



**BEM VINDOS!**  
**VENTURUS<sup>4</sup>TECH**



## Módulo IV Internet das Coisas

**VENTURUS4TECH**

Pedro Dionísio  
[pedro.pereira@venturus.org.br](mailto:pedro.pereira@venturus.org.br)

# Na Aula 2: (link: <https://goo.gl/tykpbl>)

```
Serial.available()  
Serial.read();  
Serial.readStringUntil();
```

```
// verifica se há dados na serial  
// le um caractere da serial  
// le uma string até encontrar caractere
```

```
ArduinoJson  
Adafruit_MCP9808();
```

```
// codificar e decodificar um Json  
// sensor de temperatura
```

# Aula 3

- Broker MQTT Mosquitto
- MQTT PubSubClient
- Versão Final do código do Arduino
- Iniciar Aplicação Android

# Broker MQTT Mosquitto

Mosquitto é um broker de mensagens que implementa o protocolo MQTT. MQTT é uma ferramenta robusta que permite a troca de mensagens usando o modelo publish/subscribe. Isso se encaixa perfeitamente no tema “Internet das Coisas” em dispositivos de sensoriamento, smartphones, computadores e microcontroladores como o Arduino.

- Mostrar funcionando!
- MQTTLens para conectar ao broker a enviar/receber mensagens

# Exemplo 12 - MQTT no Arduino

- Baixar o projeto: <https://goo.gl/pFMwSH>
- Ler e entender como funciona a API MQTT
- Executar
  - Digitar texto na porta serial
  - Verificar resposta na porta serial
  - Verificar resposta na telona

Aulas 1 e 2: (link: <https://goo.gl/3JpX4S>)

# Exemplo 13 - Versão Final

```
const char topico[] = "v4tech/vntpejr/iot"; // use seu nome

// funções
void atualiza_led();           // usada somente dentro do loop(). ligar, desligar ou piscar led vermelho
void atualiza_status();       // usada somente dentro do loop(), chama enviaJson() em intervalos fixos de 20s

// envia status completo no formato:
// {"led":ESTADO,"temperatura":VALOR} onde ESTADO: "ligado", "desligado", "piscando"
void enviaJson();

// usada dentro do callback() do MQTT para ler extrair a mensagem que chegou
// {"status":0} -> Deve chamar enviaJson() para informar status da placa.
// {"seta_led":ESTADO} -> Seta estado do LED. ESTADO: "ligado", "desligado", "piscando". Chama
enviaJson() também.
// {"pisca_led":VALOR} -> Seta tempo e pisca led. VALOR: Milisegundos. Chama enviaJson() também.
void analisaJson(String json);

void ligaLed(int porta);
void desligaLed(int porta);
void inverteLed(int porta);
void conectaMqtt();
```

# Exemplo 13 - Versão Final

- Projeto: <https://goo.gl/mDcKJf>