

Engenharia de Software I

Exame Tipo

Nome: _____	Número: _____
-------------	---------------

Considere o enunciado das aulas práticas

O hospital HEAL pretende informatizar parte do seu sistema de prescrições médicas. Neste contexto a prescrição está associada a três tipos possíveis: a prescrição SOS, VERBAL ou de TOMA ÚNICA e a REGULAR. A prescrição regular, diz respeito às tomas que são prescritas pelo médico antecipadamente.

A prescrição SOS é efetuada pelo médico e consiste em medicamentos ou outros que podem ser administrados aos doentes em caso de emergência ou caso este esteja muito queixoso, seguindo as seguintes indicações:

1. Intervalo Mínimo da Toma (Por exemplo PARACETAMOL no máximo de 1 por 8 horas)
2. Posologia da Toma ou Via de Administração
3. Dosagem da Toma (atenção às unidades da toma);

A prescrição verbal ou de toma única consiste em medicamentos ou outros que são administrados ao doente mediante autorização explícita do médico (normalmente pedida via telefone).

A medicação é preparada pelos enfermeiros no início de cada turno, sendo depois administrada posteriormente sem qualquer ordem definida em função do serviço de cada enfermeiro, respeitando os horários de administração das tomas.

Para o controlo da prescrição ser eficiente o sistema deve permitir introduzir e consultar o estado de doentes, assim como a sua medicação associada e pendente. Esta medicação deve ser controlada pelo sistema, no entanto, não é propósito deste sistema a sua utilização para controlo de stocks. A medicação deve ter associada à descrição, código e todos os campos que sejam necessários para o bom desempenho funcional do sistema.

Para responder às questões de escolha múltipla, deverá usar a seguinte grelha.

I	II	III	IV	V	VI	VII	VIII

A duração da prova é de 120 minutos

É descontado ¼ da cotação das respostas incorretas nas questões de escolha múltipla

Poderá entregar a prova 30 minutos após o início da mesma.

I) A metodologia evolutiva com prototipagem é adequada para o projeto, uma vez que

1. os requisitos são bem conhecidos;
2. o projeto apresenta poucos riscos;
3. as características do projeto obrigam à entrega incremental;
4. todas as afirmações anteriores estão erradas.

II) Considerando a escolha da metodologia SCRUM e a implementação do processo de autenticação como parte do primeiro sprint, a equipa

1. vai discutir a funcionalidade implementada durante a reunião de retrospectiva desse sprint;
2. vai discutir o processo de desenvolvimento durante a reunião de retrospectiva desse sprint;
3. vai testar a funcionalidade implementada durante a reunião de retrospectiva desse sprint;
4. nenhuma das anteriores.

III) Considerando o registo da prescrição utilizando um formulário web, a encriptação da comunicação entre o *browser* e o servidor

1. é um requisito funcional da aplicação (perspetiva da segurança);
2. é um requisito funcional externo à aplicação (perspetiva da segurança);
3. não é considerado um requisito da aplicação;
4. nenhuma das anteriores.

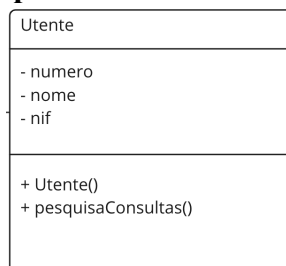
IV) Um erro introduzido na fase de análise de requisitos

1. tem uma mais probabilidade de ser corrigido utilizando uma metodologia de desenvolvimento iterativa;
2. pode ser facilmente corrigido utilizando testes automáticos;
3. pode ser facilmente corrigido através de *code refactoring*;
4. nenhuma das anteriores.

V) A rastreabilidade entre os requisitos e o desenho do sistema permite identificar

1. as classes (do diagrama de classes) associadas a um determinado requisito;
2. as interfaces gráficas associadas a um determinado requisito;
3. as CSS associadas a um determinado requisito;
4. nenhuma das anteriores.

VI) A classe Utente seguinte apresenta fraca coesão, uma vez que



1. todos os métodos estão associados ao conceito associado ao nome da classe;
2. os atributos não estão associados aos métodos que os manipulam;
3. existem métodos que não estão associados ao conceito associado ao nome da classe;
4. nenhuma das anteriores;

VII) Considerando a classe da questão VI, o princípio de design *single responsibility*¹ das regras SOLID

1. é válido para a classe;
2. não é válido para a classe, uma vez que esta apresenta fraca coesão;
3. não é válido para a classe, uma vez que faltam os métodos *setter* e *getter* (ex: *getNumero()*);
4. nenhuma das anteriores;

VIII) A utilização de *frameworks* para o desenvolvimento da aplicação

1. força à criação de código muito estruturado;
2. força à criação de código pouco estruturado, garantindo muita flexibilidade no desenvolvimento da aplicação;
3. deve ser evitada no desenvolvimento de aplicações para a web;
4. nenhuma das anteriores;

Prática

Considere a seguinte *user story*

Enquanto Enfermeiro

Pretendo registar a preparação da medicação da prescrição

Para controlar o processo de administração da medicação pendente

Critérios de aceitação

Dada a lista de prescrições pendentes

Quando registo a preparação da prescrição

Então o estado da prescrição é alterado para *administrada*

Dada a página de pesquisa de prescrições

Quando o estado da prescrição pesquisada é *administrada*

Então não é possível registar a preparação da prescrição

Dado um utilizador não autenticado

Quando a página de pesquisa de prescrições é solicitada

Então é apresentada a respetiva mensagem de erro

- a) **Elabore o diagrama de casos de uso.**

¹ responsabilidade única

b) Elabore o *diagrama de classes*.