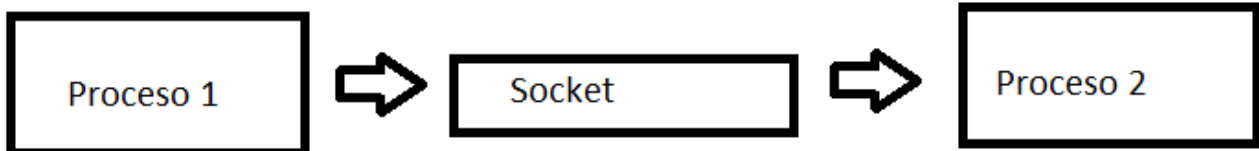


# Sockets

Un Socket es un punto de conexión (Una Tubería o un Canal) entre 2 procesos e identificado por una IP y un Puerto. En este caso serán 2 procesos ejecutándose simultáneamente conectados por un socket.



Un socket funciona de una forma similar a la de los ficheros.

## Cliente

```
Socket socket = new Socket("127.0.0.1", 5000);

BufferedReader br = null;
PrintWriter pw = null;

br = new BufferedReader(new InputStreamReader(socket.getInputStream()))//Buffer de lectura
pw = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()))//Buffer de Escritura

br.readLine();
pw.println("texto a enviar");//Texto a mandar por el Socket
pw.flush();//Similar al fflush stdin de C OJO
pw.push();//
```

## Servidor

```
//Primero creamos el socket
ServerSocket serverSocket = new ServerSocket(2013)//Indicamos el puerto de escucha, en este caso 2013

//esperamos una petición
Socket socket = ServerSocket.accept();//Esto para la ejecución y nos devuelve un socket

BufferedReader br = null;
PrintWriter pw = null;

br = new BufferedReader(new InputStreamReader(socket.getInputStream()))//Buffer de lectura
pw = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()))//Buffer de Escritura

br.readLine();
pw.println("texto a enviar");//Texto a mandar por el Socket
pw.flush();//Similar al fflush stdin de C OJO
pw.push();
```

# Implementación Servidor

```
1 package jelouda;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.io.OutputStreamWriter;
7 import java.io.PrintWriter;
8 import java.net.ServerSocket;
9 import java.net.Socket;
10
11 public class Servidor {
12
13     public static int PUERTO = 5000;
14     public void ejecutar() {
15         try {
16             ServerSocket serverSocket = new ServerSocket(Servidor.PUERTO);
17
18             Socket socket = serverSocket.accept();
19
20             BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));
21             PrintWriter pw = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()));
22
23             String cadenaRecibida = br.readLine();
24
25             pw.print(cadenaRecibida);
26             pw.flush();
27
28             System.out.println("Fin del Servidor");
29
30
31             }catch(IOException e){
32                 e.printStackTrace();
33             }
34         }
35     }
36
37     public static void main(String[] args) {
38
39     }
40
41 }
42
```

From:

<https://www.knoppia.net/> - Knoppia

Permanent link:

<https://www.knoppia.net/doku.php?id=dad:sockets&rev=1696498014>

Last update: **2023/10/05 09:26**

