

Exame de Época Normal – 2019-06-14 10h00. Duração: 90min.

B-2

NOME: _____

Nr.MEC: _____

Questões de escolha múltipla (1 a 20): **responda na grelha**, assinalando uma opção por pergunta (pretende-se a opção (mais) verdadeira, exceto se a pergunta der outra orientação); as não-respostas valem zero; **respostas erradas descontam** $\frac{1}{4}$ da cotação; as respostas assinaladas de forma ambígua serão consideradas não-respostas. Questões 21 e 22: responder em folha separada, identificada.

Grelha de respostas (perguntas 1 a 20):

	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)																				

P1.

O ciclo de desenvolvimento de sistemas (SDLC) engloba quatro etapas genéricas. Qual o trabalho típico da fase de Análise?

- a) Analisar sistemas existentes, recolher requisitos para o novo sistema em articulação com os promotores, desenvolver o conceito/proposta para a solução.
- b) Analisar os novos processos de trabalho, definir a arquitetura do novo sistema, selecionar tecnologias para a implementação da solução.
- c) Analisar o valor que o sistema pode trazer à organização, determinar a viabilidade do projeto, estabelecer um plano de projeto inicial.
- d) Analisar os casos de utilização, extrair histórias de utilização, criar modelos UML para as classes principais do sistema.
- e) Identificar tendências emergentes na evolução da tecnologia, propor oportunidades de desmaterialização de processos, caracterizar os casos de utilização alvo.

P2.

O método *Unified Process* (UP) prevê quatro fases principais no desenvolvimento do projeto, divididas em iterações. Qual das seguintes afirmações NÃO caracteriza o método UP?

- a) O método UP preconiza o desenvolvimento iterativo e incremental e adota, por isso, uma das ideias principais dos métodos ágeis.
- b) As fases do UP são de tamanho variável entre si, mas as iterações são regulares.
- c) É normal que haja lugar ao desenvolvimento de código em iterações anteriores à fase de Construção.
- d) A verificação dos incrementos, com recurso a testes, pode ser feita em iterações iniciais da Construção.
- e) Cada iteração foca uma disciplina técnica específica (e.g.: planeamento, levantamento de requisitos, definição da arquitetura, etc.)

P3.

O método Unified Process é orientado por casos de utilização (CaU), porque neste método:

- a) Os CaU estabelecem uma divisão funcional do sistema que é usada ao longo do desenvolvimento do projeto, na análise, no desenho e nos testes.
- b) A primeira tarefa do SDLC é o levantamento e especificação de CaU, incluindo cenários típicos e alternativos.
- c) A UML e os Diagramas de Casos de Utilização são usados para descrever o projeto.
- d) É recomendada a identificação de personas e a exploração de cenários de uso através de histórias ("user stories").
- e) Os CaU são usados para delinear os planos de teste.

P4.

A utilização de modelos em engenharia de software pode contribuir positivamente para o desenvolvimento dos produtos. Identifique um benefício que NÃO decorre do uso de modelos.

- a) Promover uma comunicação mais clara e sucinta dentro da equipa (técnica), recorrendo a uma linguagem visual partilhada;
- b) Manter o desenho (planeamento da solução) e a implementação (construção) mais coerentes; o modelo orienta a construção.
- c) Antecipar problemas de integração entre módulos, e dúvidas quanto aos requisitos dos produtos de software, mesmo antes de iniciar as atividades de implementação;
- d) Construir uma visão partilhada do progresso do projeto e prioridades de cada iteração/Sprint.
- e) Facilitar a geração automática de código (gerar a solução a partir do modelo).

P5.

Num projeto de modelação (e.g.: do VisualParadigm), foi criada a entidade A (por hipótese, um Ator). Como é que uma dada entidade de modelação se relaciona com os vários diagramas?

- Um elemento do modelo pertence a um diagrama (1:1); quando se apaga o diagrama do projeto, as entidades nele visualizadas são apagadas do projeto.
- Um elemento do modelo pode ser visualizado em vários diagramas (1:N), desde que sejam do mesmo tipo (e.g.: o Actor só pode ser representado no diagramas de casos de utilização).
- Um elemento do modelo pode ser visualizado em vários diagramas (1:N), que podem ser de tipos distintos.
- A UML prevê que um elemento de modelação possa ter diferentes representações visuais (*stereotypes*), mas isso é distrativo e não deve ser usado.
- A representação de um elemento do modelo em diagramas distintos, mesmo que permitida pela ferramenta, é desaconselhada na UML.

P6.

O conceito de “velocidade” da equipa num projeto, tal como é usado nos métodos ágeis/SCRUM, é um importante instrumento de planeamento.

- A determinação da velocidade requer que todas as *user stories* (que aguardam implementação) tenham uma pontuação definida, pelo *project owner*.
- A velocidade é o número de *user stories* implementadas por iteração.
- A velocidade é a média dos pontos realizados por iteração, respeitantes às *user stories* implementadas.
- A velocidade é essencial para ajustar a duração de cada iteração para o realizar o número de pontos pretendido.
- A uma equipa maior corresponde uma velocidade maior.

P7.

Uma das principais razões para se utilizar métodos ágeis em engenharia de software, em detrimento dos métodos sequenciais, é a diminuição do risco do projeto. Que prática é decisiva para a mitigação do risco:

- O projeto está dividido em iterações, o que assegura uma integração frequente dos incrementos parciais, e menor probabilidade de desvio face aos requisitos.
- A gestão do risco é baseada em *feedback* frequente: a comunicação entre equipa de desenvolvimento e os *stakeholders* (Cliente e/ou *product owner*) é requerida em vários eventos das iterações.
- Os projetos são mais pequenos e não se gasta tanto tempo em tarefas de coordenação e documentação.
- A ordem dos itens na pilha do *backlog* é imutável, tornando o projeto mais previsível.
- O teste do software é sobretudo feito pelo Cliente, melhorando as condições de aceitação do produto.

P8.

No desenho de código por objeto, devemos observar princípios que levam a soluções mais fáceis de expandir e manter.

- Uma classe VideoClip contém métodos para editar o trecho de vídeo que representa. Desde que os seus métodos se foquem apenas em atributos do vídeo, a coesão é mantida.
- A elevada coesão de um módulo significa que não usa dados de outros objetos para realizar as suas operações.
- As classes de um sistema devem ter um conhecimento abrangente (do desenho) das outras classes; quando um método enviar mensagens apenas para objetos próximos/conhecidos, favorecemos a coesão interna.
- A colaboração entre objetos, desde que baseada na invocação de métodos públicos, não influencia o acoplamento.
- A busca do baixo acoplamento e elevada coesão no desenho por objetos é impossível de atingir, porque são ideias contraditórias.

P9.

A “pirâmide dos testes” evidencia que existem diferentes tipos de teste, com granularidades e objetivos diferentes. Qual das seguintes afirmações NÃO é correta:

- Os testes unitários verificam a implementação de pequenas unidades de código (e.g. classe), de forma isolada.
- Os testes unitários são preferencialmente escritos por *testers*, e não pelo programador que implementou a unidade sob teste.
- Os testes de integração verificam o comportamento de um módulo cujo funcionalidade requer a colaboração com um grupo pequeno de outros componentes.
- Nos testes realizados sobre a camada de UI, procura-se automatizar os cenários descritos nas *user stories*.
- Os testes de aceitação verificam se os requisitos estão satisfeitos, na perspetiva do utilizador final.

P10.

O sistema VitalsRecorder, para aquisição de sinais vitais de um grupo de participantes numa experiência, define o seguinte requisito: “Rj: As recolhas de dados em curso nos dispositivos móveis devem continuar, mesmo quando ocorra a desconexão pontual entre algum dispositivo e o *tablet* de controlo”.

- É um requisito de desempenho, relacionado com o tempo de resposta das unidades móveis.
- É um requisito de usabilidade, crítico para estabelecer a facilidade de utilização do sistema.
- É um requisito de fiabilidade, que foca a integridade e a recuperação das operações, face a falhas da camada física.
- É um requisito relacionado com a manutenção do sistema (*maintainability*), visto que se pretende assegurar a continuidade das operações.
- É um requisito funcional (adequação à função) e não está relacionado com a lista de atributos do sistema.

P11.

Que relação pode ser estabelecida entre a arquitetura de um sistema de software e a UML?

- A UML recomenda a utilização de arquiteturas baseadas em componentes.
- A UML permite descrever a vista física de instalação de um sistema, mas não tem diagramas para ajudar a caracterizar a organização lógica (blocos de alto nível e associações).
- Podemos mostrar as interações entre objetos recorrendo a diagramas de sequência da UML para clarificar as colaborações entre componentes.
- A arquitetura pode ser comunicada recorrendo a diagramas de UML que suportam as noções de “módulos” e “dependências”. Dessa forma, podemos mostrar as partes constituintes de um sistema e as relações estruturais.
- Não há uma relação entre os dois conceitos; a UML não prevê diagramas de arquitetura.

P12.

Considerando a capacidade expressiva do Diagrama 1:

- Uma Empresa tem mais de um Departamento.
- Um Escritório pode ser partilhado por várias Empresas.
- Uma Empresa ou tem Escritórios ou tem Departamentos.
- O Escritório tem associado um Empregado, que é o seu responsável.
- Um Departamento tem um conjunto de Empregados distintos para cada Projeto.

P13.

Que alterações seria necessário fazer ao Diagrama 1 para captar o requisito: “Um Empregado participou num projeto por um período de tempo designado.”

- A classe Projeto deve ter dois atributos adicionais, para indicar o início e o fim do período.
- A classe Empregado deve ter dois atributos adicionais, para indicar o início e o fim do período.
- As classes Projeto e Empregado devem especializar uma classe abstrata com a data de início e fim do período.
- A associação de participação, entre Projeto e Empregado, deve ser caracterizada com uma classe de associação.
- Essa informação não é suscetível de ser captada num modelo UML.

P14.

Relativamente ao Diagrama 1:

- Não é possível realizar projetos, porque a classe Projeto não pode ser instanciada.
- Um Departamento não pode ser responsável por projetos internos e por projetos externos.
- Os projetos externos têm um gestor designado.
- Os Empregados têm Escritórios dos quais são responsáveis.
- Cada projeto interno tem um Departamento responsável.

P15.

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

- É sempre possível substituir o estereótipo “extend” por “include”, alterando a ordem na apresentação do diagrama.
- O estereótipo “extend” tem associados pontos de extensão que definem as condições em que o cenário adicional é inserido no cenário base.
- O estereótipo “extend” é acessório, não altera a natureza da relação de dependência.
- O fluxo de utilização de “Editar projeto” inclui sempre a sequência em “Configurar equipa”
- O fluxo de utilização de “Editar projeto” inclui por vezes o “Configurar equipa”, seguido de “Configurar o módulo de gestão de tarefas”.

P16.

Relativamente ao modelo representado no Diagrama 2:

- Está incompleto: não inclui atores que são sistemas, e não utilizadores humanos.
- Está errado: devia utilizar a relação de hierarquia entre atores.
- Deveria relacionar os casos de utilização “Criar Projeto” e “Editar Projeto”.
- Não deve misturar a utilização de «Include» e «Extend» no mesmo diagrama, são relações incompatíveis.
- O caso de utilização “Projeto” carece de revisão, uma vez que o nome não identifica uma motivação do ator.

P17.

Considerando os elementos modelados no Diagrama 2:

- O ator “Membro da equipa” pode ser representado num diagrama de classes e associado com outros conceitos.
- A especificação do ator “Responsável” deve ser suplementada com diagramas adicionais, para explicar os estados possíveis das entidades que gere (e.g.: estados de uma tarefa)
- O caso de utilização “Pesquisar tarefas e estados” é uma parte dentro do caso de utilização “Atualizar tarefas” e não deve estar evidenciado.
- Deveria ser representada a associação entre os atores Responsável e Membro da equipa, através de uma dependência marcada com <<extend>>.
- Os atores são uma parte indispensável; deveriam estar dentro da fronteira do sistema.

P18.

Relativamente ao processo descrito no Diagrama 3:

- Um trabalho é avaliado logo que é entregue.
- Um trabalho não pode ser entregue depois do fim do prazo.
- Os alunos que entregaram e os que desistiram recebem feedback do docente.
- O trabalho tem diferentes estados, ao longo da atividade.
- O trabalho tem diferentes estados, mas estão mal atribuídos às partições.

P19.

Em que fase do SDLC é mais natural que se construa um resultado como o Diagrama 3:

- Na Análise, para caracterizar processos de trabalho existentes ou os novos processos pretendidos.
- Na Análise, para fazer o levantamento dos atores.
- Na Análise, para fazer o levantamento dos casos de utilização e cenários subjacentes.
- Na Implementação, para clarificar os algoritmos que os métodos das classes devem codificar.
- Na Implementação, para mapear os objetos de informação trocados entre classes.

P20.

Relativamente ao tipo de diagrama ilustrado no Diagrama 3 e à sua relação com outros diagramas da UML:

- Os estados anotados nos objetos podem ser visualizados num diagrama de estados.
- As ações modeladas são também casos de utilização (Lançar trabalho, Realizar Trabalho,...).

- As partições de um Diagrama de Atividades são os atores do sistema.
- Os eventos temporais correspondem à mensagem inicial num diagrama de sequência.
- Todas as hipóteses anteriores são corretas.

P21. [questão de desenvolvimento]

“A eficácia e valor, tanto dos casos de utilização como das *user stories*, decorre da perspetiva centrada no utilizador e centrada na utilização. Os utilizadores formarão expectativas mais claras sobre o que o novo sistema lhes permitirá fazer do quando se utiliza uma abordagem centrada nas funcionalidades de um produto.” (K.Wiegers.)

- Explique, por palavras suas, os benefícios de usar técnicas de análise de requisitos centradas na utilização.
- Explique como é que os conceitos de caso de utilização e *user story* se relacionam, e como são usados na rotina diária da equipa de engenharia de software, numa abordagem ágil.

P22. [questão de desenvolvimento]

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

- Apresente um diagrama de classes para visualizar, de forma detalhada, a informação estrutural da classe `ConversorMoedas` e das suas dependências diretas.
- Apresente um diagrama de sequência para representar a interação entre objetos que ocorre quando é invocado o método `ConversorMoedas#converter()`.

```

3 public class ConversorMoedas { /// algum código omitido ///
4
5     private TabelaDeCambio tabelaCambio;
6
7     /**
8      * converte uma quantia monetária para o correspondente noutra moeda
9      * @param origem valor base a converter
10     * @param codigoMoedaDest código internacional da moeda pretendida (EUR, USD, GBP..)
11     * @return o valor convertido para a nova moeda
12     */
13     public Quantia converter(Quantia origem, String codigoMoedaDest) {
14         Quantia convertido;
15
16         if (tabelaCambio.existe(origem.getCodigoInternacional()) &&
17             tabelaCambio.existe(codigoMoedaDest)) {
18             String codigoMoedaOrigem = origem.getCodigoInternacional();
19             double cambio = tabelaCambio.procurarTaxa(codigoMoedaOrigem, codigoMoedaDest);
20             double valorInicial = origem.getValor();
21             convertido = new Quantia(codigoMoedaDest, valorInicial * cambio);
22         } else {
23             throw new IllegalArgumentException( "Códigos de moeda não reconhecidos.");
24         }
25         return convertido;
26     }
27 }

```

FOLHA DE DIAGRAMAS

Visual Paradigm Standard(ico(Universidade de Aveiro))

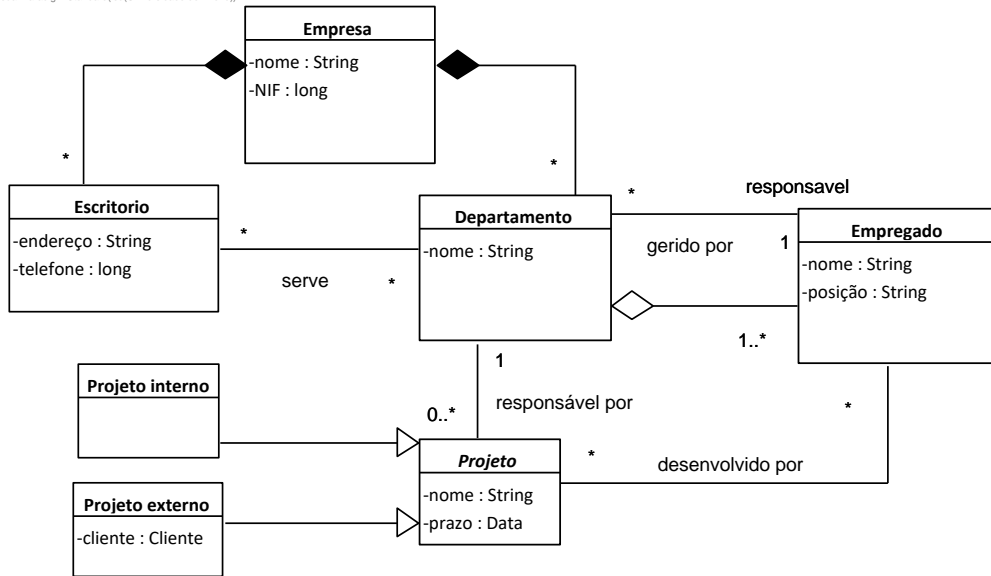
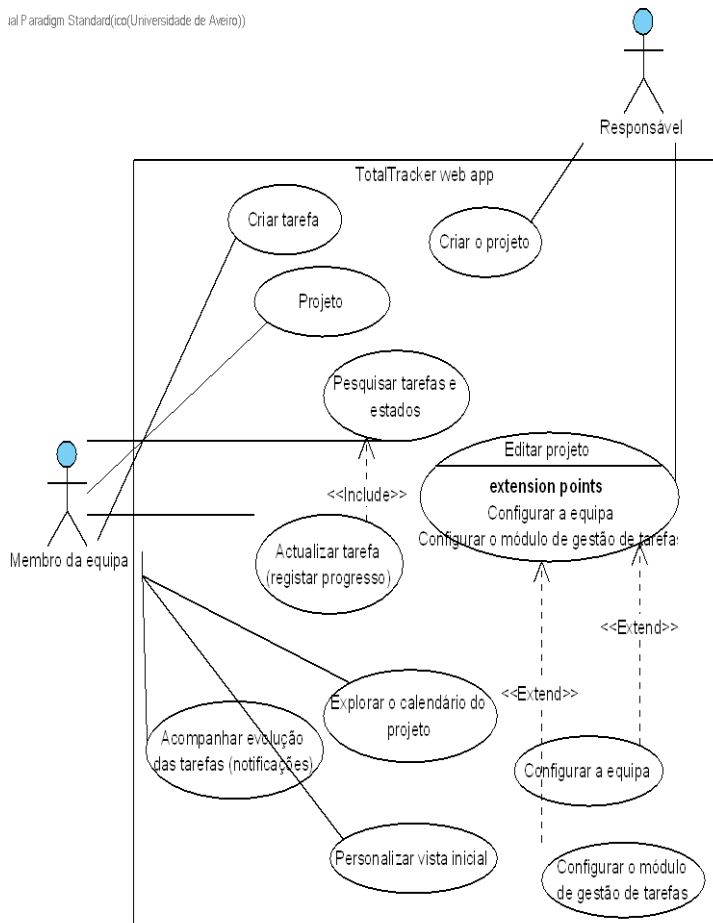


Diagrama 1- Conceitos associados à organização interna de uma empresa.

Diagrama 2- Cenários de utilização associados a um sistema de gestão de projetos.

Visual Paradigm Standard(ico(Universidade de Aveiro))



Visual Paradigm Standard(ico(Universidade de Aveiro))

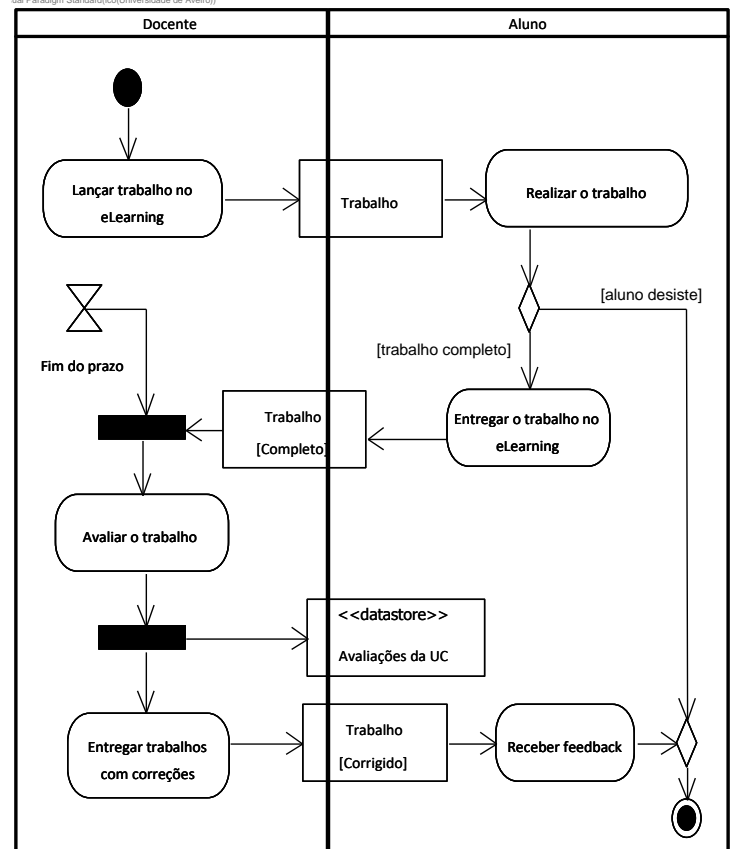


Diagrama 3: Atividades associadas à realização de trabalhos académicos.

NOME:

Nr.MEC:

