

 <p><b>FCTUC</b></p> <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p><i>Departamento de Engenharia Informática</i></p>	<p align="center"><b>Trabalho nº 4 de Algoritmos e Estruturas de Dados 2018-2019 – 2º Semestre</b></p> <p><b>Data-limite de submissão no Mooshak:</b></p> <ul style="list-style-type: none"> <li>• Tarefa A: 24 de Maio de 2019, 23h59</li> <li>• Tarefa B: 31 de Maio de 2019, 23h59</li> </ul> <p><b>Data-limite de entrega do relatório:</b></p> <ul style="list-style-type: none"> <li>• 03 de junho de 2019, 20h00.</li> </ul> <p>Em papel, manuscrito e entregue no cacifo do respetivo docente das aulas PL.</p>
<p><i><u>Nota Importante:</u> A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.</i></p>	

**Objetivos:** Tabelas de Dispersão e estruturas similares.

**Notas Importantes:**

- O relatório do trabalho deve ser realizado, MANUSCRITO, no formulário que será oportunamente fornecido.
- A não entrega do relatório do trabalho no prazo-limite estabelecido implica que o aluno tem uma classificação de ZERO na avaliação deste trabalho.

**As várias tarefas incluídas neste Trabalho Prático correspondem à resolução de um mesmo problema (descrito nas páginas seguintes, e similar ao que foi usado no Trabalho Prático N.2) usando diferentes alternativas de implementação:**

- Tarefa A – Solução baseada numa tabela de *hashing* com resolução de colisões com base em *chaining* ou *bucket chaining*.
- Tarefa B – Solução baseada numa tabela de *hashing* com resolução de colisões com base em *probing*.

### Descrição do Problema – “Base de Dados SCUT”

*O problema é essencialmente semelhante ao problema já usado anteriormente, no Trabalho Prático N.2, com exceção de alguns comandos que foram removidos.*

Para dar suporte ao registo e à cobrança de portagens nas autoestradas SCUT, foi-lhe pedido que crie uma plataforma que permita:

- atualizar rapidamente o saldo das viaturas que passam por pórticos SCUT (valor em dívida) e os pagamentos feitos nas estações de correios (abate de dívida).
- obter rapidamente o saldo em dívida para uma dada viatura, com base na matrícula.

Cada viatura é identificada pela matrícula – uma *string* com menos de 10 caracteres (tal como, por exemplo “88AA00”, podendo existir matrículas estrangeiras). O valor em dívida é sempre um inteiro (cêntimos em dívida).

#### Input:

O seu programa deve ler uma sequência de linhas (até encontrar o comando TERMINA), cada uma das quais correspondente a uma das seguintes operações:

**PORTICO** <Matrícula> <Custo>

Se o veículo ainda não está na base de dados é adicionado, com um valor em dívida de *Custo* (em cêntimos).

Se o veículo já está na base de dados o valor em dívida é atualizado com a soma do custo da nova passagem num pórtico.

Caso o valor em dívida fique nulo (zero) o veículo deve ser retirado da base de dados.

**PAGAMENTO** <Matrícula> <Valor>

Pagamento nos balcões dos correios de *Valor* cêntimos, que abatem à dívida existente do veículo. É possível que o pagamento leve a valor em dívida negativo (caso o valor pago seja superior ao valor em dívida).

Caso o valor em dívida fique nulo (zero) o veículo deve ser retirado da base de dados.

**SALDO** <Matrícula>

Esta operação indica o valor por pagar correspondente ao veículo *Matrícula*, na forma:

<Matrícula> VALOR EM DIVIDA <Saldo>

Se o veículo não estiver na base de dados é mostrada a seguinte informação:

<Matrícula> SEM REGISTO

**TERMINA**

Para terminar a execução da aplicação.

### Exemplo de *Input*:

```
PORTICO 00AA01 350[\n]
PORTICO 00BB01 40[\n]
PORTICO 00BB01 40[\n]
SALDO 00CC01[\n]
SALDO 00AA01[\n]
PORTICO 00AA01 900[\n]
PAGAMENTO 00AA01 500[\n]
SALDO 00AA01[\n]
PORTICO 00DD01 150[\n]
PAGAMENTO 00EE01 100[\n]
SALDO 00DD01[\n]
SALDO 00EE01[\n]
PORTICO 00EE01 100[\n]
TERMINA[\n]
```

### *Output* correspondente:

```
00CC01 SEM REGISTO[\n]
00AA01 VALOR EM DIVIDA 350[\n]
00AA01 VALOR EM DIVIDA 750[\n]
00DD01 VALOR EM DIVIDA 150[\n]
00EE01 VALOR EM DIVIDA -100[\n]
```

## Relatório

O *template* para o relatório, assim como os ficheiros a usar nas correspondentes medições de desempenho, serão oportunamente divulgados.

De qualquer modo, será pedido no relatório que indique e justifique:

- As alternativas técnicas escolhidas para a função de dispersão e a resolução de colisões, devidamente justificadas.
- A solução adotada para gerir dinamicamente a dimensão da tabela de dispersão.
- A solução adotada para usar a matrícula como chave da tabela.

O relatório terá ainda uma vertente de comparação entre as tabelas de dispersão a usar neste trabalho, e as árvores binárias usadas no trabalho prático N.2, em termos funcionais e de desempenho.