## línea horizontal



Práctica 2: ENVÍO DE OBJETOS ENTRE ESTACIONES REMOTAS

10/10/2022

**─**

Lucía Montañana Fuentes

Venus Escudero Fajardo

# CÓDIGO SERVIDOR

import java.io.IOException;

import java.io.ObjectInputStream;

import java.net.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

public static void main(String[] args) {

int serverPort = 6789; ***//El puerto de servicio.***

ServerSocket listenSocket; ***//El socket de escucha de peticiones.***

Socket clientSocket; ***//Socket para la comunicacion.***

ObjectInputStream in;

Ejemplo e;

try{

listenSocket = new ServerSocket(serverPort);

***//Comportamiento normal de un servidor,bucle atendiendo***

***//las peticiones los clientes:***

System.out.println("Servidor en marcha");

clientSocket=listenSocket.accept(); ***//Espera una peticion***

***//y obtiene un socket de comunicación.***

in = new ObjectInputStream(clientSocket.getInputStream());

e = (Ejemplo)in.readObject();

e.Recorrido();

System.out.println("Cierro conexion");

listenSocket.close();

}catch(IOException ex){

System.out.println("Error en socket:"+ ex.getMessage());

} catch (ClassNotFoundException ex) {

Logger.getLogger(ServidorTCP.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

# CÓDIGO CLIENTE

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author usuario

\*/

public class ClienteTCP {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

Socket s = null;

int serverPort = 6789; ***//Puerto del servidor.***

ObjectOutputStream out; ***//Para Las Escrituras Sobre El Stream***

String data;***//Para obtener la respuesta del servidor***

***//Variables que creamos nosotros***

Ejemplo e, e1, e2;

try {

***//Crea Socket local y se conecta al servidor:***

s = new Socket("localhost", serverPort);***//Al conectarse crea el socket de servicio en el servidor.***

***//Petición De Servicio:***

out = new ObjectOutputStream(s.getOutputStream()); ***//Envía El Mensaje:***

e = new Ejemplo();

e1 = new Ejemplo();

e2 = new Ejemplo();

e.x = 50;

e1.x = 100;

e2.x = 200;

//e2

e2.p1 = e1;

e2.p2 = null;

//e1

e1.p1 = e2;

e1.p2 = new Ejemplo[2];

e1.p2[0] = e1;

e1.p2[1] = e2;

//e

e.p1 = e1;

e.p2 = null;

out.writeObject(e);

} catch (IOException ex) {

Logger.getLogger(ClienteTCP.class.getName()).log(Level.SEVERE, "error inicializacion", ex);

}

}

}

# CÓDIGO EJEMPLO

public class Ejemplo implements Serializable {

int x;

Ejemplo p1 = null ;

Ejemplo[] p2 = null;

Boolean visitado=false;

Ejemplo() { **//Constructor en blanco**

}

public void Recorrido() {

System.out.println("Entro en recorrido");

if (!visitado) {

System.out.println("El objeto no está visitado aun");

visitado = true;

**//Mostramos la variable x**

System.out.println("La variable x = " + x);

**//Mostramos referencia p1, comprobamos si el objeto es nulo y llamamos recursivamente a la función**

if (p1 != null) {

p1.Recorrido();

}

**//Mostramos referencia p2, comprobamos si el objeto es nulo**

if(p2!=null){

for (Ejemplo ejemplo : p2) {

if (ejemplo != null) {

ejemplo.Recorrido();

}

}

}

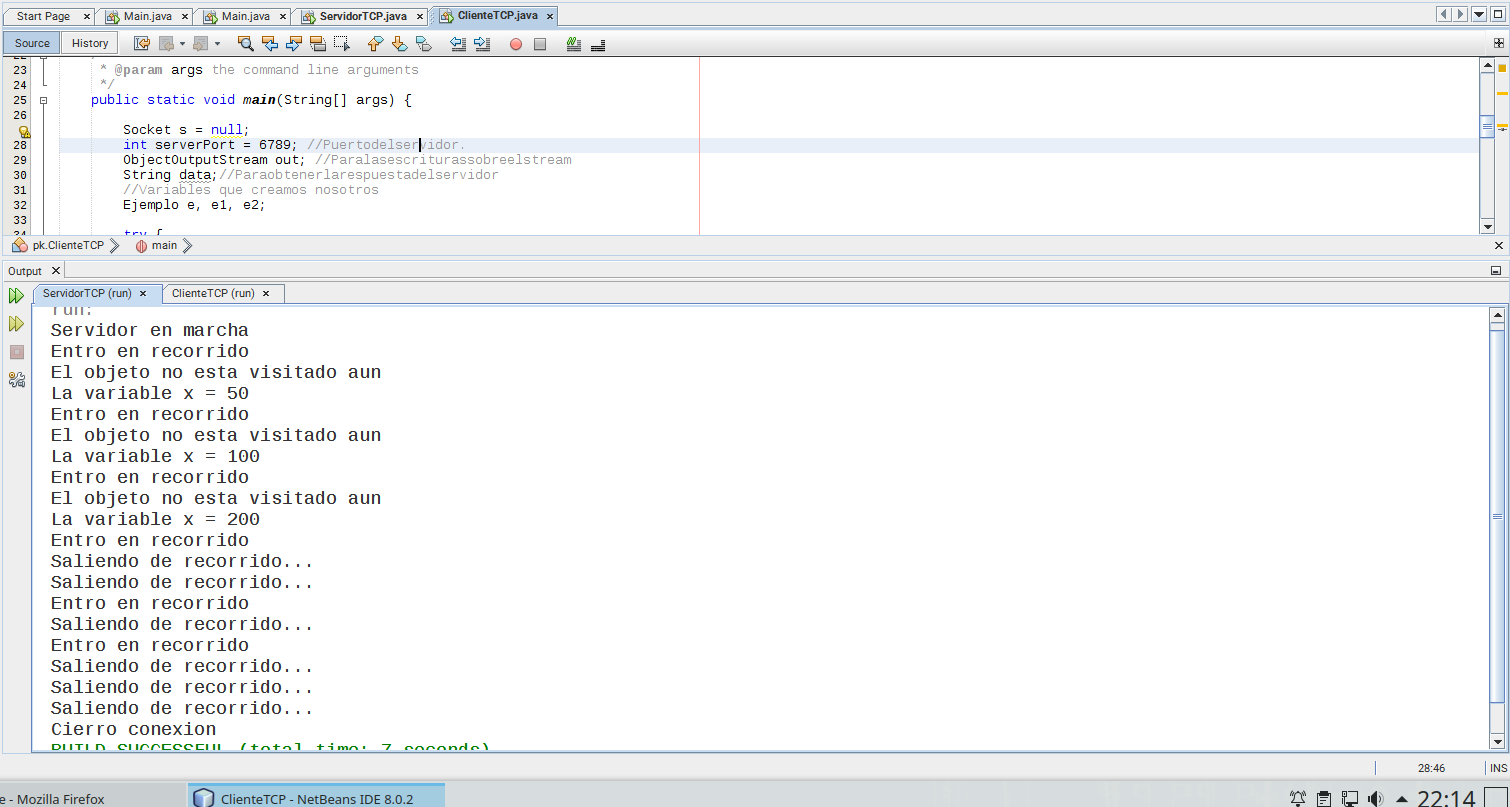
}

System.out.println("Saliendo de recorrido...");

}

}

# PRUEBA LOCAL



# PRUEBA REMOTO

Cambiamos la línea 36

s = new Socket("192.168.68.106",serverPort);

“localhost” por la ip del terminal donde tenemos ejecutando el servidor

