

Professor: Luiz Felipe Oliveira

ESP32 na Nuvem  
com MQTT

# Implantando seu ESP32 na Nuvem



# Modo Station + Cloud

CLOUD

WIFI MODE  
STATION

CLOUD



DISPOSITIVO



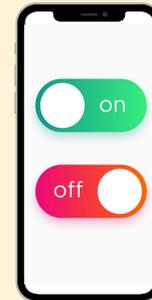
ARDUINO  
IDE



MQTT Client



CLIENTE





MQTT(Message Queuing Telemetry Transport) é um protocolo de comunicação máquina para máquina (M2M - Machine to Machine) com foco em Internet of Things (IoT) que funciona em cima do protocolo TCP/IP.

Nos anos 90, a IBM criou o protocolo MQTT. Sua origem se deu à necessidade de um protocolo simples e leve que conseguisse comunicar várias máquinas entre si, uma comunicação que ocorreria utilizando microcontroladores e que tivesse uma taxa de transmissão leve para a comunicação entre as máquinas e sensores.



Por esses motivos, em algumas redes o MQTT pode ser 93 vezes mais rápido quando comparado ao HTTP.

fonte: [medium.com](https://medium.com)

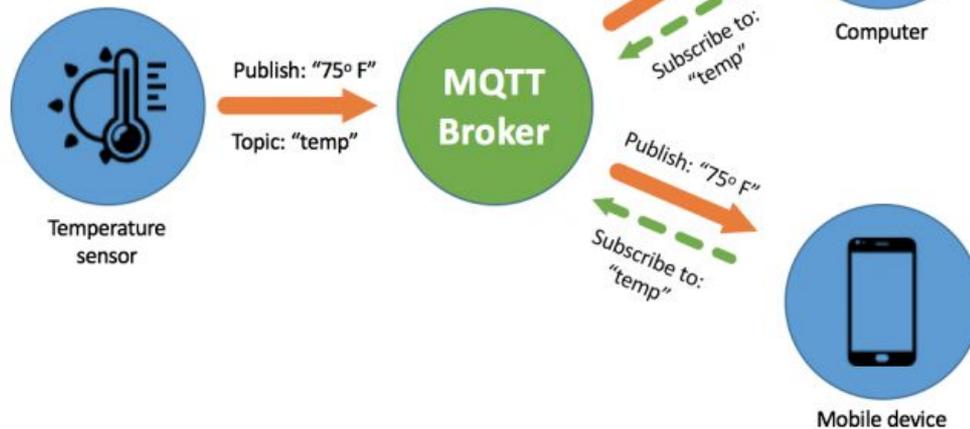
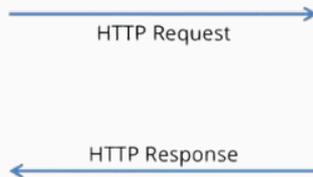
- Feito para Máquinas;
- Centrado nos dados;
- Conexão permanente;
- Super leve;
  - header pequeno;
  - mensagens de 2 bytes;
- Baixo consumo de hardware e energia (20% menos energia);

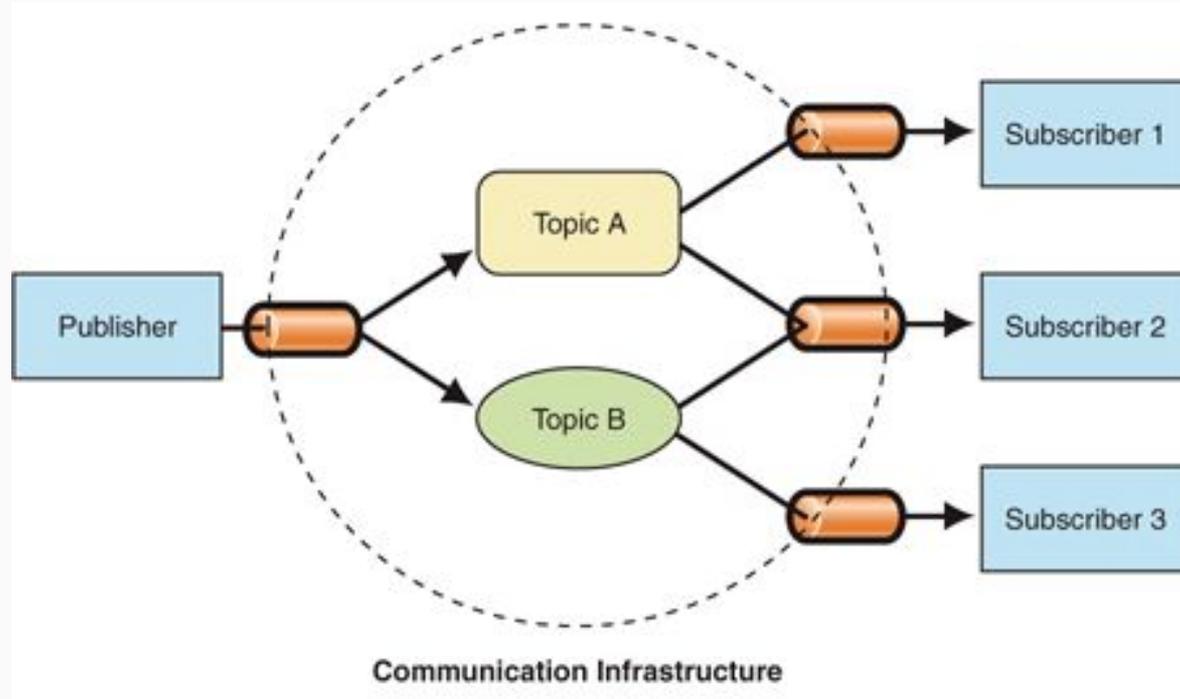
- Feito para Humanos;
- Centrado em documentos;
- Conexão intermitente;
- Menor número de conexões simultâneas;

# Por que MQTT?

Parâmetro	MQTT	HTTP
<b>Missão</b>	Transporte de telemetria de enfileiramento de mensagens	Protocolo de Transferência de Hipertexto
<b>Arquitetura</b>	Ele funciona no modelo de publicação / assinatura.	Funciona no modelo de solicitação / resposta.
<b>Complexidade</b>	Tem menos complexidade.	É mais complexo.
<b>Protocolo de Transmissão</b>	TCP	UDP/TCP
<b>Projeto de protocolo</b>	O design deste protocolo é centrado em dados.	O design deste protocolo é centrado no documento.
<b>Tamanho da Mensagem</b>	O tamanho da mensagem gerada é menor, pois usa o formato binário.	O tamanho da mensagem gerada é maior porque usa o formato ASCII.
<b>Tamanho do cabeçalho</b>	Tem 2 bytes.	É de 8 bytes.
<b>Número da porta</b>	Funciona na porta 1883.	Funciona na porta 80.
<b>Segurança de dados</b>	Fornece segurança de dados com SSL / TLS.	Não fornece segurança, mas o HTTPS foi criado para isso.

# Por que MQTT?





**MQTT  
compartilhado**

 **ECLIPSE™**  
FOUNDATION



**MQTT privado**



## Public MQTT Brokers

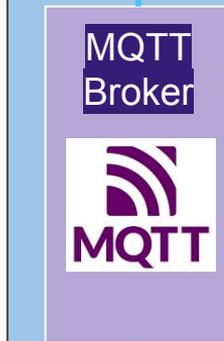
Name	Broker Address	TCP Port	TLS Port	WebSocket Port	Message Retention	Persistent Session	Sign Up Required	Link
Eclipse	mqtt.eclipse.org	1883	N/A	80, 443	YES	YES	NO	<a href="#">Link</a>
Mosquitto	test.mosquitto.org	1883	8883, 8884	80	YES	YES	NO	<a href="#">Link</a>
HiveMQ	broker.hivemq.com	1883	N/A	8000	YES	YES	NO	<a href="#">Link</a>
Flespi	mqtt.flespi.io	1883	8883	80, 443	YES	YES	YES	<a href="#">Link</a>
Dioty	mqtt.dioty.co	1883	8883	8080, 8880	YES	YES	YES	<a href="#">Link</a>
Fluux	mqtt.fluux.io	1883	8883	N/A	N/A	N/A	NO	<a href="#">Link</a>
EMQX	broker.emqx.io	1883	8883	8083	YES	YES	NO	<a href="#">Link</a>

# Modo Station + Cloud + App

CLOUD

WIFI MODE  
STATION

CLOUD



ARDUINO  
IDE



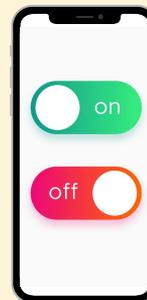
DISPOSITIVO

MQTT Client



MQTT Dash

Control the Things via MQTT protocol



CLIENTE





temp\_labrob

*publish*



temp\_nitmaker

*publish*



temp\_labgames

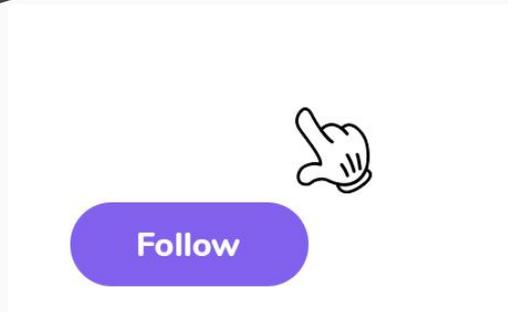
*publish*



*subscribe temp\_labrob*

*subscribe temp\_nitmaker*

*subscribe temp\_labgames*



**POST**

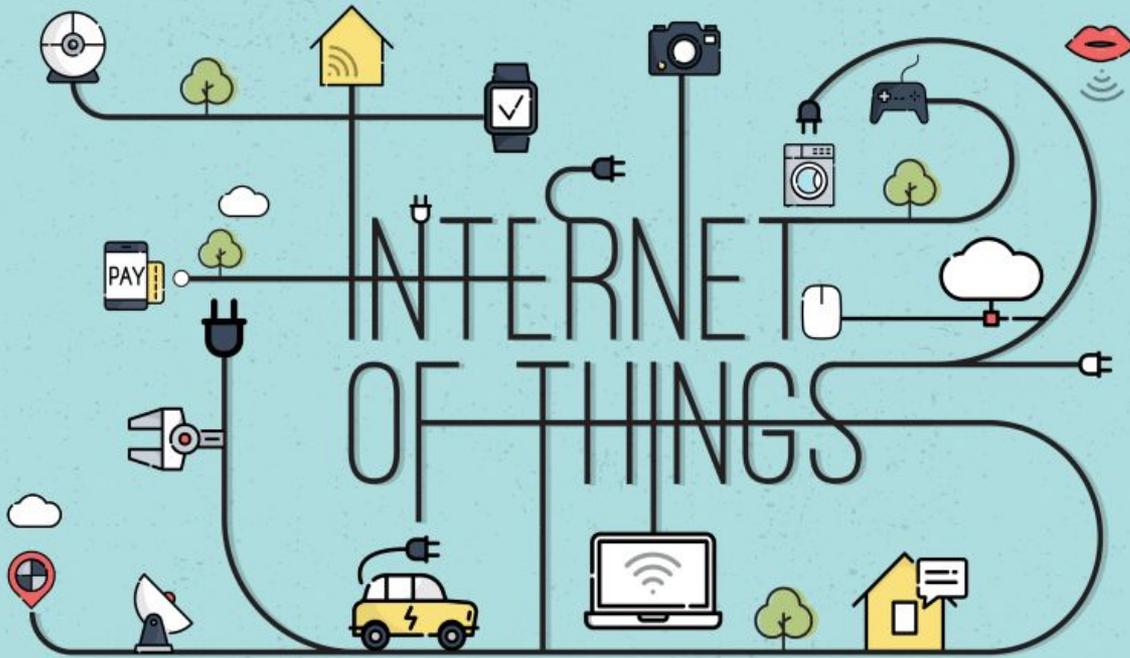
**FOLLOW**



**Parte 0: Explicando o MQTT**

**Parte 1: Publicação dos Dados**

**Parte 2: Visualização dos Dados**



Professor: Luiz Felipe Oliveira

Dúvidas