Sistemas Distribuídos

Prof. Carlos Eduardo de Barros Paes.

Projeto Chat de Mensagens usando RMI Java

A aplicação de Chat trata-se de um programa cliente-servidor que permite que diversos usuários realizem um bate-papo em tempo real. A comunicação (troca de mensagens) deve ser centralizada no servidor, ou seja, todas as mensagens enviadas pelos usuários do bate papo devem ser enviadas primeiro para o servidor, o qual deve repassá-las para os usuários envolvidos no chat. Basicamente, toda a abstração e controle dessa comunicação devem ser de responsabilidade do servidor.

O projeto consiste no desenvolvimento dessa aplicação de Chat previamente apresentada e com a arquitetura ilustrada na Figura 01.

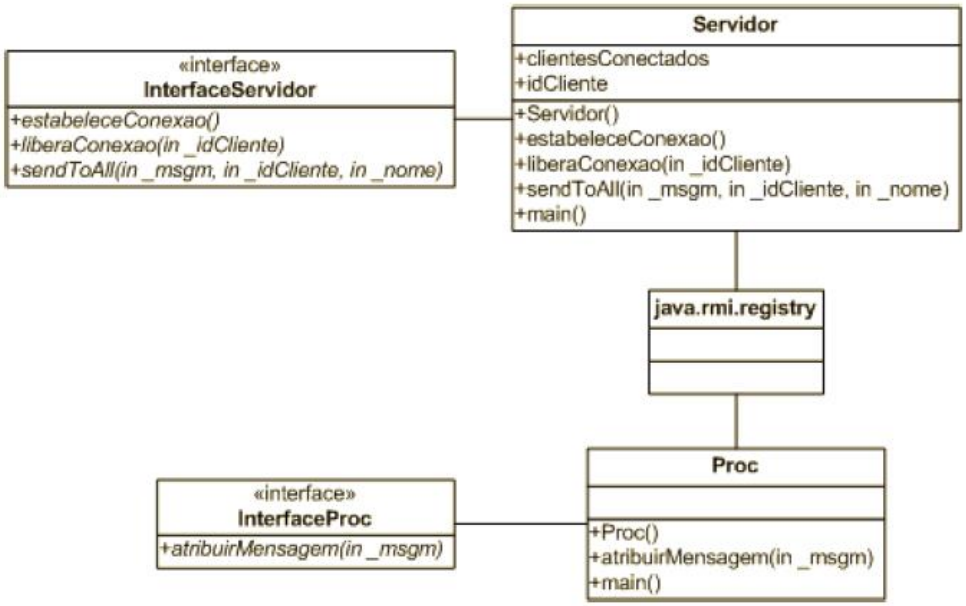

Figura 01. Arquitetura do Chat de Mensagens

Na arquitetura proposta na Figura 01 foram definidas duas interfaces, a interface do servidor (*InterfaceServidor*) e a interface do cliente (*InterfaceProc*), além de duas classes que implementam essas interfaces, a classe Servidor e a classe Proc. No lado do servidor existem três métodos que implementam a interface definida e que possuem as seguintes funcionalidades:

- ***estabeleceConexao***(): método usado para que o servidor identifique quais clientes estão conectados no chat, além de criar uma referência individual por processo dos serviços que estes disponibilizam . Com essa referência, torna-se possível ao servidor chamar os métodos dos processos clientes.

- ***liberaConexao***(): método usado para desfazer a conexão estabelecida por um cliente através do método *estabeleceConexao*(). Este método exclui o cliente conectado do conjunto de clientes conectados.
- ***sendToAll***(): método usado para enviar a mensagem digitada por um usuário do chat para todos os demais usuários. Esse método é acessado remotamente pelo cliente (classe *Proc*) o qual envia como parâmetros a mensagem digitada (_msgm), o seu id (_idCliente) conseguido no método *estabeleceConexao*(), e o nome do usuário do chat (_nome). Quando esse método é executado no servidor, ele acessa o método remoto *atribuirMensagem*() disponibilizado pelos clientes através da classe *Proc*, o que permite o envio da mensagem aos demais usuários.

Já no lado do cliente, foi definido apenas o método atribuirMensagem(), o qual é responsável por receber a mensagem enviada pelos usuários do chat e escrever na tela a string correspondente.

O servidor (classe Servidor) é basicamente um processo que deve ficar sempre executando e disponibilizando seus métodos até que seja encerrado. Ele é uma classe que implementa os métodos remotos definidos na classe *InterfaceServidor* e funciona como um servidor de RMI. O processo cliente (classe *Proc*), ao contrário do servidor, é um processo que interage com o usuário do chat, uma vez que ele lê os dados digitados e imprime na tela as mensagens digitadas e recebidas.

A interface com o usuário da aplicação poderá ser no modo texto ou gráfico (janela e componentes de interface do Java). Esta decisão fica a critério de cada grupo.

Questões de Ordem do Projeto.

- O projeto deverá ser apresentado para o professor até o dia 06/10/2016. Além disso, o código fonte do projeto deverá ser enviado previamente ao professor através do moodle.
- Usar as aulas de laboratório de SD para o desenvolvimento de projeto e com a participação de todos.
- As faltas nas aulas de laboratório que serão alocadas para o desenvolvimento do projeto impactarão na nota final do grupo.
- Todos os participantes do grupo deverão estar presentes no dia da apresentação (2 pontos serão descontados por participante ausente)
- Projeto entregue fora do prazo e através de e-mail NÃO SERÁ ACEITO
- Trabalhos idênticos ou com indícios de cópia receberão NOTA ZERO
- A nota será para o grupo e deverá ser distribuída entre os participantes até 2 dias após a divulgação das notas. Um email deverá ser enviado para o professor!
- O projeto DEVERÁ ser desenvolvido em grupo de no mínimo 2 e no máximo 3 alunos