

## Exame de Época de Normal – 2017-06-16. Duração: 90min.

Questões de escolha múltipla: **responda na grelha**; assinale de forma inequívoca apenas uma opção (se não houver outra indicação, pretende-se a opção verdadeira e, havendo várias que possam ser consideradas verdadeiras, pretende-se a mais abrangente); as **respostas erradas descontam** ½ da cotação; as respostas assinaladas de forma ambígua não são corrigidas.

NOME:

N.MEC:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a)																				
b)																				
c)																				
d)																				
e)																				

### B1.

Os casos de utilização são:

- a) Modelos de atividades criados pelos utilizadores
- b) Descrições de como o sistema vai ser implementado, incluindo os fluxos de dados.
- c) Um guião para entrevistar os utilizadores no levantamento de requisitos
- d) Uma forma de representar as situações em que o sistema interage com os atores.
- e) Requisitos funcionais do novo sistema.

### B2.

Considere o requisito a seguir formulado relativo à operação de uma loja *online* de instrumentos musicais:

R: “O website da loja deve ser responsivo, adequado à visualização em dispositivos móveis e em *desktops*”.

- a) É um requisito funcional, mas inadequado, não se deve concretizar a tecnologia de implementação.
- b) É um exemplo de um atributo de qualidade, relacionado com a facilidade de uso do sistema (*usability*).
- c) É um exemplo de um requisito não funcional, relacionado com a fiabilidade do sistema (*reliability*).
- d) É um exemplo de um requisito não funcional, relacionado com a compatibilidade do sistema (*suportability*).
- e) Não é um bom exemplo de um requisito, porque não é verificável, dada a heterogeneidade de dispositivos e *browsers* existentes.

### B3.

Os métodos ágeis de desenvolvimento preconizam a adoção de práticas que equilibrem a disciplina e o *feedback*, tais como:

- a) Entrega frequente de valor, integração em contínuo, desenvolvimento orientado por testes.
- b) Desenvolvimento por objetos, reutilização de componentes, testes com o utilizador
- c) Iterações curtas definidas pelo utilizador, demonstração de resultados frequente, priorização dos casos de utilização.
- d) Modelação em UML de todos os aspetos do sistema, documentação clara e abrangente, reuniões de pé.
- e) Detalhar requisitos à cabeça, assinar especificações por ambas as partes, cumprir as especificações iniciais.

### B4.

A separação entre Análise e Desenho pode ser, por vezes, pouco clara. Uma razão para isso é que:

- a) Os métodos ágeis são mais confusos que os métodos baseados no modelo em cascata;
- b) Os analistas são frequentemente pressionados para aligeirar a análise, não se produzindo os resultados com o detalhe adequado
- c) É frequente a alteração do âmbito do projeto.
- d) Os resultados da análise são já um primeiro passo para o desenho do novo sistema.
- e) A natureza das disciplinas subjacentes é semelhante.

### B5.

O mesmo tipo de diagrama da UML pode ser usado para criar modelos com diferentes perspetivas de análise, em diferentes fases do SDLC, como por exemplo o \_\_\_\_, usado na fase de Análise para representar \_\_\_\_ e na fase de Construção para representar \_\_\_\_.

- a) Diagrama de atividades/ os atores/ os casos de utilização.
- b) Diagrama de classes/ os conceitos do problema/ classes de código.
- c) Diagrama de objetos/ conceitos do domínio/ métodos de cada classe.
- d) Diagramas de instalação/ a arquitetura / os processos implementados.
- e) Diagrama de pacotes/ a arquitetura / os componentes.

### B6.

No cerne do TDD está um ciclo de pequenos passos que se repete, nesta ordem:

- a) Adicionar um pequeno teste; executar os testes e verificar que o novo está a falhar; implementar as alterações suficientes para o teste passar; executar os testes e verificar que o novo passa; ajeitar o código para o tornar mais limpo.
- b) Ajeitar o código para o tornar mais limpo; adicionar um teste detalhado; implementar as alterações necessárias para o teste passar; executar os testes e verificar que o novo passa.
- c) Implementar as alterações relativas ao novo incremento; adicionar um teste; executar os testes até se verificar que o novo teste passa.
- d) Remover duplicados; melhorar a clareza do código; implementar um pequeno incremento; escrever um teste para confirmar que o incremento faz o esperado.
- e) Executar o código; identificar problemas com o incremento implementado; usar ferramenta de *debugging* para encontrar a origem; resolver os erros.

**B7.**

Podemos ter várias perspetivas de arquitetura, tais como a arquitetura lógica, de componentes e de instalação. Indique a sequência de conceitos que poderíamos encontrar em cada uma delas, respetivamente:

- a) Casos de utilização / atores / sistemas.
- b) Classes / componentes / artefactos.
- c) Pacotes / dependências / redes.
- d) Camadas / pacotes / dependências.
- e) Pacotes / componentes / nós.

**B8.**

A arquitetura trata da tomada das grandes decisões técnicas em relação ao sistema a desenvolver. Um exemplo de uma decisão de arquitetura é:

- a) A formação da equipa de desenvolvimento.
- b) O desenho de classes de código que maximizam a coesão e minimizam a interdependência.
- c) A distribuição de atributos pelas classes.
- d) A capacidade máxima do sistema em termos de sessões de utilizadores simultâneas, de forma sustentada.
- e) O método de SDLC adotado.

**B9.**

Uma classe é um “molde” que pode ser usado para criar \_\_\_\_, também designados objetos.

- a) Instâncias
- b) Componentes
- c) Métodos
- d) Atributos
- e) Pacotes

**B10.**

Nos métodos ágeis, o trabalho a fazer é captado nas entradas do *backlog*-do projeto.

- a) Os requisitos prioritários estarão no topo do *backlog*.
- b) Em cada iteração, é selecionado o subconjunto de itens que podem ser feitos, do topo do *backlog*.
- c) A ordem dos itens no *backlog* pode ser alterada, com pouca “cerimónia”.
- d) O método SCRUM utiliza a noção de *backlog*.
- e) Todas as alíneas são verdadeiras.

**B11.**

Podemos avaliar a qualidade do desenho de um software por objetos utilizando um conjunto critérios, tais como interdependência (*coupling*) e coesão (*cohesion*).

- a) Quanto maior a interdependência, maior é a probabilidade que alterações numa parte do desenho impliquem alterações em outras partes do desenho.
- b) Quando um método da classe A refere/utiliza partes internas da classe B há uma situação de interdependência.
- c) O princípio da coesão indica que uma classe deve ter um único propósito, e todos os seus métodos estão relacionados com essa finalidade.
- d) Todas as opções anteriores estão corretas.
- e) As três primeiras opções estão erradas.

**B12.**

Relativamente ao Diagrama 1, quando é que o “Coordenador Pedagógico” deve “Integrar inscrições, preferências e horários”?

- a) Logo que esteja aberto o período de inscrições no PACO.
- b) Logo que esteja encerrado o período de inscrições no PACO.
- c) Logo que terminou o período para rever preferências
- d) Quando terminarem os períodos de manifestação de preferências e de inscrições no PACO.
- e) Apenas quando os períodos de manifestação de preferências e de inscrições no PACO terminam em simultâneo.

**B13.**

Relativamente ao Diagrama 1:

- a) Tem quatro nós de decisão.
- b) Usa partições para representar subsistemas.
- c) Mostra fluxos de dados entre as ações.
- d) Usa eventos para mostrar sinais de chegada baseados no tempo.
- e) Falta uma partição para representar o próprio sistema (Portal).

**B14.**

O Diagrama 2 utiliza o conceito de estereótipo da UML (*stereotype*) na relação “extend”.

- a) É sempre possível substituir o estereótipo “extend” por “include”, alterando a ordem na apresentação do diagrama.
- b) O estereótipo “extend” tem associados pontos de extensão do caso de utilização, que podem ou não se visualizados.
- c) O estereótipo “extend” é acessório, não altera a natureza da relação de dependência.
- d) O fluxo de utilização incluído em “Criar diagnóstico” inclui sempre a sequência em “Consultar exames de imagiologia”
- e) Todas as hipóteses estão corretas.

**B15.**

Relativamente ao modelo representado no Diagrama 2

- a) Está incompleto: não inclui a fronteira do sistema.
- b) Utiliza a relação de hierarquia entre atores.
- c) O “Sistema de Imagiologia Médica” não é um ator, porque é um sistema e não um utilizador.
- d) Deveria relacionar o caso de utilização “Pesquisar utente” com “Marcar consulta”.
- e) Todas as afirmações estão corretas.

**B16.**

Nos elementos modelados no Diagrama 2 há um que está claramente desajustado. Trata-se de:

- a) O ator Supervisor: este papel já está previsto no Secretariado.
- b) O ator Secretariado: está englobado pelo papel do Supervisor.
- c) O caso de utilização “consultar exames imagiologia”: é uma parte dentro do caso de utilização “criar diagnóstico”.
- d) O caso de utilização “Dados do utente”: não tem fluxo subjacente.
- e) A associação entre os atores Supervisor e Secretariado: deveria ser uma dependência marcada com <<extend>>.

**B17.**

Considere o Diagrama 3:

- a) Um Projeto pode ou não ter um gestor.
- b) Um Projeto pode agregar sub-projetos.
- c) Uma Tarefa pode agregar Projetos.

- d) Uma Tarefa pode agregar sub-tarefas.  
e) Uma Tarefa pode ser realizada em diferentes projetos.

**B18.**

O Diagrama 3 utiliza uma classe de associação, para caraterizar a alocação a projetos.

- a) A classe de associação não deve ter atributos, porque são obtidos das classes relacionadas.  
b) A classe de associação é útil quando o relacionamento entre as classes associadas precisa de ser caracterizado quanto ao período de tempo da sua vigência.  
c) A classe de associação pode ser suprimida, desde que se mova os respetivos atributos para uma das classes associadas.  
d) A classe de associação deve ser usada sempre que há uma multiplicidade de muitos para muitos (entre as classes base associadas).  
e) Um bom modelo deve evitar a utilização de classes de associação.

**B19.**

Relativamente ao Diagrama 3, assinale a afirmação **errada**:

- a) Para cada projeto, há um Colaborador que é o respetivo gestor.  
b) Uma Tarefa herda as caraterísticas definidas para o Projeto.  
c) Uma Tarefa tem sempre um Colaborador designado como responsável.  
d) Um Colaborador pode estar alocado a diferentes tarefas, em simultâneo.

- e) Um Colaborador pode participar numa Tarefa, por um período inferior ao da duração da Tarefa.

**B20.**

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

- a) Os papéis representados nas partições de um D. de Atividades podem ser Atores no D. de Casos de Utilização.  
b) Um Actor pode dar origem a uma Classe, com atributos e responsabilidades.  
c) Um método de uma classe pode ser descrito com um diagrama de atividades, para clarificar um algoritmo subjacente.  
d) As entidades de dados nos diagramas de atividade podem ser associadas as classes, modeladas num diagrama de classes.  
e) Todas as opções anteriores são corretas.

**B21. [questão de desenvolvimento]**

Um princípio central nos métodos ágeis de desenvolvimento é o “acolhimento da mudança” (*embrace change*). Em que consiste esta ideia? Que práticas podem contribuir de forma decisiva para a sua implementação?

**B22. [questão de desenvolvimento]**

Considere o trecho de código seguinte, em Java, com omissões. Apresente um diagrama de sequência para representar a interação que ocorre quando é invocado o método DemoMain.prepararEmenta(), a partir da informação que se pode inferir do trecho de código.

```
public class DemoMain {
    private int NR_PRATOS_POR_EMENTA = 4;

    // cria ementa com quatro pratos gerados à sorte
    public Ementa prepararEmenta(String[] args) {

        Ementa ementa = new Ementa("Snack da UA");
        Prato prato;
        for (int i = 0; i < NR_PRATOS_POR_EMENTA; i++) {
            prato = this.pratoAleatorio();

            // Adicionar 2 Ingredientes à sorte ao prato
            Alimento alimento = this.alimentoAleatorio();
            prato.adicionarIngrediente(alimento);

            alimento = this.alimentoAleatorio();
            prato.adicionarIngrediente(alimento);

            ementa.adicionarPrato(prato);
        }
        return ementa;
    }
    private Prato pratoAleatorio() { ... }
    private Alimento alimentoAleatorio() { ... }
}
```

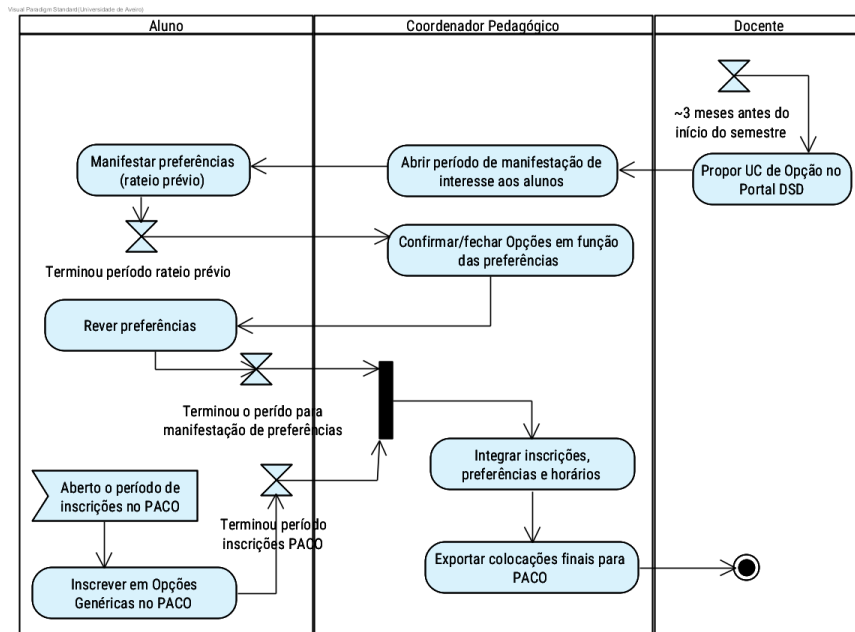
**FOLHA DE DIAGRAMAS**

Diagrama 1- Processo de escolha de Unidades Curriculares de opção ("Opções") no DETI.

(Paradigm Standard)(Universidade de Aveiro)

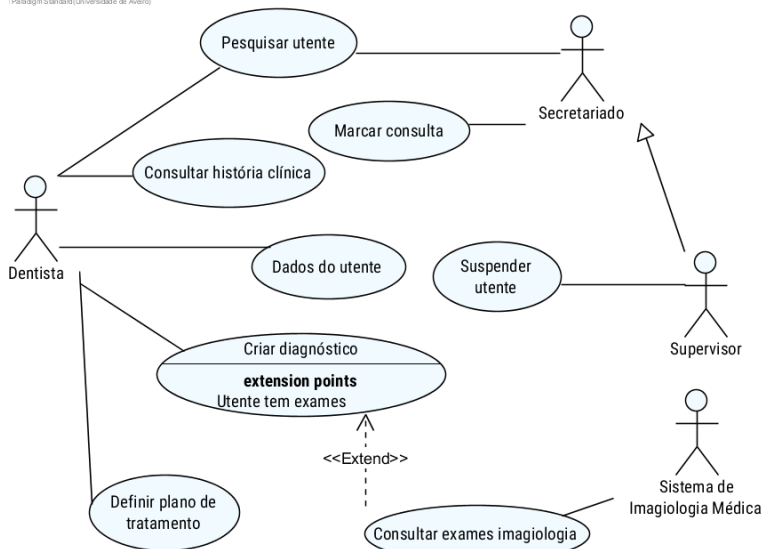


Diagrama 2- Cenários de utilização associados a uma clínica dentária.

Visual Paradigm Standard Edition(Universidade de Aveiro)

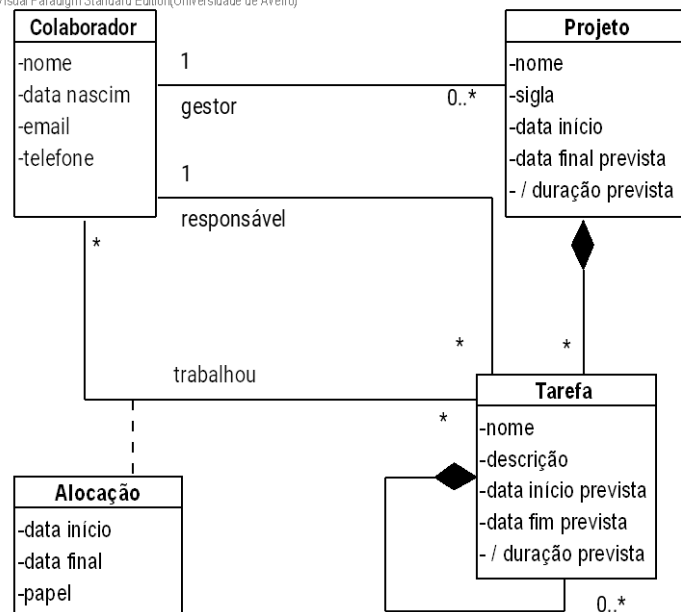


Diagrama 3 – Representação parcial dos conceitos associados à gestão de projetos.