

No hay mucho que explicar, pero básicamente, la división en paquetes ha sido en función de lo que es para el servidor, lo que es común y lo que representa la clase.

- *servidoregorilla* : es donde se encuentran todas las clases correspondientes únicamente al servidor. En este nivel está sólo el Main y derivados.
 - *datos* : están las clases que tiene los datos que sólo debe manejar el servidor. Aquí sólo está la clase ArchivoClientes, la cual es un HashTable indexado por MD5 de cada fichero (que supuestamente es único) relacionado con un Array de ConexionClientes, para saber cuántos clientes tiene ese fichero.
 - *tareas* : están las tareas principales que hace el servidor, las cuales son todas son hilos (thread's).
 - *servidor* : está la clase que representa al servidor.
- *paquete* : están a este nivel, junto al de *servidoregorilla*, porque son clases comunes tanto para el servidor como para cliente. Son clases que se van a usar como intercambio de información entre el servidor y el cliente, por lo que tienen en común en que la mayoría son serializables.
- *datos* : contiene la clase referente a datos de los ficheros que va a compartir el cliente. Al tener una única clase y ser serializable, tal vez se meta en *paquete* y quitemos este.
- *red* : tiene la clase que controla las comunicación entre socket's y que cliente y servidor usan para establecer comunicación.

NOTAS:

- No se han representado las dependencias de las clases que heredan de Thread (no se pueden pasar parámetros a run()), ya que es un diseño independiente del lenguaje de programación.
- Es un diseño básico, y aunque tiene alguna influencia de algún patrón de diseño no lo comento de momento, para futuras iteraciones.