

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 S.O. Ubuntu (GNU/Linux).
- 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 e explicando e justificando as decisões de projeto do grupo;
 - justificando a escolha dos protocolos (JGroups) utilizados e fornecendo uma [análise de desempenho da pilha de protocolos](#) configurada (latência, mensagens por segundo, etc);
 - apontando 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 06/06 via portal **meuIFMG** (<https://meu.ifmg.edu.br>)

Deverá ser agendada a apresentação parcial do trabalho ao professor. Sugestão de datas: 07 a 08/06.

Etapa 2: funcionalidades básicas e intermediárias

até 01/07 via portal **meuIFMG** (<https://meu.ifmg.edu.br>)

Deverá ser agendada a apresentação do trabalho ao professor. Sugestão de datas: de 27/06 a 29/07.

DESCRIÇÃO DO TRABALHO (v1.1):

Deverá ser projetado um **sistema distribuído** (SD) que proveja um *Serviço de Leilão Eletrônico*. A infraestrutura do SD deverá ser provida pelo [middleware JGroups](#), com a comunicação ([multicast](#), [anycast](#), [unicast](#)) entre os componentes — membros do(s) *cluster*(s) — realizada através do conector [JChannel](#).

FUNCIONAMENTO DO SISTEMA:

O **serviço** de leilão deverá permitir o anúncio de novos itens para serem leiloados, criação de sala(s) de leilão com os usuários interessados em um mesmo item, bem como receber os lances feitos pelos **usuários**. Um determinado **item** (cadastrado no sistema por um usuário) pode ser leilado em apenas uma **sala de leilão** por vez. Em caso de empate em uma rodada de lances, o leilão do item deverá continuar até que haja um maior lance. Após passado algum tempo sem novos lances, o lance de maior valor para um determinado item será considerado como **vencedor**, passando o item a ser de propriedade do usuário que enviou o lance vencedor. Observe que NÃO poderá haver dois vencedores de um mesmo leilão, bem como um mesmo item NÃO poderá ser leilado "simultaneamente" em dois ou mais leilões.

REQUISITOS DO SISTEMA:

Quando um novo componente for adicionado ao sistema (ingresso de novo membro no *cluster*) ele deverá receber a **transferência do estado** atualizado do sistema. O estado do sistema de leilão consiste em: itens anunciados, usuários autenticados, salas de leilão e seus respectivos históricos de lances, demais informações que se fizerem necessárias, conforme a função daquele componente no sistema. O sistema deverá **armazenar** periodicamente seu estado de maneira **persistente** (sugestão: serialização de objetos para arquivo em disco). Quando todos os componentes forem desligados, ao reiniciar o sistema deverá ser **carregado** para a memória principal o último estado (o mais atual possível) armazenado na memória persistente. O sistema deverá ter **distribuição** vertical (sugestão: divisão em camadas MVC) e horizontal (replicação de componentes em cada camada), de maneira que deverá ser **tolerante a falhas**, permanecendo operacional mesmo que alguns de seus componentes (membros do *cluster*) sofram falha por colapso.

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

- Serviço de [composição do cluster](#), com rápida [descoberta](#) de novos membros e [detecção de falhas](#) nos membros de um cluster JGroups;
- [Transmissão](#) de mensagens entre os [membros](#) do [cluster](#) JGroups, através de [multicast confiável](#) e com [ordenação](#) das mensagens.
- Podem ser utilizadas [flags de mensagem](#) (sincronização), [opções de requisição](#) (votação), [bloqueios / travas](#) ou [contadores atômicos](#), de acordo com o serviço necessário;
- A [pilha de protocolos](#) XML projetada deverá apresentar um [bom desempenho](#), devendo-se portanto ler a [documentação](#) do *middleware* JGroups para [conhecer](#) melhor os [protocolos](#) por ele providos e evitar incluir protocolos [desnecessários](#) ao funcionamento do serviço.

ETAPA 1) o SD deverá prover as seguintes **funcionalidades BÁSICAS**:

- Prover alguma interface com o usuário (CLI ou GUI) para acesso rápido às funcionalidades do serviço de leilões;
- Definição de um identificador único por usuário do sistema, de maneira que um mesmo usuário seja corretamente identificado ao sair e retornar;
- Cadastro de novo item a ser leilado, sem duplicidade, em acordo dos membros ([todos ou maioria](#));
- Criação de nova sala de leilão, identificada pelo [item a ser leilado](#) e [coordenada](#) por algum membro, denominado "leiloeiro";
- [Inclusão de novo membro](#) em determinada sala de leilão, realizada [através do leiloeiro](#) da sala.
- Envio de lances de um usuário [somente para](#) os outros usuários que estejam naquela sala de leilão;
- O resultado do leilão de um item deve ser [visível para todos](#) os demais membros do *cluster* JGroups;

ETAPA 2) o SD deverá prover as seguintes **funcionalidades INTERMEDIÁRIAS**:

- Armazenamento persistente do estado do sistema (cadastros, alterações);
- No reingresso de um membro, deverá ser obtido o estado do sistema, ou seja, atualizações que ele possa não ter recebido enquanto estava desconectado;
- O serviço de leilões deverá prover mecanismos de segurança, como criptografia das mensagens trocadas, autenticação dos usuários e autenticidade de seus lances.
- O serviço deverá ser capaz de continuar funcionando caso haja particionamento na rede, devendo consolidar os estados das partições quando se reunirem.

EXTRA) OPCIONALMENTE, o SD poderá prover as seguintes funcionalidades **AVANÇADAS**:

- O serviço de leilão pode utilizar (sub)canais de comunicação mais leves e baratos (ForkChannel), para implementar algumas funcionalidades (ex.: sala de leilão um de item);
- O serviço de leilão pode ser capaz de permitir que usuários que estejam em LANs diferentes possam se conectar ao mesmo *cluster*, lidando com peculiaridades da infraestrutura de rede;
- O serviço de leilão pode ser capaz de comprimir mensagens grandes para agilizar a sua transferência;
- O serviço de leilão pode implementar um limitador de taxa de dados (quantidade de dados enviada por unidade de tempo), evitando o congestionamento do serviço ou ataques de negação de serviço (DoS);
- O serviço de leilão pode ser capaz de realizar controle de fluxo, ajustando a taxa dos transmissores à taxa do receptor mais lento;
- O serviço de leilão poder ser capaz de fragmentar mensagens de aplicação muito grandes em mensagens menores, remontando-as no receptor, agilizando assim sua transferência e evitando fragmentação a nível da camada de rede.