



# Sockets

[java.net](http://java.net)

# Sockets

- La clase `ServerSocket` sirve para esperar una conexión en un puerto.
- La clase `Socket` sirve para establecer una conexión desde un cliente a un servidor en un puerto y para recibir la conexión en el servidor.

# Flujos de entrada y salida

- El flujo de salida del ServerSocket se debe comunicar con el flujo de entrada del Socket.
- El flujo de entrada del ServerSocket se debe comunicar con el flujo de salida del Socket.

# Servidor

```
ServerSocket servicio = new ServerSocket(5000);  
Socket servidor = servicio.accept();  
DataInputStream flujoE = new  
    DataInputStream(servidor.getInputStream());  
DataOutputStream flujoS = new  
    DataOutputStream(servidor.getOutputStream());  
flujoS.writeUTF("hola cliente ");  
String valor = flujoE.readUTF();  
System.out.println("servidor recibe: "+valor);  
flujoE.close();  
flujoS.close();  
servidor.close();  
servicio.close();
```

# Cliente

```
Socket cliente = new Socket("localhost",5000);
DataInputStream flujoE = new
    DataInputStream(cliente.getInputStream());
DataOutputStream flujoS = new
    DataOutputStream(cliente.getOutputStream());
String valor = flujoE.readUTF();
System.out.println("cliente recibe: "+valor);
flujoS.writeUTF("adios servidor");
flujoE.close();
flujoS.close();
cliente.close();
```

# Mantener conversación

## Servidor

```
flujoS.writeUTF("hola cliente ");  
String valor="";  
while(valor.compareTo("fin")!=0){  
    valor = flujoE.readUTF();  
    System.out.println("servidor recibe: "+valor);  
}  
flujoE.close();
```

## Cliente

```
String valor = flujoE.readUTF();  
System.out.println("cliente recibe: "+valor);  
for (int i = 0; i < 10; i++) {  
    flujoS.writeUTF(i+"");  
}  
flujoS.writeUTF("fin");  
flujoE.close();
```

# Conectar con varios clientes

```
while (condicion) {  
    new Hebra(servicio.accept()).start();  
}  
servicio.close();
```

# Hebra Servidor

```
public class Hebra extends Thread {  
    private Socket s;  
    public Hebra(Socket s) throws IOException {  
        this.s = s;  
        flujoE = new DataInputStream(s.getInputStream());  
        flujoS = new DataOutputStream(s.getOutputStream());  
        flujoS.writeUTF("hola cliente ");  
    }  
    @Override  
    public void run() {  
        String valor = "";  
        while (valor.compareTo("fin") != 0) {  
            valor = flujoE.readUTF();  
            System.out.println("servidor recibe: " + valor);  
        }  
        flujoE.close();  
        flujoS.close();  
        s.close();  
    }  
}
```