

ATIVIDADE DE REVISÃO DE ENGENHARIA DE SOFTWARE- GERALZÃO

Entregar até o final da aula.

Divirta-se!

1. O que é Engenharia de Software?

- a) A criação de código-fonte para aplicativos móveis. ✕
- b) O estudo dos componentes físicos do computador. ✕
- ☒ c) A aplicação de abordagens sistemáticas no desenvolvimento de software. •
- d) A utilização exclusiva de metodologias ágeis no desenvolvimento de sistemas. ✕
- e) A administração de redes de computadores.

2. Qual é a principal finalidade da Engenharia de Requisitos?

- a) Criar a interface gráfica do sistema. ✕
- ☒ b) Definir o que o sistema deve fazer antes da implementação. •
- c) Testar o software antes da entrega final.
- d) Corrigir falhas após a implantação do software.
- e) Criar bancos de dados para armazenar informações do sistema.

3. Os requisitos funcionais são aqueles que:

- a) Definem restrições/ou qualidades do sistema.
- ☒ b) Descrevem o comportamento esperado do sistema.

c) São opcionalmente implementados no software. ✕

d) Dizem respeito apenas à interface do usuário. ✕

e) Referem-se exclusivamente ao desempenho do sistema. ✕

4. Um exemplo de requisito não funcional é:

- a) O sistema deve permitir o login de usuários.
- b) O usuário pode cadastrar produtos no sistema.
- ☒ c) O sistema deve responder às requisições em até 2 segundos. •
- d) O sistema deve permitir a impressão de relatórios.
- e) O usuário pode criar contas e alterar senhas.

5. Qual das seguintes atividades pertence ao processo de Engenharia de Requisitos?

- a) Codificação de um sistema.
- ☒ b) Identificação e levantamento das necessidades dos usuários. •
- c) Testes automatizados no código-fonte.
- d) Implantação do sistema em produção.
- e) Configuração de servidores para hospedagem.

6. A fase de análise e negociação na Engenharia de Requisitos tem como objetivo:

a) Criar uma interface amigável para os usuários. ✓

~~b) Avaliar a viabilidade dos requisitos e resolver conflitos.~~

c) Especificar o banco de dados do sistema.

d) Criar a documentação final do sistema. ✗

e) Testar a usabilidade do software.

7. A especificação de requisitos consiste em:

a) Implementar os requisitos do software.

~~b) Criar um documento detalhado contendo todos os requisitos do sistema.~~

c) Testar os requisitos antes da implementação.

d) Alterar a arquitetura do sistema conforme os requisitos. ✗

e) Criar wireframes para o sistema. ✗

8. O que ocorre na fase de validação dos requisitos?

~~a) Os requisitos são revisados para garantir que sejam completos e consistentes.~~

b) Os requisitos são codificados no sistema. ✗

c) Os usuários finais testam o sistema em produção. ✗

d) O desempenho do sistema é avaliado com base nos requisitos. ✗

e) São definidos os prazos e custos do projeto.

9. Qual das técnicas abaixo NÃO é uma técnica de levantamento de requisitos?

a) Entrevistas com clientes e usuários.

b) Observação do ambiente de trabalho. •

c) Questionários estruturados. •

~~d) Implementação direta do sistema sem análise prévia.~~

e) Prototipação.

10. No modelo Cascata, qual é a principal característica?

a) Desenvolvimento ocorre em ciclos curtos e iterativos. ✗

~~b) As fases são executadas sequencialmente, sem retorno para etapas anteriores.~~

c) O cliente participa ativamente em cada fase do desenvolvimento.

d) Os requisitos podem ser alterados a qualquer momento do processo. ✗

e) Os testes ocorrem antes da implementação do sistema. ✗

11. No modelo Incremental, o software é desenvolvido:

a) De forma linear, sem possibilidade de revisões. ✗

~~b) Em pequenos incrementos funcionais entregues ao longo do tempo.~~ •

c) Apenas após a definição completa dos requisitos.

d) Baseando-se exclusivamente no código-fonte existente. ✗

e) De maneira aleatória, sem planejamento prévio. ✗

12. O modelo Espiral (combina elementos) dos modelos:

a) Cascata e Ágil. ✗

~~b) Incremental e Cascata.~~

c) Cascata e Prototipação. ✗

d) Ágil e Incremental.

e) Prototipação e V. ✗

13. No desenvolvimento ágil, uma das principais vantagens é:

a) A rigidez na documentação do projeto. ✗

~~b) A flexibilidade para mudanças nos requisitos durante o desenvolvimento.~~

c) A falta de contato com o cliente durante o projeto. ✗

d) A necessidade de especificação completa antes do início da codificação.

e) A eliminação da necessidade de testes. ✗

14. O Scrum organiza o trabalho em ciclos curtos chamados:

a) Releases.

b) Itens de backlog.

~~c) Sprints.~~

d) Requisitos.

e) Processos.

15. Qual é o papel do Product Owner no Scrum?

a) Desenvolver código para o sistema. ✗

b) Testar funcionalidades do software. ✗

~~c) Priorizar e gerenciar o backlog do produto.~~

d) Supervisionar o trabalho dos desenvolvedores.

e) Criar a interface gráfica do sistema.

16. A etapa de testes no desenvolvimento de software tem como principal objetivo:

a) Avaliar a viabilidade econômica do projeto. ✗

~~b) Corrigir falhas e validar o funcionamento do sistema.~~

c) Implementar novos requisitos no sistema.

d) Criar a documentação do sistema. ✗

e) Substituir a fase de levantamento de requisitos. ✗

17. Qual das opções abaixo melhor define o conceito de Stakeholder?

a) Apenas o cliente final que utilizará o sistema. ✗

b) Apenas a equipe técnica que desenvolve o software.

~~c) Todas as pessoas, grupos ou entidades impactadas pelo projeto.~~

d) Apenas o gerente de projetos e os investidores.

e) Somente os usuários finais que irão testar o sistema.

18. A Engenharia Reversa tem como finalidade principal:

a) Implementar um sistema novo do zero. ✗

b) Criar softwares sem documentação. ✗

~~c) Analisar um sistema existente para entender sua estrutura e funcionalidades.~~

d) Remover bugs de um software em produção. ✕

e) Melhorar a segurança dos bancos de dados.

19) O ciclo de vida do software descreve:

a) Apenas a fase de implementação de um sistema.

~~b) As fases de desenvolvimento, manutenção e descarte de um software.~~

c) Somente os testes de um software antes da implantação.

d) A modelagem do banco de dados antes da codificação.

e) O uso exclusivo de metodologias ágeis.

20. Qual é a primeira etapa no Processo de Desenvolvimento de Software?

a) Testes. ✕

b) Implementação. ✕

~~c) Levantamento de Requisitos.~~

d) Implantação.

e) Análise de Desempenho.

21. O que é um Diagrama de Classes?

A) Um modelo que representa os dados em um banco de dados. ✕

~~B) Um diagrama que descreve a estrutura estática do sistema, mostrando classes, atributos e relacionamentos.~~

C) Um fluxograma que descreve a sequência de execução de um algoritmo. ✕

D) Um diagrama que representa a interface gráfica de um software. ✕

22) O que define um objeto em um Diagrama de Classes?

A) O nome da classe e seus relacionamentos.

B) As funções e métodos que ele pode executar.

~~C) Sua identidade, estado (atributos) e comportamento (métodos).~~

D) Apenas sua identidade e seu nome.

23. Qual dos elementos abaixo NÃO faz parte de uma classe no Diagrama de Classes?

A) Nome da classe

B) Atributos

C) Métodos

~~D) Relacionamentos entre objetos~~

24. Qual a principal diferença entre agregação e composição?

A) A agregação indica uma relação "é um", enquanto a composição indica "tem um".

B) A composição é uma relação fraca, enquanto a agregação é uma relação forte. ✕

~~C) Na agregação, um objeto pode existir sem o outro; na composição, a existência de um depende do outro.~~

D) Não há diferença entre agregação e composição. ✓

25) Qual exemplo melhor representa uma relação de composição?

~~A) Um carro e seus pneus.~~

B) Um professor e seus alunos. ✓

C) Um banco e seus clientes.

D) Uma cidade e seus habitantes.

26. Qual exemplo melhor representa uma relação de agregação?

- A) Um computador e sua CPU. ✓
- B) Uma escola e seus professores.
- C) Uma empresa e suas filiais. ✓
- ~~D) Um pedido e seus itens. ✓~~

27. O que significa a multiplicidade 1..* em um relacionamento entre classes?

- ~~A) Pelo menos um, podendo ter vários.~~
- B) Apenas um.
- C) Nenhum ou vários.
- D) Exatamente um ou dois.

28. O que significa a multiplicidade 0..1 em um relacionamento?

- A) Um ou mais objetos. ✓
- ~~B) Nenhum ou um objeto.~~
- C) Apenas um objeto.
- D) Nenhum ou vários objetos.

29. No UML, como representamos um atributo protegido?

- A) + atributo
- B) - atributo
- ~~C) # atributo~~
- D) @ atributo

30. Qual dos exemplos abaixo representa um método em uma classe UML?

- A) - nome : String
- ~~B) + calcularSalario() : double~~
- C) # idade : int
- D) ~ endereço : String

31. Qual dessas relações indica herança no Diagrama de Classes?

- A) Associação
- ~~B) Generalização~~
- C) Dependência
- D) Composição

32. Como representamos um relacionamento de herança no UML?

- A) Uma linha sólida com um losango na ponta. ✓
- ~~B) Uma linha sólida com um triângulo na ponta.~~
- C) Uma linha tracejada com uma seta.
- D) Uma linha sólida com um círculo na ponta.

33. Qual das seguintes opções define corretamente o conceito de associação?

- ~~A) Representa um relacionamento entre classes que pode ter diferentes multiplicidades.~~
- B) Define um relacionamento obrigatório entre classes. ✓
- C) Indica herança entre classes. ✓
- D) Define um relacionamento onde um objeto faz parte do outro. ✓

34. Em uma classe UML, os métodos representam:

- ~~A) O comportamento da classe.~~
- B) Os dados armazenados pela classe. ✓
- C) As instâncias da classe. ✓
- D) A identidade do objeto. ✓

35. Em um Diagrama de Classes UML, um atributo pode ser:

- A) Apenas público. ✓
- B) Apenas privado ou protegido.
- ~~C) Público, privado ou protegido.~~
- D) Apenas privado.

~~36.~~ Qual alternativa descreve corretamente uma classe abstrata?

- ~~A)~~ Uma classe que não pode ser instanciada diretamente.
- B) Uma classe que sempre precisa de atributos privados.
- C) Uma classe sem atributos.
- D) Uma classe sem métodos.

~~37.~~ Qual das opções abaixo representa corretamente a relação entre um pedido e seus itens no UML?

- A) Composição, pois um pedido não pode existir sem seus itens.
- ~~B)~~ Agregação, pois os itens podem existir sem o pedido.
- C) Associação genérica, pois não há dependência. ✓
- D) Dependência, pois um pedido pode depender de outra classe. ✓

38. Em um diagrama de classes, o que uma classe representa?

- ~~A)~~ Um conjunto de objetos com características e comportamentos semelhantes.
- B) Um processo dentro do sistema. ✓
- C) Um banco de dados.
- D) Uma interface gráfica. ✓
- E) Um arquivo de configuração. ✓

39. O que significa a herança em um diagrama de classes?

- A) A relação entre um objeto e sua função dentro do sistema. ✓
- ~~B)~~ A reutilização de código onde uma classe filha herda atributos e métodos de uma classe pai. •
- C) A relação entre banco de dados e classes. ✓
- D) A divisão de um sistema em camadas. ✓

E) A duplicação de código para otimizar o desempenho.

40. Qual símbolo representa a herança em um diagrama de classes UML?

- A) Uma linha reta simples. ✓
- B) Um losango preenchido. ✓
- C) Um losango vazio. ✓
- ~~D)~~ Uma linha com um triângulo na ponta.
- E) Um círculo ligado à classe.

41. O que é a composição em um diagrama de classes?

- A) Um relacionamento onde um objeto pode existir independentemente do outro. ✓
- ~~B)~~ Um relacionamento onde um objeto depende da existência de outro para existir. •
- C) Um método para organizar atributos dentro de uma classe. ✓
- D) A implementação de múltiplas interfaces. ✓
- E) A separação de responsabilidades dentro de um sistema. ✓

42. No UML, como a composição é representada?

- A) Uma linha reta simples. ✓
- ~~B)~~ Um losango preenchido.
- C) Um losango vazio.
- D) Um círculo ligado à classe.
- E) Um triângulo invertido.

43. Qual das opções abaixo é um exemplo de composição?

- ~~A)~~ Uma árvore e seus galhos.
- B) Um cliente e suas compras.
- C) Um aluno e uma escola.
- D) Um professor e sua matéria.
- E) Um livro e sua editora.

44. O que é agregação em um diagrama de classes?

- ~~A) Um relacionamento fraco onde um objeto pode existir independentemente do outro.~~
- B) Um relacionamento forte onde um objeto não pode existir sem o outro. ✓
- C) Um método para compactar código. ✓
- D) Um relacionamento onde uma classe implementa outra. ~~B~~ ✓
- E) Um tipo especial de herança.

45. Como a agregação é representada em UML?

- A) Uma linha reta simples.
- ~~B) Um losango vazio.~~
- C) Um losango preenchido.
- D) Um triângulo na ponta da linha.
- E) Um círculo ligado à classe.

46. O que significa multiplicidade em um diagrama de classes?

- ~~A) O número de objetos que podem estar associados a outra classe.~~
- B) O número de métodos dentro de uma classe. ✓
- C) A quantidade de atributos que uma classe pode ter. ✓
- D) O nível de herança entre classes. ✓
- E) A repetição de código dentro da classe. ✓

47. Qual das opções representa uma multiplicidade "um para muitos" (1..*)?

- ~~A) Um livro pode ter vários autores.~~
- B) Um carro tem apenas um motor. ✓
- C) Uma cidade pertence a um estado. ✓
- D) Uma pessoa pode ter no máximo um CPF. ✓
- E) Uma casa tem um único endereço. ✓

48. O que significa a multiplicidade "0..1" em UML?

- A) O objeto deve obrigatoriamente estar associado a um outro.
- ~~B) O objeto pode ou não estar associado a outro.~~
- C) O objeto pode ter várias instâncias associadas. ✓
- D) O objeto nunca pode estar associado a outro. ✓
- E) O objeto pode se associar a qualquer outra classe. ✓

49. Quando uma classe pode se relacionar consigo mesma, isso é chamado de:

- A) Herança múltipla.
- ~~B) Relacionamento reflexivo.~~
- C) Composição hierárquica.
- D) Agregação encadeada.
- E) Multiplicidade especial.

50. Qual relação é usada quando um objeto pertence exclusivamente a outro e não pode existir separadamente?

- A) Associação.
- B) Herança.
- C) Agregação.
- ~~D) Composição.~~
- E) Dependência.

51. O que um caso de uso representa em UML?

- a) Uma entidade do sistema ✓
- b) Uma funcionalidade oferecida pelo sistema para um ator ✓
- c) Uma regra de negócio
- ~~d) Um diagrama de classes~~
- e) Um modelo de banco de dados ✓

52. Qual dos elementos abaixo NÃO faz parte de um diagrama de casos de uso? a) Ator .

- b) Caso de uso
- c) Associação
- ~~d) Diagrama de sequência~~
- e) Inclusão (<>) .

53. No diagrama de casos de uso, um ator representa:

- ~~a) Um usuário, sistema ou componente que interage com o sistema~~
- b) Uma funcionalidade do sistema ✓
- c) Uma entidade do banco de dados
- d) Um processo interno do sistema ✓
- e) Um estado do sistema

54. O que significa a relação de <> entre dois casos de uso?

- a) Representa um relacionamento opcional entre dois casos de uso ✓
- ~~b) Indica que um caso de uso sempre chama outro para executar uma funcionalidade comum~~
- c) Mostra que um caso de uso pode ser substituído por outro ✓
- d) Representa um ator secundário no diagrama ✓
- e) Indica que o caso de uso pode ser executado de maneira independente

55. Qual das opções a seguir é um exemplo válido de um ator em um diagrama de casos de uso?

- a) "Gerenciar Pagamento"
- ~~b) "Cliente"~~
- c) "Emitir Relatório"
- d) "Sistema de Banco de Dados"
- e) "Tela de Cadastro"

56. O relacionamento de <> é utilizado quando:

- a) Um caso de uso depende de outro para existir ✓

~~b) O fluxo principal do caso de uso pode ter uma funcionalidade opcional adicionada em determinados cenários ✓~~

- c) Dois casos de uso sempre precisam ser executados juntos
- d) Um ator pode desempenhar diferentes papéis no sistema ✓
- e) O sistema precisa acessar um banco de dados externo ✓

57. Em um diagrama de casos de uso, como representamos um ator?

- a) Com um retângulo
- b) Com um losango
- c) Com um círculo
- ~~d) Com um boneco estilizado (stick figure) e~~
- e) Com uma linha pontilhada

58. Quando é apropriado usar um diagrama de casos de uso?

- a) Para definir a estrutura de classes do sistema ✓
- ~~b) Para modelar a interação dos usuários e sistemas externos com o sistema~~
- c) Para modelar a estrutura física dos componentes do software ✓
- d) Para especificar a lógica interna dos algoritmos ✓
- e) Para projetar interfaces gráficas ✓

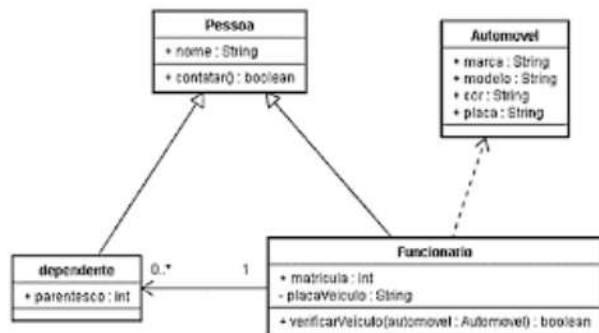
59. Se um ator A herda características de outro ator B, significa que:

- ~~a) O ator A pode realizar todos os casos de uso que o ator B pode~~
- b) O ator A substitui completamente o ator B ✓
- c) O ator A e o ator B são idênticos ✓
- d) O ator B pode interagir apenas com o ator A ✓
- e) O ator A se torna uma entidade do banco de dados ✓

60. No diagrama de casos de uso, um caso de uso pode ser representado por:

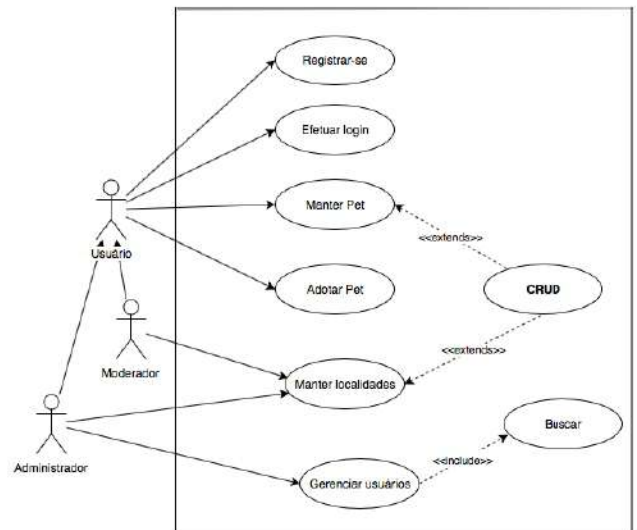
- a) Um círculo ou elipse
- b) Um retângulo
- c) Um hexágono
- d) Um losango
- e) Um paralelogramo

Sobre o Diagrama de Classe abaixo, responda V ou F nas perguntas 61 a 65.



- 61. O Funcionário é um tipo de dependente. F
- 62. A classe funcionário e automóvel é uma relação de dependência. V
- 63. O Funcionário pode ter nenhum dependente. V
- 64. O dependente pode ter vários Funcionários associados a ele. F
- 65. O dependente é um tipo de Pessoa. V

Sobre o Diagrama de Casos de Uso apresentado a seguir, responda V ou F nas perguntas 66 a 70.



- 66. O moderador pode Adotar Pet. V
- 67. O administrador pode efetuar Login V
- 68. Para o Usuário Manter pet ele obrigatoriamente tem que acessar o CRUD. F
- 69. O administrador para gerenciar usuário ele pode ou não Buscar. F
- 70. O administrador pode Registrar-se. V