



Puede utilizar como máximo tres páginas adicionales por ambas caras. Cada 10 errores ortográficos restan un punto a la nota total.

Apellidos: _____ Nombre: _____ Grupo: _____

1. (10p) Se trata de construir una pequeña aplicación P2P (Peer To Peer) formada por dos tipos de programas: servidores y clientes.

El servidor realiza las siguientes funciones:

- Pone a disposición de la red p2p los ficheros que encuentre en el directorio desde el que se ejecuta la aplicación.
- Mantiene una lista de todos los servidores amigos que haya en la red. Cada servidor es responsable de actualizar dicha lista periódicamente. Por tanto, el servidor actúa como cliente cuando pregunta a sus colegas por la aparición de nuevos servidores.
- Para poder contactar con el resto de la red p2p, el servidor necesita que, al ser invocado, se le indique un servidor que ya forme parte de la red. Si no se le indica, el servidor será el primero de una nueva red.

El servidor atiende a 3 comandos diferentes:

- **peers:** Devuelve la lista de servidores amigos que el servidor conozca en ese momento. La lista está formada por líneas de texto de la forma IP:puerto. Al final de la lista debe aparecer una línea adicional con el texto 'END'.
- **files:** Devuelve una lista con los nombres de los ficheros que comparte ese servidor, uno por línea. Esta lista también debe acabar con 'END'.
- **get <fname>:** Devuelve el fichero con nombre *fname*.

La sintaxis de llamada al servidor es:

```
p2pserv <port> <peer_ip> <peer_port>
```

donde:

- **port:** Es el puerto en el que escuchará el servidor.
- **peer_ip y peer_port:** Es un socket en el que hay otro servidor de la red p2p.

Para realizar su función, el cliente debe:

- Contactar con un servidor.
- Pedirle la lista de servidores de la red p2p.
- Buscar el fichero solicitado preguntando a todos los servidores de la red.
- Pedir el fichero al servidor que lo tenga.

La sintaxis del cliente es:

```
p2pclient <peer_ip> <peer_port> <filename>
```

Se pide:

- Realizar un análisis adecuado del problema.
- Diseñar una solución para acometer el objetivo propuesto.
- Tanto para el cliente como para el servidor, realizar una especificación en pseudocódigo, con especial énfasis en la parte de comunicaciones.
- Tanto para el cliente como para el servidor, realizar un esbozo de implementación en lenguaje Python.