

Taller: Integración de RabbitMQ con Apache Camel

Mateo Encalada

Link al repositorio: <https://github.com/MateoEncalada/TallerRabbitMQ>

Patrón de Integración Aplicado

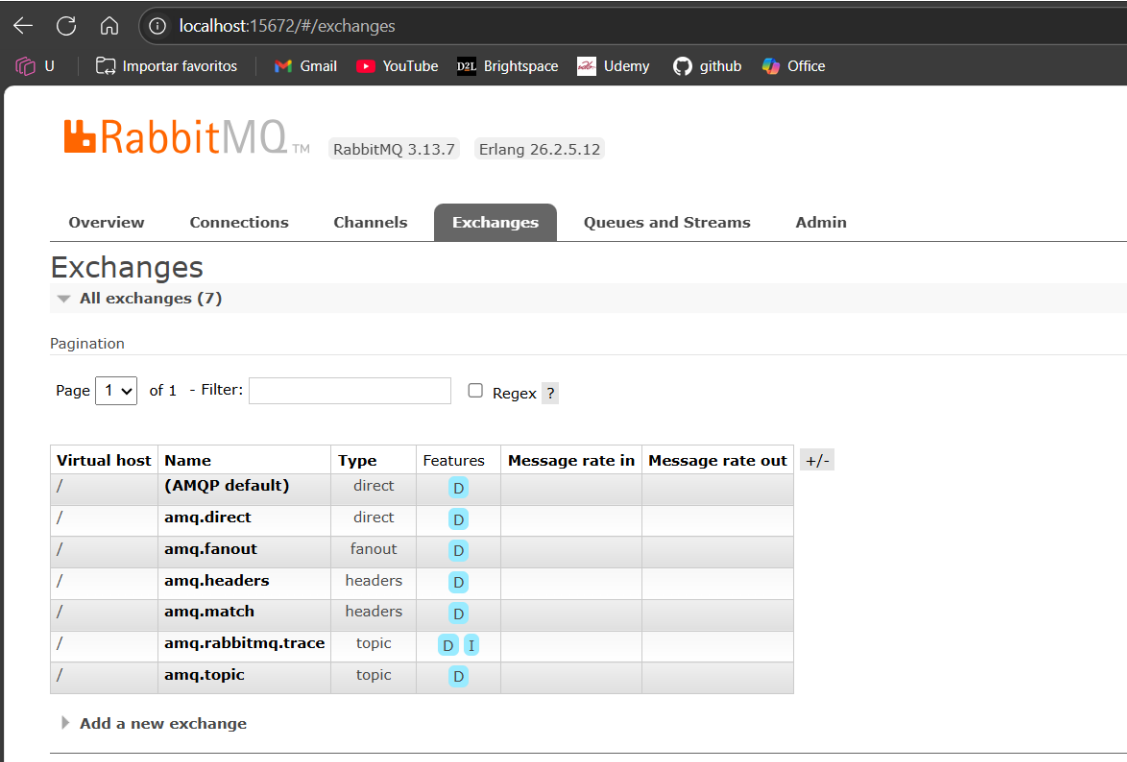
Durante el desarrollo de esta práctica se aplicó el **patrón de mensajería asincrónica**, el cual permite la comunicación entre sistemas sin necesidad de que estén disponibles al mismo tiempo. Este patrón se implementó utilizando **Apache Camel** como orquestador de rutas de integración y **RabbitMQ** como broker de mensajería.

La arquitectura se basó en dos rutas Camel:

- Una ruta **productora**, que genera mensajes periódicamente (cada 5 segundos) y los envía a una cola llamada test.camel.queue en RabbitMQ.
- Una ruta **consumidora**, que escucha esa misma cola y procesa los mensajes recibidos.

Este patrón permite la **separación completa entre productor y consumidor**, promoviendo la escalabilidad y tolerancia a fallos en sistemas distribuidos.

Antes




The screenshot shows the RabbitMQ Admin interface in a web browser. The address bar indicates the URL is localhost:15672/#/exchanges. The interface includes a navigation bar with tabs for Overview, Connections, Channels, Exchanges (selected), Queues and Streams, and Admin. The main content area is titled 'Exchanges' and shows a list of all exchanges (7). Below the title is a pagination section with 'Page 1 of 1' and a filter input. A table lists the exchanges with columns for Virtual host, Name, Type, Features, Message rate in, Message rate out, and a +/- button. The table contains 7 rows of exchange information.

Virtual host	Name	Type	Features	Message rate in	Message rate out	+/-
/	(AMQP default)	direct	D			
/	amq.direct	direct	D			
/	amq.fanout	fanout	D			
/	amq.headers	headers	D			
/	amq.match	headers	D			
/	amq.rabbitmq.trace	topic	D I			
/	amq.topic	topic	D			

Below the table, there is a link to 'Add a new exchange'.

Después

Virtual host	Name	Type	Features	Message rate in	Message rate out	+/-
/	(AMQP default)	direct	D			
/	amq.direct	direct	D			
/	amq.fanout	fanout	D			
/	amq.headers	headers	D			
/	amq.match	headers	D			
/	amq.rabbitmq.trace	topic	D I			
/	amq.topic	topic	D			
/	test.camel.queue	direct	D AD	0.20/s	0.20/s	


RabbitMQ 3.13.7 Erlang 26.2.5.12

Refreshed 2025-06-12 21:00:26
Refresh every 5 seconds

Virtual host
All

Cluster
rabbit@my-rabbit

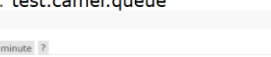
User
guest
Log out

Overview
Connections
Channels
Exchanges
Queues and Streams
Admin

Exchange: test.camel.queue

▼ Overview

Message rates last minute ⓘ



Publish (In)

0.40/s

Publish (Out)

0.40/s

Details

Type direct

Features durable: true
auto-delete: true

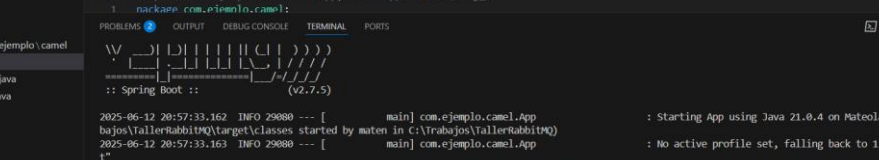
Policy

► Bindings

► Publish message

► Delete this exchange

[HTTP API](#)
[Documentation](#)
[Tutorials](#)
[New releases](#)
[Commercial edition](#)
[Commercial support](#)
[Discussions](#)
[Discord](#)
[Plugins](#)
[GitHub](#)



```

src > main > java > com > ejemplo > camel > App.java > App > main(String[])
  : nacker.com.cam.ejemplo.camel:

pom.xml
App.java
ConsumerRoute.java
ProducerRoute.java
test
target
pom.xml

main > java > com > ejemplo &
  : Spring Boot ::
  : (v2.7.5)

2025-06-12 20:57:33.162 INFO 29080 --- [main] com.ejemplo.camel.App
bajos:TallerRabbitMQ/target/classes started by maten in C:\Trabajos\TallerRabbitMQ)
2025-06-12 20:57:33.163 INFO 29080 --- [main] com.ejemplo.camel.App
t*
2025-06-12 20:57:34.058 WARN 29080 --- [main] o.a.c.c.rabbitmq.RabbitMQComponent
d. You should configure the hostname on the component
2025-06-12 20:57:34.128 INFO 29080 --- [main] o.a.c.i.m.impl.engine.AbstractCamelContext
2025-06-12 20:57:34.363 INFO 29080 --- [main] o.a.c.i.m.impl.engine.AbstractCamelContext
2025-06-12 20:57:34.364 INFO 29080 --- [main] o.a.c.i.m.impl.engine.AbstractCamelContext
2025-06-12 20:57:34.364 INFO 29080 --- [main] o.a.c.i.m.impl.engine.AbstractCamelContext
2025-06-12 20:57:34.364 INFO 29080 --- [main] o.a.c.i.m.impl.engine.AbstractCamelContext
3/mes (start:235ms)
2025-06-12 20:57:34.367 INFO 29080 --- [main] com.ejemplo.camel.App
2025-06-12 20:57:35.379 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:57:35.387 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:57:40.375 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:57:40.378 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:57:45.379 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:57:45.382 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:57:50.379 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:57:50.384 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:57:55.380 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:57:55.386 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:58:00.381 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:58:00.385 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:58:05.382 INFO 29080 --- [imer:]<generate> route2
2025-06-12 20:58:05.386 INFO 29080 --- [est.camel.queue] route1
2025-06-12 20:58:10.383 INFO 29080 --- [imer:]<generate> route2
  : Starting App using Java 21.0.4 on Matsolap with PID 29080 (C:\Vr
  : No active profile set, falling back to 1 default profile: "defau
  : The old syntax rabbitmq://hostname:port/exchangeName is deprecate
  : Apache Camel 3.20.2 (camel-1) is starting
  : Routes started (started:2)
  : Started route1 (rabbitmq://localhost/test.camel.queue)
  : Started route2 (timer:/generate)
  : Apache Camel 3.20.2 (camel-1) started in 397ms (build:25ms init:
  : Started App in 3.441 seconds (JVM running for 1.713)
  : Enviando: Mensaje generado en 2025-06-12 20:57:35
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:57:35
  : Enviando: Mensaje generado en 2025-06-12 20:57:40
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:57:40
  : Enviando: Mensaje generado en 2025-06-12 20:57:45
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:57:45
  : Enviando: Mensaje generado en 2025-06-12 20:57:50
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:57:50
  : Enviando: Mensaje generado en 2025-06-12 20:57:55
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:57:55
  : Enviando: Mensaje generado en 2025-06-12 20:58:00
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:58:00
  : Enviando: Mensaje generado en 2025-06-12 20:58:05
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:58:05
  : Enviando: Mensaje generado en 2025-06-12 20:58:10
  : Mensaje recibido: Mensaje generado en 2025-06-12 20:58:10
  
```

Desacoplamiento Productor-Consumidor

El desacoplamiento se logró mediante la introducción de **RabbitMQ como intermediario (Message Broker)** entre el productor y el consumidor.

Específicamente:

- El **productor Camel** publica los mensajes en una cola, sin saber quién o cuándo se consumirán.
- El **consumidor Camel** escucha esa cola y reacciona cuando hay mensajes disponibles.
- No existe dependencia directa entre ambos componentes, ni invocaciones directas, lo que permite ejecutar cada uno de forma independiente.

Además, ambos procesos pueden ejecutarse en **hilos, entornos o incluso máquinas diferentes**, sin afectar la funcionalidad general del sistema.

Ventajas Observadas durante la Práctica

Al realizar este taller se pudieron observar varias ventajas al usar una mensajería asincrónica, hubo un desacoplamiento total entre el productor y consumidor, también si el consumidor se detiene los mensajes seguirán almacenados en la cola hasta que vuelva a estar disponible, los logs de Apache Camel permiten observar claramente el flujo del mensaje desde su envío hasta su recepción. Una de las más importantes fue el uso práctico de herramientas como RabbitMQ y Camel.