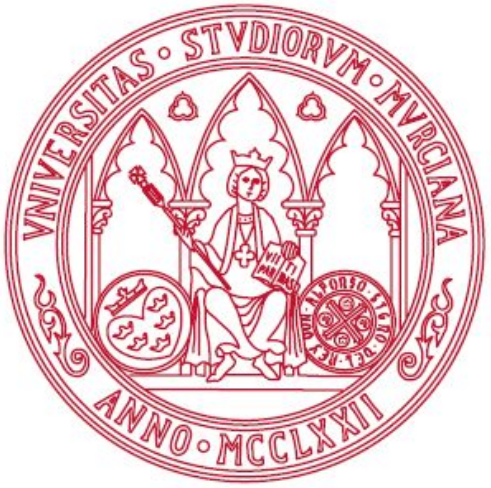


Universidad de Murcia

Grado en Ingeniería Informática

Curso 2020/2021

Programación para las comunicaciones

Profesor: Humberto Martínez Barberá

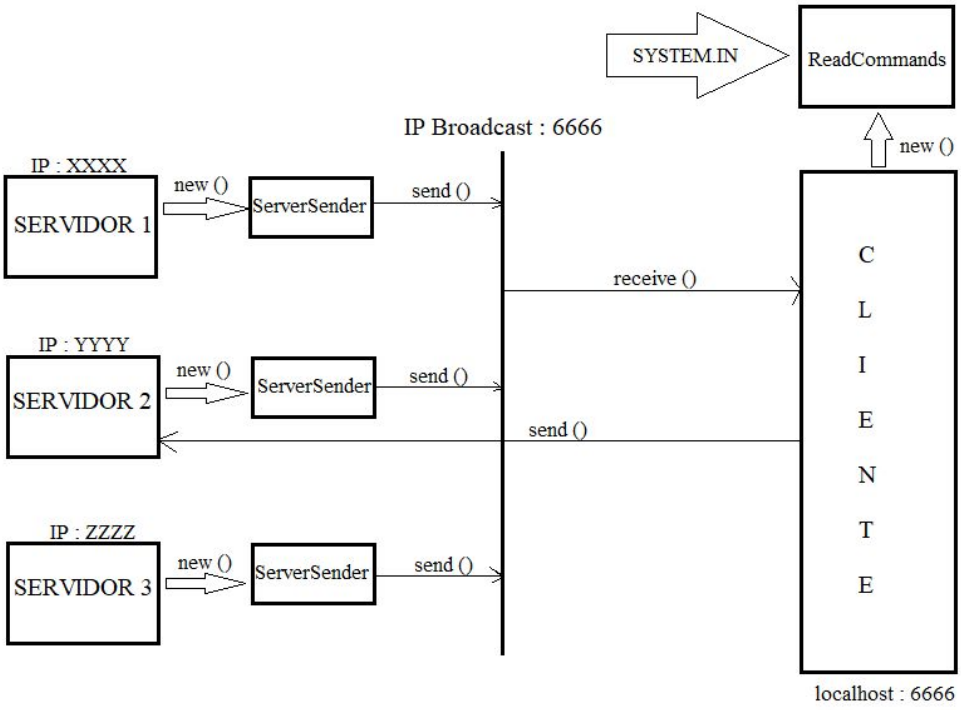
Práctica II

Jaime Ortiz Aragón - 49196689B - jaime.ortiza@um.es

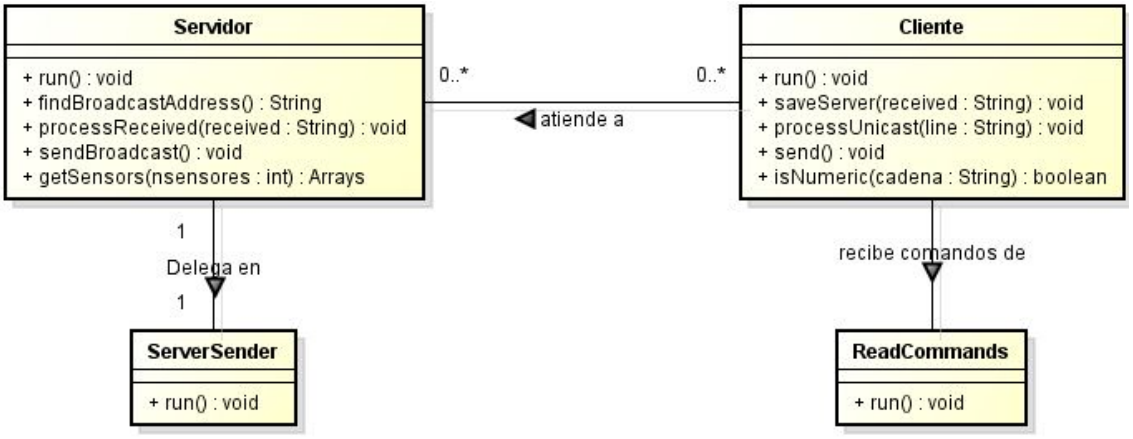
- **ESCENARIO**

Esta práctica trata del intercambio de mensajes UDP entre distintos servidores y un cliente con el fin de que este reciba información sobre el entorno proporcionada por distintos sensores conectados con los servidores. En mi caso, he cogido 6 sensores distintos para medir : temperatura, viento, humedad, partículas por millón, porcentaje de dióxido de azufre y porcentaje de ozono. El objetivo era que todo funcionara de forma simultánea en una aplicación multihilo. El funcionamiento de la aplicación consta del envío repetidamente de mensajes de broadcast por parte de los servidores, los cuales el cliente recibe y puede en cualquier momento cambiar distintos parámetros que se le muestran.

A continuación se muestra un ejemplo cómo se realiza el intercambio de mensajes:



En este diagrama se supone que el mensaje de control se manda al servidor 2, pero se podría mandar a cualquiera.



- **SERVIDOR**

La principal función del servidor es enviar continuamente información sobre los distintos sensores a la dirección de broadcast con el fin de que cualquier cliente que se conecte pueda leer la información. Estos sensores son 6, como se ha descrito en el primer punto del documento y aparecen en cada servidor de forma aleatoria. Como se comentó en clase, hay algunos sensores como temperatura y humedad que suelen ir juntos y que por su precio son más accesibles. El de la velocidad del viento, que no estaba contemplado en el boletín, también considero que puede ser bastante común. Los tres últimos que se han mencionado en el primer apartado son menos comunes y estadísticamente será más difícil que aparezcan en nuestros sensores. Estos actualizan información en cada iteración y la mandan en un DatagramPacket a la dirección de broadcast y puerto 6666, que es el "acordado" para el que debe acceder el cliente si quiere recuperar los datos. En cuanto al modelo de entrada/salida en el servidor, como hemos comentado anteriormente, el cliente podrá mandar mensajes de control a cualquier servidor, por lo que este debe estar a la espera para cualquier cambio que se deba producir en su comportamiento. Para que no se bloquee esperando ningún paquete y evitar que estos se pierdan, el servidor será quien haga la operación de receive de los mensajes unicast del cliente, mientras que los envíos los realizará un hilo auxiliar lanzado por cada servidor.

Cuando este recibe los mensajes, los procesa en función de la petición del cliente. Estos mensajes pueden ser terminar el envío de datos, en el caso de que hayamos terminado de recopilarlos por ejemplo, cambiar las unidades de temperatura entre grados centígrados y kelvin, cambiar las unidades de medición de velocidad del viento entre kilómetros por hora y metros por segundo y cambiar la frecuencia de envío indicando la nueva frecuencia en hertzios.

- **CLIENTE**

En cuanto al modelo de entrada/salida en el cliente, este lanzará un hilo encargado de leer las peticiones del usuario por consola con el fin de construir los mensajes de control. El cliente escuchará en el puerto 6666, que es al que por defecto va a ir la información de los servidores. El cliente mapea los identificadores de cada servidor con su puerto, para posteriormente saber a quién enviar. Esto podemos hacerlo porque al ejecutarlo todo en una sola máquina, la dirección IP de los servidores va a ser la misma. Realmente, para identificar cualquier servidor en internet deberíamos tener en cuenta la tupla IP:PORT. Como se indica en el fichero MultithreadCommunication, que contiene el main, la sintaxis de entrada debe ser ID operación, esta operación puede ser stop, centigrades, kelvin, m/s, km/h, X. Siendo X el número de hertzios al cual queremos cambiar la frecuencia de envío del servidor asociado a ese ID. El paquete irá destinado a la dirección serverAddress. Si se ejecutara el programa en varios equipos, habría que cambiar la implementación de esta parte con el correspondiente mapeo que distinga los servidores.

Por último señalar que se pedían dos mensajes de control, pero finalmente he decidido hacer más y aumentar también el número de sensores con el fin de sacar más rendimiento a la aplicación. También se especifican 2 servidores concurrentes, pero yo he lanzado 3 y no habría mayor problema en lanzar más, más que la incomodidad de trabajar con una consola tan cargada.