

Cliente/Servidor UDP básico en Java

Servidor	Cliente	Explicación
1. <code>ss ← new DatagramSocket(port)</code>		<ul style="list-style-type: none"> El servidor crea un socket que puede recibir por el puerto indicado. A diferencia de TCP, este socket ya puede recibir y enviar directamente.
	2. <code>sc ← new DatagramSocket()</code>	<ul style="list-style-type: none"> El cliente crea un socket con un puerto efímero (cualquiera libre). También podría indicar el puerto a usar.
3. <code>dp ← new DatagramPacket(vacio)</code>		<ul style="list-style-type: none"> El servidor crea un datagrama vacío (espacio en memoria), que se rellenará cuando reciba un datagrama.
	4. <code>dp ← new DatagramPacket(dir, data)</code>	<ul style="list-style-type: none"> El cliente crea un datagrama con la dirección destino (IP y puerto) y el mensaje a enviar.
	5. <code>sc.send(dp)</code>	<ul style="list-style-type: none"> Envía el datagrama
6. <code>ss.receive(dp)</code>		<ul style="list-style-type: none"> Recibe un datagrama (si no hay se queda esperando a recibir uno) y rellena el datagrama vacío que creamos antes.
7. <code>dp.getData / dp.getLength</code> <code>dp.getPort</code> <code>dp.getAddress</code>		<ul style="list-style-type: none"> Obtenemos los datos recibidos: <ul style="list-style-type: none"> Dirección IP del emisor del datagrama Puerto del emisor del datagrama Datos enviados. Como no estamos con nadie, en cada datagrama leído hay que examinar quién es el origen.
8. <code>dp ← new DatagramPacket(dir, data)</code> <code>ss.send(dp)</code>		<ul style="list-style-type: none"> Generamos la respuesta de acuerdo a lo indicado en el datagrama. Generamos el datagrama con la respuesta y como destino los datos recuperados en el paso anterior
	9. <code>dp ← new DatagramPacket(vacio)</code> <code>sc.receive(dp)</code> <code>dp.getData / dp.getLength</code>	<ul style="list-style-type: none"> Creamos un datagrama vacío para almacenar la respuesta. Esperamos a recibir la respuesta Analizamos la respuesta
10.	<code>sc.close</code> o Volver al paso 4	<ul style="list-style-type: none"> Cierra el socket si ya acabó el servicio En otro caso volvería al paso 4
11. Volver al paso 6		<ul style="list-style-type: none"> Vuelve a esperar otro datagrama