

# Rúbrica de la práctica de *NanoFiles*

## Redes de Comunicaciones – Curso 2022/23

Es necesario conseguir un 5 y superar la entrevista para aprobar la parte práctica. Recordad que se os preguntarán detalles de la implementación y no se admiten excusas del tiempo transcurrido ni se admiten dudas a la hora de explicar todas las partes de la práctica, en lo referido a las clases que habéis tenido que modificar.

Cada ítem de la rúbrica vale 0,25 puntos de la nota final si es totalmente correcto, la mitad (0,125 puntos) si es parcialmente correcto, 0 en caso contrario.

Cada mejora implementada puede alcanzar la puntuación máxima indicada en el enunciado del proyecto.

La nota final satura en 10. Las mejoras y los aspectos sobre robustez pueden compensar fallos en otros aspectos de la práctica a discreción del profesor.

Aspectos relacionados con el diseño de los protocolos y la documentación (máx. 2,5 puntos)	Totalmente incorrecto	Parcialmente correcto	Totalmente correcto
El documento está bien estructurado (secciones) y es correcto tanto en forma como en contenido (correctamente redactado, sin faltas ortográficas o gramaticales).			
Las diferentes secciones del documento contienen una descripción suficiente del trabajo realizado tanto en las fases de diseño como de implementación, detallando aquellos aspectos de implementación más relevantes.			
Son capaces de capturar y examinar los mensajes intercambiados en Wireshark, y dichos mensajes siguen el formato estipulado en el documento.			
El protocolo de comunicación con el directorio (sobre UDP) contiene todos los mensajes necesarios para la funcionalidad requerida			
El protocolo de transferencia de ficheros (sobre TCP) contiene todos los mensajes necesarios para la funcionalidad requerida			
Los mensajes están correctamente descritos, de manera que no dan lugar a dudas sobre su formato, y hay ejemplos de su utilización.			
En todos los autómatas, la descripción de las transiciones se han usado los mismos nombres que aparecen en los códigos de operación de los mensajes, y se consideran todos los mensajes diseñados.			
El autómata del cliente nanoFiles (UDP) es correcto y corresponde con la implementación, sirviendo como fiel referencia del intercambio de mensajes realizado entre los procesos.			
El autómata del directorio (UDP) es correcto y corresponde con la implementación.			
Los autómatas del servidor de ficheros y la parte del cliente de ficheros (TCP) es correcto y corresponde con la implementación.			
Aspectos relacionados con el directorio (máx. 2 puntos)	Totalmente incorrecto	Parcialmente correcto	Totalmente correcto
La comunicación es posible incluso cuando se producen pérdidas de datagramas (simulados por el parámetro -loss)			
La comunicación con el directorio respeta tanto el formato de los mensajes (DirMessage) como el autómatas diseñado (estados en NFController).			
Están implementados los métodos necesarios en la clase DirMessage para construir y procesar mensajes.			
Se lleva a cabo el login con el servidor de directorio y se muestra el número de servidores disponibles.			
Se registra en el directorio el nick proporcionado por el cliente tras comprobar que no está duplicado. El cliente informa sobre ello por pantalla.			
Se solicita al directorio la lista de nicks registrados (userlist), se obtiene dicha lista y se imprime por pantalla.			
La lista de nicks registrados se envía de manera estructurada, de forma que resulta posible acceder a los elementos que la componen.			
Se da de baja el nick en el directorio cuando se sale del programa (quit), de forma que el nick vuelve a estar disponible.			
Aspectos relacionados con la transferencia de ficheros (máx. 1,5 puntos)	Totalmente incorrecto	Parcialmente correcto	Totalmente correcto
La implementación del protocolo de comunicación entre pares (sobre TCP) respeta tanto el formato como los autómatas diseñados. Los mensajes que contienen datos de fichero están correctamente formateados como texto. Está implementada la clase PeerMessage, con sus atributos, constructores y los métodos fromString y toEncodedString según la especificación dada			
Se lanza un servidor de ficheros en primer plano en un puerto dado, que atiende conexiones de clientes secuencialmente			
Se accede al "modo browser", para lo cual el cliente se conecta a un servidor de ficheros dado por su IP y puerto (IP:port)			
Una vez en modo browser, es posible descargar varios ficheros (uno tras otro), en una misma conexión con un servidor. También es posible salir del "browser" sin descargar nada.			
El protocolo de transferencia de ficheros está diseñado para intercambiar ficheros de cualquier tamaño mediante mensajes textuales. Los datos de un fichero se envían respetando la limitación de readUTF/writeUTF			
Se puede salir del modo browser (cerrar la conexión con el servidor de ficheros) y volver a acceder a dicho modo con el mismo u otro servidor.			
Mejoras adicionales			Puntuación
El servidor en primer plano se puede parar mediante fgstop			
Al arrancar el servidor de ficheros, se envía al directorio un mensaje con los metadatos de los ficheros compartidos			
Es posible obtener el listado global de todos los ficheros compartidos por todos los servidores			
Se accede al "modo browser" por nick: se averigua la IP y el puerto del servidor de ficheros a partir de su nick			
Al arrancar el servidor de ficheros, se envía al directorio un mensaje para darlo de alta como servidor, indicando su nick y puerto en el que escucha			
Mantener actualizados ficheros y servidores: Se da de baja al servidor y sus ficheros en el directorio al salir (quit) o detener el servidor (fgstop)			
Una vez conectado al servidor de ficheros, se puede obtener la lista de ficheros disponibles. Esta consulta se puede realizar múltiples veces.			
Servidor de ficheros en segundo plano			
Servidor de ficheros multi-hilo			
Ampliar filelist con servidores para cada fichero			
Ampliar userlist con información sobre servidores			