

Computação Distribuída

Síntese de atividades de laboratório

Laboratório nº: 1

Data: sábado, 7 de novembro de 2020

Turma: MI1N

Grupo: 24

Integrantes:

Número	Nome
41548	Guilherme Arede
44021	Nuno Gomes

Atividades propostas:

1 – Implementação de cliente para conectar a servidor RMI

2 – Implementação de cliente e servidor RMI

Atividade 1

1. Objetivo da atividade:

Esta atividade prática foi feita de forma a executar uma implementação de um cliente capaz de comunicar com um servidor desacoplado. Para isto foi utilizado um protocolo RMI, sendo que nesta atividade apenas era necessário implementar o contrato no cliente.

2. Descrição da solução:

A nossa solução começou por analisar a implementação do servidor fornecida, bem como do contrato. Inicia-se com o cliente a efetuar a ligação através do protocolo RMI ao servidor, que corre no endereço de IP que foi fornecido. Após a ligação, utiliza os métodos apresentados para tentar encontrar as três pérolas necessárias. Para utilizar os métodos fornecidos, foi criada uma pequena interface de utilização para pedir coordenadas ao utilizador, de modo a enviar as mesmas ao servidor e ver se correspondem à localização de uma pérola. Em caso de sucesso, um contador existente do lado do cliente é incrementado, dando a indicação de uma tentativa com sucesso. A interface corre até ser encontrado o número máximo de pérolas pretendido. Se as coordenadas não corresponderem a uma localização válida, essa informação é apresentada ao utilizador e este volta a inserir novas posições para prosseguir com o jogo. Quando todas as pérolas forem encontradas, o utilizador é informado de que o jogo foi concluído.

3. Resumo dos problemas encontrados e as soluções aplicadas:

Ao início pensámos em juntar o código todo numa classe, de forma a ficar mais conciso. Após fazer dois métodos, compreendemos que seria melhor se separássemos o cliente do *Main*, e foi o que acabámos por fazer. Ao fazer isto conseguimos ter uma maior modularidade e delegação de tarefas dentro desta atividade, sendo que o *Main* ficou reservado à interface, e o cliente, ficou delegado de fazer a comunicação com servidor.

1. Objetivo da atividade:

Esta atividade prática foi feita de forma a executar uma implementação de um servidor totalmente desacoplado do cliente. Para isto foi utilizado o protocolo RMI, pelo que um dos objetivos secundários deste trabalho era criar e implementar o contrato utilizado para partilha entre as duas partes.

2. Descrição da solução:

O trabalho foi feito de forma distinta em duas partes, que são o servidor e o cliente. Também possui o contrato que é feito pelo servidor e partilhado para o cliente. Este contrato é feito para o servidor e o cliente conseguirem partilhar os métodos, sendo que o servidor especifica e implementa o contrato, enquanto que o cliente apenas utiliza o contrato. O Contrato apenas possuía os métodos de inicializar leilões, de ir buscar todos os leilões e fazer uma licitação. Para além disso também havia uma interface que serve de “token” para enviar notificações do servidor para o cliente, e que foi implementada no cliente.

No cliente é feita a ligação ao servidor, bem como a procura(*lookup*) do servidor no registo RMI, depois disso é apresentada um menu para um potencial utilizador navegar e utilizar os serviços disponíveis que interagem com o contrato, que está implementado no servidor.

A parte do servidor como já foi dita, faz implementação do contrato, e possui toda a operação do sistema, para além disso, também possui uma estrutura com os dados de todos os leilões. Aqui

3. Resumo dos problemas encontrados e as soluções aplicadas:

Um dos problemas iniciais foi garantir a comunicação entre ambos os sistemas, tendo em conta que não sabíamos qual estava a falhar. Para isto foi feito um sistema muito simples no qual o cliente apenas fazia um pedido, e na qual o servidor apenas retornava com a notificação simples, de forma a isolar o erro e perceber o que poderia estar errado. Quando testámos, acabou por não ser necessário pois conseguimos efetuar a ligação à primeira tentativa.

Lidar com os erros também foi um problema, caso um cliente fizesse um pedido que fizesse o servidor não conseguir recuperar, o que fizemos foi sempre retornar uma notificação com o erro de forma ao cliente saber o erro, e o servidor apanhava o erro e imprimia no *standard output* para o seu administrador saber o que se estava a passar.

Comum

4. Indicação se a solução final é apresentável e demonstrável:

Ambas as soluções finais são apresentáveis e demonstráveis, utilizando um IDE para o mesmo. No nosso caso recorremos aos IntelliJ, tendo em conta que não nos interpretámos que fosse necessário gerar um artefacto que funcionasse sozinho em linha de comandos.

5. Conclusão e lições aprendidas:

Com este trabalho aprendemos como utilizar e implementar um cliente, servidor e contrato utilizando Java RMI. Também aprendemos as bases gerais do sistema RMI. Foi compreendido que esta solução fornece uma forma muito simples de aceder a objectos remotos, como se estivessem localmente, ao invés de estar a fazer pedidos para uma API remota. Pelo lado negativo torna-se um serviço muito mais integrado, levando a que alterações no contrato do servidor levem a alterações em todo o lado, no entanto traz mais segurança com isso. Pensamos que seja uma boa solução para sistemas que necessitem de ser mais integrados, e transparentes à concorrência.