

Informe Tarea 2 Sistemas Distribuidos

Parte 1: Implementación con gRPC

En este caso se ha implementado un servidor el cual permite la conexión mediante gRPC en el puerto 50051, este permite la autenticación, mensajes de control y el envío de mensajes por parte de los clientes.

Ademas, el cliente abre un servidor gRPC en un puerto aleatorio, el cual indica al servidor en el mensaje de autenticación, este servidor gRPC sera utilizado para enviar los mensajes que se reciban por parte de otros usuarios dirigidos al cliente.

Parte 2: Implementación con RabbitMQ

En este caso tambien se ha implementado un servidor que procesa mensajes de control, autenticación y la comunicación entre dos usuarios, pero a diferencia de la parte 1, no es requerido que el cliente abra un servidor para escuchas debido a que al usar RabbitMQ tenemos la capacidad de enviar mensajes bidireccionales de manera asíncrona.

Complicaciones que se tuvieron al implementar la tarea

En la parte 1, como se indica más arriba, fue necesario que el cliente tambien levantara un servidor gRPC para permitir que reciba mensajes de manera asíncrona desde otros clientes.

En la parte 2, la librería utilizada para la conexión a RabbitMQ (pika) no maneja bien conexiones entre diferentes threads, por tanto fue necesario levantar multiples conexiones (3 por parte de cada cliente) y 2 + 1 por cada cliente conectado en el caso del servidor para poder hacerlo funcionar.

Recomendaciones

Entendiéndose que este chat es de juguete y no aplicable en la vida real (la autenticación es de mentira, se podrían enviar mensajes fácilmente sin el servidor python en la parte b y otros), **creemos que para el chat sería más util la implementación B**, por el hecho de que no es necesario que el cliente levante un servidor de escucha (tema que en la actualidad con los operadores realizando CGNAT (<https://openwebinars.net/blog/que-es-cgnat/>) es cada día más difícil. Ademas creemos que la necesidad de crear multiples conexiones a RabbitMQ podría ser subsanado utilizando otra librería que sea capaz de mantener una conexión a travez de multiples threads.

