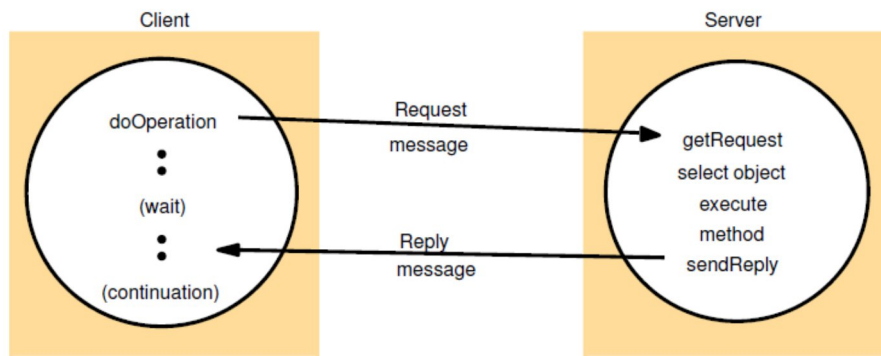




Universidade Federal do Ceará - Campus de Quixadá
Cursos: SI, ES, RC, CC e EC
Disciplina: Sistemas Distribuídos
Código: QXD0043
Professor: Rafael Braga

Trabalho 2 – *Remote Method Invocation (RMI)*

1) Reimplemente a 1ª questão do trabalho 1 através da comunicação cliente-servidor e organizado num **protocolo de requisição-resposta** como descrito na **seção 5.2** do livro texto e ilustrado na figura abaixo. A comunicação entre cliente e servidor deve ser implementada via **Invocação Remota de Método ou *Remote Method Invocation (RMI)***. **Não crie sockets nesse trabalho!**



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 como mostra a figura abaixo. A função de cada elemento da mensagem é descrita também na **seção 5.2** do livro texto.

| | |
|-----------------|---------------------------|
| messageType | int (0=Request, 1= Reply) |
| requestId | int |
| objectReference | RemoteObjectRef |
| methodId | Int |
| arguments | array of bytes |

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.

Requisitos adicionais da aplicação

- A aplicação deve ter **no mínimo 4** classes do tipo **pojos**;
- A aplicação deve ter **no mínimo 2** composição tipo agregação (“tem-um”);
- A aplicação deve ter **no mínimo 2** composição tipo extensão (“é-um”);
- A aplicação deve ter **no mínimo 4** métodos para invocação remota.
- A aplicação deve utilizar a **passagem por referência** para execução **objetos remotos**;
- A aplicação deve utilizar a **passagem valor** para execução **objetos locais** no servidor;

A **passagem de parâmetro por valor** deve utilizar a **representação externa de dados**. Para a representação externa de dados nas chamadas remotas (métodos, argumentos e resultados), sugere-se que seja implementada através de [protocol buffers](#). Contudo, versões em **XML** ou **JSON** também são aceitas.

Orientações gerais:

Cada dupla ou estudante deve entregar, além do URL do repositório do código fonte, um relatório descrevendo o serviço remoto implementado. A entrega deve ser feita via *upload* no MO-ODLE. Não há a exigência de prover Interface Gráfica com o Usuário (GUI). Modo texto é suficiente.