Relatório - Exercício Programático 0

Brian Andreossi, 11060215

1. Introdução

Este relatório pretende elucidar o sistema desenvolvido como solução para o problema **Exercício Programático ZERO**. Para efeito de entendimento mais claro sobre o sistema, foi utilizado a biblioteca Gson para tratamento de objetos que foram enviados pela rede. Além disso, foi utilizado o maven para gerenciamento de dependências.

2. Requisitos

A seguir, demonstra-se os requisitos do projeto e se o desenvolvedor os alcançou.

O cliente deveria:

- Enviar mensagens ordenadas sequencialmente;
- Enviar mensagens fora de ordem;
- Simular o envio de mensagens perdidas;
- Enviar mensagens duplicadas.

O servidor deveria:

- Receber simultaneamente mensagens dos clientes;
- Armazenar mensagens de um cliente em um buffer;
- Tratar mensagens duplicadas
- Tratar mensagens fora de ordem

3. Mensagem

3.1 Atributos da mensagem

A seguir, uma tabela dos atributos da classe mensagem, seguidas de uma breve explicação.

Atributo	Tipo	Explicação
id	**Long***	Guarda o id da mensagem. (único em relação a conversa)
content	String	O conteúdo referente a mensagem (não há distinção entre mensagem de cliente e servidor no conteúdo)
to	**User***	Usuário que receberá a mensagem (para este contexto, será sempre o servidor)
from	**User***	Usuário que enviou a mensagem

* - Classe Long, definido na biblioteca principal do java.
* - User, classe definida neste projeto.

---

4. Tratamento de mensagens fora de ordem

Quando as mensagens chegam fora de ordem, o cliente espera as mensagens que estão faltantes (até que a ordem se confirme) para as mostrar (para testar, basta usar o comando → **/testOrder**). Todas as mensagens recebidas são guardadas, até que se restabeleça a ordem.

---

5. Tratamento de mensagens duplicadas

Caso uma mensagem chegue duplicada, ela é recebida e validada (foi entendido como duplicata uma mensagem que tem somente o mesmo id que alguma outra). Caso seja uma duplicata, ela é descartada e é exibido em tela uma mensagem condizente. É possível testar usando o comando → **/testDuplicated**

6. Consumo do Buffer e Tratamento de mensagens perdidas

O servidor consome o buffer continuamente (todas mensagens que estão válidas para exibir são colocadas em fila). Mensagens perdidas são consideradas fora de ordem. A ideia é que o servidor implemente uma política de *timeout* e peça a mensagem novamente ao cliente.

Como não era escopo deste projeto esse tipo de tratamento (uma vez que o requisito não incluia mensagens repetidas), foi considerada somente uma mensagem fora de ordem (para testar, basta usar o comando → **/testLost**).