

## **EP 01 - Sistemas distribuídos**

**Elton Silva - 11201810955**

**Gabriel Zambianco - 11201722750**

**Pedro Serra - 21044215**

Neste primeiro EP da disciplina de sistemas distribuídos nós implementámos um sistema de mensageria, seguindo os pré-requisitos solicitados no enunciado. Há produtores que podem publicar as mensagens em canais específicos, bem como, consumidores que assinam esses canais para receber as mensagens. As mensagens são armazenadas e toda a comunicação é executada conforme solicitado. No servidor (server.py), nós implementamos as funções que realizam a criação de canais/filas de mensagens (CreateChannel), remoção de canais/filas de mensagens (RemoveChannel), listagem dos canais disponíveis (incluindo tipo e o número de mensagens - ListChannels), função que realiza a publicação de mensagens a partir do nome do canal especificado (PublishMessage), realiza a assinatura de canais e recebe as mensagens publicadas nesses canais e salva as mensagens em disco (log). Nós implementamos canais simples e múltiplos, e o servidor lida de maneira transparente com clientes que não estão mais disponíveis. Sobre as assinaturas, implementamos duas funções, uma chamada gRPC no modo stream (função SubscribeChannelStream) e outra que o cliente pode fazer uma chamada independente para receber apenas uma das mensagens enfileiradas no canal (função SubscribeChannelUnary). Para mostrarmos o servidor em funcionamento, criamos o código de dois clientes utilizando as linguagens Python e Scala. Foi de grande aprendizado para nós o desenvolvimento do sistema de mensageria e a utilização do gRPC, que permite definir um serviço, especificando os métodos que podem ser chamados remotamente com seus parâmetros e tipos de retorno. Para a configuração inicial do gRPC, seja em Python ou Scala, seguimos o passo a passo disponível no site “grpc.io” e “scalapb.github.io”, e fomos desenvolvendo as funções que o exercício solicitava. Tivemos bastante descobertas interessantes durante o desenvolvimento do projeto, lidamos com alguns erros de compilação, construímos e reconstruímos funções e métodos, e tiramos aprendizados. Uma das partes mais complicadas foi a implementação do multiple, da função subscribe, e para a implementação do cliente em Scala.

Em suma, nosso código segue o padrão solicitado, implementando um servidor gRPC para gerenciamento de canais de mensagens. Ele utiliza múltiplas threads para processar requisições de forma concorrente e organizada, que suportam a comunicação eficiente entre clientes e o servidor, permitindo operações de publicação e assinatura em canais de mensagens. Para a execução dos testes, basta executar o arquivo proto, seguido do arquivo server.py e por fim client.py, que contém simulações de interações entre clientes e servidor. Durante a construção do código, nós procuramos enriquecer trechos com comentários e detalhes maiores apresentamos no vídeo que se encontra no link: [https://www.youtube.com/watch?v=8s\\_0ZtaN-U0](https://www.youtube.com/watch?v=8s_0ZtaN-U0)