

Disciplina: **MODELAGEM DE SISTEMAS EM UML**Aluno: **KATIA REJANE RABELO SILVA****AV****202305362843**Turma: **9001**

DGT1343_AV_202305362843 (AG)

06/02/2024 18:11:29 (F)

Avaliação: 7,00 pts**Nota SIA: 7,00 pts**

Dispositivo liberado pela Matrícula 202305362843 com o token 379666 em 06/02/2024 18:11:21.

**EM2120073 - CONCEITOS BÁSICOS DE MODELAGEM DE SISTEMAS****1.**

Ref.: 7689639

Pontos: 1,00 / 1,00

Modelos vêm ajudando cada vez mais as pessoas a desenvolverem suas ideias de forma gráfica e discutir facilmente com sua equipe e com os usuários.

Avalie as assertivas I e II, a seguir.

I. Uma realidade complexa exige maior número de perspectivas de análise do que um problema mais elementar.

PORQUE

II. Para entender a totalidade, precisamos enxergar o problema sob várias perspectivas.

E assinale a alternativa correta.

- ☐ As assertivas I e II são falsas.
- ☒ As assertivas I e II são verdadeiras e a assertiva II justifica a I.
- ☐ As assertivas I e II são verdadeiras e a assertiva II não justifica a I.
- ☐ A assertiva I é verdadeira e a assertiva II é falsa.
- ☐ A assertiva I é falsa e a assertiva II é verdadeira.

2.

Ref.: 4272282

Pontos: 1,00 / 1,00

A UML é uma linguagem unificada de modelagem que surgiu da união de alguns métodos de profissionais do mercado. Passou a ser adotada pela OMG e tornou-se uma linguagem universal, padronizada.

Sobre a UML, avalie as alternativas a seguir e assinale a ÚNICA que é falsa:

- ☐ A UML não determina os diagramas que devem ser usados e nem a ordem com que devem ser usados.
- ☐ A UML disponibiliza diagramas estruturais e comportamentais.
- ☐ A UML é independente de tecnologia.
- ☐ A UML qualifica-se para ser usada em todo processo de desenvolvimento de software.
- ☒ A UML tem restrições de uso, caso o sistema seja implementado em determinadas linguagens.

3.

Ref.: 7689477

Pontos: 1,00 / 1,00

Camadas de software ajudam na organização do código para que se possa ter uma manutenção mais saudável do mesmo. Nesse sentido, avalie as asserções a seguir.

I. Camadas separam as responsabilidades e gerenciam as dependências.

II. Inicialmente, todo o código ficava misturado numa única camada, onde misturavam-se comandos de

processamento, de construção e manipulação de interface, bem como de acesso e persistência de dados em SGBD.

III. A divisão em camadas foi uma solução encontrada para melhor organizar o código de um programa, facilitando sua manutenção.

IV. O modelo de duas camadas veio naturalmente com a tecnologia de redes cliente-servidor. São elas: a camada de código que roda no cliente (camada de interface com usuário) e a camada servidor (camadas de lógica do negócio e persistência dos dados).

V. Três camadas é o máximo possível em que se pode dividir um sistema.

Assinale a opção correta.

- ☐ II e III, apenas.
- ☐ III e V, apenas.
- ☐ I, II e III, apenas.
- ☐ I, apenas.
- ☒ I, II, III e IV.



EM2120105 - ESTUDO DE CASO DE MODELAGEM DE SISTEMAS EM UML

4.

Ref.: 4464267

Pontos: 1,00 / 1,00

(FAURGS – TJ-RS – 2018) Requisitos não funcionais - como o nome sugere - são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Podem ser provenientes das características requeridas para o software, da organização que desenvolve o software ou de fontes externas. Os requisitos não funcionais que especificam ou restringem o comportamento do software - por exemplo, o seu desempenho, seus requisitos de proteção, seus requisitos de usabilidade e a taxa aceitável de falhas - são denominados requisitos:

- ☒ de produto
- ☐ ambientais
- ☐ externos
- ☐ éticos
- ☐ organizacionais



EM2120370 - UTILIZANDO UML PARA PROJETAR O SOFTWARE

5.

Ref.: 5429522

Pontos: 1,00 / 1,00

O que é a visibilidade de um atributo no diagrama de classes?

- ☐ A visibilidade de um atributo indica a forma de acesso.
- ☐ A visibilidade de um atributo indica o tipo de acesso.
- ☒ A visibilidade de um atributo indica o seu nível de acesso.
- ☐ A visibilidade de um atributo indica a quantidade de acessos.
- ☐ A visibilidade de um atributo indica como o acesso acontece.

6.

Ref.: 7994747

Pontos: 1,00 / 1,00

Unified Modeling Language (UML) é uma linguagem padrão para a especificação, visualização, construção e documentação dos artefatos de sistemas de software. UML foi desenvolvida originalmente para ajudar a sistematizar a engenharia de software, especialmente no contexto da orientação a objetos.

Qual a notação da UML para representar uma dependência num diagrama de classes?

- ☐ Uma reta tracejada ligando as classes envolvidas.
- ☐ Uma reta cheia ligando as classes envolvidas.
- ☐ Uma seta cheia ligando as classes envolvidas.
- ☐ Uma reta curva ligando as classes envolvidas.
- ☒ Uma seta tracejada ligando as classes envolvidas.



EM2120539 - UML PARA MODELAGEM DO DOMÍNIO

7.

Ref.: 4416261

Pontos: 1,00 / 1,00

Fonte: UFU-MG - 2017 - UFU-MG - Analista de Tecnologia da Informação

O Diagrama de Casos de Uso da UML, apresentado a seguir, expressa que:



- ☐ O ator Gerente é uma extensão do ator Vendedor.
- ☐ O ator Vendedor pode aprovar o crediário de um cliente, se seu CPF for válido.
- ☐ O ator Gerente solicita ao ator Vendedor que cadastre um cliente.
- ☐ O ator Vendedor é uma especialização do ator Gerente.
- ☒ O ator Gerente pode realizar cadastros de clientes e aprovar crediários.

8.

Ref.: 4392274

Pontos: 0,00 / 1,00

Fonte: Adaptado de Q747519 - CESPE - 2018 - TCE-MG - Analista de Controle Externo - Ciência da Computação

Em um diagrama de caso de uso, o ator representa:

- ☒ papéis que humanos e outros sistemas podem assumir ao interagirem com o sistema.
- ☒ programadores que vão implementar os casos de uso.
- ☐ humanos específicos que interagem com o sistema.
- ☐ nomes das pessoas que interagem com o sistema.
- ☐ uma elipse e um rótulo com o nome do caso de uso.



EM2120540 - INTRODUÇÃO A PADRÕES DE PROJETO - DESIGN PATTERNS

9.

Ref.: 5429720

Pontos: 0,00 / 1,00

Um software deve ser integrado a diferentes API de reconhecimento de digitais oferecidas por diferentes fabricantes. Qual padrão podemos usar, de modo a permitir que os módulos desse software possam utilizar essas

diferentes API a partir de uma mesma interface?

- ☐ Strategy
- ☒ Factory Method
- ☐ Template Method
- ☐ Adapter
- ☐ Facade

10.

Ref.: 5429560

Pontos: 0,00 / 1,00

Assinale a alternativa com uma afirmativa verdadeira sobre o padrão GRASP Criador:

- ☒ Recomenda que as responsabilidades devem ser distribuídas pelos módulos de forma que as dependências entre eles sejam gerenciáveis.
- ☐ Indica a classe mais recomendada para instanciar um objeto em uma relação de agregado x partes.
- ☐ Evita a criação de dependências em relação a estruturas de código condicionais (if-then-else ou switch-case).
- ☐ Evita a duplicação de algoritmos que apresentem a mesma estrutura, com alguns pontos de variação entre eles.
- ☐ Recomenda que cada módulo deve ter uma responsabilidade bem-definida.