

Exercício 8 de MC833 — Programação em Redes de Computadores

Raul Rabelo Carvalho, 105607, turma A

4 de Junho de 2014

1

Na linguagem Java, os vários passos necessários para obter uma conexão TCP do sistema operacional — que em C são executados com chamadas de funções em sequência, como `socket()`, `bind`, `connect`, etc. — são feitos pela implementação da classe `java.net.Socket`. Assim, no cliente de eco para Android, a conexão com o servidor é estabelecida pela linha:

```
socket = new Socket(dstAddress, dstPort);
```

que dá à aplicação um socket conectado ao servidor no endereço *dstAddress* na porta *dstPort*.

2

A classe `java.net.Socket` tem diversos construtores diferentes, incluindo

```
public Socket (String dstName, int dstPort),
```

o qual o argumento `String dstName` aceita tanto um *hostname* quanto um endereço IP, contando que o endereço IP seja passado como uma *string* na forma “x.x.x.x”.

3

O método `getLocalPort()` da classe `java.net.Socket` retorna a porta local pela qual o cliente está conectado na forma de uma *string* que pôde ser concatenada com a mensagem indicando uma conexão bem sucedida.

4

Para implementar a funcionalidade de eco, foi criado um objeto `OnClickListener` para o botão “Enviar”, o qual altera uma flag no objeto `MyClientTask`. No método `doInBackground` do objeto `MyClientTask`, foi criado um *loop* que envia ao servidor a *string* do campo `TextMessage` e recebe a resposta do servidor, imprimindo a resposta.

```
while (true) {  
    if (newline) {  
        response = "";  
        saida.println(editTextMsg.getText().toString());  
        response = entrada.readLine();  
        publishProgress();  
        newline = false;  
    }  
}
```
