

# Enunciado do Projeto

---

## 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 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 apresente e critique os resultados obtidos.

No final, devem fazer uma apresentação, em PowerPoint, do projeto desenvolvido.

Toda a gestão documental e respetivo trabalho colaborativo deverá estar suportado na aplicação OneDrive, do Office 365 (<http://login.microsoftonline.com/>), usando as credenciais de acesso ao portal do ISEP.

## Enunciado

A aplicação pedida permite a análise de **redes sociais** (LinkedIn, Facebook, Twitter,...).

Uma rede social pode ser vista como uma estrutura formada por pessoas (os nós) e o relacionamento entre as pessoas (as ligações entre os nós). Esta estrutura denomina-se grafo e pode ser representada como na Figura 1 e na Figura 2-a.

A análise das redes sociais pretende analisar e detetar padrões de relacionamento entre pessoas numa rede.

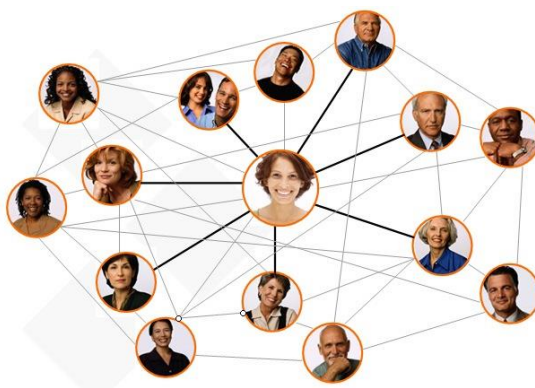


Figura 1: Representação simplificada da rede social (LinkedIn)

Uma forma de representar um grafo é a matriz de adjacências (ver Figura 2-b). Trata-se de uma matriz quadrada, com o número de linhas (colunas) igual ao número de nós do grafo. Se houver uma ligação entre quaisquer dois nós surge um 1 na posição correspondente, senão surge um 0.

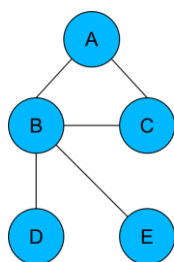


Figura 2: a) Grafo

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	1
C	1	1	0	0	0
D	0	1	0	0	0
E	0	1	0	0	0

b) Matriz de adjacências

A representação por matrizes (matriz de adjacências) permite a utilização das ferramentas da álgebra linear para analisar uma rede social.

**P1.** Dada a matriz de adjacências num ficheiro HTML, desenhe o grafo correspondente. Para tal, apenas podem ser utilizados métodos estáticos da classe **Janela** do projeto base disponibilizado no Moodle.

### Medidas de centralidade numa rede

Numa rede, os nós mais centrais são aqueles a partir dos quais podemos atingir qualquer outro nó mais facilmente. Existem diferentes medidas de centralidade que avaliam a importância dos nós numa rede de acordo com as suas ligações.

#### Centralidade de grau (*degree centrality*)

O grau de cada nó é dado pelo número de ligações desse nó, ou número de contactos de uma pessoa,  $d(i)$ . É calculado somando os elementos da linha correspondente ao nó, na matriz de adjacências (de grau  $n$ ). Quanto mais ligações, maior é a centralidade do nó.

**P2.** A aplicação deve determinar a pessoa com maior grau de conectividade normalizado ( $Cd$ ) e destacá-la graficamente (valor entre  $[0,1]$ ).

$$Cd(i) = \frac{d(i)}{n-1}$$

Nota: quanto maior o grau, mais influenciável é o nó na rede.

#### Centralidade de proximidade (*closeness*)

Esta medida de centralidade diz-nos que um nó é tanto mais importante quanto mais próximo está de todos os outros (valor entre  $[0,1]$ ).

$$Cc(i) = \frac{n-1}{\sum_{j=1}^n d(i,j)}$$

Sendo  $d(i,j)$  = distância entre  $i$  e  $j$  (ou seja, o menor número de ligações que temos de percorrer para ir de  $i$  até  $j$ ).

**P3.** A aplicação deve calcular as matrizes  $A^2, \dots, A^n$ , sendo  $A$  a matriz de adjacências e  $n$  a dimensão da matriz.

**P4.** Deve também calcular a matriz das distâncias mais curtas entre cada 2 nós.

Nota: Para calcular o número de caminhos de comprimento  $n$  deve calcular-se  $A^n$ , sendo  $A$  a matriz de adjacências.

**P5.** Deve ainda calcular o valor de centralidade de proximidade ( $C_c$ ) para todos os nós e encontrar o nó mais central (mais próximo de 1), destacando-o graficamente.

### Centralidade de intermediação (*betweenness*)

A medida de centralidade de intermediação de um nó  $k$  num grafo  $G$  com  $n$  nós é dada pela expressão

$$Cb(k) = \sum_{\substack{1 \leq i < j \leq n \\ i, j \neq k}} \frac{b_{ij}(k)}{b_{ij}}$$

Este valor, para cada nó, obtém-se somando o número de caminhos mais curtos entre  $i$  e  $j$  que passam pelo nó  $k$ ,  $b_{ij}(k)$ , a dividir por  $b_{ij}$ , número de caminhos mais curtos entre os nós  $i$  e  $j$ . Esta medida diz-nos que um nó é importante se faz parte de muitos caminhos.

**P6.** A aplicação deve calcular a matriz com o número de caminhos mais curtos entre os nós.

**P7.** Deve também calcular a centralidade de intermediação para cada nó ( $C_b$ ). Deve ainda destacar graficamente o nó mais central (o maior).

### Centralidade do vetor próprio

As medidas de centralidade espectrais procuram obter propriedades das redes a partir das propriedades dos valores próprios e dos vetores próprios da matriz de adjacências. A centralidade de vetor próprio atribui relevância a um nó, em função da sua relação com os seus vizinhos. Ou seja, se um nó  $k$  está ligado somente a outro nó  $i$  (tendo assim uma baixa centralidade de grau) mas os vizinhos de  $i$  forem importantes, o nó  $k$  também será importante, tendo uma elevada centralidade de vetor próprio.

**P8.** A aplicação deve calcular o nó mais central (ou seja, o nó correspondente à componente mais alta do vetor próprio associado ao maior valor próprio e com todas as componentes positivas) e assinala-lo graficamente.

Para tal, deve:

- Determinar os valores próprios, chamando o método **evaluate** da classe **EigenValues** do projeto base disponibilizado no Moodle.

`double vp[] = EigenValues.evaluate(A)`, sendo  $A$  a matriz de adjacências

- Calcular um vetor próprio,  $v$ , com todas as componentes positivas, correspondente ao valor próprio mais elevado  $\lambda$  (valor próprio principal). Para calcular o vetor próprio, deve resolver o sistema  $Av = \lambda v$  usando o método da condensação.

## Dados

Os dados de entrada da aplicação devem ser lidos a partir de um ficheiro HTML. Também devem ser guardados, num ficheiro HTML, os cálculos intermédios.

### Entrada de dados

Ler os dados a partir de um ficheiro em formato HTML, onde consta a matriz de adjacências com as ligações entre as pessoas na rede (ver Figura 3). Esta matriz pode ter qualquer dimensão. A entrada de dados deve ser visualizada na janela da aplicação (método adicionarDados da classe Janela).

---

### Matriz de adjacências

	Miguel	Carolina	Maria	Alice	Joana
Miguel	---	1	1	0	0
Carolina	1	---	0	1	0
Maria	1	0	---	1	1
Alice	0	1	1	---	0
Joana	0	0	1	0	---

---

Figura 3: Ficheiro HTML com os dados de entrada

### Resultados

Apresentar, num ficheiro em formato HTML os diferentes passos intermédios de cálculo de vetores e de matrizes. Estes resultados devem ser visualizados na janela da aplicação (método adicionarResultados da classe Janela).

## Notas gerais

- O LinkedIn é uma rede social para contactos profissionais. Para uma melhor compreensão do objetivo deste trabalho, sugere-se que se registe no LinkedIn (caso ainda não esteja) e que adira ao grupo DEI-ISEP.
- O trabalho deverá ser desenvolvido num ÚNICO projeto NetBeans. No Moodle encontra-se um projeto base para desenvolvimento da aplicação. Este projeto disponibiliza as classes Janela e EigenValues que pertencem à package lapr1.
- A aplicação deverá ser chamada da linha de comandos (java -jar nome\_programa.jar nome\_fich\_input.html nome\_fich\_output.html).
- Os ficheiros em HTML utilizados pela aplicação, deverão ser armazenados na pasta principal do projeto NetBeans.
- A aplicação deve ser estruturada e organizada em módulos.

## LAPR1 – 2014/15

- Será valorizada uma correta decomposição modular e o reaproveitamento de módulos.
- Deve ser criada a classe **CodificadorHTML**, contendo um conjunto de métodos estáticos que retornem strings com tabelas e texto formatado em HTML (incluindo caracteres acentuados, cabeçalhos, etc.).
- Deve também ser implementada a classe **DescodificadorHTML**, dotada de um conjunto de métodos que permitam descodificar tags HTML, para extração dos dados de entrada.
- É proibido usar classes já implementadas em Java com métodos que resolvam alguma das alíneas propostas.

### Prazos e entregas

- Na aula de orientação do grupo da semana de **15 a 19 de Dezembro** mostrar ao Orientador:
  - Esboço/mini-especificação da aplicação (máximo de 4 páginas), com uma breve descrição das classes da aplicação e dos respetivos métodos.
- Entregas (no Moodle) :
  - dia **15 de Janeiro** (até às **23:30h**)
    - o projeto completo, incluindo toda a estrutura de diretorias e ficheiros do projeto (incluindo o executável), num único ficheiro comprimido (ZIP). Este código será usado na avaliação do projeto, pelo que eventuais problemas na compilação da aplicação na apresentação serão sempre imputados ao grupo.
    - o relatório
  - dia **16 de Janeiro** (até às **23:30h**)
    - a apresentação do trabalho.

Nota: Os ficheiros deverão ter **obrigatoriamente** a designação do grupo (Ex: "DAB13\_projeto.ZIP", "DAB13\_relatorio.PDF" e "DAB13\_apresentacao.PDF").

- As apresentações do trabalho decorrerão nos dias **20 e 21 de Janeiro**.