Simulado PSOO

Parte 1 **Questões – POO e UML**

Parte 2 - **Questões – CICLO DE VIDA DE DESENVOLVIMENTO DE SOFTWARE**

1. A etapa de especificação de requisitos tem como objetivo principal:

a) Codificar o sistema em uma linguagem de programação.

b) Criar o banco de dados do sistema.

c) Definir claramente as necessidades e expectativas dos stakeholders.

d) Montar os diagramas UML.

e) Elaborar casos de teste automatizados.

1. A etapa de especificação de requisitos tem como objetivo principal:

a) Codificar o sistema em uma linguagem de programação.

b) Criar o banco de dados do sistema.

c) Definir claramente as necessidades e expectativas dos stakeholders.

d) Montar os diagramas UML.

e) Elaborar casos de teste automatizados.

Gabarito: c

Justificativa: O objetivo da especificação de requisitos é justamente levantar, documentar e esclarecer o que os usuários e interessados esperam do sistema, transformando essas necessidades em requisitos funcionais e não funcionais.

2. Qual das opções representa uma atividade da gestão de requisitos?

a) Criar algoritmos eficientes.

b) Definir a arquitetura do sistema.

c) Identificar e rastrear modificações de requisitos.

d) Construir protótipos navegáveis.

e) Implementar testes unitários.

2. Qual das opções representa uma atividade da gestão de requisitos?

a) Criar algoritmos eficientes.

b) Definir a arquitetura do sistema.

c) Identificar e rastrear modificações de requisitos.

d) Construir protótipos navegáveis.

e) Implementar testes unitários.

Gabarito: c

Justificativa: A gestão de requisitos tem como funções principais: controlar, rastrear e gerenciar mudanças nos requisitos ao longo do ciclo de vida do software.

3. A UML é definida como:

a) Uma linguagem de programação orientada a objetos.

b) Um conjunto de ferramentas CASE para gerar código.

c) Uma linguagem de modelagem padronizada e de uso geral.

d) Um padrão de banco de dados relacional.

e) Um protocolo de comunicação entre sistemas.

3. A UML é definida como:

a) Uma linguagem de programação orientada a objetos.

b) Um conjunto de ferramentas CASE para gerar código.

c) Uma linguagem de modelagem padronizada e de uso geral.

d) Um padrão de banco de dados relacional.

e) Um protocolo de comunicação entre sistemas.

Gabarito: c

Justificativa: A UML (Unified Modeling Language) é justamente isso: uma linguagem padronizada pelo OMG (Object Management Group), de uso geral, para modelar sistemas orientados a objetos e até outros tipos de sistemas.

4. Quantos tipos de diagramas UML existem, segundo a especificação do Object Management Group (OMG)?

a) 5

b) 8

c) 10

d) 14

e) 20

4. Quantos tipos de diagramas UML existem, segundo a especificação do Object Management Group (OMG)?

a) 5

b) 8

c) 10

d) 14

e) 20

Gabarito: d

Justificativa: De acordo com a **UML 2.5** (especificação do OMG), existem **14 tipos de diagramas**, divididos em **7 estruturais** e **7 comportamentais**.

5. O diagrama de casos de uso tem como função principal:

a) Representar o fluxo de execução de métodos.

b) Identificar atores e funcionalidades do sistema.

c) Mostrar a estrutura estática do sistema.

d) Representar a sequência de mensagens trocadas.

e) Definir os atributos das classes.

5. O diagrama de casos de uso tem como função principal:

a) Representar o fluxo de execução de métodos.

b) Identificar atores e funcionalidades do sistema.

c) Mostrar a estrutura estática do sistema.

d) Representar a sequência de mensagens trocadas.

e) Definir os atributos das classes.

Gabarito: b

Justificativa: O **diagrama de casos de uso** serve para **mostrar os atores (usuários ou sistemas externos)** e as **funcionalidades (casos de uso)** que o sistema deve oferecer. É muito utilizado na fase de **levantamento de requisitos**.

6. No diagrama de casos de uso, os atores representam:

a) Somente usuários humanos.

b) Apenas desenvolvedores do sistema.

c) Qualquer elemento externo que interage com o sistema.

d) Somente dispositivos de hardware.

e) Apenas clientes finais.

6. No diagrama de casos de uso, os atores representam:

a) Somente usuários humanos.

b) Apenas desenvolvedores do sistema.

c) Qualquer elemento externo que interage com o sistema.

d) Somente dispositivos de hardware.

e) Apenas clientes finais.

Gabarito: c

Justificativa: Num caso de uso, um ator representa uma **entidade externa ao sistema** – seja uma **pessoa** (humana) ou um **sistema externo/componente** – que interage com o sistema para atingir um objetivo. Atuam como iniciadores ou respondedores de um caso de uso, definindo o que o sistema deve fazer e servindo para delimitar o escopo do sistema.

7. No relacionamento <<include>> entre casos de uso:

a) O caso de uso é opcional.

b) O caso de uso incluído só ocorre se houver condição satisfeita.

c) A execução de um caso obriga a execução do outro.

d) Não existe obrigatoriedade de execução.

e) Ocorre apenas quando há generalização.

7. No relacionamento <<include>> entre casos de uso:

a) O caso de uso é opcional.

b) O caso de uso incluído só ocorre se houver condição satisfeita.

c) A execução de um caso obriga a execução do outro.

d) Não existe obrigatoriedade de execução.

e) Ocorre apenas quando há generalização.

Gabarito: c

8. O relacionamento <<extend>> em casos de uso indica:

a) Obrigatoriedade de execução de dois casos de uso.

b) Cenário opcional, executado apenas em determinadas condições.

c) Exclusão de um caso de uso.

d) Substituição de atores.

e) Um tipo de herança entre casos de uso.

8. O relacionamento <<extend>> em casos de uso indica:

a) Obrigatoriedade de execução de dois casos de uso.

b) Cenário opcional, executado apenas em determinadas condições.

c) Exclusão de um caso de uso.

d) Substituição de atores.

e) Um tipo de herança entre casos de uso.

Gabarito: b

9. O diagrama de atividades é uma versão avançada de:

a) Organogramas.

b) Mapas conceituais.

c) Fluxogramas.

d) Diagramas de classe.

e) Diagramas de sequência.

9. O diagrama de atividades é uma versão avançada de:

a) Organogramas.

b) Mapas conceituais.

c) Fluxogramas.

d) Diagramas de classe.

e) Diagramas de sequência.

Gabarito: c

10. A classificação em POO refere-se a:

a) Criar algoritmos de ordenação.

b) Agrupar objetos com características semelhantes.

c) Definir regras de herança múltipla.

d) Especificar atributos de banco de dados.

e) Construir diagramas de sequência.

10. A classificação em POO refere-se a:

a) Criar algoritmos de ordenação.

b) Agrupar objetos com características semelhantes.

c) Definir regras de herança múltipla.

d) Especificar atributos de banco de dados.

e) Construir diagramas de sequência.

Gabarito: b

11. Uma classe representa:

a) Uma instância de objeto.

b) Um algoritmo em execução.

c) A descrição de um conjunto de objetos do mundo real.

d) Apenas entidades abstratas.

e) O comportamento dinâmico do sistema.

11. Uma classe representa:

a) Uma instância de objeto.

b) Um algoritmo em execução.

c) A descrição de um conjunto de objetos do mundo real.

d) Apenas entidades abstratas.

e) O comportamento dinâmico do sistema.

Gabarito: c

12. Um objeto pode ser:

a) Apenas uma entidade física.

b) Apenas uma entidade lógica.

c) Entidade física, conceitual ou de software.

d) Somente uma variável em memória.

e) Apenas um processo em execução.

12. Um objeto pode ser:

a) Apenas uma entidade física.

b) Apenas uma entidade lógica.

c) Entidade física, conceitual ou de software.

d) Somente uma variável em memória.

e) Apenas um processo em execução.

Gabarito: c

13. No diagrama de classes, a representação gráfica é feita em:

a) Círculos.

b) Retângulos.

c) Elipses.

d) Losangos.

e) Triângulos.

13. No diagrama de classes, a representação gráfica é feita em:

a) Círculos.

b) Retângulos.

c) Elipses.

d) Losangos.

e) Triângulos.

Gabarito: b

14a. O modificador de visibilidade “+” em UML indica:

a) Privado.

b) Protegido.

c) Público.

d) Padrão.

e) Estático.

14a. O modificador de visibilidade “+” em UML indica:

a) Privado.

b) Protegido.

c) Público.

d) Padrão.

e) Estático.

Gabarito: c

14b. Com base na imagem, qual das alternativas identifica corretamente os modificadores de acesso utilizados na classe Pessoa?

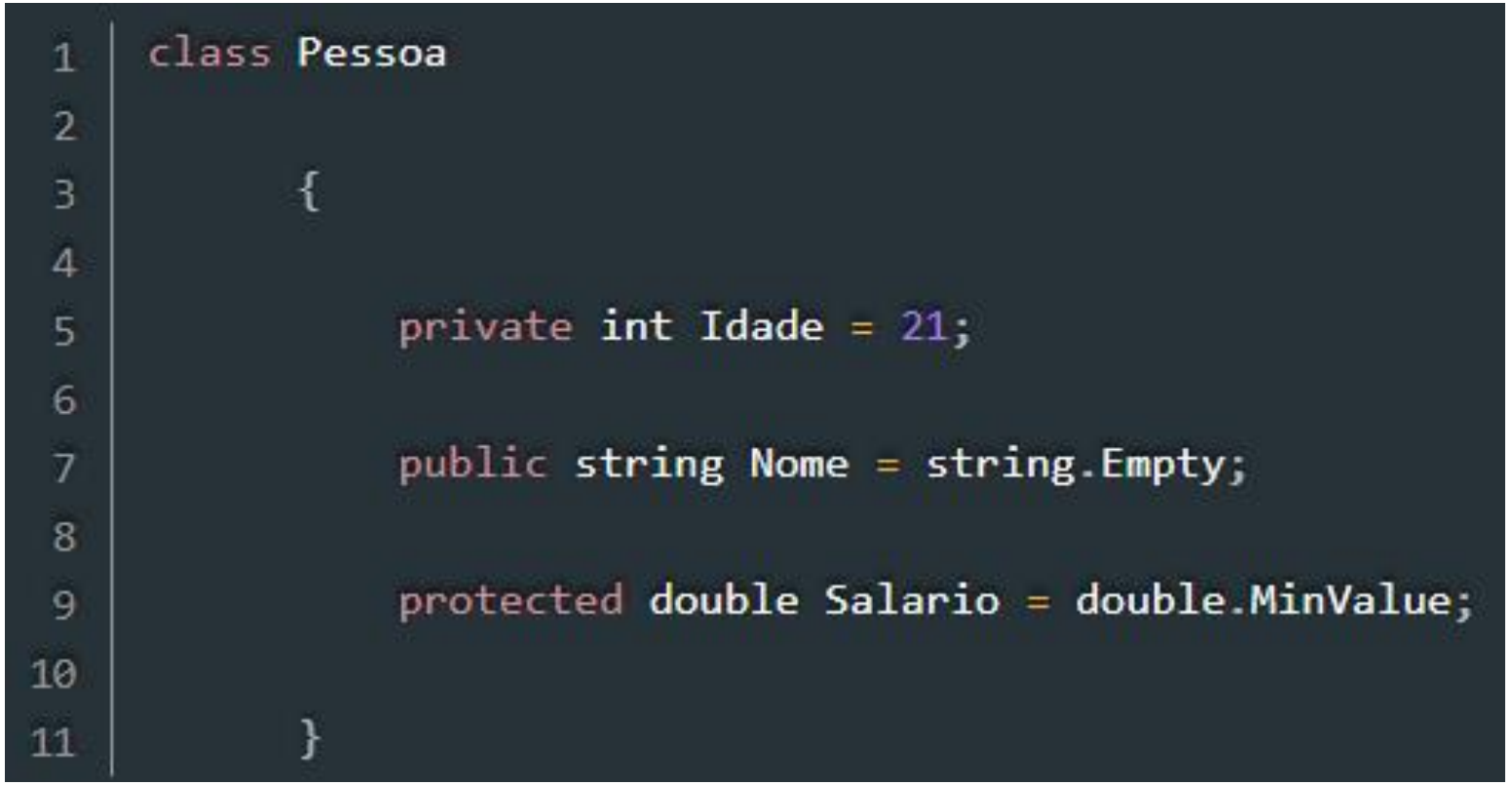
a) private, public, protected
b) public, string, protected
c) Idade, Nome, Salario
d) class, int, string
e) Pessoa, public, private

```
1  class Pessoa
2
3      {
4
5          private int Idade = 21;
6
7          public string Nome = string.Empty;
8
9          protected double Salario = double.MinValue;
10
11     }
```

14b. Com base na imagem, qual das alternativas identifica corretamente os modificadores de acesso utilizados na classe Pessoa?

a) private, public, protected
b) public, string, protected
c) Idade, Nome, Salario
d) class, int, string
e) Pessoa, public, private

Gabarito: a



```
1  class Pessoa
2
3      {
4
5          private int Idade = 21;
6
7          public string Nome = string.Empty;
8
9          protected double Salario = double.MinValue;
10
11     }
```

15. O conceito de herança em POO significa:

a) Classes podem se associar sem compartilhar atributos.

b) Subclasses herdam atributos e métodos da superclasse.

c) Um objeto pode ser convertido em classe.

d) Métodos não podem ser sobrescritos.

e) Instâncias podem criar novas classes.

15. O conceito de herança em POO significa:

a) Classes podem se associar sem compartilhar atributos.

b) Subclasses herdam atributos e métodos da superclasse.

c) Um objeto pode ser convertido em classe.

d) Métodos não podem ser sobrescritos.

e) Instâncias podem criar novas classes.

Gabarito: b

16. O polimorfismo está relacionado principalmente a:

a) Duplicação de código.

b) Redefinição de métodos herdados.

c) Criação de atributos privados.

d) Exclusão de herança.

e) Encapsulamento de dados.

16. O polimorfismo está relacionado principalmente a:

a) Duplicação de código.

b) Redefinição de métodos herdados.

c) Criação de atributos privados.

d) Exclusão de herança.

e) Encapsulamento de dados.

Gabarito: b

17. O encapsulamento em POO tem como finalidade:

a) Tornar o código mais rígido.

b) Garantir que todos os atributos sejam públicos.

c) Isolar partes do programa, facilitando alterações e manutenção.

d) Impedir a criação de subclasses.

e) Rejeitar uso de métodos polimórficos.

17. O encapsulamento em POO tem como finalidade:

a) Tornar o código mais rígido.

b) Garantir que todos os atributos sejam públicos.

c) Isolar partes do programa, facilitando alterações e manutenção.

d) Impedir a criação de subclasses.

e) Rejeitar uso de métodos polimórficos.

Gabarito: c

18. No diagrama de classes, a relação de composição indica:

a) Um objeto pode existir independentemente do outro.

b) O objeto só existe enquanto o outro existir.

c) Um relacionamento transitório e temporário.

d) Uma associação entre métodos estáticos.

e) Exclusão de herança múltipla.

18. No diagrama de classes, a relação de composição indica:

a) Um objeto pode existir independentemente do outro.

b) O objeto só existe enquanto o outro existir.

c) Um relacionamento transitório e temporário.

d) Uma associação entre métodos estáticos.

e) Exclusão de herança múltipla.

Gabarito: b

19. O diagrama de sequência tem como característica principal:

a) Enfatizar a estrutura estática do sistema.

b) Mostrar a sequência temporal de interações entre objetos.

c) Substituir o DER em banco de dados.

d) Definir os atributos das classes.

e) Representar apenas fluxos paralelos.

19. O diagrama de sequência tem como característica principal:

a) Enfatizar a estrutura estática do sistema.

b) Mostrar a sequência temporal de interações entre objetos.

c) Substituir o DER em banco de dados.

d) Definir os atributos das classes.

e) Representar apenas fluxos paralelos.

Gabarito: b

20. No DER (Diagrama Entidade-Relacionamento), a cardinalidade 0..* significa:

a) Exatamente um.

b) Um ou mais.

c) Zero ou mais.

d) Zero ou um.

e) Faixa de valores de 4 a 7.

20. No DER (Diagrama Entidade-Relacionamento), a cardinalidade 0..* significa:

a) Exatamente um.

b) Um ou mais.

c) Zero ou mais.

d) Zero ou um.

e) Faixa de valores de 4 a 7.

Gabarito: c

21. O ciclo de vida de software pode ser definido como:

a) Apenas a fase de desenvolvimento e testes de um sistema.

b) A sequência de atividades exclusivamente de codificação de software.

c) O conjunto de etapas desde a concepção até a desativação do sistema.

d) O processo de manutenção contínua sem fases intermediárias.

e) A implementação prática de apenas requisitos funcionais.

21. O ciclo de vida de software pode ser definido como:

a) Apenas a fase de desenvolvimento e testes de um sistema.

b) A sequência de atividades exclusivamente de codificação de software.

c) O conjunto de etapas desde a concepção até a desativação do sistema.

d) O processo de manutenção contínua sem fases intermediárias.

e) A implementação prática de apenas requisitos funcionais.

Gabarito: c

22. De acordo com a NBR ISO/IEC 12207:1998, o ciclo de vida de software corresponde a:

a) Uma lista de técnicas de programação que reduzem o custo do projeto.

b) A estrutura contendo processos, atividades e tarefas do desenvolvimento à descontinuação do software.

c) O modelo de prototipagem aplicado ao processo de requisitos.

d) Um conjunto de regras que determinam a linguagem de programação usada.

e) O uso obrigatório de metodologias ágeis em todo projeto de software.

22. De acordo com a NBR ISO/IEC 12207:1998, o ciclo de vida de software corresponde a:

a) Uma lista de técnicas de programação que reduzem o custo do projeto.

b) A estrutura contendo processos, atividades e tarefas do desenvolvimento à descontinuação do software.

c) O modelo de prototipagem aplicado ao processo de requisitos.

d) Um conjunto de regras que determinam a linguagem de programação usada.

e) O uso obrigatório de metodologias ágeis em todo projeto de software.

Gabarito: b

23. No modelo Cascata (Waterfall), as fases do projeto:

a) São realizadas de forma paralela, com entregas incrementais.

b) Permitem mudanças frequentes e de baixo custo.

c) Dependem apenas de reuniões diárias com o cliente.

d) Ocorrem sem necessidade de documentação formal.

e) São executadas de forma linear e sequencial, sem revisões anteriores.

23. No modelo Cascata (Waterfall), as fases do projeto:

a) São realizadas de forma paralela, com entregas incrementais.

b) Permitem mudanças frequentes e de baixo custo.

c) Dependem apenas de reuniões diárias com o cliente.

d) Ocorrem sem necessidade de documentação formal.

e) São executadas de forma linear e sequencial, sem revisões anteriores.

Gabarito: e

24. Uma desvantagem do modelo em Cascata é:

a) Ser pouco formalizado e não exigir planejamento.

b) Dificuldade em lidar com mudanças de requisitos após o início do desenvolvimento.

c) Exigir a participação constante do cliente em todo o ciclo.

d) A inexistência de estimativas de custo no início.

e) A entrega em ciclos curtos, que dificulta testes.

24. Uma desvantagem do modelo em Cascata é:

a) Ser pouco formalizado e não exigir planejamento.

b) Dificuldade em lidar com mudanças de requisitos após o início do desenvolvimento.

c) Exigir a participação constante do cliente em todo o ciclo.

d) A inexistência de estimativas de custo no início.

e) A entrega em ciclos curtos, que dificulta testes.

Gabarito: b

25. No modelo Ágil, o desenvolvimento é caracterizado por:

a) Rigidez na definição inicial de requisitos.

b) Entrega de software somente ao final do projeto.

c) Iteratividade, colaboração e flexibilidade para mudanças.

d) Planejamento único e inalterável durante o ciclo.

e) Uso de documentação pesada para validação.

25. No modelo Ágil, o desenvolvimento é caracterizado por:

a) Rigidez na definição inicial de requisitos.

b) Entrega de software somente ao final do projeto.

c) Iteratividade, colaboração e flexibilidade para mudanças.

d) Planejamento único e inalterável durante o ciclo.

e) Uso de documentação pesada para validação.

Gabarito: c

26. Uma característica fundamental das metodologias ágeis é:

a) O cliente participa apenas no início do processo.

b) A entrega é feita em uma única versão final.

c) A ênfase está no software funcional em vez da documentação detalhada.

d) O processo não admite erros.

e) O fluxo de atividades é sempre linear e sem retorno.

26. Uma característica fundamental das metodologias ágeis é:

a) O cliente participa apenas no início do processo.

b) A entrega é feita em uma única versão final.

c) A ênfase está no software funcional em vez da documentação detalhada.

d) O processo não admite erros.

e) O fluxo de atividades é sempre linear e sem retorno.

Gabarito: c

27. Na comparação entre Cascata e Ágil, pode-se afirmar que:

a) Cascata aceita mudanças de requisitos a qualquer momento sem custo.

b) Ágil concentra a participação do cliente apenas na fase inicial.

c) Cascata é iterativo e incremental, enquanto Ágil é sequencial.

d) Cascata prioriza documentação, enquanto Ágil prioriza entregas incrementais.

e) Ambos não exigem planejamento inicial.

27. Na comparação entre Cascata e Ágil, pode-se afirmar que:

a) Cascata aceita mudanças de requisitos a qualquer momento sem custo.

b) Ágil concentra a participação do cliente apenas na fase inicial.

c) Cascata é iterativo e incremental, enquanto Ágil é sequencial.

d) Cascata prioriza documentação, enquanto Ágil prioriza entregas incrementais.

e) Ambos não exigem planejamento inicial.

Gabarito: d

28. O modelo de Prototipagem é indicado quando:

a) Os requisitos estão totalmente claros e definidos.

b) O cliente não sabe expressar exatamente suas necessidades.

c) A equipe deseja evitar contato com o usuário.

d) O projeto é simples e sem mudanças previstas.

e) Não se deseja o feedback do usuário.

28. O modelo de Prototipagem é indicado quando:

a) Os requisitos estão totalmente claros e definidos.

b) O cliente não sabe expressar exatamente suas necessidades.

c) A equipe deseja evitar contato com o usuário.

d) O projeto é simples e sem mudanças previstas.

e) Não se deseja o feedback do usuário.

Gabarito: b

29. Um risco da prototipagem é:

a) O cliente se basear em desempenho irreal, já que protótipos muitas vezes não acessam bases reais.

b) A impossibilidade de descartar protótipos.

c) O excesso de documentação gerada.

d) A ausência total de participação do cliente.

e) O uso restrito a sistemas pequenos.

29. Um risco da prototipagem é:

a) O cliente se basear em desempenho irreal, já que protótipos muitas vezes não acessam bases reais.

b) A impossibilidade de descartar protótipos.

c) O excesso de documentação gerada.

d) A ausência total de participação do cliente.

e) O uso restrito a sistemas pequenos.

Gabarito: a

30. O RUP (Rational Unified Process) pode ser caracterizado como:

a) Um processo linear sem iterações.

b) Um modelo iterativo e incremental orientado a casos de uso.

c) Um ciclo de vida sem documentação associada.

d) Uma técnica de prototipagem descartável.

e) Uma adaptação do modelo cascata sem fases distintas.

30. O RUP (Rational Unified Process) pode ser caracterizado como:

a) Um processo linear sem iterações.

b) Um modelo iterativo e incremental orientado a casos de uso.

c) Um ciclo de vida sem documentação associada.

d) Uma técnica de prototipagem descartável.

e) Uma adaptação do modelo cascata sem fases distintas.

Gabarito: b

31. As quatro fases principais do RUP são:

a) Planejamento, Análise, Programação e Testes.

b) Concepção, Elaboração, Construção e Transição.

c) Escopo, Desenvolvimento, Entrega e Descontinuidade.

d) Requisitos, Implementação, Avaliação e Suporte.

e) Iteração, Incremento, Feedback e Finalização.

31. As quatro fases principais do RUP são:

a) Planejamento, Análise, Programação e Testes.

b) Concepção, Elaboração, Construção e Transição.

c) Escopo, Desenvolvimento, Entrega e Descontinuidade.

d) Requisitos, Implementação, Avaliação e Suporte.

e) Iteração, Incremento, Feedback e Finalização.

Gabarito: b

32. No RUP, a fase de Concepção tem como objetivo:

a) Desenvolver código-fonte.

b) Finalizar a documentação de entrega.

c) Elaborar a arquitetura detalhada do sistema.

d) Definir o escopo do projeto e avaliar se deve prosseguir.

e) Realizar testes finais.

32. No RUP, a fase de Concepção tem como objetivo:

a) Desenvolver código-fonte.

b) Finalizar a documentação de entrega.

c) Elaborar a arquitetura detalhada do sistema.

d) Definir o escopo do projeto e avaliar se deve prosseguir.

e) Realizar testes finais.

Gabarito: d

33. Sobre as fases de um projeto no modelo Cascata, assinale a correta:

a) Implementação ocorre antes do projeto.

b) A análise de requisitos é realizada após os testes.

c) A integração ocorre após a implementação.

d) Testes são feitos antes da definição de requisitos.

e) Projeto e análise ocorrem simultaneamente.

33. Sobre as fases de um projeto no modelo Cascata, assinale a correta:

a) Implementação ocorre antes do projeto.

b) A análise de requisitos é realizada após os testes.

c) A integração ocorre após a implementação.

d) Testes são feitos antes da definição de requisitos.

e) Projeto e análise ocorrem simultaneamente.

Gabarito: c

34. Um benefício do modelo Cascata é:

a) Maior facilidade para mudanças durante o desenvolvimento.

b) Simplicidade de entendimento e estimativas iniciais de custo.

c) Garantia de ausência de erros nos requisitos.

d) Eliminação da necessidade de documentação formal.

e) Redução do tempo total de desenvolvimento.

Gabarito: b

34. Um benefício do modelo Cascata é:

a) Maior facilidade para mudanças durante o desenvolvimento.

b) Simplicidade de entendimento e estimativas iniciais de custo.

c) Garantia de ausência de erros nos requisitos.

d) Eliminação da necessidade de documentação formal.

e) Redução do tempo total de desenvolvimento.

Gabarito: b

35. No ciclo de vida Ágil, os sprints geralmente têm duração de:

a) Entre 1 e 2 dias úteis.

b) Entre 2 e 4 semanas.

c) Entre 6 e 12 meses.

d) Entre 3 e 6 meses.

e) 1 ano.

35. No ciclo de vida Ágil, os sprints geralmente têm duração de:

a) Entre 1 e 2 dias úteis.

b) Entre 2 e 4 semanas.

c) Entre 6 e 12 meses.

d) Entre 3 e 6 meses.

e) 1 ano.

Gabarito: b

36. Durante o sprint em metodologias ágeis, é prática comum:

a) Eliminar o contato com o cliente até o fim da entrega.

b) Revisar diariamente o andamento com reuniões rápidas.

c) Produzir apenas documentação, sem código funcional.

d) Executar todo o projeto sem testes intermediários.

e) Congelar requisitos desde o início.

36. Durante o sprint em metodologias ágeis, é prática comum:

a) Eliminar o contato com o cliente até o fim da entrega.

b) Revisar diariamente o andamento com reuniões rápidas.

c) Produzir apenas documentação, sem código funcional.

d) Executar todo o projeto sem testes intermediários.

e) Congelar requisitos desde o início.

Gabarito: b

37. A fase de Descontinuação em um ciclo de vida Ágil ocorre quando:

a) O sistema está em fase inicial de testes.

b) O software deve ser substituído ou tornou-se obsoleto.

c) O cliente solicita uma nova funcionalidade.

d) O produto é implantado pela primeira vez.

e) A equipe define requisitos de negócio.

37. A fase de Descontinuação em um ciclo de vida Ágil ocorre quando:

a) O sistema está em fase inicial de testes.

b) O software deve ser substituído ou tornou-se obsoleto.

c) O cliente solicita uma nova funcionalidade.

d) O produto é implantado pela primeira vez.

e) A equipe define requisitos de negócio.

Gabarito: b

38. Qual dos modelos é mais apropriado para sistemas complexos, com alta incerteza de requisitos?

a) Cascata.

b) Modelo em Blocos.

c) Modelo Linear Sequencial.

d) Prototipagem.

e) Nenhum, pois todos exigem requisitos claros desde o início.

38. Qual dos modelos é mais apropriado para sistemas complexos, com alta incerteza de requisitos?

a) Cascata.

b) Modelo em Blocos.

c) Modelo Linear Sequencial.

d) Prototipagem.

e) Nenhum, pois todos exigem requisitos claros desde o início.

Gabarito: d

39. Um processo de software pode ser entendido como:

a) Uma linguagem de programação utilizada para implementar sistemas.

b) Conjunto de atividades que constituem o desenvolvimento de um sistema computacional.

c) Apenas a etapa de análise de requisitos.

d) A execução de testes automatizados.

e) Um método exclusivo de manutenção de sistemas legados.

39. Um processo de software pode ser entendido como:

a) Uma linguagem de programação utilizada para implementar sistemas.

b) Conjunto de atividades que constituem o desenvolvimento de um sistema computacional.

c) Apenas a etapa de análise de requisitos.

d) A execução de testes automatizados.

e) Um método exclusivo de manutenção de sistemas legados.

Gabarito: b

40. Sobre modelos de ciclo de vida:

a) São estruturas pré-definidas que organizam as fases de um processo de software.

b) São apenas metodologias ágeis.

c) Exigem o uso de protótipos descartáveis obrigatoriamente.

d) São usados apenas em projetos pequenos.

e) Impedem a documentação de requisitos.

40. Sobre modelos de ciclo de vida:

a) São estruturas pré-definidas que organizam as fases de um processo de software.

b) São apenas metodologias ágeis.

c) Exigem o uso de protótipos descartáveis obrigatoriamente.

d) São usados apenas em projetos pequenos.

e) Impedem a documentação de requisitos.

Gabarito: a

41. (POO/UML)

Uma empresa de logística deseja informatizar o processo de rastreamento de encomendas. Durante o levantamento de requisitos, foram identificados os seguintes atores: Cliente, Atendente e Sistema de Rastreamento Externo. O cliente poderá consultar o status da encomenda, enquanto o atendente poderá registrar reclamações e o sistema externo informará a localização atual do pacote.

Considerando a UML, o diagrama mais adequado para representar esse cenário:

a) Diagrama de Classes.

b) Diagrama de Casos de Uso.

c) Diagrama de Sequência.

d) Diagrama de Atividades.

e) Diagrama Entidade-Relacionamento.

41. (POO/UML)

Uma empresa de logística deseja informatizar o processo de rastreamento de encomendas. Durante o levantamento de requisitos, foram identificados os seguintes atores: Cliente, Atendente e Sistema de Rastreamento Externo. O cliente poderá consultar o status da encomenda, enquanto o atendente poderá registrar reclamações e o sistema externo informará a localização atual do pacote.

Considerando a UML, o diagrama mais adequado para representar esse cenário:

a) Diagrama de Classes.

b) Diagrama de Casos de Uso.

c) Diagrama de Sequência.

d) Diagrama de Atividades.

e) Diagrama Entidade-Relacionamento.

Gabarito: b

42. (Gestão de Requisitos)

No processo de especificação de software para um sistema bancário, foi definido que a operação “realizar saque” sempre gera automaticamente a operação “registrar transação”. Esse tipo de relacionamento entre casos de uso é representado em UML como:

a) Generalização.

b) Especialização.

c) <<extend>>.

d) <<include>>.

e) Associação simples.

42. (Gestão de Requisitos)

No processo de especificação de software para um sistema bancário, foi definido que a operação “realizar saque” sempre gera automaticamente a operação “registrar transação”. Esse tipo de relacionamento entre casos de uso é representado em UML como:

a) Generalização.

b) Especialização.

c) <<extend>>.

d) <<include>>.

e) Associação simples.

Gabarito: d

43. Uma equipe de desenvolvimento foi questionada sobre falhas na segurança de um sistema, pois os atributos das classes estavam sendo acessados diretamente por qualquer outra classe. O analista sugeriu o uso de modificadores de visibilidade em UML para restringir o acesso e permitir apenas a manipulação por meio de métodos.

Essa recomendação refere-se ao princípio da orientação a objetos chamado:

a) Herança.

b) Polimorfismo.

c) Encapsulamento.

d) Abstração.

e) Generalização.

43. Uma equipe de desenvolvimento foi questionada sobre falhas na segurança de um sistema, pois os atributos das classes estavam sendo acessados diretamente por qualquer outra classe. O analista sugeriu o uso de modificadores de visibilidade em UML para restringir o acesso e permitir apenas a manipulação por meio de métodos.

Essa recomendação refere-se ao princípio da orientação a objetos chamado:

a) Herança.

b) Polimorfismo.

c) Encapsulamento.

d) Abstração.

e) Generalização.

Gabarito: c

44. Uma empresa de e-commerce deseja modelar o processo de atendimento de pedidos. Assim que o pedido é confirmado, duas atividades ocorrem em paralelo: separação do produto e emissão da nota fiscal. Somente após ambas as atividades serem concluídas, o processo segue para a etapa de envio ao cliente.

O diagrama UML mais adequado para modelar essa situação é:

a) Diagrama de Casos de Uso.

b) Diagrama de Classes.

c) Diagrama de Sequência.

d) Diagrama de Atividades.

e) Diagrama Entidade-Relacionamento.

44. Uma empresa de e-commerce deseja modelar o processo de atendimento de pedidos. Assim que o pedido é confirmado, duas atividades ocorrem em paralelo: separação do produto e emissão da nota fiscal. Somente após ambas as atividades serem concluídas, o processo segue para a etapa de envio ao cliente.

O diagrama UML mais adequado para modelar essa situação é:

a) Diagrama de Casos de Uso.

b) Diagrama de Classes.

c) Diagrama de Sequência.

d) Diagrama de Atividades.

e) Diagrama Entidade-Relacionamento.

Gabarito: d

45. Considere um sistema para um zoológico que possui uma classe Animal, da qual derivam as classes Leão, Garça e Coala. Todas herdam os métodos da classe Animal, mas cada uma redefine o método emitirSom() com uma implementação específica.

Esse cenário ilustra os conceitos de:

a) Encapsulamento e Abstração.

b) Herança e Polimorfismo.

c) Generalização e Associação.

d) Composição e Dependência.

e) Agregação e Multiplicidade.

45. Considere um sistema para um zoológico que possui uma classe Animal, da qual derivam as classes Leão, Garça e Coala. Todas herdam os métodos da classe Animal, mas cada uma redefine o método emitirSom() com uma implementação específica.

Esse cenário ilustra os conceitos de:

a) Encapsulamento e Abstração.

b) Herança e Polimorfismo.

c) Generalização e Associação.

d) Composição e Dependência.

e) Agregação e Multiplicidade.

Gabarito: b

46. (DER – Cardinalidade)

Em um sistema acadêmico, um professor pode ministrar várias disciplinas, mas cada disciplina deve ser obrigatoriamente ministrada por apenas um professor. No diagrama Entidade-Relacionamento, essa relação é expressa como:

a) 1:1

b) 0:1

c) 1:N

d) N:M

e) 0:N

46. (DER – Cardinalidade)

Em um sistema acadêmico, um professor pode ministrar várias disciplinas, mas cada disciplina deve ser obrigatoriamente ministrada por apenas um professor. No diagrama Entidade-Relacionamento, essa relação é expressa como:

a) 1:1

b) 0:1

c) 1:N

d) N:M

e) 0:N

Gabarito: c

47. (Integração POO e UML)

Durante o desenvolvimento de um sistema de vendas, o analista optou por iniciar a modelagem com um diagrama de casos de uso para compreender as interações externas, depois passou para o diagrama de classes para estruturar entidades e, por fim, utilizou o diagrama de sequência para detalhar a comunicação entre objetos.

Essa prática representa:

a) A aplicação de diferentes diagramas UML ao projeto.

b) A substituição de diagramas desnecessários.

c) A eliminação da necessidade de documentação textual.

d) A modelagem orientada a processos de negócio.

e) O uso exclusivo de diagramas estruturais.

47. (Integração POO e UML)

Durante o desenvolvimento de um sistema de vendas, o analista optou por iniciar a modelagem com um diagrama de casos de uso para compreender as interações externas, depois passou para o diagrama de classes para estruturar entidades e, por fim, utilizou o diagrama de sequência para detalhar a comunicação entre objetos.

Essa prática representa:

a) A aplicação de diferentes diagramas UML ao projeto.

b) A substituição de diagramas desnecessários.

c) A eliminação da necessidade de documentação textual.

d) A modelagem orientada a processos de negócio.

e) O uso exclusivo de diagramas estruturais.

Gabarito: a

Até a próxima