

Clean Architecture

Criada por Uncle Bob (Robert C. Martin), é um padrão de arquitetura de software que visa criar sistemas bem estruturados, testáveis e independentes de frameworks, UI (User Interface) e bases de dados. O objetivo é manter o código flexível e adaptável a mudanças, facilitando sua manutenção e evolução ao longo do tempo.

Clean Architecture é baseada em princípios sólidos de design de software, como o princípio da separação de preocupações e a dependência de direção única (DIP - Dependency Inversion Principle). Ela propõe a organização do sistema em camadas concêntricas, onde cada camada representa um nível de abstração e possui suas próprias responsabilidades.

As camadas comuns na Clean Architecture incluem:

1. Entidades: Representam as regras de negócio e são o núcleo da aplicação. São independentes de qualquer detalhe técnico.

2. Casos de Uso (Use Cases): Também conhecidos como Interactors ou Application Layer, essa camada contém a lógica de aplicação. Ela coordena as operações do sistema sem depender de detalhes de implementação.

3. Interfaces de Interface do Usuário (UI): Essa camada lida com a interação do usuário, podendo ser uma interface de linha de comando, interface gráfica ou interface web. É a camada mais externa e depende das camadas internas.

4. Frameworks e Drivers Externos: Inclui frameworks, ferramentas e componentes externos, como bancos de dados, bibliotecas, etc. Essa camada é a mais externa e é responsável por conectar o sistema com o mundo exterior.

Clean Architecture

Uma das principais vantagens da Clean Architecture é a facilidade de testabilidade, já que as regras de negócio e lógica de aplicação são separadas das camadas externas, o que permite a realização de testes automatizados sem depender de detalhes de implementação.

Essa arquitetura promove a manutenção de um código mais limpo, escalável e adaptável a mudanças, pois facilita a identificação e alteração de partes específicas do sistema sem afetar outras áreas.

