

---

---

# Protótipo de estação meteorológica para usinas fotovoltaicas

— Trabalho de Seminários de  
Engenharia de Produção IV —

---

---

# O que são estações meteorológicas?

Uma estação meteorológica é um conjunto de sensores e equipamentos que tem a capacidade de medir, gravar e recolher dados sobre o tempo.

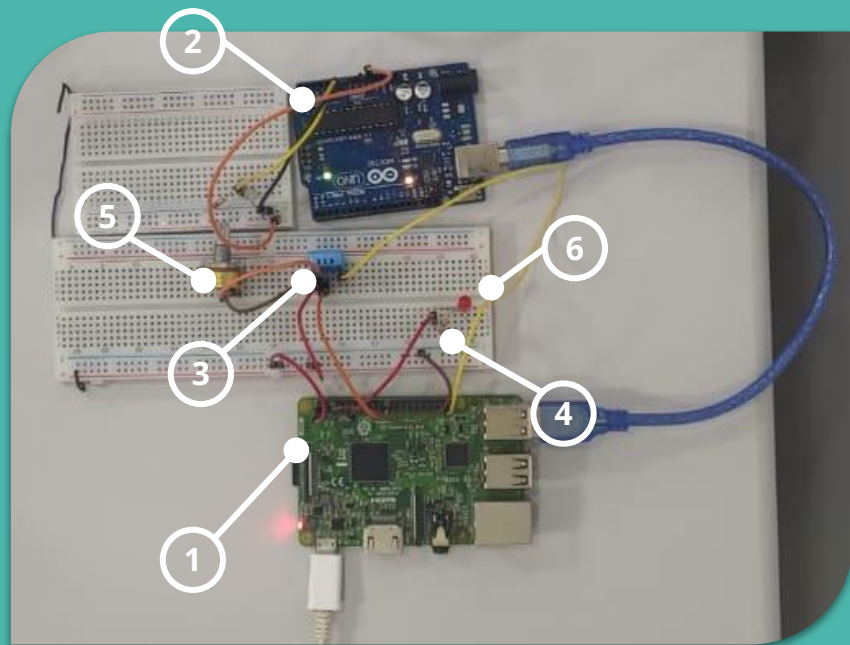
São essenciais para a gestão da energia produzida em uma usina solar.



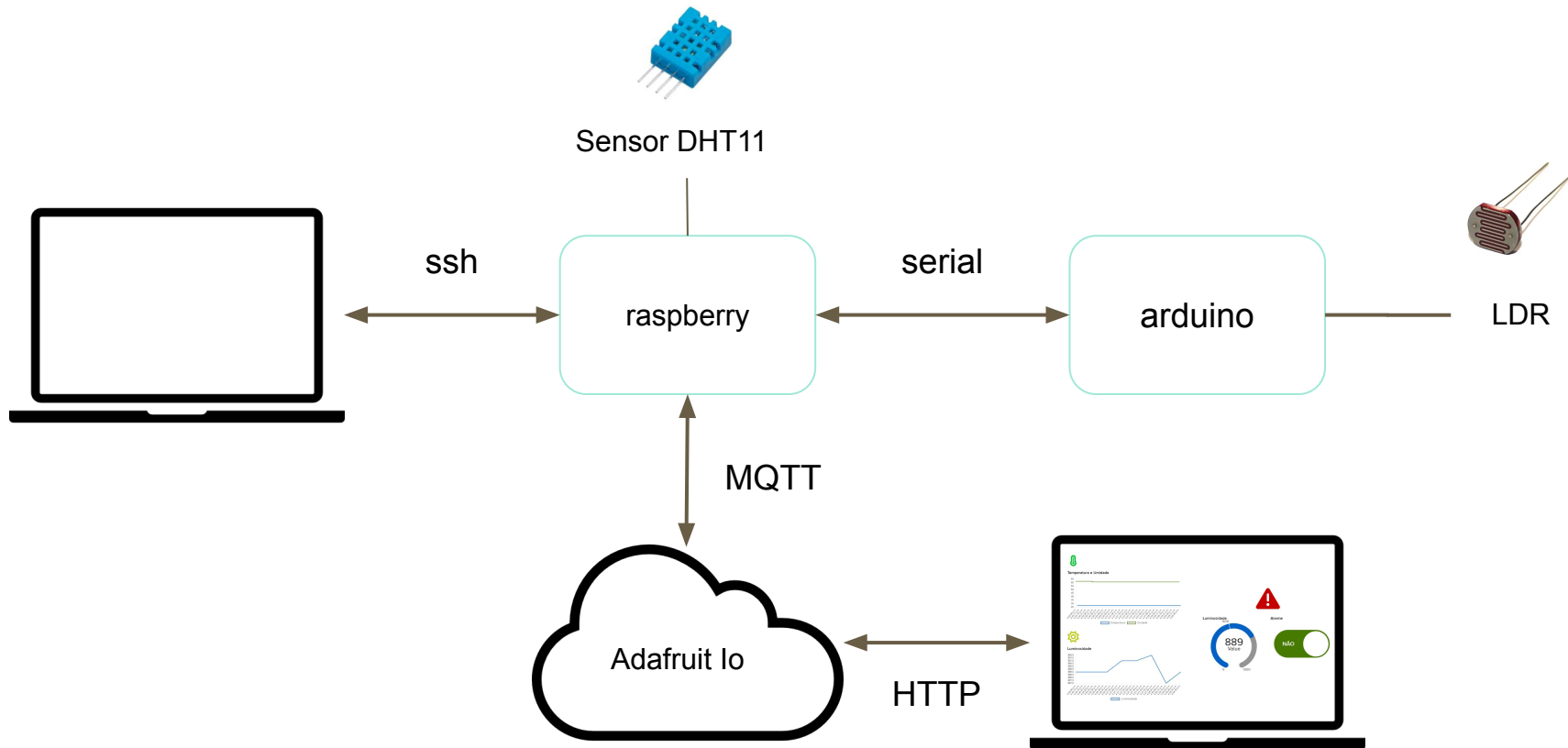
# Protótipo de estação metereológica

Protótipo de aquisição e envio de dados:

1. Raspberry Pi 3;
2. Arduino UNO;
3. Sensor de umidade e temperatura DHT11;
4. Sensor de luminosidade LDR;
5. Resistor de 4,7K $\Omega$  e de 10K $\Omega$
6. LED vermelho



# Diagrama de comunicações



# Código no Arduino IDE

```
#define pin_ldr A0
#include <Servo.h>

Servo myservo;

int pos = 0;    // Variavel para armazenar a posição do servo

void setup() {

  pinMode(pin_ldr, INPUT);    // Definir o pin do LDR A0 como entrada analógica

  Serial.begin(9600);    // Informar taxa de transmissão de bits

  myservo.attach(9);    // Fixar o objeto servo a porta 9
}

void loop() {

  if (Serial.available()) {    // Conferir se a comunicação serial está disponível

    char lido = char(Serial.read());    //Realizar leitura na comunicação serial

    if (lido == '1')    //Impor condição para garantir que é o momento certo de enviar informação
    {
      Serial.println(analogRead(pin_ldr));    //Fazer leitura de luminosidade no LDR e enviar por comunicação serial

      delay(100);}}}

```

# Código na Raspberry Pi 3

```
# Programa para detectar temperatura e umidade

#Importar bibliotecas
import Adafruit_DHT
import RPi.GPIO as GPIO
import time
import serial

#Definições iniciais das portas e comunicação serial
sensor = Adafruit_DHT.DHT11
led=11
arduino = serial.Serial('/dev/ttyACM0',9600)
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(led, GPIO.OUT)
pino_DHT = 25
GPIO.output(led,0)

# Conectar ao Dashboard Adafruit
ADAFRUIT_IO_USERNAME = "Caian_Jesus"
ADAFRUIT_IO_KEY = "aio_hrPU04BNGGe6bqelRwPY4ahrMzXe"
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
```

# Código na Raspberry Pi 3

```
print("Lendo os Valores de Temperatura e Umidade")

while(1):
    #ler dados de temperatura e umidade na Raspberry
    umid,temp = Adafruit_DHT.read_retry(sensor, pino_DHT)

    if umid is not None and temp is not None: #conferir se foi medido corretamente
        print("Temperatura = ",temp, "Umidade = ",umid)
        #enviar dados para Adafruit.io

        campo1 = aio.feeds('umidade')
        aio.send_data(campo1.key, umid)

        campo2 = aio.feeds('temperatura')
        aio.send_data(campo2.key, round(temp,2))

    else:
        print("Falha ao ler dados do DHT11")
```

# Código na Raspberry Pi 3

```
#enviar comando para arduino
arduino.write(b'1')

#ler dados arduino
lumi = int(arduino.readline().decode('utf-8'))
print('Luminosidade = ',lumi)

#enviar dados para Adafruit.io
campo3 = aio.feeds('luminosidade')
aio.send_data(campo3.key, round(lumi,2))

#retirar comando para arduino
arduino.write(b'0')
print("Aguardando próxima leitura")
```



# Código na Raspberry Pi 3

```
#lógica para acionar alarme no dashboard e ascender led caso luminosidade esteja baixa
estado= int(aio.receive("alarme").value)

if (lumi<600 or estado == 1):
    GPIO.output(led, 1)
    aio.send("texto","CONFIRA O ESTADO DO ALARME!")
    estado=1
    aio.send("alarme",estado)
elif(lumi<600 and estado ==0):
    GPIO.output(led, 1)
    aio.send("texto","CONFIRA O ESTADO DO ALARME!")
    estado=1
    aio.send("alarme",estado)
else:
    GPIO.output(led, 0)
    estado=0
    aio.send("alarme",estado)
    aio.send("texto","Tudo em ordem")


time.sleep(15)
```








Os códigos serão disponibilizados em anexo junto com a apresentação

# Campos criados no Adafruit para receber os valores das variáveis enviadas pela Raspberry Pi3

+ New Feed

+ New Group

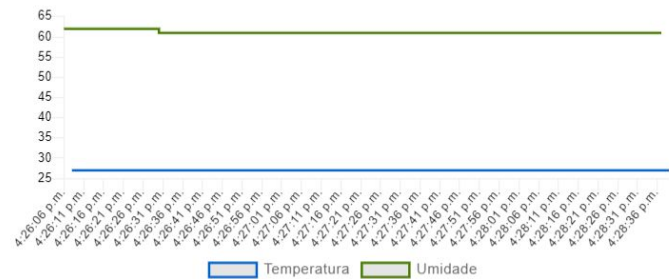


Default				 
Feed Name	Key	Last value	Recorded	
<input type="checkbox"/> <a href="#">Alarme</a>	alarme	0	1 minute ago	
<input type="checkbox"/> <a href="#">Luminosidade</a>	luminosidade	832	15 days ago	
<input type="checkbox"/> <a href="#">Temperatura</a>	temperatura	27.0	15 days ago	
<input type="checkbox"/> <a href="#">Texto</a>	texto	Tudo em ordem	15 days ago	
<input type="checkbox"/> <a href="#">Umidade</a>	umidade	62.0	15 days ago	

# Dashboard para exibir os campos criados anteriormente



Temperatura e Umidade



Luminosidade



Alarme

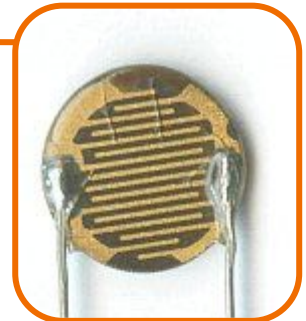
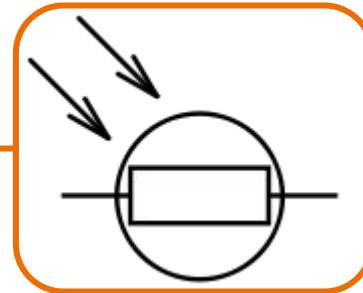
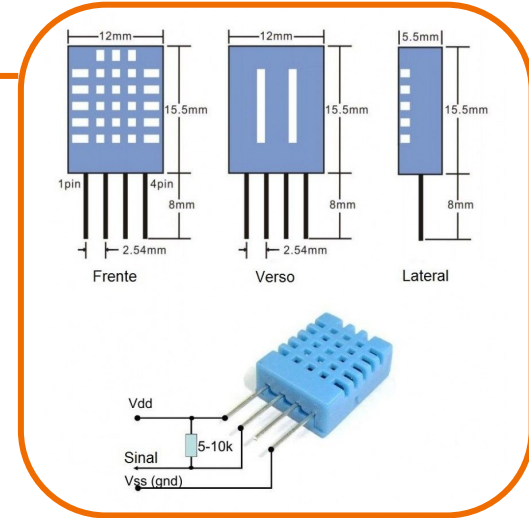


# Protocolos de comunicação usados:

1. SSH (Secure Socket Shell): protocolo específico de segurança de troca de arquivos entre cliente e servidor de internet, usando criptografia;
2. Serial: consiste na transmissão de dados em SÉRIE, ou seja, UM bit de cada vez e depois de 8 pulsos de clock monta-se o primeiro byte;
3. MQTT (Message Queuing Telemetry Transport): é um protocolo de transporte de mensagens de formato Cliente/Servidor, que possibilita a comunicação entre máquinas (Machine to Machine – M2M) e é amplamente usado para conectividade de IoT (Internet of Things).
4. HTTP (Hypertext Transfer Protocol): é um protocolo de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Ele é a base para a comunicação de dados da World Wide Web.

# Sensores usados:

1. DHT11: O DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, muito usado para projetos com Arduino.
2. LDR (Light Dependent Resistor): é um componente eletrônico passivo do tipo resistor variável, mais especificamente, é um resistor cuja resistência varia conforme a intensidade da luz (iluminamento) que incide sobre ele. Tipicamente, à medida que a intensidade da luz aumenta, a sua resistência diminui.



# Custo de produção

Componente	Valor
Raspberry Pi 3	R\$659.00
Arduino UNO	R\$94.90
Sensor de umidade e temperatura DHT11	R\$14.16
Sensor de luminosidade LDR	R\$1.85
Resistor de 4,7K $\Omega$ e de 10K $\Omega$	R\$0.13
LED vermelho	R\$0.24
Total	R\$770.28