



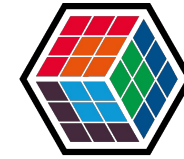
# THE DEVELOPER'S CONFERENCE

## Automação Residencial com Python

Ricardo Salveti <[rsalveti@gmail.com](mailto:rsalveti@gmail.com)>

Ursula Junque <[ursinha@gmail.com](mailto:ursinha@gmail.com)>

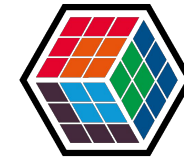
# Por que automação residencial com Python?



THE  
DEVELOPER'S  
CONFERENCE

- Pela diversão e facilidade
- Trabalhar com hardware de forma fácil e divertida
- Conectar dispositivos diferentes com base em uma mesma linguagem
- Tornar sua vida mais fácil, utilizando apenas Python!

# Casos básicos em automação de hardware



THE  
DEVELOPER'S  
CONFERENCE

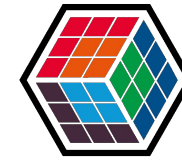
- Placas de baixo custo, principal exemplo: Arduino
- Programação em Arduino não feita em Python, mas sim com sua própria linguagem (próxima a C/C++), editor e ferramentas de flash
- Simples de prototipar uma solução, mas não tão divertido
- Será que seria possível utilizar Python para programar dispositivos embarcados como feito com Arduino?

# MicroPython



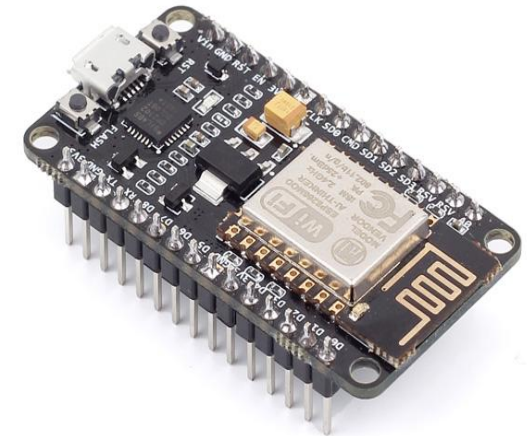
- Python 3.x para microcontroladores
- Subconjunto da biblioteca padrão do Python
- Capacidade de rodar em CPUs como ARM Cortex-M0
  - Mínimo de 256kB de ROM (flash), 16k de RAM e 16MHz
- OS + Interpretador
- Prompt interativo
- Criado via Kickstarter
- Hardware de referência: **pyboard**
- Vários hardwares adicionais suportados, ex: **ESP8266**

# ESP8266

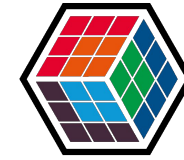


THE  
DEVELOPER'S  
CONFERENCE

- Microcontrolador da Espressif Systems
- Suporte a WiFi e TCP/IP
- 32bit rodando a 80MHz
- 16 pinos para GPIO
- 1 pino 10-bits ADC
- Muito barato, inclusive para mercados brasileiro
  - ~R\$20/30 no Mercado Livre
  - ~R\$45 no estande FILIPEFLOP



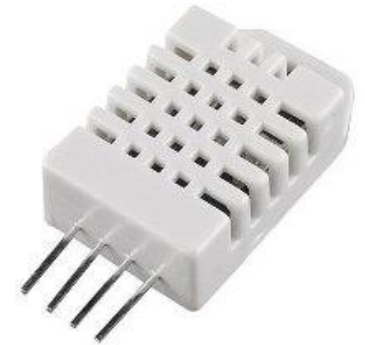
# MicroPython: Exemplo básico com LED



THE  
DEVELOPER'S  
CONFERENCE

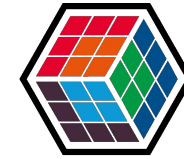
```
>>> from machine import Pin
>>> led = Pin(2, Pin.OUT)
>>> led.on()
>>> led.off()
>>> print("Ola TDC 2017 Sao Paulo")
Ola TDC 2017 Sao Paulo
```

# MicroPython: DHT22



```
>>> import dht
>>> sensor = dht.DHT22(Pin(4))
>>> sensor.measure()
>>> print("Temperatura ", sensor.temperature())
Temperatura  22.54
>>> print("Umidade ", sensor.humidity())
Umidade  78.4
```

# Comunicação com outros serviços? MQTT FTW!



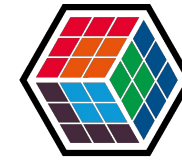
THE  
DEVELOPER'S  
CONFERENCE

- Protocolo de mensagens leve
- Muito utilizado em IoT, para comunicação entre máquinas
- ISO/IEC PRF 20922
- Publisher e Subscriber através de um Message Broker
- Uso de TCP, mas ainda assim bem utilizado em casos com pouca banda
- Broker muito conhecido e de fácil uso: Mosquitto

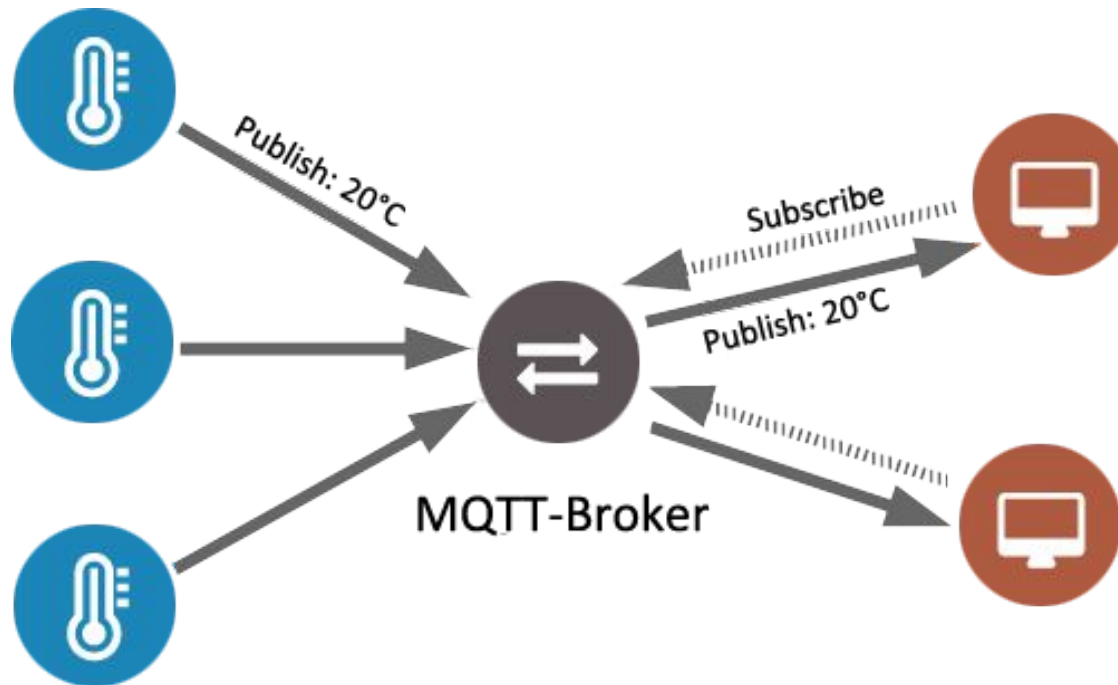


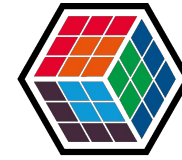


# Funcionamento do MQTT



THE  
DEVELOPER'S  
CONFERENCE

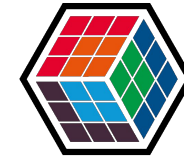




# MicroPython: MQTT

```
>>> client = MQTTClient("esp_tdc", "192.168.1.1", 1883)
>>> client.connect()
>>> sensor = dht.DHT22(Pin(4))
>>> sensor.measure()
>>> client.publish("sala/temperatura",
str(sensor.temperature()))
>>> client.publish("sala/umidade", str(sensor.humidity()))
```

# E como automatizar meus sensores?

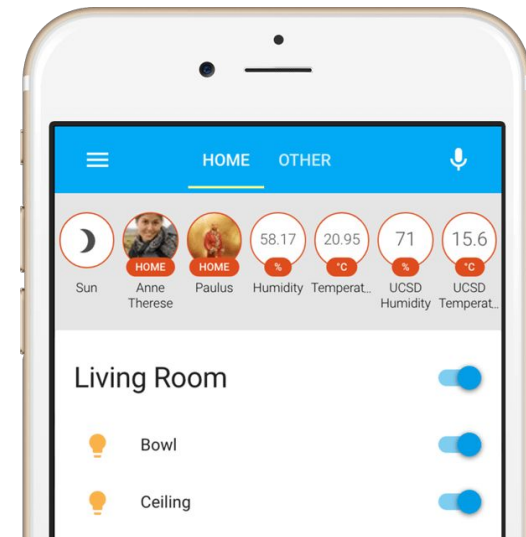


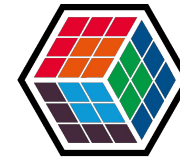
THE  
DEVELOPER'S  
CONFERENCE

- Controle dos sensores de forma remota
- Envio das informações dos sensores para serviços IoT
- Risco do envio de informações confidenciais para serviços externos.
- Será que existe uma solução Open Source?
- Melhor, será que existe uma solução a qual eu pudesse ter o controle completo?

# Home Assistant

- Open Source (Apache V2)
- Feito em Python 3
- Comunidade bem ativa
- Suporte a uma infinidade de sensores e devices
- Execução local ou remota
- Interface web, incluindo interface para mobile
- Fácil desenvolvimento
- **Automation**

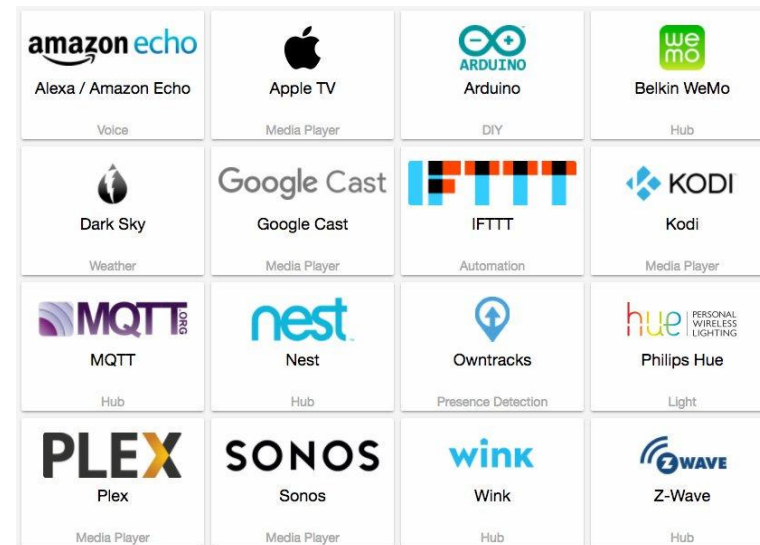




THE  
DEVELOPER'S  
CONFERENCE

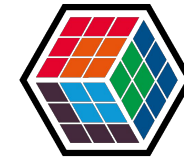
# Home Assistant: Usos

- Controle de lâmpadas inteligentes
- Controle de switches, locks
- Localização de usuários
- Controle de TVs/Media Center
  - Google Cast, Kodi, Apple TV
- Exibição e controle de câmeras
- Notificações (ex: Telegram)



<https://home-assistant.io/components>

# Configurando o HA para controlar um LED



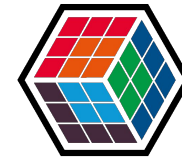
THE  
DEVELOPER'S  
CONFERENCE

light:

- platform: mqtt
- name: "Luz LED"
- command\_topic: "tdc/led/switch"
- qos: 1
- payload\_on: "on"
- payload\_off: "off"

automation:

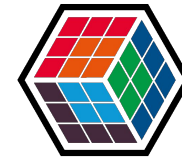
- alias: "Ligar no entardecer"
- trigger:
- platform: sun
  - event: sunset
  - offset: "-01:00:00"
- action:
- service: light.turn\_on
  - entity\_id: light.luz\_led



THE  
DEVELOPER'S  
CONFERENCE

# Demo!

<https://home-assistant.io/demo/>



THE  
DEVELOPER'S  
CONFERENCE

Obrigado(a)!