

Cliente/Servidor TCP básico e iterativo en Java

Servidor	Cliente	Explicación
1. <code>ss ← new ServerSocket(puerto)</code>		<ul style="list-style-type: none"> El servidor crea un <u>socket pasivo</u> (sólo recibe peticiones) asociado al puerto pasado. También es posible crearlo sin vincularlo a un puerto y posteriormente con el método bind indicar por qué puerto recibirá peticiones. También se puede indicar el tamaño de la cola de clientes en espera.
2.	<code>sc ← new Socket(dirSocketServidor)</code>	<ul style="list-style-type: none"> El cliente crea un <u>socket activo</u> para conectarse al servidor. La petición se almacenará en la cola de clientes pendientes del servidor
3. <code>s ← ss.accept</code>		<ul style="list-style-type: none"> El servidor saca un cliente de la cola de clientes pendientes y crea un nuevo <u>socket activo</u> conectado con el cliente.
4. <code>in ← s.getInputStream out ← s.getOutputStream</code>	<code>in ← sc.getInputStream out ← sc.getOutputStream</code>	<ul style="list-style-type: none"> El cliente y el servidor obtienen los flujos que permiten la conexión. Cuando escribe en su flujo de salida, eso traduce en un envío. Igualmente una lectura es la recepción desde el extremo opuesto. Para usar métodos de lectura y escritura simples (readLine o println), hay que convertirlos a otras clases (ver apuntes de Socket en Java).
5. <code>in.readLine out.println</code>	<code>out.println in.readLine</code>	<ul style="list-style-type: none"> Envío/recepción. Observe que el cliente y servidor hacen la operación opuesta al otro extremo en cada momento. El orden y la cantidad de envíos/recepción y su orden depende del protocolo solicitado a nivel de aplicación.
6. <code>s.close</code>	<code>sc.close</code>	<ul style="list-style-type: none"> Se indica el cierre de las conexión en ambos extremos (cerramos los sockets activos). Recuerde en cerrar también todos los flujos utilizados antes de cerrar el socket.
7. Volver a paso 3		<ul style="list-style-type: none"> Una vez el servidor ya ha atendido al cliente actual, vuelve a mirar si en la cola hay más clientes pendientes.