

```
pip install cdlib
```

```
→ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: cdlib in /usr/local/lib/python3.7/dist-packages (0.2.6)
Requirement already satisfied: angel-cd in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: thresholdclustering in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: python-Levenshtein in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: python-igraph in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: pyclustering in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: markov-clustering in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: pulp in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: eva-lcd in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: cython in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: ppoche in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: dynetx in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: python-louvain>=0.16 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: chinesewhispers in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: demon in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: nf1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: bimlpa in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: cyclerg>=0.10 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from cdlib)
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)

```
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: Pillow>=5.2.0 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: igraph==0.10.1 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from cdlib)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from cdlib)
```

```
from cdlib import algorithms, viz
import networkx as nx
import matplotlib.pyplot as plt
import networkx.algorithms.link_prediction as link_prediction
```

```
import pandas as pd  
import numpy as np
```

Note: to be able to use all crisp methods, you need to install some additional packages.  
Note: to be able to use all overlapping methods, you need to install some additional packages.  
Note: to be able to use all bipartite methods, you need to install some additional packages.



## ▼ Season

### ▼ Importação

```
S1 =pd.read_csv( '/content/drive/MyDrive/got-s1-edges_semanas7_8.csv' , sep=',')  
Grafo_Temporada1 = nx.from_pandas_edgelist(S1, source='Source', target= 'Target', edge_att
```

```
S8 =pd.read_csv( '/content/drive/MyDrive/got-s8-edges_semanas7_8.csv' , sep=',')  
Grafo_Temporada8 = nx.from_pandas_edgelist(S8, source='Source', target= 'Target', edge_att
```

▼ (Tarefa 1) Discuta quais as principais diferenças entre ambas as redes com base na distribuição de graus dos vértices (apresente os gráficos de histograma), grau médio, densidade e transitividade. Para estas três medidas, apresente os valores obtidos durante a discussão.

```
plt.figure(figsize=(12, 7))  
nx.draw(Grafo_Temporada1)
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)

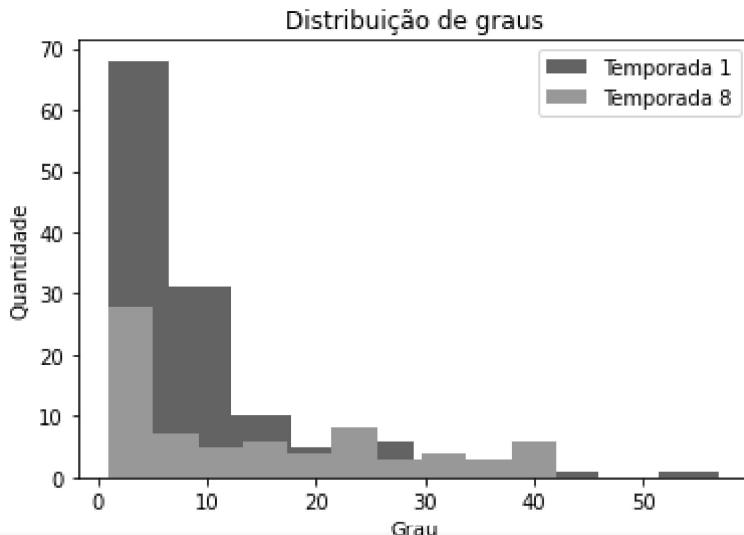


```

graus_S1 = [v for k, v in Grafo_Temporada1.degree()]
graus_S8 = [v for k, v in Grafo_Temporada8.degree()]
fig = plt.figure();
plt.hist(graus_S1)
plt.hist(graus_S8)
plt.title("Distribuição de graus")
plt.ylabel("Quantidade")
plt.xlabel("Grau")
plt.legend(['Temporada 1', 'Temporada 8'])

```

<matplotlib.legend.Legend at 0x7f2337da8a90>



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

Mostrar diferenças

```

print('Grau médio da Temporada 1: ', np.mean(graus_S1))
print('Grau médio da Temporada 8: ', np.mean(graus_S8))

```

```

Grau médio da Temporada 1:  8.714285714285714
Grau médio da Temporada 8:  14.945945945945946

```

```

print('Densidade da Temporada 1:', nx.density(Grafo_Temporada1))
print('Densidade da Temporada 8:', nx.density(Grafo_Temporada8))

```

```

Densidade da Temporada 1: 0.06971428571428571
Densidade da Temporada 8: 0.20473898556090336

```

```

print('Transitividade da Temporada 1:', nx.transitivity(Grafo_Temporada1))
print('Transitividade da Temporada 8:', nx.transitivity(Grafo_Temporada8))

```

```

Transitividade da Temporada 1: 0.38325183374083127
Transitividade da Temporada 8: 0.6441837028505615

```

(Tarefa 2) Para cada uma das redes, identifique quem são os 3 personagens mais centrais de acordo com as medidas de centralidade de grau, intermediação, proximidade e autovetor. Algum destes personagens é central em ambas as temporadas?

```
maior_caminho = -1
for i in Grafo_Temporada1:
    for j in Grafo_Temporada1:
        if i != j:
            try:
                tamanho_caminho = nx.shortest_path_length(Grafo_Temporada1,i,j)
                if tamanho_caminho > maior_caminho:
                    maior_caminho = tamanho_caminho
                    personagem1 = i
                    personagem2 = j
            except nx.NetworkXNoPath:
                pass

print(f'Os personagens mais distantes da Temporada 1 são {personagem1} e {personagem2}, e o caminho é composto por: ', nx.shortest_path(Grafo_Temporada1, personagem1, personagem2))

Os personagens mais distantes da Temporada 1 são IROGENIA e MELESSA, e o custo do caminho é composto por: ['IROGENIA', 'DOREAH', 'DAENERYS', 'NED', 'JON', 'SAM', 'MELESSA']

maior_caminho = -1
for i in Grafo_Temporada8:
    for j in Grafo_Temporada8:
        Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.
        Mostrar diferenças
        tamanho_caminho = nx.shortest_path_length(Grafo_Temporada8,i,j)
        if tamanho_caminho > maior_caminho:
            maior_caminho = tamanho_caminho
            personagem1 = i
            personagem2 = j
        except nx.NetworkXNoPath:
            pass

print(f'Os personagens mais distantes da Temporada 2 são {personagem1} e {personagem2}, e o caminho é composto por: ', nx.shortest_path(Grafo_Temporada8, personagem1, personagem2))

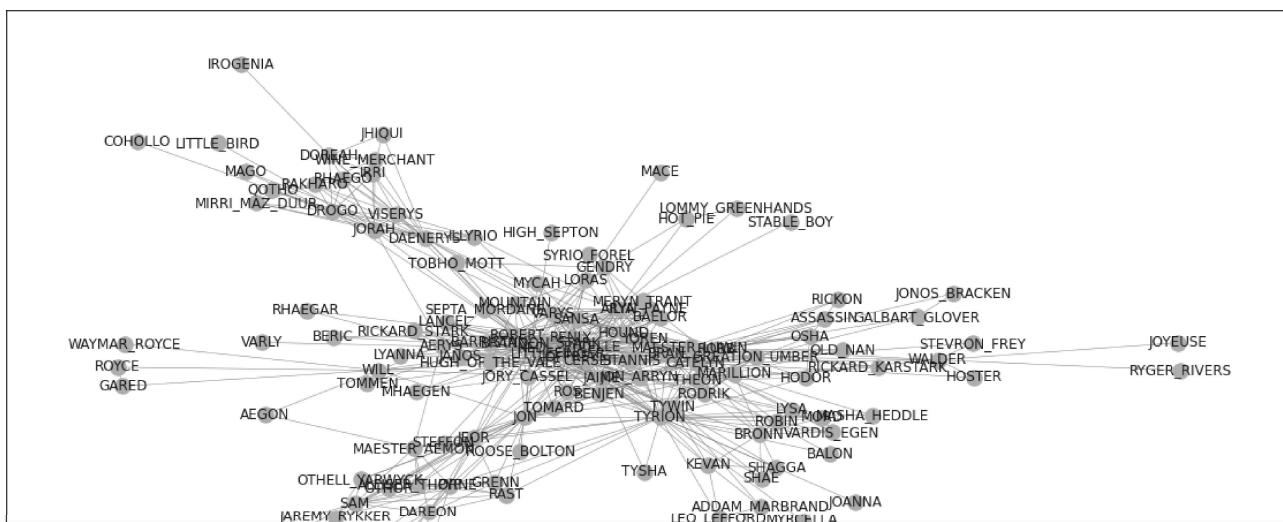
Os personagens mais distantes da Temporada 2 são ALANNA e WILLIAM, e o custo do caminho é composto por: ['ALANNA', 'OWEN', 'ARYA', 'TYRION', 'BRONN', 'CRAYAH', 'WILLIAM']
```

(Tarefa 3) Execute o mesmo algoritmo de detecção de comunidades (de sua preferência) nas redes das duas temporadas. Apresente o grafo com os vértices coloridos de acordo com as comunidades encontradas e com os nomes dos vértices para sua identificação. O número de comunidades é o mesmo em ambos os casos? A partir de uma análise visual, é possível identificar uma comunidade similar (em termos de personagens) em ambas as temporadas?

```
pos = nx.spring_layout(Grafo_Temporada1)
```

```
pos8 = nx.spring_layout(Grafo_Temporada8)
```

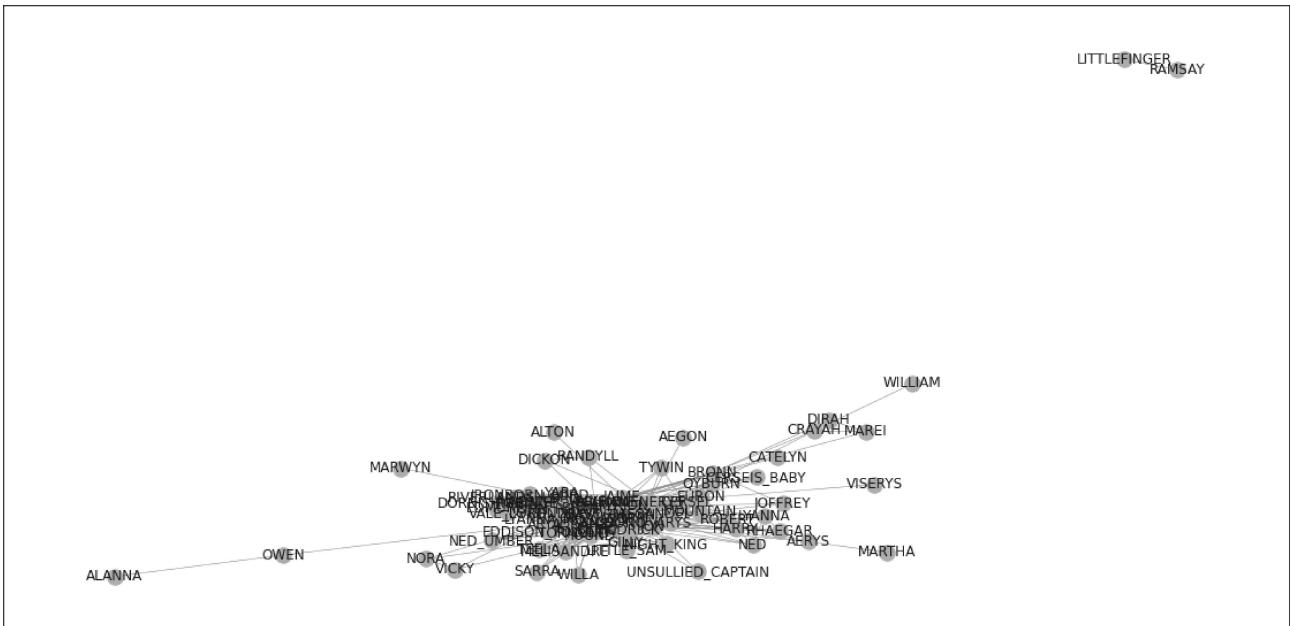
```
plt.figure(figsize=(20,10))
nx.draw_networkx(Grafo_Temporada1, pos, with_labels=True, node_size=180, node_color='Oran
```



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra quia.

[Mostrar diferenças](#)

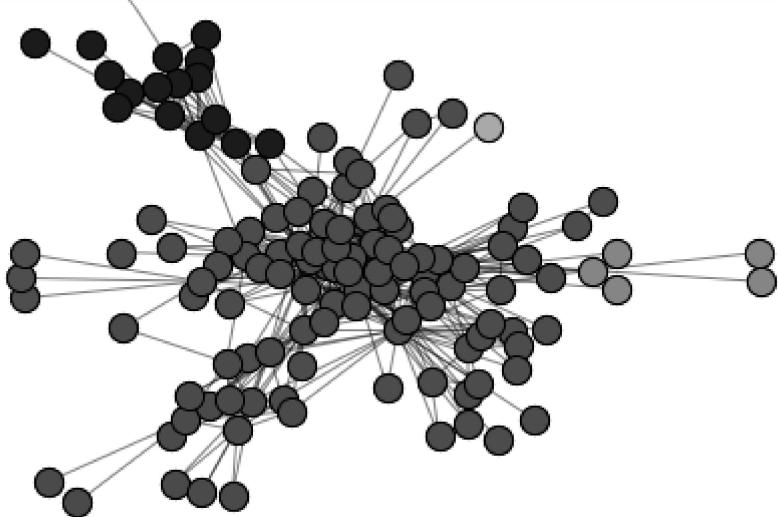
```
plt.figure(figsize=(20,10))
nx.draw_networkx(Grafo_Temporada8, pos8, with_labels=True, node_size=180, node_color='OrangeRed')
```



```
comunidades = algorithms.girvan_newman(Grafo_Temporada1, level=5)
viz.plot_network_clusters(Grafo_Temporada1, comunidades, pos, figsize=(8,6))
```

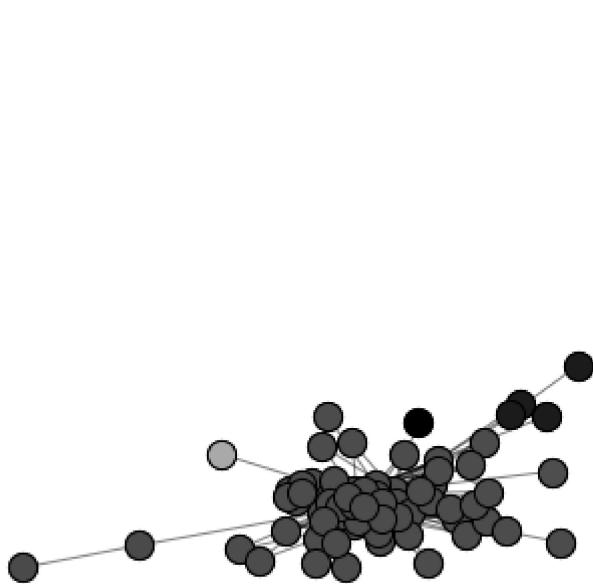
```
<matplotlib.collections.PathCollection at 0x7f2337b99bd0>
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)



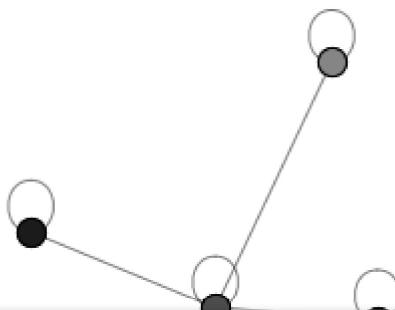
```
comunidades = algorithms.girvan_newman(Grafo_Temporada8, level=5)
viz.plot_network_clusters(Grafo_Temporada8, comunidades, pos8, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23379175d0>
```



```
viz.plot_community_graph(Grafo_Temporada1, comunidades, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f233ddb0890>
```

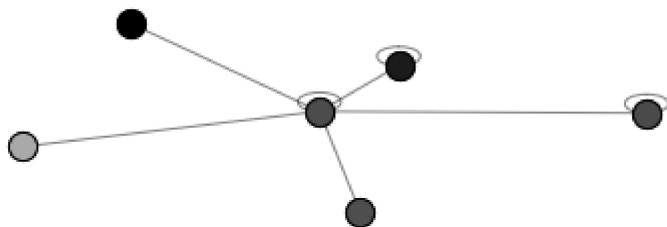


Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)



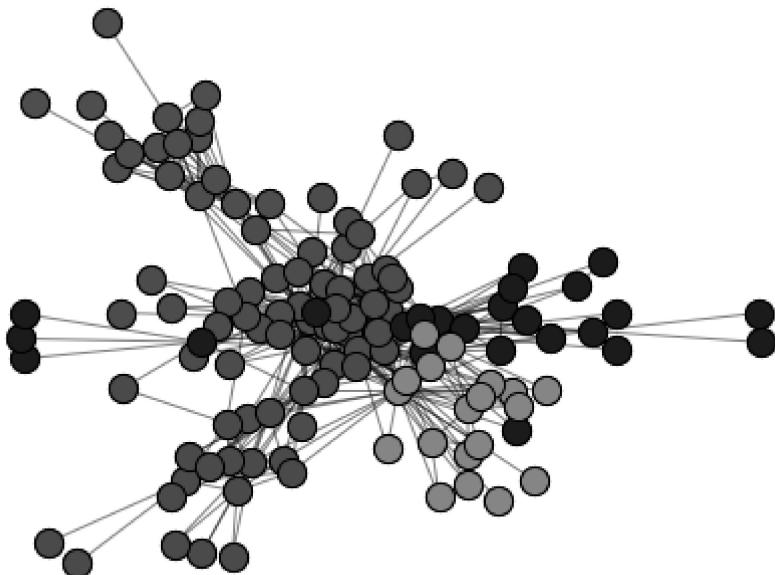
```
viz.plot_community_graph(Grafo_Temporada8, comunidades, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2338b69610>
```



```
comunidades2 = algorithms.louvain(Grafo_Temporada1)
viz.plot_network_clusters(Grafo_Temporada1, comunidades2, pos, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23387e0e90>
```



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

[Mostrar diferenças](#)

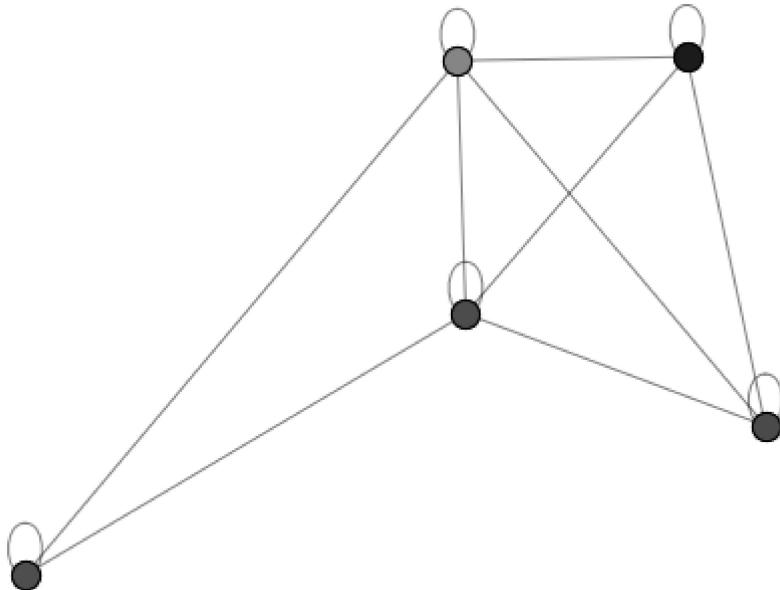
```
viz.plot_network_clusters(Grafo_Temporada8, comunidadess2, pos8, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23384941d0>
```



```
viz.plot_community_graph(Grafo_Temporada1, comunidades2, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23387ceed0>
```



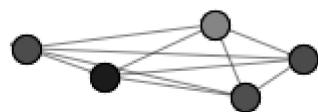
```
viz.plot_community_graph(Grafo_Temporada8, comunidadess2, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2337fa3e10>
```



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.

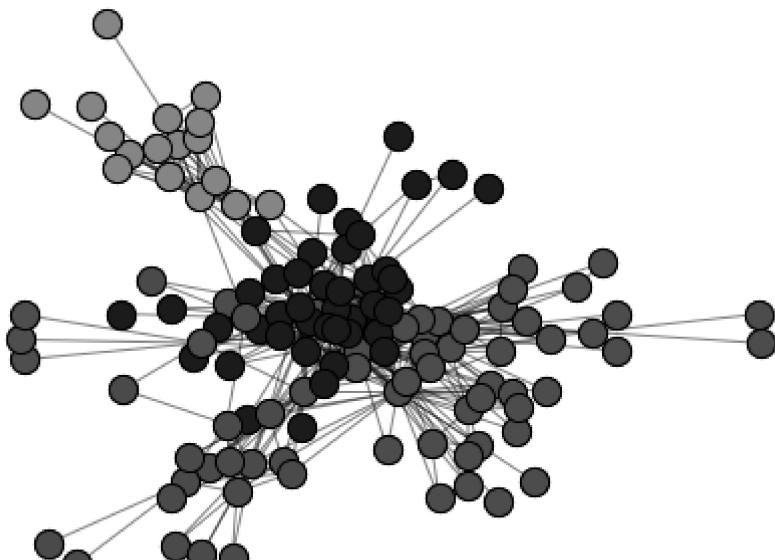
[Mostrar diferenças](#)



```
comunidades3 = algorithms.greedy_modularity(Grafo_Temporada1)
```

```
viz.plot_network_clusters(Grafo_Temporada1, comunidades3, pos, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23386b4b90>
```



```
comunidades3 = algorithms.greedy_modularity(Grafo_Temporada8)
viz.plot_network_clusters(Grafo_Temporada8, comunidades3, pos8, figsize=(8,6))
```

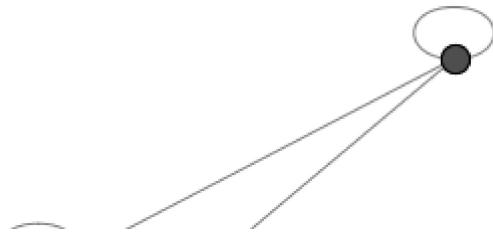
```
<matplotlib.collections.PathCollection at 0x7f23396cd210>
```



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)

```
viz.plot_community_graph(Grafo_Temporada1, comunidades3, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2338629bd0>
```



```
viz.plot_community_graph(Grafo_Temporada8, comunidades3, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2337e1cf50>
```

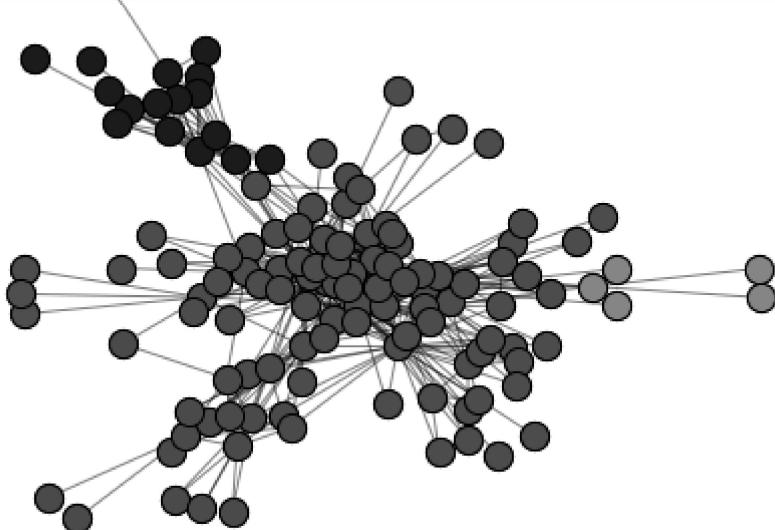


```
comunidades4 = algorithms.label_propagation(Grafo_Temporada1)
```

```
viz.plot_network_clusters(Grafo_Temporada1, comunidades4, pos, figsize=(8,6))
```

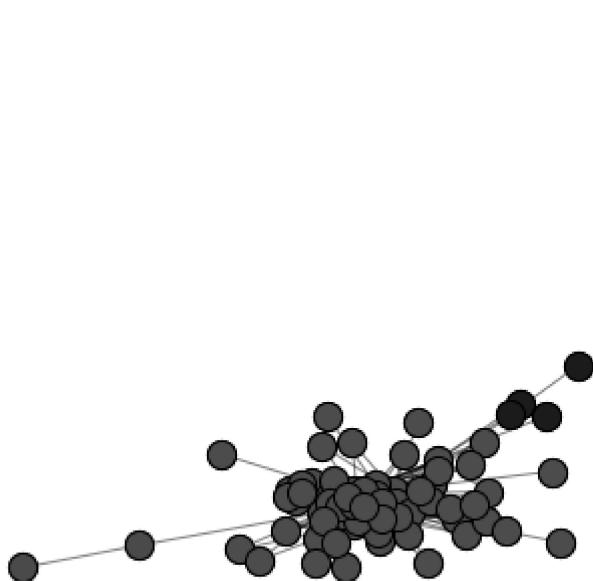
```
<matplotlib.collections.PathCollection at 0x7f2338629bd0>
```

Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)



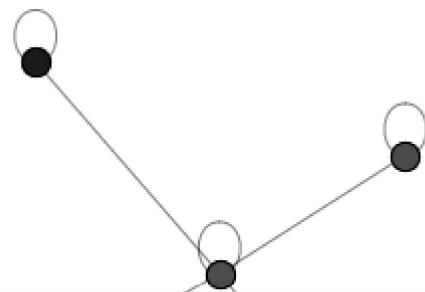
```
comunidadess4 = algorithms.label_propagation(Grafo_Temporada8)
viz.plot_network_clusters(Grafo_Temporada8, comunidadess4, pos8, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2337e03790>
```



```
viz.plot_community_graph(Grafo_Temporada1, comunidades4, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f23384efc90>
```

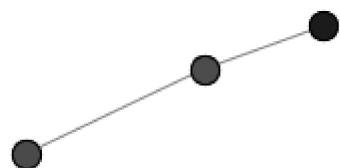


Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)



```
viz.plot_community_graph(Grafo_Temporada8, comunidadess4, figsize=(8,6))
```

```
<matplotlib.collections.PathCollection at 0x7f2338285ed0>
```



Falha no salvamento automático. Este arquivo foi atualizado remotamente ou em outra guia.  
[Mostrar diferenças](#)