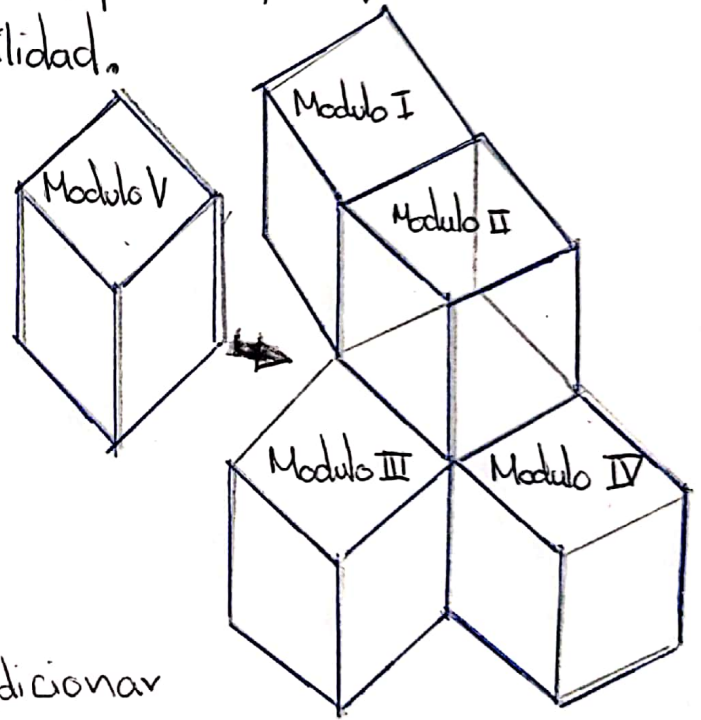


Capítulo 2:

Arquitectura Clean.

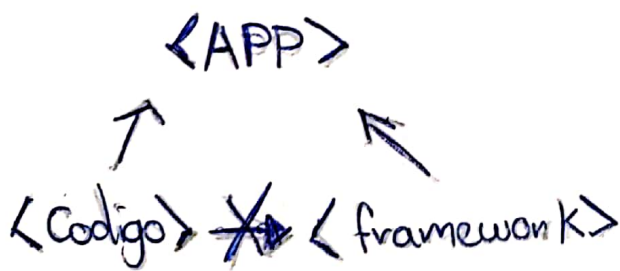
"Arquitectura Clean" o "Arquitectura Limpia" tiene su origen en un artículo escrito por Robert C. Martin, También conocido por unde BoB, titulado "the clean Archi-ecture".

Una arquitectura limpia es aquella que pretende conseguir unas estructuras modulares bien separadas, de fácil lectura, limpieza de código y testabilidad.



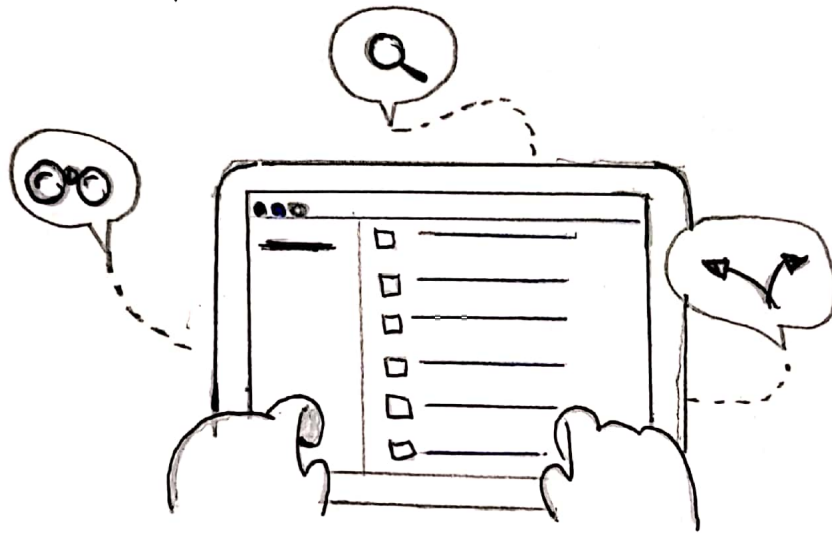
Los sistemas construidos bajo esta arquitectura han de ser:

Independiente del framework
Las librerías utilizadas no deben condicionar nuestra forma de desarrollo



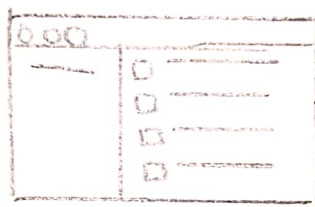
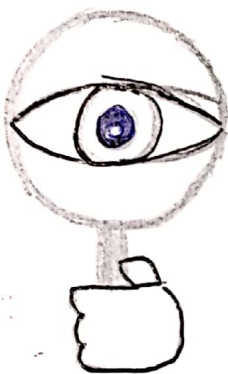
Testeables

Se ha de poder hacer test fáciles, indistintamente la GUI (interfaz Gráfica del usuario), modelo, base de datos o peticiones empleadas de la api



Independientes de la interfaz gráfica

Se debe utilizar para cambiar la GUI de acuerdo a las necesidades. Para ello se utilizarán patrones como MVP, MVVM o MVC.



X <Codigo>

Independientes del origen de datos

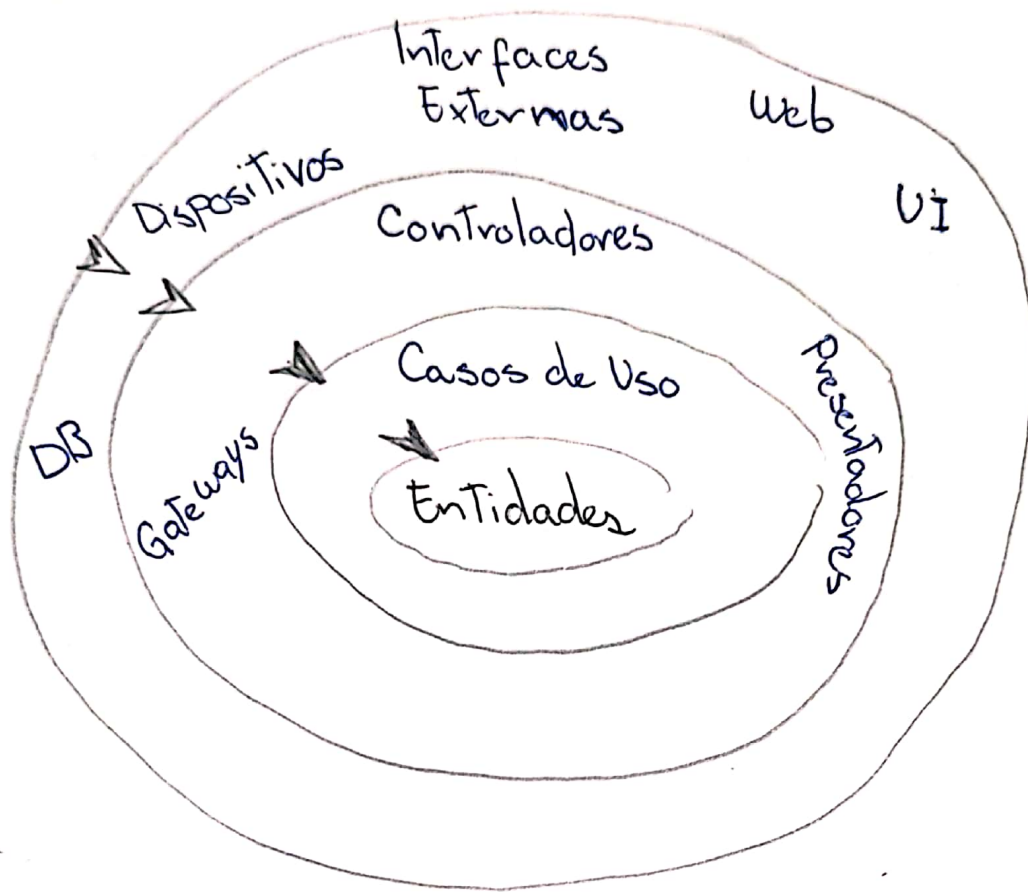
Se debe poder sustituir fácilmente nuestro origen de datos ya sea relacional o no relacional.

BDD Vieja



<APP>

En la arquitectura limpia se debe buscar que la Logica que buscamos implementar deba ser el centro de nuestro sistema.



Para atravesar estas capas se debe llevar un orden, gracias a esta separación se puede llevar una ordenada lógica fácil de entender.

Entidades

Son las interacciones que puede tener el usuario con el sistema

Adaptadores de interfaz


Se encargan de convertir los datos de acuerdo a las necesidades del sistema

Frameworks y Drivers

En esta Capa se encuentran las plataformas externas
interfaz de usuario y bases de datos. Es la capa más
externa que debe comunicarse hacia las capas inferiores



Pantallas

UI


Navegación entre
Ventanas


Controladores
de pantalla


Presentadores


Validaciones


Registro de
negocios

CASOS
de
USOS


Comunicación
con servicios web


Modelo
de
Datos


BD/
Gestión de
Cache

Entidades