

Análise e Projeto de Sistemas

1. Qual é a primeira fase do ciclo de vida tradicional de desenvolvimento de software?

- A) Codificação
- B) Análise de requisitos
- C) Testes
- D) Implantação

2. O que diferencia o desenvolvimento ágil do tradicional?

- A) Maior documentação
- B) Iterações curtas e entregas incrementais
- C) Uso exclusivo de linguagens de programação visuais
- D) Ausência de testes

3. Qual é o principal objetivo da análise de sistemas?

- A) Criar a interface do usuário
- B) Entender os requisitos e necessidades do usuário
- C) Escrever código sem especificação
- D) Gerar relatórios financeiros

4. No desenvolvimento ágil, qual prática permite adaptação rápida às mudanças?

- A) Modelo em cascata
- B) Entregas incrementais
- C) Documentação rígida
- D) Planejamento anual fixo

5. Qual técnica de levantamento de requisitos utiliza perguntas estruturadas a usuários e especialistas?

- A) Brainstorming
- B) Entrevistas
- C) Diagrama de classes
- D) Testes de software

6. Qual é a principal vantagem da prototipação?

- A) Reduz o tempo de codificação ao gerar automaticamente código final
- B) Permite validar requisitos com usuários antes do desenvolvimento completo
- C) Substitui completamente os casos de uso
- D) Elimina a necessidade de testes

7. Em workshops de elicitação de requisitos, qual é o principal benefício?

- A) Reduz custos de software
- B) Promove colaboração e alinhamento entre stakeholders
- C) Automatiza a criação de diagramas UML
- D) Substitui testes de aceitação

8. Qual dos seguintes não é uma forma comum de documentar requisitos?

- A) Casos de uso
- B) User stories
- C) Diagramas de sequência
- D) Cenários

9. Qual a diferença entre casos de uso e user stories?

- A) Casos de uso descrevem interações detalhadas, user stories são mais simples e focadas em valor de negócio
- B) Casos de uso são usados apenas em metodologias ágeis
- C) User stories descrevem código, casos de uso descrevem interfaces
- D) Não há diferença significativa

10. Em UML, qual diagrama mostra a interação entre objetos ao longo do tempo?

- A) Diagrama de classes
- B) Diagrama de sequência
- C) Diagrama de pacotes
- D) Diagrama de implantação

11. Qual diagrama UML representa a estrutura estática de um sistema?

- A) Diagrama de atividades
- B) Diagrama de classes
- C) Diagrama de sequência
- D) Diagrama de estados

12. O que é um diagrama de pacotes em UML?

- A) Mostra o fluxo de atividades em um sistema
- B) Agrupa classes relacionadas e define dependências entre elas
- C) Representa interfaces gráficas
- D) Substitui diagramas de sequência

13. Qual diagrama UML é mais indicado para modelar estados de um objeto?

- A) Diagrama de estados
- B) Diagrama de classes
- C) Diagrama de pacotes
- D) Diagrama de casos de uso

14. Qual é o objetivo principal do design de software?

- A) Codificar rapidamente sem planejamento
- B) Criar sistemas flexíveis, manuteníveis e eficientes
- C) Substituir a análise de requisitos
- D) Definir cores e fontes da interface

15. Qual padrão de design garante que uma classe tenha apenas uma instância?

- A) Factory
- B) Singleton
- C) Observer
- D) Strategy

16. Qual padrão de design permite criar objetos sem expor a lógica de criação ao cliente?

- A) Strategy
- B) Observer
- C) Factory
- D) Singleton

17. O padrão Observer é usado para:

- A) Criar uma única instância global
- B) Notificar automaticamente objetos dependentes sobre mudanças em outro objeto
- C) Produzir objetos complexos passo a passo
- D) Separar lógica de negócios e interface

18. Qual padrão permite alterar o comportamento de um objeto em tempo de execução?

- A) Singleton
- B) Factory
- C) Strategy
- D) Observer

19. Em uma arquitetura multicamadas, qual camada normalmente lida com lógica de negócios?

- A) Apresentação
- B) Persistência de dados
- C) Negócios (ou serviço)
- D) Rede

20. Qual a principal vantagem de utilizar arquitetura multicamadas?

- A) Reduz o número de usuários
- B) Facilita manutenção, escalabilidade e testes
- C) Substitui testes unitários
- D) Garante 100% de segurança

21. Qual técnica de elicitação de requisitos é mais interativa e envolve brainstorming?

- A) Entrevista individual
- B) Workshop
- C) Prototipação
- D) Diagrama de classes

22. Qual diagrama UML pode ser usado para representar interações de componentes em sistemas distribuídos?

- A) Diagrama de implantação
- B) Diagrama de classes
- C) Diagrama de atividades
- D) Diagrama de estados

23. Em um diagrama de sequência, o que representa a linha vertical (linha de vida)?

- A) Uma classe do sistema
- B) Um objeto participante na interação
- C) O fluxo de controle condicional
- D) Um pacote UML

24. Qual a finalidade dos cenários em documentação de requisitos?

- A) Detalhar fluxos alternativos e excepcionais do sistema
- B) Substituir diagramas de classes
- C) Criar código automaticamente
- D) Representar somente a camada de apresentação

25. Qual diagrama UML mostra atividades e decisões dentro de um processo?

- A) Diagrama de estados
- B) Diagrama de atividades
- C) Diagrama de sequência
- D) Diagrama de classes

26. O que caracteriza uma metodologia ágil no contexto de análise e projeto de sistemas?

- A) Planejamento extenso e rígido
- B) Iterações curtas, feedback constante e adaptação às mudanças
- C) Documentação completa antes de iniciar o desenvolvimento
- D) Substituição de testes por protótipos

27. Em UML, o que um diagrama de colaboração enfatiza?

- A) Estrutura estática do sistema
- B) Interações entre objetos e mensagens trocadas
- C) Sequência temporal exata
- D) Estrutura de pacotes

28. Qual é a principal função de diagramas de componentes em sistemas distribuídos?

- A) Mostrar a estrutura física e a comunicação entre componentes
- B) Substituir diagramas de classes
- C) Documentar requisitos funcionais
- D) Criar interfaces gráficas

29. O padrão Factory é considerado um padrão:

- A) Comportamental
- B) Estrutural
- C) Criacional
- D) De interface

30. Qual prática de design ajuda a criar software manutenível e flexível?

- A) Acoplar fortemente as classes
- B) Aplicar padrões de design e separar responsabilidades
- C) Evitar documentação
- D) Ignorar testes unitários