Lista 5: Network Models

Nome (Nro. USP): Aimê G. da Nobrega (11882429)
Disciplina: Redes Complexas (SME0130)
Docente: Francisco A. Rodrigues
USP São Carlos - ICMC

In [1]:
```
from numpy import *
import numpy as np
```

In [2]:
```
import networkx as nx
```

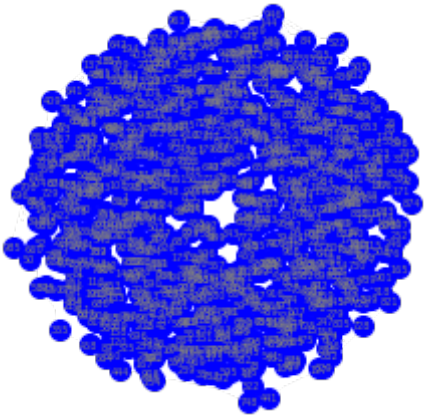
In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
def degree_distribution(GER):
    vk = dict(GER.degree())
    vk = list(vk.values()) # we get only the degree values
    maxk = np.max(vk)
    mink = np.min(min)
    kvalues = arange(0, maxk+1) # possible values of k
    Pk = np.zeros(maxk +1) #P(k)
    for k in vk:
        Pk[k] = Pk[k] +1
    Pk = Pk/sum(Pk) # sum of elements of P(k)
    return kvalues, Pk

def momment_of_degree_distribution(G, m):
    k, Pk = degree_distribution(G)
    M = sum((k**m)*Pk)
    return M
def shannon_entropy(G):
    k,Pk = degree_distribution(G)
    H = 0
    for p in Pk:
        if(p > 0):
            H = H - p*math.log(p, 2)
    return H

def plot_net(net):
    pos = nx.fruchterman_reingold_layout(net);
    plt.figure(figsize = (4,4));
    plt.axis('off');
    nx.draw_networkx_nodes(net, pos, node_size=100, node_color="b
lue");
    nx.draw_networkx_edges(net, pos, alpha=0.500, edge_color = 'w
hite');
    nx.draw_networkx_labels(net, pos, font_color="gray", font_siz
e = 5);
    plt.show();
```

1 -Gere um grafo aleatório com N=1000 e p = 0.1. Qual o valor do grau médio, segundo momento do grau e coeficiente de aglormeração médio (average clustering coefficient)?

In [5]:
```
N = 1000 # nodes
p = 0.1
av_degree = p*(N-1)
GER = nx.gnp_random_graph(N, p, seed=42, directed = False)
```

In [6]:
```
plot_net(GER)
```



Grau médio:

In [7]:
```
k = momment_of_degree_distribution(GER, 1)
```

Segundo momento do grau:

In [8]:
```
k2 = momment_of_degree_distribution(GER, 2)
```

Coeficiente de aglomeração médio:

In [9]:
```
avc = nx.average_clustering(GER)
```

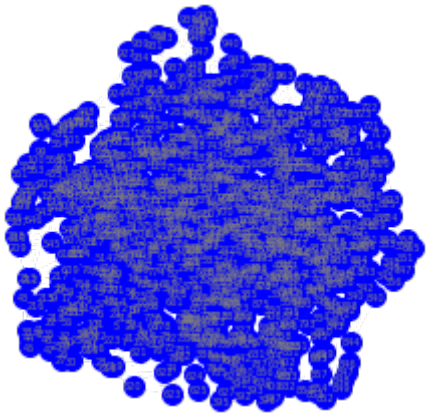
In [10]:
```
print('Grau médio: ', '%.1f'%k)
print('Second momment of P(k): ', k2)
print('Average clustering: ', '%3.4f'%avc, '|p = ', p)
```

```
Grau médio:  99.9
Second momment of P(k):  10062.856
Average clustering:  0.1002 |p =  0.1
```

2 -Gere um small-world com N=1000, grau médio igual 10 e p = 0.1. Qual o valor do grau médio, segundo momento do grau e coeficiente de aglormeração médio (average clustering coefficient)?

In [11]:
```
N = 1000
k = 10
p = 0.1 #probability of rewiring
GWS = nx.watts_strogatz_graph(N, k, p, seed=None)
```

In [12]:
```
plot_net(GWS)
```



Grau médio:

In [13]:
```
k = momment_of_degree_distribution(GWS, 1)
```

Segundo momento do grau:

In [14]:
```
k2 = momment_of_degree_distribution(GWS, 2)
```

Coeficiente de aglomeração médio:

In [15]:
```
avc = nx.average_clustering(GWS)
```

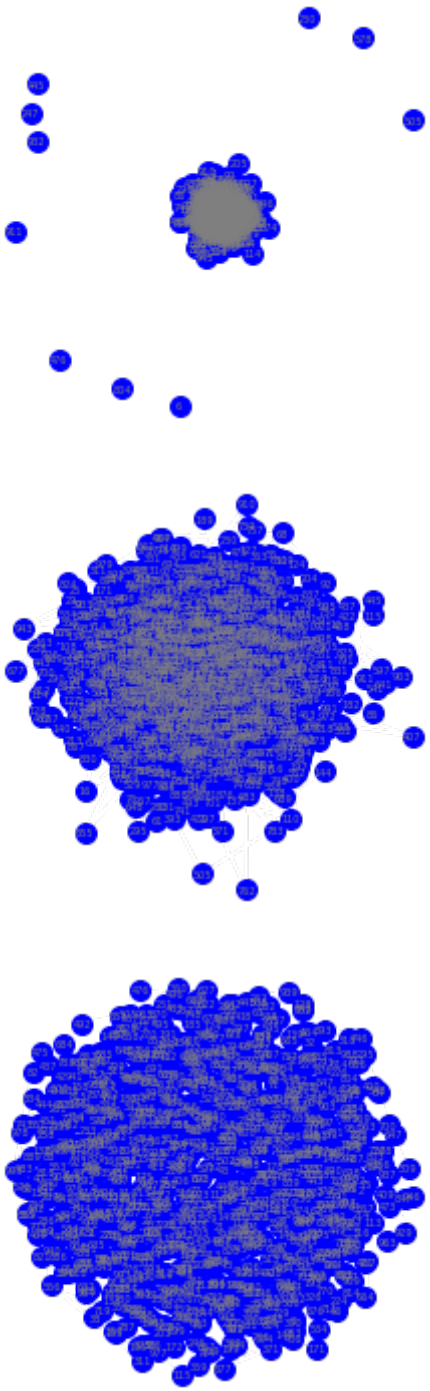
In [16]:
```
print('Grau médio: ', '%.1f'%k)
print('Second momment of P(k): ', '%3.4f'%k2)
print('Average clustering: ', '%.1f'%avc, '|p = ', p)
```

```
Grau médio:  10.0
Second momment of P(k):  100.9480
Average clustering:  0.5 |p =  0.1
```

3 -Considere uma rede aleatória (Erdos-Renyi) com N=1000 vértices. Qual o valor da entropia de Shannon do grau para $\langle k \rangle$= 5, $\langle k \rangle$=10,$\langle k \rangle$=50.

In [17]:
```
k = [5,10,50]

GER_l = []
for i in k:
    p = i/(N-1)
    GER_l.append(nx.erdos_renyi_graph(1000,p, seed=42, directed =
False))
```

In [18]:
```
for i in GER_l:
    plot_net(i)
```



In [19]:
```
for i in np.arange(3):
    print('Entropia de Shannon para K =',k[i],':', '%.1f'%shannon
_entropy(GER_l[i]))
```

```
Entropia de Shannon para K = 5 : 3.2
Entropia de Shannon para K = 10 : 3.6
Entropia de Shannon para K = 50 : 4.7
```

4 -Para o modelo small-world, calcule o valor da menor distância média (average shortest path) para p=0; p=0.01; p=0.05 e p=0.1. Considere grau médio igual a 10.

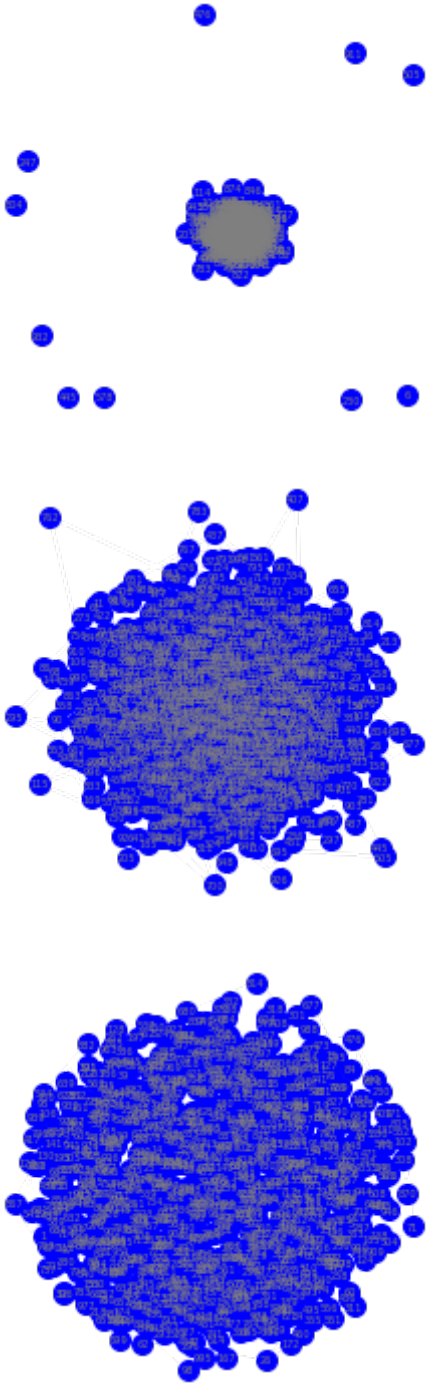
In [20]:
```
N = 100
av_degree = 8
k = int(av_degree/2)
p = [0, 0.01, 0.05, 0.1] #probability of rewiring
av_shortest_path = []

for i in p:
    for j in range(30):
        GWS = nx.watts_strogatz_graph(N, k, i, seed=j)
        av_shortest_path.append(nx.average_shortest_path_length(G
WS))
    media = sum(av_shortest_path)/30
    print('Menor distancia media para p = ', i,':','%.1f'%media)
    av_shortest_path = []
```

```
Menor distancia media para p =  0 : 12.9
Menor distancia media para p =  0.01 : 10.2
Menor distancia media para p =  0.05 : 6.2
Menor distancia media para p =  0.1 : 4.9
```

5 -Considere o modelo de Erdos-Renyi. Gere redes com grau médio igual a 5, 10 e 50 e N=1000. Qual o valor da assortatividade?

In [21]:
```
N = 1000 # nodes
av_degree = [5, 10, 50]
GER_l = []
for i in av_degree:
    p = i/(N-1)
    GER_l.append(nx.gnp_random_graph(N, p, seed=42, directed = Fa
lse))
```

In [22]:
```
for i in GER_l:
    plot_net(i)
```



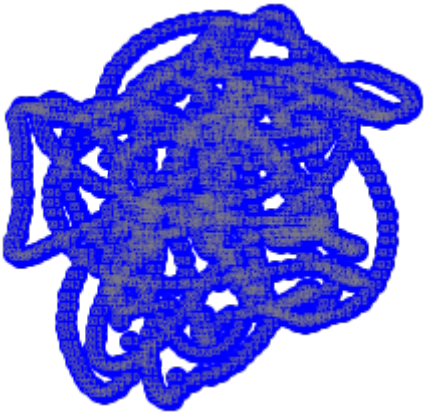
In [23]:
```
for i in range(3):
    print('Valor de assostatividade para grau médio igual a', av_
degree[i],':', '%3.2f'%nx.degree_assortativity_coefficient(GER_l
[i]))
```

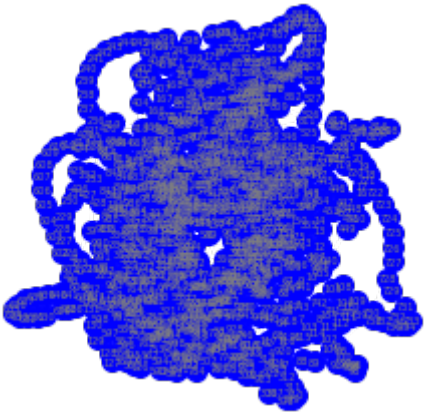
```
Valor de assostatividade para grau médio igual a 5 : -0.01
Valor de assostatividade para grau médio igual a 10 : -0.01
Valor de assostatividade para grau médio igual a 50 : -0.00
```

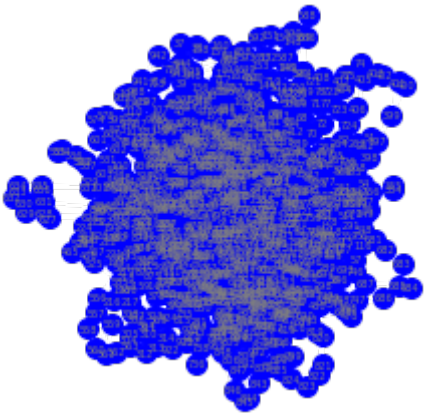
6 -Considere o modelo small-world. Gere redes com grau médio 10 e N=1000. Qual o valor da assortatividade para p=0.01; 0.05 e 0.2?

In [24]:
```
N = 1000
av_degree = 10
k = int(av_degree/2)
p = [0.01, 0.05, 0.2] #probability of rewiring
GWS_l = []
for i in p:
    GWS_l.append(nx.watts_strogatz_graph(N, k, i, seed=42))
```

In [25]:
```
for i in GWS_l:
    plot_net(i)
```







In [26]:
```
for i in range(3):
    print('Valor da assortividade para p = ',p[i],':', '%3.1f'%n
x.degree_assortativity_coefficient(GWS_l[i]))
```

```
Valor da assortividade para p =  0.01 : 0.0
Valor da assortividade para p =  0.05 : -0.0
Valor da assortividade para p =  0.2 : -0.0
```