



UNIVERSIDADE FEDERAL DO CEARÁ

13, Junho 2025

Sistemas Distribuídos - 01A - 2025.1

Trabalho 2 - Relatório - Sistema de Controle de Passagens Aéreas com RMI

Alisson Rodrigues Fernandes - 510357

Monalisa Silva Bezerra - 535614

1. Introdução

Este relatório descreve a implementação de um sistema de controle de passagens aéreas utilizando Java RMI (Remote Method Invocation), conforme especificado no documento do trabalho 2. O sistema implementa o protocolo de requisição-resposta descrito na seção 5.2 do livro texto, seguindo os padrões de comunicação distribuída.

2. Arquitetura do Sistema

O sistema utiliza uma arquitetura cliente-servidor com os componentes principais abaixo:

- **Servidor RMI:** Hospeda os serviços remotos e gerencia as operações.
- **Cliente RMI:** Consome os serviços remotos para operações de reserva.

- **Objetos Remotos:** Implementam a lógica de negócios do sistema.
- **Modelo de Dados:** Classes POJO representam entidades do domínio.

3. Implementação do Protocolo Requisição-Resposta

Como dito, o sistema utiliza-se do protocolo de requisição-resposta através do RMI, que encapsula automaticamente as operações:

3.1 Mapeamento para RMI

- **doOperation:** Implementado implicitamente pelo stub RMI.
- **getRequest:** Implementado pelo skeleton RMI no servidor.
- **sendReply:** Gerenciado automaticamente pelo runtime RMI.

3.2 Estrutura de Mensagens

| Campo | Tipo | Implementação no Sistema |
|-----------------|------------------|-----------------------------------|
| messageType | int | Gerenciado pelo RMI |
| requestId | int | Gerado automaticamente pelo RMI |
| objectReference | RemoteObject Ref | Representado por nome de registro |
| methodId | int | Mapeado para métodos da interface |
| arguments | byte[] | Serialização Java de objetos |

4. Classes de Entidade

O sistema implementa 4 classes de entidade conforme é exigido no documento do trabalho 2:

4.1 Passagem (classe abstrata)

- Atributos: codigo, proprietario, voo, preco.
- Implementa interface transferível.

4.2 PassagemPrimeiraClasse (subclasse de Passagem)

- Atributos específicos: acessoSalaVIP, milhasBonus.

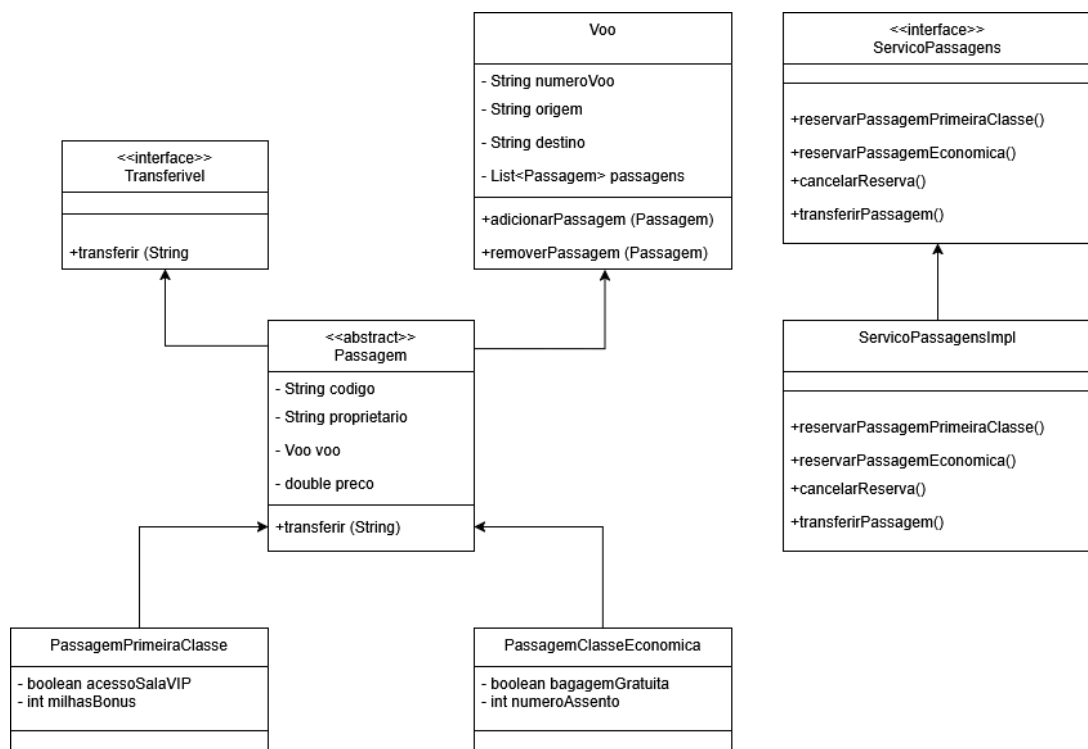
4.3 PassagemClasseEconomica (subclasse de Passagem)

- Atributos específicos: bagagemGratuita, numeroAssento.

4.4 Voo

- Atributos: numeroVoo, origem, destino.
- Agrega coleção de Passagens.

4.5 Diagrama de Classes Simplificado:



5. Relacionamentos

5.1 Agregações (“tem-um”)

1. Voo contém múltiplas Passagem
 - Implementado como lista de passagens.
 - Métodos: adicionarPassagem(), removerPassagem().
2. Passagem contém múltiplas Voo
 - Cada passagem está associada a um voo específico.

5.2 Extensões (“é um”)

3. PassagemPrimeiraClasse é uma Passagem
 - Herda atributos e métodos da superclasse.
 - Adiciona funcionalidades específicas.
4. PassagemClasseEconomica é uma Passagem
 - Especialização da classe base.
 - Implementa características econômicas.

6. Métodos Remotos

A interface `ServicoPassagens` define quatro métodos para invocação remota:

6.1 `reservarPassagemPrimeiraClasse()`

- Parâmetros: Voo, codigo, proprietario, preco.
- Retorna: `PassagemPrimeiraClasse` criada

6.2 `reservarPassagemEconomica()`

- Parâmetros: Voo, codigo, proprietario, preco, assento.
- Retorna: `PassagemClasseEconomica` criada.

6.3 `cancelarReserva()`

- Parâmetros: Voo, Passagem.
- Efeito: Remove passagem do voo.

6.4 `transferirPassagem()`

- Parâmetros: Passagem, novoProprietario.
- Efeito: Atualiza proprietário da passagem

7. Passagem de parâmetros

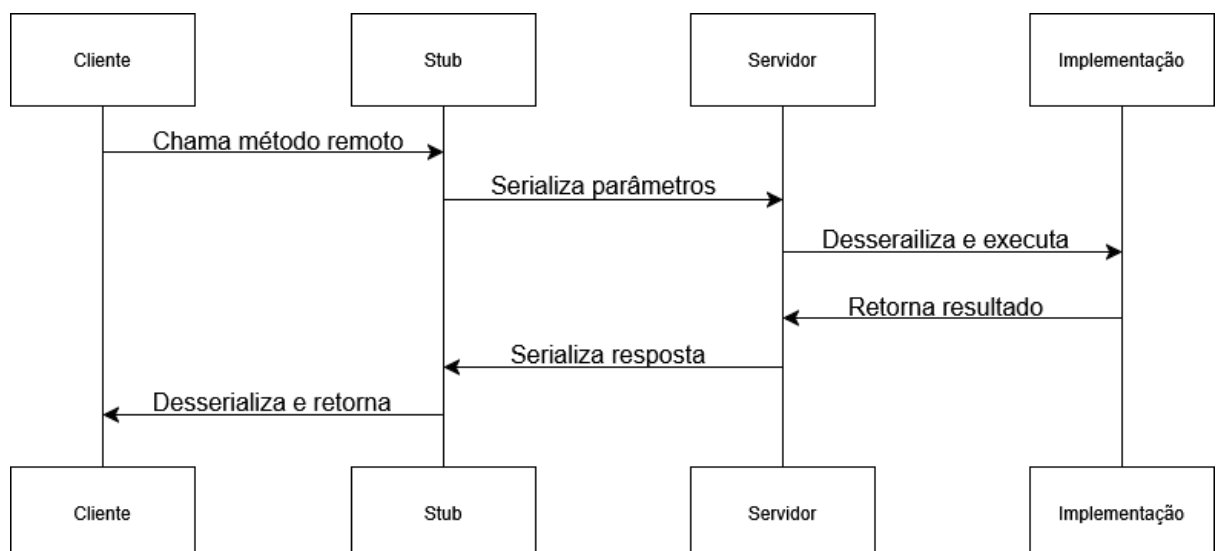
7.1 Passagem por Referência

- Objetos remotos (stubs) são passados por referência.
- O cliente recebe um stub para o serviço remoto.
- Todas as chamadas de um método são encaminhadas ao servidor.

7.2 Passagem por Valor

- Objetos locais (Passagem, Voo) são serializadas.
- Utiliza representação externa de dados via serialização Java.
- Objetos são convertidos em fluxo de bytes para transmissão.
- Desserialização no destino reconstrói os objetos.

8. Fluxo da operação



9. Testes e resultados

O sistema foi testado seguindo o fluxo abaixo:

1. Reserva de passagem de primeira classe.
2. Reserva de passagem econômica.
3. Transferência de passagem.
4. Cancelamento de reserva.

Saída do cliente:

Reservas realizadas:

- PC-001: João Silva
- EC-205: Maria Souza

Passagem transferida: PC-001 -> Carlos Oliveira

Reserva cancelada: EC-205

10. Conclusão

O sistema elaborado atende a todos os requisitos especificados no documento do trabalho:

1. Implementa quatro classes de entidade.
2. Contém quatro métodos para invocação remota.
3. Utiliza passagem por referência para objetos remotos.
4. Utiliza passagem por valor com serialização Java para objetos locais.
5. Segue o protocolo de requisição-resposta através do RMI

A solução apresenta como o RMI abstrai a complexidade da comunicação distribuída, permitindo destacar a lógica de negócios enquanto o *runtime* gerencia automaticamente o protocolo de comunicação.