

Fundamentos de Arquitetura Limpa

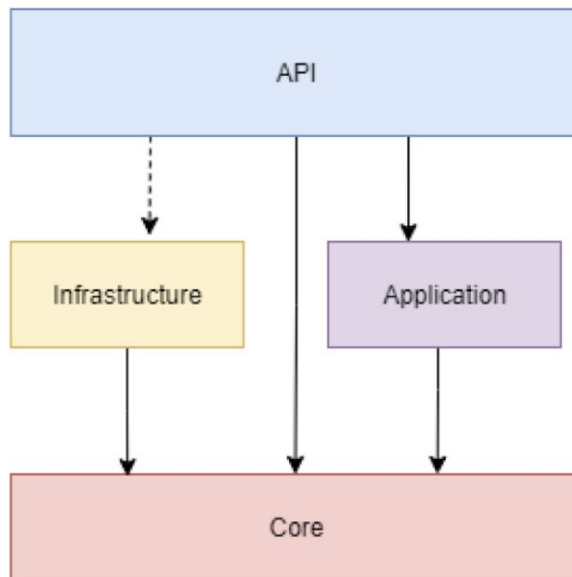
Fundamentos de Arquitetura Limpa

- Arquitetura é a organização fundamental de um sistema, incluindo seus componentes, o relacionamento entre eles, o ambiente, e os princípios em que se baseia sua construção
- Geralmente representados com a descrição de uma estrutura, se utilizando também às vezes de um diagrama ou um desenho

Fundamentos de Arquitetura Limpa

- É uma arquitetura amplamente utilizada em projetos .NET
- Proposta por Robert Martin (Uncle Bob), com foco em promover a testabilidade, desacoplamento, coesão, e reusabilidade de código
- Tem como foco o domínio do sistema, tendo em sua essência o Domain-Driven Design
- Tem diversas variações, mas a essência é a mesma
- Sua estrutura é dividida em 4 camadas principais, sendo elas
 - Core
 - Infrastructure
 - Application
 - API / UI

Fundamentos de Arquitetura Limpa





Até a próxima aula!

Camada Core

Camada Core

- Camada central da Clean Architecture, contendo a expressão em código do domínio do negócio
- Alguns conceitos e componentes do Domain-Driven Design são expressados aqui, como
 - Entidades
 - Value Objects
 - Repositórios (interfaces)
 - Eventos

Camada Core

- Os principais componentes que geralmente estão contidos aqui são:
 - Entidades e Agregados
 - Value Objects
 - Factories
 - Repositórios (interface)
 - Enums relacionados a outros componentes dessa camada
 - Serviços de Domínio
 - Exceções de Domínio

Camada Infraestrutura

Camada Infrastructure

- Camada responsável por integração com componentes de infraestrutura, como acesso a dados, caching, serviços de computação em nuvem, outros sistemas (externos ou internos à organização), entre outros
- Cada um dos subitens pode ser dividido em projetos próprios, como:
 - Persistence
 - Integration
 - Cloud Services
 - Caching
 - Authentication

Camada Infrastructure

- Os principais componentes que geralmente estão contidos aqui são:
 - Classes relacionadas a acesso a dados, como contexto de dados e implementação de Repositórios
 - Serviços de Infraestrutura, que permite integração com serviços anteriormente citados neste tópico

Camada Application

Camada Application

- Camada responsável pela implementação de casos de uso, que poderá variar com o padrão arquitetural, como:
 - Serviços de aplicação, caso siga com a abordagem padrão da Clean Architecture
 - Commands e Queries, caso utilize o CQRS
 - Em algumas implementações, as Queries são encontradas na camada Infrastructure, lidando diretamente com a projeção de dados a partir da biblioteca utilizada para acesso a dados (como EF Core, Dapper, ADO.NET)
 - DTOs de casos de uso, como
 - View Models
 - Input Models

Camada UI/API

Camada UI/API

- Camada responsável pelo código de interface com outros componentes externos, como usuários ou aplicações
- Pode ser uma API (puramente back-end), ou mesmo conter Views, como em modelo MVC
- É configurada a parte de injeção de dependência para a aplicação
- Os principais componentes que geralmente estão contidos aqui são:
 - Controllers e Actions
 - Filters
 - Outros recursos do framework Web



Até a próxima aula!