

Trabajo Practico Final - Redes

Crear un modelo cliente - servidor utilizando TCP/IP Sockets a partir de los siguientes puntos:

1) Crear un servidor:

- a) Debe aceptar conexiones entrantes
- b) Usar el port number 3000
- c) Utilizar telnet para emular el lado del cliente, y realizar conexiones entrantes (Para desarrollar el lado del servidor)
- d) Recibir data en el TCP/IP socket (Utilizar telnet para emular dicho comportamiento)
- e) La data enviada debe ser texto y se debe mostrar en la consola tanto del lado del cliente como del servidor los mensajes enviados / recibidos.
- f) Utilizar una aplicación telnet de celular para mandar datos al server
 - i) Utilizar el hostname en vez de la IP para mandar información
- g) Verificar que el firewall tenga en inbound rules los permisos correctos para la entrada y salida de data.
- h) El servidor debe tener la funcionalidad de poder responderle al cliente.
- i) Si el usuario presiona 'X', se corta la conexión. La conexión se debe poder terminar tanto del lado del cliente como del servidor.

2) Crear un cliente:

- a) Debe poder conectarse con un servidor por medio de la IP y el puerto. Dichos datos se ingresan por consola (configurables). Puede haber más de un cliente corriendo.
- b) Manejar los errores si el usuario ingresa mal la dirección IP y el puerto, y devolver un mensaje coherente.
- c) Mostrar el estado de conexión paso a paso.
- d) El cliente debe poder tanto enviar como recibir data.

Preguntas

1. ¿Que es un puerto?

2. ¿Como estan formados los endpoints?
3. ¿Que es un socket?
4. ¿A qué capa del modelo TPC/IP pertenecen los sockets? ¿Porque?
5. ¿Cómo funciona el modelo cliente-servidor con TCP/IP Sockets?
6. ¿Cuales son las causas comunes por la que la conexión entre cliente/servidor falle?
7. Diferencias entre sockets UDP y TCP
8. Diferencia entre sync & async sockets?

Notas

- Se puede utilizar cualquier lenguaje de programación. Recomendando utilizar **JAVA**.
- La aplicación debe ser de consola. Se debe loguear toda ejecución que se realice del lado del cliente como del servidor.
- El código debe estar subido a un repositorio github de manera publica. El repositorio debe tener la información necesaria en un readme para poder clonarlo y dejar andando ambas app. Recomendando una carpeta de build con los ejecutables ya buildeos.
- El trabajo se mostrará y se defenderá frente al docente. El grupo traerá un ejemplo (en vivo) para mostrar y se le hará unas preguntas al equipo para que explique cómo lo realizó. Puede haber alguna pregunta teórica del trabajo. No hace falta que haya un documento en su entrega.
- Los grupos serán de 3 o 2 personas **únicamente**.
- **Tener en cuenta el tipo de conexión que utiliza con los sockets, ya que se pide sockets TCP/IP.**
- La entrega del trabajo es hasta el jueves 20 de Junio inclusive. Se enviará un email a lgarciavandenbosch@gmail.com, donde el subject dira "Trabajo Final de Redes" seguido del apellido de los integrantes del grupo. El body del email repetirán los nombres de los integrantes del grupo, un link al repositorio de código donde está alojado tanto el server como el client, y algunos screenshots de la app funcionando.
- **El día 21 de Junio y 24 de Junio**, se presentarán los trabajos de forma presencial y oral con el profesor. **Únicamente** se tomaran estos dos días para tomar examen del trabajo que realizaron. Si alguno, finaliza antes de la fecha de entrega el trabajo, puede pedir rendir antes de forma oral. Se llamará de manera aleatoria los días 21 y 24 de Junio.