EJEMPLOS PROGRAMACIÓN SOCKET - JAVA

1.- Calcular el Cuadrado –TCP

En esta aplicación el cliente se conecta al servidor, para ello debe introducir la dirección IP del servidor y los parámetros sobre los cuales el servidor debe hacer las operaciones. Una vez conectado con el servidor este toma los parámetros y calcula su cuadrado y lo imprime por pantalla. Posteriormente envía los resultados al cliente, el cual también los imprime por pantalla.

En esta aplicación la conexión se realiza mediante conexión TCP, lo que permite al cliente y al servidor disponer de un stream que facilita una comunicación libre de errores.

El comportamiento para usar este tipo de socket es diferente en el cliente y el servidor. Cada uno de ellos utilizará unos métodos distintos. El esquema básico pasa por suponer que el servidor adoptará un papel pasivo y procederá a esperar conexiones de los posibles clientes. Mientras que los clientes serán los encargados de solicitar conexiones a los servidores de forma activa.

Código de ClienteTCP.java

En primer lugar se determina la dirección IP del host a partir de un string representando su dirección IP.

Creamos el socket, extraemos los flujos de entrada y salida, los escribimos y leemos el resultado final devuelto por el servidor, y lo imprimimos en pantalla.

Finalmente cerramos los flujos y el socket.

```
//ClienteTCP.java

import java.io.*;
import java.net.*;

class ClienteTCP
{
	public static void main(String args[])
	{
		// Leemos el primer parámetro, donde debe ir la dirección
		// IP del servidor
		InetAddress direcc = null;
		try
		{
			direcc = InetAddress.getByName(args[0]);
		}
		catch(UnknownHostException uhe)
		{
			System.err.println("Host no encontrado : " + uhe);
			System.exit(-1);
		}
		// Puerto que hemos usado para el servidor
		int puerto = 1234;
```

```
            // Para cada uno de los argumentos...
            for (int n=1;n<args.length;n++)
            {
                  Socket sckt = null;
                  DataInputStream dis = null;
                  DataOutputStream dos = null;
                  try
                  {
                        // Convertimos el texto en número
                        int numero = Integer.parseInt(args[n]);
                        // Creamos el Socket
                        sckt = new Socket(direcc,puerto);
                        // Extraemos los streams de entrada y salida
                        dis = new
DataInputStream(sckt.getInputStream());
                        dos = new
DataOutputStream(sckt.getOutputStream());
                        // Lo escribimos
                        dos.writeInt(numero);
                        // Leemos el resultado final
                        long resultado = dis.readLong();
                        // Indicamos en pantalla
                        System.out.println( "Solicitud = " + numero +
"\tResultado = " +resultado );
                        // y cerramos los streams y el socket
                        dis.close();
                        dos.close();
                  }
                  catch(Exception e)
                  {
                        System.err.println("Se ha producido la
excepción : " +e);
                  }
                  try
                  {
                        if (sckt!=null) sckt.close();
                  }
                  catch(IOException ioe)
                  {
                        System.err.println("Error al cerrar el socket :
" + ioe);
                  }
            }
      }
}
```

Código de ServidorTCP.java

Primero se obtiene la dirección IP del servidor, abrimos un socket del servidor TCP en el puerto 1234, esperamos a que alguien se conecte a nuestro socket, una vez conectado el cliente extraemos los flujos de entrada y salida a través del puerto remoto y realizamos la operación del cuadrado y enviaremos el resultado al cliente. Luego se escribe ese resultado en la pantalla de servidor y se cierran los flujos.

```
//ServidorTCP.java

import java.io.*;
import java.net.*;

class ServidorTCP
{
      public static void main(String args[])
      {
            // Primero indicamos la dirección IP local
            try
            {
                  System.out.println("LocalHost = " +
InetAddress.getLocalHost().toString());
            }
            catch (UnknownHostException uhe)
            {
                  System.err.println("No puedo saber la dirección IP
local : " + uhe);
            }
            // Abrimos un "Socket de Servidor" TCP en el puerto 1234.
            ServerSocket ss = null;
            try
            {
                  ss = new ServerSocket(1234);
            }
            catch (IOException ioe)
            {
                  System.err.println("Error al abrir el socket de
servidor : " + ioe);
                  System.exit(-1);
            }
            int entrada;
            long salida;
            // Bucle infinito
            while(true)
            {
                  try
                  {
                        // Esperamos a que alguien se conecte a nuestro
Socket
                        Socket sckt = ss.accept();
                        // Extraemos los Streams de entrada y de salida
                        DataInputStream dis = new
                        DataInputStream(sckt.getInputStream());
                        DataOutputStream dos = new
DataOutputStream(sckt.getOutputStream());
                        // Podemos extraer información del socket
                        // N° de puerto remoto
                        int puerto = sckt.getPort();
                        // Dirección de Internet remota
                        InetAddress direcc = sckt.getInetAddress();
                        // Leemos datos de la peticion
                        entrada = dis.readInt();
                        // Calculamos resultado
                        salida = (long)entrada*(long)entrada;
                        // Escribimos el resultado
                        dos.writeLong(salida);
                        // Cerramos los streams
                        dis.close();
                        dos.close();
```

```
                        sckt.close();
                        // Registramos en salida estandard
                        System.out.println( "Cliente = " + direcc + ":"
+ puerto
                        + "\tEntrada = " + entrada + "\tSalida = " +
salida );
                  }
                  catch(Exception e)
                  {
                        System.err.println("Se ha producido la
excepción : " +e);
                  }
            }
      }
}
```

2.- Calcular el Cuadrado –UDP

En esta aplicación el cliente envía un paquete al servidor, para ello debe introducir la dirección IP del servidor y los parámetros sobre los cuales el servidor debe hacer las operaciones. Una vez enviado el paquete al servidor este toma los parámetros y calcula su cuadrado y lo imprime por pantalla. Posteriormente el servidor envía un paquete con los resultados al cliente con la dirección IP y el puerto del cliente obtenidos anteriormente, el cual también los imprime por pantalla.

En este caso se trata de un mecanismo más simple, puesto que el servicio sin conexión tan sólo nos ofrece un mero envío de datos. Puesto que no existe aquí la conexión no hay proceso previo alguno antes de enviar información. Para poder comunicar con otro proceso lo único que hay que hacer es crear el socket y utilizar sus métodos para el envío y recepción de información.

Código de clienteUDP.java

En primer lugar se determina la dirección IP del host a partir de un string representando su dirección IP.

Se construye un socket para datagramas y lo “conecta” al primer puerto disponible.

Creamos un paquete con los datos y lo enviamos al servidor, preparemos un buffer para recibir los datos enviados por el servidor, creamos el contenedor del paquete y lo recibimos, creamos un stream de lectura a partir del buffer, leemos el resultado final y lo imprimimos en pantalla.

```
//clienteUDP.java

import java.net.*;
import java.io.*;

class clienteUDP
{
      public static void main(String args[])
      {
            // Leemos el primer parámetro, donde debe ir la dirección
            // IP del servidor
            InetAddress direcc = null;
            try
            {
                  direcc = InetAddress.getByName(args[0]);
            }
            catch(UnknownHostException uhe)
            {
                  System.err.println("Host no encontrado : " + uhe);
                  System.exit(-1);
            }
            // Puerto que hemos usado para el servidor
            int puerto = 1234;
            // Creamos el Socket
            DatagramSocket ds = null;
            try
            {
                  ds = new DatagramSocket();
            }
            catch(SocketException se)
            {
                  System.err.println("Error al abrir el socket : " +
se);
                  System.exit(-1);
            }
            // Para cada uno de los argumentos...
            for (int n=1;n<args.length;n++)
            {
                  try
                  {
                        //creamos un buffer para escribir
                        ByteArrayOutputStream baos = new
ByteArrayOutputStream();
                        DataOutputStream dos = new
DataOutputStream(baos);
                        // Convertimos el texto en número
                        int numero = Integer.parseInt(args[n]);
                        // Lo escribimos
                        dos.writeInt(numero);
                        // y cerramos el buffer
                        dos.close();
                        // Creamos paquete
                        DatagramPacket dp = new
DatagramPacket(baos.toByteArray(),4,direcc,puerto);
                        // y lo mandamos
                        ds.send(dp);
                        // Preparamos buffer para recibir número de 8
bytes
                        byte bufferEntrada[] = new byte[8];
                        // Creamos el contenedor del paquete
                        dp = new DatagramPacket(bufferEntrada,8);
```

```
                        // y lo recibimos
                        ds.receive(dp);
                        // Creamos un stream de lectura a partir del
buffer
                        ByteArrayInputStream bais = new
ByteArrayInputStream(bufferEntrada);
                        DataInputStream dis = new
DataInputStream(bais);
                        // Leemos el resultado final
                        long resultado = dis.readLong();
                        // Indicamos en pantalla
                        System.out.println( "Solicitud = " + numero +
"\tResultado = " +resultado );
                  }
                  catch (Exception e)
                  {
                        System.err.println("Se ha producido un error :
" + e);
                  }
            }
      }
}
```

Código de servidorUDP.java

Primero se obtiene la dirección IP del servidor, abrimos un socket UDP en el puerto 1234, a través de este socket enviaremos los datagramas, crearemos un contenedor de datagrama, cuyo buffer será un array, esperamos a recibir un paquete, extraemos la información del paquete recibido por parte de algún cliente, obtenemos el puerto y la dirección IP del cliente desde donde se envió el paquete, obtenemos el resultado y generamos un paquete con dicho resultado para enviárselo al cliente que solicitó la operación, para el envío del paquete se utiliza la dirección IP y el puerto obtenidos anteriormente del datagrama enviado por el cliente, posteriormente el servidor imprimirá el resultado por pantalla.

```
//servidorUDP.java

import java.net.*;
import java.io.*;

class servidorUDP
{
      public static void main(String args[])
      {
            // Primero indicamos la dirección IP local
            try
            {
                  System.out.println("LocalHost       =       "       +
InetAddress.getLocalHost().toString());
            }
            catch (UnknownHostException uhe)
            {
                  System.err.println("No  puedo  saber  la  dirección  IP
local :" + uhe);
            }
```

```
            // Abrimos un Socket UDP en el puerto 1234.
            // A través de este Socket enviaremos datagramas del tipo
DatagramPacket
            DatagramSocket ds = null;
            try
            {
                  ds = new DatagramSocket(1234);
            }
            catch(SocketException se)
            {
                  System.err.println("Se ha producido un error al abrir
el socket : " + se);
                  System.exit(-1);
            }
            // Bucle infinito
            while(true)
            {
                  try
                  {
                        // Nos preparamos a recibir un número entero
(32 bits = 4 bytes)
                        byte bufferEntrada[] = new byte[4];
                        // Creamos un "contenedor" de datagrama, cuyo
buffer
                        // será el array creado antes
                        DatagramPacket dp = new
DatagramPacket(bufferEntrada,4);
                        // Esperamos a recibir un paquete
                        ds.receive(dp);
                        // Podemos extraer información del paquete
                        // N° de puerto desde donde se envió
                        int puerto = dp.getPort();
                        // Dirección de Internet desde donde se envió
                        InetAddress direcc = dp.getAddress();
                        //    "Envolvemos"    el    buffer    con    un
ByteArrayInputStream...
                        ByteArrayInputStream bais = new
ByteArrayInputStream(bufferEntrada);
                        //  ...  que  volvemos  a  "envolver"  con  un
DataInputStream
                        DataInputStream dis = new
DataInputStream(bais);
                        // Y leemos un número entero a partir del array
de bytes
                        int entrada = dis.readInt();
                        long salida = (long)entrada*(long)entrada;
                        // Creamos un ByteArrayOutputStream sobre el
que podamos escribir
                        ByteArrayOutputStream baos = new
ByteArrayOutputStream();
                        // Lo envolvemos con un DataOutputStream
                        DataOutputStream dos = new
DataOutputStream(baos);
                        // Escribimos el resultado, que debe ocupar 8
bytes
                        dos.writeLong(salida);
                        // Cerramos el buffer de escritura
                        dos.close();
                        // Generamos el paquete de vuelta, usando los
datos
                        // del remitente del paquete original
```

```
                        dp = new

      DatagramPacket(baos.toByteArray(),8,direcc,puerto);
                        // Enviamos
                        ds.send(dp);
                        // Registramos en salida estandard
                        System.out.println( "Cliente = " + direcc + ":"
+ puerto + "\tEntrada = " +
                        entrada + "\tSalida = " + salida );
                  }
                  catch(Exception e)
                  {
                        System.err.println("Se ha producido el error "
+ e);
                  }
            }
      }
}
```