INSTITUTO POLITÉCNICO NACIONAL.

ESCUELA SUPERIOR DE CÓMPUTO.

INGENIERÍA DE SOFTWARE.

“Socket Básico de Datagramas”

Que presenta.

Martínez Alvarado Bryan Alexis

Del grupo

3CM16

A cargo del profesor:

Ing. Ricardo Martinez Rosales

Código del Cliente

```
import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.ArrayList;
public class Client {
 public static void main(String[] args) throws InterruptedException {
 try {
 int port = 3500;
 InetAddress host = InetAddress.getByName("127.0.0.1");
  DatagramSocket socket = new DatagramSocket();
 socket.setSendBufferSize(20);
 socket.setReceiveBufferSize(20);
  ByteArrayOutputStream baos = new ByteArrayOutputStream();
 DataOutputStream dos = new DataOutputStream(baos);
```

```
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
 System.out.print("Message: ");
String message = br.readLine();
dos.writeUTF(message);
dos.flush();
 byte[] message_buffer = baos.toByteArray();
 ArrayList<byte[]> buffers = new ArrayList<>();
 int i = 0;
while(i < message_buffer.length && message_buffer[i] != 0x0A) {
byte[] buffer = new byte[20];
 for(int j = 0; j < buffer.length; ++j) {
if(i < message_buffer.length) {
buffer[j] = message_buffer[i++];
} else break;
}
 buffers.add(buffer);
}
 baos.reset();
dos.writeInt(buffers.size());
dos.flush();
 byte[] counter_buffer = baos.toByteArray();
DatagramPacket packet1 = new DatagramPacket(counter_buffer,
counter_buffer.length, host, port);
socket.send(packet1);
 for(byte[] buffer: buffers) {
DatagramPacket packet = new DatagramPacket(buffer, buffer.length, host,
port);
socket.send(packet);
Thread.sleep(10);
}
System.out.println("Packets: " + buffers.size());
 DatagramPacket packet = new DatagramPacket(new byte[20], 20);
socket.receive(packet);
DataInputStream dis = new DataInputStream(new
ByteArrayInputStream(packet.getData()));
 int counter = dis.readInt();
System.out.println("Packets from server: " + counter);
String response = "";
for(i = 0; i < counter; ++i) {
packet = new DatagramPacket(new byte[20], 20);
socket.receive(packet);
//dis = new DataInputStream(new ByteArrayInputStream(packet.getData()));
String partial = new String(packet.getData());
response += partial;
}
```

```
  response = response.substring(2);
 String ans = "";
 for(i = 0; i < response.length(); ++i) {
 if((response.charAt(i) >= 'a' && response.charAt(i) <= 'z') ||
(response.charAt(i) >= 'A' && response.charAt(i) <= 'Z') || response.charAt(i) == ' ' ||
response.charAt(i) == '.' || response.charAt(i) == ',') {
 ans += response.charAt(i);
 }
 }
 System.out.println("From server: " + ans);
  dos.close();
 socket.close();
  } catch(IOException e) {
 e.printStackTrace();
 }
 }
}
```

Código del Servidor

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.util.ArrayList;
public class Server {
 public static void main(String[] args) throws InterruptedException {

 try {

 DatagramSocket socket = new DatagramSocket(3500);

 socket.setReceiveBufferSize(20);

 socket.setSendBufferSize(20);

 System.out.println("Server listen at port: 3500");

  while(true) {

 System.out.println("Waiting for a client...");

  DatagramPacket packet = new DatagramPacket(new byte[20], 20);

 socket.receive(packet);

  DataInputStream dis = new DataInputStream(new

ByteArrayInputStream(packet.getData()));

 ByteArrayOutputStream baos = new ByteArrayOutputStream();

 DataOutputStream dos = new DataOutputStream(baos);

  int counter = dis.readInt();

 System.out.println("packets: " + counter);

  String message = "";

  for(int i = 0; i < counter; ++i) {

 packet = new DatagramPacket(new byte[20], 20);

 socket.receive(packet);

 //dis = new DataInputStream(new

ByteArrayInputStream(packet.getData()));

 String partial = new String(packet.getData());
 message += partial;
  }
 message = message.substring(2);
 System.out.println("message: " + message);
 message = "ECO " + message;
 dos.writeUTF(message);
 dos.flush();
  byte[] message_buffer = baos.toByteArray();
 ArrayList<byte[]> buffers = new ArrayList<>();
  int i = 0;
 while(i < message_buffer.length && message_buffer[i] != 0x0A) {
 byte[] buffer = new byte[20];
 for(int j = 0; j < buffer.length; ++j) {
 if(i < message_buffer.length) {
 buffer[j] = message_buffer[i++];
 } else break;
 }
 buffers.add(buffer);
 }
  baos.reset();
 dos.writeInt(buffers.size());
 dos.flush();
 Thread.sleep(10);
```

```
 byte[] counter_buffer = baos.toByteArray();
 DatagramPacket packet1 = new DatagramPacket(counter_buffer,
counter_buffer.length, packet.getAddress(), packet.getPort());
 socket.send(packet1);
 for(byte[] buffer: buffers) {
 DatagramPacket packet2 = new DatagramPacket(buffer, buffer.length,
packet.getAddress(), packet.getPort());
 socket.send(packet2);
 Thread.sleep(10);
 }
  dos.close();
 baos.close();
 dis.close();
 }
 } catch(IOException e) {
 e.printStackTrace();
 }
 }
}
```