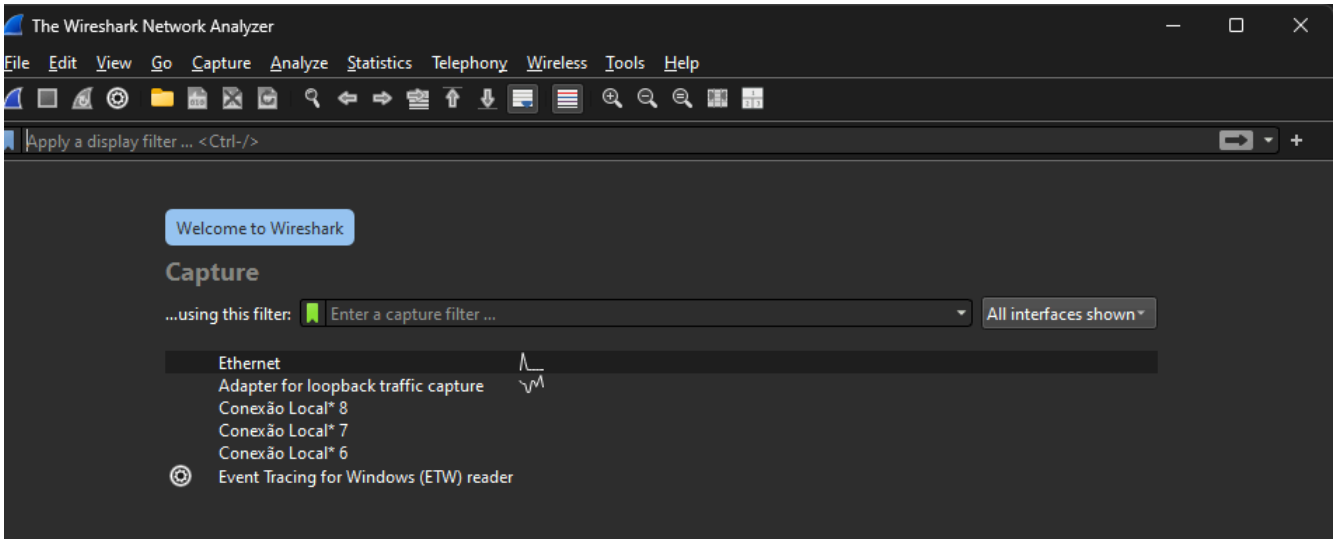
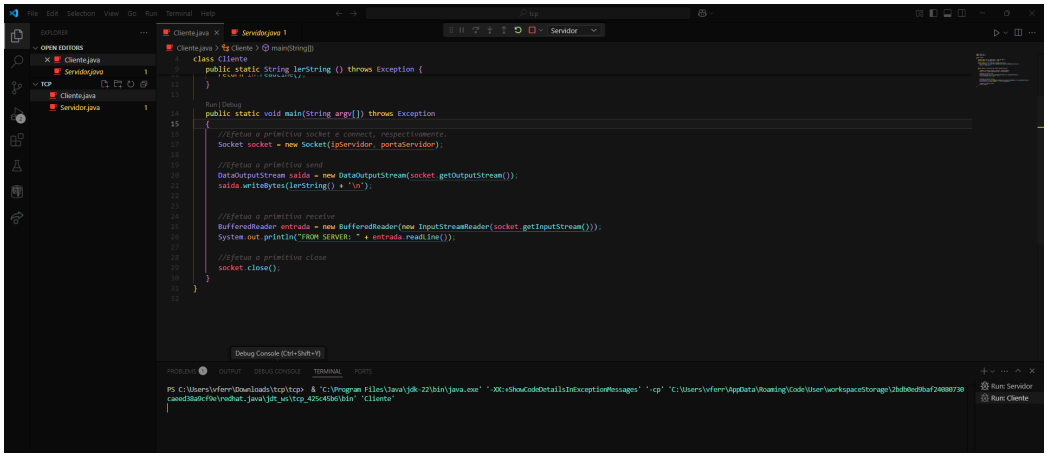
Lista 7 - Identificar os Serviços do TCP com o Wireshark

Nome: Victor Ferraz de Moraes
Matrícula: 802371

Exercício 2:



Exercício 3:

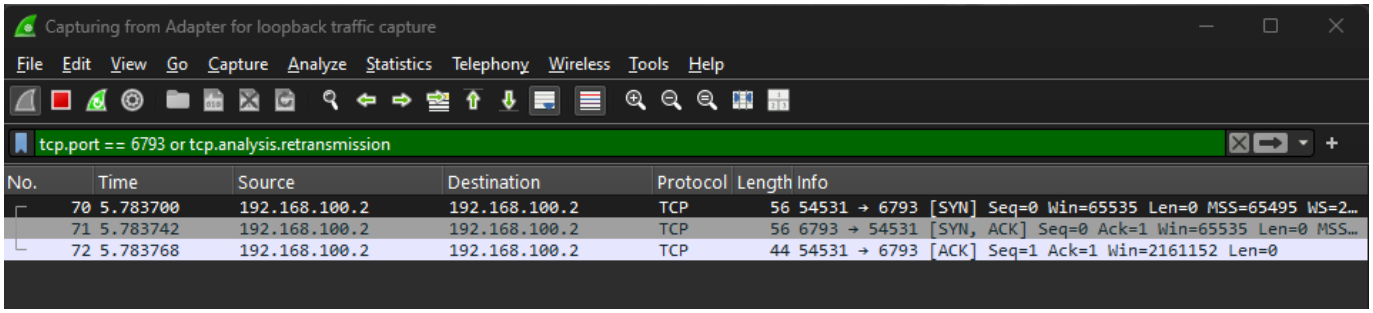


Exercício 4:

a)

1540	88.504081	192.168.100.2	192.168.100.2	TCP	45 54531 → 6793 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=1
1541	88.504105	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=1 Ack=2 Win=2161152 Len=0
1542	88.504269	192.168.100.2	192.168.100.2	TCP	45 54531 → 6793 [PSH, ACK] Seq=2 Ack=1 Win=2161152 Len=1
1543	88.504279	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=1 Ack=3 Win=2161152 Len=0
1544	88.504302	192.168.100.2	192.168.100.2	TCP	45 54531 → 6793 [PSH, ACK] Seq=3 Ack=1 Win=2161152 Len=1
1545	88.504309	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=1 Ack=4 Win=2161152 Len=0
1546	88.504327	192.168.100.2	192.168.100.2	TCP	45 54531 → 6793 [PSH, ACK] Seq=4 Ack=1 Win=2161152 Len=1
1547	88.504333	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=1 Ack=5 Win=2161152 Len=0
1548	88.504661	192.168.100.2	192.168.100.2	TCP	45 6793 → 54531 [PSH, ACK] Seq=1 Ack=5 Win=2161152 Len=1
1549	88.504673	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=2 Win=2161152 Len=0
1550	88.504696	192.168.100.2	192.168.100.2	TCP	45 6793 → 54531 [PSH, ACK] Seq=2 Ack=5 Win=2161152 Len=1
1551	88.504703	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=3 Win=2161152 Len=0
1552	88.504719	192.168.100.2	192.168.100.2	TCP	45 6793 → 54531 [PSH, ACK] Seq=3 Ack=5 Win=2161152 Len=1
1553	88.504725	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=4 Win=2161152 Len=0
1554	88.504740	192.168.100.2	192.168.100.2	TCP	45 6793 → 54531 [PSH, ACK] Seq=4 Ack=5 Win=2161152 Len=1
1555	88.504746	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=5 Win=2161152 Len=0
1556	88.504794	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [FIN, ACK] Seq=5 Ack=5 Win=2161152 Len=0
1557	88.504801	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=6 Win=2161152 Len=0
1558	88.506706	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [FIN, ACK] Seq=5 Ack=6 Win=2161152 Len=0
1559	88.506734	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=6 Ack=6 Win=2161152 Len=0

b)



c)

1556	88.504794	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [FIN, ACK] Seq=5 Ack=5 Win=2161152 Len=0
1557	88.504801	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [ACK] Seq=5 Ack=6 Win=2161152 Len=0
1558	88.506706	192.168.100.2	192.168.100.2	TCP	44 54531 → 6793 [FIN, ACK] Seq=5 Ack=6 Win=2161152 Len=0
1559	88.506734	192.168.100.2	192.168.100.2	TCP	44 6793 → 54531 [ACK] Seq=6 Ack=6 Win=2161152 Len=0

d)

Forçado a alterar o Window Size mandando dados contínuos e diminuindo o processamento do servidor.

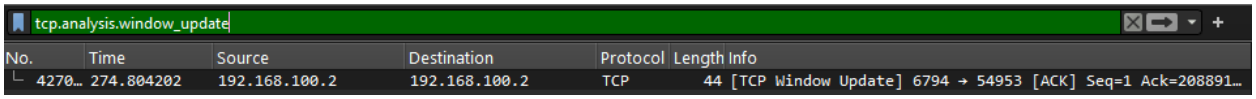
Cliente:

```
Run | Debug
public static void main(String argv[]) throws Exception
{
   //Efetua a primitiva socket e connect, respectivamente.
   Socket socket = new Socket(ipServidor, portaServidor);

   //Efetua a primitiva send
   DataOutputStream saida = new DataOutputStream(socket.
   getOutputStream());
   for (int i = 0; i < 10000; i++) {
      saida.writeBytes("Mensagem número " + i + "\n");
   }
```

Servidor:

```
String linha;
while ((linha = entrada.readLine()) != null) {
   System.out.println("Recebido: " + linha);
   Thread.sleep(millis:100); // processa devagar
}
```



Exercício 5:

a) É possível observar que para cada pacote ACK, os números ACK/SEQ são atualizados, indicando que cada parte dos dados foi transferida. Além disso, ao usar o filtro tcp.analysis.retransmission também é possível concluir que não houve retransmissões de pacotes por parte do TCP.

b) O three-way handshake é realizado no início da conexão, ao iniciar o Cliente. Nele, é possível ver o lado do Cliente enviando um pacote com um SYN (deseja conectar), em seguida o servidor responde com SYN e ACK (Ele deseja se conectar e confirma a chegada do pedido) e então, por fim, ACK do lado do cliente (confirmando o recebimento do servidor).

c) Também conhecido como four-way termination, ele é realizado da seguinte forma: O lado A pretende terminar a conexão enviando um FIN para o lado B. O lado B reconhece que recebeu a informação com um ACK. Após isso, o lado B também envia um FIN para o lado A que, por fim, reconhece o envio com um ACK e encerra a conexão.

d) Ao forçar que o servidor não processe os dados tão rapidamente e ao enviar dados contínuos, é possível ver a mudança de Window size, ou seja, uma maneira de controlar a quantidade de dados que pode ser recebida pelo servidor sem a necessidade de um ACK. Isto é feita de maneira automática pelo protocolo TCP a fim de não sobrecarregar o buffer do receptor.