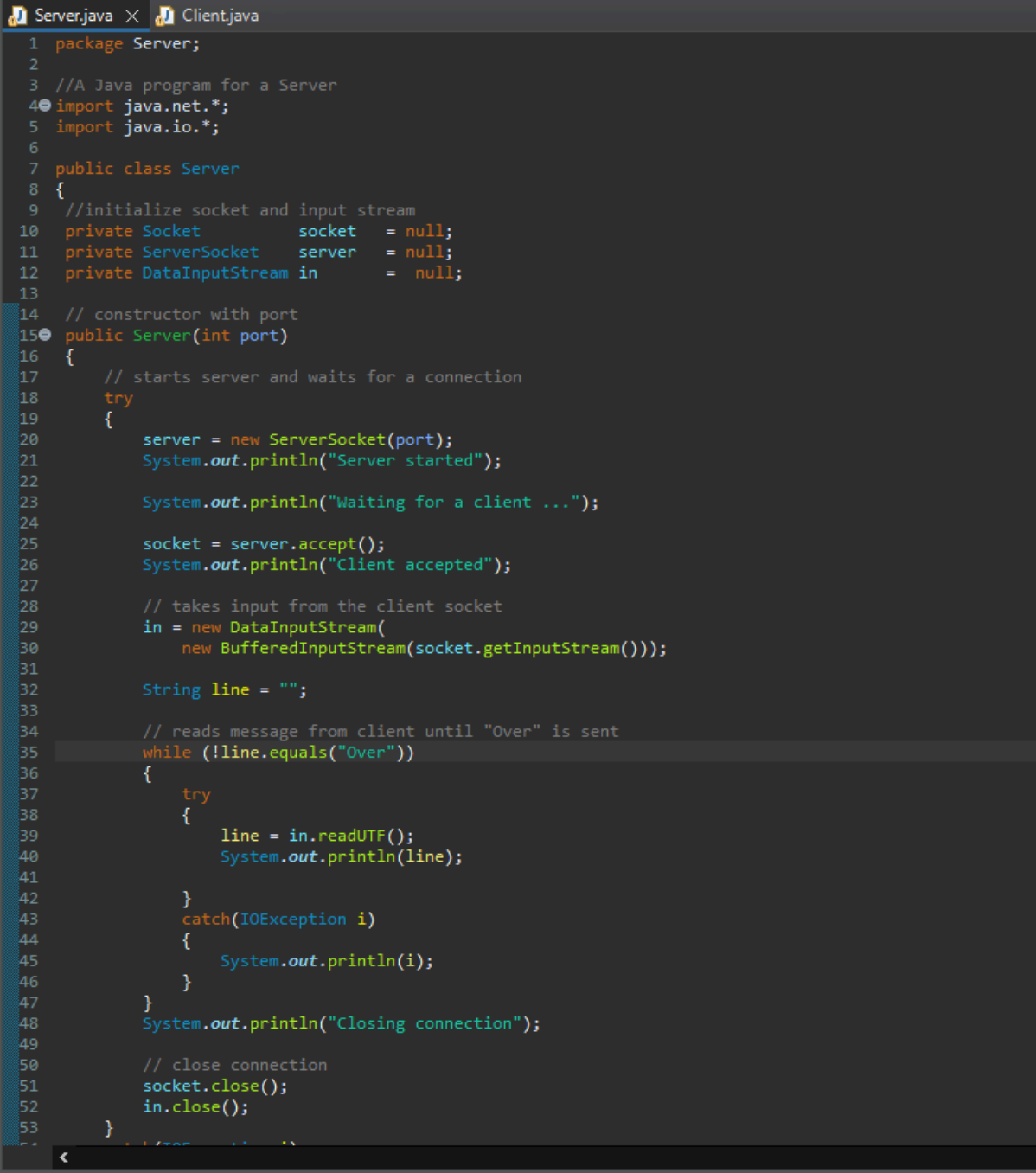


Lo primero va a ser crear los programas de java Client y Server. En la página <https://www.geeksforgeeks.org/socket-programming-in-java/> están los 2 programas. Simplemente los copiamos, creamos en eclipse los programas y pegamos ahí mismo.



```
1 package Server;
2
3 //A Java program for a Server
4 import java.net.*;
5 import java.io.*;
6
7 public class Server
8 {
9     //initialize socket and input stream
10    private Socket      socket    = null;
11    private ServerSocket server    = null;
12    private DataInputStream in      = null;
13
14    // constructor with port
15    public Server(int port)
16    {
17        // starts server and waits for a connection
18        try
19        {
20            server = new ServerSocket(port);
21            System.out.println("Server started");
22
23            System.out.println("Waiting for a client ...");
24
25            socket = server.accept();
26            System.out.println("Client accepted");
27
28            // takes input from the client socket
29            in = new DataInputStream(
30                new BufferedInputStream(socket.getInputStream()));
31
32            String line = "";
33
34            // reads message from client until "Over" is sent
35            while (!line.equals("Over"))
36            {
37                try
38                {
39                    line = in.readUTF();
40                    System.out.println(line);
41
42                }
43                catch(IOException i)
44                {
45                    System.out.println(i);
46                }
47            }
48            System.out.println("Closing connection");
49
50            // close connection
51            socket.close();
52            in.close();
53        }
54    }
55 }
```

```

1 package Client;
2
3 import java.io.*;
4 import java.net.*;
5
6 public class Client {
7     // initialize socket and input output streams
8     private Socket socket = null;
9     private DataInputStream input = null;
10    private DataOutputStream out = null;
11
12    // constructor to put in address and port
13    public Client(String address, int port)
14    {
15        // establish a connection
16        try {
17            socket = new Socket(address, port);
18            System.out.println("Connected");
19
20            // takes input from terminal
21            input = new DataInputStream(System.in);
22
23            // sends output to the socket
24            out = new DataOutputStream(
25                socket.getOutputStream());
26        }
27        catch (UnknownHostException u) {
28            System.out.println(u);
29            return;
30        }
31        catch (IOException i) {
32            System.out.println(i);
33            return;
34        }
35
36        // string to read message from input
37        String line = "";
38
39        // keep reading until "Over" is input
40        while (!line.equals("Over")) {
41            try {
42                line = input.readLine();
43                out.writeUTF(line);
44            }
45            catch (IOException i) {
46                System.out.println(i);
47            }
48        }
49
50        // close the connection
51        try {
52            input.close();
53            out.close();
54        }
55    }
56 }

```

También debemos asegurarnos que van al mismo puerto(En mi caso he puesto el puerto 5555):

```
public static void main(String args[])
{
    Client client = new Client("127.0.0.1", 5555);
}
```

```
60 public static void main(String args[])
61 {
62     Server server = new Server(5555);
63 }
64 }
65 }
```

Ahora vamos a ir a la workspace de eclipse:

Nombre	Fecha de modificación	Tipo	Tamaño
.metadata	12/03/2024 18:48	Carpeta de archivos	
Client	20/03/2024 18:46	Carpeta de archivos	
Server	20/03/2024 18:44	Carpeta de archivos	
Socket	12/03/2024 18:48	Carpeta de archivos	
Socket1	20/03/2024 18:41	Carpeta de archivos	

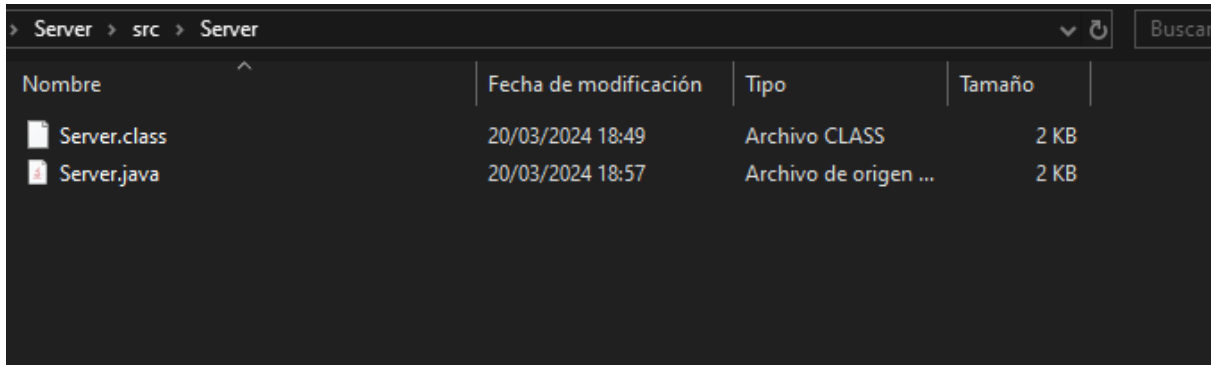
Tras eso vamos a meternos en la carpeta Server:

Nombre	Fecha de modificación	Tipo	Tamaño
.settings	20/03/2024 18:44	Carpeta de archivos	
bin	20/03/2024 18:45	Carpeta de archivos	
src	20/03/2024 18:45	Carpeta de archivos	
.classpath	20/03/2024 18:44	Archivo CLASSPATH	1 KB
.project	20/03/2024 18:44	Archivo PROJECT	1 KB

Después vamos a src:

Nombre	Fecha de modificación	Tipo	Tamaño
Server	20/03/2024 18:49	Carpeta de archivos	
module-info.java	20/03/2024 18:44	Archivo de origen ...	1 KB

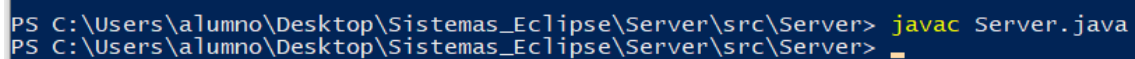
Y por último a Server:



The screenshot shows a file explorer window with the path 'Server > src > Server'. It contains a table with the following data:

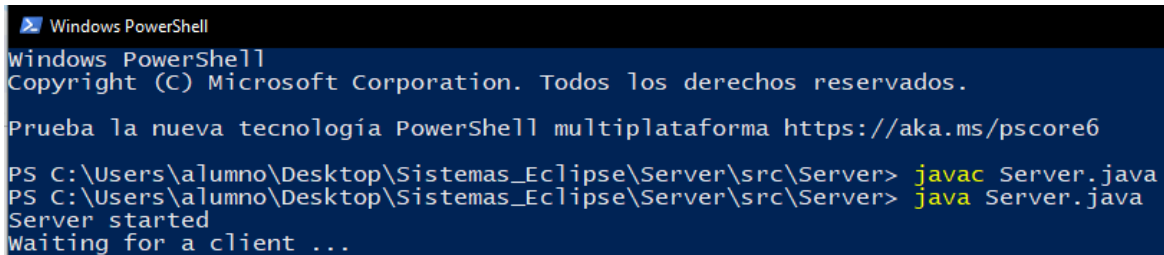
Nombre	Fecha de modificación	Tipo	Tamaño
Server.class	20/03/2024 18:49	Archivo CLASS	2 KB
Server.java	20/03/2024 18:57	Archivo de origen ...	2 KB

Ahí podemos ver el server.java ya preparado. Ejecutamos el powershell en esta carpeta y escribimos javac Server.java para COMPILAR EL PROGRAMA.



```
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> javac Server.java
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server>
```

Tras eso ejecutamos el programa con java Server.java:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> javac Server.java
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> java Server.java
Server started
Waiting for a client ...
```

Como podemos observar ya tenemos el servidor y está esperando a que se conecte un cliente.

Ahora vamos a seguir los mismos pasos pero con el cliente.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> _
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
_
```

Ahora si nos fijamos en el otro terminal podemos observar cómo el servidor ha recibido el cliente correctamente:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

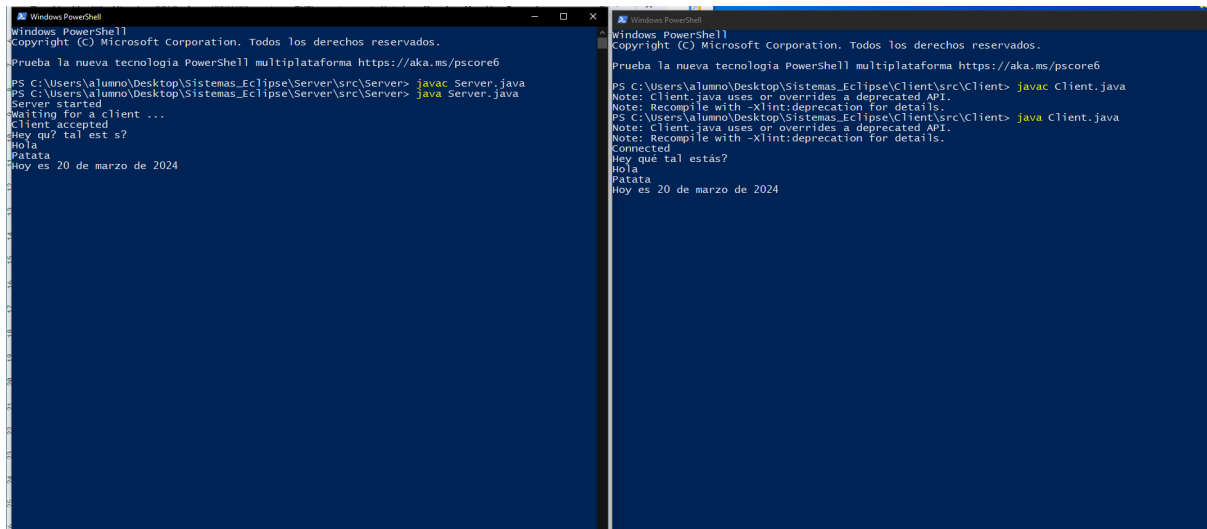
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> javac Server.java
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> java Server.java
Server started
waiting for a client ...
Client accepted

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
```

También podemos enviar líneas al servidor desde el cliente y ver cómo le llegan:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> javac Server.java
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> java Server.java
Server started
waiting for a client ...
Client accepted
Hey qu? tal est s?
Hola
Patata
Hoy es 20 de marzo de 2024

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

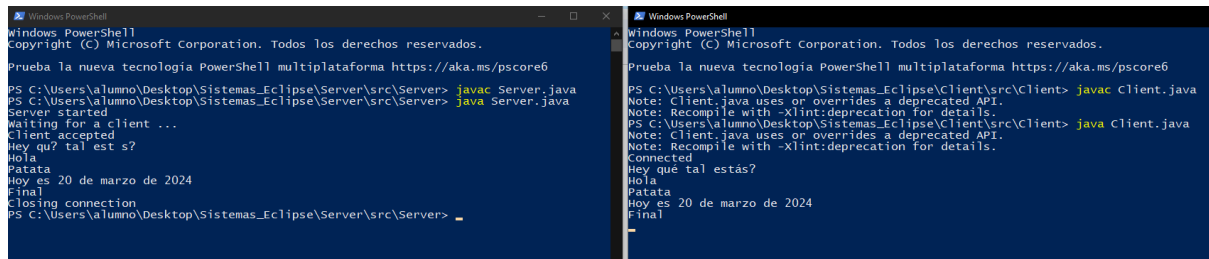
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Hey que tal estás?
Hola
Patata
Hoy es 20 de marzo de 2024
```

Podemos observar en el programa del servidor como hay un while que significa que mientras no sea “Final” haga eso. Que significa que mientras la línea introducida por el cliente no sea final siga el programa.

```
33
34     // reads message from client until "Over" is sent
35     while (!line.equals("Final"))
36     {
37         try
38         {
39             line = in.readUTF();
40             System.out.println(line);
41
42         }
43         catch(IOException i)
44         {
45             System.out.println(i);
46         }
47     }
48     System.out.println("Closing connection");
49
50     // close connection
51     socket.close();
52     in.close();
53 }
54 catch(IOException i)
55 {
56     System.out.println(i);
57 }
```

Ahora probemos qué pasa cuando ponemos Final desde el cliente:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> javac Server.java
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> java Server.java
Server started
waiting for a client ...
Client accepted
Hey qu? tal est s?
Hola
Patata
Hoy es 20 de marzo de 2024
Final
Closing connection
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server>

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

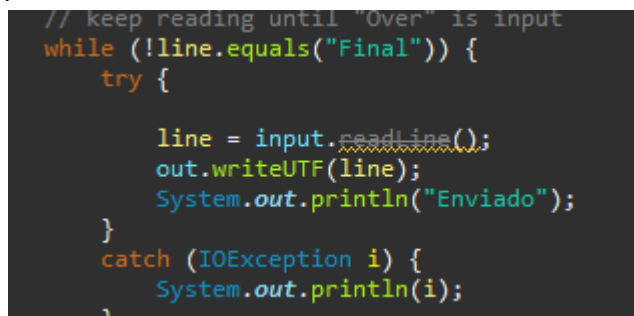
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Hey qué tal estás?
Hola
Patata
Hoy es 20 de marzo de 2024
Final
```

Cuando recibe la palabra final sale del while y cierra automáticamente la conexión.

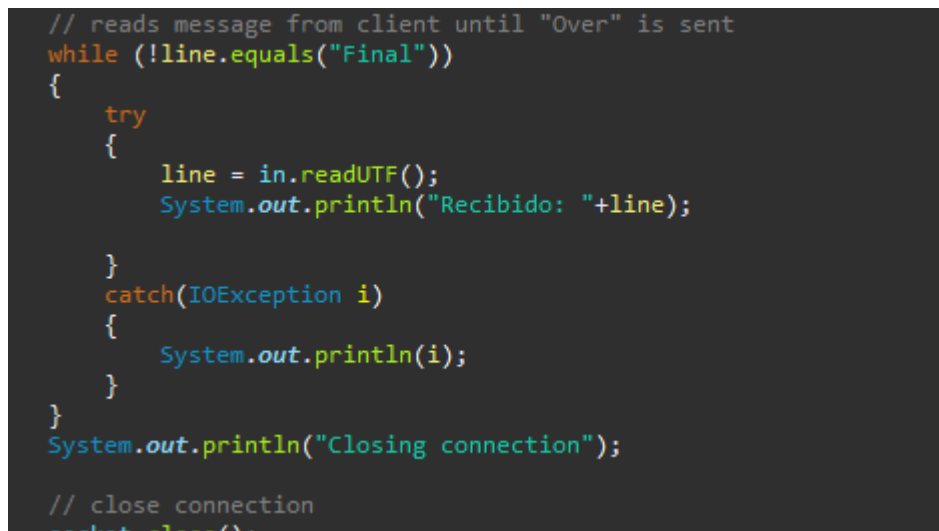
EXTRA:

También podemos hacer que salga cuando ha sido enviado un mensaje poniendo que printee un enviado en el de cliente:



```
// keep reading until "Over" is input
while (!line.equals("Final")) {
    try {
        line = input.readLine();
        out.writeUTF(line);
        System.out.println("Enviado");
    }
    catch (IOException i) {
        System.out.println(i);
    }
}
```

y un print de recibido en el de servidor:



```
// reads message from client until "Over" is sent
while (!line.equals("Final"))
{
    try
    {
        line = in.readUTF();
        System.out.println("Recibido: "+line);
    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
System.out.println("Closing connection");

// close connection
socket.close();
```

Para que el servidor no se cierre a pesar de que el cliente se vaya simplemente vamos a meter un while infinito de esta forma:

```

System.out.println("Server started");

while (true) {
    System.out.println("Waiting for a client ...");

    socket = server.accept();
    System.out.println("Client accepted");

    // takes input from the client socket
    in = new DataInputStream(
        new BufferedInputStream(socket.getInputStream()));

    String line = "";

    // reads message from client until "Over" is sent
    while (!line.equals("Final"))
    {
        line = in.readLine();
        System.out.println("Received: " + line);
    }

    System.out.println("Closing connection");
    socket.close();
}
}

```

De esta manera aunque se cierre el cliente espera otro:

The image shows two side-by-side Windows PowerShell terminal windows. The left window, titled 'Windows PowerShell', shows the execution of the server program. The right window, also titled 'Windows PowerShell', shows the execution of the client program. Both windows show the output of the programs, including messages like 'Server started', 'Waiting for a client ...', 'Client accepted', 'Received: Hola', 'Received: Patata', 'Received: Final', 'Closing connection', and 'Waiting for a client ...' on the server side, and 'Note: Client.java uses or overrides a deprecated API.', 'Note: Recompile with -Xlint:deprecation for details.', 'Connected', 'Enviado', 'Hola', 'Enviado', 'Patata', 'Enviado', 'Final', and 'Enviado' on the client side.

```

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Server\src\Server> java Server.java
Server started
Waiting for a client ...
Client accepted
Received: Hola
Received: Patata
Received: Final
Closing connection
Waiting for a client ...

PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Enviado
Hola
Enviado
Patata
Enviado
Final
Enviado
PS C:\Users\alumno\Desktop\Sistemas_Eclipse\Client\src\Client>

```