

**Grupo 1 de programas:**

- assista o vídeo "Exemplos de Redes parte\_1.m4v";
- execute o Servidor80, ele ficará esperando um cliente. Em seguida no mesmo computador execute um navegador (e.g. Internet Explorer, Google Chrome ou Firefox), colocando no endereço "localhost" ou "127.0.0.1". Explique o que acontecerá no navegador;
- execute o Servidor80 e em seguida, em outra janela de linha de comandos ("cmd" do Windows), no mesmo computador execute o Cliente80. Observe que os dados serão trocados entre eles;
- no Cliente80 substitua "127.0.0.1" por "[www.fc.unesp.br](http://www.fc.unesp.br)";. Observe que na tela do cliente será exibido um texto parcialmente em HTML;
- explique o que este texto representa e de onde ele vem.

Aparecerá "Este texto aparecerá no browser!" em negrito.

Como o cliente não entende a formatação HTML5, o texto vira sem formatação, mostrando até as instruções HTML5, ele veio do servidor [www.fc.unesp.br](http://www.fc.unesp.br) e representa um erro 400 bad request.

**Grupo 2 de programas:**

- assista o vídeo "Exemplos de Redes parte\_2.m4v";
- execute o programa Servidor (sem 80), ele ficará esperando até três clientes para troca de mensagens. Execute dois programas Cliente (um em cada janela "cmd") e, em seguida, execute o programa ClienteFrame. Neste momento, quatro programas estarão executando ao mesmo tempo, um Servidor, dois Clientes e um ClienteFrame;
- digite em qualquer um dos clientes uma mensagem e verifique o que ocorre.

A mensagem será impressa em ambos os clientes, como se fosse um chat mesmo.

**1. O que faz a instrução: "serverSocket.accept();" no Servidor?**

Espera uma conexão cliente ser feita ao serverSocket, fica travada até um cliente ser aceito.

**2. Ainda no Servidor, para que servem as instruções abaixo?**

- **Scanner is = new Scanner(clientSocket.getInputStream());**

Cria uma instância is de Scanner recebendo no seu construtor o fluxo de dados de entrada em bits do clientSocket, transformando esse fluxo de bits em algo que possamos entender, uma cadeia de caracteres ou números.

- `os[cont++] = new PrintStream(clientSocket.getOutputStream());`

vetor instancia os recebe no construtor do PrintStream o fluxo de dados de saída em bits do clientSocket, transforma em uma fluxo imprimível conveniente, como exemplo uma cadeia de caracteres ou numeros.

## **2. Como é possível enviar a mesma mensagem para três clientes praticamente ao mesmo tempo?**

No programa exemplo, o servidor utiliza um for que percorre todos os clientes enviando a mensagem para cada um, além do uso de threads no cliente e servidor, fazendo com que assim que o cliente envie uma mensagem o servidor já a recebe e repassa para todos os outros clientes.

## **4. Qual a vantagem do ClienteFrame em relação ao Cliente?**

O chat fica mais legível e organizado, no Cliente sem frame, assim que enviamos uma mensagem ele já recebe ela no cmd com isso fica uma coisa estranha e que nada parece um chat. Agora com uma Frame podemos fazer um campo para digitar a mensagem e outro para receber as mensagens.