Sistemas Autónomos

Jéssica Lemos (a82061)

1 Introdução

Neste exercício individual foi pedido para que fosse implementado um programa em Java ou web para subscrever e publicar em feeds do Adafruit IO. Pelo que neste pequeno documento será descrita a implementação bem como apresentado o resultado desta implementação em Java.

2 Implementação

Primeiro é necessário conectar o MQTT definindo as propriedades da mesma, como o *username*, *password* e o *server URI*. E ainda definir o identificador do cliente para uma sessão persistente, com o intuito de quando este se reconectar as informações ficarem disponíveis imediatamente.

Depois de obter a coneção podemos subscrever o feed, pelo que sempre que é enviada uma mensagem para o mesmo o cliente deverá ser notificado. Assim para além de termos de fazer *subscribe* foi necessário criar um mecanismo *callback* que permite a aplicação notificar quando acontece um evento assíncrono, tal como o envio de uma mensagem. Desta forma, implementei a classe *MyCallback* com os respetivos métodos tal como podemos observar no seguinte excerto de código da mesma.

```
public class MyCallback implements MqttCallback {
    //metodo connectionLost

    public void messageArrived(String topic, MqttMessage msg) {
        System.out.println(String.format("Message received: [%s] %s\n", topic, new
            ↪ String(msg.getPayload())));
    }

    //metodo deliveryComplete
}
```

De seguida recorrendo à classe *scanner* do Java é possível ler o input no terminal que será a mensagem a publicar no feed através do *publish*. Nesta implementação foi permitido a realização de várias publicações se assim for desejado, pelo que para tal é lido o input até que este seja um pedido para desconectar tal como é possível observar no seguinte excerto de código da classe *MQTT_Test*.

```
    MyCallback callback = new MyCallback();
    mqtt_client.setCallback(callback);
    mqtt_client.subscribe(topic,qos);
    Scanner sn = new Scanner(System.in);
    while(true) {
        text = sn.nextLine();
        if(text.equals("disconnect")){
            mqtt_client.disconnect();
            System.out.println("Disconnected");
            System.exit(0);
        }
        MqttMessage message = new MqttMessage(text.getBytes());
        message.setQos(qos);
        mqtt_client.publish(topic, message);
    }
```

3 Resultados

Com a implementação abordada anteriormente torna-se assim possível subscrever e publicar as vezes que pretendemos. Tendo em conta que subscrevemos o feed recebemos notificações das publicações. No exemplo apresentado na Figura 1 conseguimos verificar que podemos indicar a mensagem a ser publicada e recebemos notificação tanto da mensagem por nós publicada ("Hello from Java!") como uma publicada através do Adafruit IO ("Hello there!"). Seria possível realizarmos mais publicações e receber notificações até à desconexão que é feita quando o indicamos, escrevendo no terminal "disconnect".

```
Connection to broker: tcp://io.adafruit.com:1883
Hello from Java!
Publishing message: Hello from Java!
Message published
Message received: [jessicalemos/feeds/sensorfeed] Hello from Java!
Message received: [jessicalemos/feeds/sensorfeed] Hello there!
disconnect
Disconnected

Process finished with exit code 0
```

Figura 1: Resultado obtido no terminal

A publicação feita através do terminal bem como a feita através do Adafruit IO encontra-se assim no feed deste site, tal como podemos verificar na Figura 2.

Created at	Value
2020/03/25 6:50:17pm	Hello there!
2020/03/25 6:50:02pm	Hello from Java!

Figura 2: Resultado obtido no feed *sensorfeed*