

My Hero App:
desenvolvimento de um
aplicativo utilizando os
conceitos de programação
do Uncle Bob

O Trabalho

À medida que a codificação de um software progride, cresce o número de linhas de código e sua complexidade.

Com a finalidade de solucionar esse desafio, estabelecer regras de organização que ajudaram no crescimento e manutenção dos softwares passa a ser uma preocupação comum na comunidade.

O Trabalho

My Hero App é o nome da aplicação desenvolvida em Flutter apresentada neste trabalho.

Basicamente implementar um CRUD aplicando as ideias do Uncle Bob, através da organização do fluxo de dados dentro do aplicativo.

Dart

- Criada pela empresa Google em 2011
- linguagem de tipagem forte
- orientação a objetos
- compila código de máquina ARM e x64 para dispositivos móveis, desktop e back-end

Flutter

- Criada pela empresa Google
- É um framework de desenvolvimento híbrido
- Usa uma engine de games para mostrar telas e animações
- Uso de Widgets

Flutter - Dependências

- provider - controle de estado das telas ;
- image_picker - acesso a galeria e a câmera ;
- path - gerencia o caminho dos arquivos permitindo o acesso dentro da aplicação;
- sqflite - banco de dados no aparelho;
- http - cuida da parte de rede, permitindo acesso em APIs ou Bancos de dados remotos;

Casos de Uso

- contém as regras de negócios específicas do aplicativo
- Definem: o que o aplicativo fará, suas ações, e como serão feitas

Clean Architecture

Clean Architecture é a arquitetura proposta em um artigo publicado por Robert C. Martin em 2012.

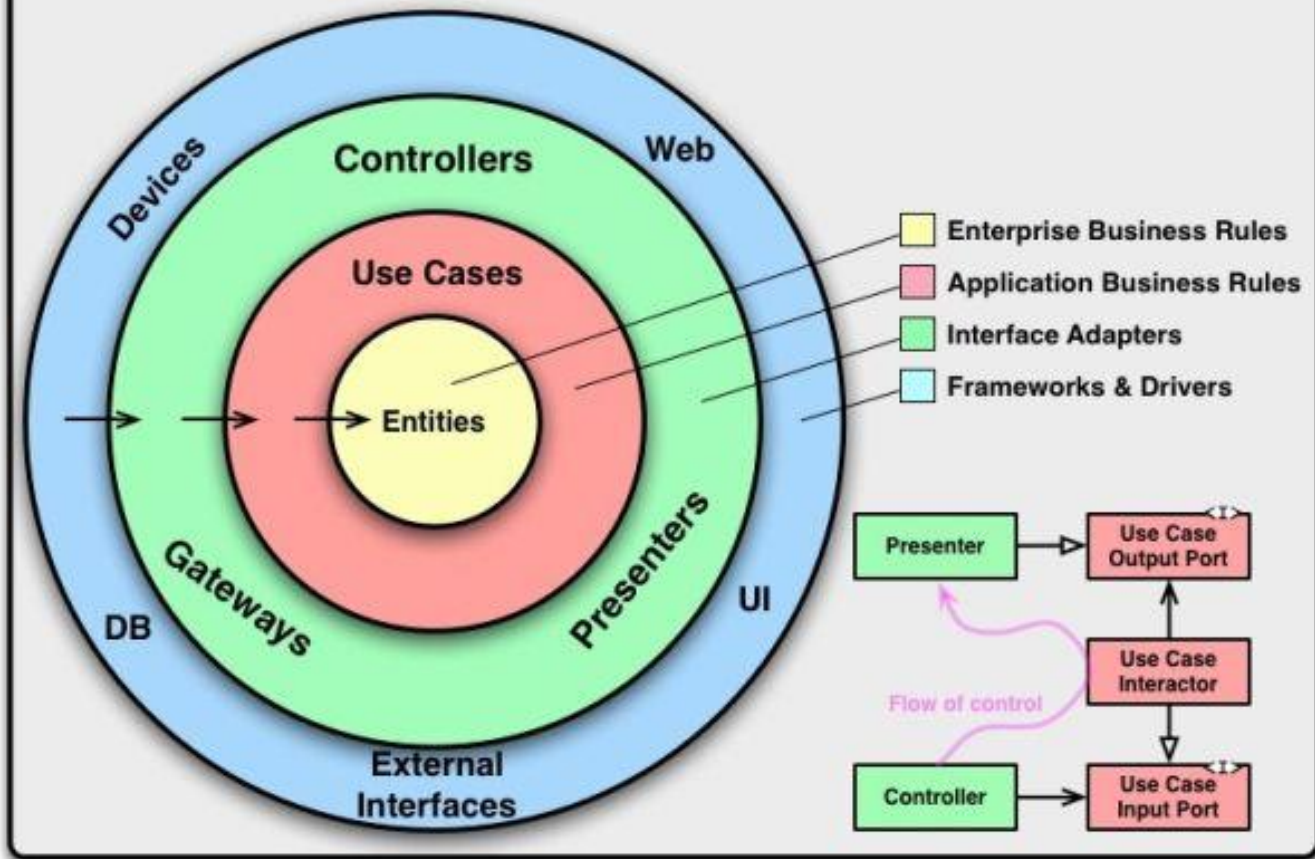
Tem como objetivo facilitar o desenvolvimento de aplicações e manutenções futuras de forma arquitetural.

Clean Architecture

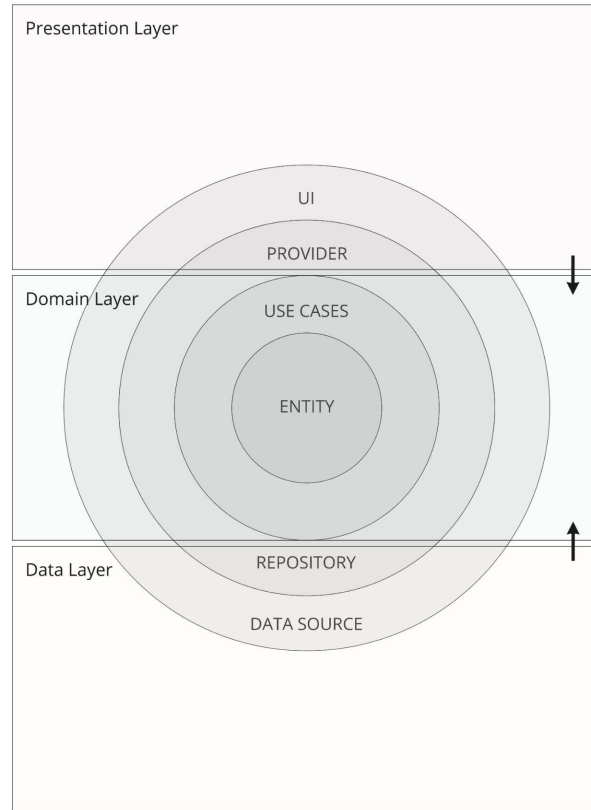
Ao se trabalhar com a arquitetura limpa deve-se seguir o SOLID:

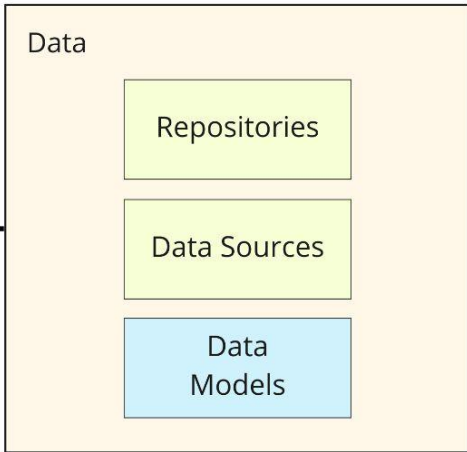
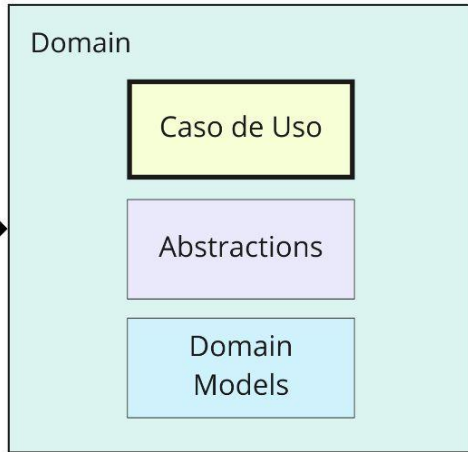
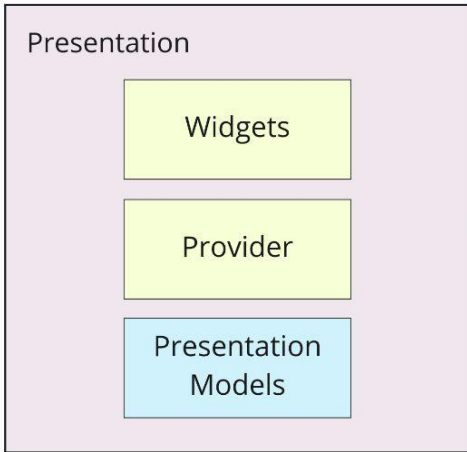
- S —Single Responsibility Principle (Princípio da responsabilidade única)
- O —Open-Closed Principle (Princípio Aberto-Fechado)
- L —Liskov Substitution Principle (Princípio da substituição de Liskov)
- I —Interface Segregation Principle (Princípio da Segregação da Interface)
- D —Dependency Inversion Principle (Princípio da inversão da dependência)

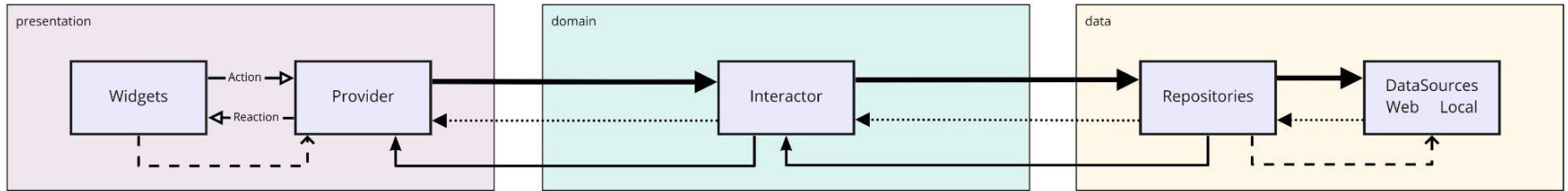
The Clean Architecture



→ Dependency







Aplicativo