

Universidade do Minho
Departamento de Informática

Mestrado Integrado em Engenharia Informática

Inteligência Ambiente: Tecnologias e Aplicações



Questão-Aula Nº2 - Conhecimento em Redes Sociais

a86617 Gonçalo Nogueira

Braga
Janeiro, 2021

Conteúdo

1	Introdução	3
2	Objetivos	4
3	Tratamento de Dados	5
4	Análise do grafo	7
4.1	Distribuição de grau	7
4.2	Diâmetro e distância	9
4.3	Medidas de centralidade	12
4.3.1	Grau	12
4.3.2	Eigenvector	13
4.3.3	Closeness	14
4.3.4	Betweenness	15
5	Grafo	16
6	Conclusão	17

Lista de Figuras

1	Leitura do ficheiro e criação do grafo	5
2	Grafo inicial	5
3	Grafo após remoção dos nodos	6
4	Método utilizado para a criação do histograma da distribuição de grau	7
5	Histograma da distribuição de grau	8
6	Diâmetro do grafo	9
7	Cálculo das distâncias	10
8	Exemplo das distâncias de um nodo para todos os outros	10
9	Eccentricity de todos os nodos e exemplo	11
10	Método para o cálculo da eccentricity	12
11	Output produzido	12
12	Método para cálculo de Eigenvector	13
13	Output produzido	13
14	Método para cálculo de Closeness	14
15	Output produzido	14
16	Método para cálculo de Betweenness	15
17	Output produzido	15
18	Grafo	16

1 Introdução

No âmbito da unidade curricular de Inteligência Ambiente: Tecnologias e Aplicações foi proposto aos alunos a conceção, manipulação e aplicação de conhecimento em **Redes Sociais** na linguagem *Python* com auxílio das bibliotecas *Networkx* e *Matplotlib* para observações dos grafos mais facilmente.

2 Objetivos

Pretende-se com a realização deste trabalho:

- Criar um grafo com as informações contidas em *listas-arestas.txt*
- Análise da distribuição de grau da rede
- Análise do diâmetro e distância de todos os nós da rede
- Análise de medidas de centralidade da rede

3 Tratamento de Dados

Para a criação do grafo foi utilizado o método `nx.Graph()` e para a adição de nodos e arestas foi feito o parse das informações contidas no ficheiro `listas-arestas.txt` e mediante cada linha do ficheiro foi-se adicionando uma nova aresta com o método `add.edge()` contendo os dois nodos bem como o seu peso.

```
in_file=csv.reader(open('listas_arestas.txt','r'))

g=nx.Graph()
listaux=[]
for line in in_file:
    g.add_edge(line[0],line[1],weight=int(line[2]),conf=int(line[3]))
    listaux.append(line[0])
```

Figura 1: Leitura do ficheiro e criação do grafo

Utilizando a biblioteca `pyplot` observamos a seguinte imagem que representa a rede criada.

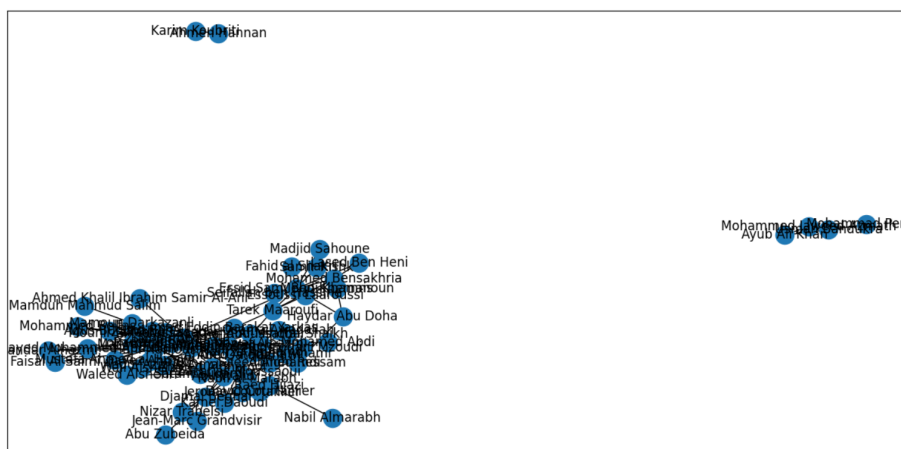


Figura 2: Grafo inicial

Depois da análise da figura 2 decidi retirar do ficheiro os nodos que não estavam conectados e que estavam a tornar o grafo não completo tornando excessivamente complicado os cálculos das medidas de análise.

Sendo assim os seguintes nodos foram eliminados:

- Usman Bandukra
- Mohammad Pervez
- Ahmen Hannan
- Karim Koubriti
- Ayub Ali Khan
- Mohammed Jaweed Azmath

A figura 3 representa o novo grafo após a remoção dos nodos não conectados.

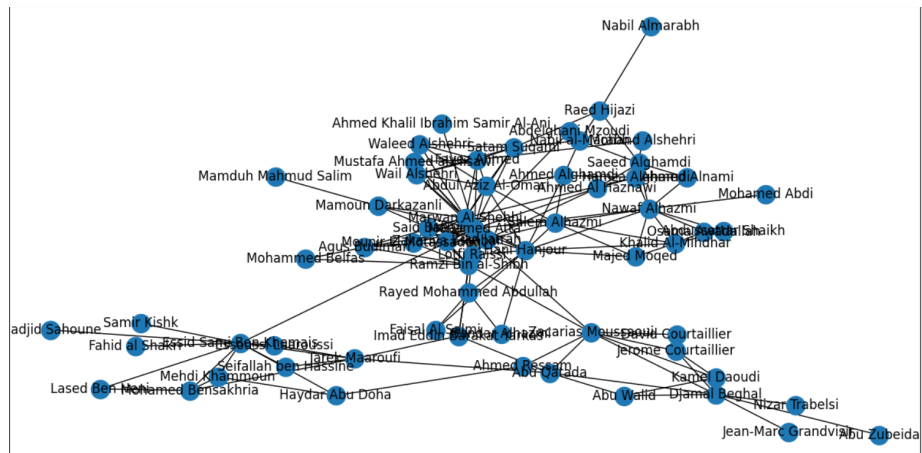


Figura 3: Grafo após remoção dos nodos

4 Análise do grafo

4.1 Distribuição de grau

No estudo de Grafos e Redes Complexas, o grau de um nó numa rede é o número de conexões que esse nó tem com outros nós, e a Distribuição de Graus é a distribuição de probabilidade dos graus dos nós de toda a rede.

Sendo assim usando o método `degree()` é possível obter o grau de cada nodo e depois ordenando todos os nodos com os seus respectivos graus e com o auxílio da livreria de desenho é possível criar um histograma com a distribuição de grau apresentado na figura 5.

```
# # Distribuição de grau
degree_sequence = sorted([d for n, d in g.degree()], reverse=True) # degree sequence
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())
#
fig, ax = plt.subplots()
plt.bar(deg, cnt, width=0.80, color="b")
#
plt.title("Degree Histogram")
plt.ylabel("Count")
plt.xlabel("Degree")
ax.set_xticks([d + 0.4 for d in deg])
ax.set_xticklabels(deg)
#
# # draw graph in inset
plt.axes([0.4, 0.4, 0.5, 0.5])
Gcc = g.subgraph(sorted(nx.connected_components(g), key=len, reverse=True)[0])
plt.axis("off")
plt.show()
```

Figura 4: Método utilizado para a criação do histograma da distribuição de grau

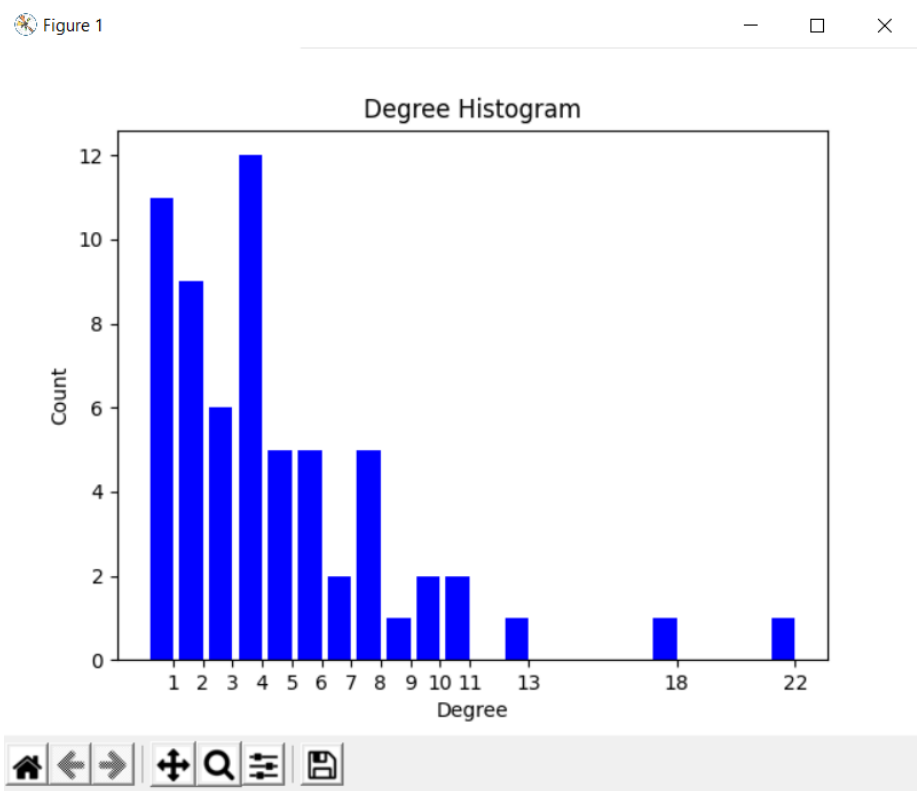
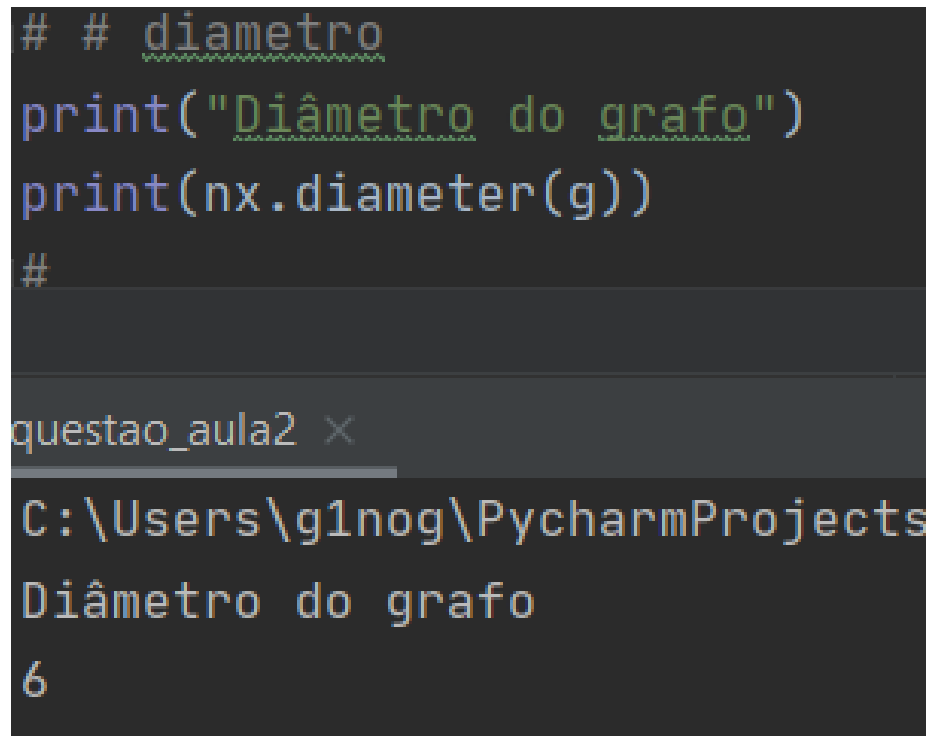


Figura 5: Histograma da distribuição de grau

4.2 Diâmetro e distância

O diâmetro é a máxima *eccentricity* ou seja o maior caminho entre dois nodos entre todos os nodos do grafo e é obtido diretamente com o método **diameter()**.



```
# # diametro
print("Diâmetro do grafo")
print(nx.diameter(g))
#
```

questao_aula2 ×

C:\Users\g1nog\PycharmProjects

Diâmetro do grafo

6

Figura 6: Diâmetro do grafo

Relativamente á distância utilizando o método all-pairs-dijkstra-path-length é possível calcular para todos os nós do grafo a distância para todos os outros nós da rede, sendo o resultado desta invocação do método um dicionário com todos esses valores, o que permite depois imprimir todas essas distâncias.

Na figura 7 está representado um exemplo para o nodo Hani Hanjour com as distâncias para todos os outros nodos.

```
# # medidas de distâncias
#
# #distância (menor distância e o seu comprimento)
print(" Comprimento do caminho mais curto entre todos os nodos")
list1 = list(nx.all_pairs_dijkstra_path_length(g))
i = 0
while i < len(list1):
    print(list1[i])
    i += 1
```

Figura 7: Cálculo das distâncias

```
Comprimento do caminho mais curto entre todos os nodos
('Hani Hanjour', {'Hani Hanjour': 0, 'Majed Moqed': 1, 'Nawaf Alhazmi': 1, 'Khalid Al-Mihdhar': 1,
'Bandar Alhazmi': 1, 'Rayed Mohammed Abdullah': 1, 'Salem Alhazmi': 1, 'Abdul Aziz Al-Omari': 1,
'Ahmed Alghamdi': 1, 'Faisal Al Salmi': 1, 'Mohamed Atta': 2, 'Waleed Alshehri': 2,
'Marwan Al-Shehhi': 2, 'Wail Alshehri': 2, 'Satam Suqami': 2, 'Fayez Ahmed': 2,
'Hamza Alghamdi': 2, 'Ziad Jarrah': 3, 'Essid Sami Ben Khemais': 3, 'Ahmed Al Haznawi': 3,
'Zacarias Moussaoui': 3, 'Mustafa Ahmed al-Hisawi': 3, 'Ahmed Khalil Ibrahim Samir Al-Ani': 3,
'Agus Budiman': 3, 'Abdelghani Mzoudi': 3, 'Imad Eddin Barakat Yarkas': 3, 'Mohand Alshehri': 3,
'Mohamed Abdi': 4, 'Nabil al-Marabh': 4, 'Haydar Abu Doha': 4, 'Mohamed Bensakhria': 4,
'Lased Ben Heni': 4, 'Madjid Sahoune': 4, 'Djamal Beghal': 4, 'Kamel Daoudi': 4,
'Ramzi Bin al-Shibh': 4, 'Lotfi Raissi': 5, 'Mamoun Darkazanli': 5, 'Mounir El Motassadeq ': 5,
'Raed Hijazi': 5, 'Mehdi Khammoun': 5, 'Ahmed Ressay': 5, 'Tarek Maaroufi': 5,
'Jerome Courtaillier': 5, 'Abu Zubeida': 5, 'Abu Walid': 5, 'Mohammed Belfas': 5, 'Ahmed Alnami': 6,
'Saeed Alghamdi': 6, 'Abdussattar Shaikh': 6, 'Osama Awadallah': 6, 'Essoussi Laaroussi': 6,
'Samir Kishk': 6, 'Fahid al Shakri': 6, 'Seifallah ben Hassine': 6, 'Abu Qatada': 6,
'Said Bahaji': 6, 'Nabil Almarabh': 6, 'Zakariya Essabar': 7, 'Jean-Marc Grandvisir': 7,
'David Courtaillier': 8, 'Mamduh Mahmud Salim': 8, 'Nizar Trabelsi': 9})
```

Figura 8: Exemplo das distâncias de um nodo para todos os outros

Relativamente á eccentricity que é a maior distância entre dois nodos de todos os nós constituintes da rede foi feita uma coleção de todos os nodos existentes na rede e para cada um foi invocado o método eccentricity.

Na figura 8 está contido o código bem como um excerto do output produzido.

```
# eccentricity (a maior distância entre um nodo e todos os outros)
print("Maior distancia entre cada nodo e todos os outros")
index = 0
while index < len(listaux):
    print(listaux[index])
    print(nx.eccentricity(g,listaux[index]))
    index += 1
```



```
questao_aula2 x
Hani Hanjour
4
Majed Moqed
5
Nawaf Alhazmi
4
Salem Alhazmi
5
Khalid Al-Mihdhar
5
Mohamed Atta
3
```

Figura 9: Eccentricity de todos os nodos e exemplo

4.3 Medidas de centralidade

4.3.1 Grau

O número de conexões de um grau para todos os outros.

```
print("Centralidade do grau")
print(nx.degree_centrality(g))
```

Figura 10: Método para o cálculo da eccentricity

```
Centralidade do grau
{'Hani Hanjour': 0.2096774193548387, 'Majed Moqed': 0.06451612903225806,
'Nawaf Alhazmi': 0.1774193548387097, 'Khalid Al-Mihdhar': 0.0967741935483871,
'Lotfi Raissi': 0.08064516129032258, 'Bandar Alhazmi': 0.03225806451612903,
'Rayed Mohammed Abdullah': 0.06451612903225806, 'Salem Alhazmi': 0.12903225806451613,
'Hamza Alghamdi': 0.11290322580645161, 'Ahmed Alnami': 0.04838709677419355,
'Saeed Alghamdi': 0.0967741935483871, 'Abdussattar Shaikh': 0.04838709677419355,
'Osama Awadallah': 0.04838709677419355, 'Mohamed Atta': 0.3548387096774194,
'Abdul Aziz Al-Omari': 0.14516129032258063, 'Marwan Al-Shehhi': 0.29032258064516125,
'Ziad Jarrah': 0.16129032258064516, 'Said Bahaji': 0.11290322580645161,
'Ramzi Bin al-Shibh': 0.16129032258064516, 'Zakariya Essabar': 0.08064516129032258,
'Essid Sami Ben Khemais': 0.1774193548387097, 'Waleed Alshehri': 0.0967741935483871,
'Wail Alshehri': 0.0967741935483871, 'Satam Suqami': 0.12903225806451613,
'Fayez Ahmed': 0.12903225806451613, 'Mohand Alshehri': 0.03225806451612903,
'Mustafa Ahmed al-Hisawi': 0.06451612903225806, 'Ahmed Alghamdi': 0.08064516129032258,
'Ahmed Al Haznawi': 0.06451612903225806, 'Zacarias Moussaoui': 0.12903225806451613,
'Djamal Beghal': 0.12903225806451613, 'Jerome Courtaillier': 0.06451612903225806,
'David Courtaillier': 0.03225806451612903, 'Ahmed Ressay': 0.03225806451612903,
'Kamel Daoudi': 0.06451612903225806, 'Tarek Maaroufi': 0.0967741935483871,
'Faisal Al Salmi': 0.03225806451612903, 'Mohamed Abdi': 0.016129032258064516,
'Mamoun Darkazanli': 0.06451612903225806, 'Ahmed Khalil Ibrahim Samir Al-Ani': 0.016129032258064516,
'Agus Budiman': 0.08064516129032258, 'Mounir El Motassadeq': 0.06451612903225806,
'Abdelghani Mzoudi': 0.016129032258064516, 'Imad Eddin Barakat Yarkas': 0.06451612903225806,
'Nabil al-Marabh': 0.06451612903225806, 'Raed Hijazi': 0.06451612903225806,
'Abu Qatada': 0.08064516129032258, 'Mamduh Mahmud Salim': 0.016129032258064516,
'Mohammed Belfas': 0.03225806451612903, 'Abu Walid': 0.04838709677419355,
'Nabil Almarabh': 0.016129032258064516, 'Haydar Abu Doha': 0.04838709677419355,
'Mohamed Bensakhria': 0.06451612903225806, 'Lased Ben Heni': 0.03225806451612903,
'Mehdi Khammoun': 0.04838709677419355, 'Essoussi Laaroussi': 0.03225806451612903,
'Samir Kishk': 0.016129032258064516, 'Fahid al Shakri': 0.016129032258064516,
'Seifallah ben Hassine': 0.03225806451612903, 'Nizar Trabelsi': 0.016129032258064516,
'Jean-Marc Grandvisir': 0.016129032258064516, 'Abu Zubeida': 0.016129032258064516,
'Madjid Sahoune': 0.016129032258064516}
```

Figura 11: Output produzido

4.3.2 Eigenvector

O quão importante é um nodo em função de quão bem conectado está.

```
print("Eigenvector")
print(nx.eigenvector_centrality(g))
```

Figura 12: Método para cálculo de Eigenvector

```
Eigenvector
{'Hani Hanjour': 0.24855477543754168, 'Majed Moqed': 0.07494537631193433,
 'Nawaf Alhazmi': 0.14414554325478365, 'Khalid Al-Mihdhar': 0.08100272785369825,
 'Lotfi Raissi': 0.15798451142987605, 'Bandar Alhazmi': 0.03495916083672969,
 'Rayed Mohammed Abdullah': 0.05489124688535319, 'Salem Alhazmi': 0.17682068059796088,
 'Hamza Alghamdi': 0.09784977862339043, 'Ahmed Alnami': 0.03379845949219039,
 'Saeed Alghamdi': 0.05137316315262917, 'Abdussattar Shaikh': 0.02931670589054792,
 'Osama Awadallah': 0.02931670589054792, 'Mohamed Atta': 0.41159667925639287,
 'Abdul Aziz Al-Omari': 0.23736562716750392, 'Marwan Al-Shehhi': 0.39875763663923103,
 'Ziad Jarrah': 0.25751758482098774, 'Said Bahaji': 0.19846162931656117,
 'Ramzi Bin al-Shibh': 0.22342018766779756, 'Zakariya Essabar': 0.17162868593047698,
 'Essid Sami Ben Khemais': 0.05876111183218088, 'Waleed Alshehri': 0.15523154998479277,
 'Wail Alshehri': 0.18386395336365258, 'Satam Suqami': 0.19169926044941774,
 'Fayez Ahmed': 0.20131188186527696, 'Mohand Alshehri': 0.03446519093394617,
 'Mustafa Ahmed al-Hisawi': 0.13443329795374057, 'Ahmed Alghamdi': 0.09252192199884879,
 'Ahmed Al Haznawi': 0.09427780211536431, 'Zacarias Moussaoui': 0.08370910897081707,
 'Djamal Beghal': 0.017101399169235955, 'Jerome Courtaillier': 0.014533720895710688,
 'David Courtaillier': 0.011318507952501443, 'Ahmed Ressam': 0.010686807703606128,
 'Kamel Daoudi': 0.014019601056274813, 'Tarek Maaroufi': 0.022008018364731294,
 'Faisal Al Salmi': 0.03495916083672969, 'Mohamed Abdi': 0.01660680020100549,
 'Mamoun Darkazanli': 0.117784923297117, 'Ahmed Khalil Ibrahim Samir Al-Ani': 0.04741848302897727,
 'Agus Budiman': 0.15377097948367366, 'Mounir El Motassadeq': 0.1419610261352748,
 'Abdelghani Mzoudi': 0.04741848302897727, 'Imad Eddin Barakat Yarkas': 0.0784492469883354,
 'Nabil al-Marabh': 0.04250385613161701, 'Raed Hijazi': 0.03334264945124957,
 'Abu Qatada': 0.02391863215243939, 'Mamduh Mahmud Salim': 0.01356950485992108,
 'Mohammed Belfas': 0.043454723779354935, 'Abu Walid': 0.006341442516383653,
 'Nabil Almarabh': 0.0038412696990255666, 'Haydar Abu Doha': 0.009051039783135172,
 'Mohamed Bensakhria': 0.011285220538016425, 'Lased Ben Heni': 0.008070087704747724,
 'Mehdi Khammoun': 0.009112903673256388, 'Essoussi Laaroussi': 0.009305468433012318,
 'Samir Kishk': 0.006769837890331657, 'Fahid al Shakri': 0.006769837890331657,
 'Seifallah ben Hassine': 0.009305468433012318, 'Nizar Trabelsi': 0.001970397068852983,
 'Jean-Marc Grandvisir': 0.001970397068852983, 'Abu Zubeida': 0.001970397068852983,
 'Madjid Sahoune': 0.006769837890331657}
```

Figura 13: Output produzido

4.3.3 Closeness

Importância de um nodo em função da sua proximidade com os outros da rede.

```
print("closeness centralidade")
print(nx.closeness_centrality(g))
```

Figura 14: Método para cálculo de Closeness

```
closeness
{'Hani Hanjour': 0.4397163120567376, 'Majed Moqed': 0.3263157894736842,
 'Nawaf Alhazmi': 0.4397163120567376, 'Khalid Al-Mihdhar': 0.32978723404255317,
 'Lotfi Raissi': 0.4, 'Bandar Alhazmi': 0.30845771144278605,
 'Rayed Mohammed Abdullah': 0.31155778894472363, 'Salem Alhazmi': 0.36257309941520466,
 'Hamza Alghamdi': 0.36046511627906974, 'Ahmed Alnami': 0.3229166666666667,
 'Saeed Alghamdi': 0.3333333333333333, 'Abdussattar Shaikh': 0.31,
 'Osama Awadallah': 0.31, 'Mohamed Atta': 0.5794392523364486,
 'Abdul Aziz Al-Omari': 0.4217687074829932, 'Marwan Al-Shehhi': 0.4626865671641791,
 'Ziad Jarrah': 0.4189189189189189, 'Said Bahaji': 0.39490445859872614,
 'Ramzi Bin al-Shibh': 0.4305555555555556, 'Zakariya Essabar': 0.3875,
 'Essid Sami Ben Khemais': 0.42758620689655175, 'Waleed Alshehri': 0.33513513513513515,
 'Wail Alshehri': 0.4, 'Satam Suqami': 0.4105960264900662,
 'Fayez Ahmed': 0.40522875816993464, 'Mohand Alshehri': 0.31,
 'Mustafa Ahmed al-Hisawi': 0.38509316770186336, 'Ahmed Alghamdi': 0.33879781420765026,
 'Ahmed Al Haznawi': 0.3974358974358974, 'Zacarias Moussaoui': 0.4305555555555556,
 'Djamal Beghal': 0.328042328042328, 'Jerome Courtailier': 0.31313131313131315,
 'David Courtailier': 0.30392156862745096, 'Ahmed Ressam': 0.32124352331606215,
 'Kamel Daoudi': 0.31313131313131315, 'Tarek Maaroufi': 0.33879781420765026,
 'Faisal Al Salmi': 0.30845771144278605, 'Mohamed Abdi': 0.3069306930693069,
 'Mamoun Darkazanli': 0.38509316770186336, 'Ahmed Khalil Ibrahim Samir Al-Ani': 0.36904761904761907,
 'Agus Budiman': 0.3875, 'Mounir El Motassadeq': 0.38509316770186336,
 'Abdelghani Mzoudi': 0.36904761904761907, 'Imad Eddin Barakat Yarkas': 0.40522875816993464,
 'Nabil al-Marabh': 0.32460732984293195, 'Raed Hijazi': 0.3163265306122449,
 'Abu Qatada': 0.33879781420765026, 'Mamduh Mahmud Salim': 0.27927927927927926,
 'Mohammed Belfas': 0.30392156862745096, 'Abu Walid': 0.2616033755274262,
 'Nabil Almarabh': 0.24124513618677043, 'Haydar Abu Doha': 0.31794871794871793,
 'Mohamed Bensakhria': 0.3163265306122449, 'Lased Ben Heni': 0.3024390243902439,
 'Mehdi Khammoun': 0.3054187192118227, 'Essoussi Laaroussi': 0.31313131313131315,
 'Samir Kishk': 0.30097087378640774, 'Fahid al Shakri': 0.30097087378640774,
 'Seifallah ben Hassine': 0.31313131313131315, 'Nizar Trabelsi': 0.248,
 'Jean-Marc Grandvisir': 0.248, 'Abu Zubeida': 0.248, 'Madjid Sahoune': 0.30097087378640774}
```

Figura 15: Output produzido

4.3.4 Betweenness

Quantifica quantas vezes um nodo aparece nos caminhos mais curtos entre dois nodos

```
print("Betweenness centralidade")
print(nx.betweenness centrality(g))
```

Figura 16: Método para cálculo de Betweenness

```
Betweenness
{'Hani Hanjour': 0.1236167484316612, 'Majed Moqed': 0.0, 'Nawaf Alhazmi': 0.15207834940780468,
'Khalid Al-Mihdhar': 0.005429226158998767, 'Lotfi Raissi': 0.011469114351187326,
'Bandar Alhazmi': 0.0, 'Rayed Mohammed Abdullah': 0.0007932310946589106,
'Salem Alhazmi': 0.012347830499602044, 'Hamza Alghamdi': 0.021632373443584443,
'Ahmed Alnami': 0.0, 'Saeed Alghamdi': 0.017020472916824052,
'Abdussattar Shaikh': 0.0, 'Osama Awadallah': 0.0, 'Mohamed Atta': 0.5853750104411132,
'Abdul Aziz Al-Omari': 0.02251878945427331, 'Marwan Al-Shehhi': 0.08833725032826036,
'Ziad Jarrah': 0.01644821334139156, 'Said Bahaji': 0.0018861272695222984,
'Ramzi Bin al-Shibh': 0.046631043907617124, 'Zakariya Essabar': 0.0,
'Essid Sami Ben Khemais': 0.24879610958405246, 'Waleed Alshehri': 0.0008593336858804865,
'Wail Alshehri': 0.0023664463511105503, 'Satam Suqami': 0.0726383273580524,
'Fayez Ahmed': 0.02532461075559966, 'Mohand Alshehri': 0.0005288207297726071,
'Mustafa Ahmed al-Hisawi': 0.0023664463511105503, 'Ahmed Alghamdi': 0.007246103094860366,
'Ahmed Al Haznawi': 0.014955134177767716, 'Zacarias Moussaoui': 0.22979023444385693,
'Djamal Beghal': 0.10348140313766967, 'Jerome Courtaillier': 0.0015335801163405608,
'David Courtaillier': 0.0, 'Ahmed Ressam': 0.00691873788119161, 'Kamel Daoudi': 0.006248898290146307,
'Tarek Maaroufi': 0.03252247488101534, 'Faisal Al Salmi': 0.0, 'Mohamed Abdi': 0.0,
'Mamoun Darkazanli': 0.03225806451612903, 'Ahmed Khalil Ibrahim Samir Al-Ani': 0.0,
'Agus Budiman': 0.011175744755861097, 'Mounir El Motassadeq': 0.0, 'Abdelghani Mzoudi': 0.0,
'Imad Eddin Barakat Yarkas': 0.028922910334861683, 'Nabil al-Marabh': 0.0035208548429066676,
'Raed Hijazi': 0.033083612433162934, 'Abu Qatada': 0.04171514190022915, 'Mamduh Mahmud Salim': 0.0,
'Mohammed Belfas': 0.0, 'Abu Walid': 0.0008549268464657148, 'Nabil Almarabh': 0.0,
'Haydar Abu Doha': 0.006610259122157588, 'Mohamed Bensakhria': 0.0034373347435219465,
'Lased Ben Heni': 0.0, 'Mehdi Khammoun': 0.0005288207297726071, 'Essoussi Laaroussi': 0.0,
'Samir Kishk': 0.0, 'Fahid al Shakri': 0.0, 'Seifallah ben Hassine': 0.0, 'Nizar Trabelsi': 0.0,
Jean-Marc Grandvisir': 0.0, 'Abu Zubeida': 0.0, 'Madjid Sahoune': 0.0}
```

Figura 17: Output produzido

5 Grafo

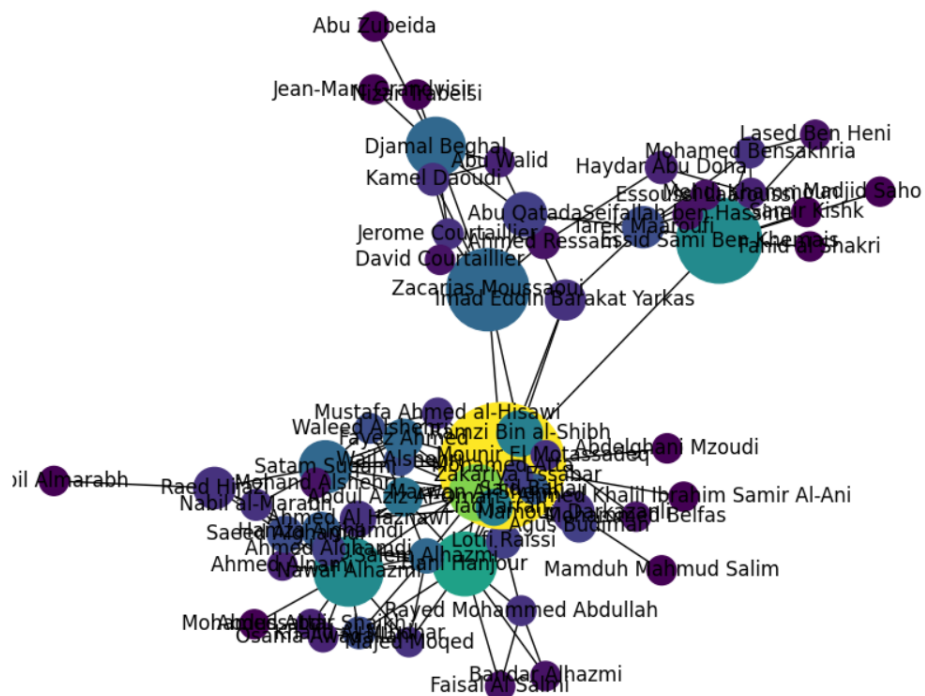


Figura 18: Grafo

6 Conclusão

Tendo em conta os objetivos que me propus a realizar, penso que posso concluir que os cumpra na íntegra pois consegui apresentar e calcular todas as medidas sobre o grafo.

Não encontrei nenhuma grande dificuldade durante toda a conceção do sistema visto a simplicidade do projeto proposto.