<https://www.youtube.com/watch?v=9Ya3yieB8aI>

<https://www.youtube.com/watch?v=6hIvPCkwMds>

<https://www.youtube.com/watch?v=UrIFdZZwo6g>

<https://www.youtube.com/watch?v=j47CSoJ_Hc4>

Cada patrón responde a una necesidad diferente, no hay un patrón de diseño mejor que otro, sino que responden a los diferentes requerimientos.

ViewModel Android

* Ayuda a gestionar problemas del ciclo de vida.
* Aplicaciones más robustas, mantenibles y testeables.
* Las activity tienen un ciclo de vida y los componentes view model otro aparte.

Capas

* **View o UI**: interfaz de usuario, más la activity. Puede ser un observer, en ocasiones deberá tener más lógica para interpretar lo que viene de viewmodel y en esos casos será necesario usar espresso para los test, pero en otros casos se puede entregar toda la lógica al viewmodel y hacer mas simples las vistas lo que hace que nuestro viewmodel no sea tan reutilizable
* **viewmodel:** modelo para la vista, obtiene datos del modelo y provee a la vista de los datos que necesita o requiere. No depende de las vistas
* **clase model:** proveedor de datos

Lifecycle es una herramienta útil para sincronizar viewmodel y view