deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

Exame de Análise e Modelação de Sistemas – *Recurso* | 2019-01-31 15h00. Duração: 90min.

NOME: NR. MEC:

Questões de escolha múltipla: **responda na grelha**, assinalando uma opção por pergunta (pretende-se a opção verdadeira e, havendo várias que possam ser consideradas verdadeiras, pretende-se a mais específica para o enunciado dado); as não-respostas valem zero; **respostas erradas descontam** ¼ da cotação; as respostas assinaladas de forma ambígua serão consideradas não-respostas.
Questões 22 e 23: responder no espaço vazio, no final do enunciado.

Grelha de respostas da escolha múltipla (perguntas 1 a 21):

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
a)																						**a)**
b)																						**b)**
c)																						**c)**
d)																						**d)**
e)																						**e)**

P1.

Qual o papel caraterístico do Analista numa equipa de desenvolvimento?

a) É o representante dos interesses dos *stakeholders* do projeto, na definição dos requisitos do produto.
b) É um profissional especializado no domínio do problema em que o projeto se situa e, por isso, compreende bem o funcionamento do negócio.
c) Analisa os processos da organização para identificar oportunidades de melhoria, e delineia o sistema de informação que as implementa.
d) Apresenta uma competência técnica acima da média, o que lhe permite fazer as escolhas tecnológicas da implementação.
e) Assegura a gestão diária do projeto e o diálogo com o cliente.

P2.

Wiegers carateriza a determinação de requisitos como um exigente desafio de interação humana. Porquê?

a) A determinação de requisitos é propensa a equívocos e origina resultados incompletos.
b) As conversas face-a-face são o método de determinação de requisitos por eleição.
c) Os requisitos devem principalmente ser deduzidos de cenários de utilização.
d) A descrição de situações concretas de interação pessoa/sistema (*user stories*) são decisivas para a construção de uma especificação de requisitos.
e) As tecnologias, por si só, não resolvem a necessidade de desenvolver uma cooperação eficiente com o cliente.

P3.

O método *Unified Process* prevê quatro fases principais no desenvolvimento do projeto, cada qual com objetivos e atividades próprios. Neste contexto:

a) A *Inception* inclui a análise detalhada dos casos de utilização, para mitigar os riscos de má compreensão das necessidades.
b) A *Elaboration* deve implementar as partes críticas necessárias para confirmar a viabilidade da arquitetura.
c) A *Elaboration* é facultativa na maior parte de projetos de implementação de aplicações para a Web.
d) Na *Construction*, o analista deve detalhar o modelo da base de dados (se necessário).
e) Na *Construction*, a solução deve ser instalada no ambiente de produção e aceite pelo cliente.

P4.

Os requisitos devem apresentar as caraterísticas conhecidas por S.M.A.R.T. Identifique, na lista, um requisito funcional adequadamente formulado.

a) "O sistema deve permitir a transferência de saldo entre contas do mesmo Investigador."
b) "O utilizador deve poder consultar todos os seus dados pessoais, sempre que quiser."
c) "Usar um ambiente Web, compatível com os *browsers* (produto e versão) que representam 85% do mercado."
d) "A renovação do aluguer tem de ser realizada até 24 horas antes de terminar o tempo já contratado."
e) "O Portal deve integrar com todos os sistemas de micro-pagamentos *online* existentes."

P5.

Na terminologia dos projetos de desenvolvimento ágil, o que é a velocidade da equipa (numa iteração)?

a) O número de submissões para o repositório partilhado (*commits*), por iteração.
b) O número de histórias (*user stories*) completadas e aceites por iteração.
c) O número de entregas (de incrementos à solução) feitas ao cliente, e aceites.
d) Os pontos acumulados das histórias implementadas e aceites, por iteração.
e) O número de critérios de aceitação verificados e aceites pelo cliente.

P6.

Os sistemas de software podem ser cateterizados de acordo com diferentes perspetivas de modelação. Um modelo funcional…

a) Trata o sistema sob especificação como uma "caixa opaca", especificando o comportamento observável "de fora".
b) Recorre a modelos de casos de utilização, mas sem detalhar os cenários, para evitar a especificação interna dos módulos na fase de Análise.
c) Utiliza, principalmente, diagramas de sequência para caraterizar a troca de mensagens entre objetos que cooperam.
d) Utiliza principalmente diagramas de classes, privilegiando a definição dos métodos (ou funções), em detrimento dos atributos (estado).
e) Explica como é que um caso de utilização é realizado, isto é, a interação que é necessária entre as entidades de software para implementar o cenário de utilização.

P7.

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

a) O desenho de classes de código que maximizam a coesão e minimizam a interdependência.
b) A distribuição de atributos e métodos pelas classes.
c) Os casos de utilização que são incluídos no âmbito do projeto.
d) A tipologia de plataformas de utilização que devem ser suportadas (*web*, Android, etc.).
e) A ferramenta CASE que a equipa vai adotar.

P8.

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 Desenho para representar ___.

a) Diagrama de atividades/ os atores/ partições dos dados.
b) Diagrama de objetos/ conceitos do domínio/ métodos de cada classe.
c) Diagrama de pacotes/ a arquitetura / os componentes.
d) Diagrama de classes / sistemas externos / a implementação interna.
e) Diagrama de sequência / colaboração entre o sistema sob especificação e sistemas externos / colaboração entre objetos.

P9.

É preciso documentar o protocolo de interação entre uma aplicação móvel e um dispositivo médico *Bluetooth*, de modo a clarificar as mensagens que devem ser trocadas ao longo do tempo. Que modelo UML pode ser utilizado para isso?

a) Um diagrama de instalação, clarificando os nós que intervêm.
b) Um diagrama de sequência, com ativações correspondentes aos intervenientes no protocolo.
c) Um diagrama de atividades, para caraterizar as responsabilidades do atores.
d) Um diagrama de classes, em que o dispositivo deve estender a interface Bluetooth.
e) O Diagrama de Casos de Utilização, para identificar os usos possíveis do sistema.

P10.

Durante as atividades de implementação, o programador deve ter em conta os padrões de desenho de software (*software design patterns*). O que são os padrões de desenho?

a) Soluções recomendadas para problemas recorrentes na programação por objetos.
b) Orientações para distribuir corretamente a responsabilidade de instanciar objetos.
c) Princípios para diminuir a interdependência entre objetos (*coupling*).
d) Boas práticas para organizar visualmente os modelos UML em que há a modelação de entidades de software.
e) Regras para avaliar a qualidade do desenho de uma implementação (por objetos).

P11.

Um dos princípios "SOLID" afirma que as entidades de software devem estar abertas a extensões, mas fechadas para modificações. O que é isto significa, na prática?

a) Uma classe, depois de implementada, não pode ser modificada.
b) Construir classes coesas, que apresentam uma única responsabilidade.
c) Para incrementar a funcionalidade de uma entidade, é preferível criar novo código, do que editar a implementação que já existe.
d) Manter reduzido o número de classes que implementam uma interface.
e) A criação de subclasses tem impacto na interação dos objetos já existentes e, por isso, deve ser evitada.

P12.

As histórias do utilizador (*user stories)* podem ser usadas para montar uma estratégia de garantia de qualidade do software:

a) Os cenários explorados nas histórias são usados para criar testes de aceitação sobre a web.
b) As histórias identificam *personas* que podem validar (aceitar) a implementação.
c) As histórias descrevem os objetivos que as *personas* pretendem realizar no sistema, utilizando o modelo "Sendo um…Quero [realizar]…De modo a…".
d) As histórias utilizam exemplos para descrever cenários de utilização, que constituem as condições de aceitação do incremento.
e) As histórias incluem cenários concretos, representativos das situações de sucesso.

P13.

Qual das seguintes sequências de passos deve ocorrer num processo de Integração Contínua?

a) Entrega de código (*commit*) pelo programador, testes de aceitação, testes de integração.
b) Entrega de código (*commit*), testes unitários automáticos, testes de aceitação automáticos, instalação em produção.
c) Entrega de código (*commit*), resolução de dependências e compilação no ambiente de integração, execução dos testes automáticos, visão partilhada do estado da *build*.
d) Deteção de alterações ao código na *workstation* do programador, execução de testes automáticos, instalação no ambiente de pré-produção.
e) Testes à cabeça, deteção de novo código no repositório, correção automática dos erros.

P14.

Relativamente ao Diagrama 1, o que se pode rever quanto aos atores modelados?

a) A Base de Dados é parte do sistema sob especificação, logo não é um ator.
b) A Base de Dados é um sistema, logo não é um ator.
c) O Gestor dos espaços deve especializar o Utente, pois também pode beneficiar do aluguer de espaços.
d) Faltam associações entre a Base de Dados e todos os casos de utilização (de alguma forma leem ou escrevem na BD).
e) Não há problemas a assinalar; os atores são adequados.

P15.

No Diagrama 1, as duas situações de <<include>> modeladas:

a) Refletem a dependência temporal dos casos de utilização (e.g.: Procurar não pode ser feito antes do Adicionar)
b) Estão mal aplicadas: o caso de utilização "incluído" não deve ter associação direta com atores.
c) Servem para evidenciar a necessidade da intervenção dos vários atores naqueles cenários.
d) Estão incompletas: faltam os respetivos pontos de extensão ("*extension points*").
e) Estão mal aplicadas: os cenários em causa têm objetivos diferentes e são autónomos entre si.

P16.

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.

P17.

Considere o modelo do Diagrama 2:

a) Um Equipamento encontra-se em uso em vários Complexos desportivos.
b) Só as Entidades de um Município podem operar/gerir Piscinas.
c) Um Utente pode, numa Reserva, incluir vários Equipamentos.
d) Um Complexo Desportivo destina-se à prática de uma Modalidade desportiva.
e) A Disponibilidade semanal (períodos de funcionamento) de um Equipamento é igual ao longo das várias semanas.

P18.

Segundo o Diagrama 2, o que é que um Utente reserva?

a) A utilização de um Complexo Desportivo, por um período de tempo.
b) A utilização de Uma Piscina municipal, por um período de tempo.
c) A utilização de um Equipamento desportivo, para realizar uma Modalidade bem definida.
d) A utilização de um Equipamento desportivo, por um período de tempo.
e) A utilização de um Equipamento desportivo, com um custo dependente da Modalidade desportiva.

P19.

Que alterações ao Diagrama 2 seriam necessárias para que o modelo tivesse a capacidade expressiva para representar o requisito "O custo hora de um Equipamento depende da Modalidade que o vai usar"?

a) Nenhuma (já é possível representar corretamente essa informação).
b) O atributo custo, que existe em Equipamento, deve ser movido para a classe Modalidade.
c) O atributo custo deve ser movido para uma classe de associação (entre Equipamento e Modalidade).
d) Deve ser acrescentado um atributo em Equipamento para representar a Modalidade.
e) O atributo custo em Equipamento pode ser descartado, porque é redundante com o que existe em Reserva de espaço.

P20.

O Diagrama 3 representa o fluxo de trabalho orientado por Histórias ("user stories"), em que:

a) A criação de Histórias leva à atualização do *Backlog*.
b) A aplicação dos controlos automáticos de garantia de qualidade não aborta o circuito de tratamento da História.
c) O Dono do produto pode rejeitar uma História, que retorna ao Programador.
d) O circuito de tratamento da História não é cancelado por eventos externos.
e) Todas as hipóteses anteriores estão corretas.

P21.

Em que ponto(s) da atividade modelada no Diagrama 3 seria de esperar que se utilizasse, como *input*, os resultados (entidades) do Diagrama 1?

a) Para escrever a "user story".
b) Para entregar a implementação.
c) Para aceitar a "user story".
d) Todas as alíneas anteriores.
e) Na verdade, a atividade modelada no Diagrama 3 não recorre às entidades modeladas no Diagrama 1.

P22. **[questão de desenvolvimento]**

"As empresas agora operam num ambiente que está a mudar de forma incrivelmente rápida. Novos produtos aparecem e desaparecem, as leis e regulamentos mudam, as empresas fundem-se e reestruturam-se. Os novos pacotes de software têm de ser rapidamente concebidos, implementados e entregues. Não há tempo para processos de engenharia de requisitos prolongados. O desenvolvimento começa logo que uma visão para o software está disponível; os requisitos emergem e são clarificados durante o processo de desenvolvimento." In: Sommerville, I. (2005). Integrated requirements engineering: A tutorial. *IEEE software*, *22*(1), 16-23.

A citação de alguma forma passa a ideia que o desenvolvimento de software precisa de se adaptar ao novo ritmo (acelerado) com que acontecem os negócios.

O que pode ser feito para adequar as metodologias da engenharia de software ao "novo" contexto?

P23. **[questão de desenvolvimento]**

Considere o trecho de código seguinte, em Java, com omissões.

a) Apresente um diagrama de classes para visualizar a informação estrutural que se pode depreender deste código.
b) Apresente um diagrama de sequência para representar a interação entre objetos que ocorre quando é invocado o método WeatherRepository#**refeshForecastIfNeeded**.

```java
/**
 * Manages the requests to get the weather forecast. If the local data
 * is still recent, no remote requests are made. Otherwise, the IPMA's
 * API is invoked.
 */
public class WeatherRepository {
    private static final String BASE_URL = "https://api.ipma.pt/open-data/";
    private static final int REFRESH_PERIOD = 300;

    private static RemoteWeatherAPI apiService = new IpmaApiClient( BASE_URL);
    private MyDatabase localDb;

    public void refreshForecastIfNeeded( int placeId,  Date day) {
        boolean goodLocalData = localDb.hastUpdatedForecast(placeId, day, REFRESH_PERIOD);
        if (! goodLocalData) {
            WeatherForecast forecast = null;
            try {
                forecast = apiService.getForecastForPlace( placeId);
                if (forecast != null) {
                    localDb.save(forecast);
                }
            } catch (IOException e) { e.printStackTrace(); }
        }
    }
}
```

Folha de Diagramas

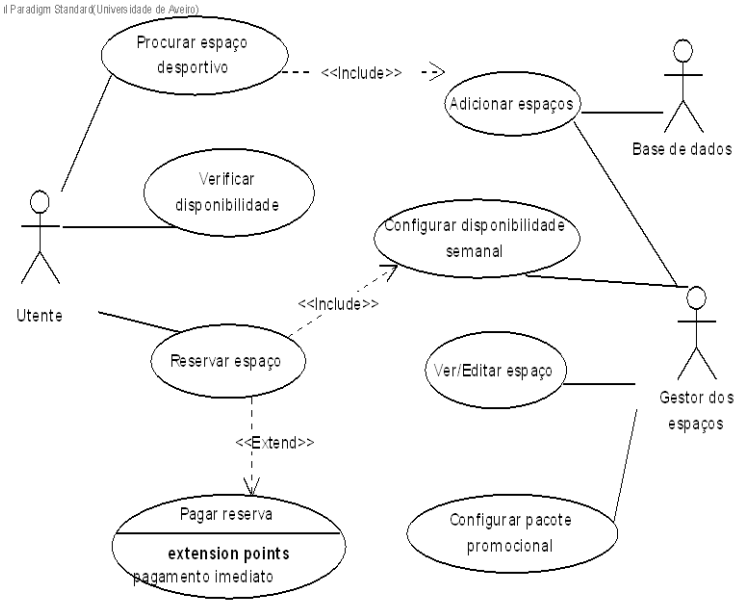

Diagrama 1- Cenários de utilização associados a um sistema de reserva de espaços desportivos.

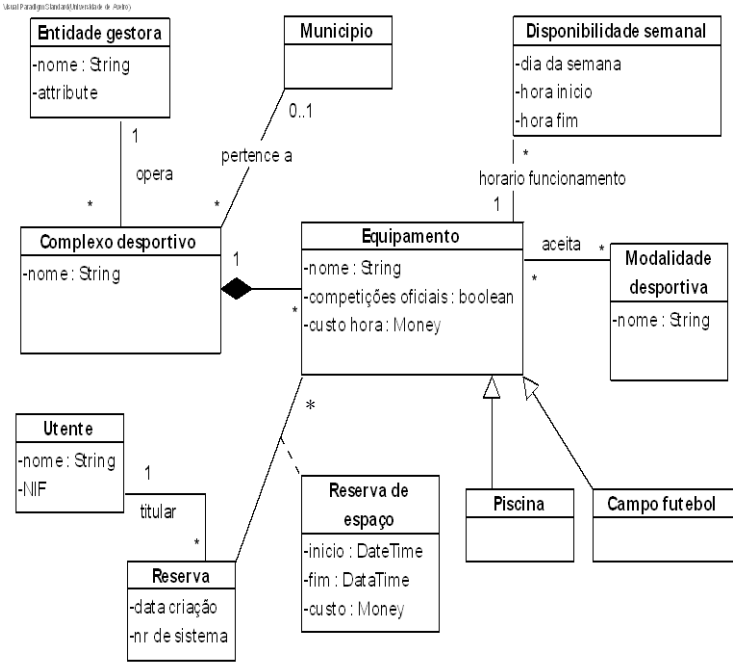

Diagrama 2 – Representação parcial dos conceitos associadas à gestão de alugueres de espaços desportivos.

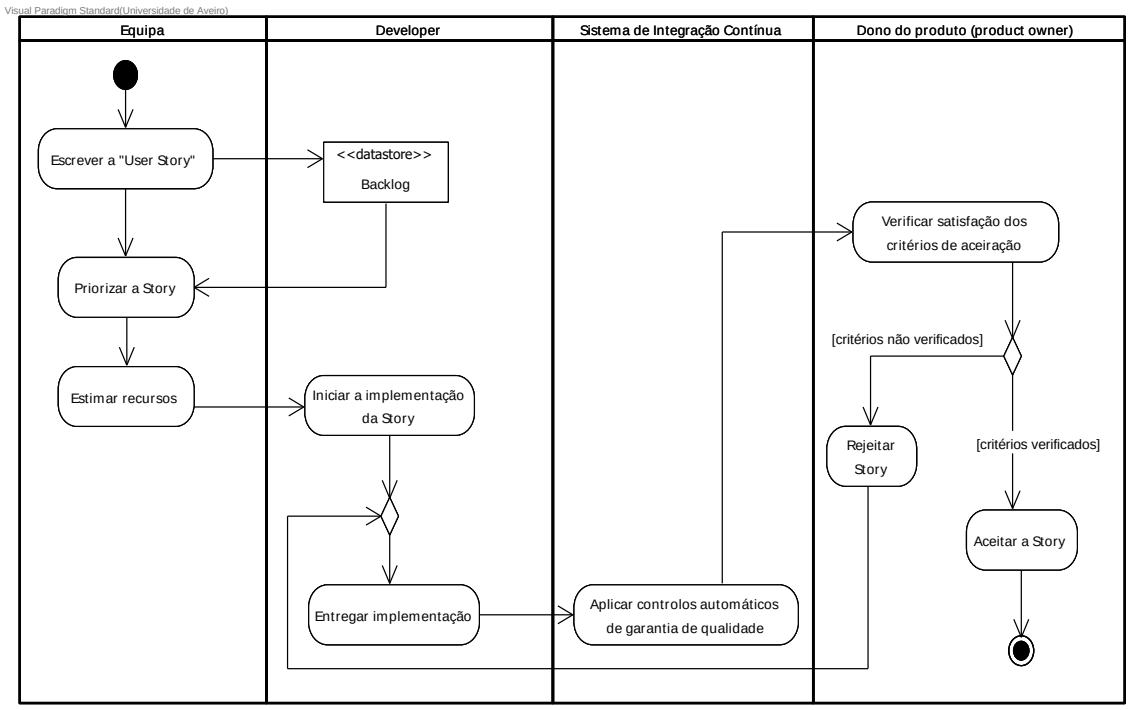

*Diagrama 3- Processo de trabalho numa equipa de desenvolvimento, recorrendo a "*user stories*".*

Questões de desenvolvimento

NOME: NR. MEC: