

TRABALHO EM GRUPO

VALOR: 30 PONTOS

IMPORTANTE, **leia atentamente as instruções a seguir**:

- Trabalho prático em grupo, a ser realizado em trios (três alunos).
 - Exceções devem ser previamente comunicadas e aprovadas pelo professor.
- Para a implementação do trabalho, recomenda-se a utilização do middleware JGroups;
 - Sugere-se a utilização da IDE Netbeans, no SO Linux Ubuntu.
- Deverá ser submetido no portal *meuIFMG* (<https://meu.ifmg.edu.br>) um arquivo compactado (.zip) contendo:
 - a implementação do trabalho: códigos-fonte + programa compilado (.java e .class);
 - o arquivo XML contendo a configuração dos protocolos do JGroups a serem utilizados pelo programa;
 - um breve relatório:
 - apresentando a arquitetura do sistema;
 - explicando e justificando as decisões de projeto do grupo;
 - justificando os protocolos do JGroups utilizados;
 - fornecendo uma [análise de desempenho da pilha de protocolos](#) configurada (latência, mensagens por segundo, etc);
 - apresentando os pontos fortes e fracos da solução desenvolvida;
 - um arquivo executável (ex.: script.sh ou programa.jar) para facilitar a implantação e execução do trabalho.
 - um arquivo texto README.txt contendo informações de (i) como compilar e de (ii) como executar (via terminal CLI) o programa no sistema operacional Linux Ubuntu;

DATAS DAS ENTREGAS:

Etapa 1: funcionalidades básicas

até 15/06 via portal *meuIFMG* (<https://meu.ifmg.edu.br>)

Deverá ser agendada a apresentação parcial do trabalho ao professor. Sugestão de data: 16/06 (sexta-feira).

Etapa 2: funcionalidades básicas e intermediárias

até 30/06 via portal *meuIFMG* (<https://meu.ifmg.edu.br>)

Deverá ser agendada a apresentação do trabalho ao professor. Sugestão de datas: de 03/07 a 06/07.

DESCRIÇÃO DO TRABALHO:

Deverá ser projetado um **sistema distribuído** (SD) que proveja um **serviço de chat**, inspirado em aplicativos populares como WhatsApp, Telegram e afins. A infraestrutura do SD deverá ser provida pelo [middleware JGroups](#), com a comunicação entre os componentes ([multicast](#), [unicast](#), [anycast](#)) realizada através do conector [JChannel](#).

REQUISITOS DO SISTEMA:

O SD deverá possuir as seguintes características do **middleware** (JGroups):

- [Transmissão](#) de mensagens entre os [membros](#) do [cluster](#) JGroups;
- [Multicast confiável](#) e [ordenação total](#) das mensagens para os membros do cluster;
- Serviço de [composição do cluster](#), com rápida [descoberta](#) de novos membros e [detecção de falhas](#);
- Podem ser utilizadas [flags de mensagem](#) e [opções de requisição](#), de acordo com o serviço necessário;
- A [pilha de protocolos](#) XML projetada deverá apresentar um [bom desempenho](#).

O SD deverá prover as seguintes **funcionalidades BÁSICAS** para o chat:

- Prover alguma interface com o usuário (CLI ou GUI) para acesso às funcionalidades do chat;
- Envio de mensagens de um membro para [todos](#) os membros do cluster JGroups;
- Definição do "apelido" (*nickname*) a ser adotado pelo usuário do chat e por ele definido;
- Envio de mensagem [somente para](#) determinado membro do Chat;
- Definição do "nome de batismo" de um grupo de chat, escolhido pelo usuário;
- Envio de mensagem [somente para os membros](#) de um determinado grupo de chat;
- Criação de novo grupo de chat, identificado pelo nome do grupo e [coordenado por um membro](#);
- [Inclusão de novo membro](#) em um determinado grupo de chat, pelo [membro coordenador](#).

O SD deverá prover as seguintes **funcionalidades INTERMEDIÁRIAS** para o chat:

- Armazenamento persistente dos históricos de comunicação do chat (usuários, grupos) e da composição dos grupos de chat (membros);
- O chat deverá ser capaz de continuar funcionando caso haja [particionamento](#) na rede, devendo [consolidar os estados](#) das partições quando se [reunirem](#).
- No [reingresso](#) de um membro, deverá ser [obtido o estado](#) do chat, ou seja, o histórico de comunicações que ele possa [não ter recebido](#) enquanto estava desconectado;
- O serviço de chat deverá prover [mecanismos de segurança](#), como criptografia e autenticação.

OPCIONALMENTE, o SD poderá prover as seguintes **funcionalidades AVANÇADAS** para o chat:

- O serviço de chat pode utilizar, para implementar algumas funcionalidades, canais de comunicação mais leves e baratos ([ForkChannel](#)).
- O serviço de chat pode implementar um [limitador de taxa de dados](#) (quantidade de dados enviada por unidade de tempo), evitando o congestionamento do serviço.
- O serviço de chat pode ser capaz de [comprimir](#) mensagens grandes para agilizar a sua transferência.
- O serviço de chat poder ser capaz de [fragmentar](#) mensagens grandes em mensagens menores, remontando-as no receptor.
- O serviço de chat pode ser capaz de realizar [controle de fluxo](#), ajustando a taxa dos transmissores à taxa do receptor mais lento.
- O serviço de chat pode ser capaz de permitir que usuários que estejam em [LANs diferentes](#) possam se conectar ao cluster, [lidando com](#) peculiaridades da infraestrutura de rede;