

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

Sistemas Distribuídos – Turmas 01 e 02 – 2023/1

Prof. Rodolfo da Silva Villaça – [rodolfo.villaca@ufes.br](mailto:rodolfo.villaca@ufes.br)

Monitor: Eduardo M. Moraes Sarmento – [eduardo.sarmento@ufes.br](mailto:eduardo.sarmento@ufes.br)

Laboratório IV – Comunicação Indireta (*Publish/Subscribe*)

## 1. Objetivos

- Experimentar a implementação de sistemas de comunicação indireta por meio de *middleware Publish/Subscribe* (Pub/Sub) e Filas de Mensagens (*Message Queues*);
- Sincronizar a troca de mensagens entre os componentes do sistema;
- Utilizar o *broker* EMQX MQTT para gerenciamento da fila de mensagens na implementação de sistemas distribuídos.

## 2. Conceitos Básicos

*Publish/Subscribe* é um padrão arquitetural onde existem os *Publishers* (Publicadores) que enviam as mensagens e os *Subscribers* (Assinantes) que recebem as mensagens. Em uma maneira mais prática, sempre que houver algum evento, o publicador vai enviar uma mensagem para que os assinantes sejam notificados. O padrão arquitetural *Publish/Subscribe* (Pub/Sub) se resume na imagem a seguir: existe um publicador, que vai enviar uma mensagem em um canal (*Channel*) que redistribui uma cópia da mensagem para cada assinante. Em nosso caso, o canal é representado por um conjunto de filas gerenciados pelo *broker*.



Nenhuma das partes se conhece, muito menos sabem os detalhes de suas implementações. Apenas conhecem o endereço do *broker*. O modelo de comunicação Pub/Sub pode trazer baixo acoplamento e facilidade de escalabilidade em um ecossistema de software. Por ser assíncrono não haverá problemas com *timeouts* ou quebras de processamento por erros encadeados.

EMQX<sup>1</sup> é um *broker* de mensagens com versão livre e de código aberto que implementa o protocolo MQTT (*Message Queue Telemetry Transport*). O protocolo MQTT fornece um método leve e simples para implementação do modelo de comunicação indireta usando Pub/Sub.

<sup>1</sup> <https://www.emqx.com/en/try?product=broker>

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA

O código-fonte dos exemplos usados neste laboratório estão disponíveis no repositório<sup>2</sup> da disciplina, especificamente na pasta lab4.

### 3. Atividade I: Uso do EMQX MQTT (MQTTX)

- a) Se necessário, faça a instalação local do *broker* MQTTX, ou busque o endereço do *broker* disponível no laboratório;
- b) Executar uma instância do *publisher* “LightPub-X” (*mqttlighpub.py*), em Python, que fará o envio de números aleatórios no intervalo [0..100] na fila “sd/X/light” do *broker* MQTTX, onde “X” é o número do seu grupo. A ideia aqui é simular dois sensores de luminosidade em um dispositivo IoT;
- c) Executar uma instância do *publisher* “TempPub-X” (*mqtttempub.py*), em Python, que fará o envio de números aleatórios no intervalo [15..40] na fila “sd/X/temp” do *broker* MQTTX, onde “X” é o número do seu grupo. A ideia aqui é simular um sensor de temperatura em um dispositivo IoT;
- d) Executar uma instância do *subscriber* “LightSub-X” (*mqttlighsub.py*), em Python, que assinará a fila “sd/X/light” do *broker* MQTTX, onde “X” é o número do seu grupo;
- e) Executar uma instância do *subscriber* “TempSub-X” (*mqtttempsub.py*), em Python, que assinará a fila “sd/X/temp” do *broker* MQTTX, onde “X” é o número do seu grupo;
- f) Executar uma instância do *subscriber* “LightTempSub-X” (*mqttlighsub.py*), em Python, que assinará as filas “sd/light” e “sd/temp” do *broker* MQTTX, onde “X” é o número do seu grupo;
- g) Experimente publicar mensagens alterando o código do *publisher* para diferentes valores de *qos* conforme tutorial disponível na documentação da biblioteca paho<sup>3</sup>. Observe os diferentes valores de *qos=1*, *qos=2*, *qos=3* para implementação de filas e recepção de mensagens *offline* por parte dos *subscribers*;
- h) Observe (ou, provoque, se necessário) a interferência das publicações dos demais grupos no seu programa assinante (*subscriber*): publique com `<group_id>` de outros grupos no mesmo *broker*;
- g) Reinicie todos os processos passando como parâmetro o endereço de um *broker* público, cujo nome é “*broker.emqx.io*” e refaça o experimento. Busque por apps para celulares Android/iPhone que sejam publicadores/ assinantes MQTT e experimente-os, interagindo com seus códigos em Python.

<sup>2</sup> <https://github.com/nerds-ufes/sistdist.git>

<sup>3</sup> <http://www.steves-internet-guide.com/publishing-messages-mqtt-client/>