

AC3 - Desenvolvimento para Internet das Coisas

Seu endereço de e-mail será registrado quando você enviar este formulário.

Não é djalma.lima@aluno.faculdadeimpacta.com.br? [Trocar de conta](#)

***Obrigatório**

Considere os tópicos descritos a seguir. Qual comando você usaria em um 1 ponto programa do tipo Subscriber para assinar apenas nos três primeiros tópicos. *

```
1 TOPICO_1 = "/dev_iot_impacta/equipe/100/comando_luz/01/"
2 TOPICO_2 = "/dev_iot_impacta/equipe/101/comando_luz/01/"
3 TOPICO_3 = "/dev_iot_impacta/equipe/102/comando_luz/01/"
4 TOPICO_4 = "/dev_iot_impacta/equipe/100/comando_luz/02/"
5 TOPICO_5 = "/dev_iot_impacta/equipe/101/comando_luz/02/"
6 TOPICO_6 = "/dev_iot_impacta/equipe/102/comando_luz/02/"
```

- ☐ client.subscribe("/dev_iot_impacta/equipe/100/101/102/comando_luz/01/02")
- ☐ client.subscribe("/dev_iot_impacta/equipe/*")
- ☐ client.subscribe("/dev_iot_impacta/equipe/#")
- ☒ client.subscribe("/dev_iot_impacta/equipe+/comando_luz/")
- ☐ client.subscribe("/dev_iot_impacta/equipe/+")



Considerando o programa a seguir, qual comando você usaria para receber todas as mensagens publicadas por ele. *

1 ponto

```
1  #!/usr/bin/env python3
2  import paho.mqtt.client as mqtt
3  from time import sleep
4
5  TOPIC01 = "/IoT_Impacta/valor1/"
6  TOPIC01 = "/IoT_Impacta/valor2/"
7
8  client = mqtt.Client()
9  client.connect("iot.eclipse.org", 1883, 60)
10
11 while True:
12     dado = "1".encode()
13     client.publish(TOPIC01, dado, qos=0)
14     dado = "2".encode()
15     client.publish(TOPIC02, dado, qos=0)
16     sleep(2)
```

- ☐ mosquitto_sub -h iot.eclipse.org -t /IoT_Impacta/valor+
- ☐ mosquitto_pub -h iot.eclipse.org -t /IoT_Impacta/#
- ☐ mosquitto_pub -h iot.eclipse.org -t /IoT_Impacta/+
- ☐ mosquitto_sub -h iot.eclipse.org -t /IoT_Impacta/*
- ☒ mosquitto_sub -h iot.eclipse.org -t /IoT_Impacta/#



Selecione a opção que contém uma consulta válida a um broker Mosquitto 1 ponto
que está sendo executado em um computador na internet com uma URL
válida de onde você fará a consulta. A consulta deve receber as
mensagens publicadas no seguinte tópico: /dev_iot_impacta/valor/ *

- ☒ mosquitto_sub -h iot.eclipse.org -t /dev_iot_impacta/+/-v
- ☐ mosquitto_sub -h 127.0.0.1 -t /dev_iot_impacta/# -v
- ☐ mosquitto_pub -h 127.0.0.1 -t /dev_iot_impacta/valor/
- ☐ mosquitto_sub -h iot.eclipse.org -t /dev_iot_impacta/*
- ☐ mosquitto_pub -t /dev_iot_impacta/valor/+/-v

Com relação ao protocolo MQTT (Message Queue Telemetry Transport) é 1 ponto
incorreto afirmar que: *

- ☐ Podemos dividir as mensagens no MQTT em duas estruturas: Tópico e Payload.
- ☐ O MQTT consiste em um protocolo de mensagens leve, criado para comunicação M2M (Machine to Machine).
- ☐ No MQTT publishers enviam informação para o Broker, subscribers recebem informação do Broker e o Broker gerencia a troca de mensagens.
- ☒ O Payload corresponde a parte da mensagem no MQTT que contém o endereço do publisher.
- ☐ Teoricamente, não há limite especificado de subscribers e publishers em uma mesma comunicação MQTT, pois o limite nesse aspecto é do Broker em lidar com as conexões.



Analisando o código a seguir podemos afirmar que: *

1 ponto

```
import paho.mqtt.client as mqtt
from time import sleep
TOPICO = "/IoT_Impacta/valor/"

def conectou(client, userdata, flags, rc):
    print("Conectado! Código recebido:"+str(rc))
    client.subscribe(TOPICO)

def chegou_mensagem(client, userdata, msg):
    v = msg.payload()
    print(msg.topic+" "+v.decode())

client = mqtt.Client()
client.on_connect = conectou
client.on_message = chegou_mensagem
client.connect("127.0.0.1", 1883, 60)
client.loop_forever()
```

- ☐ O programa publica uma mensagem no tópico /dev_iot_impacta/valor/ a cada 5 segundos.
- ☐ O programa publica a temperatura ambiente no tópico /dev_iot_impacta/valor/ .
- ☒ O endereço do Broker conectado é o local host.
- ☐ O programa publica uma mensagem no tópico /dev_iot_impacta/valor/ .
- ☐ O programa recebe mensagens pelo tópico /dev_iot_impacta/sensor/.



Com relação ao programa a seguir qual alternativa está correta? *

1 ponto

```
import paho.mqtt.client as mqtt
from random import randint
from time import sleep

client = mqtt.Client()
client.connect("127.0.0.1", 1883)

while True:
    t = randint(0,50)
    payload = str(t).encode()
    if t>25:
        client.publish("/iot/ac/x/",payload,qos=0)
    else:
        client.publish("/iot/ac/y/",payload,qos=0)
    sleep(5)
```

- ☒ Para funcionar este programa deve estar na mesma máquina que o Broker.
- ☐ Este programa utiliza um Broker externo para publicar suas mensagens.
- ☐ Para receber apenas valores maiores que 25 você deve utilizar a seguinte linha de comando: `mosquitto_sub -h iot.eclipse.org -t /iot/ac/x/ -v`
- ☐ Para receber apenas valores maiores que 25 você deve utilizar a seguinte linha de comando: `mosquitto_sub -h 127.0.0.1 -t /iot/ac/+/ -v`
- ☐ Para receber apenas valores maiores que 25 você deve utilizar a seguinte linha de comando: `mosquitto_sub -h iot.eclipse.org -t /iot/ac/+/ -v`



Podemos afirmar sobre a biblioteca subprocess que ao utilizarmos o script 1 ponto
abaixo, teremos como resposta: *

```
import subprocess  
  
command = "date"  
output = subprocess.check_output(command, shell=True)  
print(output.decode())
```

- ☐ a checagem do subprocess em dados
- ☒ a data decorrente do sistema operacional
- ☐ a data referente da fabricação do subprocess
- ☐ a data decorrente do programa check_output
- ☐ o retorno de True de acordo com o Shell

Ao utilizarmos o script abaixo, teremos como retorno: *

1 ponto

```
import subprocess  
  
subprocess.check_call("wget  
https://wordpress.org/latest.zip", shell=True)
```

- ☐ uma chamada do sistema operacional
- ☐ um erro no comando CLI, pois o browser não é executado devido ao argumento estar como shell=True
- ☒ o download do arquivo da página descrita no comando
- ☐ a abertura do Browser WGET como a página da wordpress descrita no comando
- ☐ o retorno TRUE da checagem de chamada caso a página exista



Podemos dizer da biblioteca psutil que: *

1 ponto

- ☐ serve para a geração de números randômicos inteiros
- ☒ serve para monitoria do sistema operacional
- ☐ serve como requisição de url pelo sistema operacional
- ☐ serve como protocolo de comunicação para IoT
- ☐ serve para utilizar comandos do terminal shell dentro do python

O script abaixo funciona para *

1 ponto

```
import psutil  
  
r = psutil.virtual_memory()  
  
print(str(r))
```

- ☒ listar todos os parâmetros da memória virtual do Sistema Operacional
- ☐ utilizar memória virtual
- ☐ limpar a memória virtual
- ☐ lista somente a memória cache do sistema operacional
- ☐ criar memória virtual em programa python

Uma cópia das suas respostas será enviada por e-mail para djalma.lima@aluno.faculdadeimpacta.com.br.

Enviar

Este formulário foi criado em FACULDADE IMPACTA DE TECNOLOGIA - FIT. [Denunciar abuso](#)

Google Formulários

