



INE5418-05208 (20211) - Computação Distribuída

[Painel](#) ▶ [Cursos](#) ▶ [INE5418-05208 \(20211\)](#) ▶ [Unidade 2: Plataformas e Tecnologias para Computação Distribuída](#) ▶ [A.3 Prática - RMI](#)

NAVEGAÇÃO DO QUESTIONÁRIO

1
✓[Terminar revisão](#)

Iniciado em	Thursday, 29 Jul 2021, 10:46
Estado	Finalizada
Concluída em	Saturday, 31 Jul 2021, 12:30
Tempo empregado	2 dias 1 hora
Notas	1,00/1,00
Avaliar	10,00 de um máximo de 10,00(100%)

Questão 1

Completo

Atingiu 1,00 de
1,00🚩 Marcar
questão

Implemente um contador global atômico compartilhado entre múltiplos clientes. O contador deve ser implementado por um objeto remoto e ele descreve um contador monotônico crescente, ou seja, cada leitura usando o método `getCount()` deve retornar o próximo valor inteiro. Garanta que duas invocações ao método nunca retornem o mesmo valor. O primeiro valor lido pelo contador é 1.

Enviei um arquivo zip contendo os arquivos ".java" (interface remota, o código que implementa a interface remota, o código que implementa o servidor responsável por registrar o objeto e por atender requisições dos clientes e o código que implementa o cliente). Inclua também um arquivo texto com os passos necessários para a compilação e execução do seu programa.

Dicas para teste e execução do seu código:

Ao iniciar tanto o cliente quanto o servidor, é necessário informar uma política de acesso de segurança. Isto é feito usando o SecurityManager do Java.

1) Você pode criar um arquivo com a configuração da política de segurança e referenciá-lo na execução dos seus programas RMI. A seguir é apresentado o conteúdo do arquivo "server.policy" e, em seguida, um exemplo de inicialização de um programa cliente, chamado HelloClient e um servidor, chamado Server:

```
grant {  
    permission java.security.AllPermission;  
};
```

Inicialização do cliente:

```
java -Djava.security.policy=server.policy HelloClient
```

Inicialização do servidor:

```
java -Djava.security.policy=server.policy Server
```

2) Lembre de iniciar o serviço `rmiregistry` antes de executar o servidor.

3) Ao criar os seus arquivos .java, lembre que em o nome da classe/interface deve ser exatamente igual ao nome do arquivo.

Schaly_RMI_Counter.zip

Comentário:

Usou o bloco "synchronized" para garantir a atomicidade. Apenas para conhecimento: você também poderia ter usado `java.util.concurrent.atomic.AtomicInteger`, um método `synchronized`, o `java.util.concurrent.Semaphore` ou ainda o `java.util.concurrent.locks.ReentrantLock`.

[Terminar revisão](#)[◀ 15 - Espaço de Tuplas](#)[Seguir para...](#)[A.4 Serviços Web \(opcional\) ▶](#)

Você acessou como [Matheus Henrique Schaly \(18200436\) \(Sair\)](#)
[INE5418-05208 \(20211\)](#)
[Obter o aplicativo para dispositivos móveis](#)