FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Universidad de las Américas

INTEGRACION DE SISTEMAS

Taller – Configuración de RabbitMQ e Integración con Apache Camel

Camila Cabrera

Kristiany Cerón

Tito Jaramillo

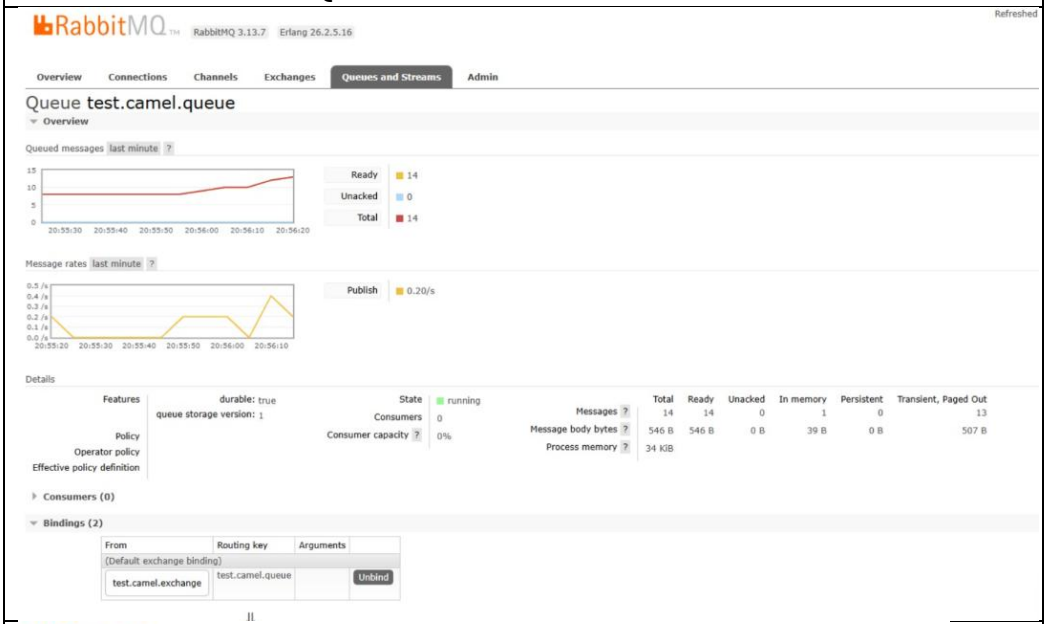
Camila Vega

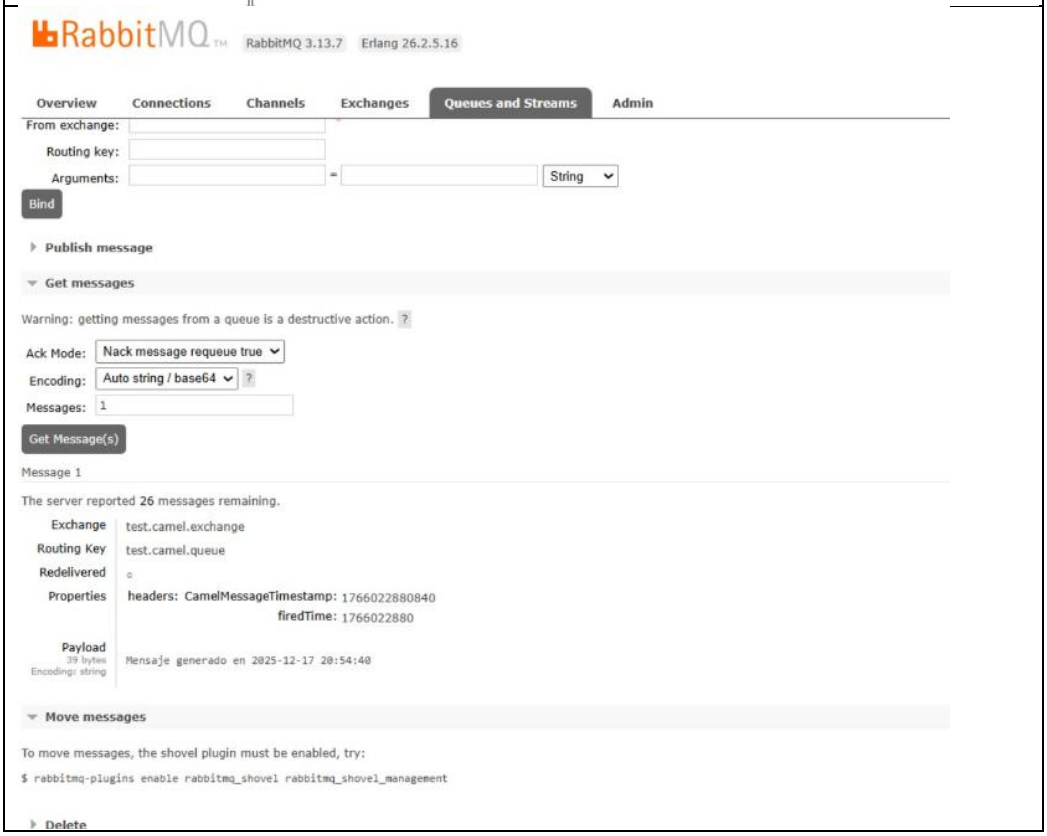
17 de diciembre del 2025

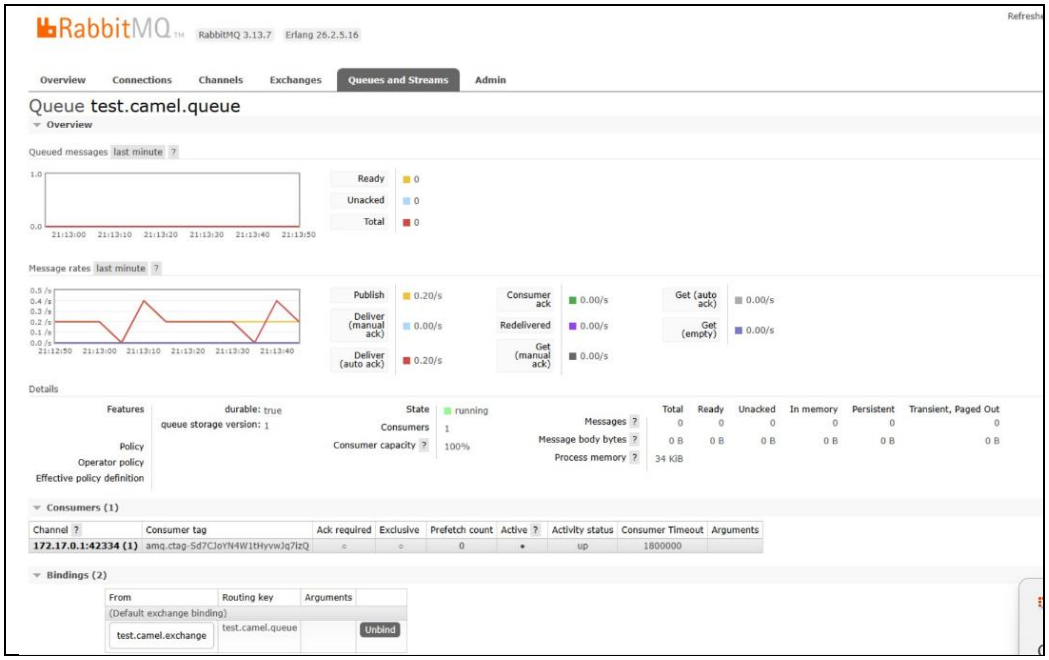
1. **Repositorio Github:**
   https://github.com/krissceron/EntregablesDeTaller.git

2. **Capturas de pantalla**

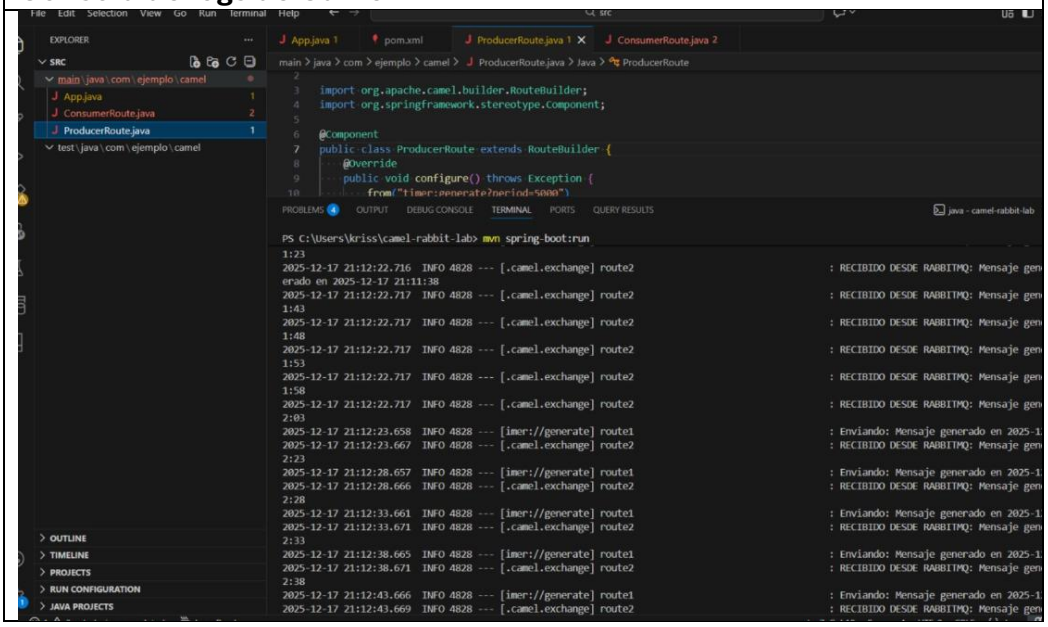
Consola de RabbitMQ







Consola de logs de Camel



1. **Qué patrón de integración se aplicó.**

   En el taller se aplicó el patrón de mensajería asincrónica (Asynchronous Messaging), específicamente el patrón Productor–Consumidor mediante un Message Broker.

   Este patrón permite que un productor envíe mensajes a una cola gestionada por un broker (RabbitMQ) sin necesidad de conocer directamente al consumidor. Apache Camel se encargó de orquestar el envío y la recepción de mensajes utilizando rutas de integración.

2. **Cómo se logró el desacoplamiento productor consumidor.**

   El desacoplamiento se logró mediante el uso de RabbitMQ como intermediario entre el productor y el consumidor.

   El productor Camel únicamente publica mensajes en una cola (test.camel.queue) sin saber quién los consumirá ni cuándo. Por su parte, el consumidor escucha esa cola y procesa los mensajes de forma independiente. Gracias a esto, ambos componentes pueden ejecutarse, modificarse o reiniciarse sin afectar directamente al otro, siempre que la cola esté disponible.

3. **Ventajas que observaron durante la práctica.**
   - Desacoplamiento de sistemas, ya que productor y consumidor no dependen entre sí.
   - Mayor escalabilidad, permitiendo agregar más consumidores sin cambiar el productor.
   - Tolerancia a fallos, porque los mensajes permanecen en la cola si el consumidor no está disponible.
   - Procesamiento asincrónico, lo que mejora el rendimiento y evita bloqueos.
   - Facilidad de monitoreo, gracias a la consola de administración de RabbitMQ.