

1 Layered Responsibility

Señales de decadencia: problemas reales de la arquitectura por capas

1. CONNECT

- ¿Cómo se organiza el código?
- ¿Qué impacto tiene las decisiones de diseño?
- ¿Qué se entiende por **responsabilidad** única?

Objetivo: entender el principal valor de la arquitectura por capas y los problemas que se presentan.

2. CONCEPT

Ventajas de la arquitectura por capas:

- > Sencillez.
- > Organización.
- > Separación de **responsabilidades**.

Problemas típicos de una arquitectura por capas:

- › Acoplamientos.
- › Flujo de dependencias.
- › Lógica dispersa.
- › Dominio inexpressivo.
- › Pruebas frágiles o complejas.
- › Potencialmente un *Big Ball of Mud*.

3. CONCRETE PRACTICE

Partimos de un diseño de capas intencionalmente deficiente.

Corregir los problemas de **responsabilidad** que se encuentran en el código.

- › Las reglas de validación de estructura y negocio están dispersas.
 - › [] Usar DTOs de entrada para validar estructura.
 - › [] Llevar validación de valores a nivel de negocio.
 - › [] Evitar llamadas a repositorio desde el controlador.
- › Las excepciones no siguen una estructura consistente.
 - › [] Definir excepciones de negocio.
 - › [] Centralizar respuestas de error.

Objetivo: familiarizarse con el código y sus deficiencias actuales

4. CONCLUSIONS

- La arquitectura por capas tiende a degradarse con el tiempo.
 - Existen principios que permiten mejorar la arquitectura.
- ¿Qué cambiarías mañana mismo en tu código para evitar que se convierta en un 'big ball of mud'?