

ABNER ENOQUE MONTEIRO SILVA GABRIEL BRITO DA CRUZ

TRABALHO 1: COMUNICAÇÃO ENTRE PROCESSOS.

Professor: Prof. Dr. Antonio Rafael Braga

QUIXADÁ 2025

SUMÁRIO

OBJE	ETIVO DO TRABALLHO	14
QUESTÃO 1QUESTÃO 2		
2.	Teste de implementação utilizando como origem a entrada padrão	19
<i>3</i> .	Teste de implementação utilizando como origem um arquivo	20
QUE	STÃO 3	21
OUESTÃO 4		22

OBJETIVO DO TRABALLHO

O objetivo do trabalho é implementar e testar diferentes formas de comunicação entre processos em sistemas distribuídos utilizando Java. O foco está na criação de serviços remotos e no uso de sockets para transmissão de dados, tanto em comunicação unicast (TCP) quanto multicast (UDP). Além disso, o trabalho envolve a serialização de dados para comunicação eficiente entre cliente e servidor, e a aplicação de técnicas como protocolo buffers, XML ou JSON para representação externa de dados.

Primeiramente, dentro do Model, execute o ServidorTCP.

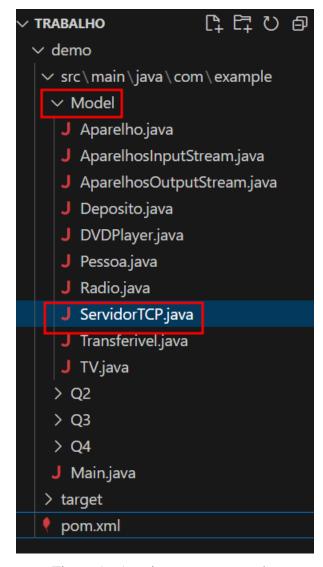


Figura 1 - Arquivo a ser executado.

Após a execução, você receberá uma mensagem informando que o servidor foi iniciado:

Figura 2 - Mensagem do servidor.

O próximo passo é executar a main, que está isolada.



Figura 3 - Arquivo Main.java que deve ser executado. Como resultado o teste utilizando as 3 saídas requiridas no trabalho.

```
__\Documents\SD_2024.2-main\Trabalho> & 'C:\Users\
ions\redhat.java-1.38.0-win32-x64\jre\17.0.13-win32-x86_64\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessage
                                     \Documents\SD_2024.2-main\Trabalho\demo\target\classes' 'com.example.Main'
Samsung 4K adicionado ao depósito Depósito Central
Sony DVD adicionado ao depósito Depósito Central
Philips FM adicionado ao depósito Depósito Central
Depósito: Depósito Central
TV - Modelo: Samsung 4K, Preço: 2500.0, Tamanho da Tela: 55 polegadas.
DVD Player - Modelo: Sony DVD, Preço: 400.0
Rádio - Modelo: Philips FM, Preço: 150.0, Faixa de Frequência: 87.5 - 108 MHz
Enviando dados para saída padr?o:
Samsung 4K2500.Sony DVD400.0
Philips FM150.0
Enviando dados para arquivo:
Dados enviados para o arquivo 'aparelhos.dat'
Enviando dados para servidor TCP:
Dados enviados para o servidor TCP.
```

Figura 4 - Teste com as 3 saídas requeridas.

É possível visualizar que o servidor que iniciamos recebeu os dados mencionados acima.

```
Servidor aguardando conex?es na porta 5000
Conex?o estabelecida com /127.0.0.1
Número de aparelhos recebidos: 3
Aparelho recebido - Modelo: Samsung 4K, Preço: 2500.0
Aparelho recebido - Modelo: Sony DVD, Preço: 400.0
Aparelho recebido - Modelo: Philips FM, Preço: 150.0
```

Figura 5 - Dados recebidos pelo servidor. Na imagem abaixo, é possível ver os dados gravados no arquivo *aparelhos.dat*

Figura 6 - Visualização dos dados no arquvo gerado.

1. Teste de implementação utilizando como destino um servidor remoto

Dentro da pasta da questão, execute o arquivo ServidorTCP.

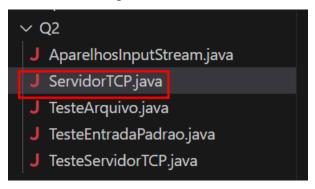


Figura 7 - Arquivo a ser executado.

Após a execução, você receberá uma mensagem informando que o servidor foi iniciado:

```
PS C:\Users\ \ \tag{\text{local} \text{Documents\SD_2024.2-main\Trabalho\demo} & 'C:\Users\ \ \text{.vscode\extensions\redhat} \ .java-1.38.0-win32-x64\jre\17.0.13-win32-x86_64\bin\java.exe' '@C:\Users \ AppData\Local\Temp\cp_ah5xhah5to1ry604smx0w3zv \ 4.argfile' 'com.example.Q2.ServidorTCP' \ Servidor aguardando conex?es na porta 5000
```

Figura 8 - Servidor em espera.

Em seguida, execute o arquivo da imagem abaixo:



Figura 9 - Arquivo responsável pelo teste do servidor.

Segue abaixo o resultado do teste.

Figura 10 - Retorno do servidor.

2. Teste de implementação utilizando como origem a entrada padrão

Execute o arquivo da imagem abaixo:

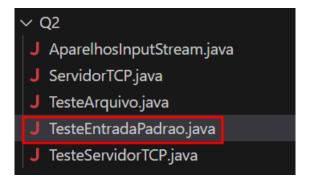


Figura 11 - Arquivo de teste a ser executado. Em seguida é pedido a quantidade de aparelhos e inseridos os dados necessários.

```
PS C:\Users\ \ \Documents\SD_2024.2-main\Trabalho\demo> & 'C:\Users\ \ \text{xtensions\redhat.java-1.38.0-win32-x64\jre\17.0.13-win32-x86_64\bin\java.exe' '@C:\Users\ \ \text{AppData\Local\Temp\cp_ah5xhah5to1ry604smx0w3zv4.argfile' 'com.example.Q2.TesteEntradaPadrao'} 
Digite os dados dos aparelhos:
2

Número de aparelhos recebidos: 2

Digite o modelo do aparelho 1:

Apple

Digite o preço do aparelho 1:
2000

Aparelho recebido - Modelo: Apple, Preço: 2000

Digite o modelo do aparelho 2:

HP

Digite o preço do aparelho 2:

1000

Aparelho recebido - Modelo: HP, Preço: 1000
```

Figura 12 - Execução do programa.

3. Teste de implementação utilizando como origem um arquivo

Executamos o arquivo indicado na imagem abaixo para o teste de arquivo:

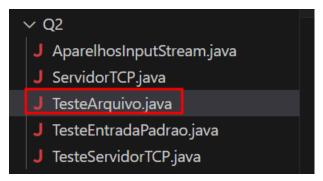


Figura 13 - Inicialização para o teste de arquivo. Ao executar, é possível verificar os dados armazenados no arquivo aparelhos.dat

Figura 14 - Retorno dos dados.

Primeiramente, execute o Servidor.



Figura 15 - Arquivo do servidor.

Após a execução, será exibida uma mensagem em que o servidor aguarda conexão:

Figura 16 - Servidor em execução.

Em seguida, executamos o arquivo referente ao cliente:



Figura 17 - Arquivo de cliente que tentará uma conexão.

Como resultado é possível perceber que nosso cliente consegui enviar corretamente os dados para o servidor.

```
PS C:\Users\\\Documents\SD_2024.2-main\Trabalho\demo> & 'C:\Users\\\.vscode\extensions\redhat.java-1.38.0-win32-x64\jre\17.0.13-win32-x86_64\bin\java.exe' '@C:\Users\\\AppData\Local\\Temp\cp_ah5xhah5to1ry604smx0w3zv4.argfile' 'com.example.Q3.Cliente'

Enviando requisiç?o para o servidor...

Resposta recebida do servidor:

TV - Modelo: Samsung 4K, Preço: 2500.0, Tamanho da Tela: 55 polegadas.
```

Figura 18 - Envio dos dados para o servidor.

Primeiro, executamos nosso servidor de votação:



Figura 19 - Servidor de votação.

Em seguida, é informado que o servidor foi iniciado

```
PS C:\Users\ Documents\trabalho_1_sd-main> & 'C:\Program Files\Java\jdk1.8.0_211\bin\java.exe' '-cp' 'C:\Users\ \AppData\Roaming\Code\User\workspaceStorage\70ea4d5f7638438f320e113d67f2df3d\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'com.example.04.VotingServer'
Servidor de Votaç?o Iniciado...
```

Figura 20 - Servidor de votação iniciado.

Agora podemos executar nosso cliente de votação:



Figura 21 -Cliente de voação.

Como resposta, obtemos que o nosso voto foi armazenado.

```
PS C:\Users \Documents\trabalho_1_sd-main> & 'C:\Program Files\Java\jdk1.8.0_211\bin\java.exe ''-cp' 'C:\Users\ ppData\Roaming\Code\User\workspaceStorage\70ea4d5f7638438f320e113d67f2df3d\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'com.example.Q4.Votingclient'

Bem-vindo, Eleitor!
Candidatos disponíveis: Candidato 1, Candidato 2, Candidato 3
Voto registrado com sucesso!
```

Figura 22 - Realizando votação.