



Universidade Federal de São Carlos
Campus Sorocaba

T1 Redes

Grupo 02

OPÇÃO 4 – Mensagem Instantânea

Alunos:

Caio Fernando Peres 769298

Eduardo Ravagnani de Melo 771004

Leonardo Valerio Morales 771030

Luís Felipe Dobner Henriques 771036

Desenvolvimento	3
Estrutura Conexão	3
Cliente	3
Servidor	3
Thread Escrita	3
Thread Leitura	3
Funcionamento do Programa	4
Erros conhecidos	4
Pacote do Programa	4
Compilação do Programa	5
Interface	5

Desenvolvimento

Estrutura Conexão

Como faremos um programa P2P, qualquer um dos dois usuários poderá exercer a função de hospedeiro e de cliente, por isso o programa oferece ambas as opções para o usuário e age de acordo com a escolha. Antes do usuário escolher entre ser Cliente e Servidor pedimos seu nome e além disso criamos um Socket global, que receberá o socket de conexão independentemente da escolha.

Cliente

Caso o usuário escolha ser cliente, deve especificar qual o ip do usuário que está agindo como hospedeiro, após isso um objeto Socket("ip", 7777) será criado para cuidar dessa conexão, este socket possui o ip do host e a porta padrão do programa 7777. Este socket será atribuído ao global que será passado para os dois threads de Escrita e Recepção.

Servidor

Caso o usuário escolha ser hospedeiro, ele apenas seleciona a opção e o sistema um objeto ServerSocket("7777") para funcionar como host, o programa então fica em espera por uma conexão. Este socket será atribuído a uma referência global Socket que será passado para os dois threads de Escrita e Recepção.

Thread Escrita

O thread de escrita é iniciado em paralelo com o de recepção e exerce o comportamento padrão de receber uma string vinda da interface do programa e passar para a OutputStream do Socket global, assim jogando a string para saída do socket, logo depois da um flush(), que envia a string como um todo para o destino do Socket global. É importante dizer que como o usuário especificou seu nome no começo do programa, neste momento a string enviada é uma concatenação de nome + : + mensagem, assim mandando a mensagem juntamente com o nome de quem a enviou.

Thread Leitura

O thread de leitura é iniciado e paralelo com o de escrita, e exerce a função contrária, quando recebe uma string na InputStream do Socket global, decompõe a mesma e a envia para a interface, assim possibilitando que a interface mostre a mensagem do usuário.

Funcionamento do Programa

Quando se inicia o programa, uma aba do terminal irá abrir juntamente com a interface. A aba do terminal pode ser fechada. Na interface uma tela de escolha irá aparecer para o usuário, nesta tela o mesmo pode escolher entre ser um servidor ou um cliente, caso escolha um servidor, o programa irá chamar a função `socket.accept()` e irá esperar uma conexão do cliente, isso faz com que o programa aparenta não estar respondendo, contudo ele apenas está esperando o usuário cliente se conectar, depois de 10 segundos ocorre um timeout e o programa volta ao estado de escolha.

Nota: O programa foi testado e comprovado o funcionamento em seu servidor local, e deve funcionar com conexões externas devido ao funcionamento padrão do `ServerSocket` (Não foi testado por esse meio por exigir port-forwarding e liberação do firewall).

Após estabelecida a conexão, o usuário se vê a frente de um espaço onde as mensagens enviadas e recebidas são mostradas, uma caixa de texto onde pode escrever suas mensagens e um botão de enviar. Ele fará utilização destes elementos para se comunicar com o outro usuário.

Erros conhecidos

O único erro conhecido se manifesta quando um cliente tenta se conectar a um servidor que não existe ou não está disponível. A interface mostra 2 janelas de aviso ao invés de apenas 1, caso o usuário clique no OK de qualquer uma das janelas o programa irá fechar como esperado.

Pacote do Programa

O programa será enviado compilado em bytecode e com a biblioteca de interface gráfica que utilizamos, o `JavaFX` para o windows, que é necessário para rodar o programa já compilado. Para facilitar a execução incluímos um arquivo `.bat`, que executa o comando do java específico juntamente com a biblioteca.

Para executar basta executar o arquivo `rodar.bat`.

Os códigos fonte podem também ser encontrados no diretório base na pasta `codigo`.

Compilação do Programa

O programa foi compilado usando o módulo externo do javafx, através da IntelliJ IDE, com o seguinte comando:

```
javac --module-path "javafx-sdk-11.0.2\lib" --add-modules  
javafx.controls,javafx.fxml *.java
```

Depois, foi transformado em .jar com o seguinte comando:

```
jar cf nomeJar.jar *
```

Para executar o programa, é possível apenas rodar o rodar.bat, ou executar o seguinte código no prompt de comando estando no diretório do programa:

```
java --module-path "javafx-sdk-11.0.2\lib" --add-modules  
javafx.controls,javafx.fxml,javafx.graphics,javafx.web -jar  
AplicacaoMensagem.jar
```

Interface

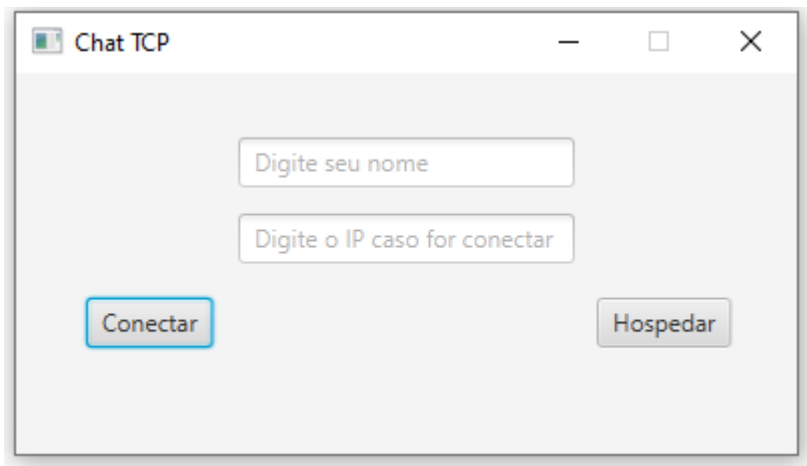


Imagem 1: tela inicial onde o usuário deve escrever seu nome e escolher hospedar ou se conectar a um servidor

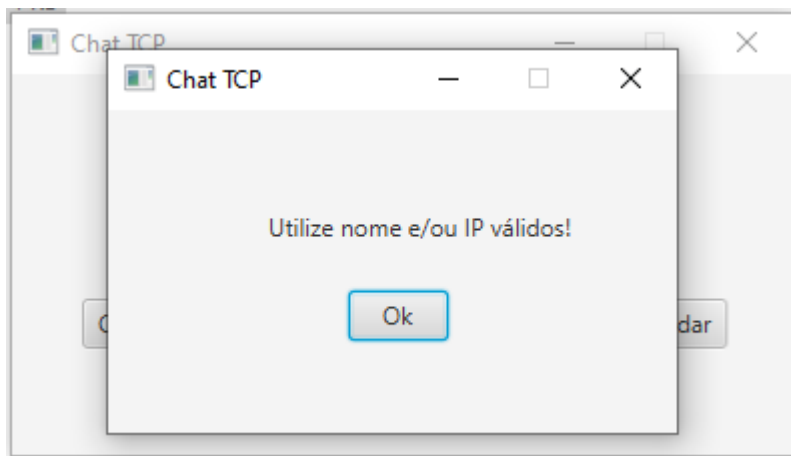


Imagem 2: mensagem de erro caso o usuário não insira um nome ou insira um IP inválido

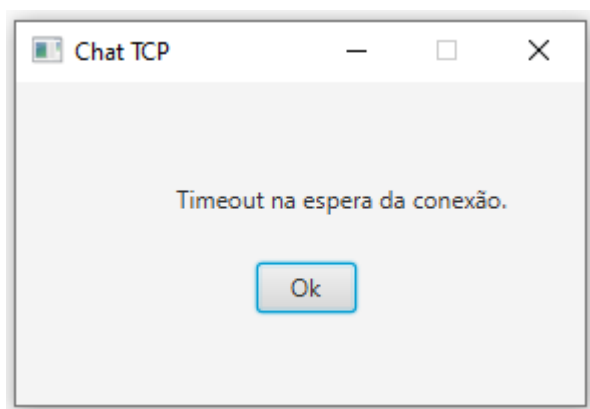


Imagem 3: mensagem de erro caso o tempo limite de espera do host foi excedido.

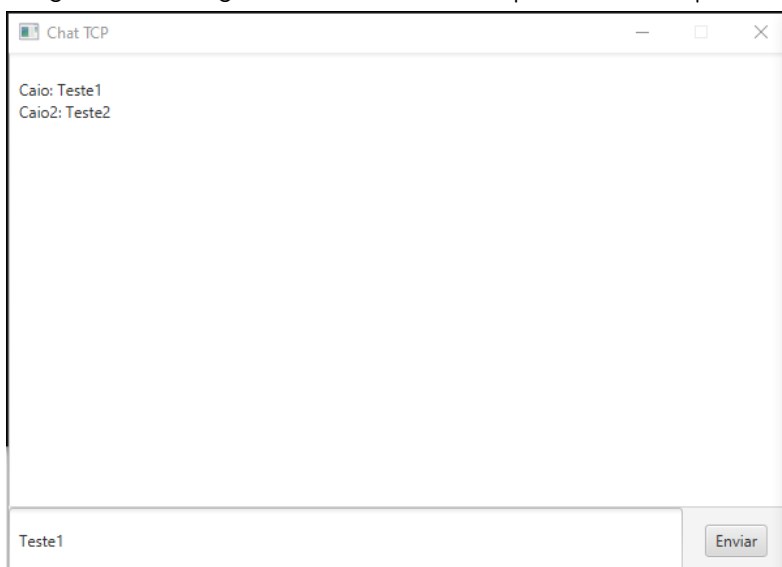


Imagem 4: Tela de Chat do usuário 1 (Caio).

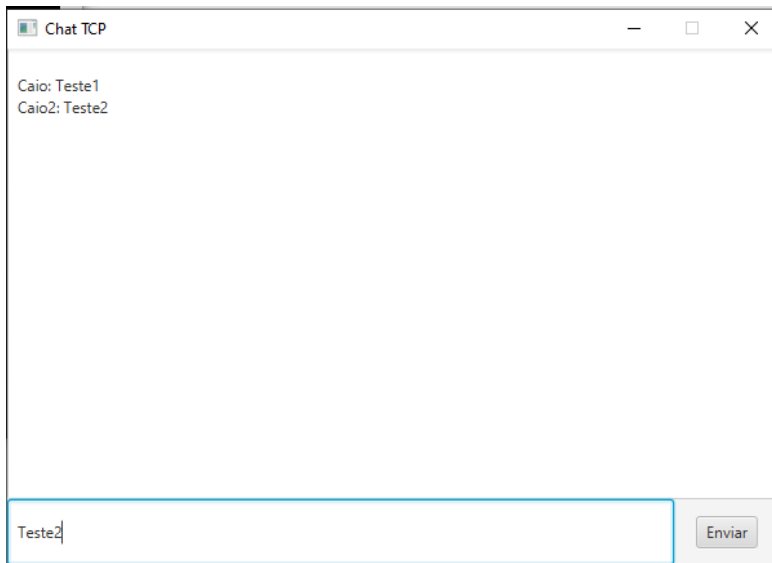


Imagem 5: Tela de Chat do usuário 2 (Caio2).

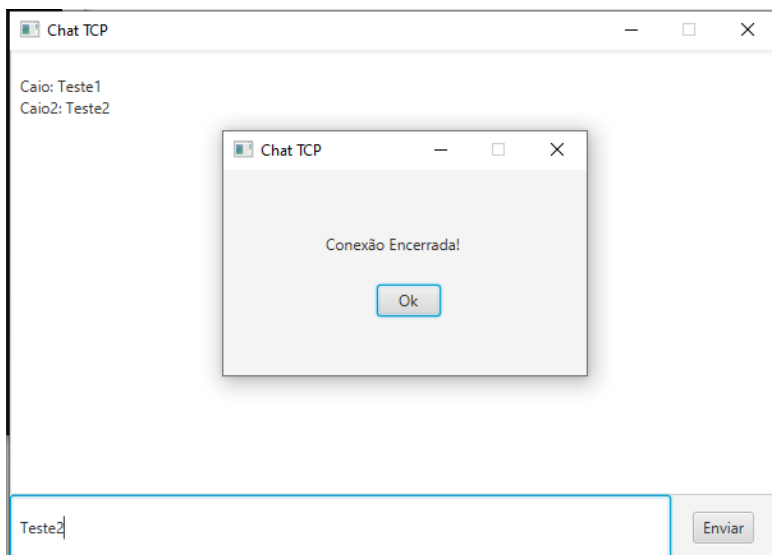


Imagem 6: Popup de encerramento que ocorre quando o outro usuário fecha o programa.