# **Servidor de**

# **transferencia**

# **de archivos.**

Texto

Descripción generada automáticamente

Javier Verdú Sánchez [javier.verdu@edu.upct.es](mailto:javier.verdu@edu.upct.es)

Ángel Madrid Morales [angel.madrid@edu.upct.es](mailto:angel.madrid@edu.upct.es)

### **ÍNDICE**

1. Instrucciones para ejecutar la aplicación.
2. Descripción de la implementación.
3. Anexo.

### Instrucciones para ejecutar la aplicación.

En primer lugar, suponemos que trabajamos con un modelo cliente servidor donde habrá una carpeta para el servidor (puede tener carpetas dentro) y una o varias carpetas en el cliente.  
  
Paso 1: Cambiar ruta del código de la variable archivoServer que será la ruta del servidor (esto se hace en el cliente).

Paso 2: Ejecutar la clase StartServer que inicia el registro RMI y el servidor.

Paso 3: Ejecutar la clase FTPClient, que mostrará por pantalla las acciones que puede realizar en cliente.

1. Descripción de la implementación.

Para la implementación del servicio hemos creado dos clases llamadas FileInfo y FileList. La primera de ellas nos sirve para que cada objeto de dicha clase tenga asociado el nombre del archivo que representa y de él se pueda extraer su tamaño en bytes. La clase FileList, nos permite listar objetos del tipo FileInfo en dos ArrayList, uno que contendrá los nombres de las carpetas contenidas en la carpeta actual, y otro que contendrá los nombres de los archivos de la carpeta actual.

Al estar usando Java RMI, todos los métodos que el servidor implemente deberán estar declarados en la interfaz remota, para que así el cliente pueda hacer uso de dichos métodos.

El método listFiles recibe como argumento un String que será la ruta de donde listaremos los archivos y carpetas mediante un objeto FileList.

El método ImprimemeMiLista recibe como argumento un objeto FileList que contendrá los archivos y carpetas de la ruta, que devolverá un String con todos los archivos y carpetas, y se imprimirán por pantalla.

El método CreateFolder recibe como argumentos la ruta donde queremos crear la carpeta o el archivo, el nombre que le queremos poner y si queremos que sea una carpeta lo que creamos tendremos que escribir por consola “si”. De lo contrario creará un archivo.

El método DeleteFolder recibe como argumentos la ruta donde queremos eliminar la carpeta y el nombre de la carpeta que queremos eliminar. Sin embargo, desde el cliente te preguntará si lo que queremos eliminar es una carpeta, en ese caso escribiremos “si” por consola, en caso contrario estaremos eliminando un archivo con el nombre que recibe, en la ruta que recibe.

El método downloadFile devuelve bloques de bytes mediante un array y recibe como argumentos el nombre del archivo que queremos descargar, la lista de los archivos disponibles para ver que exista, el offset y el resto. El offset es el byte desde el que empezaremos a leer del archivo y copiar en el array que devolveremos. El resto es el tamaño del último paquete ya que posiblemente no llene el buffer como todos los anteriores enviados.

El método subirArchivo recibe como argumentos el nombre del archivo que queremos subir al servidor y la lista de los archivos disponibles en el cliente para subir.

Respecto a la clase cliente se conecta al servidor FTP remoto y proporciona al usuario un menú de opciones para realizar diversas operaciones. Utiliza RMI para establecer una conexión con el servidor remoto identificado por el nombre "FTPInterfaz" en la dirección "rmi://localhost:5670". Donde deberemos sustituir localhost por la IP del servidor al que nos estamos conectando.

Primero se obtienen listas de archivos en el servidor y en el cliente utilizando el objeto remoto. Después se mostrará un menú de opciones para el usuario mediante un bucle infinito permitiéndole elegir entre ver archivos, descargar y subir archivos, gestionar carpetas, entre otras opciones (todas declaradas en la interfaz remota).

La opción de descarga calcula la cantidad de paquetes necesarios para la descarga y realiza la transferencia de archivos desde el servidor al cliente.

La opción de subida permite al usuario cargar archivos al servidor desde el cliente.

La opción de gestión de carpetas y archivos permite al usuario crear y eliminar carpetas o archivos tanto en el servidor como en el cliente.