

Trabalho Prático de Avaliação (TP1)

Objetivo: Utilização do *middleware* gRPC na implementação de um sistema com múltiplos servidores ligados em anel para suportar distribuição de carga nos pedidos dos múltiplos clientes

Notas prévias:

- [A aula de 09 de Novembro de 2021, na turma diurna e a aula de 15 de Novembro de 2021, na turma noturna](#) serão totalmente alocadas para realizarem o trabalho. No entanto, é pressuposto e faz parte dos ECTS da Unidade Curricular que têm de dedicar horas de trabalho fora das aulas para a realização do trabalho. Para eventual apoio e esclarecimento de dúvidas fora das aulas devem agendar com os professores o pedido de ajuda que poderá ser feito presencial ou remoto via Zoom. (nos *links* disponíveis no Moodle de cada turma);
- De acordo com as regras de avaliação definidas no slide 5 do conjunto *CD-01 Apresentação.pdf*, este trabalho tem um peso de 20% na avaliação final;
- A entrega será realizada em Moodle com um ficheiro Zip, incluindo os projetos desenvolvidos (src e pom.xml sem incluir os artefactos), bem como outros ficheiros que considerem pertinentes para a avaliação do trabalho. Por exemplo, podem juntar um ficheiro PDF ou txt, tipo readme.txt que explica os pressupostos que usaram bem como a forma de configurar e executar o sistema;
- [Nas aulas de 06 e 07 de dezembro \(turma diurna\) e 13 de dezembro \(turma noturna\)](#), cada grupo terá de apresentar e demonstrar, durante 10 minutos e para toda a turma, a funcionalidade do trabalho realizado;
- A entrega no Moodle é: [turma diurna até dia 5 de dezembro \(23:59\)](#), [turma noturna até dia 12 de dezembro \(23:59\)](#).

Pretende-se o desenvolvimento de um sistema de distribuição de mensagens (*ChatRing*) baseado num anel de servidores para suportar equilíbrio de carga com um elevado número de clientes conectados aos vários servidores do anel, como se apresenta na Figura 1.

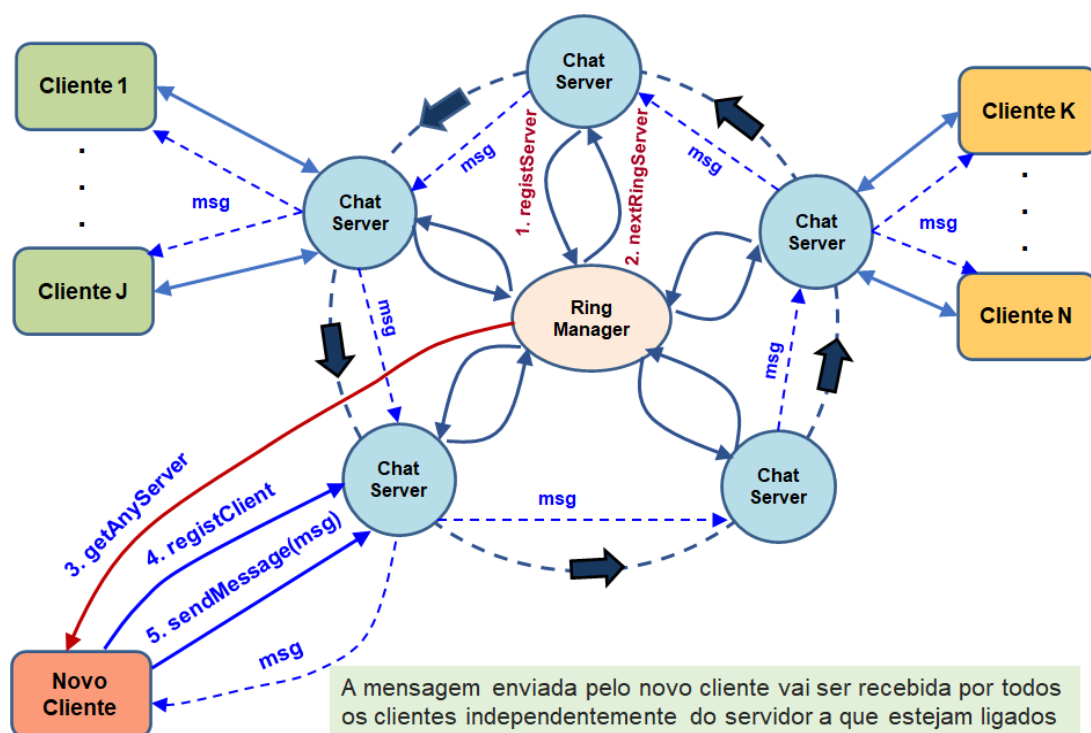


Figura 1 - Diagrama geral do sistema ChatRing

Requisitos funcionais

- Existe um servidor central que faz a gestão do anel (*RingManager*), assumindo que existe um número fixo (N) de servidores *ChatServer*, conhecido inicialmente, aquando do arranque do servidor *RingManager*;
- Um cliente para aceder ao serviço de Chat disponibilizado por cada servidor terá de previamente pedir ao *RingManager* um *EndPoint* (IP, port) de um servidor;
- O servidor *RingManager* só disponibiliza acessos aos servidores de Chat, após a conclusão da configuração do anel que se processa da seguinte forma:
 - Cada um dos N servidores (*ChatServer*) inicia-se com o registo no servidor central *RingManager*, indicando o seu *EndPoint* (IP, port) (ações 1 e 2);
 - Após todos os servidores *ChatServer* se registarem, o *RingManager* devolve a cada servidor o seu sucessor no anel, garantindo que o último que se registou tem como sucessor o primeiro que se registou;
- Após a configuração do anel o *RingManager* aceita pedidos dos múltiplos clientes (ação 3) para obterem o *EndPoint* do servidor a que se irão ligar usando uma distribuição *Round Robin* pelos vários servidores;
- Após obterem um servidor do *RingManager*, os clientes registam-se no servidor *ChatServer* (ação 4);
- A partir desse registo cada cliente poderá enviar mensagens de texto (ação 5) para o seu servidor;
- Quando um servidor *ChatServer* recebe uma mensagem, a mesma é encaminhada para todos os clientes registados, nesse servidor, e também para o próximo servidor no anel que a entrega a todos os seus clientes, e assim sucessivamente. Assim, independentemente do servidor a que um cliente está ligado, as mensagens são sempre recebidas por todos os clientes;
- Para que possa ser detectado a fim do reencaminhamento, cada mensagem, que circula no anel, deve ter a marca do servidor onde foi originalmente recebida.

Requisitos não funcionais

- Por questões de isolamento entre as partes (*Loose Coupling*) devem existir 4 contratos:
 1. Do servidor *ChatServer* para o servidor *RingManager* para registo e obtenção do próximo no anel;
 2. Do Cliente para o servidor *RingManager* para obtenção do *EndPoint* de um servidor *ChatServer*;
 3. Do servidor *ChatServer* para o servidor *ChatServer* para comunicação no anel entre servidores;
 4. Do cliente para o servidor *ChatServer*
- Assim, cada servidor *ChatServer* terá de disponibilizar dois serviços em dois portos diferentes, um para acesso dos clientes e outro para acesso pelo servidor predecessor no anel;
- Da mesma forma, o servidor *RingManager* disponibiliza dois serviços, um para registo de servidores *ChatServer* e outro para os clientes obterem os *Endpoints* dos servidores *ChatServer*.
- Assume-se que o servidor *RingManager* está disponível num *EndPoint* (IP, port) sempre igual e bem conhecido;
- Cada servidor *ChatServer* pode estar em *Endpoints* da mesma VM ou de VM diferentes;
- A construção do protótipo de demonstração deve ter pelo menos 3 instâncias de servidores *ChatServer* com pelo menos um cliente em cada servidor;
- As várias instâncias da aplicação Cliente podem executar-se tanto nas máquinas locais dos elementos do grupo (computadores pessoais) como nas instâncias de VM onde se executam os vários servidores.

Sugestões Gerais

- Qualquer questão ou dúvida sobre os requisitos deve ser discutida com o professor;
- Antes de começar a escrever código desenhe a arquitetura do sistema, os contratos dos serviços bem como os diagramas de interação mais importantes;
- Tenha em atenção o tratamento e propagação de exceções para assim o sistema ser mais fiável e permitir tratar as falhas;
- Quando tiver dúvidas sobre os requisitos, verifique no site *Moodle* se existem "*Frequently Asked Questions*" com esclarecimentos sobre o trabalho.