

# *Programação IoT com Python*

AULA #01 #02 E #03

Prof. Francisco  
**Calaça** ●●●●●



# Francisco Calaça

- Mestre em Ciência da Computação
- Cursa Doutorado em Ciência da Computação
- Professor
- Empresário
- Maker e Hacker
- Fabrica Cerveja
- Youtuber



# Agenda

- Ambiente de Trabalho
- Tópicos de programação embarcada
- Implementação de exemplos
- Obtenção de dados de sensores
- Acionamentos de atuadores.

# Ambiente de trabalho sugerido

- Equipamentos
  - Fonte
  - Protoboard
  - Ferro de Solda
  - Cabos
  - Arduino/Outras placas
- Computador
- Componentes eletrônicos
- Multímetro
- Mini retífica
- Chaves
- Alicates

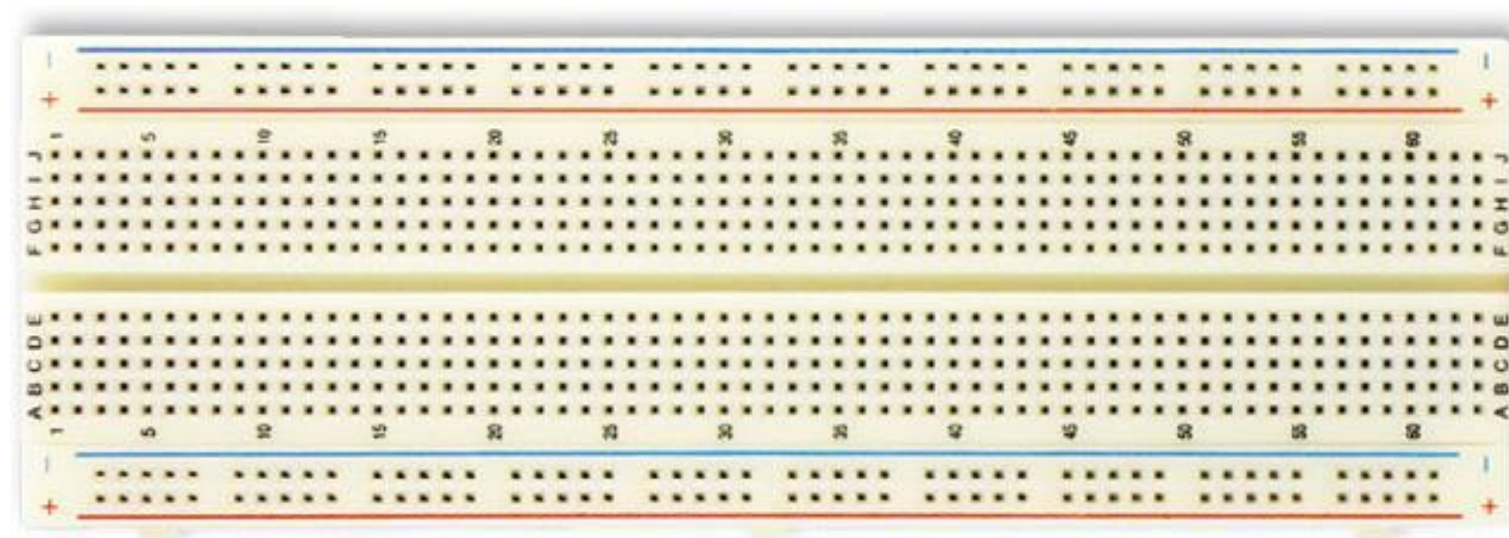
# Ambiente de Trabalho

- Fonte



# Ambiente de Trabalho

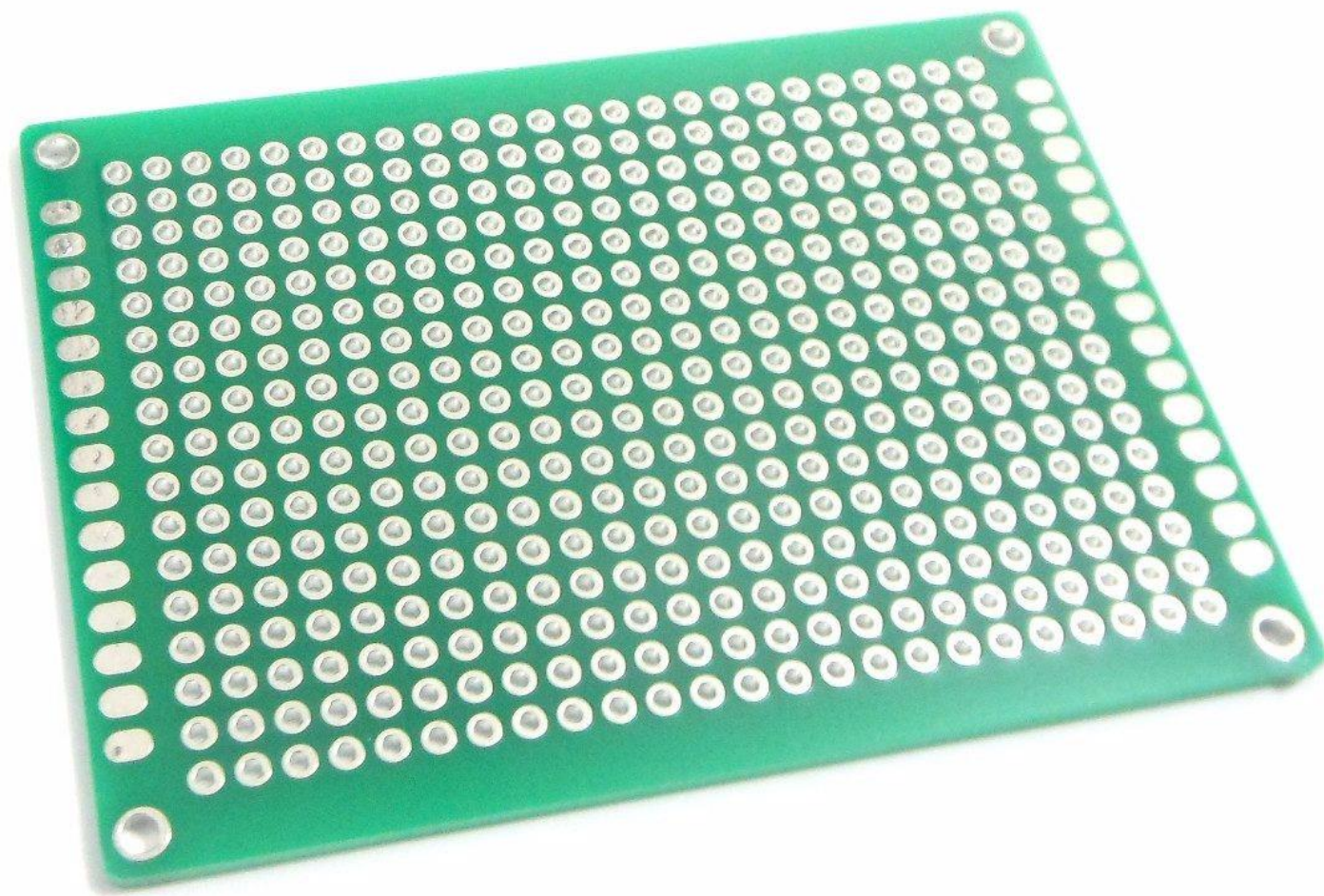
- Protoboard





# Ambiente de Trabalho

- Placa perfurada



# Ambiente de Trabalho

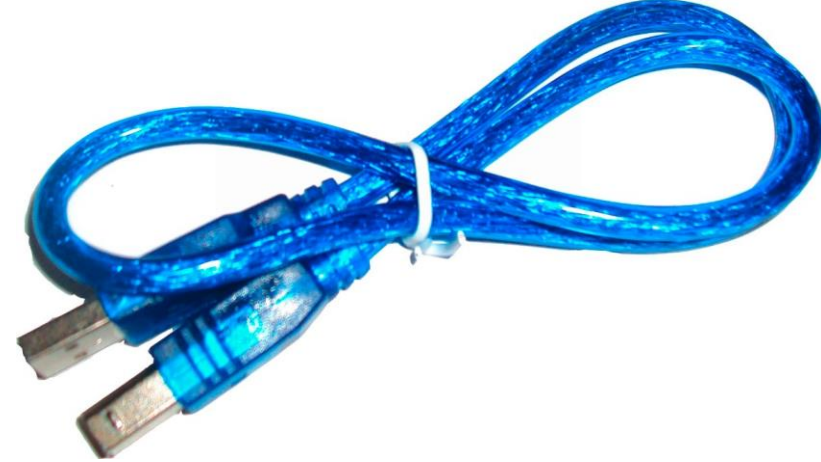
- Ferro de Solda
- 30W ou 40W





# Ambiente de Trabalho

- Cabos



# Ambiente de Trabalho

- Arduino e Outras placas

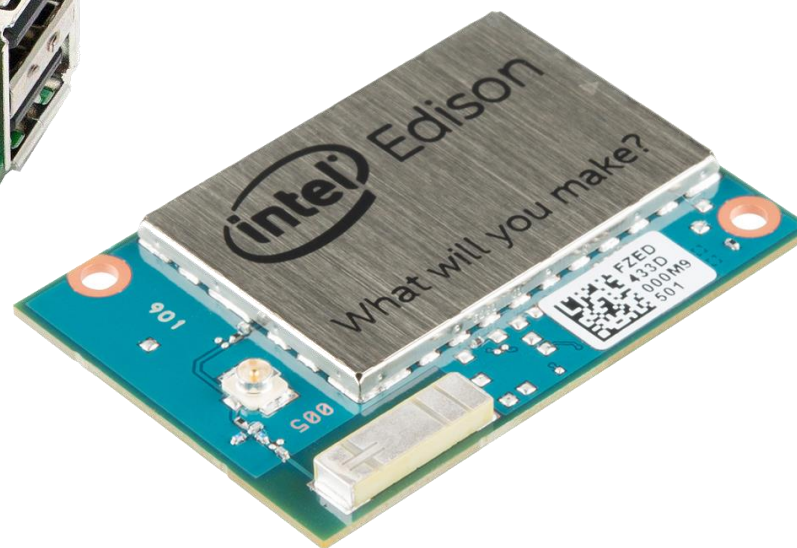
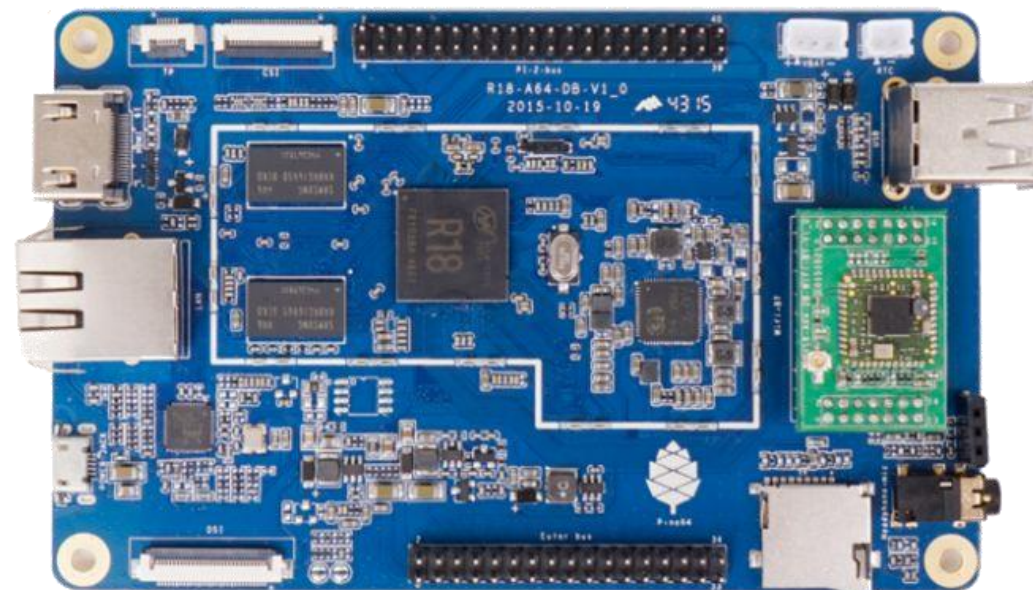
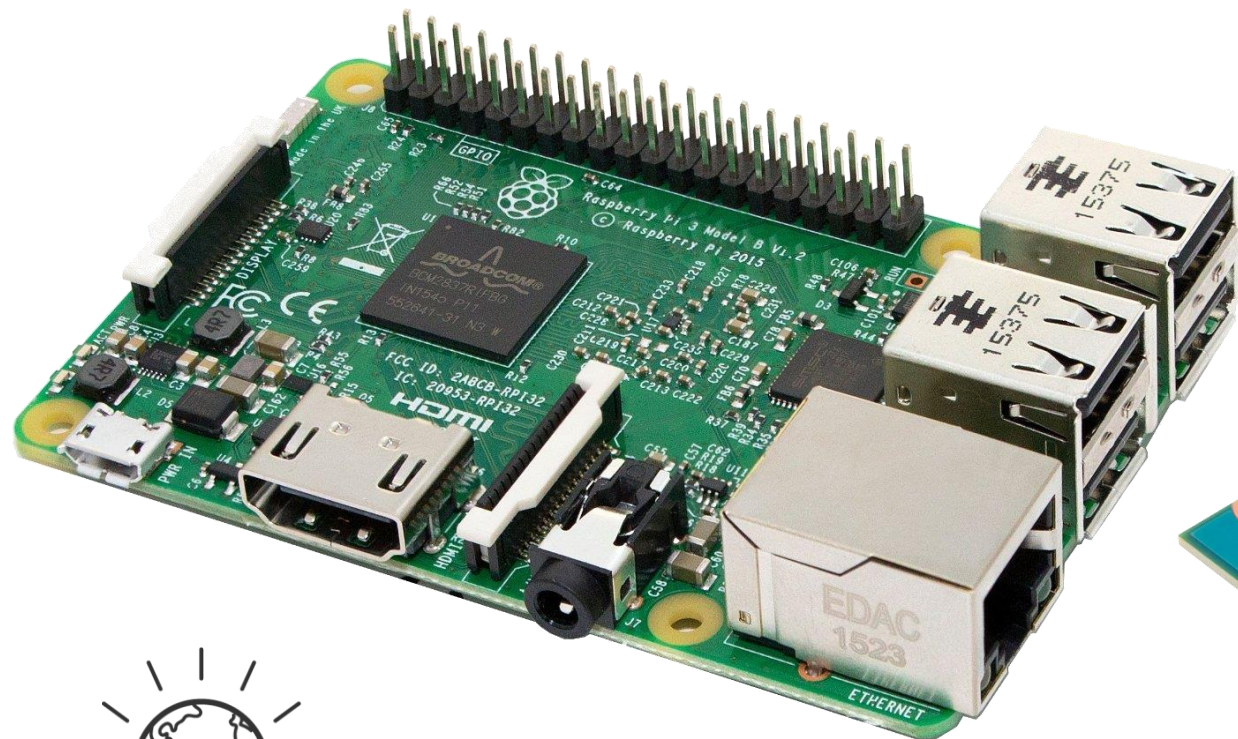


Prof. Francisco  
**Calaça**



# Ambiente de Trabalho

- Outras placas



# Ambiente de Trabalho

- Multímetro



Prof. Francisco  
**Calaça**

# Ambiente de Trabalho

- Minirretífica



Prof. Francisco  
**Calaça** ●●●●●



# Ambiente de Trabalho

- Chaves e alicates



# Ambiente de Trabalho

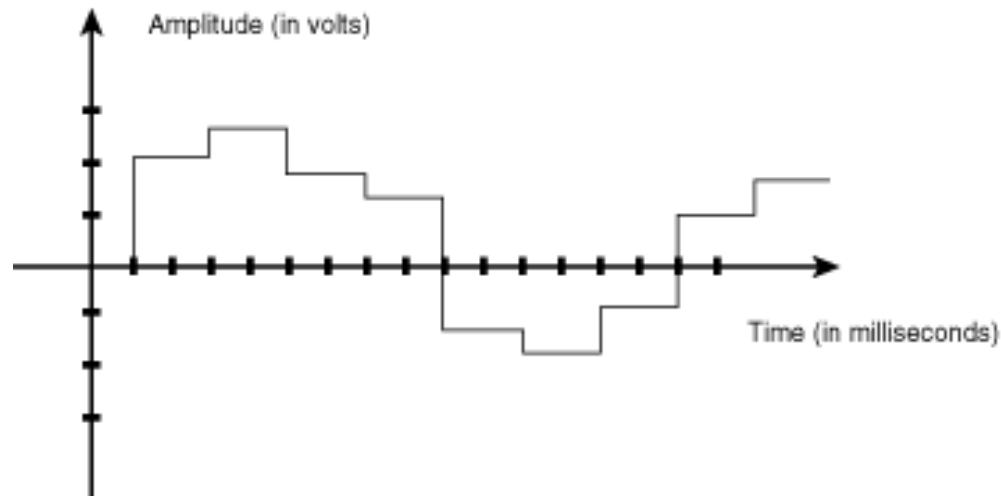
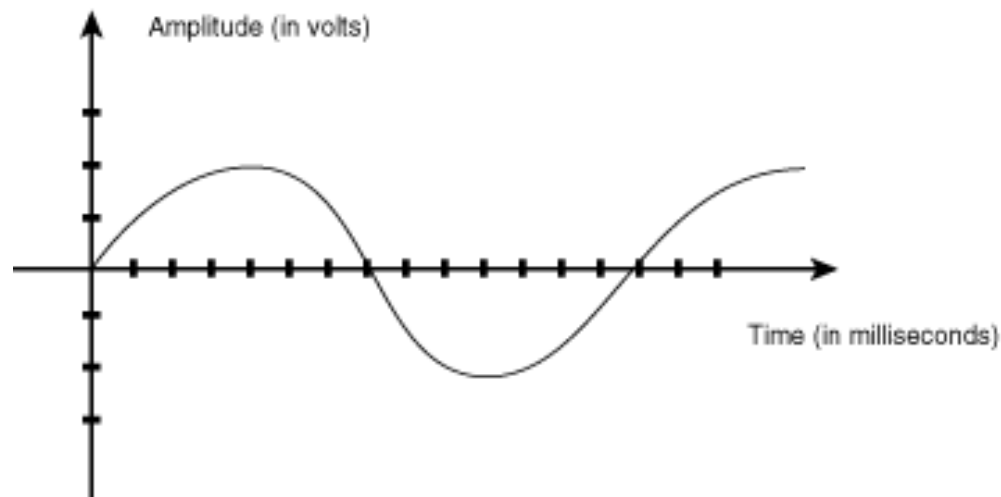
- Softwares
  - uPyCraft
  - Arduino IDE
  - Eclipse/IntelliJ/Netbeans/Visual Studio Code/Notepad++
  - Fritzing
- Sites/Foruns
  - <https://playground.arduino.cc/Portugues/Referencia>
  - <https://stackoverflow.com/questions/tagged/arduino>
  - YouTube
  - Etc

# Tópicos de programação embarcada

- Linguagem Python
  - Micropython
  - <http://docs.micropython.org/en/latest/>
  - uPyCraft
- Linguagem Arduino
  - Baseada em C++
  - <https://playground.arduino.cc/Portugues/Referencia>
  - **setup** vs **loop**
  - 5V ou 3.3V

# Tópicos de programação embarcada

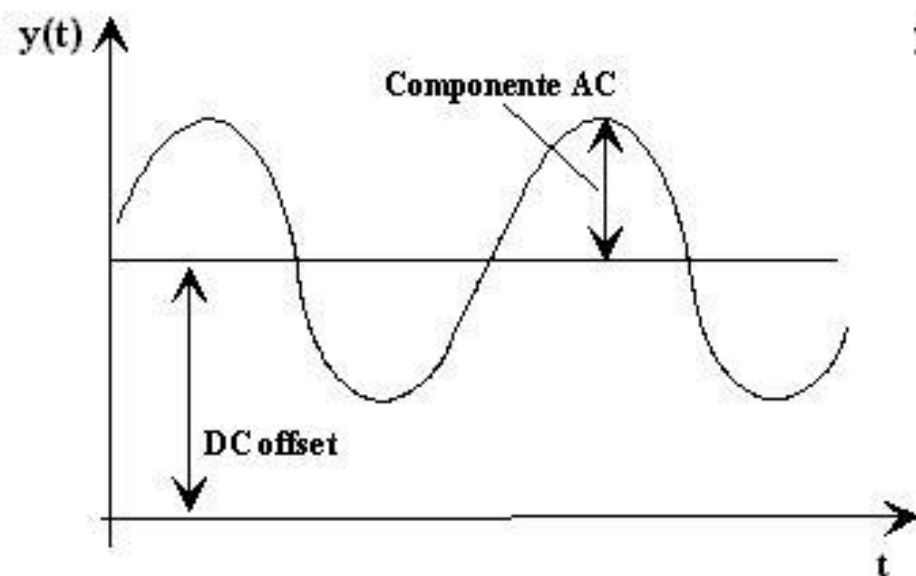
- Digital vs Analógico



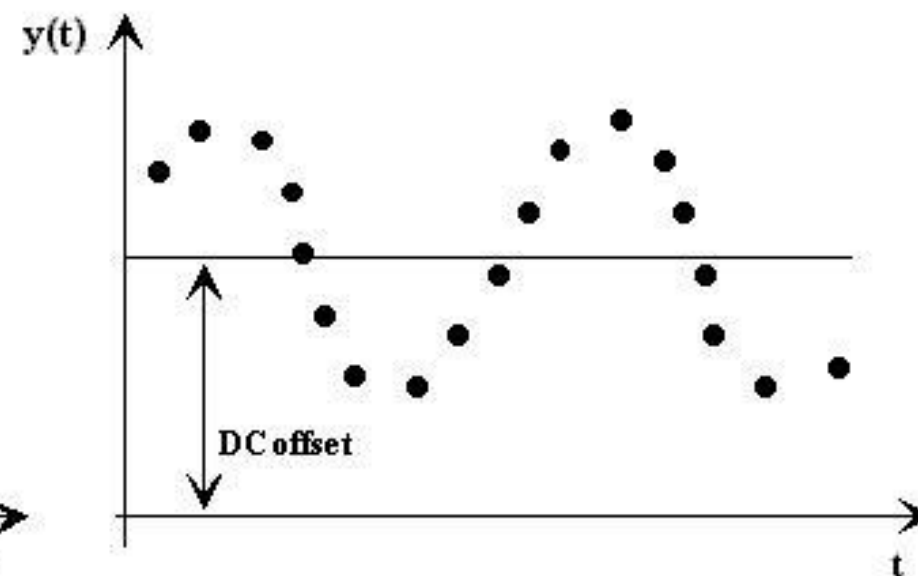
Prof. Francisco  
**Calaça** ●●●●●

# Tópicos de programação embarcada

- Digital vs Analógico



analógico



digital



# Introdução ao Python

- Ambiente
- Sintaxe
- Variáveis
- Strings e números
- Instruções de controle

# Python - Ambiente

- uPyCraft
- Python comum: vscode, sublime, atom, etc...
- Anaconda

# Python - sintaxe

- Extensão .py
- Indentação
- Comentários
  
- `print("hello world")`

# Python - variáveis

```
1  nome = input("Qual seu nome?")
2  idade = input("Qual sua idade?")
3
4  print("{} tem {} anos".format(nome, idade))
5
```

# Python – strings e ints

```
1  idade = input("Qual sua idade?")
2
3  idade_int = int(idade)
4
5  print(idade + "2")
6  print(idade_int + 2)
```



# Python – instrução controle

```
1  idade = int(input("Digite a idade:"))
2
3  if idade < 16:
4      |   print("nao pode votar")
5  else:
6      |   print("pode votar")
```

# Python – laço de repetição for

```
usuarios = ["Maria", "Jose", "Carlos"]  
  
for u in usuarios:  
    print("=====\nPrezado {},\n\n Precisamos que venha
```

# Python – laço de repetição while

```
1  qtd = int(input("Digite a quantidade:"))
2  num = 0
3  while(num <= qtd):
4      print(num)
5      num = num + 1
```

# Nosso primeiro exemplo: Pisca Pisca

```
1  from machine import Pin
2  from time import sleep
3
4  led = Pin(2, Pin.OUT)
5
6  while True:
7      led.value(not led.value())
8      sleep(1.0)
9
10
```

# Nosso primeiro exemplo: Pisca Pisca

```
from machine import Pin
from time import sleep

led = Pin(2, Pin.OUT)

while True:
    led.value(not led.value())
    sleep(1.0)
```



# Micropython

- <http://docs.micropython.org/en/latest/pyboard/tutorial/script.html>
- boot.py – executado quando a placa é ligada
- main.py – executado após o boot.py

# Pisca pisca com Arduino

```
Arquivo  Editar  Sketch  Ferramentas  Ajuda

PISCA_OFICINA_4

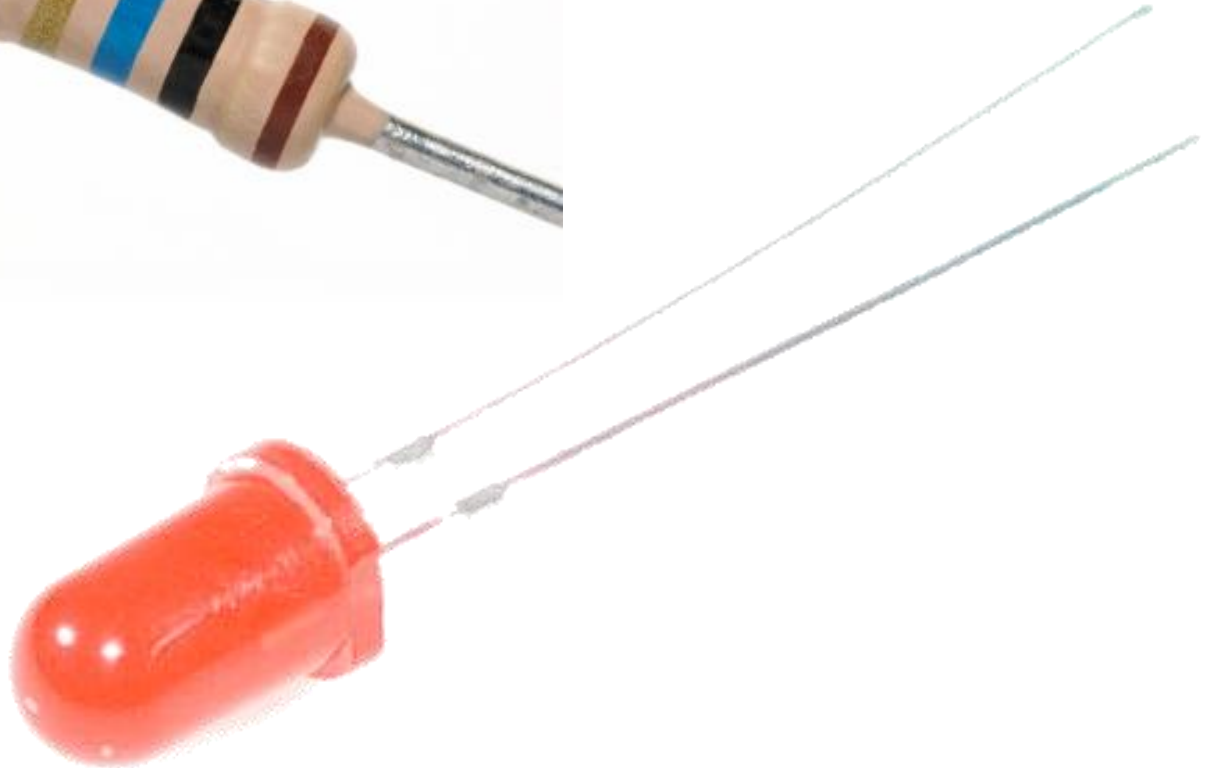
int tempo = 1000;

void setup() {
    pinMode(8, OUTPUT);
}

void loop() {
    digitalWrite(8, HIGH);
    delay(tempo);
    digitalWrite(8, LOW);
    delay(tempo);
}
```

# Mais exemplos de exemplos

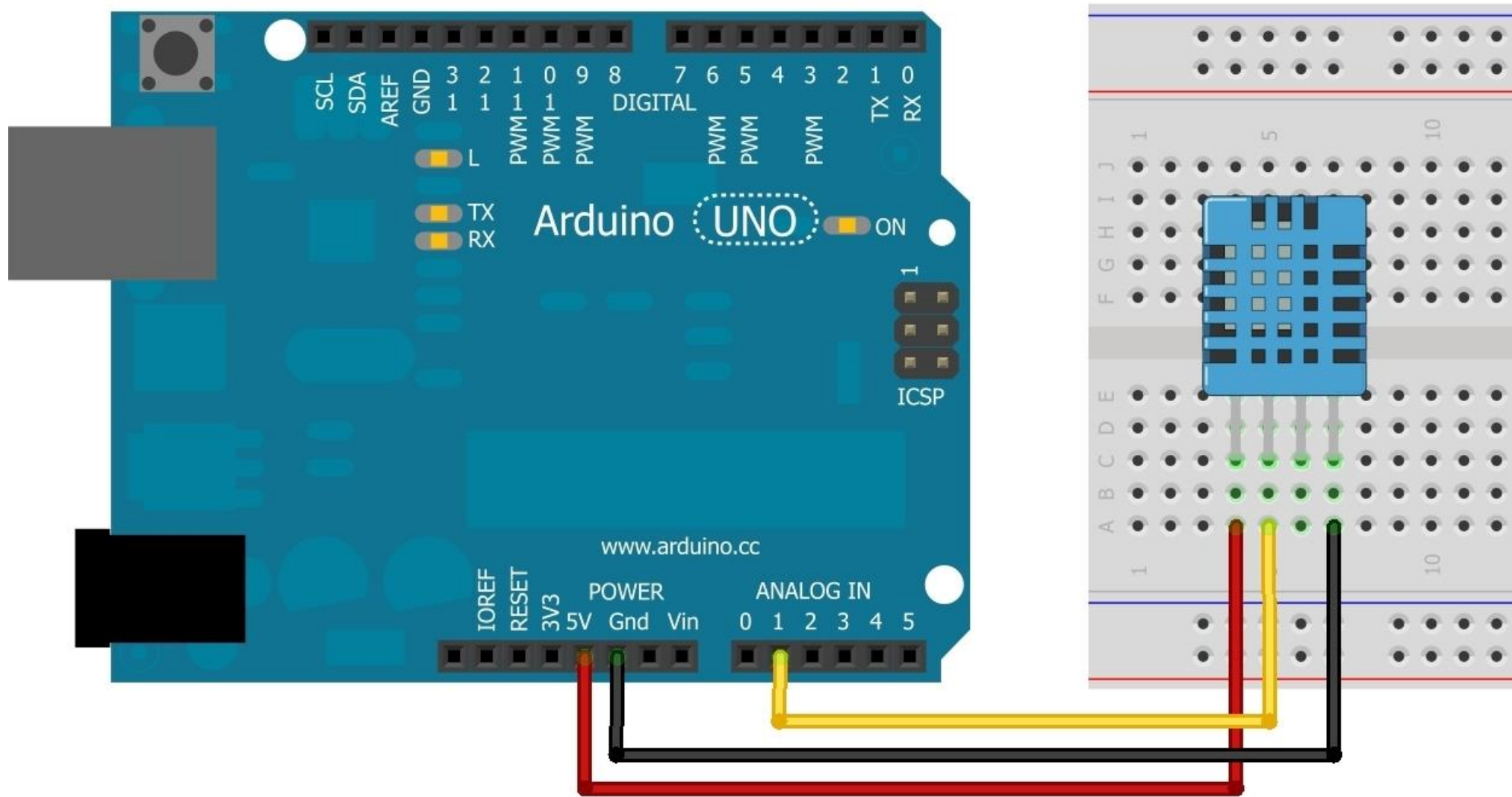
- Pisca-pisca
- Semáforo
- Resistores?
- LEDs?



# Obtenção de dados de sensores

- Sensor de Humidade e Temperatura

# Obtenção de dados de sensores



# Código

```
1  import dht
2  import machine
3  from time import sleep
4
5  d = dht.DHT22(machine.Pin(17))
6
7  =while True:
8      m = d.measure()
9      t = d.temperature()
10     h = d.humidity()
11
12     print(m)
13     print(t)
14     print(h)
15
16     sleep(1.0)
17
```



# Código

```
import dht
import machine
from time import sleep

d = dht.DHT22(machine.Pin(17))

while True:
    m = d.measure()
    t = d.temperature()
    h = d.humidity()

    print(m)
    print(t)
    print(h)

    sleep(1.0)
```

```
#include <DHT.h>

#define DHTPIN A1 // pino que estamos conectado
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

void setup()
{
    Serial.begin(9600);
    Serial.println("DHTxx test!");
    dht.begin();
}

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(t) || isnan(h)) {
        Serial.println("Failed to read from DHT");
    }
    else {
        Serial.print("Umidade: ");
        Serial.print(h);
        Serial.print(" %t");
        Serial.print("Temperatura: ");
        Serial.print(t);
        Serial.println(" *C");
    }
}
```



# Conexão wifi

```
= def do_connect():  
    import network  
    sta_if = network.WLAN(network.STA_IF)  
    if not sta_if.isconnected():  
        print('connecting to network...')  
        sta_if.active(True)  
        sta_if.connect('Iot Smart House', 'IotSmartHousei8#!')  
    while not sta_if.isconnected():  
        pass  
    print('network config:', sta_if.ifconfig())
```

# Dados em json

```
name = "Francisco"
temp = 12.3
humid = 89

data = {"nome": name, "temp": temp, "humid": humid}
json = ujson.dumps(data)

headers = {'Content-Type': 'application/json'}
print(json)
```

# Comunicação com servidor

```
=while True:  
    response = urequests.post("http://192.168.0.34:8080/temphumid/send", data=json, headers=headers)  
    print("send")  
    print(response.text)  
    sleep(2.0)
```

# Exercício

- Enviar a temperature e umidade reais (dados do sensor)

```
name = "Francisco"
temp = 12.3
humid = 89

data = {"nome": name, "temp": temp, "humid": humid}
json = ujson.dumps(data)

headers = {'Content-Type': 'application/json'}
print(json)
```

# Server

- Obs.: este código executa no computador

```
1  from flask import Flask, abort, request
2  import json
3
4  app = Flask(__name__)
5
6
7  @app.route('/temphumid/send', methods=['POST'])
8  def send():
9      print(request.json)
10     return "Ok... Python!"
11
12  app.run(host='0.0.0.0', port=8080, debug=True)
13
```





# Controle remoto

- Este código executa no servidor

```
class Contexto:
    ligado = False

@app.route('/temphumid/send', methods=['POST'])
def send():
    print(request.json)
    return "Ok... Python!"

@app.route('/temphumid/verifica')
def verifica():
    if Contexto.ligado:
        return "on"
    else:
        return "off"

@app.route('/temphumid/on')
def on():
    Contexto.ligado = True
    return "ligado"

@app.route('/temphumid/off')
def off():
    Contexto.ligado = False
    return "desligado"
```

# Controle remoto – micropython

```
response = urequests.get("http://192.168.25.54:8080/temphumid/verifica", data=json, headers=headers)
print(response.text)
if(response.text == "on"):
    led.value(1)
else:
    led.value(0)
sleep(2.0)
```

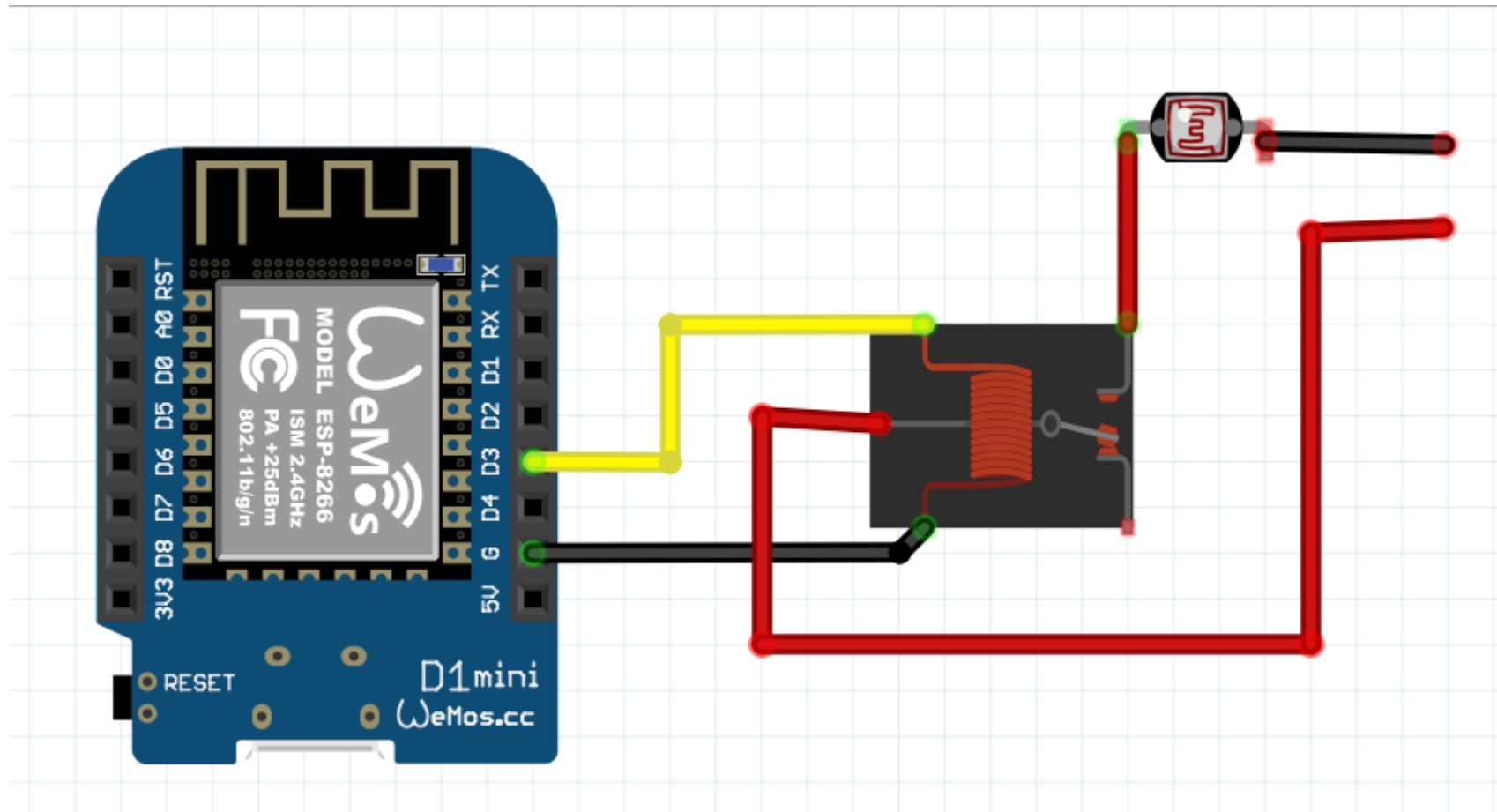
# Iniciar quando ligar

```
main.py X
1
2  exec(open('./wlan-poll.py').read(),globals())
3
```

# Controle remoto – outras formas

- MQTT
  - Mosquitto
  - Publish e Subscribe
- 
- Assunto do próximo módulo

# Acionamento de cargas maiores



Prof. Francisco  
**Calaça**

# Exercício

- Propor uma solução com IoT para um problema real.
- Em grupo
- Apresentar



# Quer mais?

- [contato@franciscocalaca.com](mailto:contato@franciscocalaca.com)
- <https://www.facebook.com/professor.francisco.calaca/>
- <https://www.youtube.com/c/ProfFranciscoCalaça>

