

# Sistemas Distribuidos

## Tarea 2

“Arquitecturas distribuidas usando microservicios”

Diego Montecinos O., 201473601-9  
Ian Mora P., 201473609-6

### Diferencias técnicas.

Las principales diferencias técnicas que se pudieron encontrar entre las implementaciones basadas en gRPC y RabbitMQ fueron:

- RabbitMQ y su sistema de colas de mensajes son más adecuadas cuando no es un problema tener un cierto delay en los mensajes. Mientras que gRPC utiliza un modelo de mensajería instantánea mediante stubs lo cual genera un menor delay en el envío de mensajes.
- gRPC utiliza “remote procedure call” haciendo uso de un servicio localizado en otro equipo de la red (en este caso el cliente) mientras que RabbitMQ utiliza un servidor que proporciona el servicio de cola de mensajes el cual puede estar ubicado en el lado del servidor de chat, un servidor dedicado u otro equipo de la red.
- RabbitMQ provee “servicios” de confiabilidad, ya que cuenta con “delivery acknowledgement” en cual lanza excepciones al cliente si falla la conexión hacia algún nodo de RabbitMQ, además éste broker de mensajes posee diversas librerías que permiten recuperarse de las fallas de conexión. gRPC no posee estos “servicios” implementados y quedan como opción para el desarrollador su implementación.
- gRPC genera una conexión directa con el cliente, donde si este no se encuentra disponible el mensaje se pierda, mientras que RabbitMQ permite recuperar mensajes de la cola aún que el cliente no estuviera disponible al momento de ser enviado.
- gRPC requiere que se defina previamente la estructura de los mensajes y respuestas, las cuales son necesarias para los stubs de comunicación, mientras que para RabbitMQ solo hace falta crear las colas y enviar los mensajes a través de ellas, no siendo tan estricto con la estructura o tipo de datos del mensaje.
- Con RabbitMQ el servidor y los clientes pueden enviar los mensajes sin que el otro se haya “levantado” aún, los mensajes se quedan en el nodo de RabbitMQ y cuando el cliente o servidor crea la cola correspondiente estos son entregados. En cambio con gRPC es necesario que las partes involucradas en la comunicación estén funcionando para poder enviar los mensajes sin que se pierdan.

## Recomendación técnica

En base a las diferencias antes mencionadas, como equipo creemos que no se puede hacer una recomendación universal, ya que ambos servicios están orientados a cumplir la misma necesidad de envío de mensajes pero con diferentes modelos. Es por esto que depende de los requisitos del sistema cuál es más apropiado para utilizar. En este caso particular donde se trabajó con un sistema de mensajería tipo “Chat” se recomienda utilizar gRPC si el número de usuarios (clientes) es reducido y si el delay de los mensajes es un requisito crítico para el sistema. En caso de que se tenga un sistema de mensajería que escalará a un gran número de usuarios y donde se requiera robustez en el paso de mensajes sin importar un delay, RabbitMQ es la mejor opción, ya que proporciona un servicio mucho más completo de broker de mensajería con herramientas ya implementadas, lo que simplifica el desarrollo del servicio distribuido de mensajería.