

Arquitetura de Software

Uma Introdução Detalhada

Introdução à Arquitetura de Software

- Definição:
 - - Estrutura fundamental de um sistema de software.
 - - Componentes e suas interações.
 - - Diretrizes que orientam design e evolução.
- Importância:
 - - Influencia qualidade do software.
 - - Facilita comunicação entre equipes.
 - - Auxilia na tomada de decisões informadas.

Padrões Arquiteturais

- Monolítica:
 - - Sistema implementado como um único código base.
- Microservices:
 - - Sistema dividido em pequenos serviços independentes.
- MVC (Model-View-Controller):
 - - Separação do sistema em Model, View e Controller.
- Camadas (Layers):
 - - Organização em camadas com responsabilidades específicas.

Qualidades Arquiteturais

- Manutenibilidade:
 - - Facilidade de modificar e estender o sistema.
- Escalabilidade:
 - - Capacidade de lidar com aumento de carga.
- Desempenho:
 - - Eficiência em tempo de resposta e uso de recursos.

Estilos Arquiteturais

- Arquitetura em Camadas:
 - - Estrutura em camadas horizontais.
- Arquitetura Hexagonal (Ports and Adapters):
 - - Isola lógica de negócios das dependências externas.
- Event-Driven:
 - - Comunicação por eventos.

Componentes Arquiteturais

- Módulos:
 - - Unidades de código com funcionalidade específica.
- Serviços:
 - - Componentes independentes com funções específicas.
- APIs:
 - - Interfaces de comunicação entre componentes.
- Banco de Dados:
 - - Armazenamento persistente de dados.

Exemplo Prático: Arquitetura Monolítica

- [Frontend/UI] --> [Backend/Business Logic] --> [Database]
- Vantagens:
 - - Simplicidade, fácil de desenvolver e implantar.
- Desvantagens:
 - - Dificuldade de escalar, manutenibilidade complicada.

Exemplo Prático: Arquitetura de Microservices

- [User Service] --> [Database]
- [Product Service] --> [Database]
- [Order Service] --> [Database]

- Vantagens:
 - - Escalabilidade, facilidade de manutenção.

- Desvantagens:
 - - Complexidade na comunicação e gestão de serviços.

Conclusão

- Arquitetura de software é vital para qualidade e sucesso de sistemas.
- Compreensão dos conceitos e boas práticas é essencial.
- Ajuste propriedades conforme necessidades específicas do projeto.