



Instituto Politécnico Nacional



Escuela Superior de Computo

# Aplicaciones para comunicaciones en red

Profesor: Martinez Rosales Ricardo

*Tarea 1. ECO TCP*

Alumna:

López Hernández Gloria Akari

Grupo:

6CM3

Periodo Escolar: 24/2

## Tarea 1. Eco TCP

### Código Servidor TCP

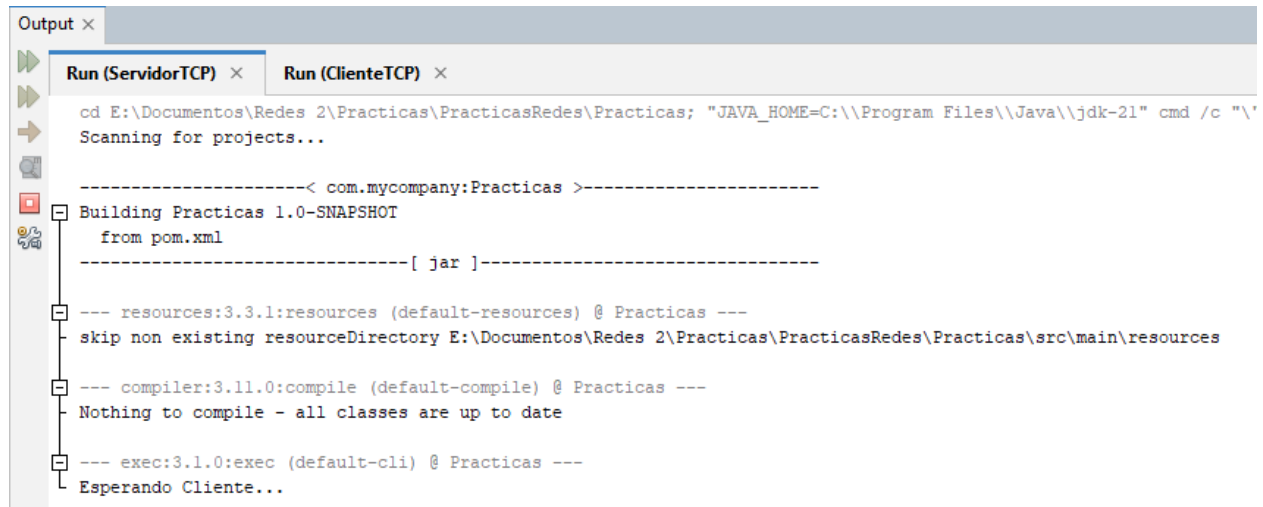
```
ClienteTCP.java x ServidorTCP.java x
Source History
1 package com.mycompany.practicas;
2
3 import java.net.*;
4 import java.io.*;
5
6 public class ServidorTCP {
7     public static void main(String[] args) {
8         try {
9             ServerSocket s = new ServerSocket(1234); //Creamos un socket de servidor que escuche en el puerto
10            System.out.println("Esperando Cliente..."); //Esperamos a que un cliente se conecte
11            BufferedReader brRespuesta = new BufferedReader(new InputStreamReader(System.in)); //Este BufferedReader
12            while (true) {
13                Socket cl = s.accept(); //Aceptamos la conexion de un cliente
14                System.out.println("Conexión establecida desde " + cl.getInetAddress() + ":" + cl.getPort());
15
16                BufferedReader br = new BufferedReader(new InputStreamReader(cl.getInputStream())); // BufferedReader
17                PrintWriter pw = new PrintWriter(new OutputStreamWriter(cl.getOutputStream()), true); // PrintWriter
18                String mensaje;
19                do { //Este bucle se ejecutará al menos una vez y seguirá ejecutándose mientras el mensaje del cliente
20                    mensaje = br.readLine(); // Leer un mensaje desde el cliente
21                    System.out.println("Cliente:\n" + mensaje); //Imprimira en consola el mensaje que envia el cliente
22
23                    if (!mensaje.equalsIgnoreCase("salir")) { //Mientras mensaje sea diferente a salir
24                        System.out.println("Servidor:");
25                        String respuesta = brRespuesta.readLine(); // Leer una respuesta desde la consola del cliente
26                        pw.println(respuesta);
27
28                        if (respuesta.equalsIgnoreCase("salir")) { // Si la respuesta es "salir", cerrar la conexion
29                            br.close();
30                            pw.close();
31                            cl.close();
32                            break;
33                        }
34                    }
35                } while (!mensaje.equalsIgnoreCase("salir")); //El ciclo se cierra cuando mensaje es salir
36                System.out.println("Esperando Cliente..."); //Una vez cerrado el socket del cliente nuestro socket
37            }
38        } catch (Exception e) {
39            e.printStackTrace();
40        }
41    }
42 }
```

## Código cliente TCP

```
1 package com.mycompany.practicass;
2
3 import java.net.*;
4 import java.io.*;
5
6 public class ServidorTCP {
7     public static void main(String[] args) {
8         try {
9             ServerSocket s = new ServerSocket(1234); //Creamos un socket de servidor que escuche en el puerto
10            System.out.println("Esperando Cliente..."); //Esperamos a que un cliente se conecte
11            BufferedReader brRespuesta = new BufferedReader(new InputStreamReader(System.in)); //Este Buffered
12            while (true) {
13                Socket cl = s.accept(); //Aceptamos la conexion de un cliente
14                System.out.println("Conexión establecida desde " + cl.getInetAddress() + ":" + cl.getPort());
15
16                BufferedReader br = new BufferedReader(new InputStreamReader(cl.getInputStream())); // Buffere
17                PrintWriter pw = new PrintWriter(new OutputStreamWriter(cl.getOutputStream()), true); // Print
18                String mensaje;
19                do { //Este bucle se ejecutará al menos una vez y seguirá ejecutándose mientras el mensaje del
20                    mensaje = br.readLine(); // Leer un mensaje desde el cliente
21                    System.out.println("Cliente:\n" + mensaje); //Imprimira en consola el mensaje que envio el
22
23                    if (!mensaje.equalsIgnoreCase("salir")) { //Mientras mensaje sea diferente a salir
24                        System.out.println("Servidor:");
25                        String respuesta = brRespuesta.readLine(); // Leer una respuesta desde la consola del
26                        pw.println(respuesta);
27
28                        if (respuesta.equalsIgnoreCase("salir")) { // Si la respuesta es "salir", cerrar la co
29                            br.close();
30                            pw.close();
31                            cl.close();
32                            break;
33                        }
34                    }
35                } while (!mensaje.equalsIgnoreCase("salir")); //El ciclo se cierra cuando mensaje es salir
36                System.out.println("Esperando Cliente..."); //Una vez cerrado el socket del cliente nuestro s
37            } catch (Exception e) {
38                e.printStackTrace();
39            }
40        }
41    }
42 }
43
```

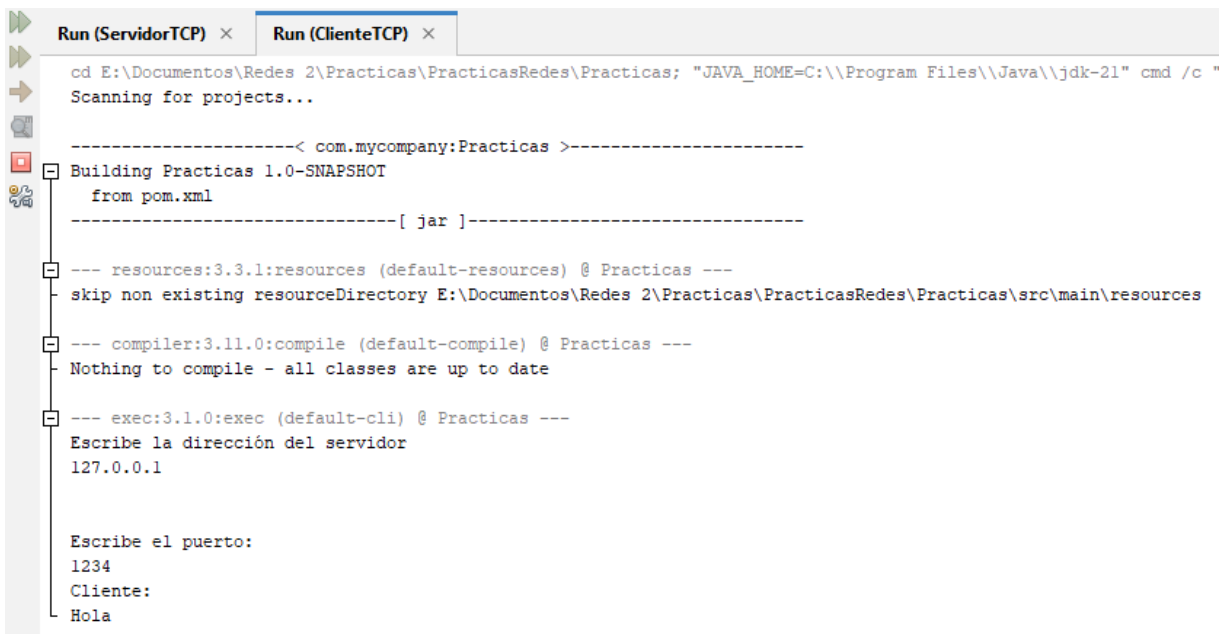
## Intercambio de mensajes

El socket del servidor comienza a esperar que un cliente se conecte.



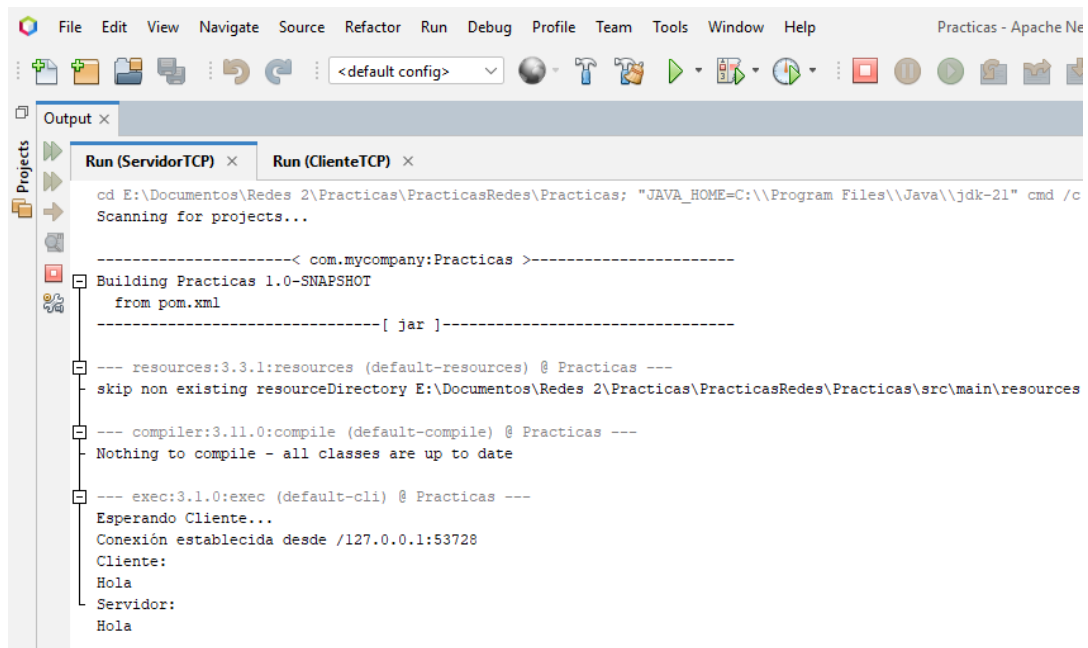
```
cd E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c " Scanning for projects... -----< com.mycompany:Practicas >----- Building Practicas 1.0-SNAPSHOT from pom.xml -----[ jar ]----- --- resources:3.3.1:resources (default-resources) @ Practicas --- skip non existing resourceDirectory E:\\Documentos\\Redes 2\\Practicas\\PracticasRedes\\Practicas\\src\\main\\resources --- compiler:3.11.0:compile (default-compile) @ Practicas --- Nothing to compile - all classes are up to date --- exec:3.1.0:exec (default-cli) @ Practicas --- Esperando Cliente...
```

El cliente ingresa la dirección y puerto a donde debe conectarse. Se envía el primer mensaje



```
cd E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c " Scanning for projects... -----< com.mycompany:Practicas >----- Building Practicas 1.0-SNAPSHOT from pom.xml -----[ jar ]----- --- resources:3.3.1:resources (default-resources) @ Practicas --- skip non existing resourceDirectory E:\\Documentos\\Redes 2\\Practicas\\PracticasRedes\\Practicas\\src\\main\\resources --- compiler:3.11.0:compile (default-compile) @ Practicas --- Nothing to compile - all classes are up to date --- exec:3.1.0:exec (default-cli) @ Practicas --- Escribe la dirección del servidor 127.0.0.1 Escribe el puerto: 1234 Cliente: Hola
```

El servidor recibe el mensaje y envia otro.



```
cd E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas; "JAVA_HOME=C:\\Program Files\\Java\\jdk-21" cmd /c
Scanning for projects...

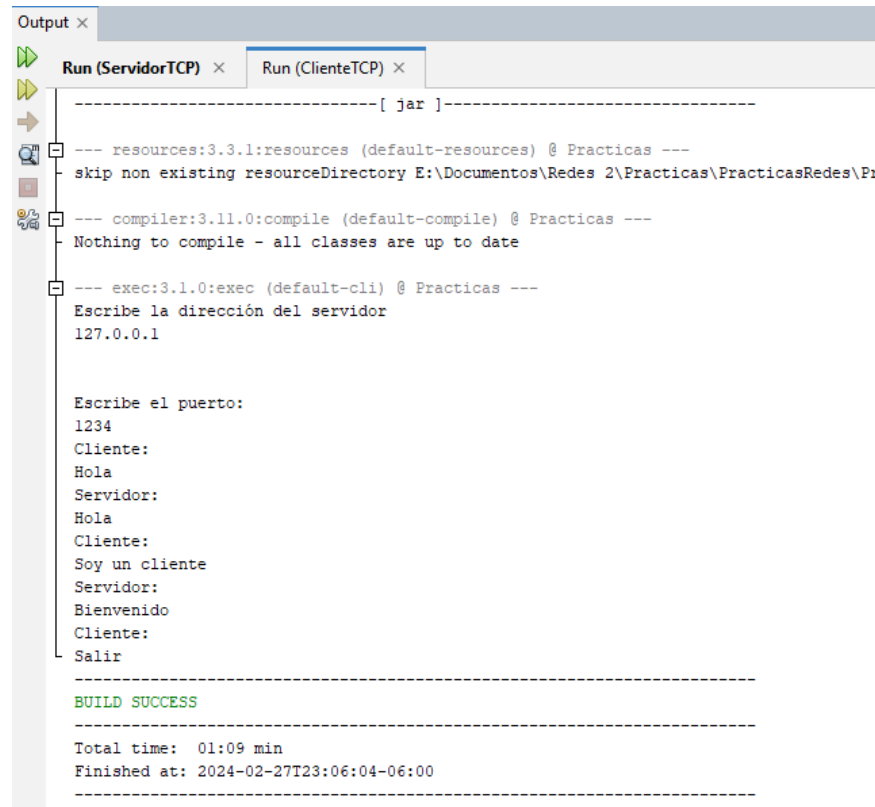
-----< com.mycompany:Practicas >-----
Building Practicas 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Practicas ---
skip non existing resourceDirectory E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Practicas ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Practicas ---
Esperando Cliente...
Conexión establecida desde /127.0.0.1:53728
Cliente:
Hola
Servidor:
Hola
```

Se cierra el socket desde el cliente.



```
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Practicas ---
skip non existing resourceDirectory E:\Documentos\Redes 2\Practicas\PracticasRedes\Pr

--- compiler:3.11.0:compile (default-compile) @ Practicas ---
Nothing to compile - all classes are up to date

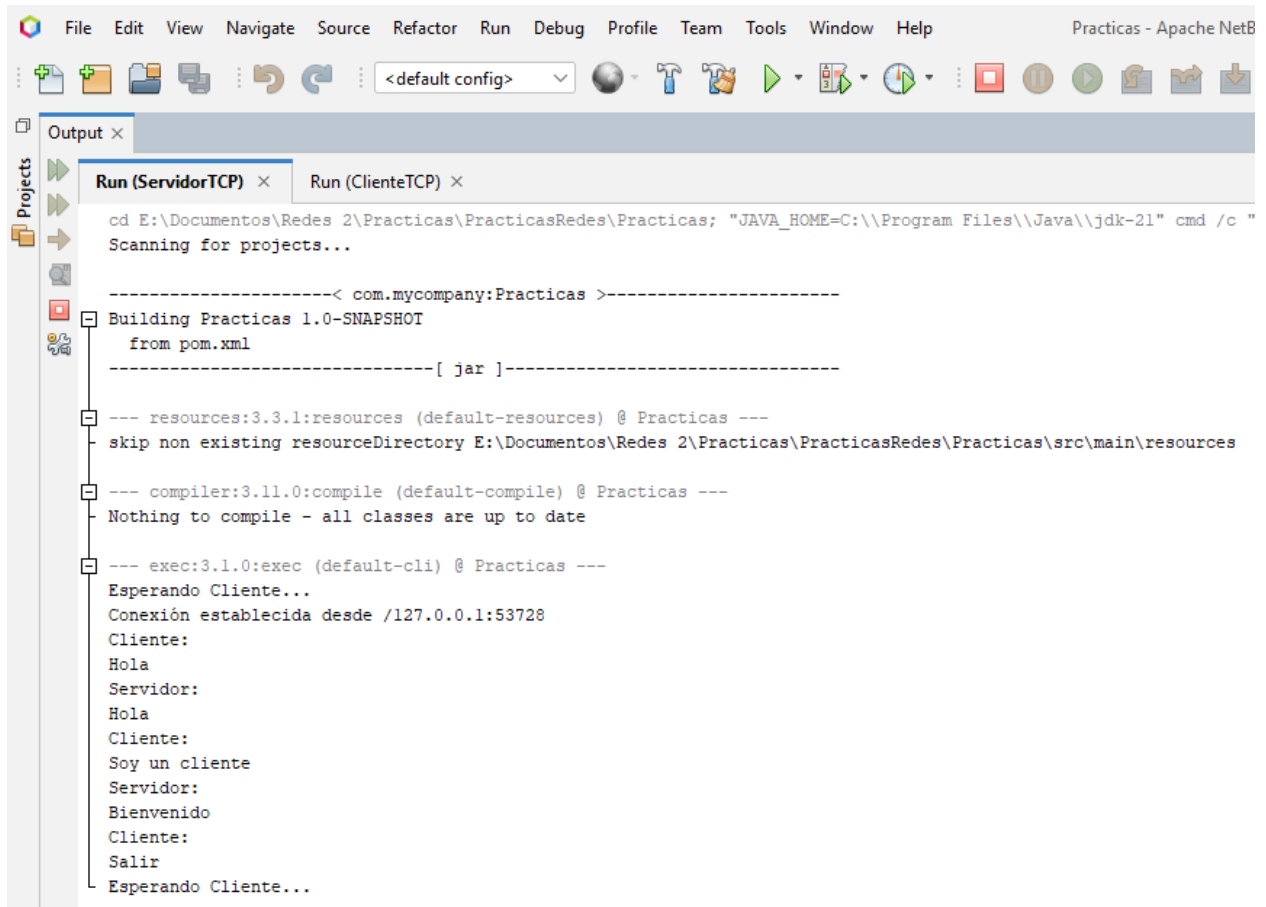
--- exec:3.1.0:exec (default-cli) @ Practicas ---
Escribe la dirección del servidor
127.0.0.1

Escribe el puerto:
1234
Cliente:
Hola
Servidor:
Hola
Cliente:
Soy un cliente
Servidor:
Bienvenido
Cliente:
Salir

BUILD SUCCESS

Total time: 01:09 min
Finished at: 2024-02-27T23:06:04-06:00
```

El servidor cierra la conexión con el cliente y vuelve al estado de espera de conexión.



```
cd E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas; "JAVA_HOME=C:\Program Files\Java\jdk-21" cmd /c "
Scanning for projects...

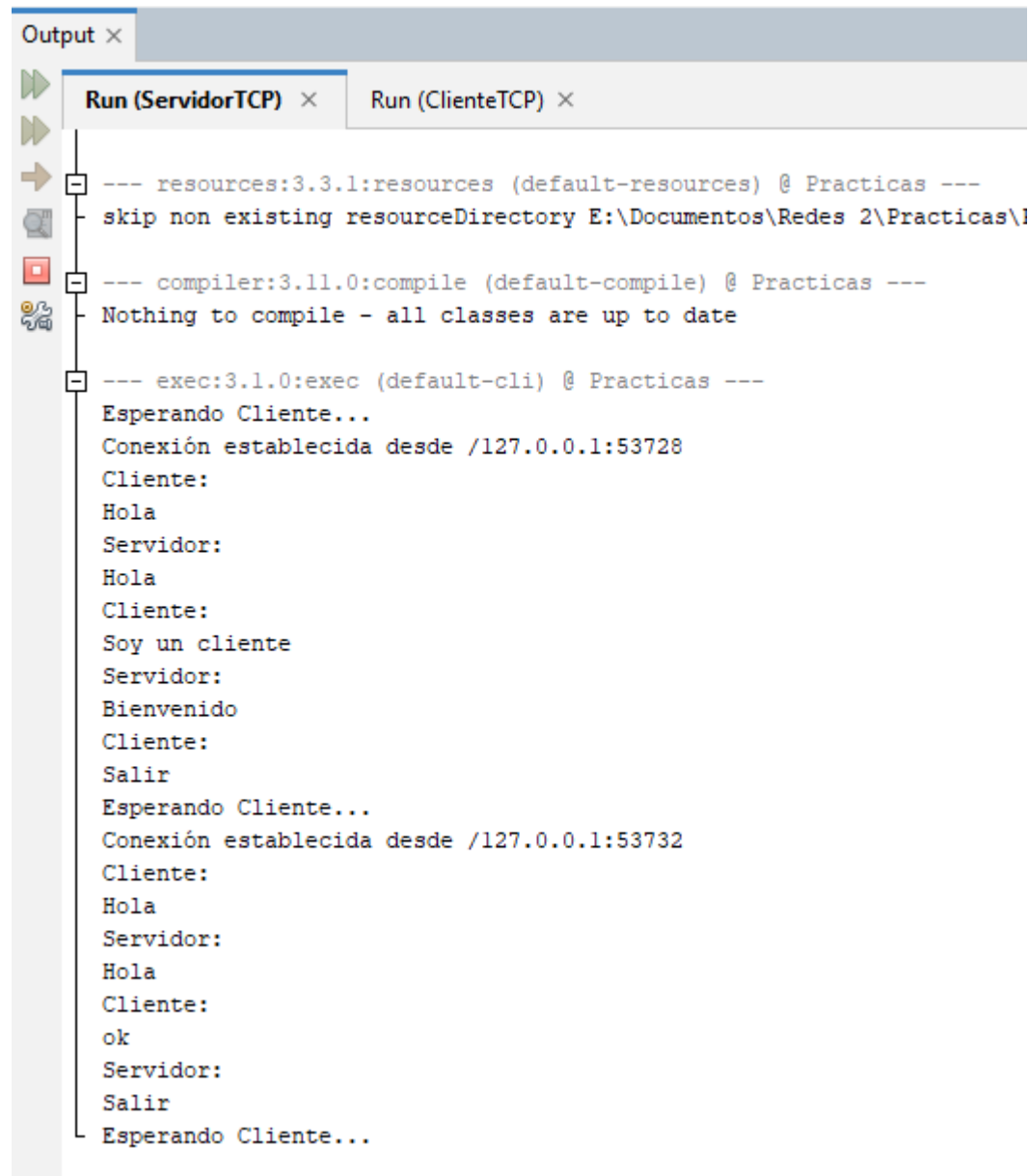
-----< com.mycompany:Practicas >-----
Building Practicas 1.0-SNAPSHOT
  from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Practicas ---
  skip non existing resourceDirectory E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Practicas ---
  Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Practicas ---
  Esperando Cliente...
  Conexión establecida desde /127.0.0.1:53728
  Cliente:
  Hola
  Servidor:
  Hola
  Cliente:
  Soy un cliente
  Servidor:
  Bienvenido
  Cliente:
  Salir
  Esperando Cliente...
```

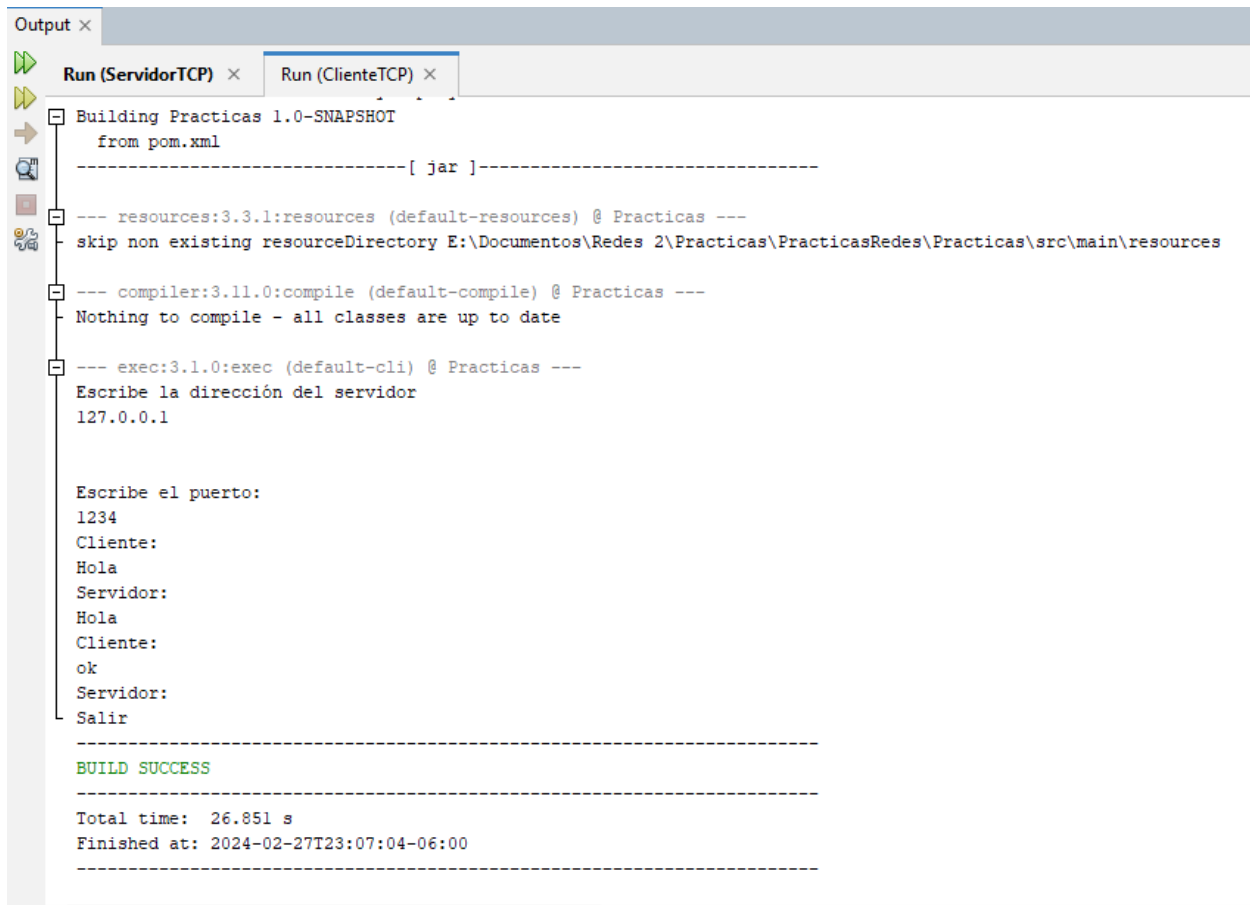
Se realiza la misma interacción que con los ejemplos anteriores pero ahora el servidor es quien cierra la conexión.



```
Output x
Run (ServidorTCP) x Run (ClienteTCP) x
--- resources:3.3.1:resources (default-resources) @ Practicas ---
skip non existing resourceDirectory E:\Documentos\Redes 2\Practicas\!
--- compiler:3.11.0:compile (default-compile) @ Practicas ---
Nothing to compile - all classes are up to date
--- exec:3.1.0:exec (default-cli) @ Practicas ---
Esperando Cliente...
Conexión establecida desde /127.0.0.1:53728
Cliente:
Hola
Servidor:
Hola
Cliente:
Soy un cliente
Servidor:
Bienvenido
Cliente:
Salir
Esperando Cliente...
Conexión establecida desde /127.0.0.1:53732
Cliente:
Hola
Servidor:
Hola
Cliente:
ok
Servidor:
Salir
Esperando Cliente...
```



Cuando el servidor cierra el socket esto es lo que podemos observar en la consola del cliente.



```
Output x
Run (ServidorTCP) x Run (ClienteTCP) x
Building Practicas 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
--- resources:3.3.1:resources (default-resources) @ Practicas ---
skip non existing resourceDirectory E:\Documentos\Redes 2\Practicas\PracticasRedes\Practicas\src\main\resources
--- compiler:3.11.0:compile (default-compile) @ Practicas ---
Nothing to compile - all classes are up to date
--- exec:3.1.0:exec (default-cli) @ Practicas ---
Escribe la dirección del servidor
127.0.0.1

Escribe el puerto:
1234
Cliente:
Hola
Servidor:
Hola
Cliente:
ok
Servidor:
Salir

BUILD SUCCESS

Total time: 26.851 s
Finished at: 2024-02-27T23:07:04-06:00
```