La parte del Cliente es una aplicación Android que incluye la vista y el controlador. Si empezamos explicando el controlador podemos decir que su función principal es gestionar las peticiones que hace la vista al modelo. El controlador está dividido en tres partes que se corresponden con el paquete **web**, el paquete **delegates** y el paquete **controllers**.

El paquete **web** es el encargado de realizar las peticiones HTTP a un más bajo nivel. Su función es mandar las peticiones GET, POST, PUT y DELETE. Con este fin, usamos la librería de Google Volley que gestiona el acceso a las comunicaciones de red a bajo nivel. Solamente es necesario saber que las peticiones se van almacenando en una cola y Volley se encargará de mandarlas. Necesitamos por tanto una clase que se ocupe de gestionar la cola y de la subida de imágenes que será **WebRequestQueue**. (Es una especie de Singleton). La clase encargada de construir las peticiones y pedirle a **WebRequestQueue** que las añada a la cola es **WebServiceClient**. Esta clase tiene los métodos putRequest(…), postRequest(…), deleteRequest(…) y getRequest(…) para las peticiones y el método postBytes(…) para las imágenes. Estos métodos aceptan 2 callbacks, que son funciones pasadas como argumento que se ejecutan cuando termina la tarea asíncrona. Uno es en caso de éxito y otro en caso de error. Estos callbacks implementan las interfaces **ErrorListenener**, **ResponseListener**, **SuccessListener** que completan el paquete **web**.

El paquete delegates es una abstracción a un niel intermedio de las peticiones HTTP al servidor. Hay un delegate por cada uno de los servicios que tiene la aplicación y es una contrapartida a cada WebServiceBroker del servidor. Aquí cada delegate es el encargado de mandar las peticiones a alto nivel correspondientes a un determinado servicio desde el controlador, y en el servidor cada WebServiceBroker es el encargado de ejecutar a alto nivel las peticiones HTTP. Todo los delegates tienen un atributo privado **WebServiceClient** que va a llamar a los métodos putRequest(…), postRequest(…), deleteRequest(…) y getRequest(…). Los métodos que tienen los delegates reciben como parámetro los dos callbacks e información adicional en forma de un objeto transfer.

El paquete controllers tiene una única clase Controller que es una abstracción a más alto nivel que los delegates. Esta es la clase que usa la vista para comunicarse con el servidor. Tiene un objeto de cada clase delegate e implementa el patrón Singleton. Si necesitamos añadir más servicios solamente sería necesario añadir otro objeto delegate. Sus funciones no reciben como argumentos objetos tranfer, sino que reciben Strings o Int y los transforman en objetos transfer para llamar a los delegates. Por tanto, estamos ocultando a la vista el uso de los objetos transfer.