RSE 2021-2022 / Alejandro Albert Casañ

Memoria práctica 3

1.- Ejecuta este código y explica el resultado obtenido.

Ejecutamos sisub.py y el resultado es el siguiente:

```
sisub: msg received with topic: $SYS/sizes/10 and payload: b'30982846'
sisub: msg received with topic: $SYS/sizes/20 and payload: b'15800416'
sisub: msg received with topic: $SYS/sizes/50 and payload: b'26991889'
sisub: msg received with topic: $SYS/sizes/100 and payload: b'10591720'
sisub: msg received with topic: $SYS/sizes/200 and payload: b'29614478'
sisub: msg received with topic: $SYS/sizes/500 and payload: b'6231302'
sisub: msg received with topic: $SYS/sizes/1k and payload: b'2408389'
sisub: msg received with topic: $SYS/sizes/2k and payload: b'8484740'
sisub: msg received with topic: $SYS/sizes/5k and payload: b'543761'
sisub: msg received with topic: $SYS/sizes/20k and payload: b'215378'
sisub: msg received with topic: $SYS/sizes/100k and payload: b'157574'
sisub: msg received with topic: $SYS/lifetimes/0 and payload: b'1167700'
sisub: msg received with topic: $SYS/lifetimes/1 and payload: b'485416'
sisub: msg received with topic: $SYS/lifetimes/2 and payload: b'124923'
sisub: msg received with topic: $SYS/lifetimes/5 and payload: b'62472'
sisub: msg received with topic: $SYS/lifetimes/10 and payload: b'48339'
sisub: msg received with topic: $SYS/lifetimes/20 and payload: b'62408'
sisub: msg received with topic: $SYS/lifetimes/50 and payload: b'67669'
sisub: msg received with topic: $SYS/lifetimes/100 and payload: b'36494'
sisub: msg received with topic: $SYS/lifetimes/200 and payload: b'10474'
sisub: msg received with topic: $SYS/lifetimes/500 and payload: b'8371'
sisub: msg received with topic: $SYS/lifetimes/1k and payload: b'4884'
sisub: msg received with topic: $SYS/lifetimes/2k and payload: b'3704'
sisub: msg received with topic: $SYS/broker/uptime and payload: b'98874 seconds'
```

Nada más ejecutarlo recibimos un mensaje de que nos hemos conectado y empezamos a recibir los mensajes publicados con el topic correspondiente.

2.- Ejecuta este codigo (en otro terminal) y explica el resultado obtenido.

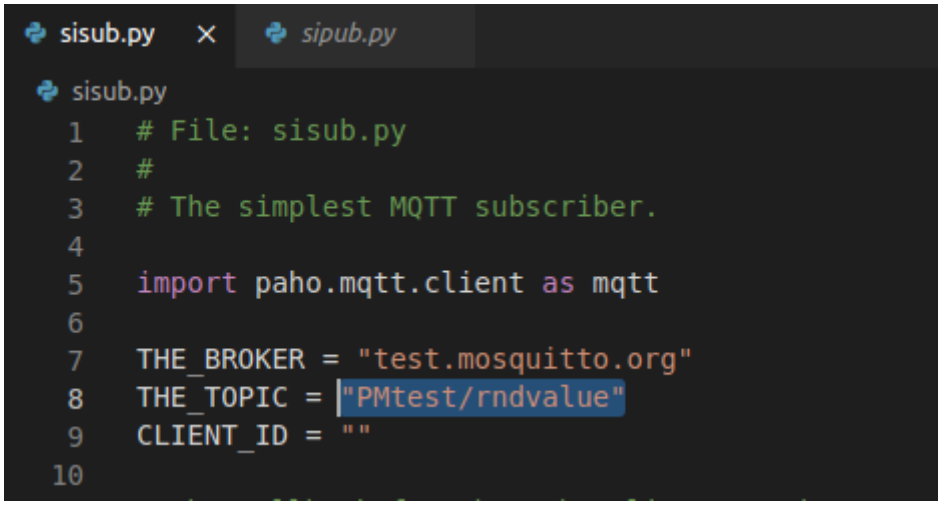
Ejecutamos sipub.py y tenemos el siguiente resultado:

```
vmrse@vmrse-VirtualBox:~/Escritorio/rse/pract3$ python3 sipub.py
sipub: msg published (mid=1)
sipub: msg published (mid=2)
Connected to  test.mosquitto.org port:  1883
Flags:  {'session present': 0} returned code:  0
sipub: msg published (mid=3)
```

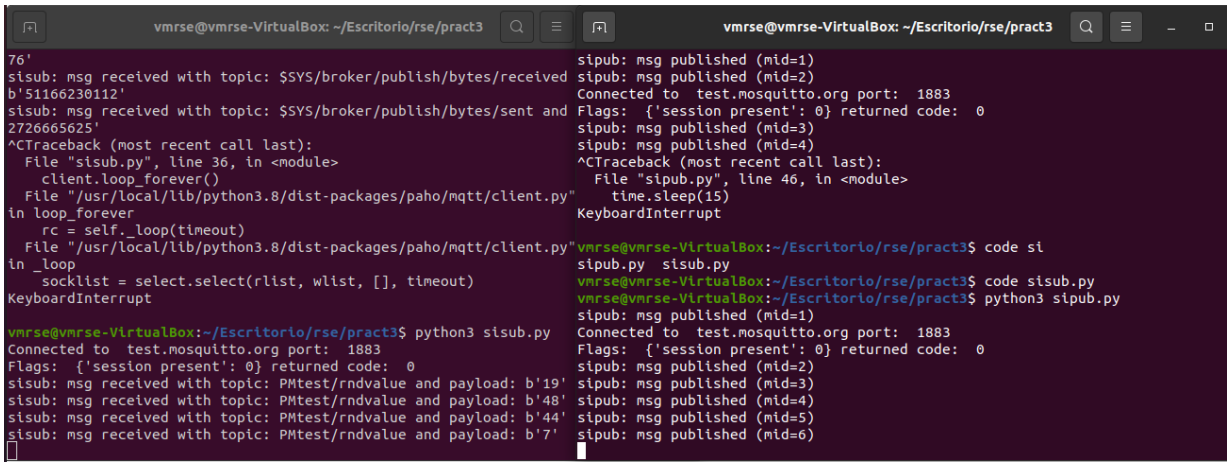
Cuando ejecutamos el publicador inmediatamente empieza a enviar mensajes en un intervalo de 15 segundos y se muestra por pantalla.

3.- Modifica sisub.py para poder recibir los datos enviados por sipub.py.

Cambiamos el topic del suscriptor para que sea el mismo que el del publicador.



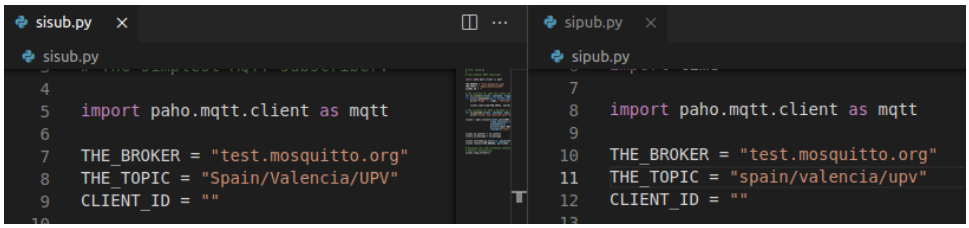
Y si los ejecutamos a la vez el resultado es el siguiente:



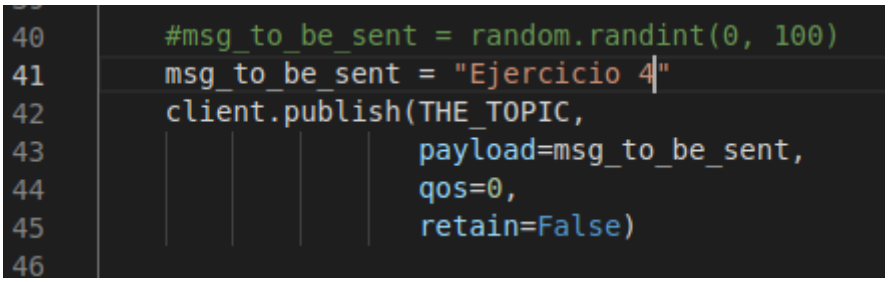
Podemos ver que funcionan correctamente. El suscriptor ve todos los mensajes que envía el publicador.

4.- Modifique el código del suscriptor para que use el topic Spain/Valencia/UPV y el del productor para que use el topic spain/valencia/upv y como mensaje use el texto que desee. Describe lo que obtienes y por qué.

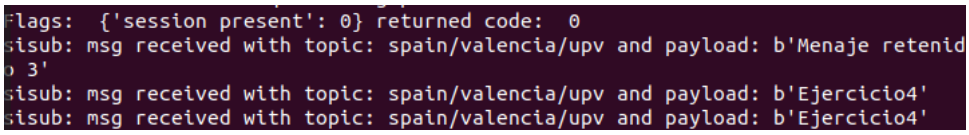
Ahora vamos a modificar los topics del suscriptor y el publicador.



Y a continuación, vamos a modificar el mensaje que envía el publicador.



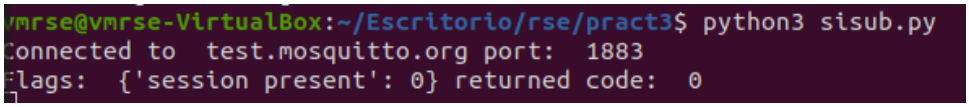
En el momento de ejecutar vemos que no se comunican y esto se debe a que no están operando en el mismo topic pues se distinguen las mayúsculas y minúsculas (case sensitive). Si ajustamos ese error vemos que funciona bien.



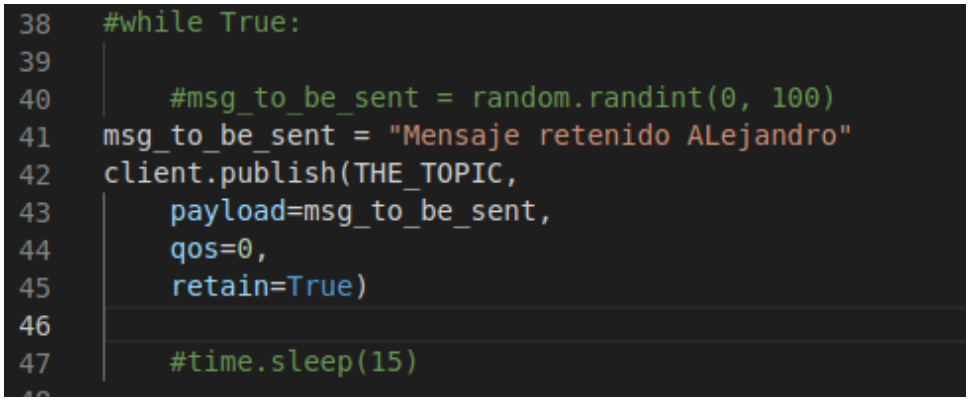
5.- Prueba los siguentes casos:

- **Publica un mensaje con la opcion de retained a “False”. ¿Qué recibe el “subscriber”?**
- **Publica un mensaje con la opcion de retained a “True”. ¿Qué recibe el “subscriber”?**
- **Publica varios mensajes (diferentes) con la opcion de retained a “True” antes de activar el “subscriber”. ¿Qué recibe el “subscriber”?**

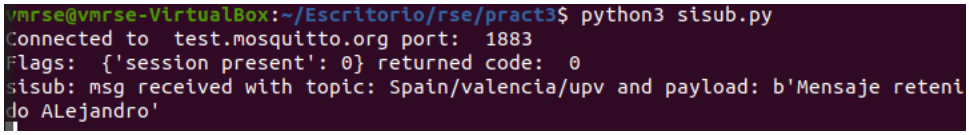
Si tenemos la opción de retained = False no recibimos ningún mensaje al suscribirse.



Si vamos modificando el código para que haya un mensaje retenido y pare



Vemos que cuando se conecta recibe un mensaje automáticamente. Este mensaje se corresponde al último mensaje publicado con ese flag activado.



Para borrar el mensaje retenido habría que enviar un mensaje vacío manteniendo el flag de retained en true.

6.- Crea una aplicación de chat muy básica, donde todos los mensajes publicados de cualquiera de los miembros sean recibidos solo por los miembros del grupo.

Para esta actividad se ha usado el código del publicador y se ha hecho que también esté suscrito al topic. De esa forma, cada usuario es publicador y suscriptor.

```
vmrse@vmrse-VirtualBox:~/Escritorio/rse/pract3$ python3 chat.py
Connected to  test.mosquitto.org port:  1883
Flags:  {'session present': 0} returned code:  0
b'[alalca3]: '
Puede escribir mensajes desde la línea de comandos. Para salir del chat escriba ESC
Buenas
b'[alalca3]: Buenas'
Todo funciona perfectamente
b'[alalca3]: Todo funciona perfectamente'
ESC
Saliendo del chat.......
vmrse@vmrse-VirtualBox:~/Escritorio/rse/pract3$
```

El código correspondiente se encuentra en el fichero chat.py

7.- Crea una aplicacion en python para replicar el ejemplo del Seminario 4 en el que leiamos datos desde TTN utilizando el cliente mqtt-explorer. En este caso es suficiente imprimir todo el JSON que llega.

En este caso utilizaremos el código del suscriptor para obtener el JSON que llega desde el TTN. Si ejecutamos el código vemos el JSON recibido

El primer JSON recibido es el siguiente:

{"end_device_ids":{"device_id":"lopy3sense","application_ids":{"application_id":"lopys2ttn"},"dev_eui":"70B3D5499EAA6C64","join_eui":"70B3D57ED002AE7C","dev_addr":"260B2681"},"correlation_ids":["as:up:01FM217XN3GWEVJ051ZYJC3D2V","gs:conn:01FKNQ3QGSJ3HBS3VMRMY0HNK8","gs:up:host:01FKNQ3QHJR0K31XHZQ5MRX7VF","gs:uplink:01FM217XECHG00EM53RDFHT8YX","ns:uplink:01FM217XEFEPJ1RE2C1G2SKA0V","rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01FM217XEE8YSC0T85JKB271SD","rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01FM217XN2V1DSE5VE25ND8H9Y"],"received_at":"2021-11-09T09:42:36.964837598Z","uplink_message":{"session_key_id":"AXx9ahQWLrp45Lpno7DT9w==","f_port":2,"f_cnt":32268,"frm_payload":"QeFgYEIePFi/gAAA","decoded_payload":{"humidity":39.558929443359375,"lux":-1,"temperature":28.17205810546875},"rx_metadata":[{"gateway_ids":{"gateway_id":"main-gtw-grc","eui":"B827EBFFFE7FE28A"},"time":"2021-11-09T09:42:36.723671Z","timestamp":4257770316,"rssi":-5,"channel_rssi":-5,"snr":8.8,"location":{"latitude":39.482534878470204,"longitude":-0.3463913363006933,"altitude":9,"source":"SOURCE_REGISTRY"},"uplink_token":"ChoKGAoMbWFpbi1ndHctZ3JjEgi4J+v//n/iihDM1qHuDxoMCIyFqYwGEPqA4eQCIODBjLf1214=","channel_index":7},{"gateway_ids":{"gateway_id":"rak-gtw-grc","eui":"B827EBFFFE336296"},"timestamp":2542672948,"rssi":-105,"channel_rssi":-105,"snr":3.8,"location":{"latitude":39.48272119427445,"longitude":-0.3471749450839346,"altitude":9,"source":"SOURCE_REGISTRY"},"uplink_token":"ChkKFwoLcmFrLWd0dy1ncmMSCLgn6//+M2KWELS4uLwJGgwIjIWpjAYQ2e+l5wIgoNb4mICqXg==","channel_index":7},{"gateway_ids":{"gateway_id":"packetbroker"},"packet_broker":{"message_id":"01FM217XFW00Q1DXHWFG1AESZP","forwarder_net_id":"000013","forwarder_tenant_id":"ttnv2","forwarder_cluster_id":"ttn-v2-eu-3","forwarder_gateway_eui":"3133303725006A00","forwarder_gateway_id":"eui-3133303725006a00","home_network_net_id":"000013","home_network_tenant_id":"ttn","home_network_cluster_id":"eu1.cloud.thethings.network"},"time":"2021-11-09T09:43:00.139179Z","rssi":-99,"channel_rssi":-99,"snr":8.25,"location":{"latitude":39.47876459,"longitude":-0.33391551},"uplink_token":"eyJnIjoiWlhsS2FHSkhZMmxQYVVwQ1RWUkpORkl3VGs1VE1XTnBURU5LYkdKdFRXbFBhVXBDVFZSSk5GSXdUazVKYVhkcFlWaFphVTlwU25aVmEwWndUMVJOTUZZd2RGTlhhMUpKVDFoQ2NVbHBkMmxrUjBadVNXcHZhVkp

zYkUxalZUaDZVMjFhZFZOSVJsWlpNbTgxVjBkd2VGSlhjRkphZVVvNUxucHhUa2xMYTJabmJubFBOVTV2UWpneWVXeFZRMEV
1ZVZndGNXdFBWMnBxUzFkb2NWOUdhaTVPVjNSMlltTlJRWEpOVTBOck1IaG9RM0pwYTFOTldFNDFTRmRJZEdSRmJYUmhN
VXhPVkd4M2RtTlpiMjVNTkdFM1dUVmZWbVEzUVVoWlUwTnFhVTFEYlVoU1NrWkRSR1psUTNWMlNXbEhkRmRxWmsxVmJH
TjFXV3gwVkhoWGJIUjBSRlptV1RocVJucGphRU5NVVZocFZEQmZXa2RMWWtKdE5GUkxTMWRJUkcxeFZYSXliSGhzTTBaV1dt
MWpRMkpMY0U1SFoycFZhRzA0WkhKMUxYZHFiREpvVTJRM05FUkhkMnR1ZW10ckxtc3hhV2RPYmpkYWIzY3lTRmRZZGxOV
mEwOWlaa0U9IiwiYSI6eyJmbmlkIjoiMDAwMDEzIiwiZnRpZCI6InR0bnYyIiwiZmNpZCI6InR0bi12Mi1ldS0zIn19"}],"settings":{"data
_rate":{"lora":{"bandwidth":125000,"spreading_factor":12}},"coding_rate":"4/5","frequency":"867900000","timestamp":4257770316,"time
":"2021-11-09T09:42:36.723671Z"},"received_at":"2021-11-
09T09:42:36.751259886Z","consumed_airtime":"1.482752s","network_ids":{"net_id":"000013","tenant_id":"ttn","cluster_id":"ttn-eu1"}}}

El código correspondiente al ejercicio se encuentra en el fichero ttn.py

Código de la práctica

El código de los ficheros de la práctica será adjuntado en el momento de la entrega, pero también se puede encontrar en [mi repositorio de GitHub de la asignatura](#).