

Anexo C: Síntese das redes de amizade e de interação do Participa.br

Importando bibliotecas:

```
In [89]: from SPARQLWrapper import SPARQLWrapper, JSON
import time, numpy as n, networkx as x
```

Definição de prefixos úteis para as consultas SparQL:

```
In [90]: PREFIX=""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX schema: <http://schema.org/>""
```

Buscando amigos do Participa.br:

```
In [91]: NOW=time.time()
q="""SELECT DISTINCT ?aname ?bname
      WHERE {
        ?a foaf:knows ?b .
        ?a foaf:name ?aname .
        ?b foaf:name ?bname .
      }"""
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todas as amizades do Participa.br"%
      (time.time()-NOW,))
```

0.07 segundos para puxar todas as amizades do Participa.br

Erigindo rede de amizades como um grafo não direcionado:

```
In [92]: g=x.Graph()
for amizade in results4["results"]["bindings"]:
    nome1=amizade["aname"]["value"]
    nome2=amizade["bname"]["value"]
    g.add_edge(nome1,nome2)
```

```
In [93]: print(u"são %i amizades entre %i pessoas no Participa.br"%
      (g.number_of_edges(), g.number_of_nodes()))
```

são 910 amizades entre 443 pessoas no Participa.br

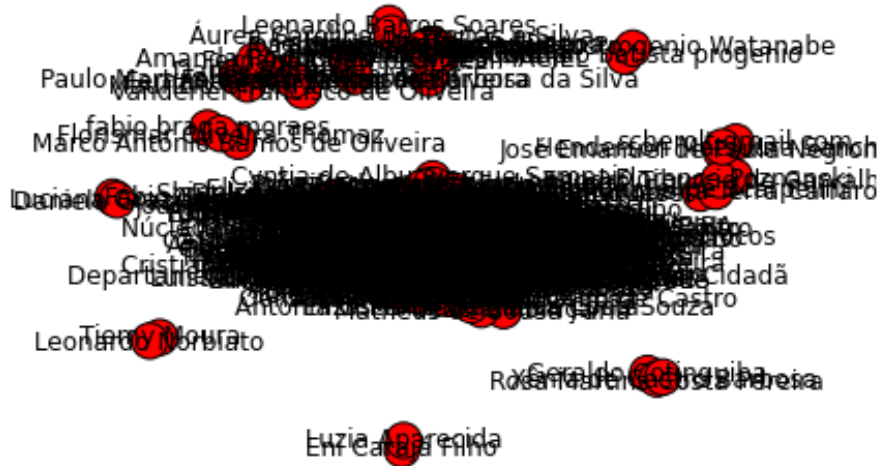
```
In [141]: x.write_graphml(g,"amizadesParticipa.graphml")
```

Para visualizar e analisar, a estrutura de amizades está pronta neste link:

<https://github.com/ttm/pnud3/blob/master/amizadesParticipa.graphml>.

Usos são imediatos, p.ex:

```
In [94]: x.draw(g,pos=x.layout.fruchterman_reingold_layout(g))
```



Puxando as interações de comentários do Participa.br:

```
In [95]: NOW=time.time()
q2="""SELECT DISTINCT ?aname ?bname
WHERE {
    ?comentario dc:type tsioc:Comment.
    ?participante1 ops:performsParticipation ?comentario.
    ?participante1 foaf:name ?aname.
    ?artigo sioc:has_reply ?comentario.
    ?participante2 ops:performsParticipation ?artigo.
    ?participante2 foaf:name ?bname.
}"""
sparql3.setQuery(PREFIX+q2)
sparql3.setReturnFormat(JSON)
results = sparql3.query().convert()
print("%.2f segundos para puxar as interações do Participa.br"%
      (time.time()-NOW,))
```

15.58 segundos para puxar as interações do Participa.br

Sintetizando rede direcionada de interação:

```
In [96]: d=x.DiGraph()
for interacao in results["results"]["bindings"]:
    nome_chegada=interacao["aname"]["value"]
    nome_partida=interacao["bname"]["value"]
    if (nome_partida,nome_chegada) in d.edges():
        d[nome_partida][nome_chegada]["weight"]+=1
```

```
else:
    d.add_edge(nome partida,nome chegada,weight=1.)
```

```
In [142]: x.write_graphml(d,"interacoesParticipa.graphml")
```

Estrutura de interação pronta para análises e visualizações neste link:

<https://github.com/ttm/pnud3/blob/master/interacoesParticipa.graphml>.

Elencando as 15 pessoas mais conectadas via **amizades**:

```
In [100]: sorted_g = sorted(g.degree().iteritems(), key=operator.itemgetter(1))
sorted_g[::-1][:15]
```

```
Out[100]: [(u'Marcelo Branco', 122),
(u'Maria Jos\xea9lia Amaral de Menezes', 60),
(u'Ana C\xea9lia Costa', 36),
(u'Vicente Aguiar', 35),
(u'Laura Zacher', 33),
(u'Luis Felipe Coimbra Costa', 32),
(u'Ricardo Poppi', 31),
(u'LUCAS MOREIRA DE SOUSA', 30),
(u'Ronald Emerson Scherolt da Costa', 29),
(u'Valessio Brito', 27),
(u'Grazielle Machado', 20),
(u'Andr\xea9 Filipe de Assun\xea7\xea3o e Brito', 20),
(u'Renato Fabbri', 20),
(u'Daniela Feitosa', 18),
(u'Daniel Pitangueira de Avelino', 18)]
```

Para medida de centralidade, as listagens acima usam quantidade de conexão, ou grau.

Medidas de interesse imediato são: de atravessador (presente em mais geodésicas do grafo, *betweenness*), de mais "perto de todos os outros" (*closeness*) e de auto-vetor:

```
In [105]: bc=x.betweenness_centrality(g)
cc=x.closeness_centrality(g)
```

```
In [106]: sorted_bc = sorted(bc.iteritems(), key=operator.itemgetter(1))
sorted_bc[::-1][:15]
```

```
Out[106]: [(u'Marcelo Branco', 0.42882163196866385),
(u'Maria Jos\xea9lia Amaral de Menezes', 0.1500460662666134),
(u'LUCAS MOREIRA DE SOUSA', 0.09201604347662772),
(u'Ana C\xea9lia Costa', 0.07270058962510031),
(u'Grazielle Machado', 0.060658560472243744),
(u'Laura Zacher', 0.05514913696577906),
(u'Valessio Brito', 0.05235162430751249),
(u'Nilzete Pacheco ', 0.049895256955160554),
(u'C\xelssio N. Martin', 0.04578534023146533),
(u'Jesulino Alves de Souza', 0.04285934737795151),
(u'Vicente Aguiar', 0.042734420429139115),
(u'Daniel Pitangueira de Avelino', 0.03954188318259412),
(u'Luis Felipe Coimbra Costa', 0.036605448228684445),
(u'kelly cristina alves', 0.035831717510820746),
(u'Ricardo Poppi', 0.034899619183799586)]
```

```
In [107]: sorted_cc = sorted(cc.iteritems(), key=operator.itemgetter(1))
sorted_cc[::-1][:15]
```

```
Out[107]: [(u'Marcelo Branco', 0.44299145920610333),
(u'Maria Jos\xea9lia Amaral de Menezes', 0.3820683267566498),
(u'Vicente Aguiar', 0.36163480373131945),
```

```
(u'Ronald Emerson Scherolt da Costa', 0.36090643554656354),
(u'Grazielle Machado', 0.35873882932706463),
(u'Ana C\xe9lia Costa', 0.35695228137224855),
(u'Valessio Brito', 0.3562426346896),
(u'Luis Felipe Coimbra Costa', 0.34895821859565485),
(u'Ricardo Poppi', 0.34895821859565485),
(u'Jesulino Alves de Souza', 0.3469313557577324),
(u'Paulo Henrique de Lima Santana', 0.3442652166164626),
(u'Daniela Feitosa', 0.3384136831895539),
(u'Paulo Meirelles', 0.3384136831895539),
(u'Laura Zacher', 0.33524797988562915),
(u'Andr\xe9 Filipe de Assun\xe7\xe3o e Brito', 0.3346219332378502)]
```

Para as redes de interação:

```
In [108]: bcd=x.betweenness centrality(d)
          ccd=x.closeness centrality(d)
```

```
In [109]: sorted_bcd = sorted(bcd.iteritems(), key=operator.itemgetter(1))
          sorted_bcd[::-1][:15]
```

```
Out[109]: [(u'Renato Fabbri', 0.027219251336898394),
           (u'Grazielle Machado', 0.019025092554504318),
           (u'Ricardo Poppi', 0.013204442616207322),
           (u'andre luiz da silva', 0.008021390374331552),
           (u'Hylton Sarcinelli Luz', 0.007853078294254766),
           (u'JO\x30 BATISTA CAVALCANTI', 0.007260386672151378),
           (u'Daniel Pitangueira de Avelino', 0.007198683669271905),
           (u'Jose mendon\xe7a Furtado Neto', 0.004669546140134375),
           (u'Jorge', 0.004668860551213492),
           (u'Caroline Burle', 0.0046482928835870005),
           (u'Gabriela Valle', 0.004324009324009324),
           (u'Tiago Henrique Ferreira Zoroastro', 0.0033730974907445494),
           (u'Vicente Aguiar', 0.003331962155491567),
           (u'Frank Lane', 0.0031262854792266552),
           (u'edison cardoso teixeira', 0.0030234471410941995)]
```

```
In [110]: sorted_ccd = sorted(ccd.iteritems(), key=operator.itemgetter(1))
          sorted_ccd[::-1][:15]
```

```
Out[110]: [(u'Portal', 0.18563072508553138),
           (u'Renato Fabbri', 0.0896842747470639),
           (u'andre luiz da silva', 0.08298272075361858),
           (u'Henrique Parra Parra Filho', 0.07487334184166763),
           (u'Marcelo Rodrigues Saldanha da Silva', 0.07365123564218587),
           (u'Jose mendon\xe7a Furtado Neto', 0.07349614053766303),
           (u'Cleber Possani Jr', 0.07341788286278758),
           (u'Hylton Sarcinelli Luz', 0.07064662093125092),
           (u'Daniel Pitangueira de Avelino', 0.06848317922486721),
           (u'Juliano Geraldi', 0.06619519942465014),
           (u'Roberto Kodama', 0.06435394670688788),
           (u'Josenildo Fernando da Silva', 0.06338269573563692),
           (u'Maira Gon\xe7alves Lopes', 0.062471719457013575),
           (u'Ricardo Poppi', 0.06187519730611386),

           (u'Ronald Emerson Scherolt da Costa', 0.05675816292745991)]
```

C.2. Setores conectivos

Na caracterização qualitativa das redes livres de escala, há uma distinção que salta aos olhos: uma multidão de vértices pouco conectados (periféricos), pouquíssimos vértices muito conectados (hubs), e alguns intermediários (intermediários mesmo).

Esta divisão é usualmente concebida na proporção 5% hubs, 15% intermediários e 80% periféricos.

- $\approx 5 \pm 4\%$ dos vértices mais conectados (entidades operantes) possuem $\approx 70 \pm 10\%$ das arestas (recursos). São os hubs.
- $\approx 70 \pm 10\%$ dos vértices menos conectados possuem $\approx 20 \pm 12\%$ das arestas. São os periféricos.
- $\approx 25 \pm 14\%$ dos vértices restantes possuem $\approx 16 \pm 10\%$ das arestas. São os intermediários.

```
In [144]: hi=int(len(sorted_g)*0.05)
          ii=int(len(sorted_g)*0.15)
          pi=int(len(sorted_g)*0.80)
          sorted_g_=sorted_g[::-1]
          hubs=sorted_g[:hi]
          intermediarios=sorted_g[hi:ii]
          perifericos=sorted_g[ii:]
```

```
In [117]: hubs
```

```
Out[117]: [(u'Marcelo Branco', 122),
            (u'Maria Jos\x9lia Amaral de Menezes', 60),
            (u'Ana C\x9lia Costa', 36),
            (u'Vicente Aguiar', 35),
            (u'Laura Zacher', 33),
            (u'Luis Felipe Coimbra Costa', 32),
            (u'Ricardo Poppi', 31),
            (u'LUCAS MOREIRA DE SOUSA', 30),
            (u'Ronald Emerson Scherolt da Costa', 29),
            (u'Valessio Brito', 27),
            (u'Grazielle Machado', 20),
            (u'Andr\x9 Filipe de Assun\x7\xe3o e Brito', 20),
            (u'Renato Fabbri', 20),
            (u'Daniela Feitosa', 18),
            (u'Daniel Pitangueira de Avelino', 18),
            (u'Nilzete Pacheco ', 18),
            (u'Carolina de Abreu Batista Claro', 17),
            (u'Jesulino Alves de Souza', 17),
            (u'Paulo Meirelles', 17),
            (u'Paulo Henrique de Lima Santana', 15),
            (u'kelly cristina alves', 14),
            (u'Daniel Chiaretti', 14)]
```

Descomentar para acessar participantes nos setores:

```
In [ ]: # intermediarios
        # perifericos
```

E para a rede de interação:

```
In [123]: hi=int(len(sorted_d)*0.05)
          ii=int(len(sorted_d)*0.15)
          pi=int(len(sorted_d)*0.80)
          sorted_d=sorted_d[::-1]
          hubs_d=sorted_d[:hi]
          intermediarios_d=sorted_d[hi:ii]
          perifericos_d=sorted_d[ii:]
```

```
In [125]: hubs_d
          # intermediarios_d
          # perifericos_d
```

```
Out[125]: [(u'Portal', 46),
           (u'Renato Fabbri', 26),
           (u'Gabriela Valle', 19),
           (u'Hylton Sarcinelli Luz', 15),
           (u'F\xeldia Rebou\xe7as', 13),
           (u'Grazielle Machado', 12),
           (u'Jose mendon\xe7a Furtado Neto', 11),
           (u'andre luiz da silva', 11),
           (u'Daniel Pitangueira de Avelino', 10),
           (u'Henrique Parra Parra Filho', 9),
           (u'Juliano Geraldi', 8)]
```

Há correlação com a atividade de cada participante, principalmente na rede de interação.

Além disso, é recomendada a comparação com a rede de Erdős-Renyi com o mesmo número de vértices e arestas, como feito neste trabalho: <http://sourceforge.net/p/labmacambira/fimDoMundo/ci/master/tree/textos/evolutionSN/paper.pdf?format=raw>.

C.3. Detecção de comunidades

```
In [145]: [aa for aa in x.community.k_clique_communities(g,5)]
```

```
Out[145]: [frozenset([u'Daniel Pitangueira de Avelino',
                      u'Francisco Marcelo de Ara\xfajo Lima J\xfanior',
                      u'Grazielle Machado',
                      u'Arthur de Moura Del Esposte',
                      u'Carolina Fonseca',
                      u'Fabiano Rangel Cidade',
                      u'Marina Arilha Silva',
                      u'Valessio Brito',
                      u'Paulo Meirelles',
                      u'Ana C\xe9lia Costa',
                      u'Ronald Emerson Scherolt da Costa',
                      u'Ricardo Poppi',
                      u'Bertoni',
                      u'Daniela Feitosa',
                      u'Vicente de Paula Blum de Oliveira',
                      u'Gustavo Buzzatti Pacheco',
                      u'Paulo Henrique de Lima Santana',
                      u'Juliana Hil\xelrio',
                      u'Mariel Zasso',
```

```

u'Everton dos Santos Silva Rodrigues',
u'Vicente Aguiar',
u'Luis Felipe Coimbra Costa',
u'Aur\x9lio A. Heckert',
u'Nitai Bezerra da Silva',
u'Renato Fabbri',
u'Maria Jos\x9lia Amaral de Menezes',
u'Luciano Santa Brigida',
u'Christian Miranda',
u'Jesulino Alves de Souza',
u'Marcelo Branco',
u'Andr\x9 Filipe de Assun\x7\x3o e Brito',
u'Sady Jacques']],
frozenset([u'Daniela Corr\xea Jacques Brauner',
u'Daniel Chiaretti',
u'Nara de Souza Rivitti',
u'Laura Zacher',
u'JOAO FREITAS DE CASTRO CHAVES',
u'Fernando de Souza Carvalho',
u'Alexandra Rodrigues Lazzarini',
u'Bruno Vinicius Batista Arruda',
u'Fernanda Hahn',
u'aidio junior mariano gon\x7alves',
u'Sabrina Piccoli Marques',
u'ANA LUISA ZAGO DE MORAES',
u'Isabel Penido de Campos Machado',
u'Maria Jos\x9lia Amaral de Menezes',
u'MARIA L\xda CIA NIDBALLA DOS SANTOS',
u'Marcelo Branco',
u'Eraldo Silva J\xfanior']],
frozenset([u'Andrea Pacheco Pac\xedfico',
u'Carolina de Abreu Batista Claro',
u'Helisane Mahlke',
u'La\xeds Azeredo Alves',
u'erika pires ramos',
u'Shirley Cardoso Gon\x7alves de Aquino Alves']])

```

As comunidades acima foram definidas pelo número de amizades mínimo em comum no grupo. Há bons métodos espectrais (Newman) e iterativos.

|||--- FIM ---|||

