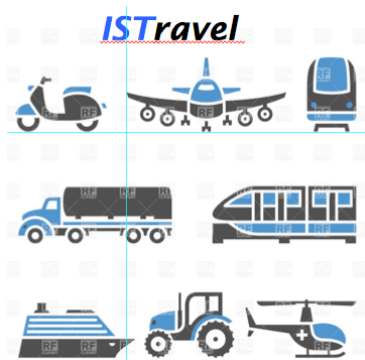


Instituto Superior Técnico

Algoritmos e Estruturas de Dados

Enunciado do Projecto



ISTRavel

Versão 1.1 (22/11/2014)

2014/2015
1º Semestre

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 2 |
| 2 | O agente de viagens ITravel | 2 |
| 3 | O programa AGENTE DE VIAGENS | 4 |
| 3.1 | Execução do programa | 4 |
| 3.2 | Formato de entrada de dados | 5 |
| 3.2.1 | Ficheiro .map | 5 |
| 3.2.2 | Ficheiro .cli | 5 |
| 3.3 | Formato de saída de dados | 7 |
| 4 | Avaliação do Projecto | 9 |
| 4.1 | Funcionamento | 10 |
| 4.2 | Código | 10 |
| 4.3 | Relatório | 10 |
| 4.4 | Discussão | 11 |
| 5 | Código de Honestidade Académica | 12 |

Revisões

| | |
|-------------------------------------|---|
| Versão 0.1 (03 de Outubro de 2014) | Versão inicial |
| Versão 0.2 (06 de Outubro de 2014) | Versão com restrições |
| Versão 1.0 (23 de Outubro de 2014) | Versão completa |
| Versão 1.1 (22 de Novembro de 2014) | Correcção do ficheiro de saída (Fig. 4) |

1 Introdução

Neste projecto pretende-se desenvolver um programa "agente de viagens", o qual determina e sugere a melhor rota, e os meios de transporte necessários, para levar um cliente de uma determinada cidade até à cidade desejada.

Os pontos de interesse (cidades de partida, de chegada e eventualmente de passagem) estão definidos numa rede de ligações, na qual se estabelecem o conjunto de cidades a considerar, a(s) ligação(ões) que existe(m) entre pares de cidades, bem como o(s) tipo(s) de transporte que as liga(m).

Para cada ligação estão definidos o tempo de duração da viagem entre as duas cidades ligadas, bem como o custo dessa viagem para o cliente da *ISTravel*.

O programa pretendido deverá fornecer para cada cliente a sequência de ligações e o respectivo meio de transporte que permita fazer a viagem entre a cidade de partida e a de chegada de forma óptima, de acordo com o critério especificado pelo cliente: tempo ou custo.

2 O agente de viagens ISTravel

O trabalho do AGENTE DE VIAGENS envolve uma rede de nós (cidades) ligados por um ou mais meios de transporte. Neste projecto consideram-se quatro meios de transporte: autocarro, comboio, avião e barco. A Figura 1 mostra um exemplo de um mapa de uma rede de cidades com as ligações (e respectivo meio de transporte) existentes. Neste exemplo, entre Lisboa e Santarém existe apenas uma ligação através de autocarro. Já entre Faro e Beja existem duas ligações, uma por comboio e a outra por autocarro.



Figura 1: Exemplo de um mapa para o agente de viagens.

Note-se por exemplo que para ir do Porto a Madrid existem várias rotas possíveis: ligação directa por comboio ou autocarro, via Chaves (comboio ou autocarro) e depois Madrid necessariamente

de avião, e ainda via Aveiro, Faro (barco ou comboio), Lisboa (os 4 meios), e finalmente Madrid (comboio ou avião).

A escolha de qual destas rotas entre o Porto e Madrid é a melhor depende da preferência do cliente: se pretende minimizar o tempo de viagem ou o custo da mesma. Para esse efeito, existem associados a cada ligação o respectivo tempo e custo da viagem.

As ligações têm uma determinada periodicidade ao longo de um dia. Para este efeito, considera-se que um dia tem uma duração de 1440 minutos, sendo o minuto a unidade de tempo padrão, nunca se considerando fracções da mesma.

Assim, na definição de uma ligação, para além do tempo e custo da viagem, definem-se três parâmetros adicionais:

- o instante de tempo t_i em que sai o primeiro transporte dessa ligação - um valor entre 0 e 1440;
- o instante de tempo t_f a partir do qual não sai mais nenhum transporte dessa ligação - um valor entre 0 e 1440 que satisfaça $t_f \geq t_i$, e
- a periodicidade do transporte p .

Neste projecto, considera-se que as ligações são bidireccionais, ou seja o transporte que existe entre A e B também existe entre B e A , e os parâmetros definidos para a ligação são iguais quer a viagem seja entre A e B quer entre B e A .

Como exemplo, suponha-se que numa ligação entre as cidades A e B os parâmetros são $t_i = 0$, $t_f = 1010$ e $p = 200$. Isto significa que para essa ligação, num dia, há transportes a saírem da cidade A para a cidade B , e da cidade B para a cidade A , nos instantes $t = 0$, $t = 200$, $t = 400$, $t = 600$, $t = 800$, e $t = 1000$. Note-se que esta situação não significa que a viagem tem que durar menos de 200, porque se assume que a frota disponível é infinita. Ou seja, não é necessário considerar que um transporte que vá de A para B , a seguir regressa a A .

Outro exemplo de parametrização de uma ligação: $t_i = 100$, $t_f = 1010$ e $p = 1000$. Neste caso, há apenas um transporte por dia que sai no instante 100. Ou seja, sempre que numa ligação sucede que $p > t_f - t_i$, a ligação em causa tem apenas um transporte diário. Para este caso, o transporte seguinte será apenas no dia seguinte, i.e. no instante $1440 + 100 = 1540$.

Adicionalmente, os clientes podem colocar algumas restrições relativamente às ligações que aceitam para construir a rota desejada. Neste projecto, devem-se considerar as seguintes restrições (**tipo A**), mas apenas uma para cada cliente:

- A.1 indicação de não mais do que um meio de transporte que o cliente **não quer** utilizar;
- A.2 tempo máximo para uma ligação;
- A.3 custo máximo para uma ligação.

As restrições acima condicionam as ligações. Como exemplo, considere-se um cliente que estelece como restrição que não quer andar de avião. Isto significa que a rota a determinar para este cliente não deve incluir nenhuma ligação por avião; mas se o cliente definir como restrição que o tempo máximo por ligação é de 1000, então a rota pode ter um tempo total superior a 1000, mas cada uma das ligações que a constituem não podem ter um tempo de viagem superior a 1000. Idêntico raciocínio para o custo.

Finalmente, ainda para cada cliente, este pode definir uma restrição adicional (**tipo B**):

B.1 se escolher como preferência (critério de optimização) o tempo, o cliente pode estabelecer que não aceita rotas cujo tempo total seja superior a um valor especificado; ou

B.2 se escolher como preferência (critério de optimização) o custo, o cliente pode estabelecer que não aceita rotas cujo custo total seja superior a um valor especificado.

O objectivo do projecto é determinar para um conjunto de clientes da *ISTravel*, que indicam a cidade de partida, a cidade de chegada, o critério de preferência, e eventualmente uma restrição do tipo A e/ou uma restrição do tipo B, qual a melhor rota entre as duas cidades, identificando as ligações que a constituem (i.e., qual o meio de transporte).

3 O programa AGENTE DE VIAGENS

O programa a desenvolver deve conseguir ler a configuração de uma rede de cidades, contendo informação sobre as cidades e as ligações que existem, bem como a informação sobre os clientes, e gerar para cada cliente a rota que melhor serve os seus interesses.

Naturalmente, o programa deve estar também preparado para lidar com um cliente que pretende viajar entre duas cidades para as quais não existe qualquer rota, ou coloca restrições que são impossíveis de satisfazer.

Neste projecto não se deve assumir qualquer pressuposto que os dados fornecidos estejam em consonância com a realidade. Por exemplo, não é necessário que uma viagem de avião demore menos tempo e/ou custe mais que a mesma viagem feita por qualquer dos outros meios de transporte.

Nas secções seguintes descreve-se a forma como o programa deve ser invocado, a forma como a informação da rede e dos clientes é descrita e o formato que têm de ter os dados de saída, ou seja a forma de descrever as rotas encontradas.

3.1 Execução do programa

O programa AGENTE DE VIAGENS deverá ser invocado na linha de comandos da seguinte forma:

```
aed$ ./istravel input1.map input2.cli
```

onde:

istravel designa o nome do ficheiro executável contendo o programa AGENTE DE VIAGENS;

input1.map é um exemplo do nome do ficheiro contendo a descrição da rede de cidades, de acordo com o formato indicado na Secção 3.2.1.

input2.cli é um exemplo do nome do ficheiro contendo a informação dos clientes, de acordo com o formato indicado na Secção 3.2.2.

Os ficheiros utilizados para a descrição da informação de partida terão de ter a extensão `.map` para a rede de cidades, e `.cli` para os clientes, mas poderão ter qualquer nome, nada sendo assumido para esse efeito. Se se pedir a execução do programa indicando-se o nome de um ficheiro que não exista ou sem indicar qualquer nome, o programa deve simplesmente abortar a execução. Assume-se que os ficheiros de entrada estão sempre correctos de acordo com o formato de entrada, pelo que o programa a desenvolver não tem de se preocupar com a detecção de erros nos dados de entrada.

3.2 Formato de entrada de dados

A descrição da configuração de partida do AGENTE DE VIAGENS é feita em dois ficheiros de entrada.

3.2.1 Ficheiro .map

A descrição da rede de cidades é feita num ficheiro com extensão `.map`, obedecendo às seguintes regras:

- a 1ª linha contém dois números inteiros, N e L , em que N representa o número de cidades e L o número de ligações.
- as seguintes L linhas correspondem às ligações existentes. Cada linha (ligação) contém oito elementos separados por um espaço em branco:
 - dois números inteiros correspondentes à identificação das duas cidades ligadas;
 - o meio de transporte da ligação, que pode ser uma das seguintes palavras: **aviao**, **comboio**, **barco** ou **autocarro**;
 - o tempo de duração da ligação;
 - o custo da ligação;
 - o instante da primeira partida num dia (t_i);
 - o instante do dia após o qual não haverá mais partidas (t_f); e
 - a periodicidade diária (p).

Para a definição das cidades de uma ligação, considera-se que as cidades estão numeradas de 1 a N .

Nota: todos os tempos e custos referidos atrás são representados por números inteiros.

A título de exemplo, o ficheiro que descreve a rede¹ da Figura 1, é o indicado na Figura 2. Considerou-se a seguinte numeração das cidades: 1 - Lisboa, 2 - Évora, 3 - Beja, 4 - Faro, 5 - Santarém, 6 - Guarda, 7 - Madrid, 8 - Aveiro, 9 - Porto, 10 - Braga, 11 - Chaves, 12 - Dili.

3.2.2 Ficheiro .cli

A descrição dos clientes é feita num ficheiro com extensão `.cli`, obedecendo às seguintes regras:

- a 1ª linha contém um número inteiro, C , que representa o número de clientes.
- as seguintes C linhas correspondem à informação dos clientes da *ISTravel*. Cada linha (cliente) contém seis elementos obrigatórios separados por um espaço em branco:
 - um número inteiro correspondente ao número de cliente (um número entre 1 e C);
 - dois números inteiros correspondentes à identificação da cidade de partida e da cidade de chegada desejada;
 - um número inteiro não negativo com o instante mínimo de disponibilidade para a partida do cliente;

¹à qual se acrescenta a cidade de Dili, sem nenhuma ligação às restantes.

```
12 21
4 3 comboio 60 10 0 1440 100
4 1 comboio 100 14 500 1200 500
4 3 autocarro 70 9 600 1440 300
1 4 aviao 10 30 480 1440 1440
4 1 barco 90 21 0 1000 200
4 1 autocarro 90 15 0 1000 200
1 5 autocarro 20 5 0 1000 200
2 6 aviao 80 14 0 1000 200
7 1 aviao 200 100 0 1000 200
1 7 comboio 1111 75 0 1000 200
4 8 barco 220 93 0 1000 200
4 8 comboio 132 61 0 1000 200
9 8 barco 22 9 0 1000 200
9 8 comboio 13 13 0 1000 200
9 10 autocarro 5 2 0 1000 200
11 9 comboio 19 9 0 1000 200
11 9 autocarro 18 8 0 1000 200
9 7 comboio 302 101 0 1000 200
7 9 autocarro 1221 90 0 1000 200
11 7 aviao 2001 110 0 1000 200
5 6 barco 5 5 0 1000 200
```

Figura 2: Exemplo do ficheiro .map para a rede apresentada na Figura 1.

- a preferência do tipo de rota, identificada por uma de duas palavras: **tempo** ou **custo**. A primeira palavra significa que o cliente pretende a rota de menor tempo, enquanto a segunda significa que pretende a rota de menor custo; e
- o número de restrições do cliente (0, 1 ou 2).

Para além dos elementos obrigatórios, a linha correspondente a um cliente pode incluir uma ou duas restrições, i.e., um ou dois pares opcionais de elementos:

- um par constituído por uma das seguintes palavras: **A1**, **A2** ou **A3**, seguida de uma das seguintes palavras: aviao, comboio, autocarro ou barco, caso a primeira seja A1, ou um número inteiro caso a primeira seja A2 ou A3.
- O outro par possível é constituído por uma das seguintes palavras: **B1** ou **B2**, seguida de um número inteiro.

Nota: todos os tempos e custos referidos atrás são representados por números inteiros.

A título de exemplo, apresenta-se na Figura 3 a informação relativa a um conjunto de 5 clientes.

No exemplo de ficheiro .cli apresentado na Figura 3, a informação associada aos clientes significa o seguinte:

- o primeiro cliente, que pretende ir da cidade 1 para a cidade 4 o mais rapidamente possível, está disponível para partir no instante 0, e tem como restrição a não utilização do avião,
- o segundo cliente, que quer ir da cidade 2 para a 3 da forma mais barata possível, está disponível no instante 250, e não aceita rotas cujo custo total seja superior a 10000,

```
5
1 1 4 0 tempo 1 A1 aviao
2 2 3 250 custo 1 B2 10000
3 1 12 1550 custo 0
4 9 10 0 tempo 2 A2 500 B1 1440
5 5 9 1000 tempo 2 A3 200 B1 2000
```

Figura 3: Exemplo do ficheiro .cli.

- o terceiro cliente, que pretende ir da cidade 1 para a 12 da forma mais barata possível, está disponível para partir no instante 1550, e não tem qualquer restrição,
- o quarto cliente, que quer ir da cidade 9 para a 10 o mais rapidamente possível, está disponível para partir no instante 0, e não quer ligações cujo tempo de viagem seja superior a 500 nem rotas cujo tempo total seja superior a 1440, e finalmente
- o quinto cliente pretende ir da cidade 5 para a cidade 9 o mais rapidamente possível, pode partir no instante 1000, e não quer ligações cujo custo de viagem seja superior a 200 nem rotas cujo tempo total seja superior a 1000.

3.3 Formato de saída de dados

O resultado da execução do programa de AGENTE DE VIAGENS consiste em determinar para cada cliente qual a rota que melhor satisfaz a sua preferência, em tempo ou custo.

A solução deve ser colocada num ficheiro de saída, cujo nome deve ser o mesmo do ficheiro de entrada .cli mas **com extensão** .sol. Este ficheiro deve ser criado e aberto pelo programa. Por exemplo, se o ficheiro de entrada de clientes se chama `input231.cli`, o ficheiro de saída deve chamar-se `input231.sol`.

O ficheiro de saída deverá ter C linhas, uma por cada cliente, com a seguinte informação, separada por um espaço em branco:

- o número do cliente, a rota determinada para o cliente, o tempo total da rota e o custo total da mesma. O tempo total da rota é a diferença entre o instante de chegada do cliente à cidade final e o instante de disponibilidade de partida da cidade inicial (ver definição do ficheiro .cli). O custo total da rota é simplesmente a soma dos custos das diferentes ligações que compõem a rota.
- a especificação da rota deve seguir o seguinte formato: começa sempre pelo número da cidade de partida (número inteiro de 1 a N), e termina no número da cidade de chegada. Entre estes dois números deve-se colocar, sempre separados por um espaço em branco, o meio de transporte (uma das seguintes palavras: aviao, comboio, barco ou autocarro) e o número da cidade para as várias ligações que constituem a rota.

Por exemplo, se uma rota entre a cidade 33 e 18 compreende as seguintes ligações: comboio entre a cidade 33 e a cidade 67, aviao entre a cidade 67 e a cidade 101, e barco entre a cidade 101 e a cidade 18, então a linha correspondente do ficheiro de saída é a seguinte²:

²O número 12 inicial corresponde ao número do cliente.

3 O PROGRAMA AGENTE DE VIAGENS

```
12 33 comboio 67 aviao 101 barco 18 1267 8197
```

Neste exemplo, a rota em causa tem um tempo total de 1267 e um custo total de 8197.

- no caso particular de não existir qualquer rota entre a cidade de partida e a cidade de chegada e/ou que satisfaça as restrições definidas por um cliente, a linha correspondente a esse cliente no ficheiro de saída deve ter apenas -1.

Considere a rede apresentada na Figura 1. Neste caso, um exemplo de ficheiro de saída é o seguinte:

```
1 1 autocarro 4 90 15
2 2 aviao 6 barco 5 autocarro 1 comboio 4 autocarro 3 1020 47
3 -1
4 9 autocarro 10 5 2
5 5 autocarro 1 autocarro 4 comboio 8 comboio 9 853 94
```

Figura 4: Exemplo do ficheiro de saída para a rede apresentada na Figura 1 e os clientes apresentados na Figura 3.

Tabela 1: Datas importantes do Projecto

| Data | Documentos a Entregar |
|--|--|
| até 23 de Outubro de 2014, 5ª feira | Enunciado do projecto disponibilizado na página da disciplina. |
| até 31 de Outubro de 2014 (24h) | Inscrição dos grupos no sistema Fenix. |
| 9 de Dezembro de 2014, 3ª feira 10h 15h | 1ª Data de entrega do projecto: Submissão electrónica do projecto. Entrega do relatório do projecto em papel. |
| 10 de Dezembro de 2014, 4ª feira 10h 15h | 2ª Data de entrega do projecto: penalização de um (1) valor Submissão electrónica do projecto. Entrega do relatório do projecto em papel. |
| 11 de Dezembro de 2014, 5ª feira 10h 15h | 3ª Data de entrega do projecto: penalização de dois (2) valores Submissão electrónica do projecto. Entrega do relatório do projecto em papel. |
| | Submissões posteriores a 11 de Dezembro têm penalização de 20 valores. |
| 15 a 19 de Dezembro de 2014 | Discussão do trabalho (data combinada com cada grupo). |

4 Avaliação do Projecto

O projecto está dimensionado para ser feito por grupos de dois alunos, não se aceitando grupos de outra dimensão. Para os alunos que frequentam o laboratório, o grupo de projecto não tem de ser o mesmo do laboratório, mas é aconselhável que assim seja.

Quando os grupos de projecto estiverem constituídos, os alunos devem obrigatoriamente inscrever-se no sistema Fenix, no grupo de Projecto correspondente, que será criado oportunamente.

A avaliação do projecto envolve duas datas distintas: numa primeira data é entregue o relatório (em papel) e o código do projecto (electronicamente) e numa segunda data é feita a discussão e defesa do mesmo. No entanto, só terão que realizar a discussão os grupos em que se verifique uma diferença superior a três valores entre a classificação do laboratório e a classificação provisória do projecto e/ou para grupos em que a diferença de desempenho no laboratório entre os dois alunos seja também superior a três valores. As datas relevantes referentes aos vários passos da avaliação do projecto estão indicadas na Tabela 1.

Note-se que o projecto só é considerado entregue aquando da entrega do relatório em papel. As submissões electrónicas do código não são suficientes para concretizar a entrega. Um grupo que faça a submissão electrónica do código e a entrega do relatório em papel, por exemplo, na 1ª data de entrega pode fazer submissões nas datas seguintes, mas se fizer a entrega de um novo relatório em papel, será este, e as respectivas submissões, o considerado para avaliação, com a penalização indicada.

A avaliação do projecto basear-se-á no funcionamento do trabalho, qualidade do código, qualidade do relatório e, eventualmente, na discussão.

4.1 Funcionamento

A verificação do funcionamento do código a desenvolver no âmbito do projecto será exclusivamente efectuada nas máquinas do laboratório da disciplina, embora o desenvolvimento possa ser efectuada em qualquer plataforma ou sistema que os alunos escolham. Esta regra será estritamente seguida, não se aceitando quaisquer excepções. Por esta razão, é essencial que os alunos, independentemente do local e ambiente em que desenvolvam os seus trabalhos, os verifiquem no laboratório antes de os submeterem, de forma a evitar problemas de última hora. Uma vez que os laboratórios estão abertos e disponíveis para os alunos em largos períodos fora do horário das aulas, este facto não deverá causar qualquer tipo de problemas.

4.2 Código

Não deve ser entregue código em papel. Os alunos devem entregar por via electrónica o código do programa (ficheiros `.h` e `.c`) e uma `Makefile` para gerar o executável. Todos os ficheiros (`*.c`, `*.h`, `Makefile`, etc.) devem estar localizados na directoria raiz.

O código deve ser estruturado de forma lógica em vários ficheiros (`*.c` e `*.h`). As funções devem ter um cabeçalho curto mas explicativo e o código deve estar correctamente indentado e com comentários que facilitem a sua legibilidade.

4.3 Relatório

Os relatórios devem ser entregues na altura indicada na Tabela 1 e dirigidos ao docente do laboratório respectivo. Os grupos compostos por alunos que frequentam laboratórios distintos devem indicar um dos docentes desses laboratórios. Os alunos que não estão a frequentar o laboratório devem mencionar essa situação.

O relatório do projecto deverá contemplar os aspectos seguidamente indicados. A apresentação do mesmo deverá iniciar-se por aspectos de ordem geral, seguindo-se descrições mais detalhadas dos módulos e estruturas de dados utilizados. Sugere-se a seguinte estrutura para o relatório:

- Uma capa com os dados dos membros do grupo, incluindo nome, número e e-mail. Esta capa deverá seguir o formato indicado na página da disciplina (oportunamente será disponibilizado);
- Uma página com o índice das secções em que o relatório se divide;
- Uma descrição do problema que foi resolvido, com indicação clara das especificações do mesmo tal como foram entendidas;
- Um texto simples que indique como os alunos abordaram e resolveram o problema;
- Uma descrição completa da arquitectura do programa, incluído um fluxograma detalhado e um texto claro, mas sucinto, indicando a divisão lógica e funcional dos módulos desenvolvidos para a resolução do problema, indicando os respectivos objectivos, as funções utilizadas e as estruturas de dados de suporte;
- Uma descrição detalhada das estruturas de dados utilizadas e justificação das mesmas;
- Descrição dos algoritmos usados (por exemplo, na manipulação das estruturas de dados);
- Uma descrição dos subsistemas funcionais que existam e, para cada um
 - a descrição dos objectivos do subsistema (até 5 linhas);

- o nome do módulo onde estão definidas as estruturas de dados (ficheiros .h) a utilizar no subsistema;
- o nome do módulo C onde estão as funções do respectivo subsistema;
- listagem das funções a implementar no subsistema, indicando para cada uma a respectiva assinatura e os objectivos da função (descrição sumária, sem código);
- Uma análise dos requisitos computacionais do programa desenvolvido, tanto em termos da memória que utiliza como da complexidade computacional, com particular ênfase no custo das operações de processamento sobre as estruturas de dados;
- Uma análise crítica do funcionamento do programa e a avaliação do desempenho do projecto implementado;
- Pelo menos, um pequeno exemplo completo de aplicação, com descrição da utilização das estruturas de dados em cada passo.

Oportunamente será disponibilizado na página da disciplina um relatório modelo.

A nota final do projecto será calculada com base nas diferentes componentes de avaliação mas estará condicionada ao funcionamento do programa de acordo com os valores máximos indicados na Tabela 2.

Tabela 2: Grelha de Avaliação do Projecto

| Situação | Nota Máxima |
|---|-----------------------|
| Programa não compila, termina sistematicamente de formas ilegais com erros de execução, etc. | Trabalho não é aceite |
| Programa funciona de forma algo incompleta ou incorrecta, i.e., determina alguns valores mas apresenta resultados total ou parcialmente errados, não obtém a solução completa para o problema ou tem qualquer outra limitação considerada grave | 12 valores |
| Programa determina a solução do problema mas funciona de forma ineficiente (má gestão de memória ou computacionalmente ineficaz). | 16 valores |
| Programa funciona de forma eficiente e completa, com código correctamente comentado e estruturado e relatório bem escrito, claro e completo | 20 valores |

4.4 Discussão

Só terão de realizar discussão os grupos em que se verifique uma diferença entre a classificação provisória do projecto e a obtida em laboratório superior a três valores, ou grupos em que os dois elementos possuam uma diferença na classificação do laboratório superior a três valores. A discussão, a ser efectuada, será sempre feita em grupo embora possam ser colocadas questões distintas aos dois elementos de cada grupo e os mesmos possam vir a obter nota diferente no projecto, quando tal se justificar.

5 Código de Honestidade Académica

Espera-se que os alunos conheçam e respeitem o Código de Honestidade Académica que rege esta disciplina e que pode ser consultado na página da cadeira. Lembramos igualmente que a verificação de potenciais violações a este código é feita de forma automática com recurso a sofisticados métodos de comparação de código, que envolvem não apenas a comparação directa do código mas também da estrutura do mesmo. Esta verificação é feita com recurso ao software disponibilizado em

<http://moss.stanford.edu/>

.