6º) Se pide la escritura de un sistema clinte-servidor de *backup* remoto de archivos **rbackup**, de forma que sólo se actualicen en el servidor las copias de los archivos cuyos contenidos hayan cambiado en el cliente. Para ello se usará una función resumen (*hash*), que el servidor y el cliente usarán para comprobar si los archivos a copiar se mantienen iguales o no y, por tanto, si hay que transferirlos (los archivos que no existan en el servidor, se crearán).
El sistema constará de un **cliente *rbackup.py*** y de un **servidor *rbackupd.py*** siendo el uso de estos programas:

***rbackup.py* [-h | -rd]** camino al archivo o directorio

-h	**ayuda**
-r	**recursivo**
-d	**borra los archivos en el servidor que no están en el cliente**

rbackupd.py camino al directorio de *backups*

El proyecto se debe escribir en **Python**. El desarrollo debería seguir estos pasos:

- Diseñar y especificar el protocolo [sintaxis, semántica y temporización] que seguirán servidor y clientes para comunicarse (en el LEEME). Entre otras cosas hay que establecer cómo pasar la lista de archivos junto con sus resúmenes. En este punto, hay que decidir hasta qué punto nos conviene crear un nuevo protocolo o adaptar uno ya existente. De manera que este paso conecta con el siguiente:
- Separar muy claramente la lógica del *backup* de la de los protocolos de red. Para esto último se puede hacer uso de cualquier librería estándar de Python (teniendo en cuenta que a más bajo nivel, más trabajo de diseño de protocolo; y que si podemos adaptarnos a un protocolo ya implementado, nos ahorramos la mayor parte del dichoso diseño):
  - *socket*
  - en el caso del servidor: *SocketServer* o cualquier otra que pueda facilitar el diseño del protocolo (capa de aplicación, como p. ej. *BaseHTTPServer*)
  - algo parecido sucede en el caso del cliente
- Escribir una primera versión del servidor que sólo admita la conexión de un único cliente a la vez.
- Escribir el cliente.
- Escribir una segunda versión del servidor que permita atender a múltiples clientes de forma simultánea. Debe tenerse en cuenta qué hacer para almacenar las copias de clientes en distintos nodos (con distintas IP). ¿Qué sucede si dos o más clientes en un mismo nodo intenten hacer *backup* de los mismos archivos?
- Un estudio de problemas, mejoras y extensiones.

*Para la entrega se creará el proyecto **rbackup.py** en el directorio $HOME/repositorio que contendrá, al menos, tres ficheros: un **LEEME**, que explicará el proyecto, versiones, enfoque, uso, problemas, mejoras y extensiones posibles, un **rbackup.py** y un **rbackupd.py**. Las versiones que se vayan desarrollando de estos ficheros se controlarán con **Mercurial-Hg**.*