

## PRACTICA

Nombre: Carmela Sandra Calliconde Carrasco

Lic. Gallardo

Paralelo: Martes

### Ejercicio 1 Piedra Papel o Tijera

#### TCP

##### Cliente

```
2 import java.io.*;
3 import java.net.*;
4
5 public class ClienteTCP {
6     public static void main(String[] args) throws IOException {
7         Socket socketCliente = null;
8         BufferedReader entrada = null;
9         PrintWriter salida = null;
10        System.err.println("Carmela Sandra Calliconde Carrasco \n");
11        System.err.println("juego iniciado para terminar 'Salir' \n");
12        // creamos un socket en cliente que se enlazara con un servidor y en el mismo puerto
    }
}
```

Console

ClienteTCP [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (8 jun. 2020 22:10:45)

Carmela Sandra Calliconde Carrasco

juego iniciado para terminar 'Salir'

Elige [1=Piedra, 2=Papel, 3=Tijera]:  
2  
Respuesta del servidor: Es empate!...Cpu eligio 'Papel'

Elige [1=Piedra, 2=Papel, 3=Tijera]:  
3  
Respuesta del servidor: Usted gana!...Cpu eligio 'Papel'

Elige [1=Piedra, 2=Papel, 3=Tijera]:  
1  
Respuesta del servidor: La computadora gana!...Cpu eligio 'Papel'

Elige [1=Piedra, 2=Papel, 3=Tijera]:  
Salir

##### Servidor

ejercicio1TCP/ClienteTCP.java - Eclipse IDE

Search Project Run Window Help

ClienteTCP.java

```
30
31 try {
32     boolean sw = true;
33     while (sw) {
34         System.out.println("Elige [1=Piedra, 2=Papel, 3=Tijera]:");
35         linea = stdIn.readLine();
36         salida.println(linea);
37
38         if (linea=="Salir") sw = false ;
39         else {
40             linea = entrada.readLine();
41         }
    }
}
```

Console

ServidorTCP [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (8 jun. 2020 22:10:39)

Escuchando: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=4444]

--

Connexion aceptada: Socket[addr=/127.0.0.1,port=52402,localport=4444]

--

52402

\*\*\*

Papel

Empate!

Cliente: 2

Papel

Usted gana!

Cliente: 3

Papel

La computadora gana!

Cliente: 1

##### Servidor

```
package ejercicio1TCP;
import java.net.*;
```

```

import java.io.*;
import java.nio.channels.SocketChannel;

public class ServidorTCP {

    //creando el puerto donde escucharemos
    public static final int PORT = 4444;
    //metodo para seleccionar generar piedra papel o tijera
    public static String PiedraPapelTijera(String seleccionUsuario)
    {

        int seleccionCompu = (int)(Math.random()*3+1); // genera
        un número al azar
        // entre 1 y 3 La computadora escoge
        String res = "";
        switch ( seleccionCompu )
        {
            case 1:
                System.out.println("Piedra");
                switch ( seleccionUsuario )
                {
                    case "1": System.out.println("Empate!"); res = "Es
empate!"+"...Cpu eligio 'Piedra'"; break;
                    case "2": System.out.println("Usted gana!"); res
="Usted gana!"+"...Cpu eligio 'Piedra'"; break;
                    case "3": System.out.println("La computadora
gana!"); res = "La computadora gana!"+"...Cpu eligio 'Piedra'"; break;
                }
                break;

            case 2:
                System.out.println("Papel");
                switch ( seleccionUsuario )
                {
                    case "1": System.out.println("La computadora
gana!"); res = "La computadora gana!"+"...Cpu eligio 'Papel'"; break;
                    case "2": System.out.println("Empate!"); res = "Es
empate!"+"...Cpu eligio 'Papel'"; break;
                    case "3": System.out.println("Usted gana!"); res
="Usted gana!"+"...Cpu eligio 'Papel'"; break;
                }
                break;

            case 3:
                System.out.println("Tijera");
                switch ( seleccionUsuario )
                {
                    case "1": System.out.println("Usted gana!"); res
="Usted gana!"+"...Cpu eligio 'Tijera'"; break;
                    case "2": System.out.println("La computadora
gana!"); res = "La computadora gana!"+"...Cpu eligio 'Tijera'"; break;
                    case "3": System.out.println("Empate!"); res = "Es
empate!"+"...Cpu eligio 'Tijera'"; break;
                }
                break;
        }
        return res;
    }
}

```

```

    }

    public static void main(String[] args) throws IOException {
        // establece el puerto en el que escucha peticiones
        ServerSocket socketServidor = null;
        try {
            //creando el servidor con el puerto en que escuchara
            socketServidor = new ServerSocket(PORT);
        } catch (IOException e) {

            System.out.println("No se puede acceder al puerto: " +
PORT);

            System.exit(-1);
        }
        //creando el socket cliente
        Socket socketCliente = null;

        BufferedReader entrada = null;
        // creando donde mostraremos
        PrintWriter salida = null;
        ///mensaje muestra en q puerto y direccion escucha el servidor
        System.out.println("Escuchando: " + socketServidor+"\n--");
        try {

            while(true) {

                socketCliente = socketServidor.accept();
                //mensaje de conexion aceptada del cliente
                System.out.println("Connexión aceptada: "+
socketCliente+"\n--");

                //muestra el puerto en el que esta el cliente
                System.out.println(socketCliente.getPort()+"\n***");

                // establece un canal de entrada
                entrada = new BufferedReader(new
InputStreamReader(socketCliente.getInputStream()));
                // establece un canal de salida
                salida = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socketCliente.getOutputStream()), true));

                boolean sw = true;
                // hace eco de lo que le proporciona el cliente,
hasta que recibe "Salir"
                while (sw) {
                    String palabra = entrada.readLine();
                    if (palabra.equals("Salir")) sw = false;
                    else {
                        //enviamos la respuesta al metodo
                        String cad = PiedraPapelTijera(palabra);

                        //enviando respuesta al cliente
                        System.out.println("Cliente: " +palabra);
                        salida.println(cad);
                    }
                }
            }
        }
    }
}

```

```

    } catch (IOException e) {
        System.out.println("IOException: " + e.getMessage());
    }
    //liberando recursos
    salida.close();
    entrada.close();
    socketCliente.close();
    socketServidor.close();
}

}

```

## CLIENTE

```

package ejercicio1TCP;
import java.io.*;
import java.net.*;

public class ClienteTCP {
    public static void main(String[] args) throws IOException {
        Socket socketCliente = null;
        BufferedReader entrada = null;
        PrintWriter salida = null;
        System.err.println("Carmela Sandra Calliconde Carrasco \n");
        System.err.println("juego iniciado para terminar 'Salir' \n");
        // crearemos un socket en cliente que se enlazara con un servidor y
        en el mismo puerto

        try {
            socketCliente = new Socket("localhost", 4444);
            // Obtenemos el canal de entrada

            entrada = new BufferedReader(new
InputStreamReader(socketCliente.getInputStream()));

            // Obtenemos el canal de salida
            salida = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socketCliente.getOutputStream()), true));
        } catch (IOException e) {

            System.err.println("No puede establecer canales de E/S para la
conexión");
            System.exit(-1);
        }
        //leendo el buffer de entrada
        BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in));
        String linea;

        try {
            boolean sw = true;
            while (sw) {
                System.out.println("Elige [1=Piedra, 2=Papel, 3=Tijera]: ");
                linea = stdIn.readLine();

                salida.println(linea);
            }
        }
    }
}

```

```

        if (linea=="Salir") sw = false ;
        else {
            linea = entrada.readLine();
            System.out.println("Respuesta del servidor: " + linea);
            // Si es "exit" es que finaliza la comunicación
        }
    }
} catch (IOException e) {
    System.out.println("IOException: " + e.getMessage());
}

//liberando recursos
salida.close();
entrada.close();
stdIn.close();
socketCliente.close();
}
}

```

## Ejercicio 2

### Contador de palabras UDP

#### Servidor

```

1 package ejercicio2UDP;
2 import java.io.*;
3 import java.net.*;
4 import java.util.StringTokenizer;
5 public class servidorUDP {
6     public static String recuperar(String cad,int tam){
7         String respuesta ="";
8         for(int i=0;i<tam;i++) {
9             respuesta +=cad.charAt(i);
10        }
11        return respuesta;
12    }
13    public static void main(String[] args)throws IOException {
14        System.out.println(" contador de palabras-");
15        try {
16            DatagramSocket socketUDP= new DatagramSocket(5555);
17            byte [] mensaje=new byte[20000];
18            while(true) {
19                DatagramPacket peticion=new DatagramPacket(mensaje,mensaje.length);
20                socketUDP.receive(peticion);
21                String res= new String(peticion.getData());
22                String x=recuperar(res,peticion.getLength());
23
24                StringTokenizer stringTokenizer = new StringTokenizer(x, " ");
25                String env ="numero de palabras ->"+stringTokenizer.countTokens();
26
27                byte [] enviar= env.getBytes();
28                DatagramPacket mensaje1= new DatagramPacket(enviar,enviar.length(),peticion.getAddress(),peticion.getPort());
29                socketUDP.send(mensaje1);
30
31                System.out.print("Datos:"+new String(peticion.getData()));
32            }
33        } catch (SocketException e) {
34            // TODO Auto-generated catch block
35            //e.printStackTrace();
36        }
37    }
38 }
39

```

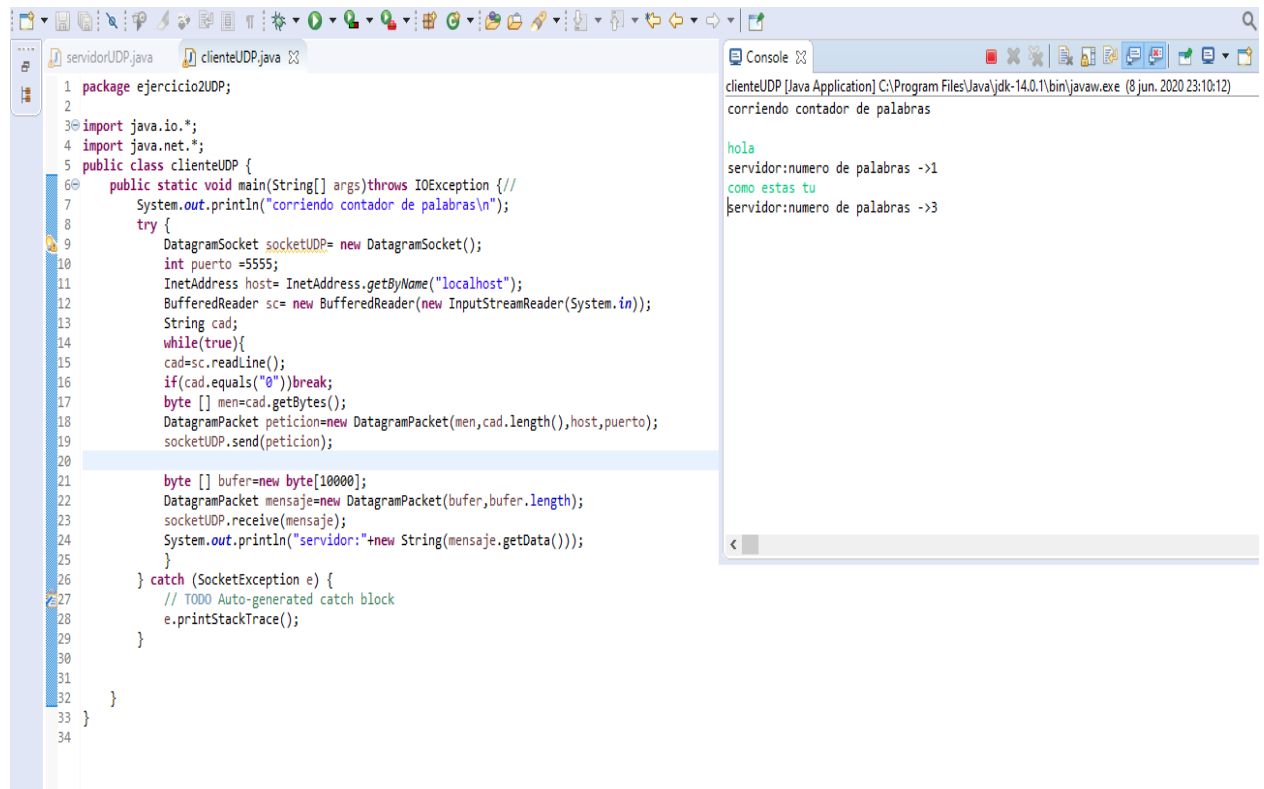
Console

```

servidorUDP [Java Application] C:\Program Files\Java\jdk-
contador de palabras-
Datos:hola

```

## Cliente



The screenshot shows an IDE with two tabs: `servidorUDP.java` and `clienteUDP.java`. The `clienteUDP.java` tab is active, displaying the following code:

```
1 package ejercicio2UDP;
2
3 import java.io.*;
4 import java.net.*;
5 public class clienteUDP {
6     public static void main(String[] args) throws IOException {
7         System.out.println("corriendo contador de palabras\n");
8         try {
9             DatagramSocket socketUDP= new DatagramSocket();
10            int puerto =5555;
11            InetAddress host= InetAddress.getByNme("localhost");
12            BufferedReader sc= new BufferedReader(new InputStreamReader(System.in));
13            String cad;
14            while(true){
15                cad=sc.readLine();
16                if(cad.equals("0"))break;
17                byte [] men=cad.getBytes();
18                DatagramPacket peticion=new DatagramPacket(men,cad.length(),host,puerto);
19                socketUDP.send(peticion);
20
21                byte [] bufer=new byte[10000];
22                DatagramPacket mensaje=new DatagramPacket(bufer,bufer.length);
23                socketUDP.receive(mensaje);
24                System.out.println("servidor:"+new String(mensaje.getData()));
25            }
26        } catch (SocketException e) {
27            // TODO Auto-generated catch block
28            e.printStackTrace();
29        }
30    }
31 }
32
33 }
34 }
```

The console window on the right shows the output of the program:

```
clienteUDP [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (8 jun. 2020 23:10:12)
corriendo contador de palabras

hola
servidor:numero de palabras ->1
como estas tu
servidor:numero de palabras ->3
```