

Tutorial (por Estêvão Bissoli Saleme)

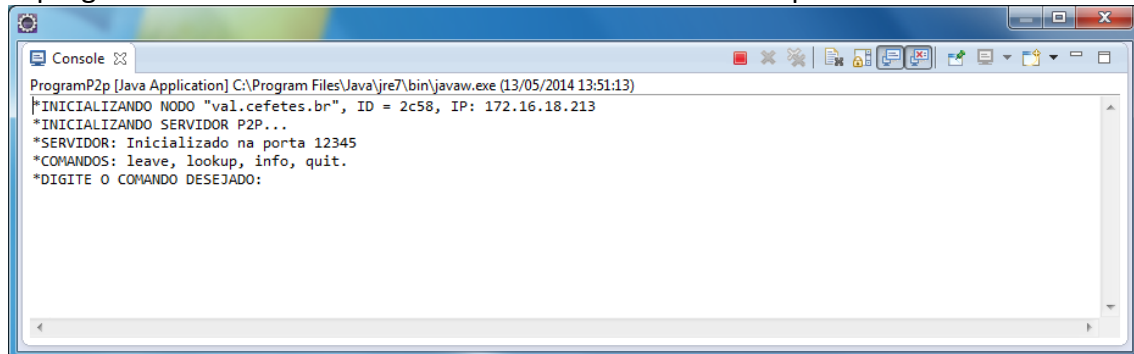
O tutorial está estruturado em passos com exemplos de uso de todos os comandos disponíveis no programa.

Requisitos: Sistema operacional Windows ou Linux e Java 7

Passo 1) Execute o arquivo jar “p2p.jar” no terminal com o seguinte comando:

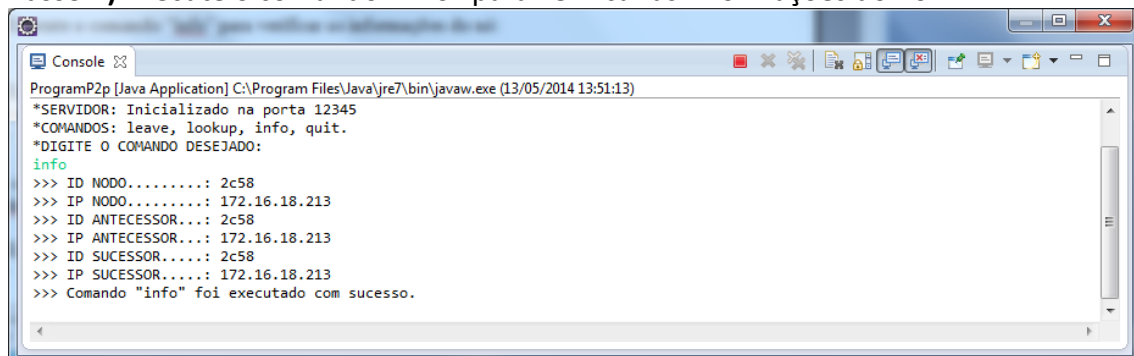
`java -jar p2p.jar`

O programa será inicializado e uma lista de comandos será apresentada:



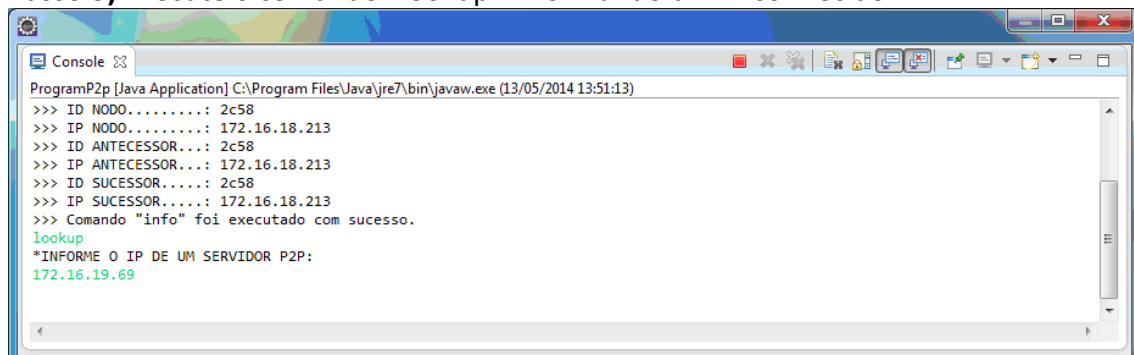
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*INICIALIZANDO NODO "val.cefetes.br", ID = 2c58, IP: 172.16.18.213
*INICIALIZANDO SERVIDOR P2P...
*SERVIDOR: Inicializado na porta 12345
*COMANDOS: leave, lookup, info, quit.
*DIGITE O COMANDO DESEJADO:
```

Passo 2) Execute o comando “info” para verificar as informações do nó:



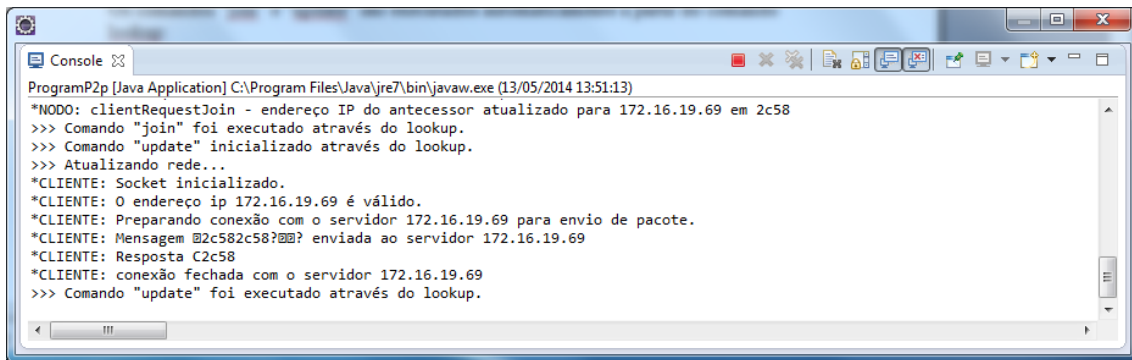
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*SERVIDOR: Inicializado na porta 12345
*COMANDOS: leave, lookup, info, quit.
*DIGITE O COMANDO DESEJADO:
info
>>> ID NODO.....: 2c58
>>> IP NODO.....: 172.16.18.213
>>> ID ANTECESSOR...: 2c58
>>> IP ANTECESSOR...: 172.16.18.213
>>> ID SUCESSOR.....: 2c58
>>> IP SUCESSOR.....: 172.16.18.213
>>> Comando "info" foi executado com sucesso.
```

Passo 3) Execute o comando “lookup” informando um IP conhecido:



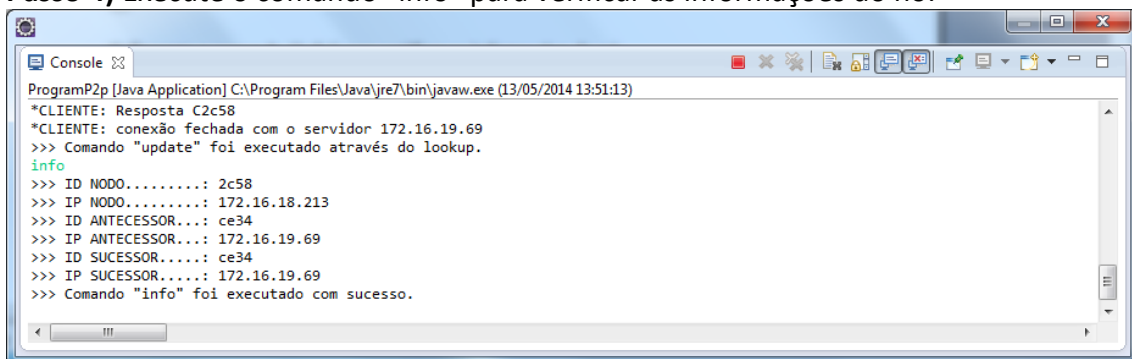
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
>>> ID NODO.....: 2c58
>>> IP NODO.....: 172.16.18.213
>>> ID ANTECESSOR...: 2c58
>>> IP ANTECESSOR...: 172.16.18.213
>>> ID SUCESSOR.....: 2c58
>>> IP SUCESSOR.....: 172.16.18.213
>>> Comando "info" foi executado com sucesso.
lookup
*INFORME O IP DE UM SERVIDOR P2P:
172.16.19.69
```

Os comandos “join” e “update” são executados automaticamente a partir do comando lookup:



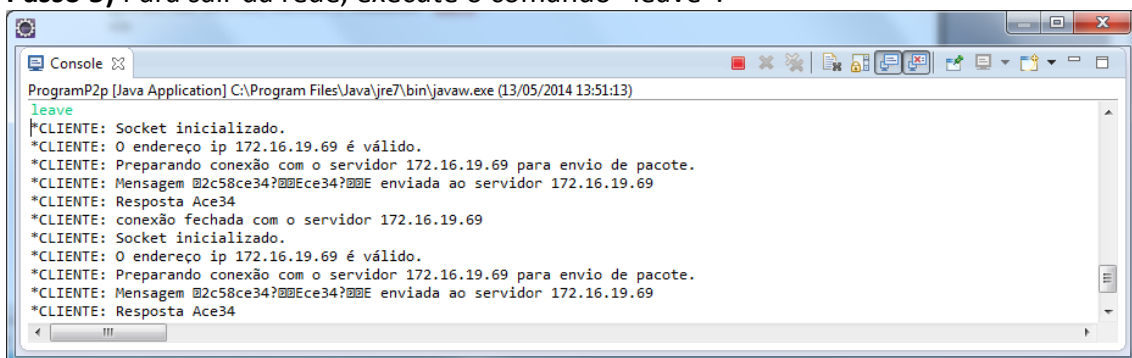
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*NODO: clientRequestJoin - endereço IP do antecessor atualizado para 172.16.19.69 em 2c58
>>> Comando "join" foi executado através do lookup.
>>> Comando "update" inicializado através do lookup.
>>> Atualizando rede...
*CLIENTE: Socket inicializado.
*CLIENTE: O endereço ip 172.16.19.69 é válido.
*CLIENTE: Preparando conexão com o servidor 172.16.19.69 para envio de pacote.
*CLIENTE: Mensagem @2c582c58?@? enviada ao servidor 172.16.19.69
*CLIENTE: Resposta C2c58
*CLIENTE: conexão fechada com o servidor 172.16.19.69
>>> Comando "update" foi executado através do lookup.
```

Passo 4) Execute o comando “info” para verificar as informações do nó:



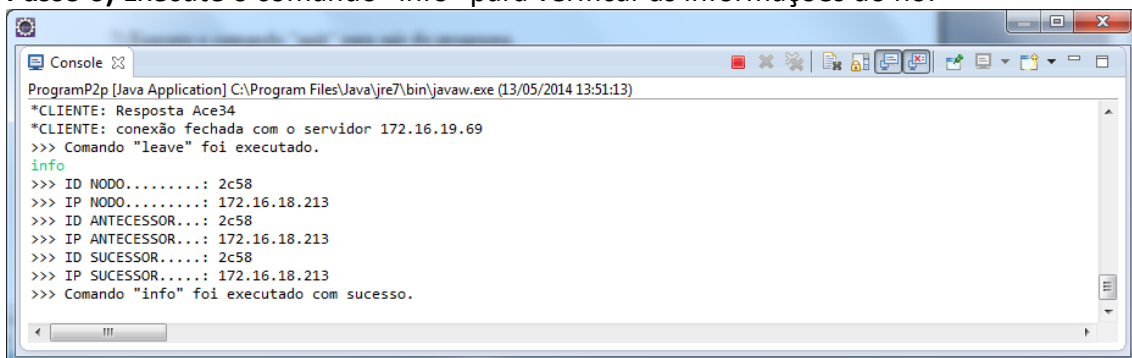
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*CLIENTE: Resposta C2c58
*CLIENTE: conexão fechada com o servidor 172.16.19.69
>>> Comando "update" foi executado através do lookup.
info
>>> ID NODO.....: 2c58
>>> IP NODO.....: 172.16.18.213
>>> ID ANTECESSOR...: ce34
>>> IP ANTECESSOR...: 172.16.19.69
>>> ID SUCESSOR.....: ce34
>>> IP SUCESSOR.....: 172.16.19.69
>>> Comando "info" foi executado com sucesso.
```

Passo 5) Para sair da rede, execute o comando “leave”:



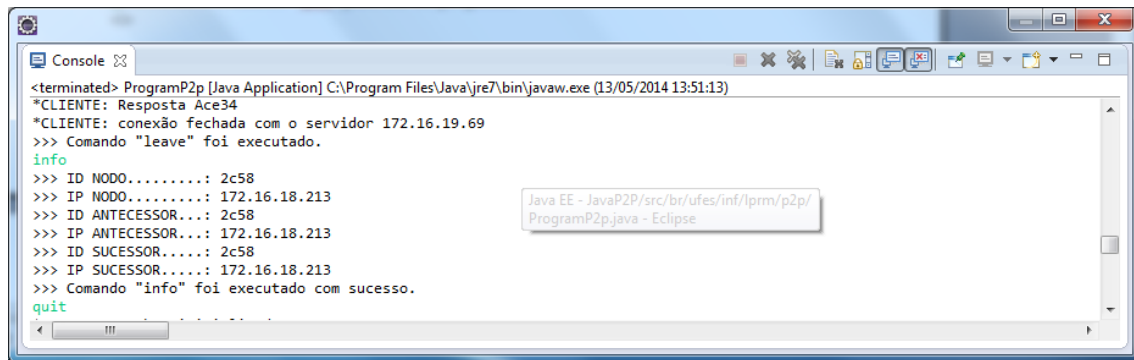
```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
leave
*CLIENTE: Socket inicializado.
*CLIENTE: O endereço ip 172.16.19.69 é válido.
*CLIENTE: Preparando conexão com o servidor 172.16.19.69 para envio de pacote.
*CLIENTE: Mensagem @2c58ce34?@Ece34?@E enviada ao servidor 172.16.19.69
*CLIENTE: Resposta Ace34
*CLIENTE: conexão fechada com o servidor 172.16.19.69
*CLIENTE: Socket inicializado.
*CLIENTE: O endereço ip 172.16.19.69 é válido.
*CLIENTE: Preparando conexão com o servidor 172.16.19.69 para envio de pacote.
*CLIENTE: Mensagem @2c58ce34?@Ece34?@E enviada ao servidor 172.16.19.69
*CLIENTE: Resposta Ace34
```

Passo 6) Execute o comando “info” para verificar as informações do nó:



```
ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*CLIENTE: Resposta Ace34
*CLIENTE: conexão fechada com o servidor 172.16.19.69
>>> Comando "leave" foi executado.
info
>>> ID NODO.....: 2c58
>>> IP NODO.....: 172.16.18.213
>>> ID ANTECESSOR...: 2c58
>>> IP ANTECESSOR...: 172.16.18.213
>>> ID SUCESSOR.....: 2c58
>>> IP SUCESSOR.....: 172.16.18.213
>>> Comando "info" foi executado com sucesso.
```

Passo 7) Execute o comando “quit” para sair do programa.



The screenshot shows a Java console window titled "Console" with a standard Windows-style title bar. The console output includes a termination message, a client response, a connection status message, and a series of commands and responses. A tooltip is visible over the console area, displaying the file path and the application name.

```
<terminated> ProgramP2p [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13/05/2014 13:51:13)
*CLIENTE: Resposta Ace34
*CLIENTE: conexão fechada com o servidor 172.16.19.69
>>> Comando "leave" foi executado.
info
>>> ID NODO.....: 2c58
>>> IP NODO.....: 172.16.18.213
>>> ID ANTECESSOR...: 2c58
>>> IP ANTECESSOR...: 172.16.18.213
>>> ID SUCESSOR.....: 2c58
>>> IP SUCESSOR.....: 172.16.18.213
>>> Comando "info" foi executado com sucesso.
quit
```

Java EE - JavaP2P/src/br/ufes/inf/lprm/p2p/
ProgramP2p.java - Eclipse