

SIMULADOR TryRabbitMQ

1. Nombre del Exchange, tipo y bindings creados:

1.1 Nombre del Exchange: compute.direct

1.2 Tipo del Exchange: Direct

1.3 Bindings creados:

- ✓ **Exchange:** compute.direct
- ✓ **Queue:** compute_queue
- ✓ **Binding Key:** calculate

1.4 Routing keys usadas:

2. Routing keys usadas y colas que recibieron cada mensaje:

- ✓ **Routing keys usadas:** calculate (Utilizada para enrutar los mensajes del Producer al Exchange y luego a la Queue).
- ✓ **Colas que recibieron cada mensaje:** compute_queue

3. Capturas de pantalla del flujo observado.

The screenshot displays the TryRabbitMQ Simulator web application. The interface includes a legend on the left with icons for exchange (triangle), queue (square), producer (circle), and consumer (star). The main area shows a message flow diagram: a producer sends a message to an exchange named 'compute.direct', which then routes it to a queue named 'compute_queue' using the routing key 'calculate'. The queue has two consumers, 'consumer1' and 'consumer2'. The 'Properties' panel on the right shows the 'Edit Producer' section with a 'producer' field and 'Delete'/'Edit' buttons. The 'New Message' section has a text input with the value '["task": "sumar 2 + 3"]', a 'calculate' button, and a '0' value in a text input, with 'Stop' and 'Send' buttons below. The 'Message Log' at the bottom shows two messages received by 'consumer2' and 'consumer1' with the task 'sumar 2 + 3'.

tryrabbitmq.com

RabbitMQ Simulator About

and drag from a source node to connect it to a destination node.

Advanced Mode

Properties

Edit Producer

producer

Delete Edit

New Message

{ "task": "sumar 2 + 3" }

compute

seconds

Stop Send

Message Log

Consumer: consumer2 got msg: { "task": "sumar 2 + 3" }

Consumer: consumer1 got msg: { "task": "sumar 2 + 3" }

Los mensajes llegan a compute_queue y Luego se entregan a los consumidores **en orden alternado (round-robin)**.

RabbitMQ Simulator About

and drag from a source node to connect it to a destination node.

Advanced Mode

Properties

Edit Producer

producer

Delete Edit

New Message

{ "task": "multiplicar 4*5" }

compute

seconds

Stop Send

Message Log

Consumer: consumer1 got msg: { "task": "sumar 2 + 3" }

Consumer: consumer2 got msg: { "task": "sumar 2 + 3" }

Si el routing key del productor no es el mismo al del **Binding Key**, los mensajes no se envían ni procesan. Ver parte de los message Log que solo muestra los de suma.

