**Universidade Federal do ABC**

**Exercício programático 0**

**EP0**

**Aluno: Ramon Neres Teixeira Jardim RA: 11068616**

**Professor: Vladimir Emiliano Moreira Rocha**

**Santo André**

**2019**

**b) Formato da mensagem transferida**

A carga de dados transferida do cliente para o servidor é um objeto java serializado em um vetor de bytes. A classe deste objeto foi nomeada "DadosPacote" e possui 3 propriedades com seus devidos métodos *get*/*set*:

* ***Mensagem****:* Uma String contendo a mensagem enviada;
* ***NumeroSequencia****:* Um inteiro que indica a ordem das mensagens enviadas, utilizado pelo servidor para identificar mensagens repetidas, perdidas ou desordenadas;
* ***IdCliente***: Um inteiro único para cada cliente, serve para que o servidor possa diferenciar cada cliente.

**c) Explicação em “alto nível” do tratamento de mensagens fora de ordem.**

O servidor instancia um novo objeto da classe ClienteBuffer sempre que um pacote de um cliente novo é recebido e todas as mensagens recebidas deste cliente são armazenadas neste objeto criado. O buffer é um vetor de String presente na classe ClienteBuffer e é nesta estrutura que o servidor guarda as mensagens recebidas, sempre armazenando a mensagem no índice igual ao seu *numeroSequencia*. Como o buffer é consumido de forma sequencial, a mensagem no índice 9, por exemplo, só será consumida quando a mensagem no índice 8 for consumida ou atingir o *timeout* (1000 ms).

**d) Explicação em “alto nível” do tratamento de mensagens duplicadas.**

Como dito anteriormente, as mensagens são armazenadas em um vetor de Strings utilizando o *numeroSequencia* como índice. Dessa forma, caso alguma mensagem chegue ao servidor duplicada, ela será armazenada no mesmo índice do buffer, o que não irá afetar no consumo, ou seja, todas as mensagens que forem recebidas com o *numeroSequencia* *=* *x* serão armazenadas na posição *x*, e portanto nunca haverá mais de uma mensagem com o mesmo número de sequência armazenada no buffer.

**e) Explicação em “alto nível” do consumo do buffer e tratamento de mensagens perdidas.**

Um contador itera pelo buffer tentando a todo o momento consumir uma mensagem na posição em que está parado, quando consegue consumir, avança para a próximo posição. Caso passe 1 segundo sem que uma mensagem seja colocada naquela posição, incrementa o contador em 1 de forma a repetir o processo na próxima posição do buffer, assumindo que o pacote que viria a preencher a posição anterior foi perdido.

O consumo consiste em exibir no console a mensagem que estava gravada naquela posição do buffer e gravá-la em um arquivo, gravando/printando “##MENSAGEM PERDIDA##” caso o pacote seja considerado como perdido.

Ao perder 5 mensagens seguidas de um cliente, o servidor finaliza a execução para aquele cliente e considera que a transmissão foi concluída.

**f) Links dos lugares de onde baseou seu código.**

Os métodos para serialização e desserialização de um objeto para uma array de bytes foi retirado de: <https://stackoverflow.com/questions/2836646/java-serializable-object-to-byte-array>.

O código para escrever texto em arquivos foi baseado em: <https://www.javatpoint.com/java-filewriter-class>.

Os códigos para envio e recebimento de pacotes foram baseados nos slides de aula e em implementações feitas na disciplina de Redes de Computadores.