

SERVIDOR

```
import java.net.* ;

public class EchoServer {

    static final int PORT = 8000 ;

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

        // Inicializar o socket com o porto de atendimento
        DatagramSocket socket = new DatagramSocket( PORT ) ;

        for(;;) { //ciclo infinito de atendimento...

            // Preparar uma mensagem (datagrama) vazia para recepção...
            byte[] buffer = new byte[65536] ;
            DatagramPacket echoRequest = new DatagramPacket( buffer, buffer.length ) ;

            // Receber a mensagem...
            socket.receive( echoRequest ) ;

            // Obter o conteúdo da mensagem recebida...
            byte[] echoRequestData = echoRequest.getData();
            int echoRequestLength = echoRequest.getLength() ;

            // Preparar uma mensagem para envio com a resposta...
            DatagramPacket echoReply = new DatagramPacket( echoRequestData, echoRequestLength ) ;

            // Indicar o destino da mensagem... (máquina + porto)
            echoReply.setAddress( echoRequest.getAddress() ) ;
            echoReply.setPort( echoRequest.getPort() ) ;

            // Enviar a mensagem...
            socket.send( echoReply ) ;

        }
    }
}
```

CLIENTE

```
import java.io.* ;
import java.net.* ;
import java.util.* ;

public class EchoClient {

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

        if( args.length != 2 ) {
            System.out.printf("usage: java EchoClient máquina_do_servidor porto\n") ;
            System.exit(0);
        }

        // Criar um objecto auxiliar para ler linhas completas do canal de estrada standard
        Scanner in = new Scanner( System.in) ;

        // Preparar endereço e o porto do servidor
        String servidor = args[0] ;
        int port = Integer.parseInt( args[1] ) ;
        InetAddress serverAddress = InetAddress.getByName( servidor ) ;

        // Preparar o socket para trocar mensagens (datagramas)
        DatagramSocket socket = new DatagramSocket() ;

        // Ler o pedido do canal "standard input"
        String request = in.nextLine() ;
        byte[] requestData = request.getBytes() ;

        // Criar a mensagem para enviar
        DatagramPacket echoRequest = new DatagramPacket( requestData, requestData.length ) ;

        echoRequest.setAddress( serverAddress ) ;
        echoRequest.setPort( port ) ;

        socket.send( echoRequest ) ;

        // Criar uma mensagem vazia para receber a resposta
        byte[] buffer = new byte[65536] ;
        DatagramPacket echoReply = new DatagramPacket( buffer, buffer.length ) ;

        socket.receive( echoReply ) ;
        socket.close() ;

        // Constrói uma string com o conteúdo da mensagem
        String echoedMsg = new String( echoReply.getData(), 0, echoReply.getLength() ) ;

        System.out.printf("A resposta foi: \"%s\"\n", echoedMsg ) ;

    }
}
```