La capa correspondiente a la vista dentro de la aplicación ha sido realizada con el programa Android Studio. La vista está basada fundamentalmente en *Activities* donde cada *Activity* representa una pantalla de la aplicación y tiene una funcionalidad asociada. La navegación entre las distintas pantallas de la aplicación se realizará cambiando de Activity con el método startActivity(new Intent(this, FavorsActivity.class)). Cada *Activity* tendrá la funcionalidad para adquirir el contenido que mostrará. Para dicha representación hemos utilizado las facilidades que nos otorga Android Studio, entre ellas el uso de Material Design y la posibilidad de añadir Adapters para personalizar los distintos listados de favores, mensajes y regalos.

En el *MainActivity* se crea un *LoginService* que será el que se use mientras la aplicación esté abierta. En dicho *Activity*, se comprueba si hay un *Token* que guardamos en memoria persistente. En caso afirmativo, éste es envidado al servidor a través de un TransferObject de la clase TransferToken. En el servidor se comprueba si el *Token* es válido, y es entonces cuando el servidor devuelve a la vista un nombre de usuario. Seguidamente, se pasa a la pantalla de inicio que es la correspondiente a *FavorsActivity*. Si no se encuentra el *Token* entonces el usuario es dirigido al *SignInActivity* donde, tras introducir su nombre y su contraseña, se lanzará una petición al servidor que le devolverá a la vista el nombre de usuario correspondiente.

Las *Activities* que corresponden al resto de pantallas de la aplicación están compuestas por distintos componentes visuales que permiten la interacción del usuario, tanto para obtener información como para cambiar de *Activity*. De entre todos estos componentes, los más destacables son los *RecyclerViews*. Contamos con un total seis *RecyclerViews* que se corresponden con: los favores que otros usuarios piden (en *FavorsActivity*), mis favores a realizar (en *MyFavorsActivity*), mis favores pedidos (en *MyFavorsActivity*), los chats abiertos (en *MessageActivity*), los mensajes de un chat (en *ChatPersonActivity*) y los regalos (en *GiftsActivity*). Los *RecyclerViews* van a funcionar como paneles que muestran al usuario la información de las distintas listas de objetos. Estos objetos *Favor*, *Gift*, *Persona* y *Mensaje* serán construidos por la vista tras peticiones al servidor que *TransferObjects* (*TransferChat*, *TransferFavor*, *TransferGift*, *TransferMessage* y *TransferUser*). Para obtener estos *TransferObjects*, desde la vista se hacen peticiones al servidor como List<TransferFavor> lf=FavorQuerier.getAllFavors(). Este ejemplo muestra que la comunicación entre la vista y el server es siempre a través de *Queriers* en los que la vista solicita unos datos que le son devueltos en forma de *TransferObjects* y son tratados para poder ser mostrados.

La mayoría de los Activities, aquellos que podríamos considerar como pantallas de navegación principales y representan los distintos módulos de la aplicación, cuentan además con un panel inferior formado por varios componentes propios de Android Studio como *ImageButton* o *TextView* que permiten navegar por la aplicación yendo de un Activity a otro. Cabe destacar que en *FavorsActivity* el usuario tiene disponible una opción de búsqueda por filtro. El usuario podrá buscar un favor por nombre, por distancia o por número de grollies y se mostrará dicha información ordenada en el *RecyclerView* del *FavorsActivity*. Internamente **PEDRO COMPLETA ESTO QUE NO SE LO QUE HACE INTERNAMENTE.**

Una de las ventajas del uso del patrón Modelo-Vista-Controlador en nuestra aplicación es que conseguimos un mayor nivel de seguridad. La vista obtiene todos sus datos a través de *Queriers* al servidor y la comunicación entre vista y servidor es siempre mediante *TransferObjects*, que permiten el intercambio bidireccional de información entre ambos. Dichas peticiones son siempre comprobadas por el servidor lo que aporta un mayor grado de seguridad.

**PEDRO HABLA DE LA PERSISTENCIA QUE NO TENGO NI IDEA DE LO QUE ES.**

En cuanto al *ShopActivity*, que es el encargado de los pagos, usaremos el sistema de pagos de GooglePlay **PEDRO EXPLICA LO DE LOS PAGOS**

**PEDRO HABLAR DE SINGLETON PORQUE LO DIJISTE Y NO SE EXACTAMENTE DONDE LO USAMOS EN LA VISTA**