

Métodos de Apoio à Decisão Multicritério

1 Objetivos

Com este projeto pretende-se que os alunos desenvolvam uma aplicação em linguagem Java onde apliquem um processo básico de desenvolvimento de aplicações informáticas, valorizando todas as fases do ciclo de desenvolvimento, desde a análise e conceção aos testes de validação. Pretende-se também que os alunos elaborem um relatório que descreva a aplicação concebida, o processo de desenvolvimento e que apresentem e critiquem os resultados obtidos.

Em particular, neste projeto pretende-se que o aluno estude Métodos de Apoio à Decisão Multicritério (MADMC) e implemente uma aplicação que permita resolver este tipo de problemas. Para desenvolver a aplicação, o aluno deverá considerar essencialmente quatro módulos: leitura, escrita e implementação dos métodos AHP (Analytic Hierarchy Process) (Saaty, 1990) e TOPSIS (Technique of Order Preference by Similarity to Ideal Solution) (Hwang & Yoon, 1981).

2 Plano de Trabalho

Para facilitar o estudo e desenvolvimento da aplicação, o trabalho de LAPR1 será dividido em duas partes/iterações. Na primeira parte o aluno terá que implementar uma versão simplificada do método AHP e os módulos necessários à leitura de dados e escrita de resultados. Na segunda parte o aluno terá que melhorar a implementação do método AHP, implementar o método TOPSIS e resolver um problema de tomada de decisão utilizando a aplicação desenvolvida.

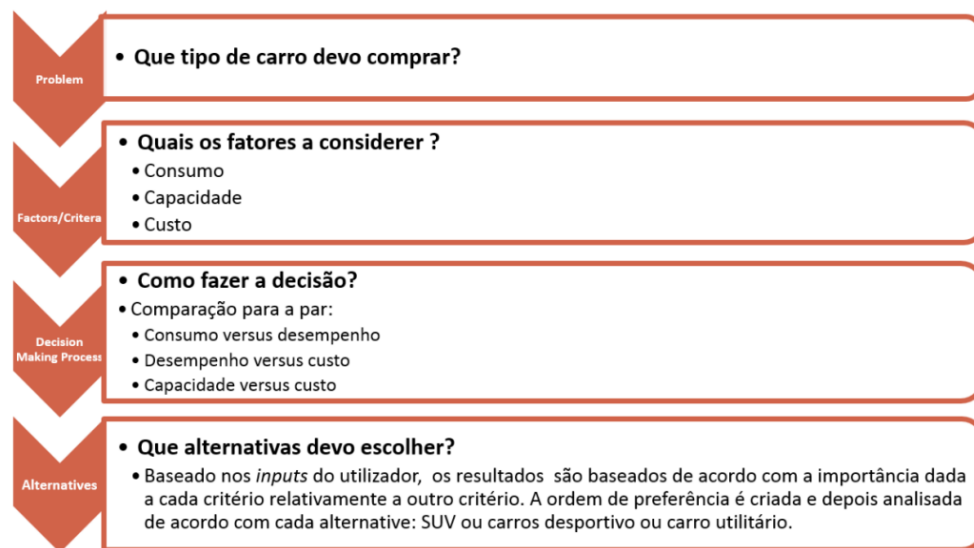


Figura 1: Problema de decisão multi-critério

3 Método de Apoio à Decisão Multicritério

Um método de decisão multicritério recorre a técnicas numéricas que auxiliam os decisores a escolher uma opção de um conjunto discreto de alternativas. Este processo é efetuado com base no cruzamento das alternativas com os critérios existentes.

Os métodos de apoio à decisão multicritério (MADMC) permitem ordenar um conjunto de alternativas numa situação de critérios conflituosos, procurando satisfazer as restrições, com objetivos conflitantes, ou seja, uma solução de compromisso. Assim sendo, os métodos MADMC podem ser utilizados para o apoio à negociação e/ou decisão em grupo. Na Figura 1 está representado um problema de tomada de decisão quando o problema é a compra de um tipo de carro. Neste processo de tomada de decisão são equacionados critérios, racional de decisão e alternativas de escolha (Roy, 2004).

3.1 AHP - Analytic Hierarchy Process

Um dos principais métodos desenvolvidos no ambiente das Decisões Multicritério Discretas é o Método de Análise Hierárquica (AHP-Analytic Hierarchy Process), criado em 1980 pelo professor Thomas L. Saaty (Saaty, 1990).

Este método permite o uso de critérios qualitativos bem como quantitativos no processo de avaliação de alternativas. A ideia principal deste método é dividir o problema de decisão em níveis hierárquicos, facilitando, assim, sua compreensão e avaliação.

O método AHP pode ser descrito através de um conjunto de passos:

- **Representação do Problema**

Construção da árvore hierárquica de decisão: definir o problema e estruturar em diagrama hierárquico. Esta fase consiste na decomposição do problema/decisão em uma hierarquia, composta, no mínimo, por um objetivo, critérios e alternativas. Na Figura 2 é apresentada a árvore hierárquica de decisão associada ao problema de escolha de um tipo de carro. Neste problema um decisor considera três critérios (Estilo, Confiabilidade e Consumo) e quatro alternativas (Car 1, Car 2, Car 3 e Car 4) na sua tomada de decisão.

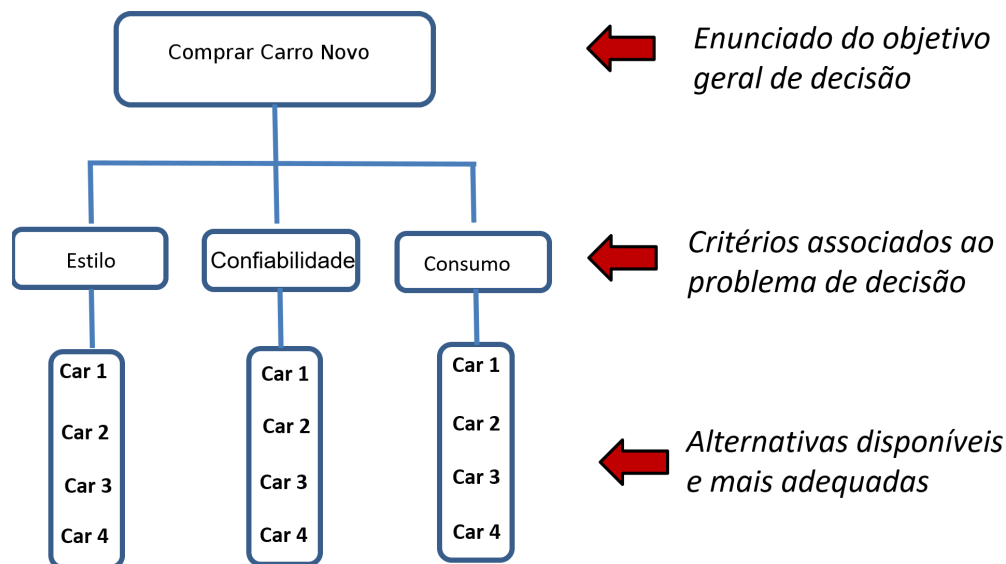


Figura 2: Exemplo de árvore hierárquica de decisão

- **Definição de critérios e alternativas**

Definição das matrizes de comparação das alternativas e critérios. Esta fase consiste em estabelecer prioridades entre os elementos para cada nível da hierarquia, por meio de uma matriz de comparação.

Considerando os três critérios (Estilo, Confiabilidade e Consumo) representados na estrutura hierárquica apresentada na Figura 2, um qualquer decisor, tendo por base a importância relativa de cada critério, pode definir a matriz de comparação de critérios apresentada na Tabela 1. De seguida, e para cada critério, um decisor define uma matriz de comparação de alternativas. Um exemplo é a matriz apresentada na Tabela 2 que compara quatro carros utilizando o critério estilo. Neste caso, o decisor também teria que construir matrizes de comparação de alternativas para os critérios Confiabilidade e Consumo.

Tabela 1: Matriz de comparação dos critérios Estilo, Confiabilidade e Consumo

	Estilo	Confiabilidade	Consumo
Estilo	1	1/2	3
Confiabilidade	2	1	4
Consumo	1/3	1/4	1

Tabela 2: Matriz de comparação de alternativas para o critério *Estilo*

	car 1	car 2	car 3	car 4
car 1	1	1/4	4	1/6
car 2	4	1	4	1/4
car 3	1/4	1/4	1	1/5
car 4	6	4	5	1

Tabela 3: Matriz de comparação de critérios normalizada juntamente com vetor de prioridades

	Estilo	Confiabilidade	Consumo	Vetor de Prioridades
Estilo	3/10	2/7	3/8	0.32
Confiabilidade	3/5	4/7	1/2	0.56
Consumo	1/10	1/7	1/8	0.12

- **Normalização e Vetor de Prioridades**

Para obter a prioridade relativa de cada critério é necessário: normalizar os valores da matriz de comparação e obter o vetor de prioridades. Na Tabela 3 é apresentada a matriz de comparação de critérios normalizada juntamente com o vetor de prioridades (vetor próprio associado ao

Tabela 4: Matriz de comparação de alternativas para critério *estilo* juntamente com vetor de prioridades

	car 1	car 2	car 3	car 4	Vetor de Prioridades
car 1	1	1/4	4	1/6	0.13
car 2	4	1	4	1/4	0.24
car 3	1/4	1/4	1	1/5	0.07
car 4	6	4	5	1	0.56

maior valor próprio).

Na Tabela 4 é apresentada a matriz de comparação de alternativas e o vetor de prioridades para o critério estilo.

• Análise de consistência da matriz

Esta fase consiste em calcular a Razão de Consistência (RC) para avaliar a consistência dos julgamentos representados nas matrizes de comparação. Para o cálculo do RC é necessário calcular o maior valor próprio da matriz de comparação.

$$RC = \frac{IC}{IR}$$

em que

$$IC = \frac{\lambda_{max} - n}{n - 1},$$

onde n é a ordem da matriz (quadrada) de comparação e IR é o valor do índice de consistência aleatório dado pela Tabela 5.

Tabela 5: Valores referência do índice de consistência aleatório

n	1	2	3	4	5	6	7	8	...
IR	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	...

• Cálculo da prioridade composta

Nesta fase são combinados os vetores de prioridades obtidos com o objetivo de obter a prioridade composta para as diferentes alternativas:

$$\underbrace{\begin{bmatrix} 0.13 & 0.38 & 0.30 \\ 0.24 & 0.29 & 0.24 \\ 0.07 & 0.07 & 0.21 \\ 0.56 & 0.26 & 0.25 \end{bmatrix}}_{\text{Matriz de Vetores de Prioridades}} \times \underbrace{\begin{bmatrix} 0.32 \\ 0.56 \\ 0.12 \end{bmatrix}}_{\text{Peso dos Critérios}} = \underbrace{\begin{bmatrix} 0.30 \\ 0.27 \\ 0.08 \\ \mathbf{0.35} \end{bmatrix}}_{\text{Prioridade Composta}} \quad (1)$$

- **Escolha da melhor alternativa**

No final do processo, a alternativa associada à prioridade composta de maior valor é considerada a melhor decisão segundo o método AHP. Neste caso a melhor alternativa é a compra do *car 4*.

Para o estudo aprofundado do método AHP os alunos deverão consultar a bibliografia da unidade curricular, os apontamentos disponíveis no Moodle e assistir às aulas da área da Matemática.

3.2 TOPSIS - Technique of Order Preference by Similarity to Ideal Solution

Tal como o método AHP, este método permite o uso de critérios qualitativos bem como quantitativos no processo de avaliação de alternativas. A ideia principal deste método é o cálculo da distância entre cada alternativa considerada e duas alternativas ideais. Utilizando o cálculo destas distâncias, o método TOPSIS seleciona a alternativa que está mais próxima da solução ideal e mais distante da solução ideal negativa (Hwang & Yoon, 1981; Triantaphyllou, Shu, S.N.Sanchez, & Ray, 1998).

O método TOPSIS pode ser descrito através de um conjunto de passos:

- **Definição de matriz de decisão e pesos de cada critério**

Considerando um conjunto de m alternativas e n critérios são definidos, pelo decisor, os pesos de cada critério e uma matriz de decisão. Na Tabela 6 são apresentados os pesos de cada critério (w_j , com $j \in 1, \dots, n$) e uma matriz de decisão definida para auxiliar na tomada de decisão de compra de um automóvel. Nesta matriz estão representadas quatro alternativas (Civic, Saturn, Ford e Mazda) e quatro critérios (Estilo, Confiabilidade, Consumo e Custo) em que cada coeficiente da matriz x_{ij} (normalmente um valor inteiro no intervalo de 0 a 10) representa a importância da alternativa i relativamente ao critério j .

Tabela 6: Peso de cada critério e matriz de decisão

Pesos	0.1	0.4	0.3	0.2
	Estilo	Confiabilidade	Consumo	Custo
Civic	7	9	9	8
Saturn	8	7	8	7
Ford	9	6	8	9
Mazda	6	7	8	6

- **Construção da matriz de decisão normalizada pesada**

Neste passo, primeiro, a matriz de decisão representada na Tabela 6 é normalizada. De seguida, cada coeficiente da matriz normalizada é multiplicado pelo respetivo peso (w_j), obtendo-se a matriz normalizada pesada apresentada na Tabela 7. Formalmente, podemos definir os coeficientes da matriz normalizada pesada com sendo $v_{ij} = w_j r_{ij}$, em que r_{ij} são os coeficientes da matriz de decisão normalizada.

- **Determinar solução ideal e solução ideal negativa**

Considerando a utilização de critérios benefício J , em que maior valor representa maior ganho, e critérios custo J' , em que o menor valor representa maior ganho, nesta fase determina-se a

Tabela 7: Matriz de decisão normalizada pesada

	Estilo	Confiabilidade	Consumo	Custo
Civic	0.046	0.244	0.162	0.106
Saturn	0.053	0.192	0.144	0.092
Ford	0.059	0.164	0.144	0.118
Mazda	0.040	0.192	0.144	0.080

solução ideal $A^* = \{v_1^*, \dots, v_n^*\}$, em que $v_j^* = \{\max_i(v_{ij}) \text{ se } j \in J \text{ ou } \min_i(v_{ij}) \text{ se } j \in J'\}$ e a solução ideal negativa $A' = \{v_1', \dots, v_n'\}$, em que $v_j' = \{\min_i(v_{ij}) \text{ se } j \in J \text{ ou } \max_i(v_{ij}) \text{ se } j \in J'\}$. No exemplo ilustrativo apresentado na Tabela 6 podemos observar que $J = \{\text{Estilo}, \text{Confiabilidade}\}$ e $J' = \{\text{Custo}\}$.

- **Calcular a distância entre cada alternativa e as soluções ideal e ideal negativa**

No caso em que a distância Euclidiana seja considerada, a distância entre cada solução alternativa e a solução ideal será obtida através da fórmula

$$S_i^* = \left[\sum_{j=1}^n (v_j^* - v_{ij})^2 \right]^{\frac{1}{2}}, \quad i = 1, \dots, m$$

e a distância à solução ideal negativa será obtida através da expressão

$$S_i' = \left[\sum_{j=1}^n (v_j' - v_{ij})^2 \right]^{\frac{1}{2}}, \quad i = 1, \dots, m.$$

- **Calcular a proximidade relativa à solução ideal**

Considerando as distâncias obtidas no passo anterior, para o cálculo da proximidade relativa de cada alternativa à solução ideal é utilizada a seguinte fórmula:

$$C_i^* = \frac{S_i'}{S_i^* + S_i'}, \quad i = 1, \dots, m.$$

- **Escolher a melhor alternativa**

Neste passo devemos ordenar o vetor C^* e escolher a alternativa com valor mais próximo do valor um. Considerando o exemplo apresentado na Tabela 7, cuja solução ideal é $A^* = \{0.059, 0.244, 0.162, 0.080\}$, cuja solução ideal negativa é $A' = \{0.040, 0.164, 0.144, 0.118\}$, e que a proximidade à solução ideal é $C^* = \{0.74, 0.41, 0.17, 0.45\}$, a alternativa escolhida será a compra do automóvel *Civic*.

Para o estudo aprofundado do método TOPSIS os alunos deverão consultar a bibliografia da unidade curricular, os apontamentos disponíveis no Moodle e assistir às aulas da área da Matemática.

4 Trabalho a Desenvolver

O trabalho a realizar no projeto de LAPR1 será desenvolvido em duas iterações.

4.1 Iteração 1 - Método AHP

O trabalho a realizar até ao dia 31 de Dezembro de 2016 consiste no:

- Estudo do método AHP.
- Desenvolvimento, em linguagem Java, de uma aplicação que auxilie o utilizador na tomada de decisão implementando o método AHP. Nesta primeira parte apenas devem implementar um versão simples do método AHP, não considerando casos especiais e implementando **simultaneamente o cálculo aproximado e exato dos valores e vetores próprios**. Esta versão simples considera apenas a resolução de **problemas com três critérios, quatro alternativas e matrizes de comparação consistentes**. O cálculo exato dos valores próprios deve ser realizado com o auxílio da biblioteca *la4j* - linear algebra for Java (<http://la4j.org/apidocs/>). O cálculo aproximado deve ser realizado utilizando o método apresentado nas aulas de Matemática. Os slides que apresentam o método, e que foram utilizados nas aulas, estão disponíveis no Moodle.
- A aplicação deverá permitir carregar a descrição de um problema (matrizes de comparação) e escrever a solução deste problema para a consola e ficheiro de texto (txt).
- O utilizador da aplicação deve ter a possibilidade de escolher entre resolver o problema utilizando um método de cálculo de valores e vetores próprios exato ou um método de cálculo aproximado.
- Elaborar um relatório em que descrevem: o método AHP, a implementação do método AHP e a metodologia de trabalho que utilizaram para desenvolver a aplicação. A descrição do método AHP deve ser acompanhada por um exemplo ilustrativo.

4.2 Iteração 2 - Método TOPSIS

Nesta segunda iteração a aplicação desenvolvida na primeira parte será revista com o objetivo de introduzir algumas melhorias, disponibilizar novas funcionalidades e implementar um novo método de apoio à decisão. Em concreto, o trabalho a realizar até ao dia 15 de Janeiro de 2017 consiste em:

- Estudar o método TOPSIS.
- Implementar do método de apoio à decisão TOPSIS. A implementação deve considerar a descrição sucinta apresentada na Secção 3.2 e a introdução a realizar nas aulas da área da Matemática.
- Implementar os métodos AHP e TOPSIS de forma a conseguir resolver problemas com um número arbitrário de alternativas e critérios.
- Melhorar a implementação do método AHP por forma a incluir a possibilidade de excluir critérios cujo peso seja inferior a um limiar definido pelo utilizador. Sempre que um ou mais critérios tenham peso inferior ao valor do limiar, os cálculos devem ser executados novamente. Os resultados obtidos utilizando o subconjunto de critérios deve ser apresentado no ficheiro de saída e consola, tal como especificado na Secção 4.3.1, devendo ser também apresentados, no ficheiro de texto e consola, os atributos eliminados. O limiar será um parâmetro definido na linha de comandos (ver Secção 6 deste documento).

Como exemplo, e considerando os pesos apresentados na Tabela 3, se o limiar fosse igual a 0.15, o critério *Consumo* seria removido e não seria considerado na resolução do problema.

- Melhoria do módulo de leitura de forma a incluir a capacidade de ler a descrição de um problema a ser resolvido pelo método TOPSIS.

- Incluir funcionalidade de validação dos dados de entrada. Além de outras melhorias que os alunos poderão introduzir, o módulo de leitura deverá rejeitar descrições de problemas cujo ficheiro de entrada contenha:
 - Linhas que não estejam completamente em branco;
 - Coeficientes das matrizes de comparação cujos valores sejam diferentes de um valor inteiro ou um número racional.

Nestes casos, em que os dados de entrada são inválidos, o módulo de leitura deve apresentar os erros na consola e num ficheiro de erros. Os erros identificados numa sessão devem ser registados num ficheiro cuja designação inclui a data e horário em que os erros foram detetados.

- Os critérios benefício e custo devem ser utilizados na resolução do método TOPSIS e devem ser identificados no ficheiro de entrada. Caso os critérios benefício e custo não estejam definidos no ficheiro de entrada, a aplicação deve prosseguir assumindo que todos os critérios são critérios benefício.
- Os pesos de cada um dos critérios utilizados na resolução do método TOPSIS devem ser especificados no ficheiro de entrada. Caso estes pesos não estejam definidos no ficheiro de entrada, a aplicação deve apresentar uma mensagem na consola e prosseguir a execução considerando uma distribuição uniforme para os pesos, i.e., $w_i = 1/n$, em que n é o número de critérios e i o índice do critério associado à coluna i da matriz de decisão.
- O módulo de escrita deve identificar o método utilizado na resolução. A identificação do método a utilizar será um parâmetro do comando de execução da aplicação e será definido pelo utilizador (ver Secção 6 deste documento).
- Resolução de um problema de tomada de decisão utilizando a aplicação desenvolvida e os métodos AHP e TOPSIS. Deve ser efetuada uma análise comparativa aos resultados obtidos utilizando cada um dos métodos. O enunciado deste problema será enviado por e-mail, para todos os elementos de cada um dos grupos, no dia 5 de Janeiro de 2017.
- Elaborar um relatório final tendo por base o relatório parcial escrito durante a primeira parte do trabalho. Além de documentar o estudo e a implementação dos requisitos apresentados nesta segunda parte do trabalho, o relatório final deve incluir um capítulo *Estudo de um Problema de Tomada de decisão* onde será descrita a resolução do problema de tomada de decisão a ser enviado no dia 5 de Janeiro de 2017.

A escrita deste relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de competências), e não deve ultrapassar as 28 páginas.

- Elaborar um conjunto de slides com o objetivo de apresentar o trabalho desenvolvido na unidade curricular de LAPR1. Nestes slides os alunos têm que apresentar, não estando limitados a, o negócio (Métodos de Apoio à Tomada de Decisão), a arquitetura da aplicação e a implementação da aplicação. Estes slides (no máximo 8 slides) serão utilizados pelos alunos para apresentar o trabalho desenvolvido aquando do momento de avaliação a ter lugar entre os dias 16 e 18 de Janeiro. A apresentação do trabalho com auxílio dos slides não deve ultrapassar a duração máxima de 10 minutos.

4.3 Formato dos Dados de Entrada e Saída

Nesta secção é descrito o formato de entrada e saída para os métodos AHP e TOPSIS.

4.3.1 Método AHP

O ficheiro que contém os dados de entrada (as matrizes de comparação de critérios e alternativas) deve incluir em primeiro lugar a matriz de comparação de critérios e de seguida cada uma das matrizes que, para cada critério, define a comparação das alternativas. Neste ficheiro a representação de qualquer das matrizes deve iniciar com uma linha de cabeçalho seguida da matriz dos coeficientes. A linha de cabeçalho descreve o problema e as linhas/colunas da matriz. A descrição de qualquer dos campos presentes na linha de cabeçalho não deve incluir espaços em branco. Os campos da linha cabeçalho aparecem sempre separados por dois espaços em branco enquanto os coeficientes das restantes linhas podem aparecer separados por um número arbitrário de espaços em branco.

Um exemplo de um ficheiro de entrada:

```
mc_criteios  Estilo  Confiabilidade  Consumo
1      1/2    3
2      1      4
1/3 1/4      1
```

```
mcp_estilo  car1  car2  car3  car4
1      1/4  4      1/6
4      1    4      1/4
1/4 1/4  1      1/5
6      4    5      1
```

```
mcp_confiabilidade  car1  car2  car3  car4
1      2      5      1
1/2  1      4      1/4
1/5 1/3  1      1/4
1      1/2  4      1
```

```
mcp_consumo  car1  car2  car3  car4
1      1/4  4      1/6
4      1    4      1/4
1/4 1/4  1      1/5
6      4    5      1
```

Os dados de saída para o ficheiro de texto são as matrizes de comparação, as matrizes de comparação de entrada e as matrizes normalizadas, o IR, o RC, o maior valor próprio, de cada matriz, as matrizes de comparação normalizadas com as respetivas prioridades relativas, o vetor de prioridade composta e a alternativa escolhida. As matrizes de entrada, vetor de prioridade composta para as alternativas e a alternativa escolhida devem ser apresentados simultaneamente na consola e no ficheiro de texto.

4.3.2 Método TOPSIS

O ficheiro que contém os dados de entrada (identificação de critérios benefício e custo, o vetor de pesos e a matriz de decisão) deve incluir a identificação de critérios benefício e custo o vetor de pesos

e a matriz de decisão. Este ficheiro deve iniciar com a identificação dos critérios benefício e custo. Depois, o ficheiro deve conter o cabeçalho do vetor de pesos e os respetivos pesos/coeficientes. Por último, deve ser representada a matriz de decisão, primeiro as três linhas do cabeçalho, identificando critérios e alternativas, depois um conjunto de coeficientes.

A descrição de qualquer dos campos presentes na linha de cabeçalho não deve incluir espaços em branco. Os campos das linhas de cabeçalho aparecem sempre separados por dois espaços em branco enquanto os coeficientes das restantes linhas podem aparecer separados por um número arbitrário de espaços em branco.

Um exemplo de um ficheiro de entrada:

```
crt_beneficio  Estilo  Confiabilidade  Consumo
crt_custo      Custo

vec_pesos      Estilo  Confiabilidade  Consumo  Custo
0.1 0.4 0.3 0.2

md crt_alt
crt Estilo  Confiabilidade  Consumo  Custo
alt Civic  Saturn  Ford  Mazda
7 9 9 8
8 7 8 7
9 6 8 9
6 7 8 6
```

Os dados de saída para o ficheiro de texto são o vetor dos pesos de entrada, a matriz de decisão, a matriz normalizada, a matriz normalizada pesada, a solução ideal, a solução ideal negativa, a distância entre cada alternativa e as soluções ideal e ideal negativa, o vetor com as proximidades relativa à solução ideal e a alternativa escolhida. Os dados de entrada, a matriz normalizada pesada, o vetor com as proximidades relativa à solução ideal e a alternativa escolhida devem ser apresentados simultaneamente na consola e no ficheiro de texto.

5 Método de Trabalho

- Todos os alunos devem utilizar a metodologia de trabalho definida no eduScrum (Delhij & Solingen, 2013). Cada um dos grupos deve escolher um Scrum Master e este deve ser responsável por gerir a execução de tarefas. Para atingir este objetivo, o grupo deve criar um documento (4 folhas A4) onde são: registadas as tarefas do projeto; a atribuição de tarefas; o estado de cada tarefa e as tarefas terminadas. Devem utilizar uma folha A4 para cada tipo de registo. Em alternativa, os alunos podem utilizar um ficheiro Excel com 4 folhas.
- A aplicação será desenvolvida utilizando o sistema de controle de versões Git e o Bitbucket (<https://bitbucket.org>). Todos os alunos terão que criar uma conta no Bitbucket com o endereço de email do ISEP (i.e. 1XXXXXX@isep.ipp.pt) e cada grupo terá que criar um repositório. A designação do repositório deve seguir o formato do exemplo "LAPR1_TurmaDAB_Grupo01". O repositório deve ser partilhado com todos os docentes que lecionam a turma onde o grupo está inserido.

- O grupo deve criar uma pasta no OneDrive onde guarda todo o material desenvolvido para a realização do projeto (ex: folha de Excel com tarefas e relatório) exceto o código Java existente no Bitbucket. A designação da pasta deve seguir o formato do exemplo "LAPR1_TurmaDAB_Grupo01". A pasta será partilhada com todos os docentes que lecionam a turma onde o grupo está inserido.
- Na segunda parte do trabalho cada grupo deve escolher novamente um elemento para desempenhar as funções de SCRUM Master. **O Scrum master que coordena a segunda parte do Trabalho será, obrigatoriamente, um elemento que não o Scrum Master que exerceu funções na primeira parte do trabalho.**

6 Processo de Desenvolvimento de Software

- A aplicação deve ser estruturada e organizada em módulos. Será valorizada uma correta decomposição modular e o reaproveitamento de módulos.
- O trabalho deverá ser desenvolvido em linguagem Java e deverá resultar num ÚNICO projeto NetBeans.
- Na implementação do método AHP, o cálculo exato dos valores e vetores próprios será utilizada a biblioteca la4j - linear algebra for Java (<http://la4j.org>).
- A aplicação deverá ser chamada da linha de comandos utilizando o comando:
`java -jar nome_programa.jar [Options] nome_ficheiro_input.txt nome_ficheiro_output.txt`, em que *Options* representa um conjunto de parâmetros:
 -M 1 #Execução do Método AHP
 -M 2 #Execução do Método TOPSIS
 -L 0.15 #Limiar a utilizar no método AHP para descartar atributos irrelevantes

```

Bool test_find_max_eigenvalue(matrix, expectedMaxEigenValue)
{

maxEigenvalue = find_max_eigenvalue(matrix);

if(expectedMaxEigenValue==maxEigenvalue)
    return True;
else
    return False;

}

```

Algoritmo 1: Exemplo de métodos de teste unitários

- Todos os métodos desenvolvidos terão, obrigatoriamente, de estar associados a testes unitários. Por exemplo, se o aluno criar o método *find_max_eigenvalue(matrix)* para determinar o maior valor próprio de uma matriz de comparação, também deve criar o método *test_find_max_eigenvalue(matrix, expectedMaxEigenValue)* (ver Algoritmo 1), em que *expectedMaxEigenValue* é o valor do maior valor próprio conhecido da matriz. Estes testes são extremamente

úteis para determinar se os métodos estão de acordo com a sua especificação e se a edição destes não alterou a funcionalidade.

7 Submissão do Trabalho

Datas e entregas de trabalho a efetuar através do Moodle:

- Dia 15 de Janeiro de 2017 (até às 20:00h)
 - O projeto, incluindo toda a estrutura de diretorias e ficheiros do projeto (incluindo o executável), num único ficheiro comprimido (ZIP).
 - Relatório em formato pdf e não ultrapassando as 28 páginas. A escrita do relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de competências).
 - Apresentação em formato pdf não ultrapassando os 8 slides.

Nota: Os ficheiros deverão identificar, obrigatoriamente, a designação do grupo e a turma a que os alunos pertencem (Exemplo: "LAPR1_TurmaDAB_Grupo01_projeto.ZIP", "LAPR1_TurmaDAB_Grupo01_relatorio.PDF" e "LAPR1_TurmaDAB_Grupo01_apresentacao.PDF").

Referências

- Delhij, A., & Solingen, R. (2013). *The eduscrum guide: The rules of the game*. (Disponível em http://eduscrum.nl/file/CKFiles/The_eduScrum_Guide_EN_December_2013_1.0.pdf)
- Hwang, C., & Yoon, K. (1981). *Multiple attribute decision making: methods and applications : a state-of-the-art survey*. Springer-Verlag. Retrieved from <https://books.google.de/books?id=X-wYAQAIAAJ>
- Roy, R. (Ed.). (2004). *The analytic hierarchy process*. London: Springer London.
- Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1), 9–26.
- Triantaphyllou, E., Shu, B., S.N.Sanchez, & Ray, T. (1998). Multi-criteria decision making: an operations research approach. *Encyclopedia of electrical and electronics engineering*, 15(1998), 175–186.