

**21177 – Modelação de Sistemas de Informação**  
**2017-2018**  
**e-fólio A**  
**Resolução e Critérios de Correção**

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

- 1) O e-fólio é constituído por 4 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.

A avaliação do estudante está contida no seguinte vetor de cotações parciais:

Questão:	1	2.1	2.2	2.3
Cotação:	10	15	10	5 décimas

1- (1 valor) Construa um Fluxograma utilizando os símbolos de início/fim, decisão e processamento, para descrever os seguintes algoritmos:



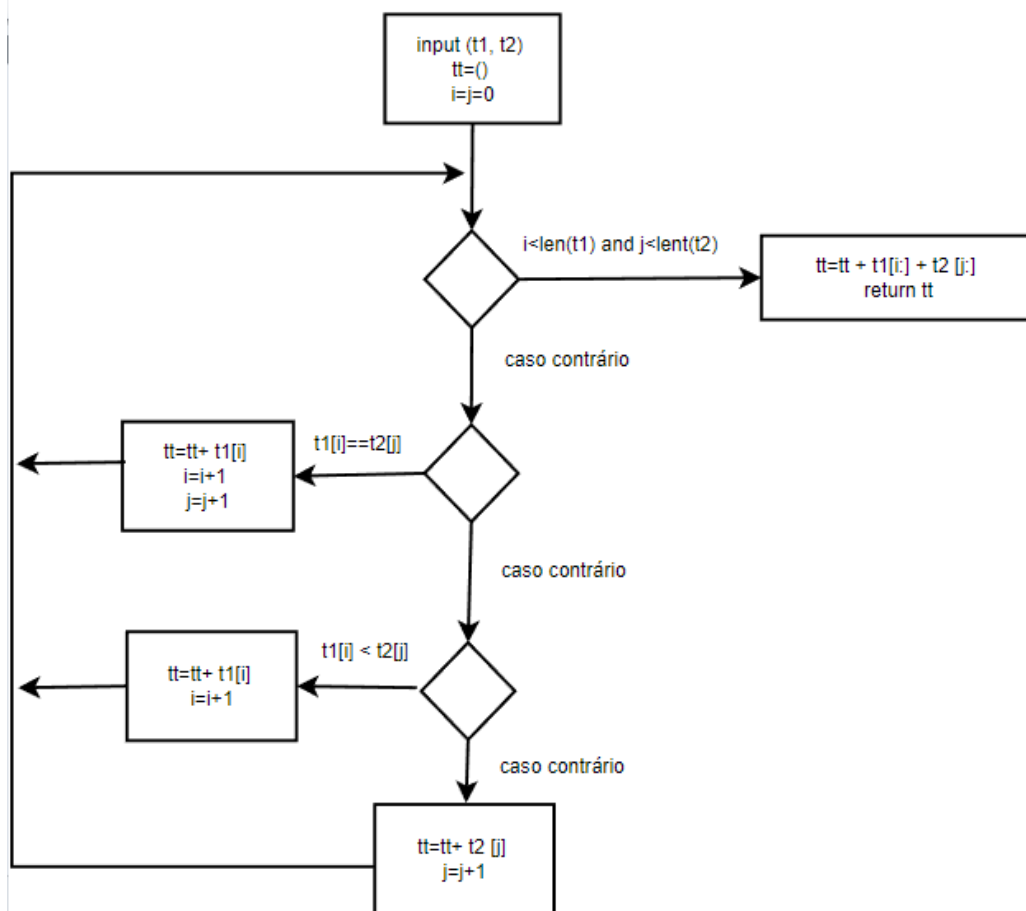
Defina uma função “junta ordenados” que recebe duas listas contendo inteiros, ordenados por ordem crescente e devolve uma lista também ordenada com os elementos das duas listas.

Por exemplo: junta\_ordenados ((2, 34, 200, 210), (1, 23)), devolve a seguinte lista (1, 2, 23, 34, 200, 210).

Resposta:

Em Python a função ‘junta\_ordenados’ (merge) será:

```
def merge (t1,t2):  
    tt=()  
    i=j=0  
    while i<len(t1) and j<len(t2):  
        if t1[i]==t2[j]:  
            tt=tt+(t1[i,]);i=i+1;j=j+1  
        elif t1[i]<t2[j]:  
            tt=tt+(t1[i,]); i=i+1  
        else:  
            tt=tt+(t2[j,]); j=j+1  
    tt=tt+t1[i:]+t2[j:]  
    return tt  
  
print (merge( (2, 34, 200, 210), (1, 23) ))
```



Comentários e Critérios de correção:

- devem ser utilizados exclusivamente os símbolos: processo (retângulo) e decisão (losango) e conectores (círculo ou similar);
- ciclo adequado com 3 losangos;
- penalização para a dificuldade de leitura (linhas cruzadas, letras com fontes desadequadas): penalização de 20% a 100%
- penalização de 70% para quem utilizar o SORT; a eficiência do SORT é de  $O(N \cdot \log N)$  e o merge de  $O(N)$
- erros e omissões: penalização de 20% a 100%

2 – Pretende-se que faça a modelação das principais funcionalidades de uma Empresa de Automóveis Usados.



Após várias consultas com a comissão da empresa, a sua empresa de *software* recolheu as seguintes informações:

A empresa compra automóveis usados, envia para o automóvel para reparação e limpeza, e coloca-o à venda no ‘stand’.

Cada automóvel é identificado pela matrícula. Tem associado os dados que constam no livrete como a marca, modelo e data da matrícula. São ainda registadas as datas de compra e venda, o preço de compra, o preço da reparação e limpeza, e por cada mês no ‘stand’ é acrescido um valor de estacionamento de 100 euros. Também tem um estado: iniciado, em negociação e concluído.

Os funcionários da agência podem ser comerciais (venda) ou angariadores (compra). Um deles é o diretor comercial da agência. O comercial atende clientes à procura de comprar automóveis. O angariador atende clientes que pretendem vender automóveis. Assim, os clientes da agência podem ter dois papéis: compradores ou vendedores. De reparar que uma mesma pessoa, ao longo do tempo, pode ser comprador e vendedor.

Para além da informação típica de funcionário, os comerciais e angariadores podem trabalhar numa mesa de atendimento diferente em cada dia de trabalho. O diretor comercial (CEO) é responsável por todos os funcionários e é ele que gere a lista de funcionários (inserção, remoção, atualização dos seus dados).

O angariador insere clientes vendedores, e possui a capacidade de introduzir novos automóveis no sistema.

O comercial insere clientes compradores, e é capaz de iniciar e concluir negócios de venda.

Um cliente tem um nome, profissão, contactos e um historial de negócios.



O diretor comercial (CEO), que na infância lia a banda desenhada do Tio Patinhas, pretende ter um gráfico dos lucros mensais. Pretende ainda saber os montantes envolvidos dos angariadores e dos comerciais em cada mês.

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.

## 2.1- (1½ valor) *Use-Cases*, Casos-de-uso ou Funcionalidades

### (a) Defina os requisitos funcionais do sistema

Resposta:

- O sistema deve permitir a inclusão, alteração e remoção de dados de funcionários (diretor, angariador e vendedor), clientes (vendedor e comprador), automóveis (compras e vendas).
- O diretor comercial (CEO) pretende ter um gráfico dos lucros mensais. Pretende ainda saber os montantes envolvidos dos angariadores e dos comerciais em cada mês.

**CrITÉrios de correção:**

- cotação 5 décimas
- inclusão, alteração e remoção de dados (50%)
- gráfico lucros (50%)

(b) Construa a Matriz Atores (nomes) versus Casos-de-uso (verbos)

Resposta:

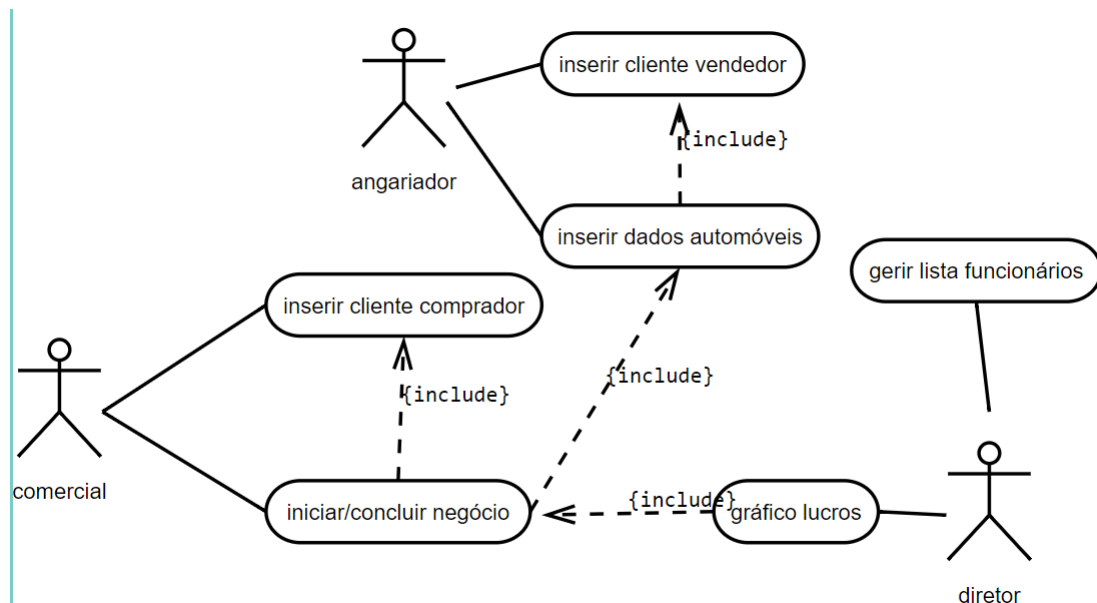
casos-de-uso \ atores	diretor comercial	angariador	comercial	cliente comprador	cliente vendedor
gerir dados funcionários	X				
inserir cliente vendedor		X			
inserir dados automóvel		X			
inserir cliente comprador			X		
iniciar/fechar negócio venda			X		
consultar dados (extra texto)				X	X
gerar gráfico dos lucros	X				

Comentários e Critérios de correção:

- cotação 5 décimas
- casos-de-uso e atores adequados (50%)
- ligações adequadas (50%)
- penalização ao colocar como atores ou use-cases desajustados

(c) Apresente o Diagrama de Casos-de-uso

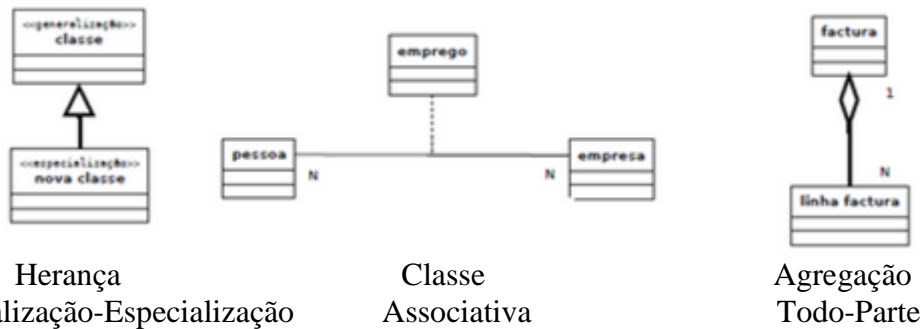
Resposta:



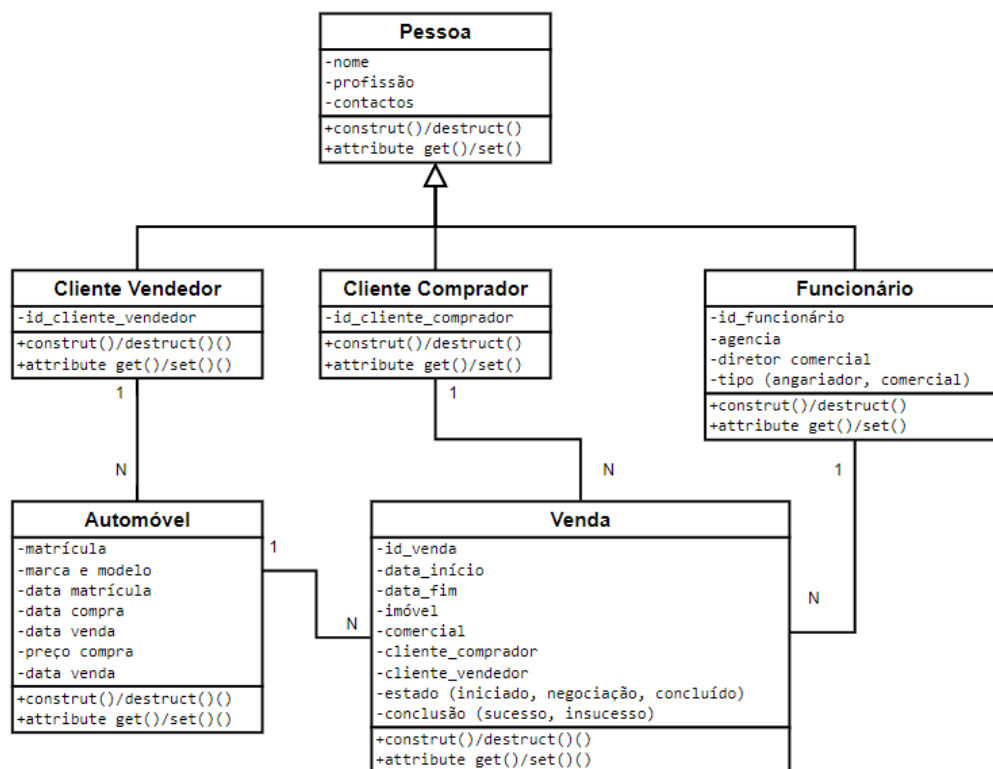
Comentários e Critérios de correção:

- cotação 5 décimas
- penalização para a dificuldade de leitura (linhas cruzadas, letras com fontes desadequadas): -2 a 3 décimas;
- penalização para atores ou ligações desadequadas: -2 a -5 décimas;

2.2- (1 valor) Construa o Diagrama de Classes com as regras definidas nas atividades formativas anteriores. Os símbolos a utilizar são os seguintes:



Resposta:



Comentários e Critérios de correção:

- classes adequadas: 5 décimas;
- cardinalidade e símbolos de relação: herança, associação e agregação: 5 décimas;
- penalização de classes não referidas no texto ou desadequadas: -5 décimas;
- penalização para a dificuldade de leitura (linhas cruzadas, letras com fontes desadequadas): -5 décimas;

2.3- (1/2 valor) Construa a matriz CRUD (create, read, update, delete), com os Casos-de-Uso versus as Classes encontradas. A estrutura da matriz é a seguinte:

Casos-de-uso \ Classes	A	B	C	D
X				
Y				
Z				
W				
Contadores C R U D				

Resposta:

<b>casos-de-uso</b> \ <b>classes</b>	C. Vendedor	C. Comprador	Funcionário	Automóvel	Venda
gerir acessos			CRUD		
inserir cliente vendedor	CRUD				
inserir dados imóvel	R			CRUD	
inserir cliente comprador		CRUD			
iniciar/fechar negócio	R	R	R	R	CRUD
gerar gráfico dos lucros				R	R
<b>contadores CRUD</b>	1311	1211	1211	1311	1211

Comentários e Critérios de correção:

- deve existir um único "use-case" que realize o "Create" e o "Delete", podendo existir um número limitado de "use-cases" para o "Update" e vários para o "Read";
- matriz CRUD: 3 décimas
- outros e contadores CRUD: 2 décimas