

# Resumo: Arquitetura de Software

## Introdução:

- Arquitetura de software representa o plano fundamental de um sistema.
- Organiza e comunica os elementos essenciais para a construção do software.
- Arquiteturas confiáveis facilitam manutenções, aumentam o sucesso do projeto e lidam com restrições econômicas e estéticas.

## Princípios SOLID:

Conjunto de diretrizes proposto por Robert C. Martin em 1995, essenciais para a Programação Orientada a Objetos (POO):

- **Responsabilidade Única:** Cada entidade deve ter uma única responsabilidade.
- **Aberto-fechado:** Sistemas devem ser expansíveis sem alterar o código existente.
- **Substituição de Liskov:** Componentes substituíveis devem manter a funcionalidade esperada.
- **Segregação de Interface:** Interfaces específicas para atender às necessidades do cliente.
- **Inversão de Dependência:** Módulos de alto nível não devem depender de módulos de baixo nível.

## Padrões de Arquitetura:

### Arquitetura Cliente-Servidor:

- Divide o sistema em cliente (interface do usuário) e servidor (banco de dados).
- Ideal para sistemas simples, como serviços de e-mail.

### Arquitetura em Camadas:

- Divide o sistema em camadas interdependentes.
- Exemplos incluem sistemas escolares com camadas de apresentação, aplicação, negócios e dados.

### Arquitetura Model-View-Controller (MVC):

- Separa lógica de domínio (Model), interface do usuário (View) e controle de ações do usuário (Controller).
- Amplamente utilizado em desenvolvimento web.

### Arquitetura de Microsserviços:

- Divide grandes sistemas em subsistemas menores e independentes.
- Facilita a escalabilidade e o uso de tecnologias variadas.

## Considerações Finais:

- Arquiteturas devem evoluir com o tempo, garantindo eficiência, escalabilidade e flexibilidade.
- Princípios SOLID promovem sistemas flexíveis e fáceis de manter.
- A aplicação de padrões arquiteturais melhora segurança e adaptabilidade do sistema às demandas futuras.

### ATIVIDADE QUIZZ:

1.

Qual é o significado da sigla MVC em inglês?

a)

Model-View-Controller

b)

Modelo-Visualizador-Controlador

c)

Models-Views-Controllers

d)

Meus Velhos Camaradas

2.

Complete a frase:

"Responsabilidade Única, Aberto-fechado, Substituição de Liskov, Segregação de Interface, Inversão de Dependência são princípios \_\_\_\_\_."

Resp.

3.

Escolha a alternativa que melhor descreve o princípio Aberto-fechado

a)

Cada entidade deve ter uma responsabilidade única

b)

Uma entidade deve ser capaz de ser expandida mantendo sua independência

c)

Uma entidade deve ser capaz de ser modificada sem que se expanda

d)

O sistema deve ser capaz de ser desligado e ligado a qualquer momento do dia

4.

Assinale a INCORRETA

a)

O princípio de inversão de dependência diz que módulos superiores devem depender de módulos inferiores

b)

Arquitetura baseada em microsserviços permite uma melhor escalabilidade

c)

Arquitetura Client-Server é recomendada para sistemas simples

d)

Microserviços é a arquitetura oposta à arquitetura monolítica

5.

Complete a frase:

"Arquitetura em camadas é uma versão mais \_\_\_\_\_ da arquitetura MVC"

Resp.

6.

Qual é o objetivo da arquitetura de software, segundo Robert C. Martin?

a)

Minimizar o uso de recursos humanos

b)

Melhora na qualidade de vida

c)

Reduzir custo de hardware

d)

Maior lucro

7.

Complete a frase:

"O princípio de substituição de \_\_\_\_\_ diz que uma classe derivada deve ser capaz de substituir uma classe base sem que haja problemas"

Resp.

8.

O que é o princípio da responsabilidade única?

a)

Um módulo deve ser capaz de se auto-gerenciar

b)

Um módulo deve ser capaz de realizar múltiplas tarefas

c)

Um módulo deve ter apenas uma razão para mudar

d)

Um módulo deve ser independente de outros módulos

9.

Qual é a diferença entre arquitetura monolítica e microserviços?

- a)  
Monolítica é mais segura do que microserviços
- b)  
Microserviços permitem desenvolvimento e implantação independentes
- c)  
Monolítica é mais fácil de escalar do que microserviços

**d)**  
**Microserviços são mais difíceis de manter do que monolítica**

10.  
Qual é a principal vantagem de usar a arquitetura de microserviços?

- a)  
Desempenho superior em todos os casos
- b)  
Facilidade de implementação

**c)**  
**Escalabilidade e flexibilidade**

d)  
Menor custo de desenvolvimento

11.  
O que significa o princípio de inversão de dependência?

- a)  
Módulos inferiores devem depender de módulos superiores
- b)  
Módulos superiores e inferiores devem ser independentes
- c)  
Módulos superiores devem depender de módulos inferiores

**d)**  
**Módulos superiores não devem depender de módulos inferiores**

12.  
Qual é uma desvantagem comum da arquitetura de microserviços?

- a)  
Facilidade de manutenção

b)

Complexidade de gerenciamento

c)

Escalabilidade limitada

d)

Dependência de um único ponto de falha

13.

O que é o princípio de segregação de interface?

a)

Interfaces devem ser dependentes de módulos superiores

b)

Interfaces devem ser independentes de implementação

c)

Interfaces devem ser grandes e abrangentes

d)

Interfaces devem ser específicas e pequenas

14.

Qual é a principal característica da arquitetura em camadas?

a)

Cada camada tem uma responsabilidade específica

b)

Permite a comunicação direta entre todas as camadas

c)

Camadas são combinadas em um único módulo

d)

Camadas são independentes e não se comunicam

15.

Quem foi a pessoa que escreveu vários livros sobre arquitetura de software, incluindo os livros "Clean Code" e "Clean Architecture"?

a)

Eder Augusto Penharbel

b)

Ricardo de la Rocha Ladeira

c)

Robert Cecil Martin

d)

Luiz Ricardo Uriarte