

PATRONES DE DISEÑO DÍA 3



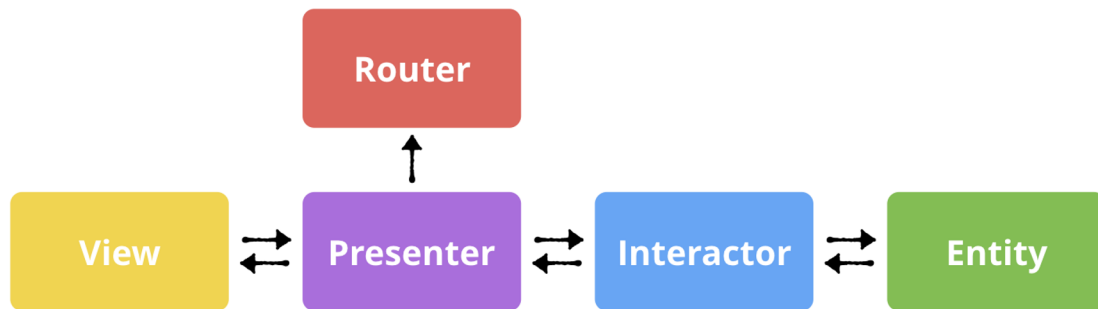
KEEPCODING
Tech School



KeepCoding®. Todos los derechos reservados

Patrones de diseño: VIPER - Intro

- Cada una de las letras en *VIPER* representa un componente de la arquitectura View, *Interactor*, *Presenter*, *Entity* y *Router*.
- Separa la lógica de visualización de la lógica del modelo de datos.
- Solo el Presenter habla con la View, y solo el Interactor habla con el Model/Entity.
- El Presenter y el Interactor se coordinan entre sí.



Patrones de diseño: VIPER - Componentes

- **View:** La responsabilidad de la vista es enviar las acciones del usuario al presentar y enseñar lo que le dice el *Presenter*.
- **Presenter:** Su responsabilidad es convertir datos desde el *Interactor* en acciones de usuario, crear una vista modelo y llevarlo hacia el *View* para mostrarlo.
- **Router:** Tiene toda la lógica de navegación en el proyecto.

Patrones de diseño: VIPER - Componentes

- **Interactor:** Esto es la columna vertebral del proyecto, ya que contiene la lógica de negocio. Su implementación es totalmente independiente de la interfaz del usuario.
- **Entity:** Contiene el modelo básico de objetos usados por el *Interactor*. Tiene parte de responsabilidades de la capa modelo en otras arquitecturas.

Patrones de diseño: VIPER - Ventajas

Ventajas de trabajar con VIPER:

- Simplifica los proyectos complejos.
- Los módulos de Viper son independientes y funcionan muy bien para equipos de gran tamaño.
- Lo hace escalable. Habilita a los desarrolladores a trabajar simultáneamente en el proyecto.
- Desacopla el código y lo vuelve más reutilizable y fácil de testearlo.
- Facilita la escritura de tests ya que la lógica de UI se separa de la lógica de negocio.

Patrones de diseño: VIPER - Ventajas

Ventajas de trabajar con VIPER:

- Crear interfaces más claras y mejor definidas, independientes de otros módulos.
- Hace que sea más fácil cambiar la manera en la que la interfaz presenta varios módulos de usuarios.
- Hace el código fuente más claro, compacto y reusable.
- Reduce el número de conflictos dentro del equipo de desarrollo.
- Hace que el codebase sea siempre parecido. Lo que hace leer el código de otras personas más rápido.

VIPER: Code Time





KEEPCODING

Tech School

Madrid | Barcelona | Bogotá

Aitor Iglesias Pubill
aiglesiaspubill@gmail.com