

## Exercicio de socket utilizando TCP/IP

1 – Criar o arquivo Servidor.java

```
package br.com.elaborata.sockets.tcp.servidor;
```

```
import java.io.DataInputStream;  
import java.io.DataOutputStream;  
import java.io.IOException;  
import java.net.ServerSocket;  
import java.net.Socket;
```

```
/**
```

```
 * @author Rogue Junior
```

```
 *
```

```
 */
```

```
public class Servidor {
```

```
    public static void main(String[] args) throws IOException{
```

```
        //Criação de objeto que gerencia conexão com cliente
```

```
        ServerSocket conexaoServidor = new ServerSocket(4321, 1);
```

```
        //Recebendo solicitação do cliente
```

```
        Socket cliente = conexaoServidor.accept();
```

```
        //Abrindo canal para receber dados
```

```
        DataInputStream entrada = new DataInputStream(cliente.getInputStream());
```

```
        //Abrindo canal para enviar dados
```

```
        DataOutputStream saida = new DataOutputStream(cliente.getOutputStream());
```

```
        //Recebendo mensagem do cliente
```

```
        String mensagemRecebida = entrada.readUTF();
```

```
        if("Olá!! tudo bem?".equals(mensagemRecebida)){
```

```
            //Exibindo mensagem recebida pelo cliente
```

```
            System.out.println("Mensagem recebida pelo cliente: " +  
mensagemRecebida);
```

```
            //Enviando resposta na requisição do cliente
```

```
            saida.writeUTF("Tudo e você??");
```

```
        }
```

```
        //Fechando canais de entrada e saida e conexão com o cliente
```

```
        saida.close();
```

```
        entrada.close();
```

```
        cliente.close();
```

```
        conexaoServidor.close();
```

```
    }
```

```
}
```

## 2 – Criar o arquivo Cliente.java

```
package br.com.elaborata.sockets.tcp.cliente;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;

/**
 * @author Roque Junior
 */
public class Cliente {

    public static void main(String[] args) throws UnknownHostException, IOException {

        //Abrindo conexão com o IP do servidor
        Socket conexaoCliente = new Socket("127.0.0.1", 4321);

        //Abrindo canal para receber dados
        DataInputStream entrada = new
DataInputStream(conexaoCliente.getInputStream());

        //Abrindo canal para enviar dados
        DataOutputStream saida = new
DataOutputStream(conexaoCliente.getOutputStream());

        String mensagemEnviada = "Olá!! tudo bem?";
        System.out.println("Mensagem que enviaremos ao servidor: " +
mensagemEnviada);

        //Enviando mensagem ao servidor
        saida.writeUTF(mensagemEnviada);

        //Ao receber resposta do servidor, exibe mensagem recebida da tela
        String mensagemRecebida = entrada.readUTF();
        System.out.println("Resposta do servidor: " + mensagemRecebida);

        //Fechando canais de entrada e saida
        saida.close();
        entrada.close();
        //fechando a conexão
        conexaoCliente.close();

    }

}
```

## Exercicio de socket utilizando UDP

### 1 – Criar o arquivo ServidorDatagrama.java

```
package br.com.elaborata.sockets.udp.servidor;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

/**
 * @author Rogue Junior
 */
public class ServidorDatagrama {

    public static void main(String[] args) throws IOException {
        // Criando objeto do tipo DatagramSocket para escutar a porta 2345
        DatagramSocket servidor = new DatagramSocket(2345);

        //Criando datagrama que receberá pacote
        DatagramPacket pacoteRecebido = new DatagramPacket(new byte[512], 512);
        //Fica aguardando até o recebimento de uma nova mensagem
        servidor.receive(pacoteRecebido);

        //Preparando a resposta
        String mensagem = "Resposta do servidor";
        //Transformando String em array de bytes
        byte[] bufferResposta = mensagem.getBytes();
        //Empacotando resposta
        DatagramPacket resposta = new DatagramPacket(bufferResposta,
                                                    bufferResposta.length,
                                                    pacoteRecebido.getAddress(),
                                                    pacoteRecebido.getPort());

        //Enviando pacote
        servidor.send(resposta);
        //Fechando comunicação
        servidor.close();
    }
}
```

## 2 – Criar o arquivo Cliente.java

```
package br.com.elaborata.sockets.udp.cliente;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

/**
 * @author Roque Junior
 *
 */
public class ClienteDatagrama {

    public static void main(String[] args) throws IOException{
        // Criando objeto DatagramaSocket para o cliente -
        // não precisa especificar porta
        DatagramSocket cliente = new DatagramSocket();

        //Definindo endereço do servidor
        InetAddress enderecoServidor = InetAddress.getByName("localhost");
        //Criando mensagem de solicitação que será enviada ao servidor
        String pergunta = "Solicitação do Cliente";
        byte[] bufferPergunta = pergunta.getBytes();
        //Preparando pacote
        DatagramPacket pacotePergunta = new DatagramPacket(
            bufferPergunta,
            bufferPergunta.length,
            enderecoServidor, 2345);

        //Enviando pacote
        cliente.send(pacotePergunta);
        System.out.println("Enviou a pergunta");
        //Preparando pacote para receber resposta do servidor
        DatagramPacket resposta = new DatagramPacket(new byte[512], 512);
        //Recebendo pacote de resposta do servidor
        cliente.receive(resposta);
        //Fechando socket do cliente
        cliente.close();
    }
}
```