

Relatório

Thaís Félix dos Santos¹, Renan Henrique Cardoso²

¹Engenharia de Computação – Universidade Federal do Ceará (UFC)

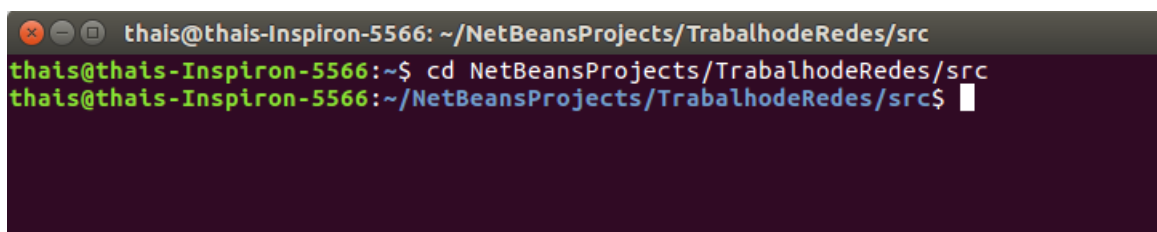
1. Introdução

O trabalho trata da elaboração de Sockets de rede, para uso dessa prática, foi necessário a implementação de dois clientes e um servidor. Os clientes enviam informações em forma de texto para o servidor, logo, o servidor recebe e exibe as mensagens. Para que essa conexão seja estabelecida, é necessário o endereço do servidor que nesse caso foi usado o IP de loopback 127.0.0.1, pois estão sendo executados na mesma máquina. A porta usada para a conexão Socket foi a 50000. As portas do protocolo TCP/UDP variam entre 0 e 65535, deve-se utilizar a porta 1024 em diante, números abaixo deste são reservados para o uso do sistema.

2. Como executar

O sistema operacional usado foi o Ubuntu com Java já instalado. O primeiro passo é abrir três instâncias do terminal com o comando CTRL+ALT+T. Logo em seguida, direcionar até a pasta onde contém o projeto com os arquivos: Servidor.java, Cliente1.java e Cliente2.java. Com o seguinte comando nos três terminais:

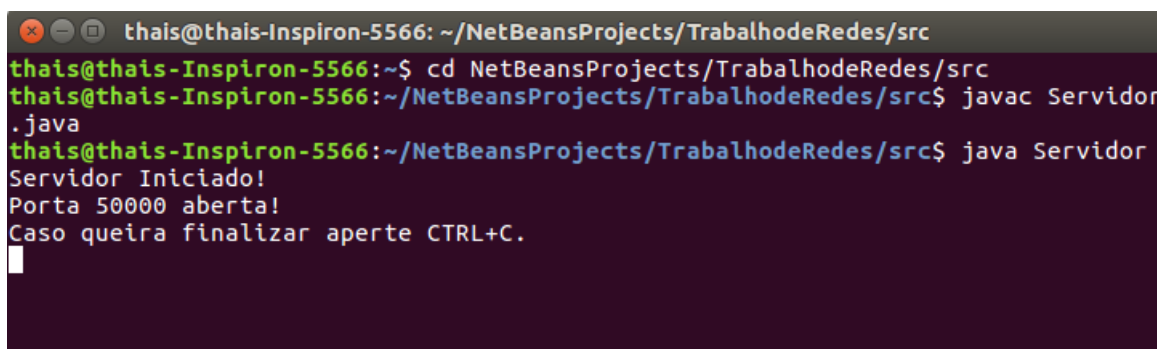
- cd NetBeansProjects/TrabalhodeRedes/src



```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$
```

Um terminal será para compilar e executar o Servidor. Logo, digite os comandos:

- javac Servidor.java
- java Servidor



```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Servidor
.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Servidor
Servidor Iniciado!
Porta 50000 aberta!
Caso queira finalizar aperte CTRL+C.
```

Agora o Servidor foi iniciado e direcionou para a porta aberta 50000.

Nos outros dois terminais será para compilar e executar os dois clientes, em um terminal digite:

- javac Cliente1.java
- java Cliente1

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Cliente1.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Cliente1
0 cliente 1 se conectou!
127.0.0.1
Assunto:
█
```

Com isso, o cliente 1 se conecta ao servidor, e espera o usuário digitar alguma mensagem. Sendo o Assunto opcional e a Mensagem obrigatória, caso o usuário não preencha o local destinado a mensagem é enviada uma mensagem pedindo para o usuário digitar alguma mensagem.

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Cliente1.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Cliente1
0 cliente 1 se conectou!
127.0.0.1
Assunto:
Mensagem:
Digite uma Mensagem!
Mensagem:
█
```

Depois disso, no outro terminal digite os comandos para conectar o Cliente2:

- javac Cliente2.java
- java Cliente2

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Cliente2.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Cliente2
0 cliente 2 se conectou!
127.0.0.1
Assunto:
█
```

Ao ser enviada as mensagens pelos clientes, o servidor exibe o que foi digitado. Depois os clientes podem continuar enviando mensagens e o servidor continuará recebendo-as.

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Servidor.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Servidor
Servidor Iniciado!
Porta 50000 aberta!
Caso queira finalizar aperte CTRL+C.
Nova conexão com cliente 127.0.0.1
Nova conexão com cliente 127.0.0.1
Cliente 1
Assunto: Sem Assunto
Mensagem: teste1
Cliente 2
Assunto: assunto2
Mensagem: teste2

```

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Cliente1.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Cliente1
0 cliente 1 se conectou!
127.0.0.1
Assunto:
Mensagem:
Digite uma Mensagem!
Mensagem:
teste1
Assunto:

```

```
thais@thais-Inspiron-5566: ~/NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~$ cd NetBeansProjects/TrabalhodeRedes/src
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ javac Cliente2.java
thais@thais-Inspiron-5566:~/NetBeansProjects/TrabalhodeRedes/src$ java Cliente2
0 cliente 2 se conectou!
127.0.0.1
Assunto:
assunto2
Mensagem:
teste2
Assunto:

```

References

- Caelum. Apêndice-Sockets. <https://www.caelum.com.br/apostila-java-orientacao-objetos/apendice-sockets/19-5-servidor>. Acesso em: 07/06/2018.
- DEVMEDIA. Java Sockets: Criando comunicações em Java. <https://www.devmedia.com.br/java-sockets-criando-comunicacoes-em-java/9465>. Acesso em: 07/06/2018.
- GUJ. Conexão de multiplos sockets em um servidor. <http://respostas.guj.com.br/23004-conexao-de-multiplos-sockets-em-um-servidor>. Acesso em: 07/06/2018.
- Slide Share. <https://pt.slideshare.net/marciopalheta/implementao-de-soc>. Acesso em: 07/06/2018.