



Universidade Federal do Ceará - Campus de Quixadá  
Disciplina: Sistemas Distribuídos  
Código: QXD0043  
Professor: Rafael Braga

## Trabalho 2 – *Remote Method Invocation (RMI)*

**(1)** Crie uma aplicação, cliente/servidor usando o serviço **RMI** (Invocação Remota de Método). **Não crie sockets nessa questão!**

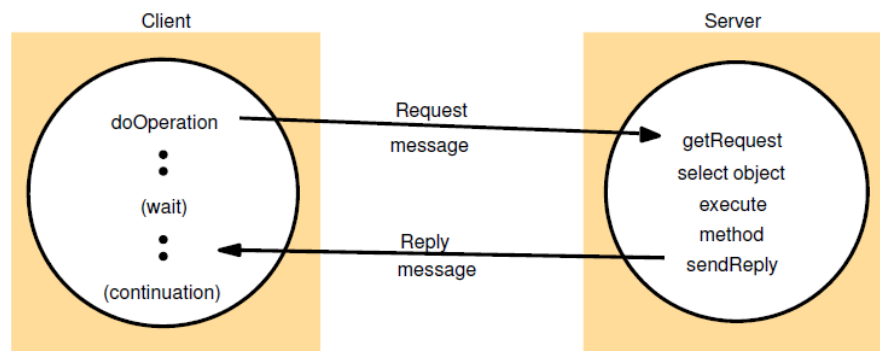
### Orientações específicas da questão 1:

- A aplicação deve ter pelo menos 4 classes do tipo pojos.
- A aplicação deve realizar a passagem por referência e por valor de “objetos”;
- A aplicação deve ter no mínimo uma composição tipo “tem-um” e uma do tipo “é-um”;
- A aplicação deve ter no mínimo 4 métodos.
- Chame todos os métodos no cliente, o servidor implementará esses métodos.

**(2)** Sobre a representação externa de dados e serialização crie uma aplicação cliente/servidor com **Socket** com no mínimo dois métodos.

### Orientações específicas da questão 2:

- O cliente deve empacotar a mensagem de **requisição** antes de enviar para o servidor. E desempacotar a mensagem de **reply** enviada pelo servidor.
- O servidor deve desempacotar a mensagem de **requisição** do cliente, e empacotar o **reply** e enviar para o cliente.
- O serviço remoto deve ser organizado num **protocolo de requisição-resposta** como descrito na **seção 4.4** do livro texto e ilustrado na figura abaixo.
- A comunicação entre cliente e servidor deve ser implementada via **sockets** (TCP ou UDP) que trocam fluxos de bytes (mensagens empacotadas).



O uso dos **métodos sugeridos pelo autor**, que implementam o protocolo requisição-resposta, **deve ser seguido**, mas alterações em suas assinaturas são permitidas:

- ***public byte[] doOperation (RemoteObjectRef o, int methodId, byte[] arguments)***: envia uma mensagem de requisição para o objeto remoto e retorna a resposta. Os argumentos especificam o objeto remoto, o método a ser chamada e os argumentos para aquele método.
- ***public byte[] getRequest ()***: obtém uma requisição de um cliente através de uma porta servidora.
- ***public void sendReply (byte[] reply, InetAddress clientHost, int clientPort)***: envia a mensagem de resposta para o cliente, endereçando-a a seu endereço IP e porta.

As mensagens de requisição/resposta devem ser empacotadas com mostra a figura abaixo. A função de cada elemento da mensagem é descrita também na seção 4.4 do livro texto.

messageType	<i>int</i> (0=Request, 1= Reply)
requestId	<i>int</i>
objectReference	<i>RemoteObjectRef</i>
methodId	<i>Int</i>
arguments	<i>array of bytes</i>

Os elementos *objectReference* e *methodID* podem ser **Strings** que representam, respectivamente, o nome do objeto que fornece o serviço e o nome do método a ser invocado.

O uso de **reflexão** no lado servidor é encorajado para torná-lo mais flexível à adição/remoção de serviços (métodos).

### Orientações gerais:

Cada estudante deve entregar, além do código fonte, um relatório descrevendo o serviço remoto implementado. A entrega deve ser feita via *upload* no MOODLE. Não há necessidade de prover Interface Gráfica com o Usuário (GUI). Modo texto é suficiente.