



En mi sistema me he centrado en el uso del patrón de diseño factory ya que según mi punto de vista es el mejor para preparar la aplicación para posibles ampliaciones sin necesidad de tocar el código.

He elegido este patrón de diseño debido a que posee muchas cosas en común unos elementos con otros, como es el caso de los diálogos por tanto he elegido la interface DialogFragment y de ahí he colgado los demás para que en un futuro podamos añadirle sin tocar nada más en las clases.

Por otro lado la clase de “**opciones de juego**” también le aplicado este patrón debido a que posee elementos como “**Media**” que por ahora solo posee el **Sonido** y podríamos en un futuro añadirle por ejemplo **Videos**. Lo mismo ocurre en Opciones Generales, ahora solo podemos cambiar el número de preguntas, pero podríamos cambiar en un futuro las Opciones de apariencia del juego por ejemplo. En mi modesta opinión es el que más se adapta al sistema que queríamos desarrollar también podría haber elegido singleton pero decidí desistir en seguir este patrón por la posibilidad de poder iniciar en un futuro el jugador más de una instancia a la vez jugando partidas por internet.

Otra de las cosas importantes que he realizado para mejorar el manejo de la base de datos es la creación de una clase adaptador que funciona de abstracción sobre la Base de Datos consiguiendo que las demás clases no necesiten saber cómo manejar la base de datos sino que simplemente llaman al objeto adaptador y él se encarga del resto.

En lo que respecta al Patrón arquitectónico me he limitado a utilizar el Modelo Vista Controlador de Android debido a que es el patrón arquitectónico que mejor funciona con android, dividimos las vistas en los archivos xml, el Controlador las distintas clases y el modelo de datos es nuestra base de datos que lo único que tenemos que hacer es modificar un archivo de texto y el sistema nos cargará todos los datos.