

Manual de execução do Socket SERVIDOR-CLIENTE através da linguagem de programação Java, usando o sistema operacional Linux.

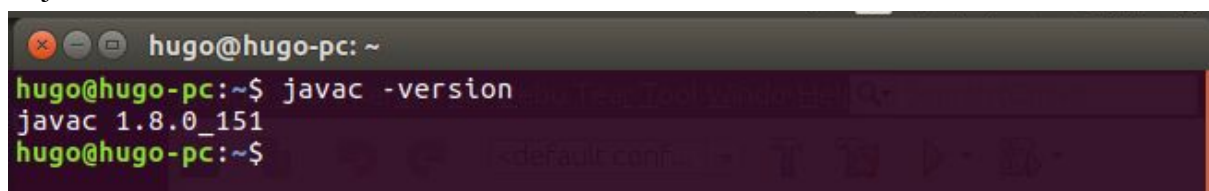
Desenvolvedores: Hugo Silveira Sousa e Kamila Amélia Sousa Gomes

Hoje, uma das tecnologias utilizadas para comunicação entre máquinas são os sockets. Eles podem ser implementadas tanto em diversas linguagens de programação, quanto em diversos sistemas operacionais. No caso implementado a seguir, temos um exemplo de um socket criado em Java no sistema operacional Linux (Ubuntu). Para uso dessa aplicação foi necessário a implementação de um servidor e um cliente. O cliente envia uma informação em forma de texto, o servidor recebe, e exibe. Para que esta conexão seja estabelecida é necessário o conhecimento dos caminhos do servidor e cliente, para os dois, tanto o endereço do servidor, como o caminho para a conexão do cliente com o servidor, foi usado o endereço 127.0.0.1(localhost) pois estão sendo executados na mesma máquina, caso fosse feito em máquinas diferentes, seria necessário a troca desse parâmetro, já a porta utilizada para a conexão socket 40000 pois é uma das portas registradas que estão entre 1024 a 49151. Na parte da comunicação o cliente envia a mensagem por essa porta, e o servidor fica “escutando” a mesma.

Estando no Linux, abra um terminal com o seguinte atalho CRTL+ALT+T, e faça novamente o atalho para abrir outro terminal(um será o servidor e o outro será o cliente).

Digite o seguinte comando, em um dos terminais, para saber se a sua máquina tem o compilador Java instalado;

→ javac -version

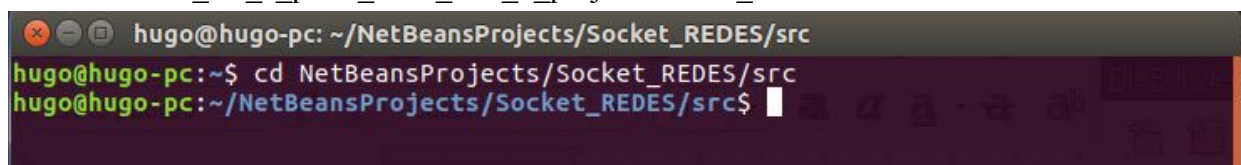


```
hugo@hugo-pc: ~  
hugo@hugo-pc:~$ javac -version  
javac 1.8.0_151  
hugo@hugo-pc:~$
```

Caso não apareça a versão do compilador instalado, irá a mensagem de que o comando não é reconhecido, vá até o site do Java(Como instalar?) e o faça a instalação.

Logo após, deve ser feito o direcionamento do caminho onde está o projeto com o Servidor.java e o Cliente.java, com o seguinte comando, nos dois terminais:

→ cd caminho_ate_a_pasta_onde_esta_o_projeto/Socket_REDES/src



```
hugo@hugo-pc: ~/NetBeansProjects/Socket_REDES/src  
hugo@hugo-pc:~$ cd NetBeansProjects/Socket_REDES/src  
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$
```

Em um dos terminais iremos compilar e executar o Servidor, com os seguintes comandos:

→ javac Servidor.java

→ java Servidor

```
hugo@hugo-pc: ~/NetBeansProjects/Socket_REDES/src
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ javac Servidor.java
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ java Servidor
Servidor Iniciado
Porta 40000 aberta
Para finalizar digite CTRL+C
█
```

Agora o servidor foi iniciado, e já aguarda a conexão do cliente.

No outro terminal execute os seguintes comandos para compilar e executar o cliente.

→ javac Cliente.java

→ java Cliente

```
hugo@hugo-pc: ~/NetBeansProjects/Socket_REDES/src
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ javac Cliente.java
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ java Cliente
Cliente criado
Cliente conectado ao servidor
Para finalizar digite CTRL+C
ASSUNTO:
█
```

Neste ponto, o cliente já está conectado e já espera o usuário digitar uma mensagem, no qual deve conter um assunto (opcional) e o corpo de mensagem (obrigatório). Como no exemplo:

```
hugo@hugo-pc: ~/NetBeansProjects/Socket_REDES/src
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ javac Cliente.java
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ java Cliente
Cliente criado
Cliente conectado ao servidor
Para finalizar digite CTRL+C
ASSUNTO:
assunto teste
MENSAGEM:
mensagem teste
ASSUNTO: Agora o servidor foi iniciado, e já aguarda a conexão do cliente
█

hugo@hugo-pc: ~/NetBeansProjects/Socket_REDES/src
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ javac Servidor.java
hugo@hugo-pc:~/NetBeansProjects/Socket_REDES/src$ java Servidor
Servidor Iniciado
Porta 40000 aberta
Para finalizar digite CTRL+C
Cliente 127.0.0.1 conectado
ASSUNTO: assunto teste
MENSAGEM: mensagem teste
█
```

Ao ser enviado a mensagem pelo cliente, o servidor exibe o que foi digitado. Depois o cliente pode continuar enviando mensagens e o servidor continuará recebendo-as.

Por fim, para finalizar a conexão, digite CTRL+C.