

## Trabalho prático 1

A sua tarefa consiste no desenvolvimento de uma aplicação Java RMI para registrar usuários (jogadores) em um servidor remoto de jogos. Ao registrar um jogador no servidor, recebe-se um identificador único (um número inteiro). Este identificador único deve ser utilizado pelos jogadores como parâmetro em dois outros métodos remotos disponibilizados pelo servidor: *joga* (simula uma atividade/jogada realizada pelo usuário) e *encerra* (finaliza o registro do usuário). São definidas as seguintes interfaces:

```
public interface JogoInterface extends Remote {  
    public int registra() throws RemoteException;  
    public int joga(int id) throws RemoteException;  
    public int encerra(int id) throws RemoteException;  
}
```

```
public interface JogadorInterface extends Remote {  
    public void inicia() throws RemoteException;  
    public void finaliza() throws RemoteException;  
    public void cutuca() throws RemoteException;  
}
```

O servidor deve ser iniciado com um parâmetro, definindo o número de N jogadores. Quando todos os jogadores houverem realizado o registro, o servidor deverá invocar o método *inicia()* de cada jogador.

Cada jogador deve realizar M jogadas (por exemplo 50, invocação do método *joga()*) e após finalizar (invocação do método *encerra()*). Cada jogada realizada pelos jogadores deve ocorrer com um intervalo de 500ms a 1500ms, sendo esse tempo gerado aleatoriamente. A cada jogada, existe uma probabilidade de 1% do servidor finalizar o jogador (invocação do método *encerrado()*, disponibilizado na interface do jogador). O servidor deve cutucar cada jogador a cada 3 segundos (para verificar se o mesmo continua ativo).

Para a elaboração da atividade, sugere-se utilizar como referência o exemplo sobre callbacks apresentado na aula anterior (aula08-add\_rmi\_callback.tar.gz).

O trabalho deve ser realizado em grupos de 2 ou 3 integrantes. Para o desenvolvimento, é sugerido que os alunos utilizem uma rede com topologia definida, e que sejam realizados testes com um número suficiente de máquinas (pelo menos 3 VMs e pelo menos 5 terminais). Juntamente com sua implementação, deve ser entregue um pequeno relatório (entre 3 e 5 páginas), descrevendo os seguintes aspectos: 1) organização do código (por exemplo, descrição das funcionalidades dos módulos/classes); 2) utilização do programa; 3) demonstração da implementação.