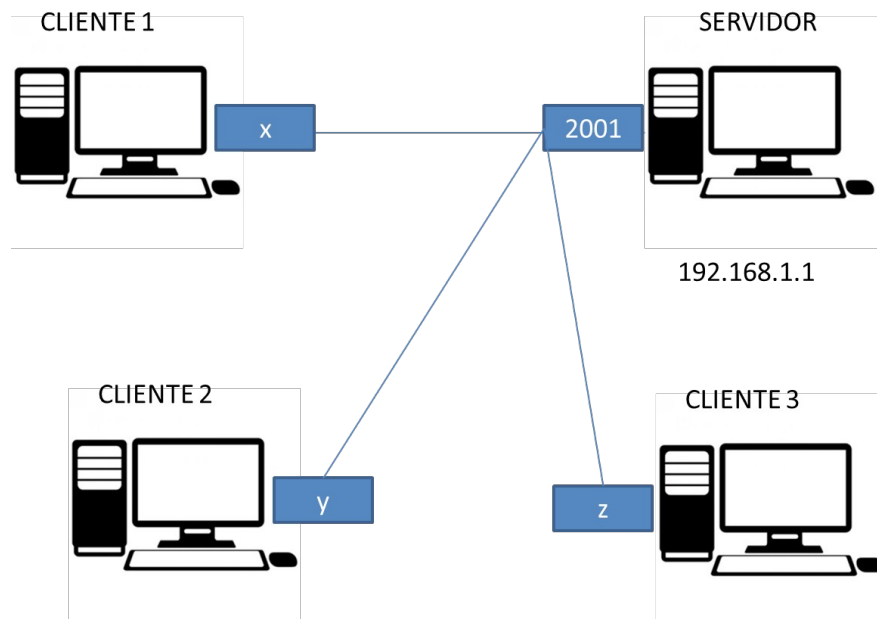


ACTIVIDADES

Se desea realizar una aplicación en red usando el protocolo UDP para la comunicación entre varias máquinas con el servidor

**El funcionamiento del programa es el siguiente.**

- Para cada cliente:
 - o El cliente envía un paquete al servidor
 - o El servidor devuelve la fecha al cliente mediante la llamada Date()
- Si el servidor detecta que no ha habido ninguna conexión en 15 segundos cerrará el socket y finalizará el programa

Sugerencia:

Para implementar el tiempo de espera de 15 segundos máximo puedes utilizar el método `setSoTimeout` de la clase `socket` que lanza una excepción **SocketTimeoutException** que es un subtipo de **SocketException**

```
public void setSoTimeout(int timeout)
    throws SocketException
```

[https://docs.oracle.com/javase/7/docs/api/java/net/DatagramSocket.html#setSoTimeout\(int\)](https://docs.oracle.com/javase/7/docs/api/java/net/DatagramSocket.html#setSoTimeout(int))

Una vez realizado el programa, completa las siguientes sentencias de forma que se consiga lo especificado en cada una de las siguientes cuestiones:

Código en la máquina CLIENTE

Ejercicio 1.

Creación de un socket para el envío de datagramas utilizando CUALQUIER PUERTO.

Ejercicio 2.

Creación de un datagrama para el envío de un datagrama a la máquina 192.168.1.1 al puerto 2001.

Ejercicio 3.

Envío del datagrama

Ejercicio 4.

Recepción del datagrama y presentación de su contenido por pantalla (sabemos que el datagrama contiene una cadena de texto)

Ejercicio 5.

Cerrar el socket

Código en la máquina 192.168.1.1

Ejercicio 6.

Creación de un socket para la recepción de datagramas a través del puerto 2001.

Ejercicio 7.

Espera del socket de 15 segundos.

Ejercicio 8.

Creación de un datagrama para la recepción de información usando un buffer de tamaño 1024.

Ejercicio 9.

Recepción del datagrama

Ejercicio 10.

Creación de un datagrama para el envío de un datagrama a la máquina que envió el paquete. El datagrama contendrá la fecha

Ejercicio 11.

Envío del datagrama

Ejercicio 12.

Cerrar el socket