

LABORATÓRIO PRÁTICO – RabbitMQ

Objetivo:

Criar um ambiente funcional com RabbitMQ, configurar uma *exchange* do tipo **fanout**, desenvolver um produtor e múltiplos consumidores, e visualizar o fluxo de mensagens dentro do painel de gerenciamento.

Pré-requisitos:

- **Docker** e **Docker Compose** instalados
- Qualquer ambiente de execução:
 - Máquina local
 - Máquina virtual
 - Ambiente em nuvem (AWS, Azure, Google Cloud, etc.)

O laboratório não depende de nenhum provedor específico.

Subindo o ambiente RabbitMQ:

Utilize o seguinte arquivo **docker-compose.yml** para disponibilizar o RabbitMQ com o painel de administração:

```
services:  
  rabbitmq:  
    image: rabbitmq:3-management  
    container_name: rabbitmq  
  
    ports:  
      - "5672:5672"  
      - "15672:15672"  
    environment:  
      RABBITMQ_DEFAULT_USER: admin  
      RABBITMQ_DEFAULT_PASS: admin
```

Comando para iniciar o serviço:

```
docker compose up -d
```

Acessando o painel de gerenciamento:

Abra no navegador

<http://localhost:15672>

Credenciais padrão (definidas no compose):

- Usuário: **admin**
- Senha: **admin**

O painel permitirá acompanhar filas, consumidores, conexões, canais e mensagens em tempo real.

Criando a exchange FANOUT

Você pode criá-la de duas maneiras:

Opção A — Pelo próprio painel

1. Acesse: **Exchanges**
2. Clique em **Add a new exchange**
3. Nome: **lab_fanout**
4. Tipo: **fanout**
5. Criar
- 6.

Opção B — Pelo código

```
channel.exchange_declare(exchange='lab_fanout', exchange_type='fanout')
```

Passos do Laboratório:

- 1) Subir o RabbitMQ com Docker
 - Garantir que o servidor está ativo e acessível no painel.
- 2) Criar duas filas
- 3) Associar as filas à exchange fanout
- 4) Implementar e executar dois consumidores:
 - Cada consumidor deve ficar escutando uma das filas criadas.
 - O objetivo é observar que **cada mensagem enviada será entregue a todos os consumidores.**
- 5) Implementar e rodar o produtor
 - O produtor deve publicar mensagens na exchange **lab_fanout**.
- 6) Verificar o painel: conexões, canais, consumidores e filas