

— Exame —  
Desenvolvimento de Sistemas de Informação

LESI/LMCC  
Chamada 2 - 2001/02

25/06/2002

Duração máxima: 2h00  
Leia as questões com atenção.

**Grupo I**

Considere o seguinte extracto de código Java:

```
public class Turma {
    // variáveis de instância
    private Hashtable turma;
    ...
    public int quantosPassam() {
        Enumeration e = this.turma.elements();
        int tot = 0;
        Aluno a;
        boolean more = e.hasMoreElements();

        while (more) {
            a = (Aluno)e.nextElement();
            if (a.getMedia()>=10)
                tot++;
            more = e.hasMoreElements();
        }
        return tot;
    }
    ....
}

public class Aluno {
    // variáveis de instância
    private String nome;
    private String numero;
    private int notaT, notaP;

    public int getMedia() {
        return (this.getNotaT()+this.getNotaP())/2;
    }
}
```

```

    public int getNotaT() {
        return this.notaT;
    }

    public int getNotaP() {
        return this.notaP;
    }
    ...
}

```

1. Escreva um **Diagrama de Sequência** que descreva o comportamento do método `quantosPassam()`.
2. Considere agora que o método `getMédia` é substituído por:

```

    public int getMedia() {
        return (this.notaT+this.notaP)/2;
    }

```

Escreva o **Diagrama de Colaboração** correspondente ao diagrama de sequência da pergunta anterior, mas com a nova versão de `getMedia()`.

## Grupo II

Como parte da preparação para o Euro 2004 foi-lhe pedido que desenvolvesse o software de controlo dos torniquetes de validação de acessos aos estádios. Os torniquetes possuem braços metálicos que podem impedir ou permitir a passagem de pessoas conforme estejam bloqueados ou não. Possuem também um leitor de bilhetes para validação dos bilhetes para os jogos.

A solução que vai propor tem o seguinte modo de funcionamento:

Ao ser ligado o torniquete fica bloqueado. Quando é inserido um bilhete válido o bilhete é retido e o torniquete é desbloqueado, ficando nesse estado até que alguém passe, altura em que bloqueia novamente. Quando é inserido um bilhete inválido, faz soar um alarme e o bilhete é devolvido. O sistema mantém o alarme até o bilhete ser retirado do leitor, altura em que volta a estar bloqueado. Enquanto está desbloqueado o torniquete não aceita bilhetes.

O sistema possuirá também um modo de teste para permitir a técnicos analisar o funcionamento do torniquete. Quando em modo de teste os técnicos podem realizar dois tipos de testes: testar os braços e testar o leitor de bilhetes. Para testar os braços o técnico pode bloquear e desbloquear o torniquete. Para testar o leitor de bilhetes o técnico insere bilhetes (que poderão ser válidos ou inválidos) que serão sempre devolvidos. No caso do bilhete ser inválido o sistema soa o alarme até o técnico retirar o bilhete. Os dois tipos de teste podem ser executados em paralelo.

O modo de teste pode ser activado em qualquer altura. Quando sai de modo de teste o sistema fica na situação de bloqueado. O sistema pode ser desligado em qualquer altura.

Escreva um **Diagrama de Estado** que represente a solução proposta.

### Grupo III

**mediateca** s. f., (neol.), arquivo organizado de material e de equipamento relativo a meios de comunicação áudio (registos discográficos ou de emissões radiofónicas), visuais (fotografias e outro material de reprodução gráfica) e audiovisuais (filmes, vídeos, etc.)

*Dicionário da Língua Portuguesa Online, Texto Editora.*

Considere que lhe é pedido o desenvolvimento de um Sistema de Informação para uma Mediateca. O sistema deverá permitir registar informação relativa aos vários tipos de documentos. Todos os documentos deverão possuir um número de registo que os identifica, bem como um nome, uma descrição sinóptica, uma classificação etária (>0, >6, >12, >16, >18) e o endereço do ficheiro que contém o documento.

O sistema deverá ainda permitir acesso aos documentos (para tal todos os documentos deverão possuir um método `play` que permita a sua visualização/audição). Nesta fase vão apenas ser considerados três tipos de documento (`mp3` para documentos áudio; `png` para documentos visuais; `mpeg` para documentos audiovisuais). A solução proposta deverá facilitar a adição de novos tipos de documento em fase posterior.

Para permitir referências cruzadas entre documentos, cada documento deverá possuir uma lista de referências para outros documentos no sistema.

O sistema deverá contemplar dois tipos de utilizadores: utentes e administradores. Todos os utilizadores possuem um nome de utilizador e uma senha de acesso. Para cada utente deverá ser registado o seu nome e morada, bem como a sua data de nascimento.

Os utentes utilizam o sistema para procurar e consultar documentos. A procura é efectuada indicando uma frase chave, a consulta é efectuada indicando o número do registo do documento (esse número poderá ser obtido através de uma procura). No entanto, o acesso a um documento só é permitido se a classificação etária do documento não for incompatível com a idade do utente.

Os utentes podem ainda propor novos documentos para inclusão na Mediateca. Nesse caso a proposta é regista e fica pendente de aprovação por um administrador.

Os administradores fazem a gestão da informação no sistema, pelo que podem adicionar e remover utentes, bem como adicionar e remover documentos. É ainda incumbência dos administradores aceitar (ou não) os documentos propostos pelos utentes. Caso o documento seja aceite, o administrador deverá atribuir-lhe uma classificação etária.

Responda às seguintes questões (note que os diagramas pedidos são interdependentes):

1. Identifique os Actores e Use Cases contidos na descrição dada e escreva o **Diagrama de Use Case** correspondente.
2. Proponha um **Diagrama de Classe** para o sistema (inclua as variáveis e métodos que considerar mais relevantes).
3. Proponha um **Diagrama de Sequência** para o método que determina um vector com os números de registo de todos os documentos que satisfazem uma dada frase chave:

```
public Vector procura(String chave)
```

**nota:** o método `public int indexOf(String str)` da classe `String` retorna o índice da primeira ocorrência de `str` ou `-1` se `str` não existe