

# Programación de Servicios y Procesos

## Programación Sockets

### Práctica 3.02

Vamos a crear un mini servicio de transferencia de ficheros.

#### ***Lado de servidor***

La aplicación se lanzará por la clase `psp.ud03.practica02.MainFileServerApp`

El servidor puede recibir, mediante la línea de comandos, el puerto en el que escuchará peticiones. Si no se especifica usará el puerto por defecto 4321.

Asimismo deberá funcionar en modo multihilo de forma que pueda servir a varios clientes de forma simultánea.

#### ***Lado de cliente***

La aplicación se lanzará por la clase `psp.ud03.practica02.MainFileClientApp`

Puede recibir, mediante la línea de comandos la dirección IP o nombre del servidor como primer parámetros y el puerto del servidor como segundo parámetro. Si no se reciben se emplean los valores por defecto `localhost` para el nombre del servidor y 4321 para el puerto.

La aplicación, usando la consola, pide al usuario la ruta de un fichero en el servidor. Cuando se introduce la ruta conecta con el servidor de ficheros y le solicita el mismo, guardándolo en la carpeta por defecto con el mismo nombre que tuviera en la ruta, sobrescribiendo cualquiera con dicho nombre si lo hubiera. Si el archivo no existe en el servidor se debe mostrar un error.

#### **Protocolo de comunicaciones**

El protocolo sigue un patrón simple solicitud - respuesta.

El tipo de conexión (TCP o UDP) deberás decidirlo tú analizando los requerimientos del resto del protocolo que veremos a continuación.

El cliente conecta, envía una solicitud, recibe una respuesta y se cierra la conexión en ambos extremos.

Las peticiones tienen la forma:

`ruta<SALTOLINEA>`

donde `ruta` es la ruta a un archivo y `<SALTOLINEA>` es un salto de línea (caracteres `\n\r`).

La respuesta puede ser una de:

OK\n\rDATOS

o

KO\n\r

indicando OK que se ha encontrado y se puede acceder al archivo y a continuación van los datos del mismo en formato binario (bytes) o KO que no se puede acceder al archivo por cualquier razón.

En ambos casos la comunicación se cierra en cuanto se termina el mensaje.