*Exame anterior de conteúdos relacionados. Apenas para apoio ao estudo *não serve como referência*.*

*Perguntas 15 a 20 incidem sobre **diagramas da UML**.*

deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

a Normal. Duração: 90min. **A2**

NOME:

Parte I

1. A expressão "entrega contínua" (*Continuous Delivery*) é usada para caraterizar a prática em que…

A) Os incrementos são entregues com regularidade ao cliente, no fim de cada iteração.
B) O software é construído de forma a que possa ser lançado para produção a qualquer momento.
C) A qualidade do software é verificada através da execução de testes automáticos.
D) A equipa de desenvolvimento entrega diariamente incrementos para as *user stories* selecionadas para a iteração.
E) A equipa adota sistemas de integração contínua, baseados na *Cloud*.

2. Como se pode argumentar que a utilização de práticas de escrita de testes à-priori (TDD) não só não atrasa, como pode acelerar o desenvolvimento, no médio prazo?

A) A escrita dos testes e do código (de produção) são escritas por pessoas diferentes e pode-se paralelizar.
B) A escrita dos testes pode ser delegada em ferramentas (automação da escrita dos testes).
C) O código usado na escrita dos testes pode ser usado para gerar automaticamente o código de produção (*forward-engineering*).
D) O TDD revela os erros mais cedo, e é mais fácil de localizar e corrigir os erros no software, quanto mais perto do momento em que foram introduzidos.
E) Os testes são escritos com base nos critérios de aceitação das histórias (*user stories*) e, por isso, já estão definidos.

3. A metáfora da "pirâmide dos testes" transmite a ideia de que:

A) O esforço da equipa com as atividades de teste é cumulativo e aumenta de iteração para iteração.
B) O número de testes diminui com o *burndown*, i.e., à medida que menos itens de trabalho subsistem no *backlog*, há menos testes para executar.
C) Os testes das camadas superiores devem usar os testes das camadas inferiores.
D) Existem diferentes classes de teste de software, que variam em quantidade e quanto aos seus objetivos.
E) Nas metodologias ágeis (associadas à metáfora da "pirâmide") as práticas de teste são o inverso do modelo tradicional ("V-Model").

4. O V-Model descreve uma forma de integrar os procedimentos de garantia de qualidade com o processo de desenvolvimento.

A) O V-Model propõe a execução de um pacote de testes no final de cada etapa do "Waterfall".
B) O V-Model prevê a definição de diferentes tipos de testes que, concetualmente, se pode emparelhar com as etapas do "Waterfall".
C) O V-Model prevê a especificação e realização de testes numa fase avançada do projeto, depois da implementação.
D) Os tipos de testes previstos no V-Model não são praticáveis em processos iterativos.
E) Cada etapa do "Waterfall" só fica completa com a especificação e execução do pacote de testes corrrespondente do V-Model.

5. O *framework* de gestão de equipas Scrum prevê a definição de alguns papéis na equipa.

A) O Scrum Master é o responsável pela definição da arquitetura da solução.
B) O Product Owner supervisiona a realização dos testes do software.
C) O Scrum Leader é responsável por conduzir as reuniões de retrospetiva, conduzidas no final das iterações.
D) O Product Owner faz a gestão ativa do *backlog*.
E) O Product Owner representa os interesses dos *stakeholders* e tem um papel preponderante na definição de prioridades.

6. Qual a relação entre as classes usadas no modelo do domínio (na análise) e as classes usadas no código (desenho e implementação)?

A) Nenhuma; apesar de se tratar da mesma técnica diagramática, são conceitos distintos.
B) As classes do modelo do domínio não são úteis no desenho porque, tipicamente, não identificam operações (métodos), necessários para o código.
C) As classes concetuais do modelo do domínio inspiram a definição das classes do código (classes candidatas, nomes, atributos candidatos, etc.).
D) Há uma continuidade; as classes concetuais identificadas no modelo do domínio são refinadas no desenho, acrescentando-se os métodos necessários.
E) As classes identificadas na análise são as mesmas do desenho; por isso, o modelo do domínio não é importante quando se vai implementar em paradigmas que não são (orientados) por objetos.

7. O modelo comportamental de um sistema inclui as realizações dos casos de utilização (*use-case realizations*), que são:

A) A construção de protótipos de interação com o utilizador, de modo a validar as interações previstas.
B) Narrativas que descrevem o fluxo de execução do caso de utilização e alternativas a considerar;
C) Um resultado do desenho, em que se desenvolve a interação necessária entre objetos e consequente distribuição de responsabilidade, para implementar o caso de utilização;
D) Diagramas de colaboração ou sequência que mostram, para um caso de utilização, os pontos de entrada no sistema (operações de sistema).
E) Modelos e atividades que mostram os fluxos possíveis dentro de um caso de utilização;

8. Identifique, das situações listadas, aquela que **não é** uma forma/causa de *coupling* da classe A para a classe B (em Java):
A) A classe A tem um atributo do tipo B.
B) A implementação classe A invoca o método "sumAll()" definido na classe B.
C) A implementação classe A inclui um método que, na sua lista de argumentos, recebe um parâmetro do tipo B.
D) A classe A é uma sub-classe (especializa) de B.
E) A classe B é uma sub-classe (especializa) de A.

9. Os padrões de desenho fornecem soluções conhecidas para problemas de programação comuns. Alguns exemplos de padrões relacionados com a criação de objetos (*creational*) são:
A) Abstract Factory, Factory Method, Singleton.
B) Façade, Adaptor, Singleton.
C) Façade, Adaptor, Strategy.
D) Observer, Visitor, Strategy.
E) Cohesion, Single Responsability, Observer

10. As narrativas dos casos de utilização suportam diferentes atividades do processo de desenvolvimento, mas **não servem** para:
A) Mostrar como é que um ator usa o sistema para realizar os seus objetivos (fluxos) e as responsabilidades esperadas do sistema;
B) Proporcionar contexto para perceber como é que certa capacidade do sistema vai ser usada pelo ator, no ambiente de produção;
C) Captar requisitos funcionais, necessários para suportar as atividades do programador.
D) Indicar a arquitetura necessária para satisfazer as ações dos atores.
E) Suportar o trabalho de escrita de testes de aceitação, ao definir o comportamento esperado do sistema.

11. Tanto os casos de utilização como as histórias (*user stories*) fazem parte dos recursos da análise de sistemas. Qual das seguintes propriedades é caraterística dos casos de utilização (e não da história)?
A) O âmbito é adequado para ser utilizado como uma entrada do *backlog*, na gestão diária do desenvolvimento.
B) Declaração concisa das necessidades de uma *persona*.
C) Presume que há um acesso fácil aos especialistas do domínio de aplicação (para completar os detalhes da especificação, quando necessário)
D) Descreve como é que o ator se imagina a interagir com o sistema para realizar os seus objetivos.
E) Explicita as condições necessárias para a aceitação da implementação daquele incremento.

12. Wiegers suporta a ideia de que "Não é necessário desenvolver a SRS para todo o produto antes de iniciar o desenvolvimento; os requisitos para cada incremento devem ser detalhados quando se vai construí-lo." Esta ideia está presente no conceito:
A) Requisitos funcionais.
B) Requisitos não funcionais.
C) Atributos de qualidade (do software).
D) Requisitos evolutivos.
E) Requisitos ágeis.

13. O Analista documenta requisitos, mas também o ambiente do negócio. Identifique um exemplo de uma regra do negócio ("Business rule"), no contexto de um sistema para pagamento de refeições, numa cafetaria.
A) O sistema deve identificar os utilizadores com cartões "contactless", compatíveis com a norma ISO-14443.
B) Para os utilizadores que se apresentam sem cartão, a refeição tem um custo suplementar de 1EUR.
C) O sistema de POS (terminal de caixa) existente deve integrar com a identificação por cartão de funcionário.
D) A identificação do utilizador, com cartão, deve ter sucesso em menos de 3segundos, ou expira.
E) O leitor de cartões deve acender um LED verde para sinalizar o sucesso da operação.

14. Assinale o conjunto que corresponde a uma coleção de técnicas de levantamento de requisitos estudadas.
A) Entrevistas, *workshops*, questionários, análise documenta, observação no local, "focus group".
B) Casos de utilização, histórias (*user stories*), diagramas de atividades e narrativas.
C) Planeamento da iteração, Scrum diário, avaliação da iteração, retrospetiva da iteração.
D) Mapas de experiência, desenvolvimento centrado no utilizador, prototipagem de "baixa fidelidade", prototipagem de "alta fidelidade".
E) Casos de utilização, narrativas estruturadas, modelos de atividades.

15. Relativamente ao Diagrama 1:
A) Quando o dentista cria um diagnóstico, pode consultar exames de imagiologia que existam.
B) Sempre que o dentista cria um diagnóstico, deve consultar exames de imagiologia que existam.
C) Quando o Sistema de Imagiologia prepara um novo exame para o utente, o dentista deve inserir um diagnóstico.
D) A secção referente aos pontos de extensão ("*extension points"*) deve ser retirada, visto que o cenário (consulta de exames) é sempre incluído.
E) O diagrama está incompleto, porque o Dentista não está associado com o caso "Consultar exames".

16. Como é que diagramas do género do Diagrama 1 são utilizados ao longo do SDLC?
A) Podem ser usados para detalhar/suplementar os conceitos identificados no modelo do domínio.
B) Podem ser detalhados/suplementados com diagramas de sequência
C) Podem ser substituídos por diagramas de atividades, em que há partições correspondentes aos atores.
D) Os métodos ágeis de desenvolvimento privilegiam a comunicação sobre a documentação e não recomendam o uso dos casos de utilização.
E) São usados apenas na fase de análise do sistema, para explorar requisitos funcionais.

17. Em relação ao Diagrama 2:
A) O serviço "ProductSearch" é implementado pelos componentes "SearchEngine", "Inventory" e "Orders".
B) O serviço "ProductSearch" é implementado pelo subsistema "WebStore".

C) O serviço "ProductSearch" é usado pelo componente "SearchEngine".
D) O subsistema "WebStore" fornece serviços de autenticação ao subsistema "Accounting"
E) Os componentes "Orders" e "Custommers" não têm dependências funcionais entre si.

18. Em relação ao Diagrama 2:
A) É uma vista funcional de caixa-fechada, visto que não se descreve as interações entre classes.
B) É uma vista lógica da arquitetura, que identifica níveis de uma arquitetura por camadas.
C) É uma vista de arquitetura que mostra a forma como a solução está dividida em componentes, e as interfaces entre eles.
D) É uma vista de instalação que destaca os subsistemas da solução, que devem ser instalados em diferentes nós.
E) É uma vista de alto nível, que mostra a colaboração entre componentes envolvidos na execução dos pedidos dos atores na loja online.

19. Relativamente ao Diagrama 3, é possível inferir que:
A) A classe "Appointment" implementa os métodos hasPendingPayments() e addRemark().
B) O objeto "result" é sempre instanciado.
C) A classe AppointManager implementa os métodos findPendingBills()
D) A execução das invocações dentro do fragmento "alt" é condicional.
E) Todas as alíneas anteriores.

20. Como é que os diagramas do género do Diagrama 3 são utilizados num projeto?
A) O analista prepara estes diagramas para mostrar a realização dos casos de utilização.
B) Este tipo de diagramas pode ser usado para clarificar a lógica da integração com um sistema externo.
C) No desenho, estes diagramas são usados para verificar a distribuição balanceada de métodos entre classes.
D) O programador usa estes diagramas para gerar o código, poupando tempo na sua escrita.
E) Estes diagramas servem por mostrar a coesão e *coupling* entre classes.

21. [*Esta pergunta* ***é de resposta facultativa*** *e vale até 1 valor, a somar no bloco de escolha múltipla*]
Distinga as abordagens de definição de requisitos centradas no produto (*product-centric*) e centradas no utilizador/utilização (*usage-centric*), exemplificando.

Parte II

22. *[questão de desenvolvimento]*
O processo OpenUP identifica quatro etapas principais, definindo objetivos e *milestones* correspondentes. O OpenUP é um processo ágil? Fundamente a sua resposta.

23. *[questão de desenvolvimento]*
Considere a ilustração da Figura 1, na qual é possível ver um painel de um sistema de informação de voos de um aeroporto.
Apresente um modelo do domínio, i.e., o mapa de conceitos que é possível identificar a partir desta informação (e do senso comum sobre esta área de aplicação), segundo a perspetiva do Analista, num diagrama de classes.
Se necessário, anote a sua resposta com os pressupostos que considerou.

Dia	Hora	Terminal	Nº do Voo	Destino	Companhia aérea	Estado
2020-01-20	08:05		TP 1122 AD 7283 EY 1533	Alicante	TAP PORTUGAL	Cancelado
2020-01-20	09:10	T1	TP 544 BT 5353 EY 1455	Dusseldorf	TAP PORTUGAL	Partiu 10:05
2020-01-20	09:30	T2	FR 2097	Rome, Ciampino	RYANAIR	Partiu 09:56
2020-01-20	09:40	T2	DS 7685	Basel	EASYJET SWITZERLAND	Partiu 10:02

Figura 1- Informação sobre voos num painel informativo.

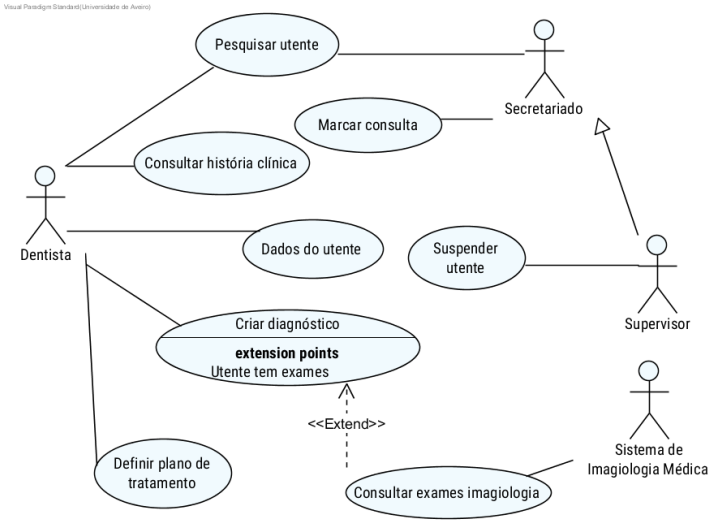

Diagrama 1- Cenários de utilização associados a um sistema de informação de uma clínica dentária.

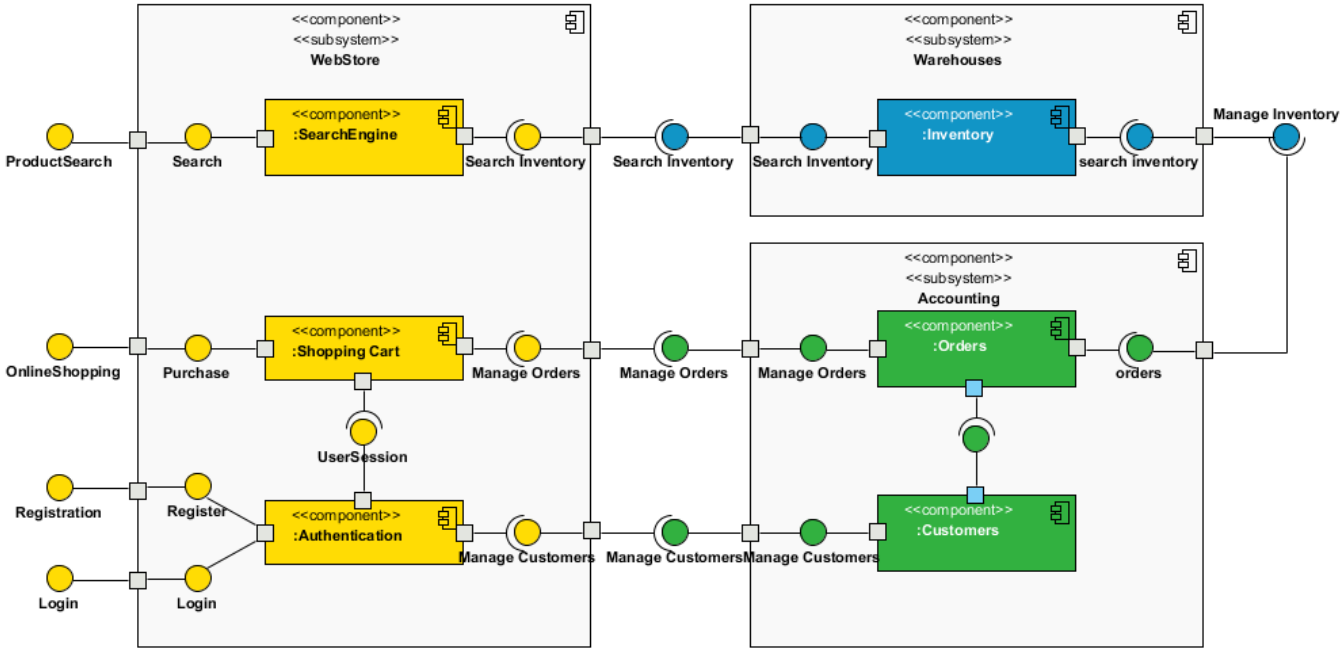

Diagrama 2- Vista da organização interna de um sistema de loja online.

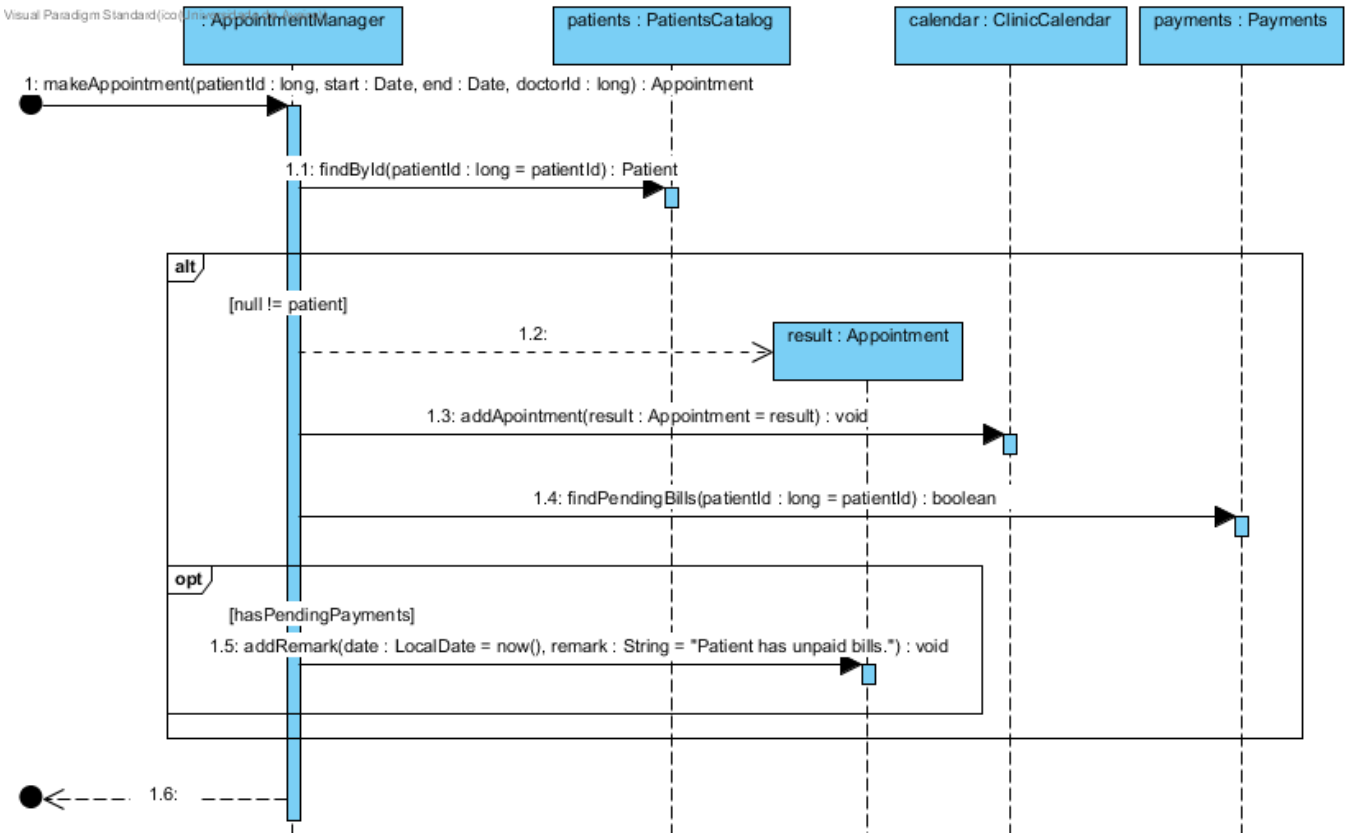

Diagrama 3- colaboração entre objetos para realizar uma nova marcação.

NOME: NR. MEC:

Questões de escolha múltipla: **responda na grelha**, assinalando a opção verdadeira/mais correta, exceto se a perguntar fornecer outra orientação. As não-respostas valem zero. **Respostas erradas descontam** ¼ da cotação; as respostas assinaladas de forma ambígua serão consideradas não-respostas. Questões de desenvolvimento: responder no espaço vazio disponível, neste enunciado.

Escolha múltipla: perguntas 1 a 20, escrever a opção de resposta (A, B, C, D, E) ou deixar vazio.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20

Questões de desenvolvimento: