

- **Modelo:** representa la capa de datos y la lógica de negocio sin haber diferencias importantes con los otros patrones mencionados.
- **Vista:** se encarga de mostrar la información directamente al usuario y es la capa con la que éste interacciona directamente a través de eventos, permitiendo la manipulación de la aplicación y sus datos.
- **View Model** o Modelo de Vista: sirve de intermediario entre el modelo y la vista, proporcionando datos a esta última permitiéndole modificarlos. Es ajena a la vista, es decir, no contiene referencias a ésta y, por lo tanto, puede ser reutilizable por varias de ellas. Por tanto, la comunicación con ella se realiza por medio de enlaces de datos o bindings, que, en nuestro caso, será LiveData. Además, una vista puede necesitar de la información proporcionada por más de un View Model.

**Vídeo:**

**<https://youtu.be/klgqCxo-6tM>**

**CLIENTE IOS Eh-Ho**

**CLIENTE ANDROID Eh-Ho**

COREDATA  
USER DEFAULTS

ROOM

Framework de Swift URLSession  
CODABLE

VOLLEY (Topics/Posts)  
RETROFIT (Login)

Patrón de diseño MVVM

MVVM sólo en el módulo de Login  
Inyección de dependencias con Dagger  
(PostsRepo)

Coroutines (login)