



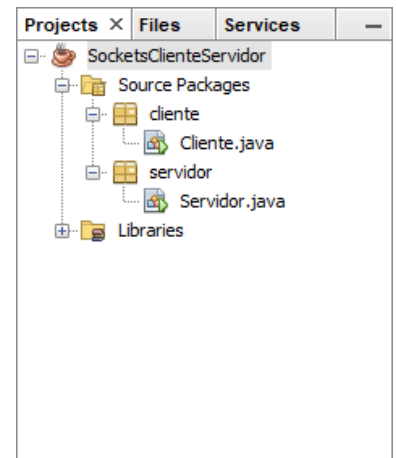
## Laboratorio 02

### Cliente Servidor con Sockets

#### Cliente Servidor con Sockets (SocketsClienteServidor)

Cliente.java

```
1 package cliente;
2
3 import java.io.*;
4 import java.net.Socket;
5 import java.util.logging.*;
6
7 public class Cliente {
8     public static void main(String[] args) {
9         final String HOST = "localhost";
10        final int PUERTO = 9393;
11        DataInputStream in;
12        DataOutputStream out;
13        try{
14            //Socket para conectar con el cliente
15            Socket sc = new Socket(HOST, PUERTO);
16            in = new DataInputStream(sc.getInputStream());
17            out = new DataOutputStream(sc.getOutputStream());
18            //Mensaje enviado al cliente
19            out.writeUTF("HolaMundoCliente");
20            //Mensaje recibido del servidor
21            String mensaje = in.readUTF();
22            System.out.println(mensaje);
23            sc.close();
24        } catch (IOException ex) {
25            Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null, ex);
26        }
27    }
28 }
```



Servidor.java

```
1 package servidor;
2
3 import java.io.*;
4 import java.net.*;
5 import java.util.logging.*;
6
7 public class Servidor {
8     public static void main(String[] args) {
9         ServerSocket servidor = null;
10        Socket sc = null;
11        DataInputStream in;
12        DataOutputStream out;
13        final int PUERTO = 9393;
14        try{
15            //Se crea el socket del servidor
16            servidor = new ServerSocket(PUERTO);
17            System.out.println("Servidor Hola Mundo preparado.");
18            while (true){
19                //Espera a que un cliente se conecte con el servidor
20                sc = servidor.accept();
21                System.out.println("Cliente conectado");
22                in = new DataInputStream(sc.getInputStream());
23                out = new DataOutputStream(sc.getOutputStream());
24                //Se lee el mensaje que me envia
25                String mensaje = in.readUTF();
26                System.out.println(mensaje);
27                //Se envia un mensaje al cliente
28                out.writeUTF("HolaMundoServidor");
29                //Cerrar el socket
30                sc.close();
31                System.out.println("Cliente desconectado");
32            }
33        } catch (IOException ex) {
34            Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
35        }
36    }
37 }
```



```
32         }
33     } catch (IOException ex) {
34         Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
35     }
36 }
37 }
```

### Run File in Servidor.java

```
run:
Servidor Hola Mundo preparado.
```

### Run File in Cliente.java

```
run:
HolaMundoServidor
BUILD SUCCESSFUL (total time: 0 seconds)
```

### SocketsClientesServidor(run)

```
run:
Servidor Hola Mundo preparado.
Cliente conectado
HolaMundoCliente
Cliente desconectado
```