

PSPD - Programação para Sistemas Paralelos e Distribuídos  
Turma A 2021/1

**Lab03\_distribuído** – Construindo aplicações distribuídas  
usando o paradigma *publish-subscriber*

**A) Objetivo:** O objetivo desse experimento é que o aluno compreenda as características inerentes à construção de aplicações distribuídas, incluindo passagem de parâmetros, envolvendo módulos cliente e servidor conectados por *um broker RabbitMQ*.

**B) Detalhes do laboratório**

A Figura 01 a seguir representa dois processos que, em conjunto, representam uma aplicação distribuída com diálogo intermediado por um representante ou *broker*, que implementa persistência e viabiliza a programação no modelo *publish-subscriber*. Os alunos devem instalar o software RabbitMQ (<https://www.rabbitmq.com/>) no ambiente local (Linux) e construir uma pequena aplicação distribuída (linguagem C), que descobre o maior e o menor valor em um vetor de 100 posições, considerando os seguintes passos:

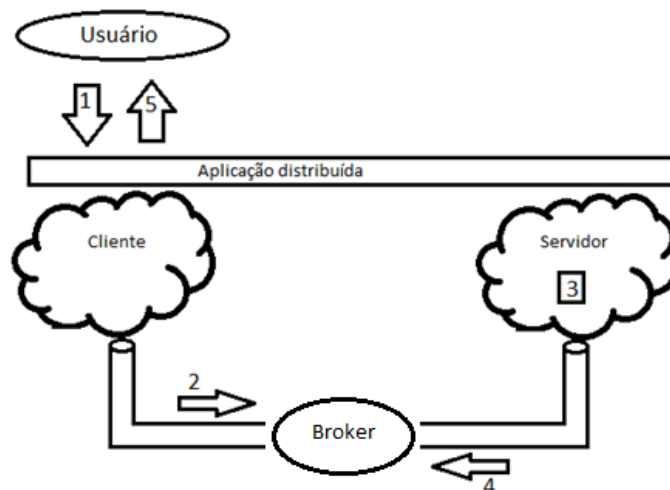
1. O vetor deve ser de números do tipo *float*, e inicializado pelo uso da fórmula a seguir:

$v[i] = (i - \text{tamanho\_do\_vetor}/2) ** 2;$
--

2. Em seguida, a aplicação deve calcular cada posição do vetor, de acordo com a seguinte fórmula:

$v[i] = \text{raiz\_quadrada}(v[i]);$
---------------------------------------

3. O lado servidor deve analisar o vetor, encontrar o maior e o menor valor e repassar esses números de volta para o cliente, de modo que ele possa imprimir o resultado para o usuário.
4. Os lados cliente e servidor dessa aplicação distribuída devem dialogar com apoio do *broker* RabbitMQ, conforme ilustrado na Figura 01.
5. Ou seja, o cliente apresenta para o usuário solicitante, o resultado da operação. Perceber que a solicitação foi realizada de maneira distribuída, ou seja, envolvendo unidades de processamento e espaço de memória de máquinas distintas.



**Figura 01** – Arquitetura de uma aplicação distribuída usando *broker publish-subscriber*

### C) Questões de ordem

- A atividade pode ser feita por grupos de até 2 alunos.
- Caso solicitados, os alunos devem estar preparados (slides e apresentação do código) para demonstração da solução em sala de aula, conforme sorteio definido pelo professor em data oportuna.
- A entrega será feita pelo envio de um arquivo zipado no ambiente Moodle da disciplina disponível em <http://aprender3.unb.br>. O arquivo zipado deve conter: (i) os arquivos .c das aplicações criadas, (ii) instruções de instalação do *broker* e dos processos que compõem a aplicação, e (iii) um relatório cujo conteúdo está descrito mais adiante.
- Os códigos, devidamente comentados e identados, devem ser entregues em conjunto com um relatório sobre a atividade realizada, o qual deve conter os seguintes pontos:
  - a) Título do experimento, dados da disciplina e do(s) aluno(s)
  - b) Introdução – pequena descrição do problema e do protocolo de filas implícito do RabbitMQ
  - c) Descrição da solução para 100 posições – apontar problemas e soluções encontradas até chegar à versão final do problema proposto. Incluir aqui eventuais limitações dos códigos entregues.
  - d) Descrição da solução para um vetor de 1.000.000.000 de posições de números do tipo *float*. Caso a aplicação original não funcione para esse vetor maior, proponha uma reorganização do código, usando o RabbitMQ, de modo a encontrar o maior e o menor valor desse novo vetor. Incluir aqui eventuais limitações dessa solução, instruções de instalação e uso dos processos comunicantes (e do *broker*, se for o caso). Caso seja gerado código diferente para esse vetor alterado, incluí-lo no mesmo arquivo zipado que será postado no Moodle.
  - e) Opinião geral sobre o experimento, apontando dificuldades encontradas e possíveis limitações percebidas.