

# Infra-estrutura para o projeto de PCS 3519 / 3819

michelet@usp.br

# Ambiente

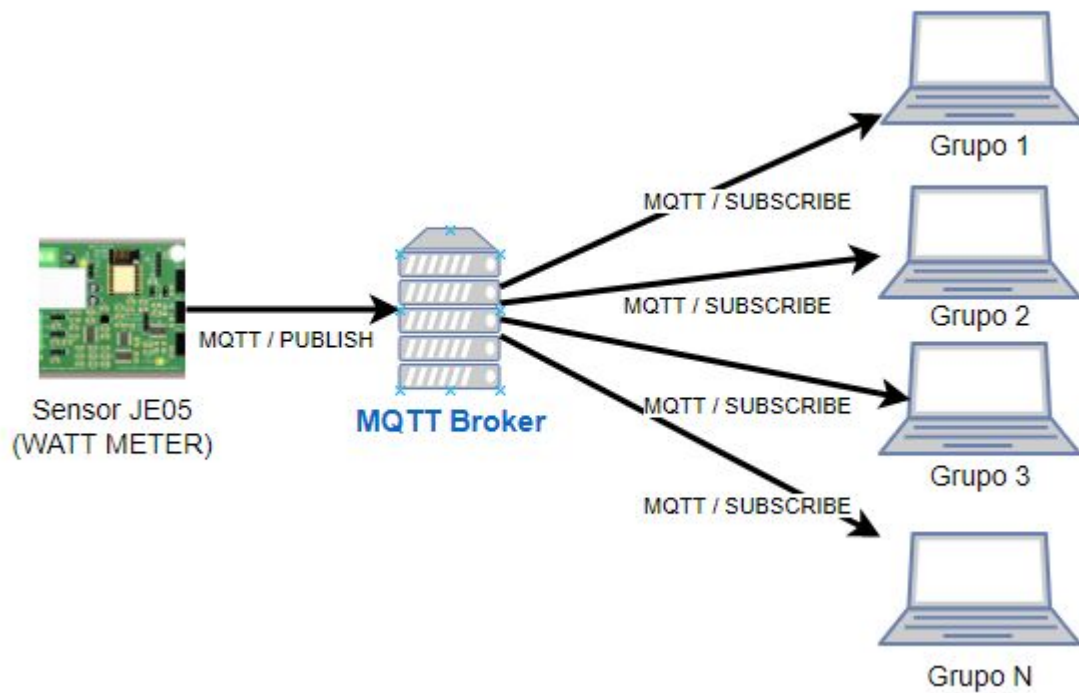
Broker MQTT que disponibiliza os dados de consumo de energia de um laboratório.

- A cada 10 segundos será publicada uma mensagem no broker contendo os dados de consumo de energia. A descrição dos dados está na página 11 do seguinte documento: <https://drive.google.com/file/d/1suTAzTpP2bmTwJ6eJj7A3j2yUCI3iAZI/view?usp=sharing>

Cada grupo tem o papel de obter os dados do broker e desenvolver o projeto a partir desses dados.

A arquitetura de software de referência é [publisher/subscriber](#).

# Elementos da Infra-estrutura



# Como se conectar?

Desenvolver um cliente mqtt e processar as mensagens.

Informações da conexão:

- URL do Broker: 3.88.203.66
- Id do Tópico: 'lab/telemetria/je05'
- Id do Cliente: 'número USP'
- Porta: 1883

No seguinte exemplo feito em python, tem o código necessário para fazer a conexão utilizando a linguagem python. Para poder executar o código é preciso instalar a biblioteca [paho-mqtt](#) (seguir as instruções no link para instalação) e executar o programa que está no seguinte slide

# Exemplo usando Python

```
# paho mqtt library
import paho.mqtt.client as mqtt
# setting up the connection
broker_origin_address='3.88.203.66'
port = 1883
topic = 'lab/telemetria/je05'
client_id = '5730682' # unique client to avoid collisions
# when connected, subscribe to the topic
def on_connect(client, userdata, flags, rc):
    client.subscribe(topic)
# when a new message arrives, print it or do otherwise
def on_message(client, userdata, msg):
    print(msg.payload.decode())
def main():
    client = mqtt.Client(client_id=client_id, clean_session=True,
        userdata=None, protocol=mqtt.MQTTv311,
        transport='tcp')
    # setting up the callbacks
    client.on_connect = on_connect
    client.on_message = on_message
    client.connect(host=broker_origin_address, port=port, keepalive=10)
    client.loop_forever()
if __name__ == '__main__':
    main()
```

# Demo

Após a execução do programa, deve ser apresentado um resultado na tela semelhante ao apresentado embaixo.

```
$ python teste.py
{'DATA': {'VARMS': 131, 'VBRMS': 131, 'VCRMS': 131, 'IARMS': 2, 'IBRMS': 0, 'ICRMS': 11, 'VABRMS': 227, 'VBCRMS': 228, 'VCARMS': 227, 'VABCTRMS': 227, 'PA': 30, 'PB': 0, 'PC': 133, 'PT': 163, 'QA': -10, 'QB': 0, 'QC': -36, 'QT': -46, 'SA': 31, 'SB': 0, 'SC': 137, 'ST': 169, 'FPA': -967, 'FPB': 1000, 'FPC': -970, 'FPT': -964, 'KVARHA': -61160, 'KVARHB': -76890, 'KVARHC': -144209, 'KVARHT': -282260, 'KWAHA': 91618, 'KWHAHB': 10171, 'KWHC': 242866, 'KWHT': 344656, 'FREQ': 5988, 'TEMP': -32768, 'SERRS': 0, 'WRSSI': -18, 'UPTIME': 1736192}, 'datetime': '2021-02-15T20:20:46.743288'}
```