

Comunicación entre Procesos.

Ejemplos

Sistemas Distribuidos
Grado en Ingeniería Informática



Ejemplo 1. Comunicación entre dos procesos mediante Datagramas (UDP)

Fichero DataGramas_Emisor.java

```
package DataGramas_Emisor;

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

public class DataGramas_Emisor {

    public static void main(String[] args) {

        InputStreamReader Entrada=new InputStreamReader(System.in);
        BufferedReader Teclado=new BufferedReader(Entrada);
        String Host = null;
        int Puerto=0;
        String Mensaje=null;
        int Nveces=0;
        try {
            System.out.print("Introduce el nombre del Host: ");
            Host=Teclado.readLine();
            System.out.print("Introduce el nº de Puerto: ");
            Puerto= Integer.parseInt(Teclado.readLine());
            System.out.println("Introduce el mensaje que queires enviar: ");
            Mensaje=Teclado.readLine();
            System.out.println("Introduce el nº de veces a enviarlo: ");
            Nveces= Integer.parseInt(Teclado.readLine());
        } catch (IOException ex) {
            System.out.println("Error en la entrada por datos por teclado....");
        }

        try
        {
            InetAddress Maquina= InetAddress.getByName(Host);
```

```
System.out.println("Maquina host: "+Maquina.getHostName());
System.out.println("Dirección Host: "+Maquina.getHostAddress());
DatagramSocket MiSocket = new DatagramSocket();
byte[] mensajeBytes;
String mensajeDatos;
DatagramPacket datagrama;
for (int i=0;i<=Nveces;i++) {

    if (i<Nveces)
        mensajeDatos=Mensaje+" "+i;
    else
        mensajeDatos=".";
    mensajeBytes=mensajeDatos.getBytes();
    System.out.println("Enviando mensaje: "+mensajeDatos);
    datagrama=new DatagramPacket(mensajeBytes,mensajeBytes.length,Maquina,Puerto);
    MiSocket.send(datagrama);
    //Thread.sleep(200);
}
MiSocket.close();

} catch(Exception ex)
{
    System.out.println("Error en la conexión con datagrama");
}

}
}
```

Fichero DataGramas_Receptor.java

```
package datagramas_receptor;

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

public class DataGramas_Receptor {
```

```
public static void main(String[] args) {  
  
    InputStreamReader Entrada=new InputStreamReader(System.in);  
    BufferedReader Teclado=new BufferedReader(Entrada);  
    int Puerto=0;  
    try {  
        System.out.print("Introduce el nº de Puerto: ");  
        Puerto= Integer.parseInt(Teclado.readLine());  
    } catch (IOException ex) {  
        System.out.println("Error en la entrada por datos por teclado....");  
    }  
    boolean Finalizar=false;  
    try  
    {  
        System.out.println("Esperando paquetes datagrama por el puerto "+Puerto);  
        DatagramSocket MiSocket=new DatagramSocket(Puerto);  
        byte[] Datos=new byte[1024];  
        DatagramPacket datagrama=new DatagramPacket(Datos,1024);  
        while (Finalizar==false)  
        {  
            MiSocket.receive(datagrama);  
            String mensaje=new String(Datos);  
            if (mensaje.getBytes()[0]=='.')  
                Finalizar=true;  
            else  
                System.out.println("Mensaje: "+mensaje);  
        }  
        MiSocket.close();  
    } catch(Exception ex)  
    {  
        System.out.println("Error en la conexión con datagrama");  
    }  
}
```

Ejemplo 2. Comunicación entre dos procesos mediante socket Stream (TCP)

Fichero TCP_Server.java

```
package TCP_Server;

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

public class TCP_Server {

    public static void main(String[] args) {

        InputStreamReader Entrada=new InputStreamReader(System.in);
        BufferedReader Teclado=new BufferedReader(Entrada);

        int PuertoReceptor=0;
        String Mensaje=null;
        try {
            System.out.print("Introduce el nº de Puerto Receptor: ");
            PuertoReceptor= Integer.parseInt(Teclado.readLine());
            System.out.println("Introduce el mensaje que quieres enviar: ");
            Mensaje=Teclado.readLine();
        } catch (IOException ex) {
            System.out.println("Error en la entrada por datos por teclado.");
        }

        try
        {
            ServerSocket socketConexion=new ServerSocket(PuertoReceptor);
            for (int i=0; i<5;i++)
            {
                System.out.println("Preparado para aceptar la conexión "+(i+1)+" de 5");
                Socket socketDatos=socketConexion.accept();
                System.out.println("Conexión acpetada");
                InetAddress Maquina= socketDatos.getLocalAddress();
            }
        }
    }
}
```

```
System.out.println("Maquina local: "+Maquina .getHostName());
System.out.println("Dirección local: "+Maquina.getHostAddress());
Maquina= socketDatos.getInetAddress();
System.out.println("Maquina Cliente: "+Maquina.getHostName());
System.out.println("Dirección Cliente: "+Maquina.getHostAddress());
System.out.println("Puerto Cliente: "+socketDatos.getPort());

OutputStream s_fout=socketDatos.getOutputStream();
InputStream s_fin=socketDatos.getInputStream();

PrintWriter fout=new PrintWriter(s_fout);
BufferedReader fin=new BufferedReader(new InputStreamReader(s_fin));

fout.println(Mensaje);
fout.flush();

boolean Salida=false;
float Numero1, Numero2, Solucion=0;
String Operacion;

while (Salida==false)
{
    Numero1= Float.parseFloat(fin.readLine()); //Numero 1

    Operacion=fin.readLine(); //Operacion

    Numero2= Float.parseFloat(fin.readLine()); //Numero 2

    switch (Operacion)
    {
        case "+": Solucion=Numero1+Numero2; break;
        case "-": Solucion=Numero1-Numero2; break;
        case "*": Solucion=Numero1*Numero2; break;
        case "/": Solucion=Numero1/Numero2; break;
    }

    fout.println(Numero1+" "+ Operacion+" "+ Numero2+ " = "+Solucion);
    fout.flush();
}
```

```
        Operacion= fin.readLine().toLowerCase();
        if ("s".equals(Operacion)==false)
            Salida=true;
    }
    fin.close();
    socketDatos.close();
    System.out.println("Socket de datos cerrado");
}
socketConexion.close();
System.out.println("Socket de conexión cerrado");
} catch (Exception ex)
{
    System.out.println("Inposible crear el servidor tcp");
}
}
```

Fichero TCP_Client.java

```
package tcp_client;

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

public class TCP_Client {

    public static void main(String[] args) {

        InputStreamReader Entrada=new InputStreamReader(System.in);
        BufferedReader Teclado=new BufferedReader(Entrada);

        int PuertoServidor=0;
        String Host=null;
```

```
try {
    System.out.print("Introduce el el nombre del Host: ");
    Host=Teclado.readLine();
    System.out.print("Introduce el nº de Puerto Receptor: ");
    PuertoServidor= Integer.parseInt(Teclado.readLine());
} catch (IOException ex) {
    System.out.println("Error en la entrada por datos por teclado...");
}

try
{
    InetAddress Maquina=InetAddress.getByName(Host);
    Socket socketDatos=new Socket(Maquina, PuertoServidor);

    System.out.println("Socket de conexión aceptado");
    InetAddress MaquinaLocal= socketDatos.getLocalAddress();

    System.out.println("Maquina local: "+MaquinaLocal.getHostName());
    System.out.println("Dirección local: "+MaquinaLocal.getHostAddress());
    System.out.println("Maquina Cliente: "+Maquina.getHostName());
    System.out.println("Dirección Cliente: "+Maquina.getHostAddress());
    System.out.println("Puerto Cliente: "+socketDatos.getLocalPort());

    OutputStream s_fout=socketDatos.getOutputStream();
    InputStream s_fin=socketDatos.getInputStream();

    PrintWriter fout=new PrintWriter(s_fout);
    BufferedReader fin=new BufferedReader(new InputStreamReader(s_fin));
    System.out.println("Preparado para Empezar a operar");

    boolean Salida=false;
    String Mensaje=fin.readLine();
    System.out.println(Mensaje); //Mensaje bienvenida

    while (Salida==false)
    {
        System.out.println("Introduce operacion <numero1> <operador> <numero2>");
```



```
System.out.print("Introduce <numero1>: ");
fout.println(Teclado.readLine());
fout.flush();

System.out.print("Introduce <operador>: ");
fout.println(Teclado.readLine());
fout.flush();

System.out.print("Introduce <numero2>: ");
fout.println(Teclado.readLine());
fout.flush();

Mensaje=fin.readLine();
System.out.println(Mensaje); //Resultado

System.out.print("Deseas una operación más (s/n)? ");
Mensaje=Teclado.readLine().toLowerCase(); //Para que server termine
fout.println(Mensaje);
fout.flush();

if (Mensaje.equals("s")==false)
    Salida=true;
System.out.println(); //Linea en blanco
}
System.out.println("Termino de Operar");

fin.close();
socketDatos.close();
System.out.println("Socket de datos cerrado");

} catch (Exception ex)
{
    System.out.println("Inposible crear el cliente tcp: "+ex.getMessage());
}
}
```

Ejemplo 3. Comunicación entre dos procesos mediante socket datagrama (UDP)

Fichero UDP_Sender.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;

namespace UDP_Sender
{
    class UDP_Sender
    {
        static void Main(string[] args)
        {
            int PuertoReceptor = 0, Nveces=0;
            String Mensaje,Host;

            Console.WriteLine("Introduce el nombre del Host: ");
            Host=Console.ReadLine();
            Console.WriteLine("Introduce el nº de Puerto Receptor: ");
            int.TryParse(Console.ReadLine(), out PuertoReceptor);
            Console.WriteLine("Introduce el mensaje que quieres enviar: ");
            Mensaje = Console.ReadLine();
            Console.WriteLine("Introduce el nº de veces a enviarlor: ");
            int.TryParse(Console.ReadLine(), out Nveces);

            IPEndPoint remoteEndPoint = new IPEndPoint(IPAddress.Parse(Host),PuertoReceptor);

            UdpClient client = new UdpClient();
            try
            {
                string mensajeDatos;
                for (int i = 0; i <= Nveces; i++)
```

```
{
    if (i<Nveces)
        mensajeDatos=Mensaje+" "+i;
    else
        mensajeDatos=".";
    byte[] Data = Encoding.UTF8.GetBytes(mensajeDatos);
    Console.WriteLine("Enviando Mensaje: " + mensajeDatos);
    client.Send(Data, Data.Length, remoteEndPoint);
}
}
catch (Exception err)
{
    Console.WriteLine(err.ToString());
}
finally
{
    client.Close();
}
}
}
```

Fichero UDP_Receiver.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;
```

```
namespace UDP_Receiver
{
    class UDP_Receiver
    {
```

```
static void Main(string[] args)
{
    int PuertoReceptor = 0;
    String Mensaje;

    Console.WriteLine("Introduce el nº de Puerto Receptor: ");
    int.TryParse(Console.ReadLine(), out PuertoReceptor);

    UdpClient client = new UdpClient(PuertoReceptor);
    Boolean Finalizar=false;
    try
    {
        IPEndPoint anyIP = new IPEndPoint(IPAddress.Any, 0);
        Console.WriteLine("Esperando Paquetes datagrama por el puerto " + PuertoReceptor);
        do
        {
            byte[] Data = client.Receive(ref anyIP);
            Mensaje = Encoding.UTF8.GetString(Data);
            if (Mensaje.ToLower() == ".")
                Finalizar = true;
            else
                Console.WriteLine("Mensaje: " + Mensaje);
        } while (Finalizar == false);
        Console.WriteLine(Mensaje);
    }
    catch (Exception err)
    {
        Console.WriteLine(err.ToString());
    }
    finally
    {
        client.Close();
    }
}
```

Ejemplo 4. Comunicación entre dos procesos mediante socket Stream (TCP)

Fichero TCP_Server.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.IO;

namespace TCP_Server
{
    class Program
    {
        static void Main(string[] args)
        {
            int PuertoReceptor = 0;
            String Mensaje;

            Console.Write("el nº de Puerto Receptor: ");
            int.TryParse(Console.ReadLine(), out PuertoReceptor);
            Console.Write("Introduce el mensaje que quieres enviar: ");
            Mensaje=Console.ReadLine();

            IPAddress ipAddress = Dns.Resolve(Dns.GetHostName()).AddressList[0];
            IPEndPoint ipLocalEndPoint = new IPEndPoint(ipAddress, PuertoReceptor);
            try {
                //TcpListener listener = new TcpListener(PuertoReceptor);
                TcpListener listener = new TcpListener(ipLocalEndPoint);
                listener.Start();

                for (int i = 0; i < 5; i++) {
                    Console.WriteLine("Preparado para aceptar la conexión {0} de 5", i + 1);

                    Socket socketDatos = listener.AcceptSocket();
                    Console.WriteLine("Conexión aceptada con {0}", socketDatos.RemoteEndPoint);
                }
            }
        }
    }
}
```

```
Stream fs = new NetworkStream(socketDatos);
StreamReader fin = new StreamReader(fs);
StreamWriter fout = new StreamWriter(fs);
fout.AutoFlush = true;

fout.WriteLine(Mensaje);
bool Salida = false;
float numero1, numero2, solucion = 0;
string operacion;
while (Salida == false) {
    fout.WriteLine("Introduce <numero1>: ");
    float.TryParse(fin.ReadLine(), out numero1);
    fout.WriteLine("Introduce <operacion>: ");
    operacion = fin.ReadLine();
    fout.WriteLine("Introduce <numero2>: ");
    float.TryParse(fin.ReadLine(), out numero2);
    switch (operacion) {
        case "+": solucion = numero1 + numero2; break;
        case "-": solucion = numero1 - numero2; break;
        case "*": solucion = numero1 * numero2; break;
        case "/": solucion = numero1 / numero2; break;
    };
    fout.WriteLine("{0} {1} {2} = {3}", numero1, operacion, numero2, solucion);
    fout.WriteLine("Deseas una operación más (s/n)? ");
    Mensaje = fin.ReadLine().ToLower();
    if (Mensaje != "s")
        Salida = true;
}
fs.Close();
}
}
}
}
}
```

Fichero TCP_Client.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.IO;

namespace TCP_Client
{
    class Program
    {
        static void Main(string[] args)
        {
            try {
                int PuertoReceptor = 0;
                String Mensaje, Host;

                Console.Write("Introduce el nº de Puerto Receptor: ");
                int.TryParse(Console.ReadLine(), out PuertoReceptor);
                Console.Write("Introduce el Host al que quieres conectarte: ");
                Host = Console.ReadLine();

                IPAddress ipAddress = Dns.Resolve(Host).AddressList[0];
                IPEndPoint ipRemoteEndPoint = new IPEndPoint(ipAddress, PuertoReceptor);

                TcpClient client = new TcpClient();
                client.Connect(ipRemoteEndPoint);

                Stream fs = client.GetStream();
                StreamReader fin = new StreamReader(fs);
                StreamWriter fout = new StreamWriter(fs);

                fout.AutoFlush = true;
                Mensaje = fin.ReadLine();
                Console.WriteLine(Mensaje); //Bienvenida
            }
        }
    }
}
```

```
do {
    Mensaje = fin.ReadLine();
    Console.Write(Mensaje); //numero 1
    fout.WriteLine(Console.ReadLine());

    Mensaje = fin.ReadLine();
    Console.Write(Mensaje); //operacion
    fout.WriteLine(Console.ReadLine());

    Mensaje = fin.ReadLine();
    Console.Write(Mensaje); //numero 2
    fout.WriteLine(Console.ReadLine());

    Mensaje = fin.ReadLine();
    Console.WriteLine(Mensaje); //Resultado

    Mensaje = fin.ReadLine(); //pregunta
    Console.Write(Mensaje); //
    Mensaje = Console.ReadLine().ToLower();
    fout.WriteLine(Mensaje);
} while (Mensaje == "s");
fs.Close();
}
catch (Exception ex)
{
    Console.WriteLine("Error en la conexión {0}", ex.Message);
}
}
}
```