

Tarea Individual 22 - Transferencia de objetos en SOCKETS UDP y TCP

Salidas TCP:

Inicia servidor y se queda a la espera de un cliente.

```
TCPServidor [Java Application] C:\Program Files\Java\jdk-21\bin
Inicio del programa servidor.....
Esperando primera conexion en puerto: 6000
```

Cuando se conecta, envia el objeto y lo recibe editado

```
Inicio del programa servidor.....
Esperando primera conexion en puerto: 6000
Enviado objeto: Jorge*18
Recibido objeto: Jorge con apellido*22
Fin del programa Servidor .....
```

Salidas UDP:

Ejecuto UDPServidor y se queda a la espera del datagrama con el objeto:

```
s> java -cp bin Tema3.objetos.UDPServidor
Esperando datagrama.....
```

Ejecuto el cliente:

```
s> java -cp bin Tema3.objetos.UDPCliente
PS C:\Users\ [redacted] \OneDrive\Escritorio\Prog
s>
```

UDPServidor recibe el objeto y lo imprime correctamente:

```
s> java -cp bin Tema3.objetos.UDPServidor
Esperando datagrama.....
Recibido objeto: Jorge*22
```

Codigos:

Persona.java

```
package Tema3.objetos;
import java.io.Serializable;
public class Persona implements Serializable {

    private static final long serialVersionUID = 1L;
    private String nombre;
    private int edad;

    public Persona() {
        super();
    }

    public Persona(String nombre, int edad) {
        super();
        this.nombre = nombre;
        this.edad = edad;
    }

    public String getNombre() {
        return this.nombre;
    }

    public int getEdad() {
        return this.edad;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

}
```

UDPServidor.java

```
package Tema3.objetos;
import java.io.ByteArrayInputStream;
import java.io.ObjectInputStream;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class UDPServidor {

    private static final int PUERTO = 12345;

    public static void main(String[] args) throws Exception{

        //buffer para recibir el datagrama
        byte[] buffer = new byte[1024];
        DatagramSocket socket = new DatagramSocket(PUERTO);

        //Esperando datagrama
        System.out.println("Esperando datagrama.....");
        DatagramPacket paquete = new DatagramPacket(buffer, buffer.length);
        socket.receive(paquete); //Recibo datagrama

        //Convertimos el buffer en un stream ByteArrayInputStream
        ByteArrayInputStream flujoBytes = new ByteArrayInputStream(buffer);
        //Extraemos el stream de objetos del ByteArrayInputStream
        ObjectInputStream flujoObjetos = new ObjectInputStream(flujoBytes);

        //Convertimos el datagrama a un objeto Persona
        Persona persona = (Persona) flujoObjetos.readObject();
        System.out.println("Recibido objeto: " + persona.getNombre() + "*" +
persona.getEdad());

        //Cerramos streams y sockets
        socket.close();
        flujoObjetos.close();
        flujoBytes.close();

    }
}
```

UDPCliente.java

```
package Tema3.objetos;

import java.io.ByteArrayOutputStream;
import java.io.ObjectOutputStream;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class UDPCliente {

    public static void main(String[] args) throws Exception{
```

```

        InetAddress destino = null;
        int puerto = 12345;
        DatagramSocket socket = new DatagramSocket();
        destino = InetAddress.getLocalHost();

        //Necesitamos un stream de array para obtener un buffer para el paquete
        ByteArrayOutputStream flujoSalida = new ByteArrayOutputStream();
        //Creamos un stream de objetos DENTRO del stream de array de bytes
        ObjectOutputStream flujoObjetos = new ObjectOutputStream(flujoSalida);

        //Creamos un objeto Persona
        Persona persona = new Persona("Jorge", 22);
        //Metemos el objeto Persona en el stream de objetos DENTRO del stream
de array de bytes
        flujoObjetos.writeObject(persona);

        //Obtenemos un buffer para enviar el datagrama del stream de array de
bytes
        byte[] buffer = flujoSalida.toByteArray();

        //Enviamos datagrama al servidor
        DatagramPacket paqueteEnviado = new DatagramPacket(buffer,
buffer.length, destino, puerto);
        socket.send(paqueteEnviado);

        //Cerramos streams y sockets
        socket.close();
        flujoObjetos.close();
        flujoSalida.close();

    }
}

```