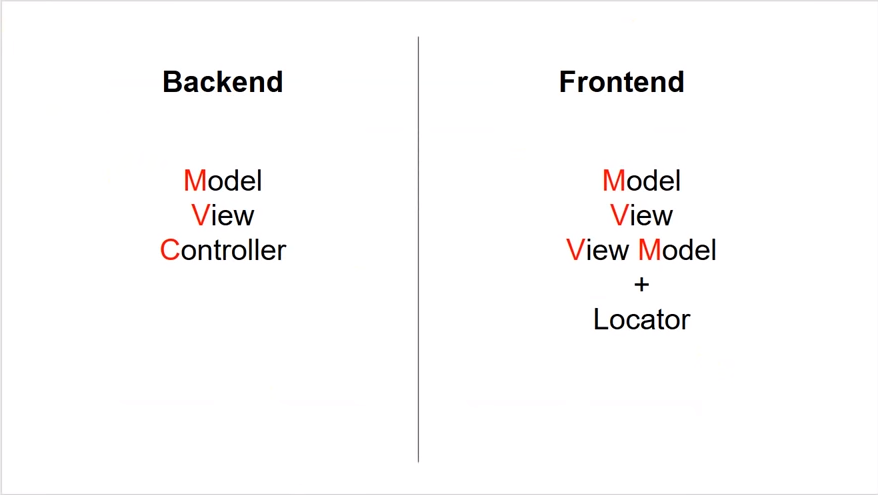


Android, iOS y UWP utilizan el proyecto “Sales”. Hay una “dependencia” de Sales.

No hay ninguna flecha apuntando a Common, por lo tanto Common no depende de nadie.

Domain usa a Common.

Backend usa a Domain y a Common



Para el Backend, nos va bien utilizar el patrón MVC, mientras que para el Frontend, nos va mejor utilizar el patrón MVVM.

MVVM: Patrón de arquitectura basado en Bindings.

Locator: Es otro patrón de arquitectura

MVC: Conectaremos los controladores con las vistas. Pero no se conectan a traves de Bindings sinó que le pasaremos los objetos. El Controlador le pasa el modelo a la vista. El usuario interactúa con la vista. Por ejemplo cuando yo voy a modificar un producto, quién es el modelo ?, El producto.!! . El controlador que hace ?, va y busca el producto en la base de datos. Lo tiene ahí en memoria. Y se lo pasa a la vista “Edit”, el usuario le cambia el precio y cuando le dice “guardar”, vuelve al controlador. El controlador valida que cumpla con las reglas de negocio y lo vuelca a la base de datos.

MVVM: Es muy diferente. El ViewModel tiene el producto. El Viewmodel dice “Yo soy producto” y la vista le dice “yo estoy ligado con vos”. Si realizo un cambio en la vista, se debe reflejar en la ViewModel.

Empezamos por el Backend. Vamos a hacer un “Catálogo de productos”. Quién sería el “modelo” ? El Producto.!. El producto representa al objeto que vamos a manipular. Producto será una clase que utilizaremos en “todo” el proyecto. En el backend, en la api, en android, etc..

Y, como se trata de una clase que la vamos a usar en todo el proyecto, en dónde debería estar ? En “Common”.!!