VIPER & MVP

Abrahán Fernández

March 10, 2021

Objetivos de arquitecturas más avanzadas

Clean Architecture

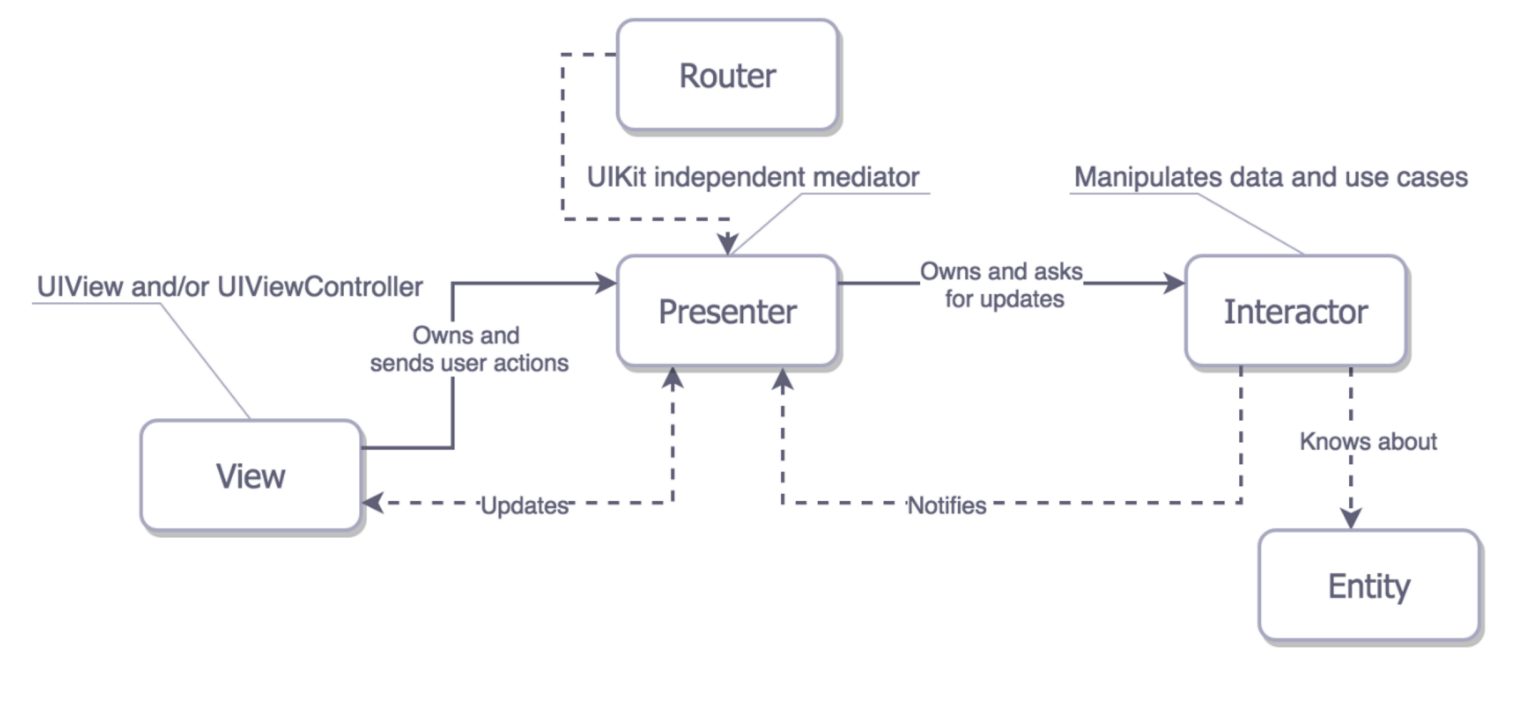
- *Testable* Que podamos desarrollar tests capaces de probar cada caso de uso por separado sin necesidad de interacción con la interfaz de usuario ni servicios externos
- Independiente de la UI Vamos a aislar la capa visual de nuestras aplicaciones

SOLID

- (S) Single Responsibility Principle
- (O) Open/Close Principle
- (L) Liskov Substitution Principle
- (I) Interface Segregation Principle
- (D) Dependency Inversion Principle
- Composición antes que herencia y programación orientada a protocolos.

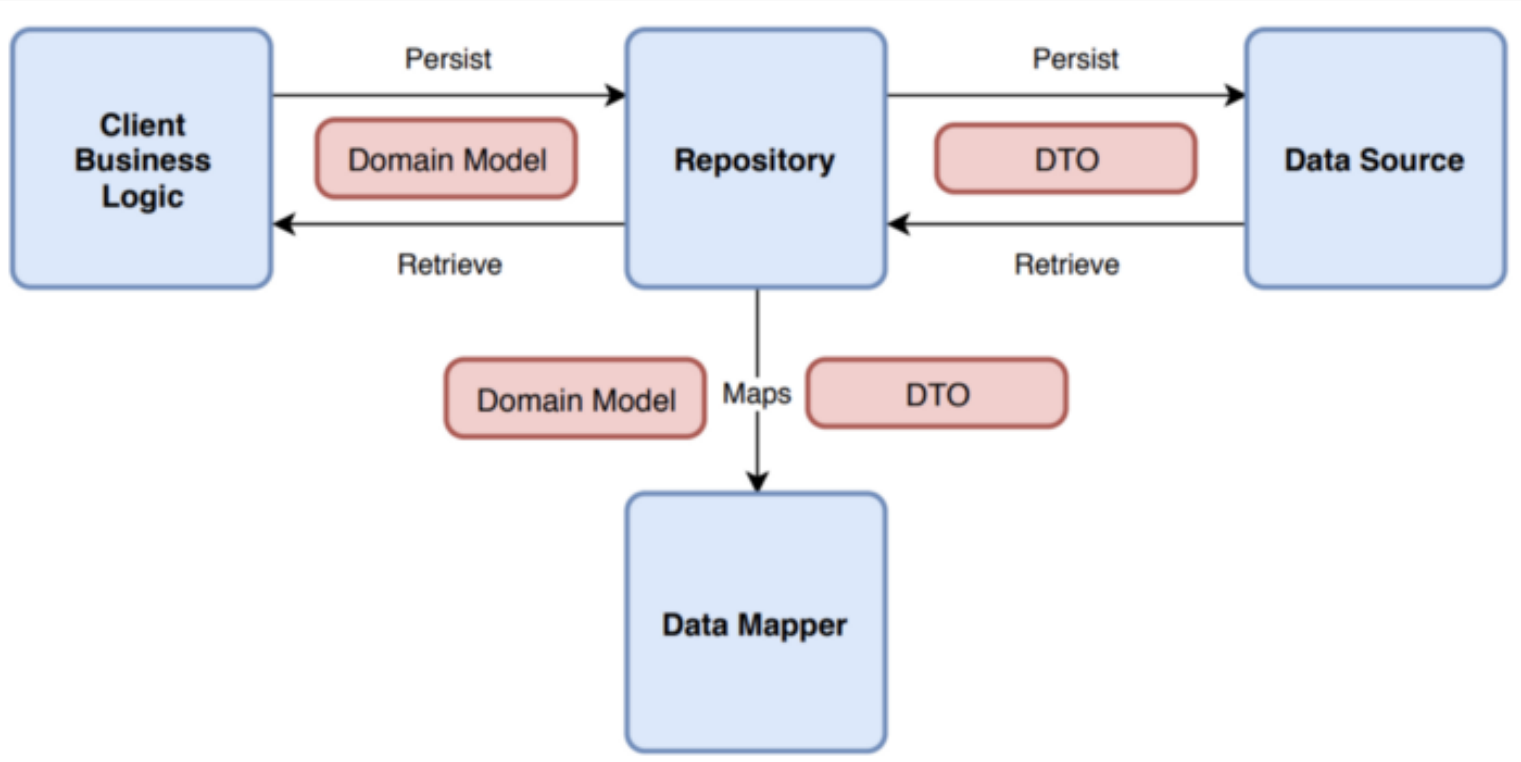
VIPER

La sigla VIPER proviene de las palabras *View*, *Interactor*, *Presenter*, *Entity* y *Routing* o *Router*



Nuestro VIPER

- Los *providers*
- El patrón *repositorio*



MVP (¿Por qué?)

- Hay proyectos que por criterios de relación coste/beneficio no son adecuados para aplicar una arquitectura VIPER.
- Menos capas y más flexibilidad implica más agilidad a la ahora de implementar cambios.
- Cualquier desarrollador puede ser productivo muy rápido sin necesidad de tener conocimientos previos de la arquitectura
- Simpre hay que pensar en como ocurrirá el mantenimiento de dicho proyecto para una decisión correcta de la arquitectura.

Ejecución hilos

Ejecución en el hilo principal vs 2 plano.

```
class Utils {
    static func dispatchToMainAsync(completion: @escapi
        if Thread.isMainThread {
            completion()
        } else {
            DispatchQueue.main.async {
                completion()
            }
        }
    }
}
```

Retención de memoria en clausuras I

Retención de memoria en clausuras II

Los pasos para decidir si es necesarios serian:

1. Es un clausura que escapa o no. Si escapa necesita weak self
2. Si no escapa puede que retrase la liberación de memoria si tarda mucho tiempo en ejecutarse, GCD, URLSessions, Timers. Dependerá si nos interesa en ese caso usar weak self o no.

Trucos

1. Realizar una referencia a lo que queremos modificar antes de la clausura y así nos libramos de usar self. Si son varias cosas las podemos agrupar en una tupla.