

SOCKETS TCP EN JAVA

Conceptos

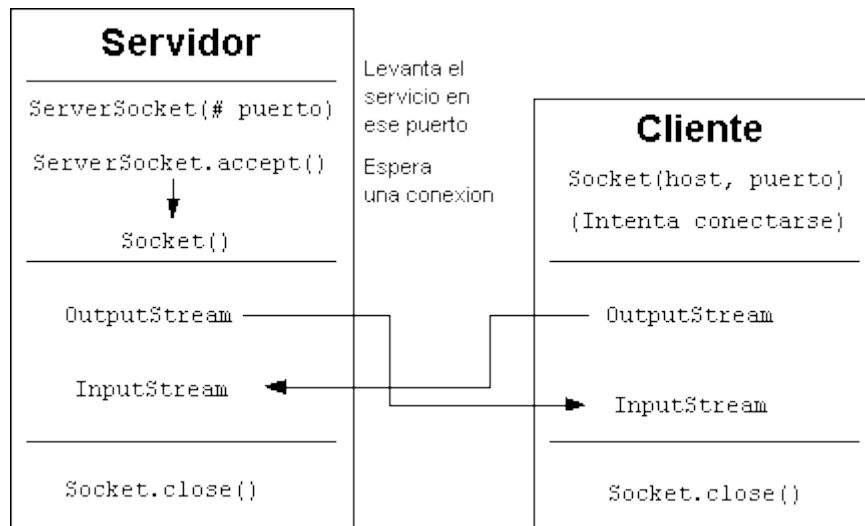
Un socket es cada uno de los extremos de una comunicación entre procesos, por lo que un socket mantiene asociados dos flujos: uno de entrada y otro de salida.

Para que dos sockets entablen una comunicación, uno de ellos (el cliente) debe iniciar enviando una petición al otro socket (el servidor). El cliente requiere 2 cosas: el nombre o la dirección del host donde está residente el proceso servidor y el puerto por donde el servicio está escuchando las peticiones de los clientes.

Los puertos son la abstracción de puntos de acceso (software). En TCP/IP (Transmission Control Protocol/Internet Protocol), los puertos son números de 16 bits y en un rango de 0-65535. En la práctica los puertos del 0 al 1023 están reservados para servicios predefinidos (tales como telnet, SMTP mail, ftp, etc.) y manejados por el administrador del sistema.

Modelo de red en Java

En Java, las clases que nos permiten la comunicación en red se encuentran en el paquete `java.net`. El manejo de flujos asociados a sockets se encuentra en clases del paquete `java.io`.



- El servidor asigna un número de puerto. Cuando el cliente solicita una conexión, el servidor abre un socket a través del método `accept()`.
- El cliente establece una conexión con el host en el puerto especificado.
- Tanto el cliente como el servidor se comunican a través de `InputStream` y de `OutputStream`.

Mínimo servidor TCP

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

public class Servidor
{
    public static void main(String a[])
    {
        ServerSocket serverSocket = null;
        Socket socket = null;
        String peticion = null;
        String respuesta = "Datos incorrectos";

        try
        {
            System.out.println("Escuchando por el puerto 8000");
            serverSocket = new ServerSocket(8000);
        }
        catch(IOException e)
        {
            System.out.println("java.io.IOException generada");
            e.printStackTrace();
        }

        System.out.println("Esperando a que los clientes se conecten...");
        while(true)
        {
            try
            {
                socket = serverSocket.accept();
                System.out.println("Se conecto un cliente: " +
socket.getInetAddress().getHostName());
                DataInputStream dis = new DataInputStream( socket.getInputStream()
);
                peticion = dis.readUTF();
                System.out.println("El mensaje que me envio el cliente es: " +
peticion);
                DataOutputStream dos = new DataOutputStream(
socket.getOutputStream() );
                if(peticion != null)
                    respuesta = "Datos correctos";
                System.out.println("El mensaje que le envio al cliente es: " +
respuesta);
                dos.writeUTF(respuesta);
                dos.close();
                dis.close();
                socket = null;
            }
            catch(IOException e)
            {
                System.out.println("java.io.IOException generada");
                e.printStackTrace();
            }
        }
    }
}
```

Mínimo cliente TCP

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

public class Cliente
{
    public static void main(String a[])
    {
        Socket socket = null;
        String peticion = "Mensaje para el servidor";
        String respuesta = null;

        try
        {
            System.out.println("Me conecto al puerto 8000 del servidor");
            socket = new Socket(a[0],8000);
            System.out.println("Le envio mi peticion: " + peticion);
            DataOutputStream dos = new DataOutputStream(
socket.getOutputStream() );
            dos.writeUTF(peticion);
            DataInputStream dis = new DataInputStream( socket.getInputStream()
);
            respuesta = dis.readUTF();
            System.out.println("El mensaje que me envio el servidor es: " +
respuesta);
            dos.close();
            dis.close();
            socket.close();
        }
        catch(IOException e)
        {
            System.out.println("java.io.IOException generada");
            e.printStackTrace();
        }
    }
}
```