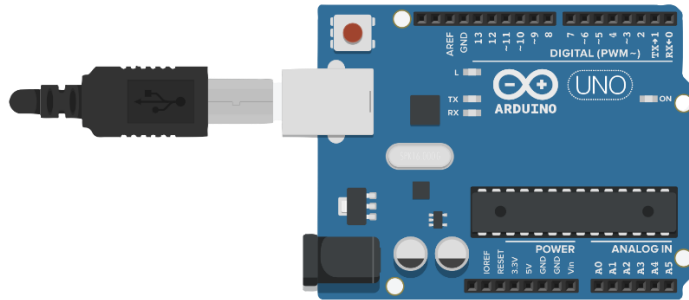


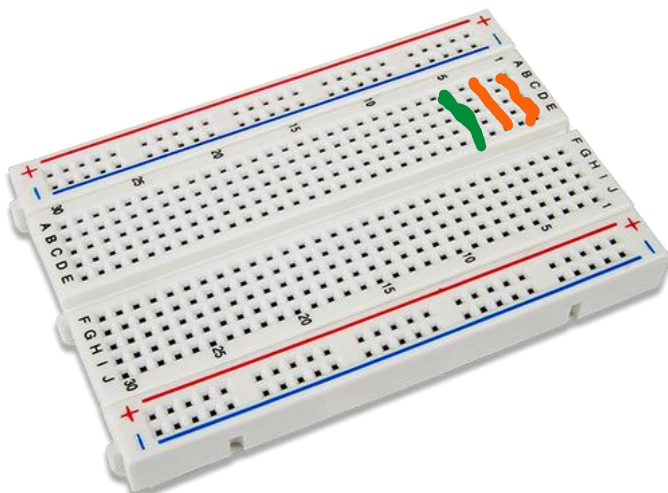
# Arduino



O Arduino é uma plataforma de prototipagem open-source. A placa de um Arduino consiste em um microcontrolador (que é um microcomputador de um único circuito que processa o código desenvolvido por nós no software chamado **IDE**), com componentes complementares para facilitar a programação e incorporação para outros circuitos. A versão mais popular e mais vendida é o Arduino UNO.

A **Arduino IDE** é uma ferramenta de desenvolvimento multiplataforma escrita em Java, com uma interface gráfica simples de usar, mas que também possui flexibilidade para os usuários mais experientes. É um programa que reúne características e ferramentas para agilizar o processo de desenvolvimento de software.

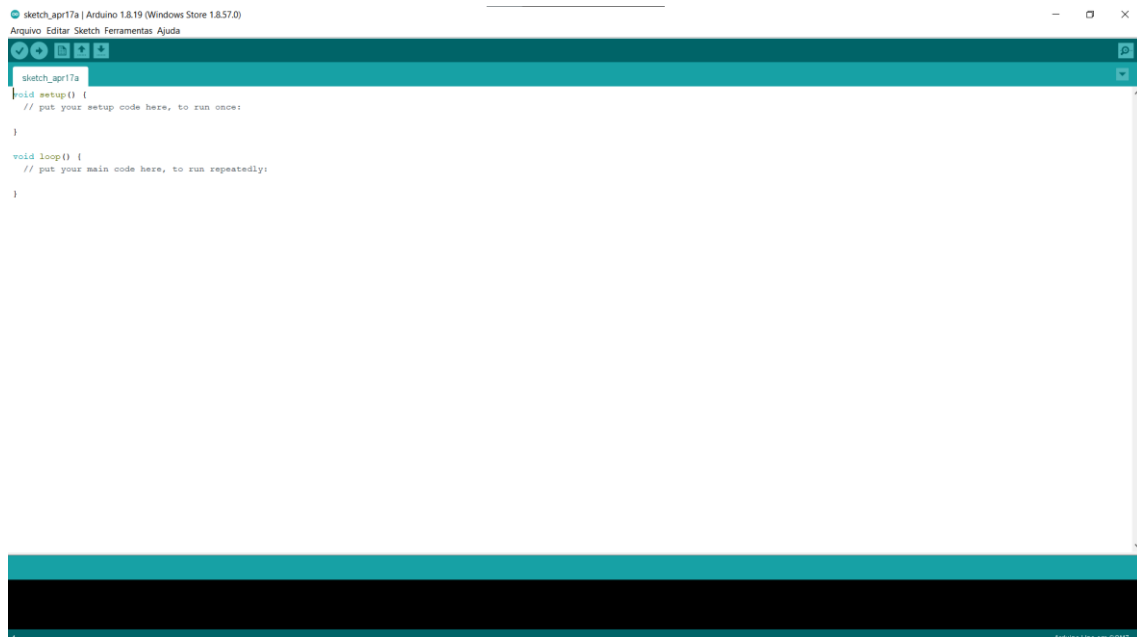
Podemos encarar o Arduino como o processador (cérebro), mas suas entradas são limitadas, então utilizamos uma coisa chamada protoboard:



Podemos encarar o protoboard como uma extensão do Arduino. Nele vamos montar nossos circuitos eletrônicos que serão comandados pelo Arduino. Por mais que cada um desses pequenos furos aparentam ser isolados eles estão interligados por dentro. Como pode ver na linha laranja. Todas essas estradas na vertical estão internamente interligadas, mas em sua horizontal permanecem isoladas e únicas. Por exemplo a as laranjas estão interligadas, mas a linha verde

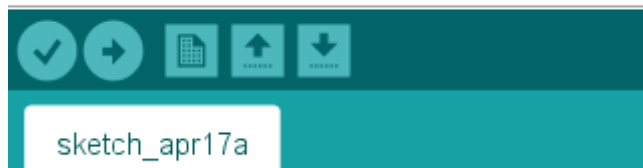
é outra completamente diferente. Isso permite que nós possamos montar circuitos eletrônicos, pois cada linha na vertical é uma linha única com entradas diferentes.


## Software:

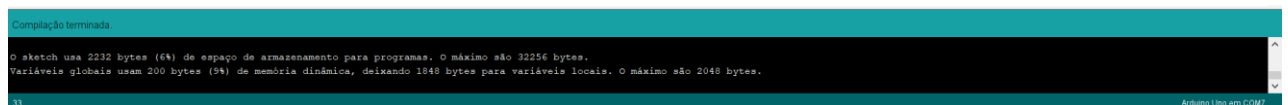


Esta é a cara do software do Arduino IDE, é aqui onde iremos escrever nossa programação antes de passar para o Arduino.

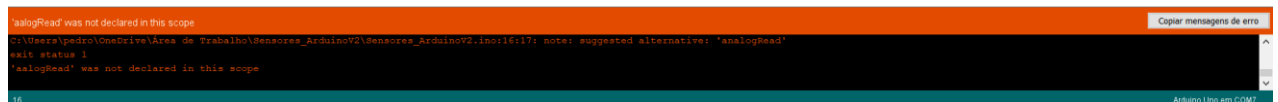
Arquivo Editar Sketch Ferramentas Ajuda




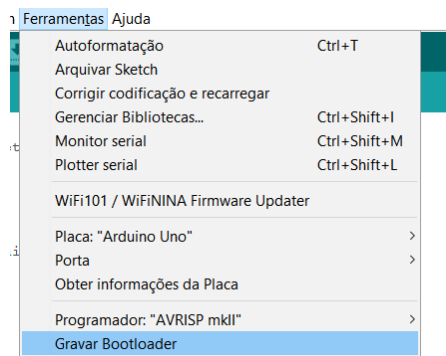
Aqui  neste ícone nos podemos checar se tem algum erro em nossa programação, erro de sintaxe, após escrevermos nosso código clicamos aqui e caso ele esteja tudo certo, o inferior da tela ficara assim:



Caso algo de errado ele apontará o erro:

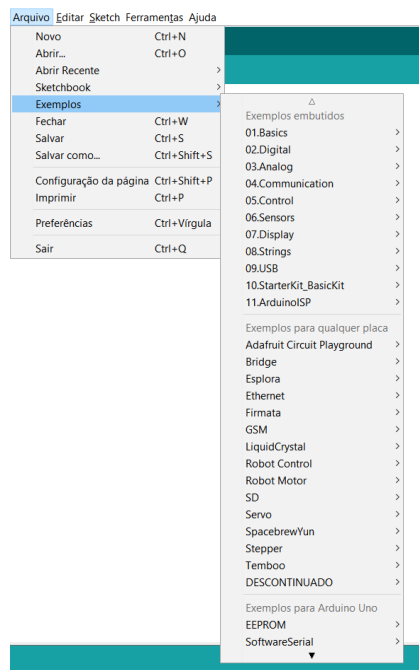


Quando tudo estiver certo clicaremos aqui . Aqui nós iremos encaminhar a programação que nós fizemos para dentro do Arduino (lembrando que o Arduino deve estar conectando em uma porta USB do PC. Para saber se o Arduino está conectado podemos clicamos em:



Ferramentas, após isso em “Placa” selecionamos Arduino UNO e clicamos em “porta” para verificar em qual nosso Arduino se encontra. OBS: Normalmente só de conectarmos o Arduino ele já faz isso automaticamente.

OBS:



Aqui nós podemos, ver alguns códigos salvos para realizar alguns testes com o Arduino e brincar um pouco com ele.

## Esquema de Ligação:

### LM35 – sensor de temperatura

Figura 1: Sensor LM35

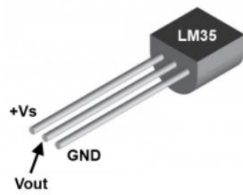
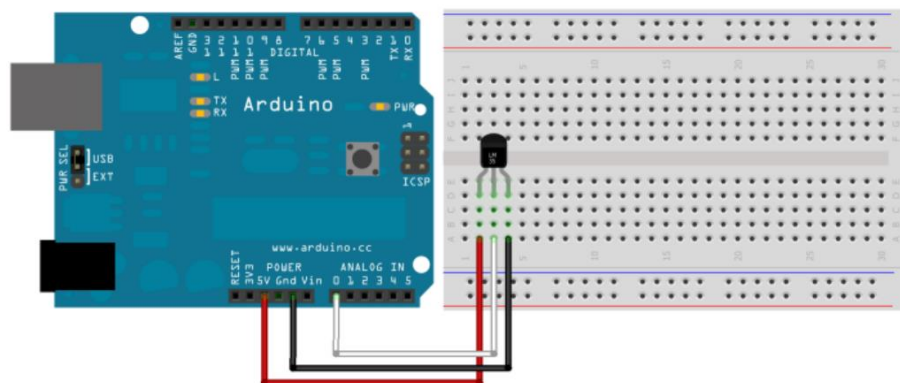


Figura 2: Ligação LM35 com o Arduino

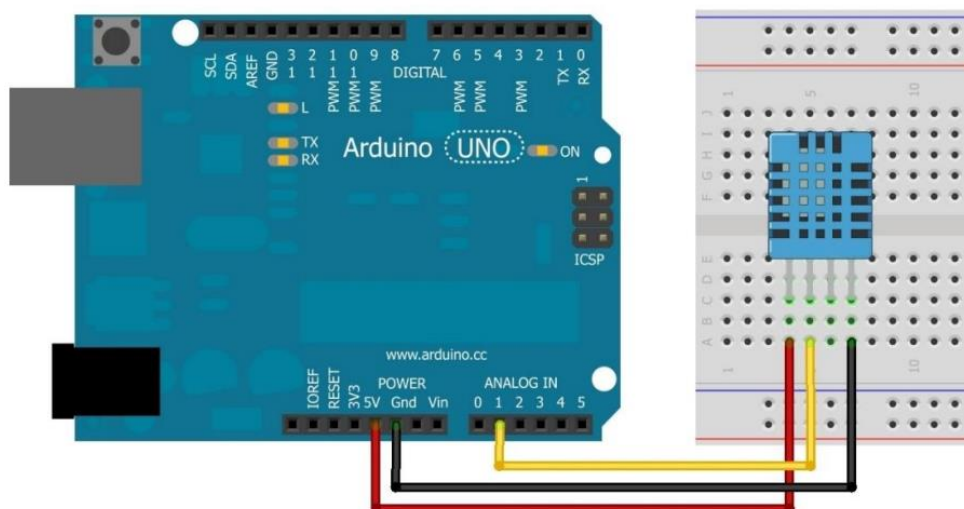


### DHT11 – sensor de umidade e temperatura

Figura 3: Sensor DHT11



Figura 4: Ligação DHT11 com o Arduino



## LDR – sensor de luminosidade

### LDR5

Figura 6: Sensor LDR5mm

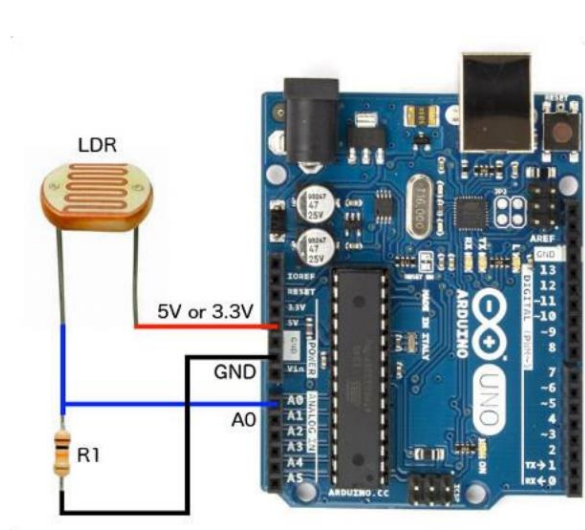
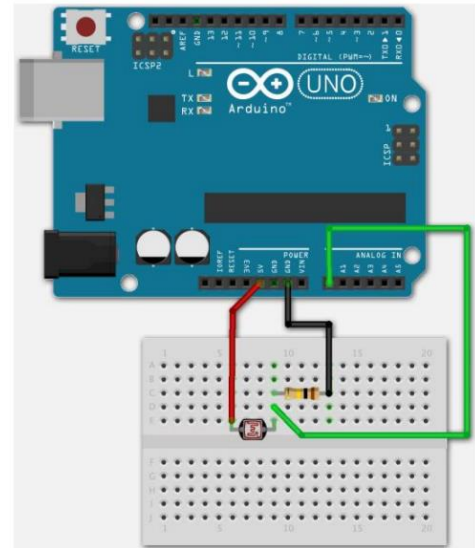


Figura 5: Ligação LDR5 com Arduino



## TRC5000 – sensor de proximidade

Figura 7: Sensor TCRT5000



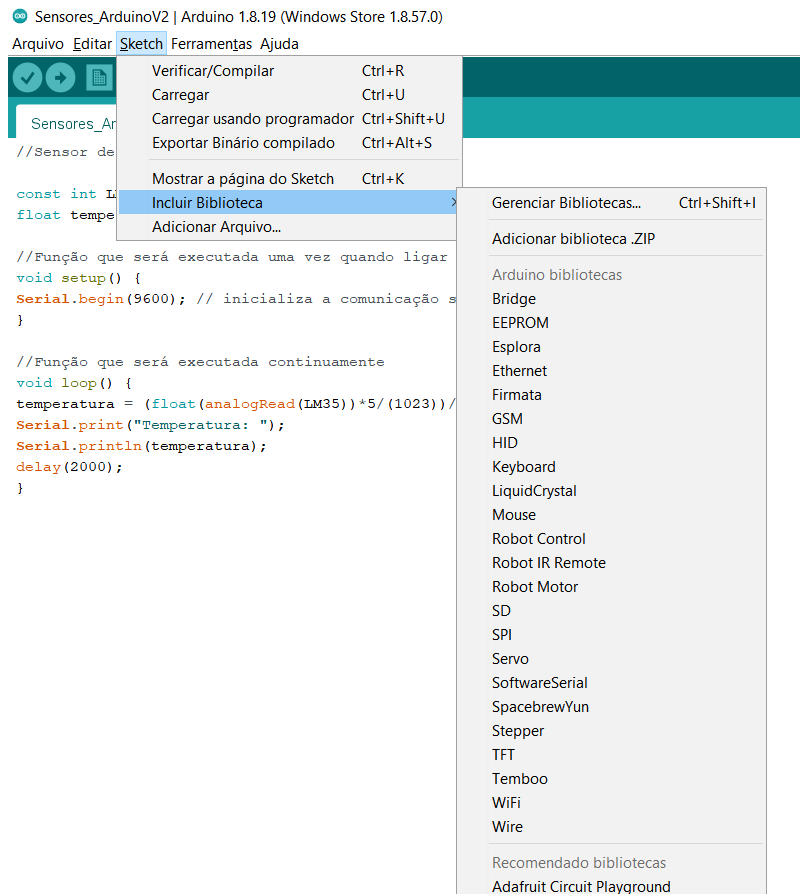
Figura 9: Ligação TCRT5000 com Arduino



## Bibliotecas:

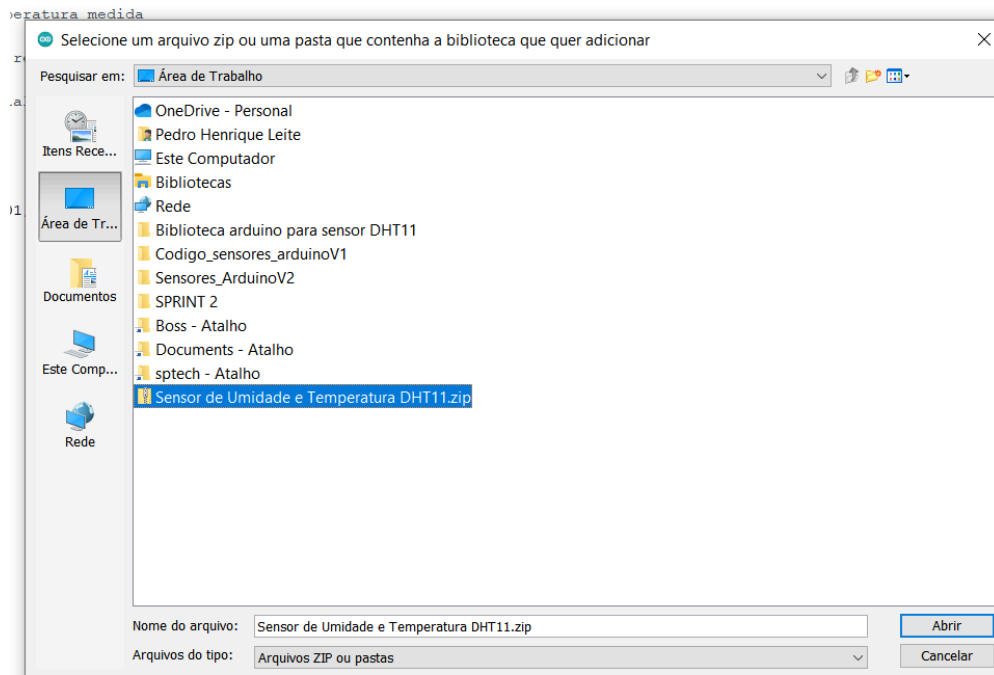
Alguns códigos sensores e outros componentes eletrônicos precisam de algo a mais para funcionarem, por isso existem as bibliotecas. Elas são como extensões do Arduino, códigos a mais que instalamos para conseguirmos comandar determinados sensores. Por exemplo para conseguir fazer o Sensor de Umidade e Temperatura DHT11 funcionar, precisamos adicionar uma nova biblioteca.

Para adicionar essas bibliotecas (extensões) seguimos esses passos:



Clicamos em “Sketch”, depois clicamos em “incluir biblioteca” e por fim em “Adicionar biblioteca .ZIP”.

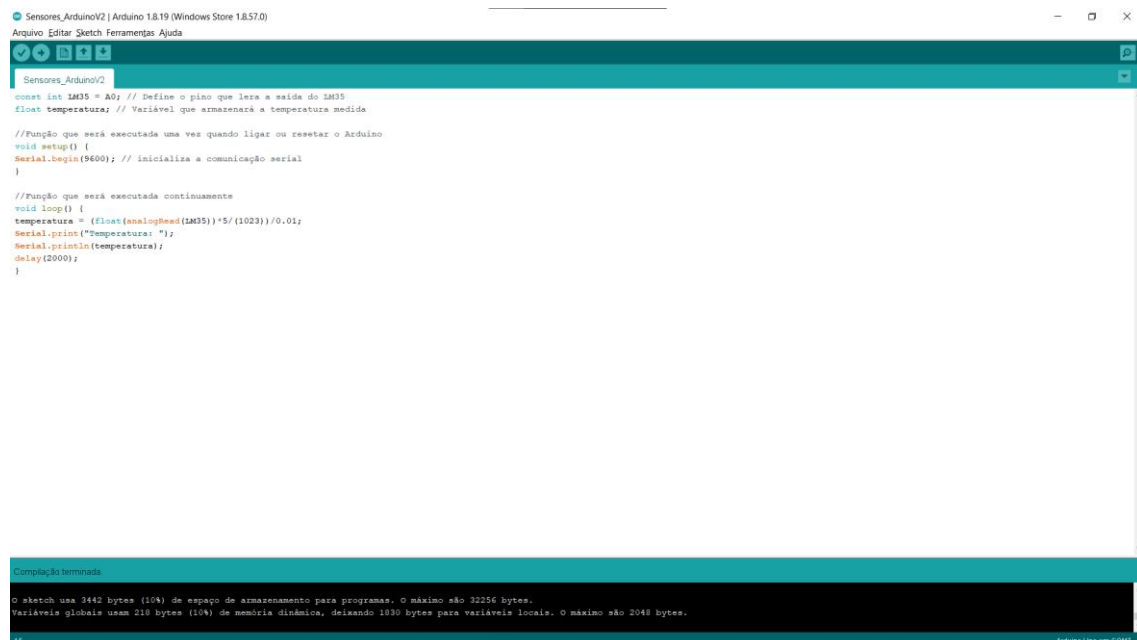
Após isso vai abrir o explorador ai escolhemos a pasta .ZIP da biblioteca que queremos adicionar e clicamos em “abrir”:



Nesse momento a biblioteca já foi importada para sua IDE.

Agora você pode usar os seguintes códigos para os sensores funcionarem. Basta achar o código do sensor que você deseja utilizar, copiar ele e colar no Arduino IDE:

Exemplo:



### Sensor de temperatura LM35:

```
const int LM35 = A0; // Define o pino que lera a saída do LM35
float temperatura; // Variável que armazenará a temperatura medida

//Função que será executada uma vez quando ligar ou resetar o Arduino
void setup() {
  Serial.begin(9600); // inicializa a comunicação serial
}

//Função que será executada continuamente
void loop() {
  temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
  Serial.print("Temperatura: ");
  Serial.println(temperatura);
  delay(2000);
}
```

---



Sensor de Umidade e Temperatura DHT11: (OBS: ESSE SENSOR PRECISA DE UMA BIBLIOTECA DIFERENTE PARA FUNCIONAR)

```
#include "dht.h" //INCLUSÃO DE BIBLIOTECA
```

```
const int pinoDHT11 = A2; //PINO ANALÓGICO UTILIZADO PELO DHT11
```

```
dht DHT; //VARIÁVEL DO TIPO DHT
```

```
void setup(){
```

```
  Serial.begin(9600); //INICIALIZA A SERIAL
```

```
  delay(2000); //INTERVALO DE 2 SEGUNDO ANTES DE INICIAR
```

```
}
```

```
void loop(){
```

```
  DHT.read11(pinoDHT11); //LÊ AS INFORMAÇÕES DO SENSOR
```

```
  Serial.print("Umidade: "); //IMPRIME O TEXTO NA SERIAL
```

```
  Serial.print(DHT.humidity); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO
```

```
  Serial.print("%"); //ESCREVE O TEXTO EM SEGUIDA
```

```
  Serial.print(" / Temperatura: "); //IMPRIME O TEXTO NA SERIAL
```

```
  Serial.print(DHT.temperature, 0); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO E  
  REMOVE A PARTE DECIMAL
```

```
  Serial.println("*C"); //IMPRIME O TEXTO NA SERIAL
```

```
  delay(2000); //INTERVALO DE 2 SEGUNDOS * NÃO DIMINUIR ESSE VALOR
```

```
}
```

---

### Sensor de bloqueio TCRT5000:

```
int pinoLed = 8; //PINO DIGITAL UTILIZADO PELO LED
```

```
int pinoSensor = 7; //PINO DIGITAL UTILIZADO PELO SENSOR
```

```
void setup(){
```

```
    pinMode(pinoSensor, INPUT); //DEFINE O PINO COMO ENTRADA
```

```
    pinMode(pinoLed, OUTPUT); //DEFINE O PINO COMO SAÍDA
```

```
    digitalWrite(pinoLed, LOW); //LED INICIA DESLIGADO
```

```
}
```

```
void loop(){
```

```
    if (digitalRead(pinoSensor) == LOW){ //SE A LEITURA DO PINO FOR IGUAL A LOW, FAZ
```

```
        digitalWrite(pinoLed, HIGH); //ACENDE O LED
```

```
    }else{//SENÃO, FAZ
```

```
        digitalWrite(pinoLed, LOW); //APAGA O LED
```

```
    }
```

```
}
```

---

## Sensor de luminosidade LDR:

```
const int analogInPin = A0;           // Pino analógico que o sensor está conectado
const int digitalPin = 2;

int sensorValue = 0;                  // valor que será lido do sensor

int setpoint = 200;                   // Este valor deverá ser alterado de acordo com a intensidade de
                                     // luz desejada.

void setup() {
    Serial.begin(9600);                // inicializa a comunicação serial com a taxa de 9600 bps
    pinMode(digitalPin, OUTPUT);
}

void loop() {

    sensorValue = analogRead(analogInPin); // Faz a leitura do pino analógico

    Serial.print("sensor = ");         // Imprime o resultado no monitor serial
    Serial.println(sensorValue);

    if(sensorValue < setpoint){

        digitalWrite(digitalPin, HIGH);

    } else {

        digitalWrite(digitalPin, LOW);

    }

    delay(2);                          // Aguarda 2 milissegundos para estabilizar o conversor ADC
}
```