

QUIZZ Fichas de trabalho**Questionário 01 - 2025/01 - MDCOO**

Total questions: 50

Tempo da planilha: 1horas 15minutos

Nome do instrutor: Fernando Alves

Nome

Turma

Data

1. Dentre as definições de Software, escolha a opção INCORRETA:

- | | |
|--|---|
| a) estruturas de dados que possibilitam aos programas manipular informações adequadamente; | b) apenas o código fonte escrito em uma linguagem de programação; |
| c) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas; | d) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados; |

2. Qual das seguintes opções melhor descreve a Engenharia de Software?

- | | |
|--|---|
| a) Um campo que lida exclusivamente com a manutenção de sistemas de computador existentes. | b) Uma disciplina que trata dos métodos, processos e melhores práticas objetivando o desenvolvimento de programa de computadores de alta qualidade. |
| c) Um ramo da engenharia que se concentra apenas no desenvolvimento de hardware. | d) Um processo de design de interfaces de usuário para aplicativos de celular. |

3. Qual evento é considerado o marco histórico da criação da área de Engenharia de Software?

- | | |
|--|--|
| a) A criação da linguagem de programação C em 1972. | b) A fundação da Microsoft em 1975. |
| c) O lançamento do primeiro computador pessoal pela IBM em 1981. | d) A Conferência da OTAN realizada em Garmisch, Alemanha, em 1968. |

4. Qual foi o principal objetivo da Conferência da OTAN de 1968 em relação ao desenvolvimento de software?
- a) Discutir a criação de hardware mais potente para suportar softwares complexos.
 - b) Promover a padronização de linguagens de programação.
 - c) Estabelecer diretrizes para a comercialização de softwares na Europa.
 - d) Chamar a atenção para os desafios do desenvolvimento de software e propor soluções baseadas em princípios de engenharia.
5. Dentre as alternativas abaixo, escolha a definição mais adequada para Engenharia de Software?
- a) É a prática de desenvolver software sem a necessidade de processos sistemáticos.
 - b) É a aplicação de abordagens artísticas e intuitivas para desenvolver software.
 - c) É o estudo exclusivo de linguagens de programação para desenvolvimento de aplicativos.
 - d) É a área da Computação que se preocupa em propor e aplicar princípios de engenharia na construção de software.
6. Como a Engenharia de Software contribui para o desenvolvimento de sistemas de software complexos?
- a) Ignorando práticas sistemáticas e disciplinadas para acelerar o desenvolvimento.
 - b) Focando exclusivamente na estética e na interface do usuário dos sistemas.
 - c) Evitando o uso de princípios de engenharia para manter a flexibilidade no desenvolvimento.
 - d) Aplicando abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir software.
7. Qual é uma das razões para a crescente importância da Engenharia de Software no mundo moderno?
- a) A substituição de software por soluções manuais em processos empresariais.
 - b) O aumento da dependência de software em diversos setores, como empresas, governos e dispositivos embarcados.
 - c) A diminuição do uso de sistemas computacionais em empresas e governos.
 - d) A redução da complexidade dos sistemas de software atuais.

8. Qual das opções descreve corretamente o que é Engenharia de Software?

- a) É o estudo das linguagens de programação baseadas em objetos.
- b) É o conjunto de processos, métodos e ferramentas para construir sistemas de software com qualidade.
- c) É a modelagem gráfica de sistemas de informação apenas com UML.
- d) É a modelagem gráfica de sistemas de informação apenas com UML.

9. Combine o elemento da Engenharia de Software com a explicação mais adequada:

Define os passos gerais para o

desenvolvimento e manutenção do software

☐ Processo

Define como fazer um passo

específico do processo

☐ Método

Facilita e automatiza o processo e os métodos

☐ Ferramenta

10. O modelo de cascata é um dos modelos de desenvolvimento mais antigos e tradicionais, ordene as fases deste modelo:

a) Análise
(Requisitos)

b) Implantação

c) Projeto
(Design)

d) Teste

e) Desenvolvimento
(Codificação)

1)

2)

3)

4)

5)

11. Combine a etapa do Processo de Desenvolvimento de Software com a sua respectiva definição:

- | | | |
|---|-----------------------|---------------------------------------|
| Verificação e validação do software | ☐ | ☐ Análise |
| Definição da arquitetura e design da solução | ☐ | ☐ Projeto |
| Codificação | ☐ | ☐ Desenvolvimento |
| Definição dos objetivos e identificação das funcionalidades | ☐ | ☐ Teste |
| Lançamento do software | ☐ | ☐ Implantação |

12. Combine os termos com as suas respectivas definições:

- | | | |
|--|-----------------------|-------------------------------------|
| está relacionado com os conceitos, informações e tipo de dados pertencentes ao Sistema de Informação | ☐ | ☐ Abstração |
| é o modo gráfico que será descrito o modelo | ☐ | ☐ Representação |
| é ação intelectual que identificamos os conceitos e características do Sistema de Informação | ☐ | ☐ Domínio |
| é a imagem criada após o processo de abstração | ☐ | ☐ Modelo |

13. Organize as ferramentas adequadas para cada etapa do processo de desenvolvimento de software

Categorize o seguinte

Editor De Texto

Ferramenta De Modelagem (UML)

Visual Studio Code

GitHub

Planilha Com Cenários De Teste

Navegador

Análise

Projeto

Desenvolvimento

Teste

14. Combine o tipo de aplicativo com sua respectiva característica

É instalado diretamente no Sistema

Operacional de um computador e é executado localmente

Não possui interface gráfica

São desenvolvidas para smartphones e tablets

Funcionam através de navegadores

Aplicativo por Linha de Comando

Aplicativo Desktop

Aplicativo WEB

Aplicativo Mobile

15. Combine os tipos de aplicativos com a sua respectiva vantagem:

Atualização centralizada

Pode ser usado em qualquer lugar

Pode ser usado sem conexão com a internet

Geralmente são mais leves e rápidos

Aplicativo de Linha de Comando

Aplicativo Desktop

Aplicativo WEB

Aplicativo Mobile

16. O Manifesto Ágil foi criado com o objetivo de:
- a) Propor valores e princípios que priorizam indivíduos, interações e entregas de software funcional.
 - b) Estabelecer uma metodologia única para todos os projetos de software.
 - c) Substituir completamente os processos tradicionais de desenvolvimento.
 - d) Eliminar a necessidade de planejamento e documentação em projetos de software.
17. Sobre os princípios de Desenvolvimento Ágil, escolha a opção incorreta:
- a) Documentação é a parte mais importante do processo de desenvolvimento
 - b) Responder a mudanças mais que seguir um plano.
 - c) Indivíduos e interações mais que processos e ferramentas.
 - d) Colaboração com o cliente mais que negociação de contratos.
18. Qual é a principal característica dos processos ágeis de desenvolvimento de software?
- a) Foco exclusivo na documentação detalhada.
 - b) Adoção de ciclos curtos e iterativos com entregas frequentes.
 - c) Execução sequencial de fases sem iteração.
 - d) Desenvolvimento centrado apenas no código, sem interação com o cliente.
19. Em quais situações os métodos ágeis podem não ser recomendados?
- a) Ambientes que valorizam entregas incrementais e feedback contínuo.
 - b) Equipes pequenas e altamente colaborativas.
 - c) Projetos com requisitos bem definidos e estáveis desde o início.
 - d) Desenvolvimento de produtos inovadores com requisitos em constante evolução.
20. O Kanban se caracteriza por:
- a) Exigir a definição completa de requisitos antes do início do desenvolvimento.
 - b) Adotar um quadro visual para gerenciar o fluxo contínuo de trabalho.
 - c) Utilizar sprints fixos de duas semanas para entregas.
 - d) Focar exclusivamente na programação em pares.

21. Combine as seguintes etapas da Engenharia de Requisitos:

e de Requisitos Os requisitos coletados são analisados para garantir que sejam completos, consistentes, claros e viáveis.



☐ Elicitação de Requisitos

Muitas vezes, surgem conflitos entre requisitos que precisam ser resolvidos.

Consiste em acompanhar as mudanças nos requisitos durante o ciclo de vida do software, controlando versões, rastreabilidade e impacto das alterações.



☐ Análise de Requisitos

Garante que os requisitos realmente representam o que o cliente deseja. Técnicas comuns incluem revisões, prototipação, simulações e testes.



☐ Especificação de Requisitos

É a documentação formal dos requisitos. Pode ser feita em linguagem natural, modelos UML, diagramas, ou linguagens formais.



☐ Validação de Requisitos

É o processo de descobrir os requisitos através de entrevistas, questionários, observações, workshops, brainstorming, etc.



☐ Gerenciamento de Requisitos

Envolve entender o problema e as necessidades do cliente.

22. Sobre Tipo de Requisitos, correlacione o termo e a sua respectiva definição:

Descrevem o que o sistema deve fazer (ex: "o sistema deve permitir login com senha").



Requisitos Funcionais

É um tipo de requisito que descreve explicitamente o que o sistema não deve fazer.



Requisitos Não Funcionais

Descrevem restrições ou qualidades do sistema (ex: desempenho, segurança, usabilidade)



Requisitos de Domínio

Regras específicas da área de negócio (ex: "a taxa de juros deve ser recalculada conforme o CDI").



Requisito Inverso

23. Qual é uma característica das histórias de usuário em comparação com os casos de uso?

a) São mais simples e orientadas ao diálogo com o usuário, sem entrar em detalhes técnicos.

b) São mais detalhadas tecnicamente e apresentam todos os fluxos alternativos.

c) São utilizadas apenas em projetos com metodologia em cascata.

d) Exigem o uso obrigatório de diagramas UML para validação.

24. O que são histórias de usuários no contexto da engenharia de requisitos?

a) Narrativas curtas que descrevem funcionalidades do sistema do ponto de vista do usuário.

b) Documentos técnicos detalhados que descrevem a arquitetura do sistema.

c) Relatórios de desempenho do sistema em produção.

d) Manuais de instrução para usuários finais.

25. Qual é o formato clássico de uma história de usuário?

a) Como [desenvolvedor], preciso de [tecnologia] para [implementar funcionalidade].

b) Como [stakeholder], desejo [recurso] com [prazo de entrega].

c) Como [tipo de usuário], quero [ação ou funcionalidade], para que [benefício ou motivo].

d) Se [condição], então [resultado esperado], caso contrário [alternativa].

26. O que melhor define um caso de uso?

- a) Um modelo matemático para descrever o fluxo de dados entre sistemas.
- b) Um resumo em linguagem informal das regras de negócio.
- c) Um conjunto de passos que descreve interações entre um usuário e o sistema para atingir um objetivo.
- d) Um conjunto de testes automatizados para validar funcionalidades.

27. Em que contexto os casos de uso são mais indicados do que histórias de usuário?

- a) Em projetos sem a necessidade de interação com stakeholders.
- b) Quando se deseja prototipar rapidamente funcionalidades com foco em interface.
- c) Em processos tradicionais ou híbridos que exigem documentação mais detalhada e cenários alternativos.
- d) Em projetos ágeis com entregas semanais.

28. Organize essas opções nas categorias certas

Categorize o seguinte

Foco No Processo	Reuso Limitado	Dados Globais	Foco Nos Objetos
Facilita O Reuso		Dados Protegidos	

Programação Procedural

Programação Orientada A Objetos

29. Combine os termos com as suas respectivas definições:

Atributo	é uma descrição de um conjunto de objetos que compartilham atributos, operações e relacionamentos
Objeto	instância de uma classe
Método	é uma característica de uma classe
Classe	é uma operação que a classe realiza

30. Qual dos seguintes não é um tipo de relacionamento em um Diagrama de Classes UML?

- a) Herança
- b) Associação
- c) Instância
- d) Agregação

31. O padrão de arquitetura para WEB em três camadas separa as funcionalidades, visando tornar o sistema mais flexível, possibilitando sua alteração de forma mais independente em cada camada. Quais são essas camadas?

- a) Camada de teste, Camada de funções e Camada de aplicativos.
- b) Camada de apresentação, Camada de Regras de Negócio e Camada de acesso a dados.
- c) Camada de componentes, Camada de aplicativos e Camada de negócios.
- d) Camada de regras de negócios, Camadas de aplicativos e Camada de funções.

32. Segundo a Linguagem de Modelagem Unificada (UML), uma classe é representada completa e corretamente como:

- a) uma elipse com dois compartimentos que são o nome da classe e seus métodos.
- b) um retângulo com três compartimentos que são o nome da classe, seus atributos e seus métodos.
- c) um retângulo com quatro compartimentos que são o nome da classe, seus objetos, seus parâmetros de visibilidade e suas operações.
- d) um losango com três compartimentos que são o nome do objeto, o nome da classe e seus atributos com as respectivas visibilidades.

33. O profissional de TI poderá utilizar uma forma de representação gráfica para descrever a estrutura de um Sistema de Informação (SI) em desenvolvimento. A linguagem de notação UML (Unified Modeling Language) provê um diagrama de estrutural que ilustra a estrutura estática dos objetos do SI. Qual é o nome deste diagrama?

 - diagrama de atividades
 - diagrama de casos de uso
 - diagrama de pacotes
 - diagrama de classes

34. O que o Diagrama de Classes na UML representa principalmente?

 - Estrutura estática de um sistema
 - Processos de negócios em um sistema
 - Sequência temporal de eventos
 - Interações entre os objetos

35. Qual dos seguintes elementos NÃO é encontrado em um Diagrama de Classes UML?

 - Atributos
 - Relacionamentos
 - Fluxos de controle
 - Métodos

36. Qual característica é típica do paradigma de programação orientado a objetos?

 - Organiza o programa em termos de objetos e classes.
 - É focado apenas em estruturas de repetição.
 - Depende do uso de instruções em ordem linear.
 - Utiliza exclusivamente sequências de decisão.

37. Qual é a finalidade do relacionamento de associação em um Diagrama de Classes UML?

 - Mostrar a herança entre classes
 - Indicar interações entre objetos
 - Representar dependência entre classes
 - Exibir um relacionamento direto entre classes

38. Associe o atributo com o seu respectivo tipo de dado:

Quantidade de Dependentes	☐ string (campo texto/alfanumérico)
Peso	☐ number (valor numérico inteiro)
Nome	☐ number (valor numérico decimal/real) ☐
Data de Nascimento	☐ date (data)

39. Um atributo de uma classe pode ser definido como "+ dataAdmissao: date = '15/01/2023' " onde o símbolo "+" determina a (a) _____ do atributo, a "dataAdmissao" significa o (b) _____ do atributo, o "date" significa o (c) _____ e "15/01/2023" é o (d) _____.

Escolha uma das palavras abaixo

Visibilidade

Nome

Tipo Do Dado

Valor Padrão Inicial

Público

Privado

Numérico

String

40. Um método de uma classe pode ser definido como "+ calcularIMC(peso,altura):float;" onde símbolo "+" determina a (a) _____ do método, o termo "calcularIMC" significa o nome do (b) _____, os termos "peso" e "altura" significam os (c) _____ do método e "float" significa o (d) _____

Escolha uma das palavras abaixo

Visibilidade

Método

Parâmetros

Tipo Do Dado De Retorno .

Público

Privado

Numérico

String

41. "Uma associação, em um Diagrama de Classe, é uma (a) _____ entre duas classes, direcionada da classe de (b) _____ para classe de (c) _____." (FOWLER, 2003, p. 54).

Escolha uma das palavras abaixo

Linha Cheia

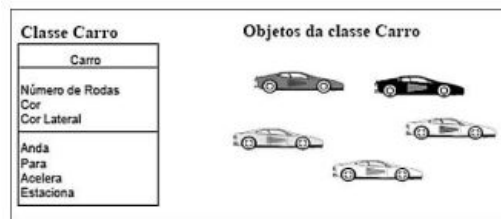
Origem

Destino

Linha Tracejada

Linha Pontilhada

42.



A modelagem de sistemas usando UML consiste em organizar o software como uma coleção de objetos discretos que incorporam a estrutura dos dados e o comportamento. Um exemplo é exibido pela figura ao lado. Sobre a UML, pode-se afirmar que:

- a) O mesmo objeto pertence a várias classes. b) Uma classe é uma descrição de um grupo de objetos com atributos, comportamentos, relacionamentos e semântica comuns.
- c) As características de um objeto são: identidade, atributos e representação. d) A classe é chamada de instância de seu objeto.

43. Considerando os diagramas de classes UML, relacione as os termos:

Multiplicidade ☹

☹ Linha cheia entre duas classes

Associação ☹

☹ Indica quantos objetos podem preencher a propriedade

Valor Padrão ☹

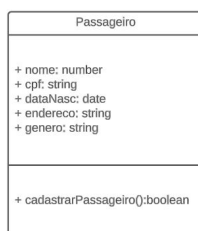
☹ Métodos presentes em uma classe

Operações ☹

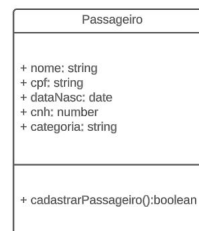
☹ Valor inicial assumido caso não seja feita uma atribuição específica

44. Você é o analista responsável pela modelagem do aplicativo UBER, considerando o domínio, dados e tipos de dados desta aplicação selecione a representação mais adequada para a classe Passageiro:

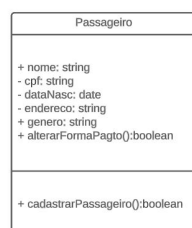
a)



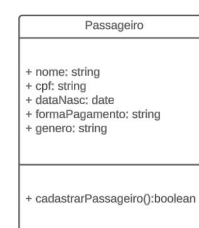
b)



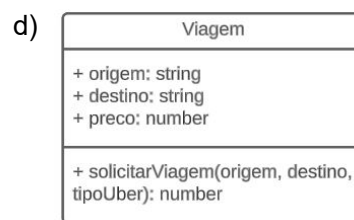
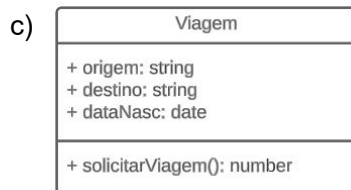
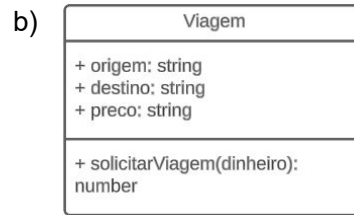
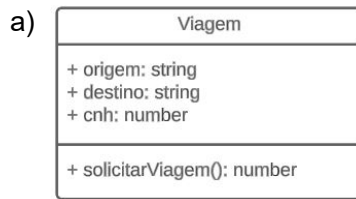
c)



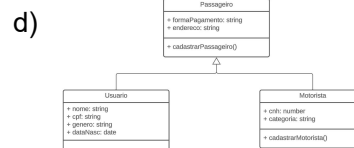
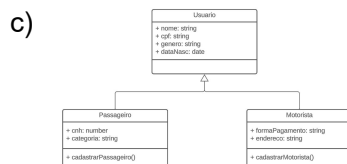
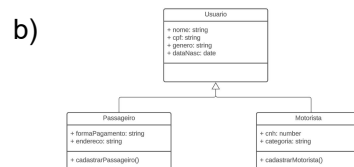
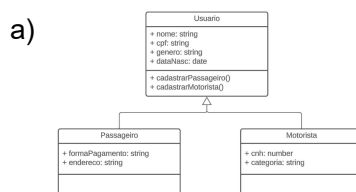
d)



45. Você é o(a) analista responsável pela modelagem do aplicativo UBER, considerando o domínio, dados e tipos de dados desta aplicação selecione a representação mais adequada para a classe Viagem:



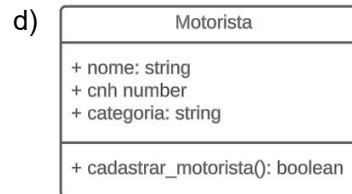
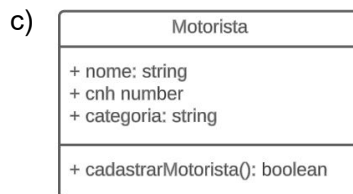
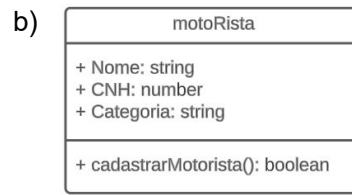
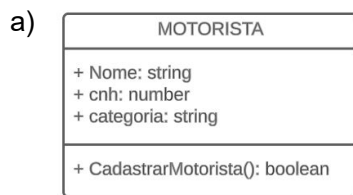
46. Você é o(a) analista responsável pela modelagem do aplicativo UBER, considerando o domínio, dados e tipos de dados desta aplicação selecione a representação mais adequada para o relacionamento do tipo Especialização/Generalização entre as classes "Usuario", "Passageiro" e "Motorista":



47. Qual elemento NÃO faz parte dos 3C's das histórias de usuário?

- a) Confirmation b) Condition
c) Card d) Conversation

48. Considerando as convenções de nomenclatura Camel Case para atributos e métodos e a Pascal Case para classes, selecione a representação da classe "Motorista" que obedece essas convenções:



49. Combine o nome da convenção de nomenclatura e o seu respectivo exemplo:

Camel case	☐ primeiroNomePessoa
Snake case	☐ PrimeiroNomePessoa
Pascal case	☐ primeiro_nome_pessoa
Screaming snake case	☐ primeiro-nome-pessoa
Kebab case	☐ PRIMEIRO_NOME_PESSOA

50. Qual é o objetivo principal da modelagem de software?

- a) Executar testes automatizados de interface. b) Criar bancos de dados relacionais completos.
- c) Preencher a lacuna entre os requisitos e o código. d) Realizar simulações em tempo real do sistema

Chaves de resposta

- | | | |
|--|---|---|
| 1. b) apenas o código fonte escrito em uma linguagem de programação; | 2. b) Uma disciplina que trata dos métodos, processos e melhores práticas objetivando o desenvolvimento de programa de computadores de alta qualidade. | 3. d) A Conferência da OTAN realizada em Garmisch, Alemanha, em 1968. |
| 4. d) Chamar a atenção para os desafios do desenvolvimento de software e propor soluções baseadas em princípios de engenharia. | 5. d) É a área da Computação que se preocupa em propor e aplicar princípios de engenharia na construção de software. | 6. d) Aplicando abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir software. |
| 7. b) O aumento da dependência de software em diversos setores, como empresas, governos e dispositivos embarcados. | 8. b) É o conjunto de processos, métodos e ferramentas para construir sistemas de software com qualidade. | 9. Define os passos gerais para o desenvolvimento e manutenção do software
- Processo,
Define como fazer um passo específico do processo
- Método,
Facilita e automatiza o processo e os métodos
- Ferramenta |
| 10. a, c, e, d, b | 11. Definição dos objetivos e identificação das funcionalidades
- Análise,
Definição da arquitetura e design da solução
- Projeto, Codificação -
Desenvolvimento, | 12. é ação intelectual que identificamos os conceitos e características do Sistema de Informação
- Abstração,
é o modo gráfico que será descrito o modelo
- Representação,
está relacionado com os conceitos, informações e tipo de dados |

Verificação e validação do software

- Teste,

Lançamento do software -

Implantação

pertencentes ao Sistema de Informação

- Domínio,

é a imagem criada após o processo de abstração

- Modelo

13. **Análise**

Editor De Texto

Projeto

Ferramenta De Modelagem (UML)

Desenvolvimento

Visual Studio Code

GitHub

Teste

Planilha Com Cenários De Teste

Navegador

14. Não possui interface gráfica

-

Aplicativo por Linha de Comando

,

É instalado diretamente no Sistema Operacional de um computador e é executado localmente

- Aplicativo Desktop,

Funcionam através de navegadores

- Aplicativo WEB,

São desenvolvidas para smartphones e tablets

- Aplicativo Mobile

15. Geralmente são mais leves e rápidos

-

Aplicativo de Linha de Comando

,

Pode ser usado sem conexão com a internet

- Aplicativo Desktop,

Atualização centralizada -

Aplicativo WEB,

Pode ser usado em qualquer lugar

- Aplicativo Mobile

16. a) Propor valores e princípios que priorizam indivíduos, interações e entregas de software funcional.

17. a) Documentação é a parte mais importante do processo de desenvolvimento

18. b) Adoção de ciclos curtos e iterativos com entregas frequentes.

19. c) Projetos com requisitos bem definidos e estáveis desde o início.

20. b) Adotar um quadro visual para gerenciar o fluxo contínuo de trabalho.

21. É o processo de descobrir os requisitos através de entrevistas, questionários, observações, workshops, brainstorming, etc. Envolve entender o problema e as necessidades do cliente.
- Elicitação de Requisitos
,
e de Requisitos Os requisitos coletados são

analisados para garantir que sejam completos, consistentes, claros e viáveis. Muitas vezes, surgem conflitos entre requisitos que precisam ser resolvidos.

- Análise de Requisitos, É a documentação formal dos requisitos. Pode ser feita em linguagem natural, modelos UML, diagramas, ou linguagens formais.

- Especificação de Requisitos

, Garante que os requisitos realmente representam o que o cliente deseja. Técnicas comuns incluem revisões, prototipação, simulações e testes.

- Validação de Requisitos , Consiste em acompanhar as mudanças nos requisitos durante o ciclo de vida do software, controlando versões, rastreabilidade e impacto das alterações.

- Gerenciamento de Requisitos

22. Descrevem o que o sistema deve fazer (ex: "o sistema deve permitir login com senha").
- Requisitos Funcionais,

23. a) São mais simples e orientadas ao diálogo com o usuário, sem entrar em detalhes técnicos.

24. a) Narrativas curtas que descrevem funcionalidades do sistema do ponto de vista do usuário.

Descrevem restrições ou qualidades do sistema (ex: desempenho, segurança, usabilidade)

-

Requisitos Não

Funcionais

,

Regras específicas da área de negócio (ex: "a taxa de juros deve ser recalculada conforme o CDI").

- Requisitos de Domínio,
É um tipo de requisito que descreve explicitamente o que o sistema não deve fazer.

- Requisito Inverso

25. c) Como [tipo de usuário], quero [ação ou funcionalidade], para que [benefício ou motivo].

26. c) Um conjunto de passos que descreve interações entre um usuário e o sistema para atingir um objetivo.

27. c) Em processos tradicionais ou híbridos que exigem documentação mais detalhada e cenários alternativos.

28. **Programação Procedural**

Foco No Processo

Reuso Limitado

Dados Globais

Programação Orientada a Objetos

Foco Nos Objetos

Facilita O Reuso

Dados Protegidos

29. Classe -
é uma descrição de um conjunto de objetos que compartilham atributos, operações e relacionamentos
, Objeto -
instância de uma classe,
Atributo -
é uma característica de uma classe
, Método -
é uma operação que a classe realiza

30. c) Instância

31. b) Camada de apresentação, Camada

32. b) um retângulo com três compartimentos que

33. d) diagrama de classes

- | | | |
|--|--|---|
| de Regras de Negócio e Camada de acesso a dados. | são o nome da classe, seus atributos e seus métodos. | |
| 34. a) Estrutura estática de um sistema | 35. c) Fluxos de controle | 36. a) Organiza o programa em termos de objetos e classes. |
| 37. d) Exibir um relacionamento direto entre classes | 38. Nome -
string (campo texto/alfanumérico)
,
Quantidade de Dependentes -
number (valor numérico inteiro)
, Peso -
number (valor numérico decimal/real)
, Data de Nascimento -
date (data) | 39. visibilidade, nome, tipo do dado, valor padrão inicial |
| 40. visibilidade, método, parâmetros, tipo do dado de retorno . | 41. linha cheia, origem, destino | 42. b) Uma classe é uma descrição de um grupo de objetos com atributos, comportamentos, relacionamentos e semântica comuns. |
| 43. Associação -
Linha cheia entre duas classes
, Multiplicidade -
Indica quantos objetos podem preencher a propriedade
, Operações -
Métodos presentes em uma classe
, Valor Padrão -
Valor inicial assumido caso não seja feita uma | 44. d) | 45. d) |

atribuição específica

46. b)

47. b) Condition

48. c)

49. Camel case -

primeiroNomePessoa,

Pascal case -

PrimeiroNomePessoa,

Snake case -

primeiro_nome_pessoa,

Kebab case -

primeiro-nome-pessoa,

Screaming snake case -

PRIMEIRO_NOME_PESSOA

50. c) Preencher a lacuna

entre os requisitos e o
código.