

Trabalho Prático:

Arquitetura de Processos Pares: Eventos e Notificações, Java RMI, JMS.

1 Objetivo

Desenvolver um sistema *peer-to-peer* que gerencie um leilão de produtos.

2 Requisitos da aplicação

1. Implementar os processos da aplicação em Java;
 2. Utilizar a *middleware* Java RMI (*Remote Invocation Method*) para prover a comunicação entre os processos pares;
 3. O leiloeiro de cada processo cadastra/publica os produtos que deseja leiloar em um Tópico de mensagens do JMS (*Java Message Service*). As informações publicadas devem conter:
 - identificação do processo par;
 - identificação do produto a ser leiloado:
 - código;
 - nome;
 - preço inicial/atual;
 - tempo do término do leilão).
 4. Processos interessados no sistema de leilão devem se registrar no Tópico de Mensagens do JMS para receber informações sobre os novos eventos (leilões abertos);
 5. Uma vez interessado em um leilão específico visualizado pelo JMS, o processo par deve entrar em contato diretamente com o processo responsável por esse leilão para dar início ao(s) lance(s). Essa comunicação deverá ocorrer via Java RMI;
-

6. As informações sobre o andamento do leilão de cada produto devem ser apresentadas aos processos em uma interface gráfica via Java RMI. Todo lance deve ser notificado de forma assíncrona aos participantes interessados no produto em questão. A informação sobre o preço do produto (lance maior oferecido até o momento) deve ser atualizada no Tópico de Mensagens do JMS.
7. Os processos participantes só podem efetuar lances crescentes (lance maior que o atual) para um determinado produto;
8. Um participante pode efetuar quantos lances desejar;
9. O leilão de um determinado produto deve ser finalizado quando o tempo definido para esse leilão expirar ou quando o vendedor do produto desejar.
10. Ao finalizar o leilão, todos os participantes devem receber as seguintes informações (via Java RMI):
 - nome completo do vencedor do produto;
 - nome do produto;
 - valor negociado.

3 Equipe

A equipe pode ser composta por, **no máximo**, 3 pessoas. É permitido que seja feito individualmente.

4 Entrega

A entrega do trabalho será por relatório de atividade, em PDF, gerado por **TEX**. No *Moodle* está disponível um diretório com exemplos incluindo o `modeloRelatorioArtigo.tex`, que poderá (apenas sugestão) ser usado. O relatório deve conter as seguintes seções:

Introdução: descrição do objetivo do trabalho;

Metodologia: dados sobre os experimentos e implementação, como a configuração das máquinas utilizadas (se for o caso), a localização das mesmas na rede, linguagem de programação, bibliotecas ou qualquer outra informação técnica relevante;

Resultados: apresente o *print* as telas da aplicação, exemplo de como funcionam, gráficos (se for o caso) etc;

Análise: para cada experimento, discuta os resultados observados. Os resultados foram de acordo com o esperado?

Conclusão: como todo trabalho técnico, algumas palavras finais sobre o resultado do trabalho, tanto das observações quanto do seu aprendizado sobre o assunto;

Código: apresente todo o código fonte comentado da aplicação. Use para isso o recurso do \LaTeX exemplificado no arquivo `algoritmo.tex`.

5 Avaliação

A avaliação se dá segundo o relatório entregue e apresentação. Cabe ao professor decidir quais alunos irão apresentar individualmente (por amostragem), sendo que a **nota será refletida em toda a equipe**.

A entrega do relatório é feita por qualquer membro da equipe (apenas um), no Moodle, até as 23:59 horas do dia da entrega. A partir de então, será aplicada uma multa de 10% do valor do trabalho por dia de atraso.

Qualquer dúvida a respeito do trabalho deve ser postada no Moodle, onde será prontamente respondida. No entanto, pede-se 24 horas de prazo para a resposta.
