

# Computação Distribuída

## MEIC e MEIM

### Laboratório 2

**Objetivo:** Experimentação de modelos de comunicação por eventos/mensagens utilizando um servidor RabbitMq alojado num container Docker e desenvolvimento, em Java usando a API *amqp-client*, de aplicações Produtoras e Consumidoras de mensagens

- 1) Tal como já realizou em laboratório anterior, crie uma VM no seu projeto GCP com sistema CentOS 8 e OpenJDK 11
- 2) Instale o runtime Docker de acordo com o slide 13 do grupo (CD-04 VMs-Containers);
- 3) verifique que está tudo a funcionar executando o seguinte comando:

```
docker run fedora cat /etc/os-release
```

- 4) Garanta que a sua VM tem regras de Firewall para abrir os portos TCP/IP **5672** e **15672**
- 5) Execute um container que aloja um servidor (Broker) RabbitMq com o seguinte comando:

```
docker run -d --hostname rabbithost --name rabbitmq \
-p 5672:5672 -p 15672:15672 rabbitmq:management
```

- 6) Execute no seu browser a aplicação de gestão do Broker no seguinte URL:  
`http://<public IP da VM>:15672`
- 7) Faça download no Moodle do ficheiro Lab2-AppsRabbitMQ.zip que contém 3 projectos Maven (aplicação de configuração de Exchanges, Filas Bindings etc e aplicações de envio e receção de mensagens).
- 8) Usando as aplicações atrás referidas, com eventuais modificações, concretize os diferentes padrões de encaminhamento de mensagens.
  - a) Utilize a aplicação web no browser de gestão do RabbitMQ para ir confirmando as configurações que efetua.
- 9) **[Desafio]** Usando exclusivamente o modelo *Publish/Subscribe* e o servidor RabbitMQ implemente uma aplicação distribuída de troca de mensagens entre múltiplos utilizadores nos seguintes cenários:
  - a) Todos os utilizadores recebem todas as mensagens;
  - b) Um utilizador só recebe mensagens associadas a um único dos seguintes temas (desporto, economia, tecnologia)
  - c) Generalize em que cada utilizador pode receber mensagens classificadas com um conjunto de temas