

## Exercício da unidade de Arquitetura de Software

1. Defina o que é arquitetura de software e explique sua importância no desenvolvimento de sistemas.
2. Liste e explique brevemente os principais objetivos da arquitetura de software.
3. Explique a diferença entre estilo arquitetural e padrão arquitetural, usando exemplos para ilustrar cada conceito.
4. Cite um exemplo de um estilo arquitetural e um exemplo de um padrão arquitetural. Explique como eles podem ser aplicados em diferentes contextos.
5. Descreva o estilo arquitetural em camadas (Layered Architecture). Quais são as vantagens e desvantagens desse estilo? Dê um exemplo de aplicação.
6. Descreva o estilo arquitetural orientado a eventos (Event-Driven Architecture). Como os componentes se comunicam nesse estilo? Dê um exemplo de caso em que essa arquitetura é adequada.
7. Explique o estilo arquitetural de tubos e filtros (Pipe and Filter Architecture). Como os dados fluem através dos componentes? Dê um exemplo de aplicação em que esse estilo seja útil.
8. Aborde o estilo arquitetural baseado em microserviços (Microservices Architecture). Quais são as características que definem esse estilo? Quais são os benefícios e desafios associados a ele?
9. Suponha que você esteja desenvolvendo um sistema de comércio eletrônico. Qual estilo arquitetural você escolheria e por quê? Quais seriam as vantagens desse estilo para o cenário de comércio eletrônico?

10. Discuta a importância da escalabilidade ao escolher um estilo arquitetural. Como diferentes estilos lidam com requisitos de escalabilidade?
11. Explique os princípios fundamentais da Clean Architecture e por que ela é considerada uma abordagem valiosa para o design de software.
12. Cite e explique brevemente as camadas definidas na Clean Architecture (Entidades, Casos de Uso, Interfaces de Interface do Usuário, Frameworks e Dispositivos).
13. Descreva a relação entre as camadas da Clean Architecture. Como a direção da dependência é estabelecida entre elas?
14. Quais são os benefícios de manter as regras de negócios na camada de Casos de Uso (Use Cases)? Como isso contribui para a testabilidade e flexibilidade do sistema?
15. Compare a Clean Architecture com outras abordagens arquiteturais mais tradicionais, como a arquitetura em camadas e a arquitetura monolítica. Quais são as principais diferenças?
16. Descreva o propósito das portas (ports) e adaptadores (adapters) na Arquitetura Hexagonal. Como eles contribuem para a separação de preocupações?
17. Cite um exemplo de implementação prática da Arquitetura Hexagonal. Descreva as diferentes camadas e como elas interagem.
18. Quais são as principais vantagens da Arquitetura Hexagonal em termos de testabilidade, flexibilidade e isolamento de dependências?
19. Descreva brevemente o que é o Domain-Driven Design (DDD) e qual é seu principal objetivo ao desenvolver sistemas de software.
20. Defina o que é um "domínio" em DDD e explique por que ele é central para a abordagem.