

## **Lab03 – Construindo aplicações distribuídas usando paradigma publish-subscriber**

Disciplina: Programação de sistemas paralelos e distribuídos

Professor: Fernando W Cruz

Turma A

Aluno: Felipe Boccardi Silva Agustini

Matrícula: 180119818

Aluno: Giovanna Borges Bottino

Matrícula: 170011267

### **Introdução:**

O objetivo desse experimento é compreender as características inerentes à construção de aplicações distribuídas, incluindo passagem de parâmetros, envolvendo módulos cliente e servidor conectados por um broker RabbitMQ

### **Descrição da solução:**

O git pode ser encontrado no seguinte link:

<[https://github.com/giovannabbottino/pspd\\_unb/tree/main/Lab03](https://github.com/giovannabbottino/pspd_unb/tree/main/Lab03)>

Nossas contribuições e reuniões podem ser observadas pela contribuição no git em:

<[https://github.com/giovannabbottino/pspd\\_unb/commits/main](https://github.com/giovannabbottino/pspd_unb/commits/main)>.

Para avaliar nosso projeto recomendo uma leitura do nosso consumer.c e do producer.c que tentamos deixar comentado para o entendimento. Além disso, também recomendo seguir o README enviado.

Nossa solução consiste em uma aplicação que utiliza o rabbitMQ para enviar um JSON, o qual é reconstruído para que possa se pegar os dados e encontrar o menor e maior valor enviado.

Isso é, criamos uma struct Mensagem com o array de float e um inteiro tamanho. Com um vetor de 100 posições tivemos muitas dificuldades em transformar essa informação em bytes. Por isso, optamos por um processo de ENCODE e DECODE de JSON.

No producer, essa struct é transformada em JSON e de JSON para string. No consumer, os bytes são transformados em string, de string para JSON e JSON para a struct Mensagem.

Com um vetor de 1.000.000.000 de posições essa solução fica mais lenta. Mas para otimizar podemos dividir em várias mensagens a serem enviadas pelo producer e no consumer receber essas mensagens e comparar entre as recebidas o maior número. No entanto, não conseguimos desenvolver pelo tempo apertado de entrega. Por isso, focamos na entrega apenas do de 100 posições.

### **Conclusão:**

O envio de mensagem utilizado o rabbitMQ é fácil e rápido, contudo, por se tratar de uma tecnologia que não estávamos familiarizados e o pouco tempo disponível para o desenvolvimento da aplicação, tivemos uma certa dificuldade, principalmente pela falta de documentação da aplicação em alguns aspectos.

**Felipe** - A divisão de tarefas permanece a mesma dos trabalhos anteriores, bem equilibrada. Neste trabalho tivemos uma certa dificuldade por se tratar de uma tecnologia nova para nós, e além disso, consideramos o tempo para a realização da atividade muito inadequado, já que tínhamos que lidar com uma ferramenta nova e a documentação dela deixa a desejar em alguns pontos. Contudo, acredito que lidamos bem com as dificuldades. Nota 10;

**Giovanna** - Como é uma dupla que trabalhei em todos os outros laboratórios, conseguimos manter a divisão equilibrada. Mas o trabalho foi bem complicado pelo tempo apertado e pela documentação da biblioteca que usamos. Relevando esse fato, nos esforçamos muito para a entrega. Nota 10 .