



**Universidade Federal de Ouro Preto**  
**Instituto de Ciências Exatas e Aplicadas**  
**Departamento de Computação e Sistemas**

# **Sistemas Distribuídos:**

## **Relatório – RPC/RMI**

**Ícaro Bicalho Quintão – 14.1.8083**

## **Introdução**

A chamada de procedimento remoto (RPC, Remote Procedure Call), que permite aos programas clientes chamarem procedimentos de forma transparente em programas servidores que estejam sendo executados em processos separados e, geralmente, em computadores diferentes do cliente. Assim como a Invocação Remota de Métodos (Remote Method Invocation – RMI) permite a invocação de um método de um objeto em outra máquina (servidor). Nesse trabalho, utilizou-se desta interface como compilador do código usando o comando “rmic” para gerar o stub.

## **Objetivos**

1. Buscar um exemplo não muito elementar de RPC;
2. Compilar os códigos cliente e servidor;
3. Gerar stubs RPC;
4. Executar o servidor e o cliente;
5. Entender o funcionamento do código

## **Procedimento experimental e Resultados**

O primeiro passo é a implementação dos códigos Calculadora, CalculadoraClient, CalculadoraImple e CalculadoraServer, que nesse trabalho foi pego de um exemplo de RPC. Após isso executa-se o comando “rmic” na classe CalculadoraImple para gerar o “stub” (responsável pela comunicação entre o cliente e servidor).

```
Prompt de Comando
Microsoft Windows [versão 10.0.16299.371]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\Users\phmbb>cd eclipse-workspace
C:\Users\phmbb\workspace>cd ProjetoCalculadora
C:\Users\phmbb\workspace\ProjetoCalculadora>cd bin
C:\Users\phmbb\workspace\ProjetoCalculadora\bin>rmic CalculadoraImple
warning: generation and use of skeletons and static stubs for JMRP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

C:\Users\phmbb\workspace\ProjetoCalculadora\bin>
```

Figura 1: A mensagem de perigo serve para avisar que o stub está depreciado, mas isto é normal.

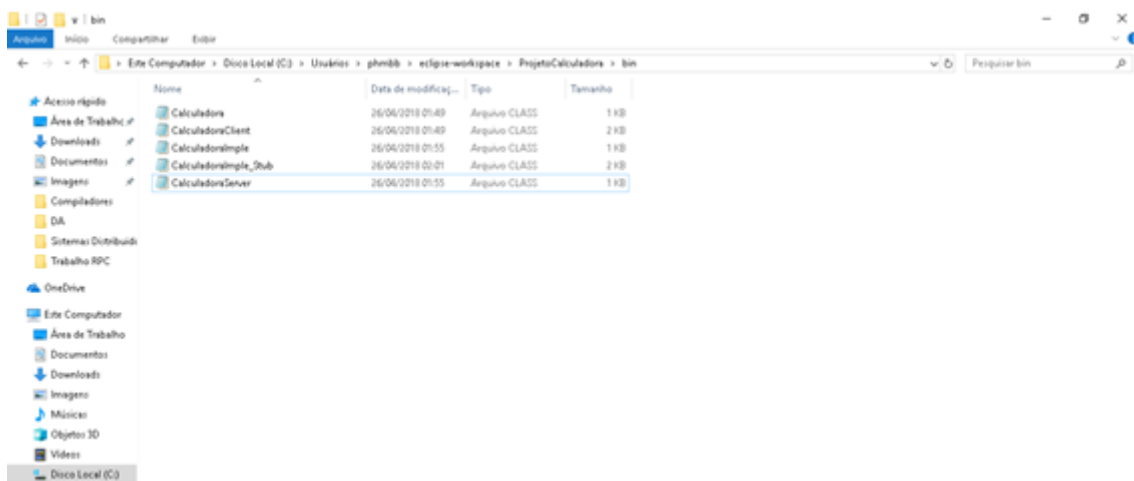


Figura 2: Pasta com as classes e o stub gerado.

O próximo comando executado é o “Java”, na classe CalculadoraServer, que iniciará a execução do servidor e permanecerá executando para que o cliente possa realizar a comunicação.

```
Prompt de Comando - java CalculadoraServer
Microsoft Windows [versão 10.0.16299.171]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\Users\phmbb>cd eclipse-workspace
C:\Users\phmbb\workspace>cd ProjetoCalculadora
C:\Users\phmbb\workspace\ProjetoCalculadora>cd bin
C:\Users\phmbb\workspace\ProjetoCalculadora\bin>rmic CalculadoraImple
warning: generation and use of skeletons and static stubs for JMRP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
C:\Users\phmbb\workspace\ProjetoCalculadora\bin>java CalculadoraServer
```

Figura 3: Este é o servidor executando.

Por fim, é aberto outro prompt de comando, executado o comando “Java” na classe CalculadoraClient. Assim, realiza a comunicação cliente/servidor para a soma dos valores nas variáveis do cliente e a exibição o resultado na tela.

```
Prompt de Comando
Microsoft Windows [versão 10.0.16299.171]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\Users\phmbb>cd eclipse-workspace
C:\Users\phmbb\workspace>cd ProjetoCalculadora
C:\Users\phmbb\workspace\ProjetoCalculadora>cd bin
C:\Users\phmbb\workspace\ProjetoCalculadora\bin>java CalculadoraClient
Adição: 25
C:\Users\phmbb\workspace\ProjetoCalculadora\bin>
```

Figura 4: Cliente executando.

## Conclusão

O RMI consiste basicamente em dois programas (cliente-servidor). O servidor instancia objetos remotos, o referencia com um nome e faz um "vínculo" dele numa porta, onde este objeto espera por clientes que invoquem seus métodos. Já o cliente referencia remotamente um ou mais métodos de um objeto remoto. A interface RMI fornece os mecanismos para que a comunicação entre cliente e servidor seja possível. Esse tipo de aplicação geralmente é denominado como Aplicação de Objeto Distribuído. Em virtude dos fatos mencionados, vimos como podemos usar diferentes métodos em diferentes máquinas virtuais java e combiná-las para usar como um único aplicativo. A interface RMI é útil em um aplicativo do servidor onde se quer comunicar com diferentes máquinas virtuais java, utilizando stub para receber objetos e serializá-los, assim reforçando o verdadeiro polimorfismo orientado a objetos.

## Referências Bibliográficas

Autor João Castelli, *Exemplo Aplicação Java RMI*, disponível em <<https://www.youtube.com/watch?v=tLtTWY6JqNo&t=239s>>. Acesso: 25/09/2018