidade de utilização, a integração dele com outros dispositivos é bastante prática, o que facilita a aceitação de mercado.

Uma das maiores dificuldades encontradas é a vinculação do dispositivo com o ambiente no qual ele está inserido, onde será necessária uma análise dos pontos de sol, pontos de sombra e de qual será o tipo desse ambiente (sala de aula, escritório, cozinha, ...), o que gera alteração em sua configuração. Em decorrência dessas dificuldades, a taxa de variação da resposta do LDR varia de ambiente para ambiente, gerando uma leitura diferente, ou seja, cada um desses ambientes deve ser tratado de forma diferente.

Ainda assim, o projeto se mostrou viável e, com o desenvolvimento necessário, pode atingir níveis de leitura ainda mais precisos e com um maior controle de variação.