

Relatório - Trabalho 2: Remote Method Invocation (RMI)

Universidade Federal do Ceará - Campus de Quixadá\ Disciplina: Sistemas Distribuídos (QXD0043)\

Professor: Rafael Braga\ Aluno: Davi Medeiros e Lucas Nóbrega

1. Objetivo

Reimplementar a primeira questão do Trabalho 1 utilizando comunicação cliente-servidor baseada em invocação remota de métodos (RMI), estruturada segundo o protocolo de requisição-resposta conforme o livro texto (Seção 5.2).

2. Descrição do Serviço Remoto

O serviço remoto implementa uma aplicação de gestão de pedidos, com as seguintes funcionalidades acessíveis via RMI:

- **listarProdutos()**: retorna uma lista de produtos disponíveis.
- **realizarPedido(Pedido pedido)**: recebe um pedido e o registra.
- **consultarCliente(String cpf)**: retorna dados de um cliente.
- **emitirNotaFiscal(Pedido pedido)**: gera uma nota fiscal com base no pedido.

Esses métodos são definidos em uma interface remota e implementados em uma classe que é registrada no `Registry` do RMI.

3. Estrutura de Comunicação

Apesar de usar RMI (que abstrai detalhes de socket), a aplicação simula a estrutura de protocolo de requisição-resposta ao empacotar as mensagens em JSON. Os objetos trocados contêm os seguintes campos:

- **objectReference**: representado pelo nome do objeto remoto.
- **methodID**: representa o nome do método.
- **arguments**: argumentos dos métodos, serializados em JSON.

O uso do JSON garante a representação externa de dados conforme exigido no trabalho.

4. Requisitos Atendidos

- **4 entidades**: `Cliente`, `Endereco`, `Produto`, `Pedido`, `NotaFiscal`.
- **2 composições do tipo agregação (tem-um)**:
 - `Pedido` tem uma lista de `Produto`.
 - `Cliente` tem um `Endereco`.
- **2 composições do tipo extensão (é-um)**:
 - `Pessoa` é superclasse de `Cliente`.
 - `ProdutoComDesconto` estende `Produto` (exemplo adicional, se usado).
- **4 métodos remotos**: listados no item 2.

- **Passagem por referência:** objetos como `Pedido` são passados remotamente.
 - **Passagem por valor com representação externa:** JSON utilizado para argumentos e retorno.
-