

Teste Modelo

1. Os métodos ágeis de desenvolvimento de software preconizam a manutenção de uma relação aberta, transparente e cooperante com o cliente através de práticas tais como?

“Participação do cliente na priorização dos casos de utilização.”

Option: b)

2. O método Unified Process prevê quatro fases principais no desenvolvimento do projeto. Cada qual tem um grande objetivo a atingir (milestone) para se poder avançar, por esta ordem?

“1/ Decisão de avançar ou parar o projeto; 2/ Arquitetura técnica definida e validada; 3/ Funcionalidades da primeira versão do produto implementadas; 4/ Solução instalada e aceite pelo cliente.”

Option: d)

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

- During the Elaboration phase, the project team is expected to capture a healthy majority of the system requirements.
- Address known risk factors and to establish and validate the system architecture.
- Common processes undertaken in this phase include the creation of use case diagrams, conceptual diagrams (class diagrams with only basic notation) and package diagrams (architectural diagrams).



“Os CaU estabelecem uma divisão funcional do sistema que é usada ao longo do desenvolvimento do projeto, na análise, no desenho e nos testes.”

Option: b)

4. Considere o requisito a seguir formulado, relativo à operação de uma loja online, no contexto da análise de requisitos de um sistema de software:

Ri: “Os utilizadores devem autenticar-se na Loja (login) para realizar todas as operações, exceto para a consulta dos artigos em promoção.”

Types of requirements:

- Functional - relates to a process or data
- Non-functional - relates to a system quality

“É um exemplo de um requisito não funcional, relacionado com a segurança do sistema.”

Option: b)

5. Os conceitos de Ator e Persona são usados na análise de sistemas para descrever cenários de uso de um sistema.

“A Persona é uma instância/concretização de um Ator.”

Option: b)

6. Na SCRUM, há lugar à gestão ativa da pilha de trabalho (backlog). Neste contexto, que propriedades são próprias do backlog?

- The Product Backlog indeed must be ordered: its ordering determines PBIs' order of delivery.

“A pilha está ordenada, das funcionalidades com maior pontuação (no topo) para as que têm menos pontuação.”

Option: d)

7. A arquitetura trata da tomada das grandes decisões técnicas em relação ao sistema a desenvolver, tendo em conta os atributos de qualidade pretendidos. Um exemplo de um assunto/decisão de arquitetura é:

- Especificar os cenários de interoperação com sistemas externos e as tecnologias selecionadas para os implementar.
- Definir estratégias de distribuição de carga para garantir a disponibilidade do sistema em utilização contínua, com 1000 sessões simultâneas.
- Definir os mecanismos técnicos para adaptar um sistema existente aos novos requisitos previstos no RGPD.

“Todas as opções anteriores são corretas.”

Option: d)

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

“A entrega frequente de valor diminui os problemas decorrentes de eventuais divergências na percepção dos requisitos.”

[Option: d\)](#)

9. O modelo do domínio é construído na Análise, para mapear os conceitos do universo de discurso.

“Os conceitos identificados no modelo do domínio serão candidatos naturais a constituir classes, na fase de desenho.”

[Option: c\)](#)

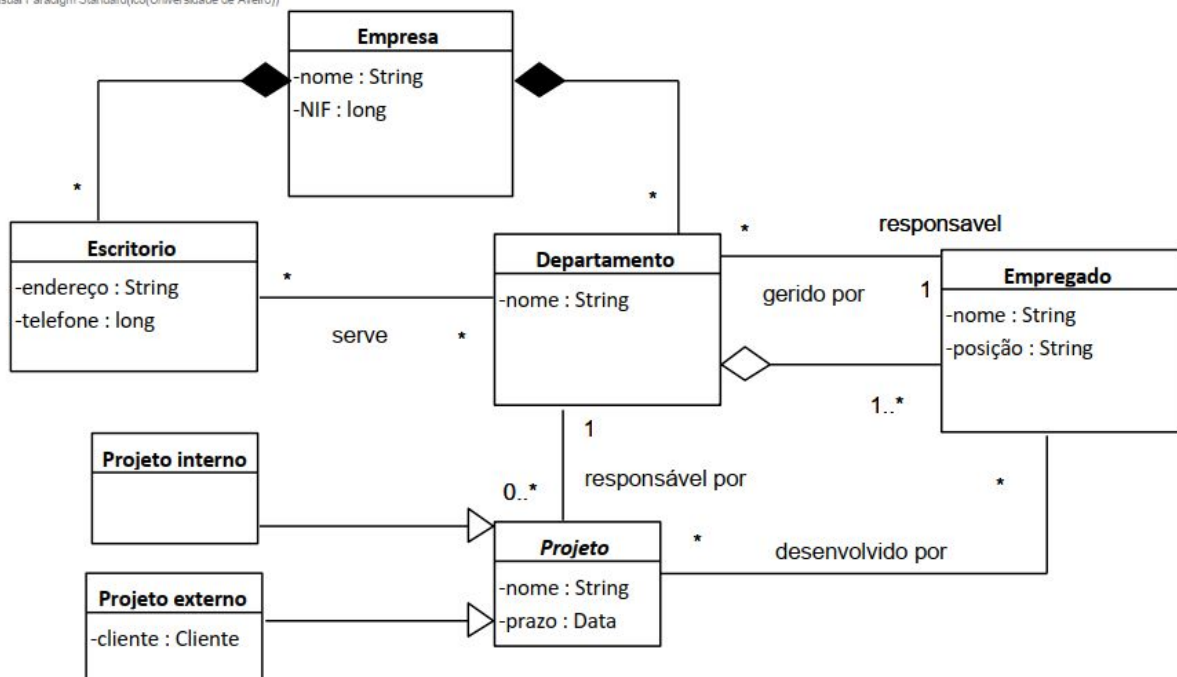
10. É possível avaliar a coerência entre diferentes diagramas de um modelo. Considerando a transposição do diagrama de objetos para o diagrama de classes, devemos garantir que:

“Todos os atributos de um Objeto são representados como atributos de uma Classe.”

[Option: a\)](#)

Diagrama 1

Visual Paradigm Standard(jco(Universidade de Aveiro))



11. Considerando a capacidade expressiva do Diagrama 1:

“Um Departamento tem sempre Empregados.”

[Option: e\)](#)

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

“A associação de participação, entre Projeto e Empregado, deve ser caracterizada com uma classe de associação.”

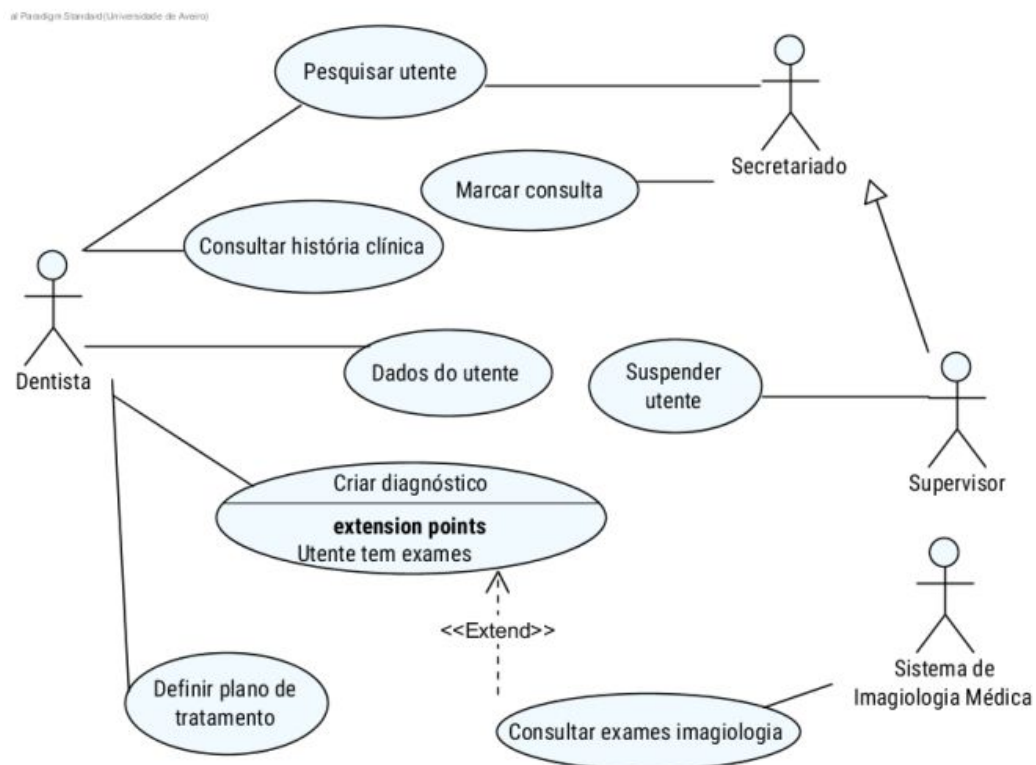
Option: d)

13. Relativamente ao Diagrama 1:

“Cada projeto interno tem um Departamento responsável.”

Option: e)

Diagrama 2



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

Stereotype is a profile class which defines how an existing metaclass may be extended as part of a profile. It enables the use of a platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass.

“O estereótipo “extend” tem associados pontos de extensão do caso de utilização, que podem ou não se visualizados.”

Option: b)

15. Relativamente ao modelo representado no Diagrama 2:

“Utiliza a relação de hierarquia entre atores.”

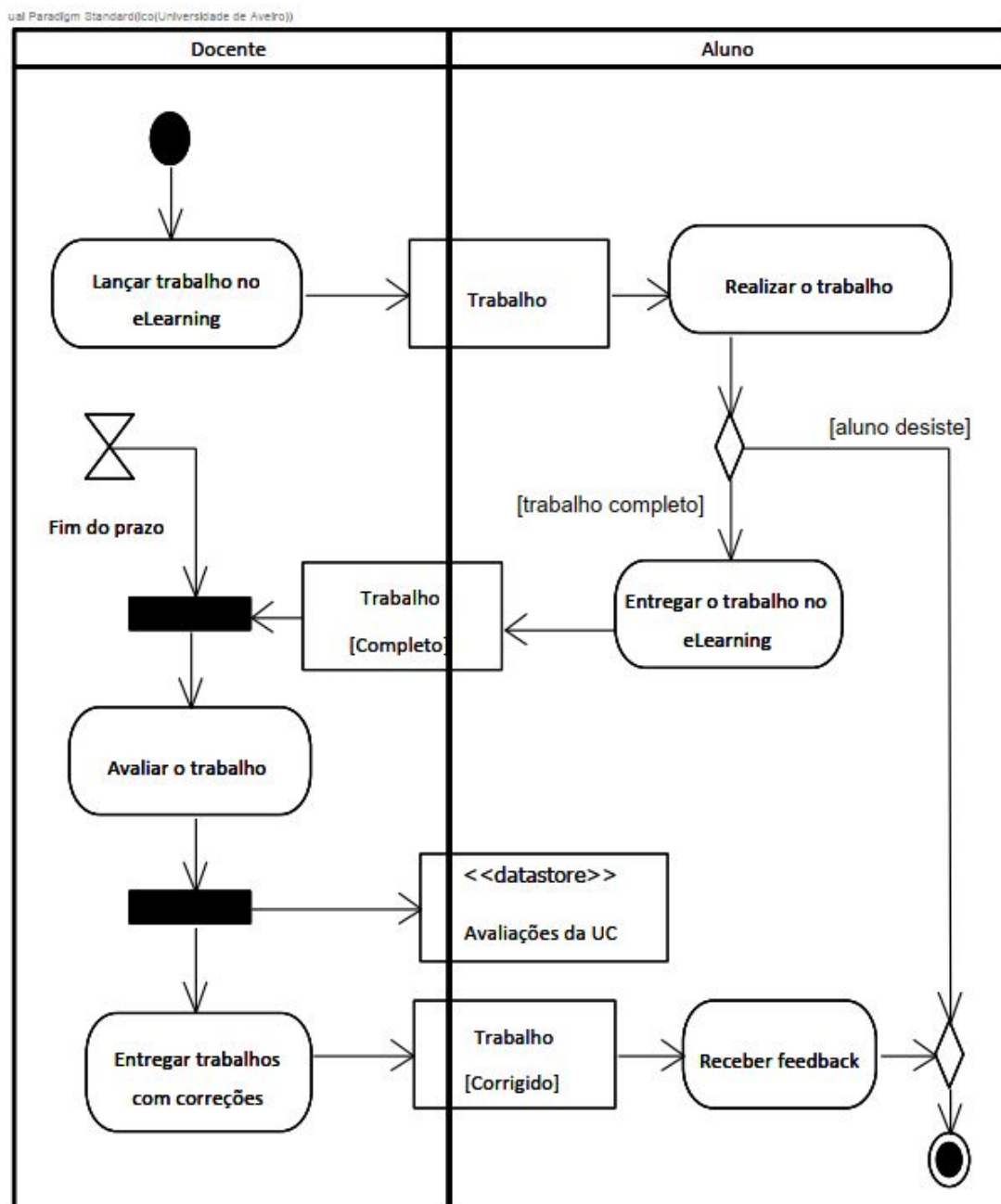
[Option: b\)](#)

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

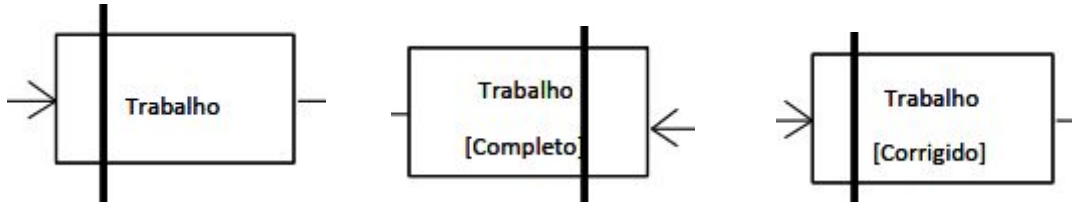
“O caso de utilização “Dados do utente”: não tem fluxo subjacente.”

[Option: e\)](#)

Diagrama 3



17. Relativamente ao processo de trabalho descrito no Diagrama 3:



“O trabalho tem diferentes estados, ao longo da atividade.”

Option: d)

18. 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.”

Option: a)

19. 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.”

Option: a)

20. O Manifesto para o Desenvolvimento Ágil de Software apresenta 12 princípios, nos quais se inclui esta ideia: “os processos ágeis fomentam um desenvolvimento sustentado. Os promotores, a equipa técnica e os utilizadores devem ser capazes de manter um ritmo [de trabalho/progresso] contínuo”.

Explique o que são os “processos ágeis” e como é que contribuem para se atingir o preceito do “desenvolvimento sustentado” num projeto de sistemas de informação.

As ideias principais do Desenvolvimento Ágil de Software é este ser contínuo, desenvolver uma boa relação com o cliente para a melhor partilha de informação acerca dos requisitos ao invés de uma relação formal. Procura portanto o progresso regular e “demonstrável” para que haja partes do sistema funcional a cada iteração deste, para apresentar ao cliente para perceber se os requisitos foram cumpridos. Note-se que esta abordagem permite que as falhas ou mais implementações dos requisitos sejam detectados mais cedo na produção para minimizar os custos de correção dos mesmos. A forma como tal é feito passa também pela simplicidade de cada iteração de trabalho. Através de backlogs e definição de prioridades, as equipas devem conseguir desenvolver o necessário com o menor trabalho possível.

- É um método contínuo
- Envolver o cliente para partilha de informação acerca dos requisitos
- Progresso regular, partes do sistema funcional a cada iteração

- Permite que falhas ou a necessidade de mais requisitos sejam detectadas mais cedo, reduzindo custos
- Simplicidade a cada iteração de trabalho

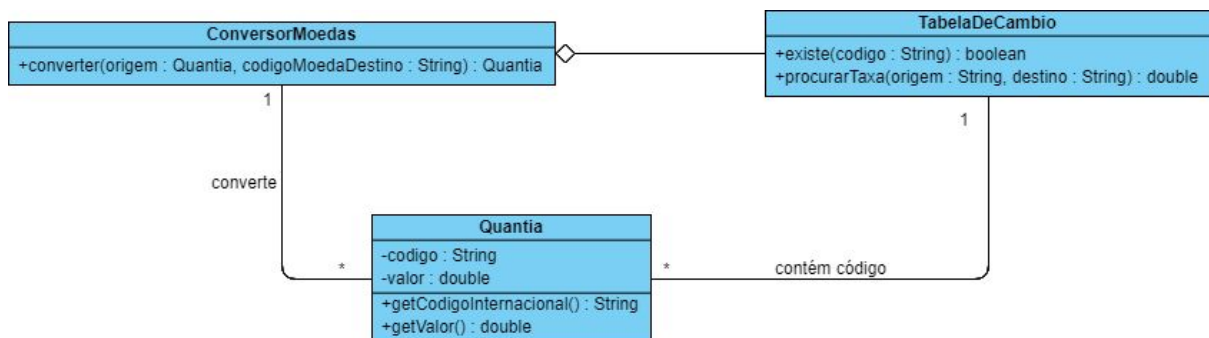
21.

```

3 public class ConversorMoedas { /// algum codigo 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 codigo 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 }

```

a)



b)

