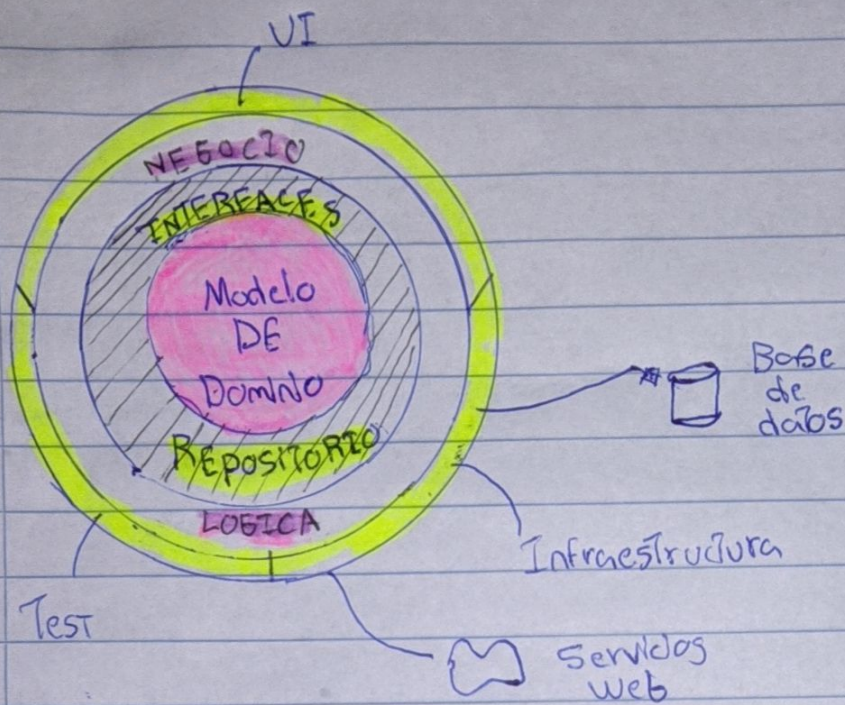


# ¿Qué es la arquitectura limpia?



Es aquella que pretende desarrollar estructuras modulares separadas que sean fáciles de leer, que muestren un código limpio y proporcione una buena Testabilidad.

Según Uncle Bob el código limpio debe regirse por los siguientes principios:

- **Independiente del framework**: Las librerías que se usen no deben condicionar el código sino que se tiene que acoplar con la estructura.
- **Testables**: La lógica de la aplicación debe ser Testable independientemente de la interfaz gráfica, modelo, base de datos o peticiones http.
- **Independientes de la interfaz gráfica**: Se debe buscar la manera de cambiar fácilmente la Interfaz grafica para ella se deben aplicar patrones de diseño de software como PUP, MVVM o MVC.



- **Independientes de los orígenes de datos:** Se debe contar con una flexibilidad en la hora de sustituir el origen de datos sin importar si se encuentra en una base de datos local, ficheros o peticiones a una api. Para asegurar este apartado se utilizan patrones de diseño como el patrón repositorio.

- **Independientes de Factores externos:** Las reglas de negocio no deben conocer nada ajeno a ellas.

Otro aspecto fundamental en el que se basa la arquitectura limpia es en la separación de responsabilidades dentro de las capas. En el interior se encuentra la lógica de negocio, luego el transporte o comunicación y en la parte más externa se encuentran las vistas, bases de datos e interfaces externas, etc.

A su vez existe un sentido para recorrer las capas, desde la capa más externa hasta el interior. Por lo tanto con esta separación la capa interior no debe saber nada de las capas exteriores. Esto significa que la lógica de negocio no debe utilizar ningún elemento de otras capas.