

## 21007 - Análise de Sistemas

### 2016-2017 e-fólio Recurso

**PARA A RESOLUÇÃO DO E-FÓLIO, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:**

- 1) O e-fólio é constituído por 3 perguntas. A cotação global é de 4 valores.
- 2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 50% a 100%.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normalização “eFolioX” + <nº estudante> + <nome estudante com o máximo de 3 palavras>. Penalização de 50% a 100%.
- 5) Na primeira página do e-fólio deve constar o nome completo do estudante bem como o seu número. Penalização de 50% a 100%.
- 6) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- 7) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 8) A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

## 1 - Projeto UML

Para usar o metropolitano, um cliente deve ter um bilhete eletrónico previamente carregado com uma certa quantia de dinheiro (por simplicidade, assumir que o pagamento só é realizado com moedas e notas, e que a máquina de venda automática pode dar troco, se necessário). O bilhete eletrónico pode ser comprado e carregado em máquinas de venda automática especiais localizadas nas estações de metro. O dono do bilhete usa o bilhete numa máquina de validação de ingressos, para aceder ao terminal de embarque/desembarque do metro, e esta ação resulta em debitar o valor de uma viagem (por simplicidade, assumir que todas as viagens custam o mesmo).

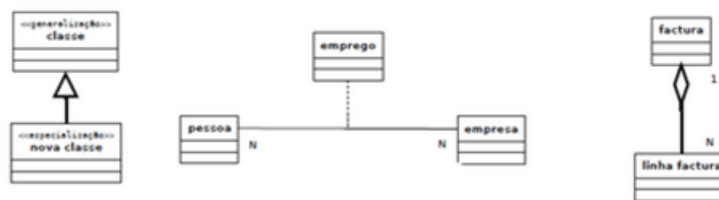
Se o ingresso não tiver crédito suficiente, as portas para o terminal não serão abertas. Quando o cliente chega ao seu destino, ele usa o bilhete numa máquina de validação de saída, para que possa terminar a sua viagem. Um cliente pode pedir um reembolso do valor restante dentro do ingresso, devolvendo o ingresso a uma máquina de vendas. Um funcionário do metro recolhe o dinheiro das máquinas de venda automática, de tempos a tempos, mas sem uma periodicidade pré-definida.

Todos os dias, o sistema gera um relatório com todas as vendas nas máquinas de venda automática e envia esse relatório para um Administrador, que tem a capacidade de atualizar os preços dos ingressos.

Na sua análise não exceda 7 atores, 7 casos de utilização (funcionalidades) e 7 classes. A utilização de atores, funcionalidades ou classes que não existam no texto levará à aplicação de penalizações.

1.1 (0,5 valor) Elabore um diagrama de "use-cases"

1.2 (0,5 valores) Construa o Diagrama de Classes com as regras definidas nas atividades formativas anteriores. Os símbolos a utilizar são os seguintes:



3 (0,5 valores) Elabore a tabela CRUD, i.e., matriz funcionalidades versus classes.

A estrutura da matriz é a seguinte:

Funcionalidades \ Classes	A	B	C	D
F1				
F2				
F3				
F4				
Contadores C R U D				

1.4 (0,5 valores) Para a funcionalidade 'Enviar relatório de vendas', crie um diagrama de sequência.

2 (1,0 valor) – Crie um diagrama de atividades para a função chamada 'particiona' que recebe como argumentos uma lista, **lst**, e dois elementos, **l**, e **s**. Esta função devolve uma lista de três elementos, contendo na primeira posição a lista com os elementos de **lst** menores que **l**, e na segunda posição a lista com os elementos de **lst** entre **l** e **s** inclusive, e na terceira posição a lista com os elementos de **lst** maiores que **s**.

Por exemplo:

```
>>> particiona([2, 0, 12, 19, 5], 5,12)
```

```
[[2, 0], [5, 12], [19]]
```

```
>>> particiona([7, 3, 4, 12], 3, 9)
```

```
[[], [7, 3, 4],[12]]
```

3 (1,0 valor) – Crie um diagrama de estados que reconhece um número binário com o mesmo número de bits **1** e **0**.

Por exemplo:

```
>>> 1011010 (não reconhece)
```

```
>>> 0110110 (não reconhece)
```

```
>>> 11100010 (reconhece)
```