Exame de Época Normal – 2016-06-24. Duração: 90min.

Nas questões de escolha múltipla: responda diretamente no enunciado; assinale de forma inequívoca apenas uma opção (a mais correta); as respostas erradas descontam ½ da cotação; as respostas ambíguas não são corrigidas.

NOME: N.MEC:

B1.

Considere o requisito a seguir apresentado, relativo a um sistema de análises clínicas:

R_i: Não pode haver mais que 5 análises perdidas em 1000 por causa de falhas no software.

a) Não é um bom exemplo de um requisito, porque não é possível conhecer de antemão o número de falhas do sistema, antes de ele estar implementado.
b) É um requisito não-funcional, mas inadequado, porque não é verificável.
c) É um requisito funcional, mas inadequado, porque não é parametrizável para assumir outros valores.
d) É um bom exemplo de um atributo de qualidade, relacionado com o desempenho (performance) do sistema de software.
e) É um bom exemplo de um requisito não funcional, relacionado com a fiabilidade (reliability) do sistema de software.

B2.

Os Casos de Utilização…

f) fornecem uma representação visual de baixo nível dos requisitos dos utilizadores.
g) fornecem uma representação visual de alto nível dos requisitos dos utilizadores.
h) fornecem uma representação estruturada dos requisitos dos utilizadores.
i) fornecem uma representação dos requisitos dos utilizadores diretos do sistema.
j) Todas as opções anteriores estão corretas.

B3.

O modelo de Casos de Utilização…

a) inclui necessariamente um ou mais diagramas de pacotes.
b) inclui necessariamente um ou mais diagramas de casos de utilização.
c) pode captar casos de utilização sem usar diagramas de Casos de Utilização, mas deve incluir Diagramas de Atividade.
d) pode captar casos de utilização sem usar diagramas, recorrendo narrativas estruturadas.
e) Nenhuma opção anterior está correta.

B4.

A análise de requisitos orientada a cenários (de utilização) valoriza a caraterização de funcionalidades que os atores sabem que precisam/querem.

a) Isto ajuda a evitar a sobre-especificação de requisitos desnecessários, que podem parecer ser uma boa ideia, mas que ninguém vai usar, porque não estão relacionados com os objetivos dos utilizadores.
b) Isto é limitativo, porque apenas capta funcionalidades de que os utilizadores se lembram na altura da especificação.
c) Isto leva a especificações incompletas, porque não é possível identificar requisitos não funcionais.
d) Isto é limitativo, porque só utiliza um tipo de diagrama da UML.
e) Nenhuma opção anterior está correta.

B5.

Qual dos seguintes princípios **não** é caraterística de uma metodologia ágil de desenvolvimento de software?

a) Entrega frequente ao cliente de software a funcionar, ao longo do processo de desenvolvimento;
b) A alteração de requisitos é bem acolhida, independentemente do momento em que ocorrem.
c) Os clientes e os programadores (developers) trabalham em conjunto para resolver as necessidades de negócio.
d) O desenvolvimento deve ser orientado por iterações, com entrega de incrementos a funcionar.
e) A principal medida de progresso do projeto é o número de horas de contacto em pessoa (face-to-face) entre o cliente e equipa de desenvolvimento.

B6.

Na Análise, os Diagramas de Classes são adequados para:

a) mostrar as classes de objetos que pertencem ao domínio aplicacional, as suas caraterísticas e relações entre classes.
b) mostrar o modelo de dados para a construção da base de dados;
c) representar as classes que serão necessárias para uma implementação numa linguagem de programação por objetos.
d) ajudar a encontrar a arquitetura candidata da solução de software
e) os diagramas de classes não são adequados para usar na etapa de Análise.

B7.

Na Análise, os Diagramas de Atividades são adequados para:

a) Mostrar como é que vários fluxos num caso de utilização se interlaçam;
b) Mostrar quais são os papéis responsáveis por certas tarefas, recorrendo a partições;
c) Modelar o fluxo (workflow) de um processo do negócio.
d) Modelar o fluxo de dados entre tarefas num dado processo de trabalho.
e) Todas as opções anteriores estão corretas.

B8.

Qual é o principal resultado a atingir (*milestone*) no final de cada uma das 4 fases do método OpenUP?

a) Visão partilhada do âmbito e objetivos do projeto; arquitetura definida e ensaiada; versão implementada do sistema; solução em produção.
b) Requisitos detalhados da solução; narrativas dos casos de utilização; sistema implementado; solução em produção.
c) Casos de utilização; implementação exploratória; arquitetura detalhada; solução em produção.
d) Modelo do domínio; requisitos detalhados; funcionalidade totalmente implementada; solução em produção.
e) Nenhuma das opções anteriores está correta.

B9.

Qual das seguintes propriedades não é caraterística de um SDLC sequencial?

a) Os requisitos do sistema completo estão claramente definidos e documentados à cabeça.
b) Os principais requisitos têm de estar definidos; no entanto, algumas funcionalidades ou melhorias podem evoluir e ser detalhadas mais tarde.
c) O progresso é visto como um fluir direcionado, em que fase anterior desagua na seguinte.
d) O Sistema é desenvolvido primeiro em subunidades, que depois são integradas para formar o sistema, e depois são feitos os testes de validação.
e) É possível voltar atrás e retomar uma fase anterior, apesar de isso obrigar a rever os resultados dessa fase.

B10.

As abordagens iterativas e incrementais contribuem para uma redução de risco nas fases iniciais do projeto muito mais acentuada do que as abordagens tradicionais sequenciais.

a) Na metodologia sequencial o risco de estar a construir um sistema errado é maior porque o utilizador não vê resultados intermédios.
b) Na metodologia iterativa o risco de falhar os prazos do projeto é maior porque se repete os mesmos passos, a cada iteração.
c) Na metodologia iterativa a validação do sistema com os utilizadores finais é tardia, contribuindo para aumentar o risco do projeto.
d) Na metodologia iterativa o risco de estar a construir um sistema errado é maior porque não aplica esforço na especificação inicial dos requisitos.
e) Na metodologia sequencial o risco é sempre elevado, porque não é possível voltar atrás e corrigir resultados construídos em fases anteriores.

B11.

Assinale a afirmação **falsa** sobre o modelo representado no Diagrama 1:

a) Cada Equipa é coordenada por um Docente.
b) Podem existir Docentes que não coordenam nenhuma Equipa.
c) A Entrega é feita por vários Alunos.
d) Uma Entrega é avaliada por um Membro do CC.
e) Um Desafio pode ter um número alargado de Entregas.

B12.

Assinale a afirmação correta sobre o modelo representado no Diagrama 1:

a) Um Membro do CC só deve avaliar entregas resolvidas com linguagens de programação para as quais é especialista.
b) Uma Equipa poder ser composta por alunos de várias Instituições (i.e., a Equipa não é de uma Instituição).
c) As Entregas de uma Equipa relativa a um Desafio podem ser avaliadas por Docentes diferentes.
d) Todas as três afirmações anteriores são corretas.
e) Todas as três primeiras afirmações são erradas.

B13.

Considere que se pretende incluir o seguinte requisito no modelo representado no Diagrama 1. Que alterações podem ser feitas?

"Um Membro do CC só deve avaliar entregas resolvidas com linguagens de programação para a qual é especialista."

a) A classe MembroCC pode incluir um atributo multi-valor para designar as linguagens de especialização do Docente.
b) A classe Docente pode incluir um atributo multi-valor para designar as linguagens de especialização do Docente.
c) A classe Coordenador de Equipa deve incluir um atributo multi-valor para designar as linguagens de especialização do Docente.
d) A Entrega deve incluir uma associação adicional com a classe MembroCC para indicar os avaliadores candidatos/preferenciais.
e) A classe Desafio deve incluir a linguagem de programação de especialização do MembroCC que o propôs.

B14.

Considerando o Diagrama 2:

a) O diagrama descreve a interação entre objetos de software.
b) O diagrama mostra eventos, fluxo condicional e ações.
c) O diagrama está errado porque não usa partições e as tarefas ficam sem responsáveis.
d) O diagrama tem nós de reunião de fluxos em excesso.
e) O diagrama está errado porque não inclui condições de acesso (*"guard conditions"*) nos nós de reunião.

B15.

Considerando o Diagrama 2:

a) Está incompleto, porque não representa os atributos das classes.
b) É um resultado próprio da etapa de modelação do negócio.
c) É um resultado próprio do desenho do software.
d) Permite captar os requisitos não-funcionais para complementar a vista de casos de utilização.
e) Permite representar a arquitetura candidata do sistema.

B16.

Considere o processo de trabalho representado no Diagrama 2:

a) Abrir *ticket*, Atualizar *ticket*, Fechar *ticket* são exemplos de estados (de uma máquina de estados).
b) Quando não é possível Reproduzir o problema, a atividade termina.
c) Depois de Identificar a causa, o caminho a seguir é ambíguo, porque há vários fluxos possíveis.
d) Depois de Verificar a solução, o ticket é fechado.
e) O processo de trabalho é desencadeado por um evento, e não por uma ação.

B17.

Considere o Diagrama 3:

a) Representa um processo de trabalho.
b) Destaca as classes que são modificadas.
c) Mostra as ações dos atores.
d) Identifica transições de estado possíveis.
e) É o diagrama da UML mais usado pelos programadores.

B18.

A informação do modelo representado no Diagrama 3:

a) Pode ser incluída na narrativa dos casos de utilização, para explicar as interações atores/sistema.
b) Pode ser representada num diagrama de sequência, em que a linhas de vida são as entidades representadas "Em Preparação", "Aceite",…).
c) Pode ser representada num diagrama de casos de utilização, com os atores Ordenante, Comprador, Sistema.
d) Pode ser representada num diagrama de atividades, destacando as ações que levam às transições de estado.
e) A informação deste diagrama não é suscetível de ser usada para construir outra vista/tipo de diagrama.

B19.

Considerando os diagramas exemplificados (1, 2, 3);

a) São exemplos de diagramas de comportamento.
b) São exemplos de diagramas estruturais.
c) São exemplos de visualização do código (de uma linguagem de programação por objetos).
d) Todas as hipóteses anteriores estão corretas.
e) Nenhuma das três primeiras opções está correta.

B20.

Considerando a complementaridade entre os tipos de diagramas exemplificados (1, 2, 3);

a) Uma classe pode ser especificada com um diagrama de estados, para mostrar possíveis estados dessa entidade.
b) Um método de uma classe pode ser descrito com um diagrama de atividades, para clarificar um algoritmo subjacente.
c) As entidades de dados nos diagramas de atividade podem ser associadas as classes, modeladas num diagrama de classes.
d) Nenhuma das opções anteriores é correta.
e) Todas as primeiras três opções são corretas.

B21. [questão de desenvolvimento]

Os métodos de desenvolvimento ágil procuram dar resposta a certas limitações do modelo "em cascata".
Caraterize os aspetos essenciais de abordagem ágil.
Explique porque é vantajosa a sua adoção no SDLC.

B22. [questão de desenvolvimento]

Considere o trecho de código seguinte, em Java (ServicoTransferencias) com omissões (sinalizadas com **[…]**):

a) Apresente um diagrama de classes para representar a informação que se pode inferir do trecho de código anterior.
b) Apresente um diagrama que modele a interação entre objetos que ocorre quando um outro módulo invoca o método **transferir**():

```
ServicoTransferencias servico;
 […]
servico.transferir( 115, 2001, 4000.0);
```

```
public class ServicoTransferencias {
    private RepositorioContas listaClientes;
[…]
    public void transferir(int codClienteOrigem, int codClienteDestino, double quantia) {
          ContaCorrente origem = listaClientes.procurarConta(codClienteOrigem);
          ContaCorrente destino = listaClientes.procurarConta(codClienteDestino);
          origem.debitar(quantia);
          destino.creditar(quantia);
          listaClientes.atualizarConta(origem);
          listaClientes.atualizarConta(destino);
    }

    public void listarTransferenciasEnviadas( Date desde, Date ate, ContaCorrente origem) {
[…]
    }
}
```

DIAGRAMAS

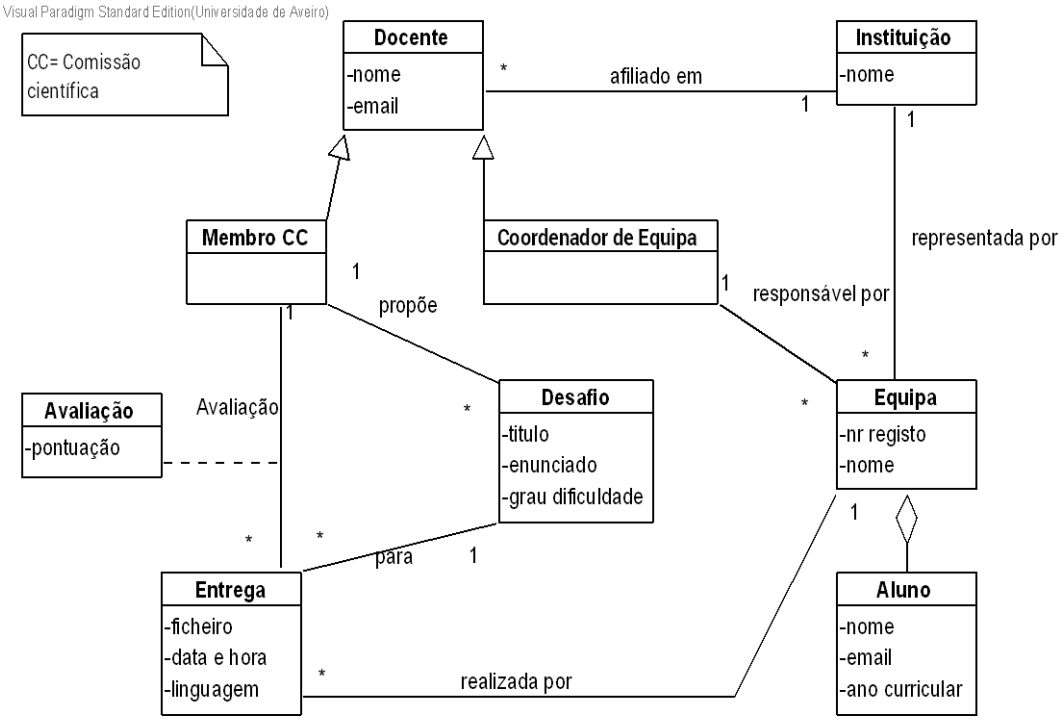

Diagrama 1- Representação parcial dos conceitos associadas à gestão de um concurso de programação, em que várias equipas de alunos resolvem desafios e entregam os programas para avaliação.

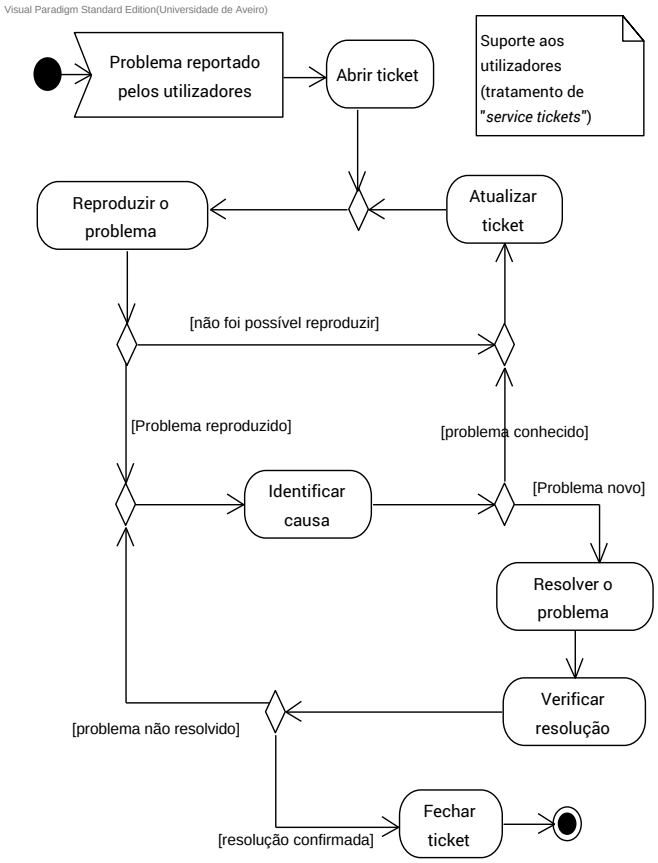

Diagrama 2- Processo de trabalho associado a um serviço de apoio ao cliente (service desk), que recebe e analisa problemas (tickets).

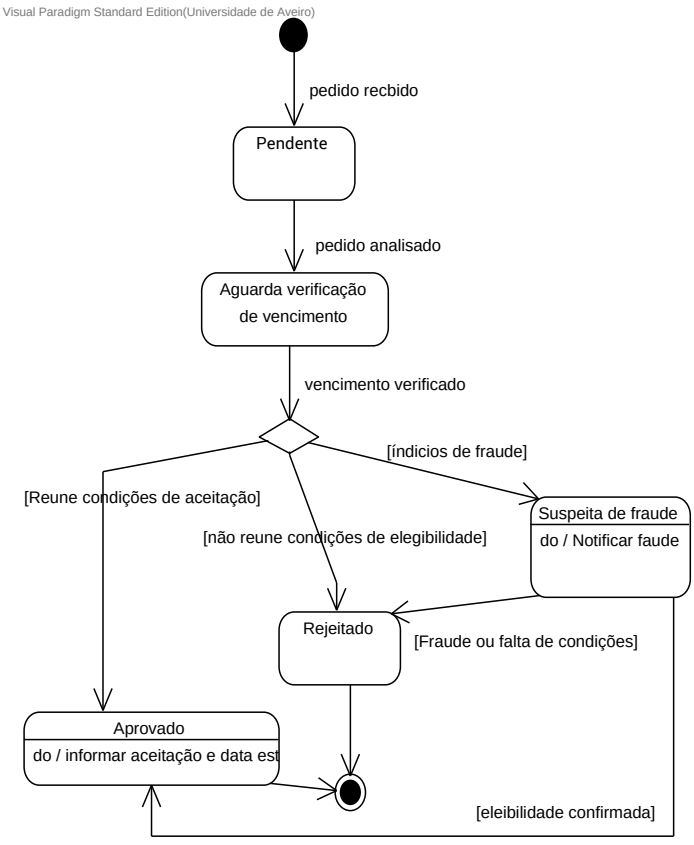

Diagrama 3 – Evoluções possíveis no tratamento de pedidos de crédito.