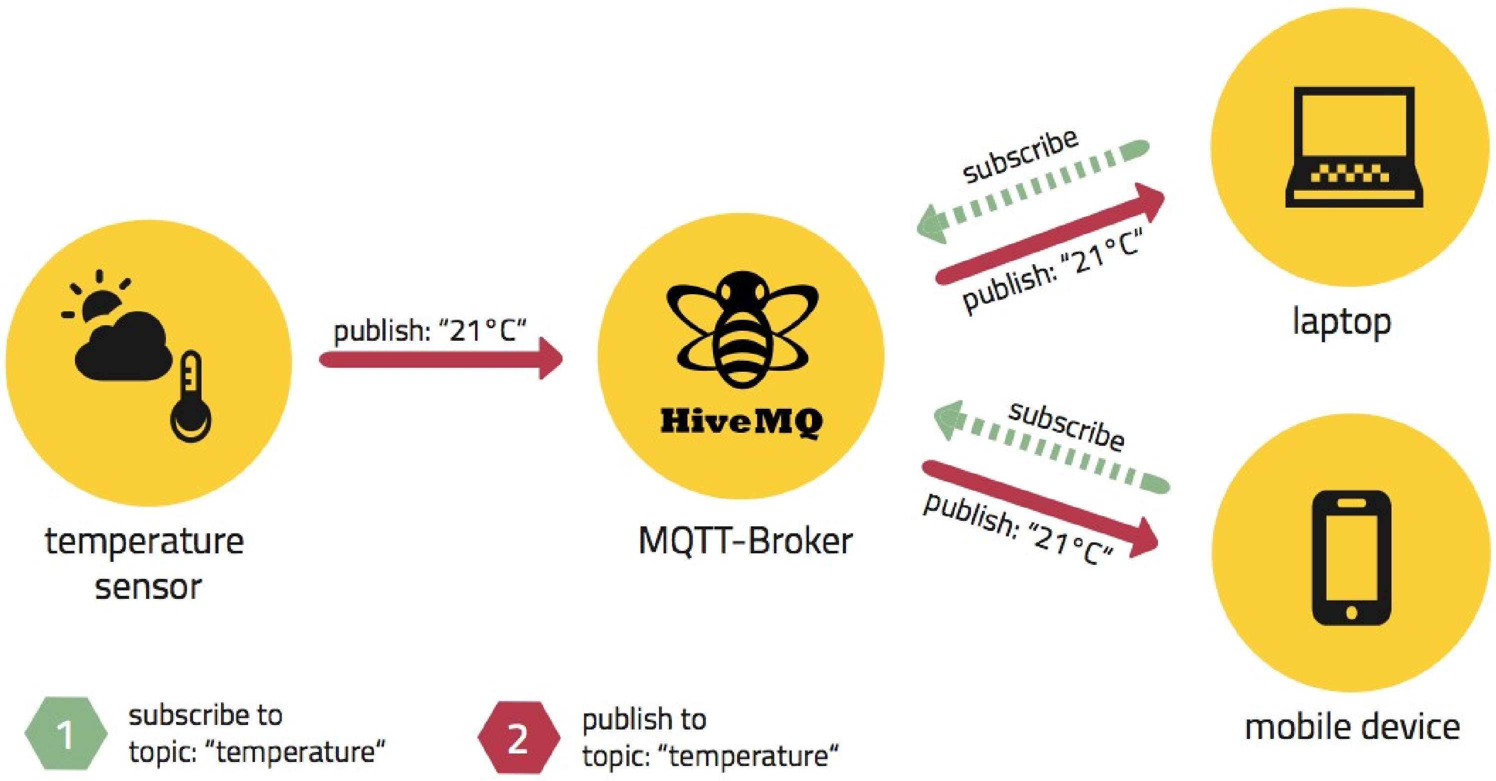
mosquitto

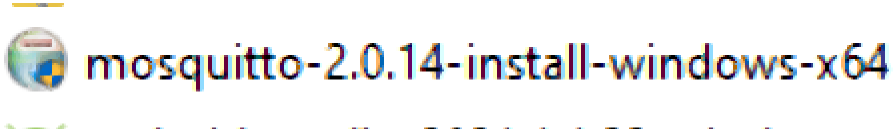
Características

- **MQTT**(Message Queuing Telemetry Transport) es un protocolo de comunicación ligero M2M(Machine2Machine).
- Eclipse **Mosquitto** es una implementación Open Source de este protocolo ampliamente usada.
- La arquitectura de una aplicación que utilice MQTT tiene: Un servidor MQTT que hace de broker de mensajes(Donde se conectan los clientes que publican o se suscriben a topics), los clientes que pueden tener un rol publicador, subscriptor o ambos.
- Un publicador es un componente que se dedica a publicar mensajes a un topic. Un subscriptor es un componente que se subscribe a uno o varios topics y cada vez que algún publicador publica algo sobre ese topic recibe el mensaje.

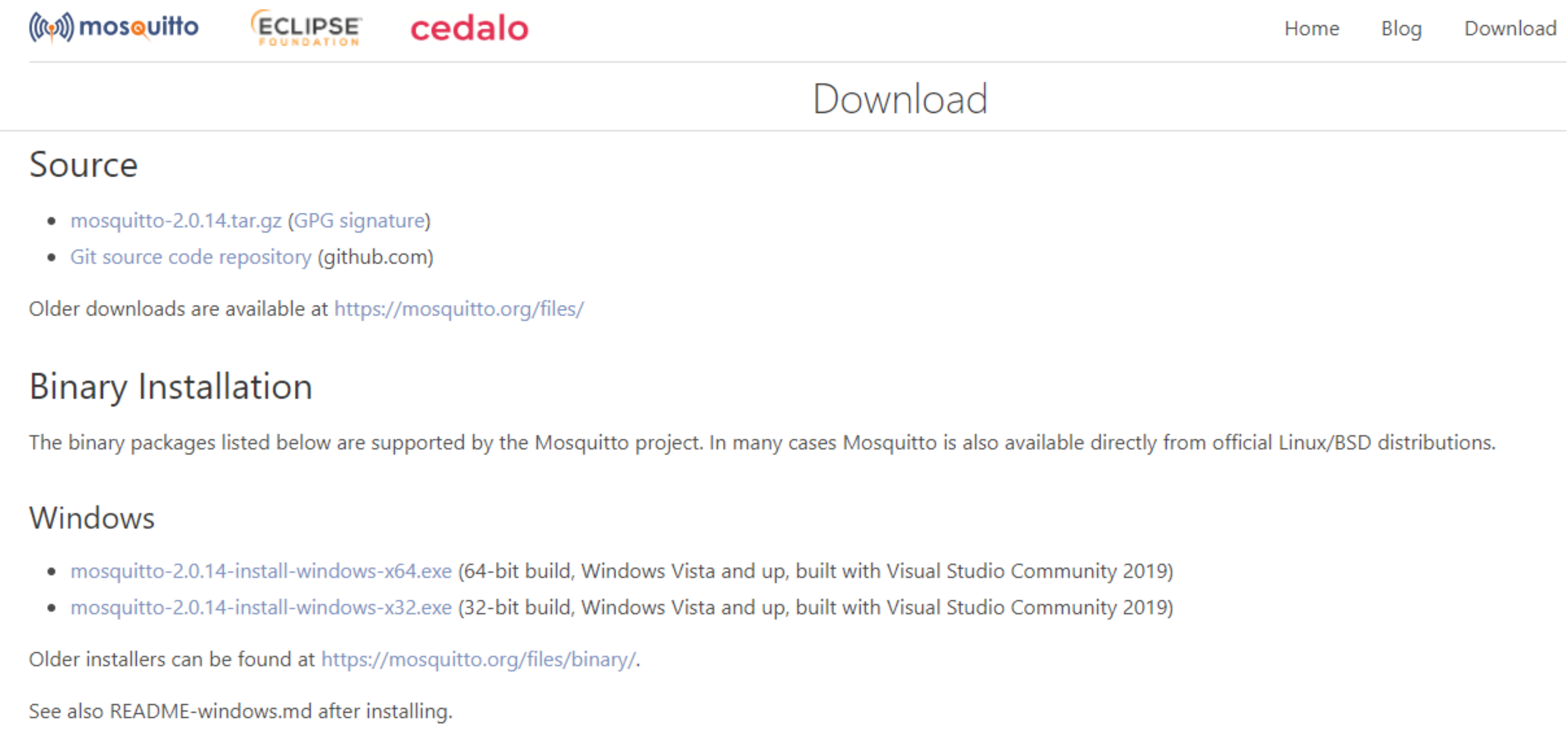
Arquitectura



El instalador {https://mosquitto.org/download/}



mosquitto-2.0.14-install-windows-x64



Download

Source

- mosquitto-2.0.14.tar.gz (GPG signature)
- Git source code repository (github.com)

Older downloads are available at https://mosquitto.org/files/

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- mosquitto-2.0.14-install-windows-x64.exe (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- mosquitto-2.0.14-install-windows-x32.exe (32-bit build, Windows Vista and up, built with Visual Studio Community 2019)

Older installers can be found at https://mosquitto.org/files/binary/.

See also README-windows.md after installing.

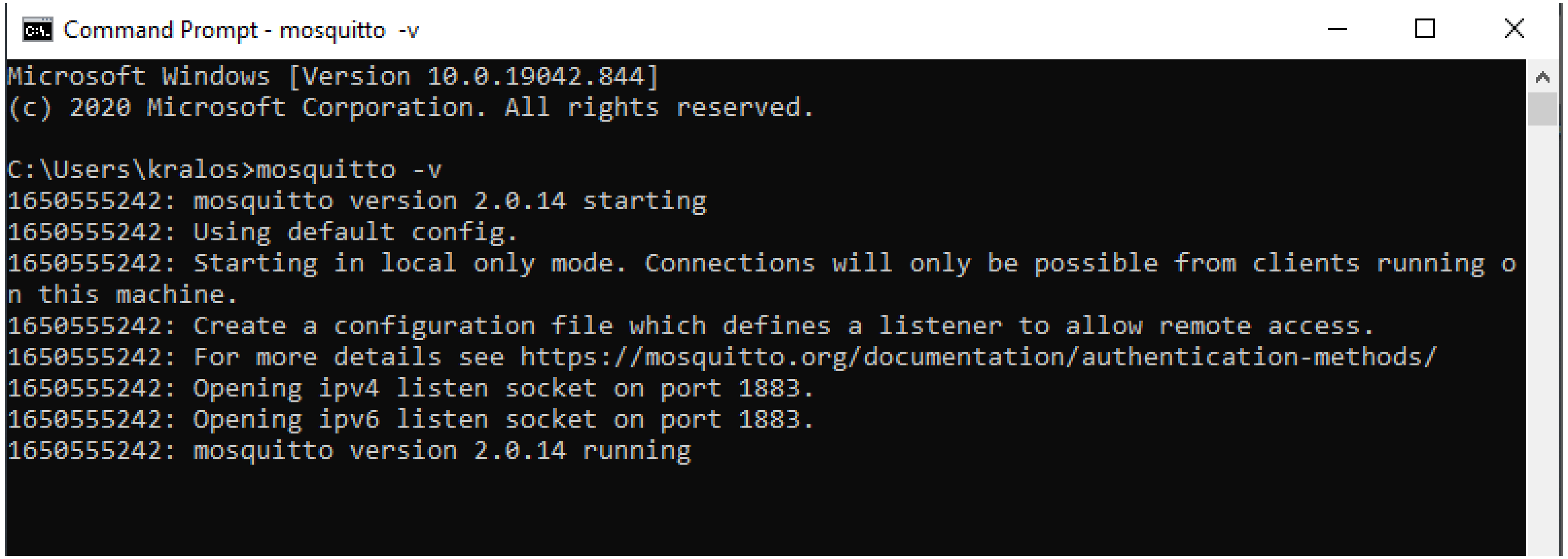
Arrancar Mosquitto

- mosquitto -v

- mosquitto -c mosquitto.conf
  - listener 1883 127.0.0.1
    allow_anonymous true

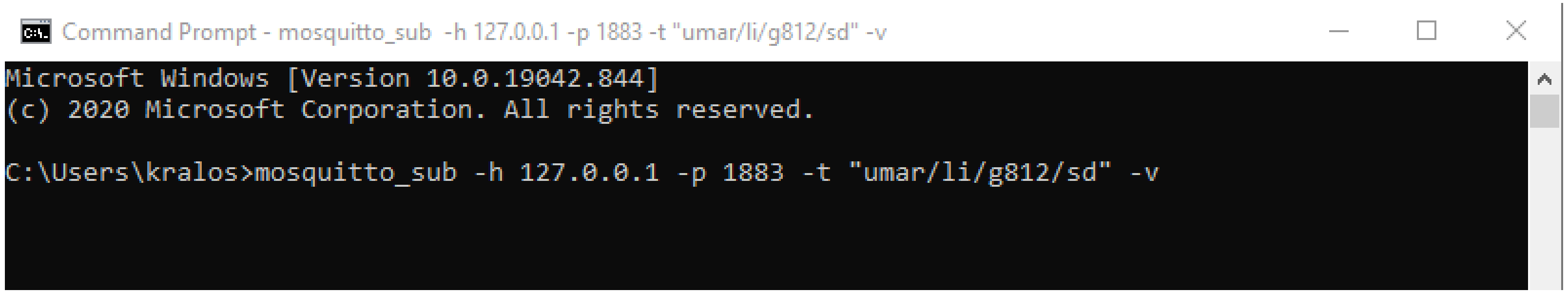
- Ctrl + c

# Iniciamos el Broker

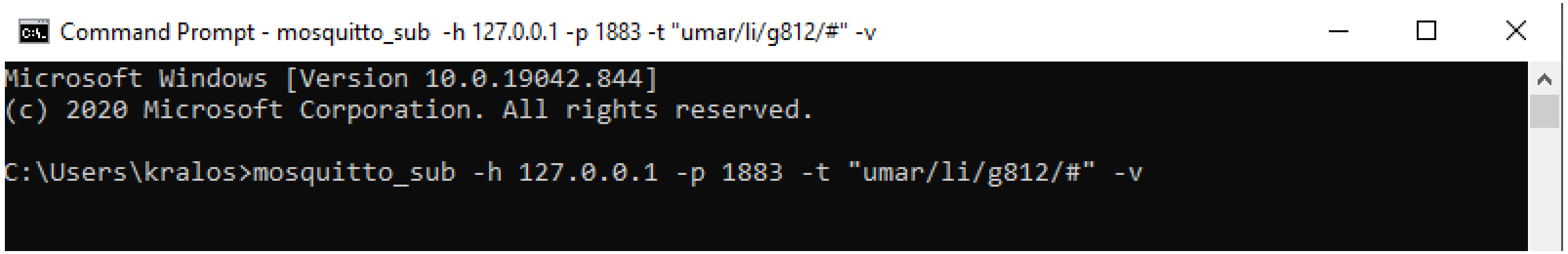


```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto -v
1650555242: mosquitto version 2.0.14 starting
1650555242: Using default config.
1650555242: Starting in local only mode. Connections will only be possible from clients running o
n this machine.
1650555242: Create a configuration file which defines a listener to allow remote access.
1650555242: For more details see https://mosquitto.org/documentation/authentication-methods/
1650555242: Opening ipv4 listen socket on port 1883.
1650555242: Opening ipv6 listen socket on port 1883.
1650555242: mosquitto version 2.0.14 running
```

# Configuramos subscriptores

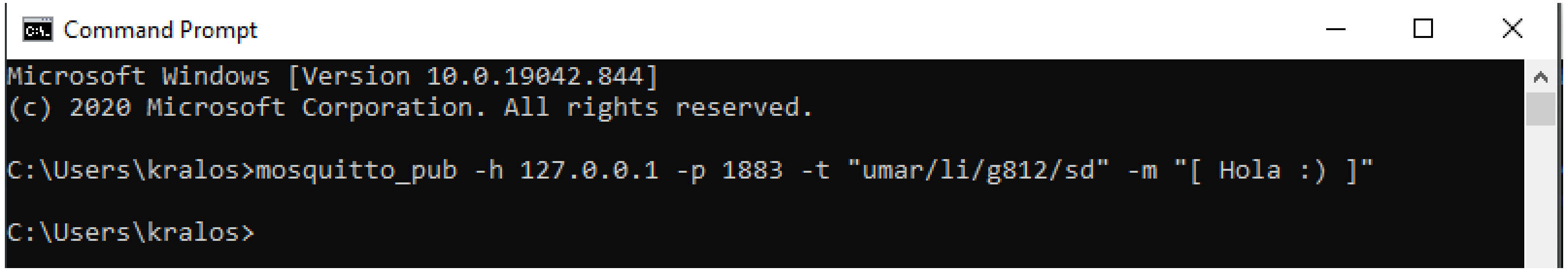

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v
```

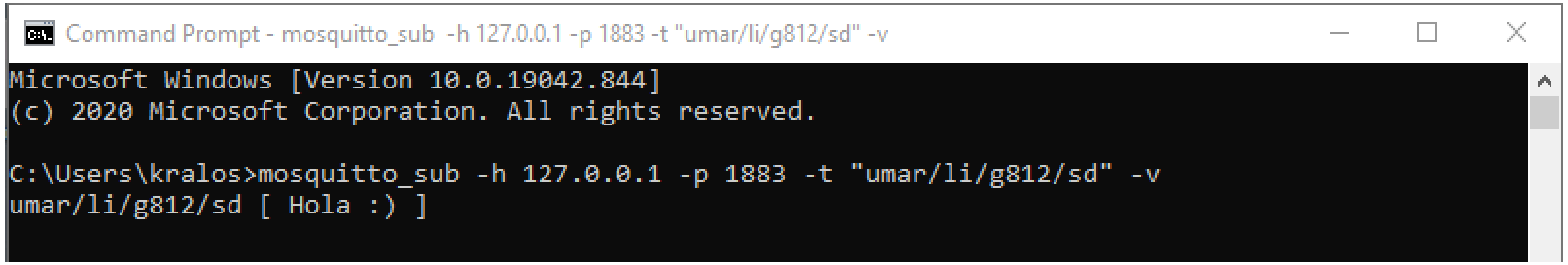

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v
```

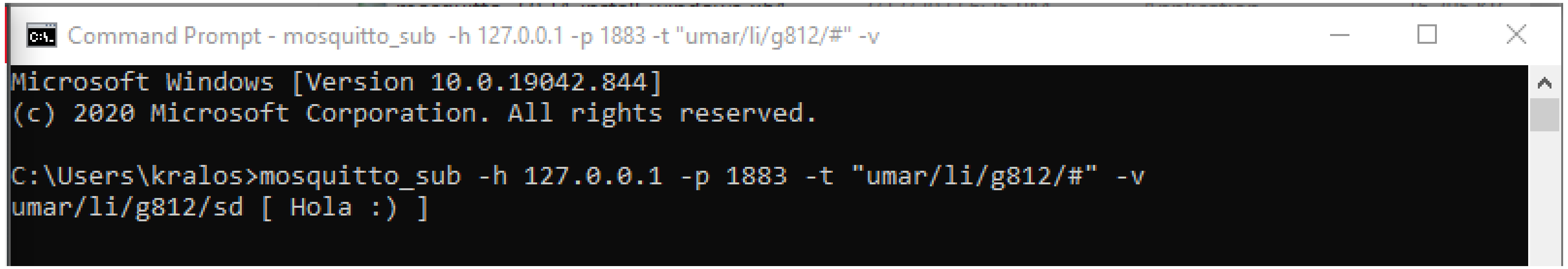
# Publicamos { umar/li/g812/sd }



```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_pub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -m "[ Hola :) ]"

C:\Users\kralos>
```

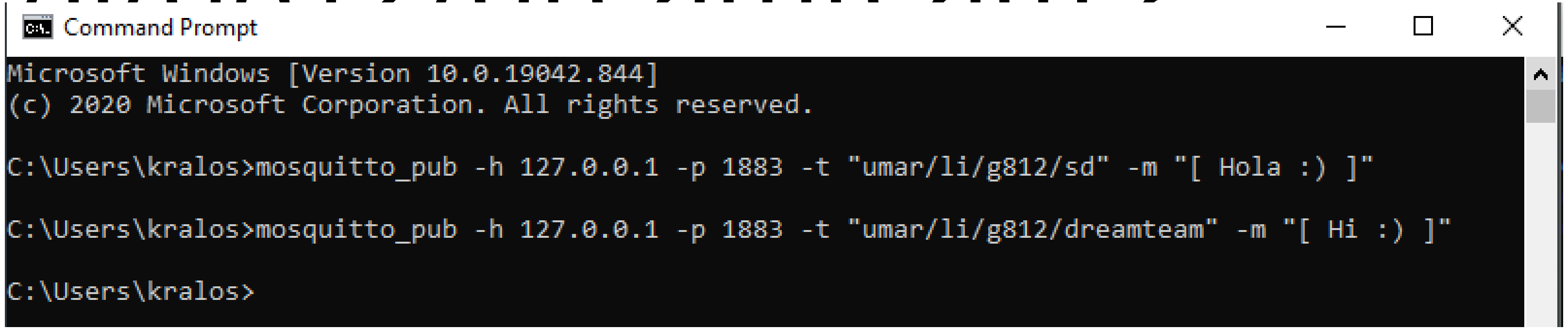

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v
umar/li/g812/sd [ Hola :) ]
```

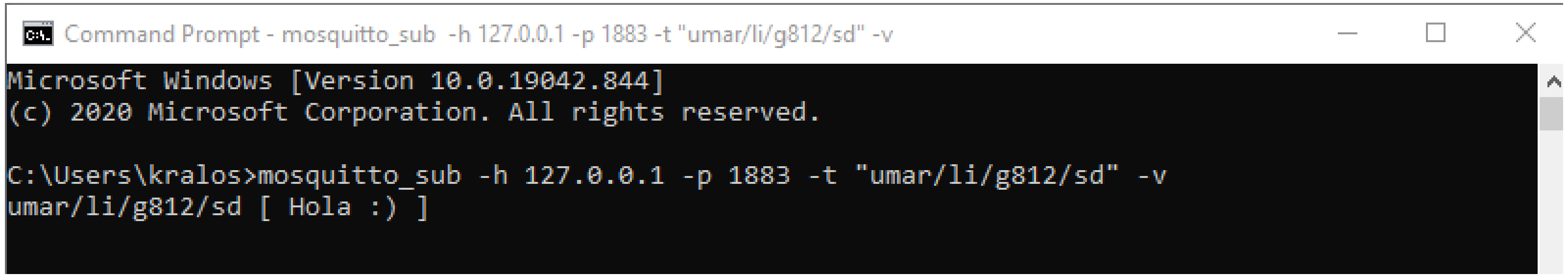

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v
umar/li/g812/sd [ Hola :) ]
```

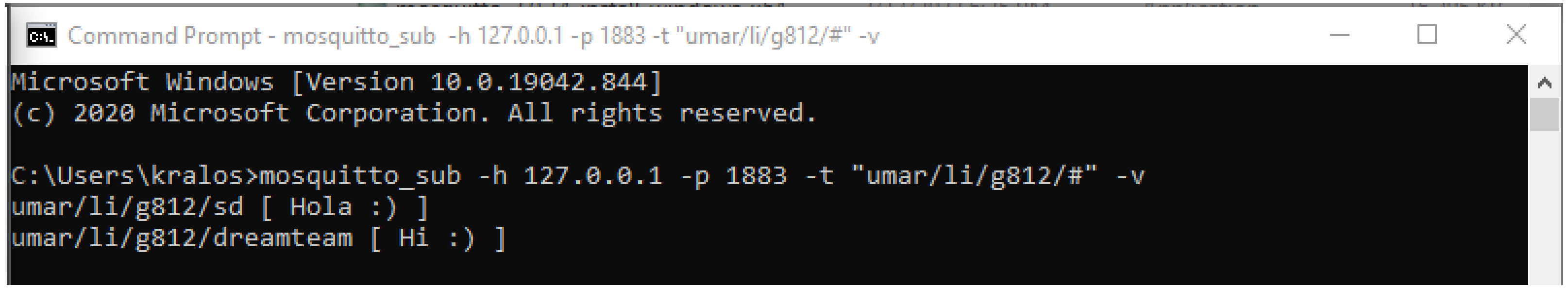
Publicamos { umar/li/g812/dreamteam }



```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_pub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -m "[ Hola :) ]"

C:\Users\kralos>mosquitto_pub -h 127.0.0.1 -p 1883 -t "umar/li/g812/dreamteam" -m "[ Hi :) ]"

C:\Users\kralos>
```

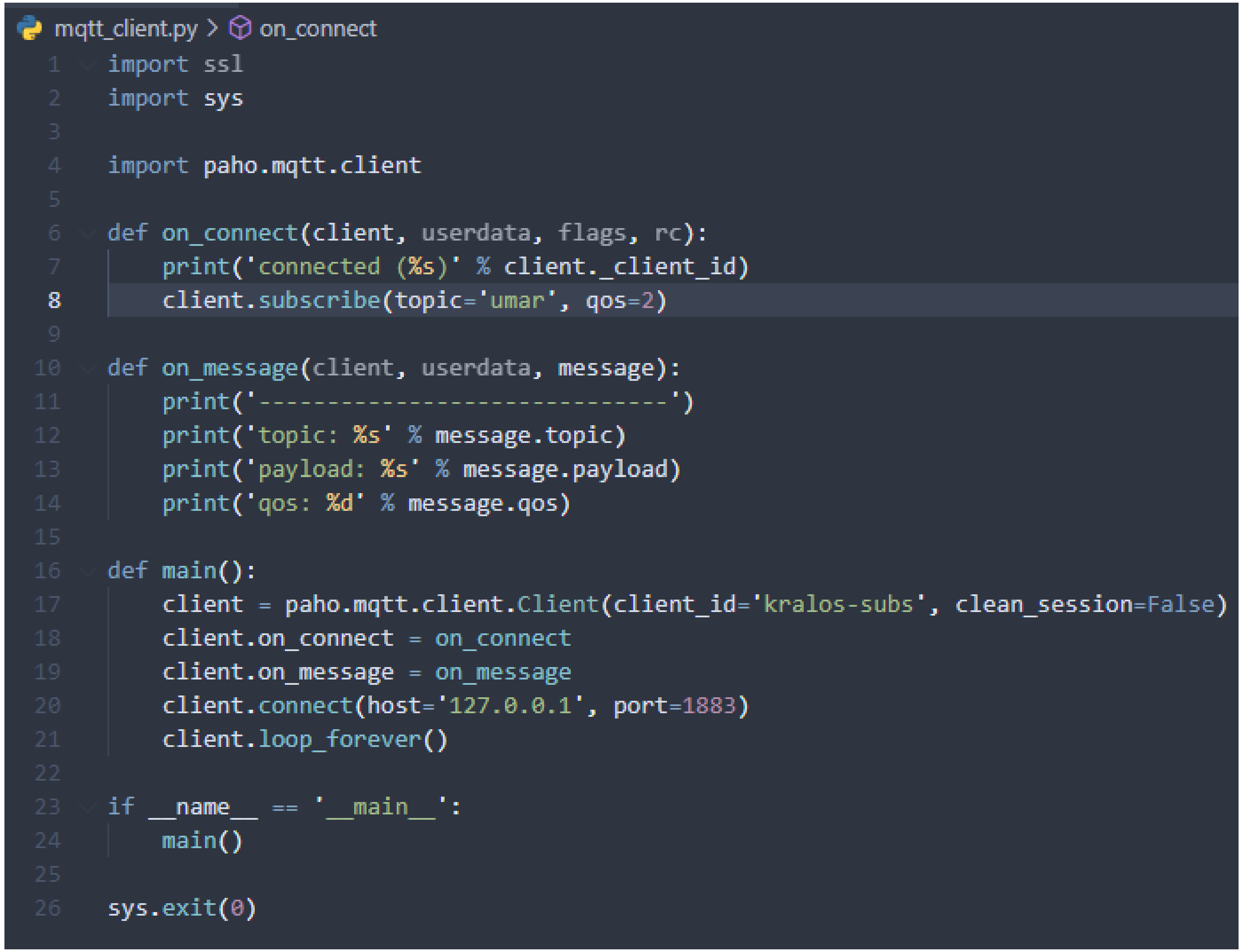

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v
umar/li/g812/sd [ Hola :) ]
```

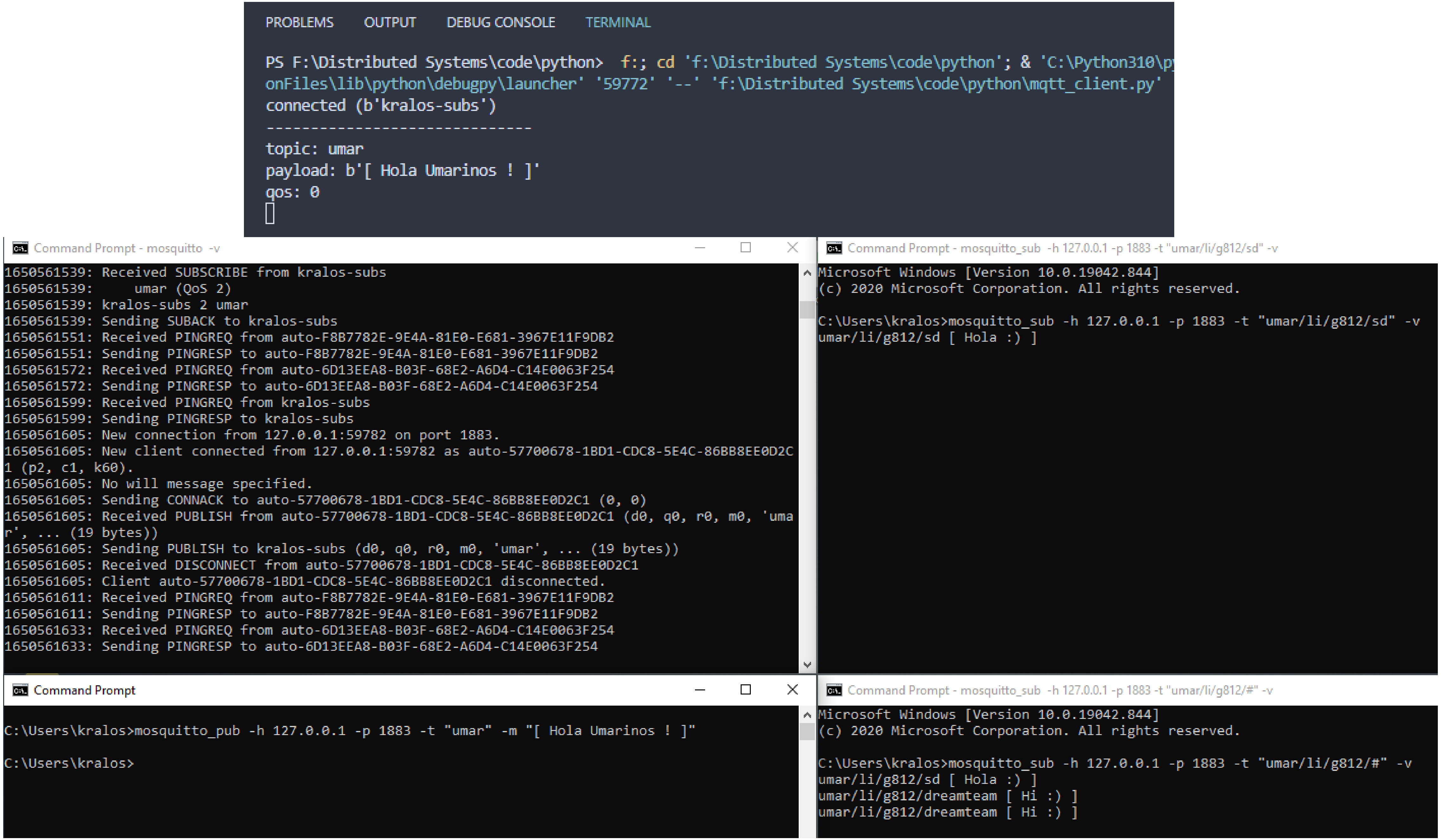

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v
umar/li/g812/sd [ Hola :) ]
umar/li/g812/dreamteam [ Hi :) ]
```

# con python



```python
import ssl
import sys

import paho.mqtt.client

def on_connect(client, userdata, flags, rc):
    print('connected (%s)' % client._client_id)
    client.subscribe(topic='umar', qos=2)

def on_message(client, userdata, message):
    print('------------------------------')
    print('topic: %s' % message.topic)
    print('payload: %s' % message.payload)
    print('qos: %d' % message.qos)

def main():
    client = paho.mqtt.client.Client(client_id='kralos-subs', clean_session=False)
    client.on_connect = on_connect
    client.on_message = on_message
    client.connect(host='127.0.0.1', port=1883)
    client.loop_forever()

if __name__ == '__main__':
    main()

sys.exit(0)
```



```
PS F:\Distributed Systems\code\python>  f:; cd 'f:\Distributed Systems\code\python'; & 'C:\Python310\py
onFiles\lib\python\debugpy\launcher' '59772' '--' 'f:\Distributed Systems\code\python\mqtt_client.py'
connected (b'kralos-subs')
-----------------------------
topic: umar
payload: b'[ Hola Umarinos ! ]'
qos: 0
```

```
1650561539: Received SUBSCRIBE from kralos-subs
1650561539:     umar (QoS 2)
1650561539: kralos-subs 2 umar
1650561539: Sending SUBACK to kralos-subs
1650561551: Received PINGREQ from auto-F8B7782E-9E4A-81E0-E681-3967E11F9DB2
1650561551: Sending PINGRESP to auto-F8B7782E-9E4A-81E0-E681-3967E11F9DB2
1650561572: Received PINGREQ from auto-6D13EEA8-B03F-68E2-A6D4-C14E0063F254
1650561572: Sending PINGRESP to auto-6D13EEA8-B03F-68E2-A6D4-C14E0063F254
1650561599: Received PINGREQ from kralos-subs
1650561599: Sending PINGRESP to kralos-subs
1650561605: New connection from 127.0.0.1:59782 on port 1883.
1650561605: New client connected from 127.0.0.1:59782 as auto-57700678-1BD1-CDC8-5E4C-86BB8EE0D2C
1 (p2, c1, k60).
1650561605: No will message specified.
1650561605: Sending CONNACK to auto-57700678-1BD1-CDC8-5E4C-86BB8EE0D2C1 (0, 0)
1650561605: Received PUBLISH from auto-57700678-1BD1-CDC8-5E4C-86BB8EE0D2C1 (d0, q0, r0, m0, 'uma
r', ... (19 bytes))
1650561605: Sending PUBLISH to kralos-subs (d0, q0, r0, m0, 'umar', ... (19 bytes))
1650561605: Received DISCONNECT from auto-57700678-1BD1-CDC8-5E4C-86BB8EE0D2C1
1650561605: Client auto-57700678-1BD1-CDC8-5E4C-86BB8EE0D2C1 disconnected.
1650561611: Received PINGREQ from auto-F8B7782E-9E4A-81E0-E681-3967E11F9DB2
1650561611: Sending PINGRESP to auto-F8B7782E-9E4A-81E0-E681-3967E11F9DB2
1650561633: Received PINGREQ from auto-6D13EEA8-B03F-68E2-A6D4-C14E0063F254
1650561633: Sending PINGRESP to auto-6D13EEA8-B03F-68E2-A6D4-C14E0063F254
```

Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v

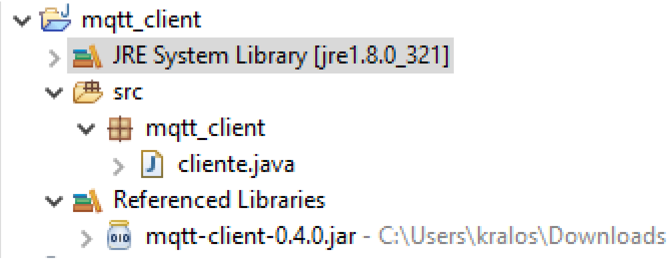
```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/sd" -v
umar/li/g812/sd [ Hola :) ]
```

```
C:\Users\kralos>mosquitto_pub -h 127.0.0.1 -p 1883 -t "umar" -m "[ Hola Umarinos ! ]"

C:\Users\kralos>
```

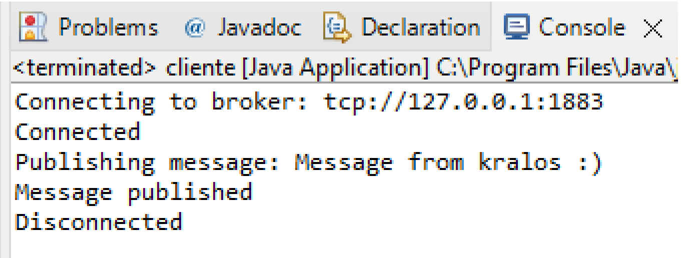
Command Prompt - mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v

```
Microsoft Windows [Version 10.0.19042.844]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kralos>mosquitto_sub -h 127.0.0.1 -p 1883 -t "umar/li/g812/#" -v
umar/li/g812/sd [ Hola :) ]
umar/li/g812/dreamteam [ Hi :) ]
umar/li/g812/dreamteam [ Hi :) ]
```

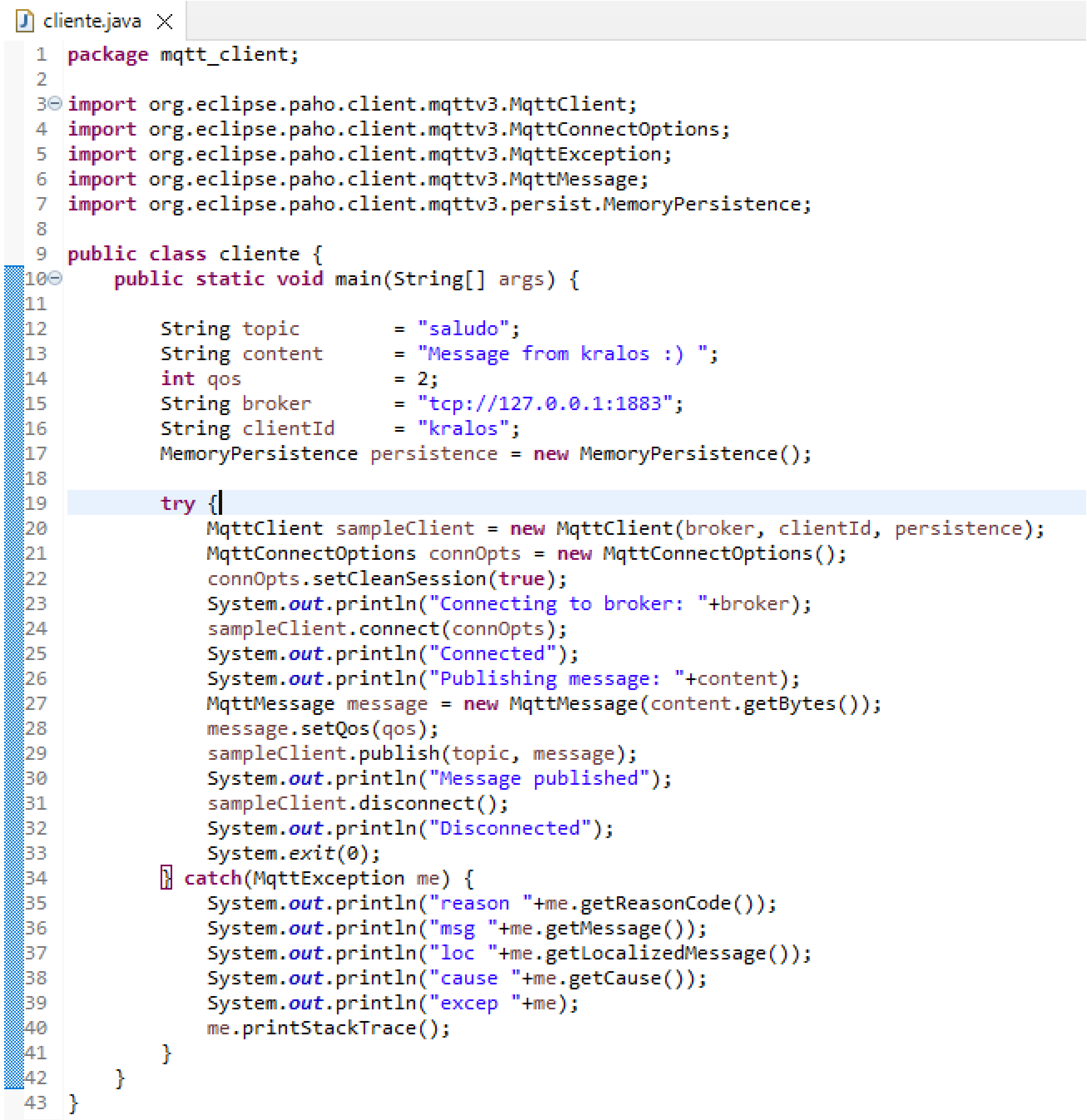
# con java



```
mqtt_client
  JRE System Library [jre1.8.0_321]
  src
    mqtt_client
      cliente.java
  Referenced Libraries
    mqtt-client-0.4.0.jar - C:\Users\kralos\Downloads
```



```
<terminated> cliente [Java Application] C:\Program Files\Java\
Connecting to broker: tcp://127.0.0.1:1883
Connected
Publishing message: Message from kralos :)
Message published
Disconnected
```



```java
package mqtt_client;

import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;

public class cliente {
    public static void main(String[] args) {

        String topic        = "saludo";
        String content      = "Message from kralos :) ";
        int qos             = 2;
        String broker       = "tcp://127.0.0.1:1883";
        String clientId     = "kralos";
        MemoryPersistence persistence = new MemoryPersistence();

        try {
            MqttClient sampleClient = new MqttClient(broker, clientId, persistence);
            MqttConnectOptions connOpts = new MqttConnectOptions();
            connOpts.setCleanSession(true);
            System.out.println("Connecting to broker: "+broker);
            sampleClient.connect(connOpts);
            System.out.println("Connected");
            System.out.println("Publishing message: "+content);
            MqttMessage message = new MqttMessage(content.getBytes());
            message.setQos(qos);
            sampleClient.publish(topic, message);
            System.out.println("Message published");
            sampleClient.disconnect();
            System.out.println("Disconnected");
            System.exit(0);
        } catch(MqttException me) {
            System.out.println("reason "+me.getReasonCode());
            System.out.println("msg "+me.getMessage());
            System.out.println("loc "+me.getLocalizedMessage());
            System.out.println("cause "+me.getCause());
            System.out.println("excep "+me);
            me.printStackTrace();
        }
    }
}
```