ANDROID:

Del diseño de la arquitectura al despliegue profesional

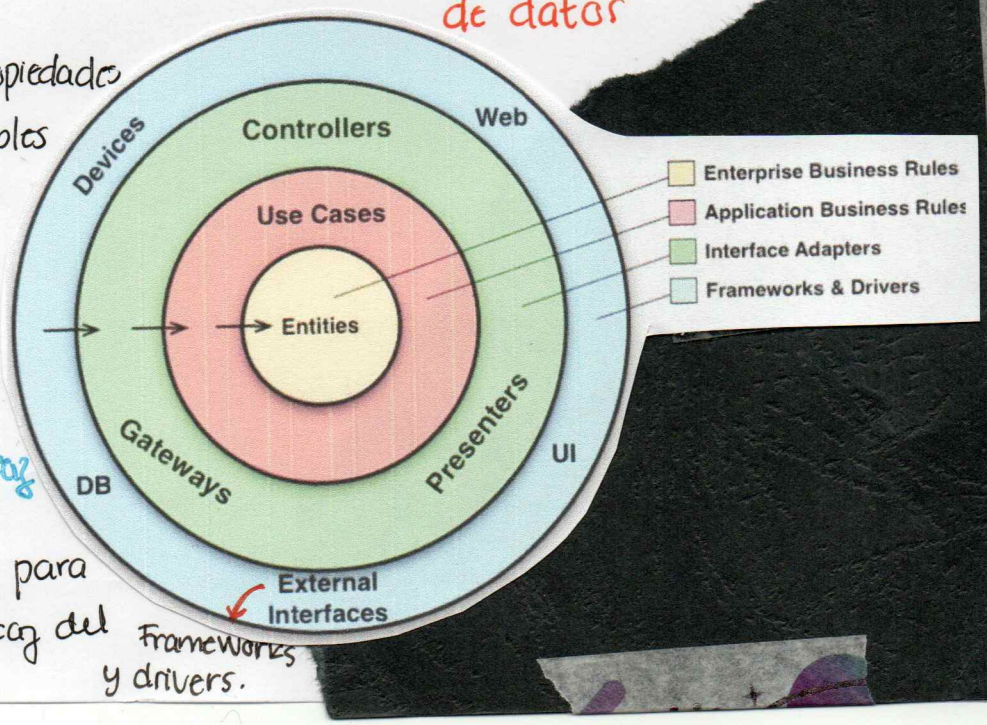
Capítulo 2:

Arquitectura CLEAN

"Arquitectura Limpia" → Robert C. Martin
- The Clean Architecture
- Clean Code

Es aquella que busca estructuras modulares, fácil lectura, limpieza del código y testabilidad.

- Independientes del framework utilizado
- Testeables
- Independientes de la interfaz gráfica
- Independientes de factores externos
- Independientes de los orígenes de datos



- **Entidades:** Poseen propiedades y métodos, son reutilizables y su estructura varia con poca frecuencia.
- **Caso de uso:** manejo del flujo de datos desde y hacia las entidades.
- **Adaptadores de interfaz:** Transforman los datos del caso de uso y entidades para la base de datos o interfaz del usuario.

ANDROID:

Del diseño de la arquitectura al despliegue profesional

Capítulo 3:

Principios SOLID

SOLID → Introducido por Robert C. Martin a principios del 2000

- **Principio de responsabilidad única (S):**
  Una clase debe tener una única funcionalidad.

- **Principio de Ser abierto y cerrado (O):**
  Nuestro código debe estar abierto para extenderse y cerrado para modificarse.

- **Principio de Sustitución de Liskov (L):**
  Si la clase A es de un subtipo de la clase B, entonces deberíamos poder reemplazar B con A sin afectar el comportamiento de nuestro programa.

- **Principio de Segregación de interfaces (I):**
  Se debe evita tener métodos de interfaces que no se implementan.

- **Principio de inversión de dependencia (D):**
  No debe existir dependencias entre los módulos.

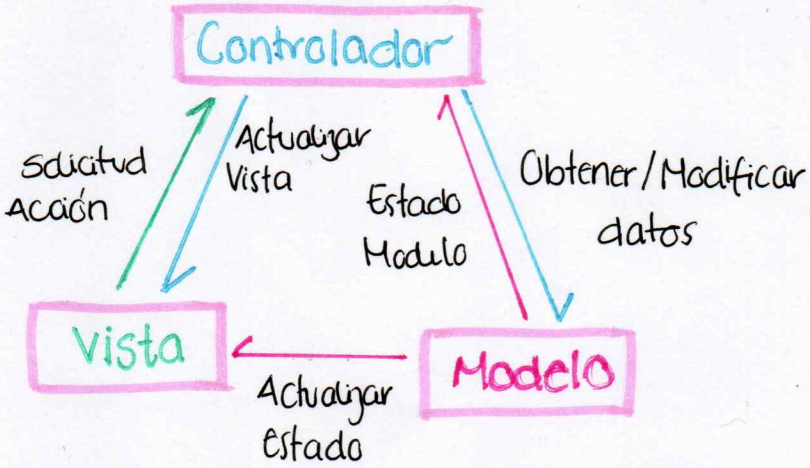
ANDROID:

Del diseño de la arquitectura al despliegue profesional.

Capitulo 4:

Patrones de diseño

- MVC



Características.

- Multiples vistas, controlador.
- Modulo sin lógica
- Controlador lógica de negocio del sistema.

- MVP

Características.

- Cada vista tiene su representador
- Lógica de negocio del sistema, modulo.
- Controlador lógica representación de la vista.

- Patrón observer: → Observables: Notificar a los subscritores sobre cambios de estado sobre estos objetos.

↓

Observadores: se subscriben a los observables y que solicitan ser avisados sobre cambios en el estado de los observables.