



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

## **MOBILIDADE URBANA E COVID-19: A ESCALA METROPOLITANA**

Leticia da Silva Cabral

Relatório de Iniciação Científica do  
programa PIBIC, orientada pelo  
Dr. Antônio Miguel Vieira Mon-  
teiro e co-orientada pelo Dr. Leo-  
nardo Barcelar Lima Santos

INPE  
São José dos Campos  
2020

## RESUMO

A população mundial está passando por uma pandemia, que se alastrou através do adensamento populacional e, principalmente, por conta da mobilidade e da globalização em que vivemos. As fronteiras são facilmente atravessadas todos os dias, recebendo um fluxo de pessoas, coisas e, eventualmente, vírus com potencial pandêmico, como o COVID-19. No Sudeste brasileiro, as capitais regionais tem forte conexão com a rede urbana paulista, com intenso tráfego diário entre os municípios por diversos motivos, sejam eles de trabalho, prestação de serviço, saúde, educação ou lazer. Sendo assim, no contexto em que estamos, faz-se muito necessário entender e analisar a forma, a intensidade e o diálogo intermunicipal brasileiro. Neste estudo, analisamos o fluxo e a conectividade dos 39 municípios que compõem a RMVPLN, conectados também aos municípios de Belo Horizonte, Campinas, Rio de Janeiro e São Paulo, através da elaboração de (geo)grafos, que são compostos por vértices de localização espacial conhecida, ligados por arestas que representam o fluxo entre eles.

## LISTA DE FIGURAS

	<u>Pág.</u>
4.1 Dado de saída do código - <i>shapefile</i> vértices e arestas . . . . .	6
4.2 Trecho do código - simetrização . . . . .	7
4.3 Opções do formato da simetrização da matriz de fluxo . . . . .	7
4.4 (geo)grafo RMVPLN - Indicador degree . . . . .	8
4.5 (geo)grafo RMVPLN - Indicador peso das arestas . . . . .	9

## LISTA DE ABREVIATURAS E SIGLAS

- RMVPLN – Região Metropolitana do Vale do Paraíba e Litoral Norte  
IBGE – Instituto Brasileiro de Geografia e Estatística



## SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO . . . . .	2
2 OBJETIVOS . . . . .	3
3 METODOLOGIA . . . . .	4
4 DISCUSSÃO DOS RESULTADOS . . . . .	5
5 CONCLUSÃO . . . . .	9
REFERÊNCIAS BIBLIOGRÁFICAS . . . . .	10

# 1 INTRODUÇÃO

Em apenas 3 meses, o vírus da COVID-19 saiu de Wuhan na China e se alastrou por 180 países, evidenciando que as fronteiras internacionais, e nacionais também, são facilmente cruzadas e que vive-se em uma rede de mobilidade complexa. Como enunciado por [Monteiro et al. \(2020\)](#), a vida não se reduz aos limites administrativos e, muito menos, um inimigo que nem as conhece, portanto apenas planejar cidades é insuficiente neste caso.

Além do adensamento populacional, fatores como conexões com eixos rodoviários, existência de aeroportos e universidades, que atraem estudantes de diversos lugares do Brasil, também influenciam na disseminação da COVID-19([PERES, 2020](#)). O movimento pendular, que acontece do domicílio ao trabalho, local de estudo ou lazer, é um dos aspectos que fazem parte da análise do deslocamento de pessoas([JARDIM, 2011](#)) e pode acontecer de forma intermunicipal, também contribuindo para o espalhamento de doenças contagiosas.

A RMVPLN é formada por 39 municípios divididos em 5 sub-regiões, abrigando uma população de aproximadamente 2,5 milhões de pessoas ([EMPLASA, 2016](#)). Tem forte interação social com a rede urbana paulista, com a capital e outras capitais regionais como Campinas, que neste ano foi considerada pelo ([IBGE, 2020](#)) como uma das 15 metrópoles brasileiras. Recebe intenso fluxo de pessoas diariamente, tendo por consequência uma grande exposição, que pode ser perigosa em fases epidêmicas, como a enfrentada atualmente no país.

(Geo)grafos ([SANTOS et al., 2017](#)) são grafos com vértices de localização espacial fixa e conhecida interligados por arestas que representam suas dependências espaciais<sup>(1)</sup>. O uso de (geo)grafos para a análise de dados de mobilidade urbana no cenário pandêmico que enfrentamos hoje, com a atual crise sanitária, é muito importante para elucidar de forma clara as conexões regionais ou intermunicipais. Principalmente com o uso de indicadores que forneçam informações valiosas para a interpretação dos dados, como o grau de conectividade ou a intensidade com que o fluxo de viagens acontece.

---

<sup>(1)</sup>[...] the concept of (geo)graphs: graphs in which the nodes have a known geographical location and the edges have spatial dependence.

## **2 OBJETIVOS**

Construção de (geo)grafos da RMVPLN, também conectada aos municípios de Belo Horizonte, Campinas, Rio de Janeiro e São Paulo, aplicando diferentes simbologias às métricas dos nós para entender a forma e a intensidade com que esses municípios se conectam. Podendo assim, analisar a importância da mobilidade urbana no contexto da crise sanitária de COVID-19.

### 3 METODOLOGIA

Inicialmente, desenvolveu-se três diferentes tutoriais, anexados à este relatório, que exemplificam como executar dois códigos, previamente escritos e cedidos à este estudo, em linguagem *Python* na plataforma gratuita *Google Colaboratory*, bem como o formato dos dados de entrada e a visualização, download e aplicações dos dados de saída.

A segunda etapa, deu-se pela construção de mapas a partir dos dados de saída desses códigos, que continham as métricas dos nós, os vértices e as arestas dos (geo)grafos da Região Metropolitana do Vale do Paraíba e Litoral Norte, conectada entre si e à outros 4 municípios, sendo eles Belo Horizonte, Campinas, Rio de Janeiro e São Paulo.

Para o primeiro código, usou-se como dado de entrada a longitude e a latitude dos centróides extraídos dos *shapefiles* dos municípios, que estão disponíveis na base de dados do IBGE, e uma matriz de fluxo, preparada a partir das matrizes ORIGEM-DESTINO geradas e cedidas à esse estudo<sup>(2)</sup>. Para usá-las fez-se necessário remover as linhas e colunas com dados de 'outros municípios', que não tinham localização espacial definida, sendo não condizentes com as definições de um (geo)grafo. Como dado de saída, obteve-se um *shapefile* com os vértices e outro com as arestas que compõem o (geo)grafo.

No segundo código, usou-se como dado de entrada as matrizes de fluxo e o limiar, que foi definido como número 1, ou seja, qualquer aresta com fluxo maior ou igual a 1 era forte o suficiente para aparecer no (geo)grafo. A partir disso, foi obtido como dado de saída as métricas dos nós.

Por fim, foram produzidos 16 mapas com o uso do software *ArcGis*. As matrizes de fluxo usadas foram: COLETIVO-ESCOLAR, COLETIVO-SERVIÇO, COLETIVO-SAÚDE, COLETIVO-TRABALHO, INDIVIDUAL-ESCOLAR, INDIVIDUAL-SERVIÇO, INDIVIDUAL-SAÚDE E INDIVIDUAL-TRABALHO, contendo o fluxo intermunicipal por dia. Para cada uma delas, foram feitos 2 mapas, usando em um deles o indicador de grau de centralidade (*degree*), dado pela métrica dos nós e, no outro, o peso das arestas, ou seja, a quantidade de fluxo de cada uma delas, fornecida pelas informações contidas na tabela de atributos do *shapefile*.

---

<sup>(2)</sup>por Bruna Pizzol - LabGeo/Eng. Transportes da Poli-USP utilizando a Pesquisa OD da RMV-PLN de 2014 da Secretaria dos Transportes Metropolitanos de SP

No período em que houve a extensão da bolsa durante o mês de agosto, esses resultados foram apresentados em uma reunião com um grupo de pesquisadores FAPESP, que abrangem diversas áreas como saúde, geografia, matemática, redes complexas e epidemiologia, e têm discutido ao longo dos últimos meses, através do olhar das diferentes áreas de conhecimento, o avanço da COVID-19 no Brasil.

Além disso, foram feitas análises dos dados<sup>(3)</sup> de mobilidade multi-modais das 342 zonas do município de São Paulo. OS dados disponibilizados são muito completos, disposto em 30 tabelas, observando não só a mobilidade das pessoas, mas também sua faixa etária, renda familiar mensal média, grau de escolaridade, gênero, entre outros.

Os dados de mobilidade estão divididos em seis matrizes. A primeira apresenta os dados de deslocamento por modo coletivo, como ônibus, metrô, transporte escolar e fretado. A segunda contém os dados de transporte por modo individual, como passageiro ou motorista de veículo particular, táxi, bicicleta, passageiro ou motorista de motocicleta. Na terceira matriz OD estão dispostos os dados das viagens feitas a pé. Na quarta tem-se o deslocamento por modo motorizado tanto coletivo quanto individual. Já na quinta estão os dados do deslocamento não motorizado, ou seja, apenas de bicicleta ou a pé. Por fim, na última matriz OD, está o total de deslocamento, independente do modo, por zona, separadas como origem ou destino.

A última matriz mencionada, foi utilizada para calcular o fluxo total por zona do município de São Paulo. Ou seja, somou-se o fluxo da zona sendo ela ponto de origem do deslocamento, ou sendo destino do deslocamento. Matematicamente, para cada zona, somou-se o total da linha com o total da coluna da matriz, subtraindo-se o elemento da diagonal principal, para que ele não seja somado duas vezes.

#### 4 DISCUSSÃO DOS RESULTADOS

A RMVPLN e a forma como ela se conecta entre si e com os municípios de Belo Horizonte, Campinas, Rio de Janeiro e São Paulo delimitam a nossa região de estudo. Foram usadas 8 matrizes de fluxo diferentes como dado de entrada para a produção dos (geo)grafos, definidas por MODO-MOTIVO, ou seja, o modo e o motivo do deslocamento, sendo elas: COLETIVO-ESCOLAR, COLETIVO-SERVIÇO, COLETIVO-SAÚDE, COLETIVO-TRABALHO, INDIVIDUAL-ESCOLAR, INDIVIDUAL-SERVIÇO, INDIVIDUAL-SAÚDE E INDIVIDUAL-TRABALHO.

---

<sup>(3)</sup>Pesquisa Origem-Destino 2017, disponibilizadas pelo Metro-SP.

Após executar o primeiro código usando a lista de centroides dos municípios e suas respectivas longitude e latitude junto com uma das matrizes de fluxo como dado de entrada, obtém-se um shapefile com os vértices e outro com as arestas de fluxo que compõem o (geo)grafo.

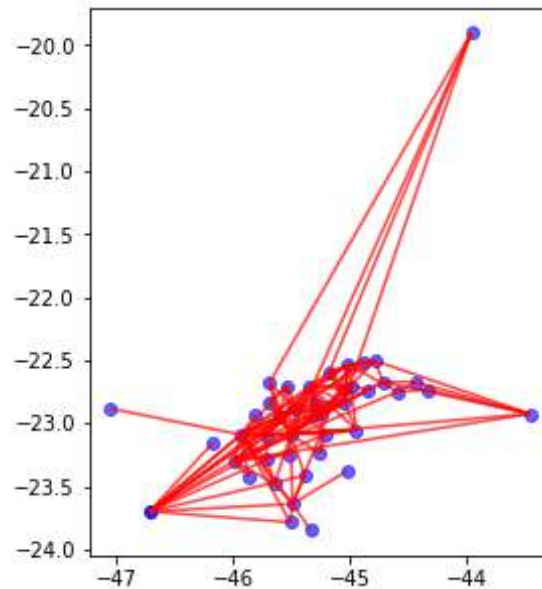


Figura 4.1 - Dado de saída do código - *shapefile* vértices e arestas

O número de viagens está disposto na matriz separando em dois valores o fluxo de um município, ora ele sendo ponto de origem ora de destino. Ao executar o código, optou-se por realizar a simetrização das matrizes através da soma dos valores dispostos acima e abaixo da diagonal principal, para que assim o fluxo, independente do seu sentido, fosse totalmente considerado. Outras opções como simetrizar pelo maior, pelo menor ou pela média também são possíveis, porém não eram do interesse do nosso estudo.

Os vértices dos mapas representam os centroides dos municípios de estudo e as arestas correspondem ao fluxo intermunicipal, ligando dois nós diferentes. Escolheu-se dois tipos de indicadores para a produção dos mapas. Usando o grau de centralidade (degree), que é dado pelas métricas dos nós, é possível visualizar o número de conexões de um determinado vértice do grafo, ou seja, com quantos municípios diferentes um determinado município se conecta. Foi aplicada uma escala de intensidade de cor, em que os centroides de cor verde mais escuro têm menor grau de conectividade do que os centroides representados em vermelho. Isso é importante para demonstrar

## ▾ Symmetrizing the matrix

```
[12] # def simetric_matrix(dados):  
#     for i in range(len(dados.columns)):  
#         for j in range(len(dados.columns)):  
#             if(dados.iloc[i,j]>dados.iloc[j,i]):  
#                 dados.iloc[j,i] = dados.iloc[i,j]  
#             elif (dados.iloc[j,i]>dados.iloc[i,j]):  
#                 dados.iloc[i,j] = dados.iloc[j,i]  
#     return dados
```

```
[13] #Symmetrizing by higher  
def simetric_matrix_higher(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[13] #Symmetrizing by lower  
def simetric_matrix_lower(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]<dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]<dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados  
  
#Symmetrizing by sum  
def simetric_matrix_sum(dados):  
    original=dados.copy()  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            dados.iloc[i,j] = original.iloc[i,j] + original.iloc[j,i]  
    return dados  
  
#Symmetrizing by average  
def simetric_matrix_avg(dados):  
    original=dados.copy()  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            dados.iloc[i,j] = (original.iloc[i,j] + original.iloc[j,i])/2  
    return dados
```

Figura 4.2 - Trecho do código - simetrização

```
[ ] #You have to choose which function you would like to apply  
  
#mtx = simetric_matrix_higher(mtx)  
#mtx = simetric_matrix_lower(mtx)  
mtx = simetric_matrix_sum(mtx)  
#mtx = simetric_matrix_avg(mtx)
```

Figura 4.3 - Opções do formato da simetrização da matriz de fluxo

que os municípios não estão isolados, se conectando diariamente.

Modo de deslocamento: **Individual** - Motivo do deslocamento: **Trabalho**, RMVPLN

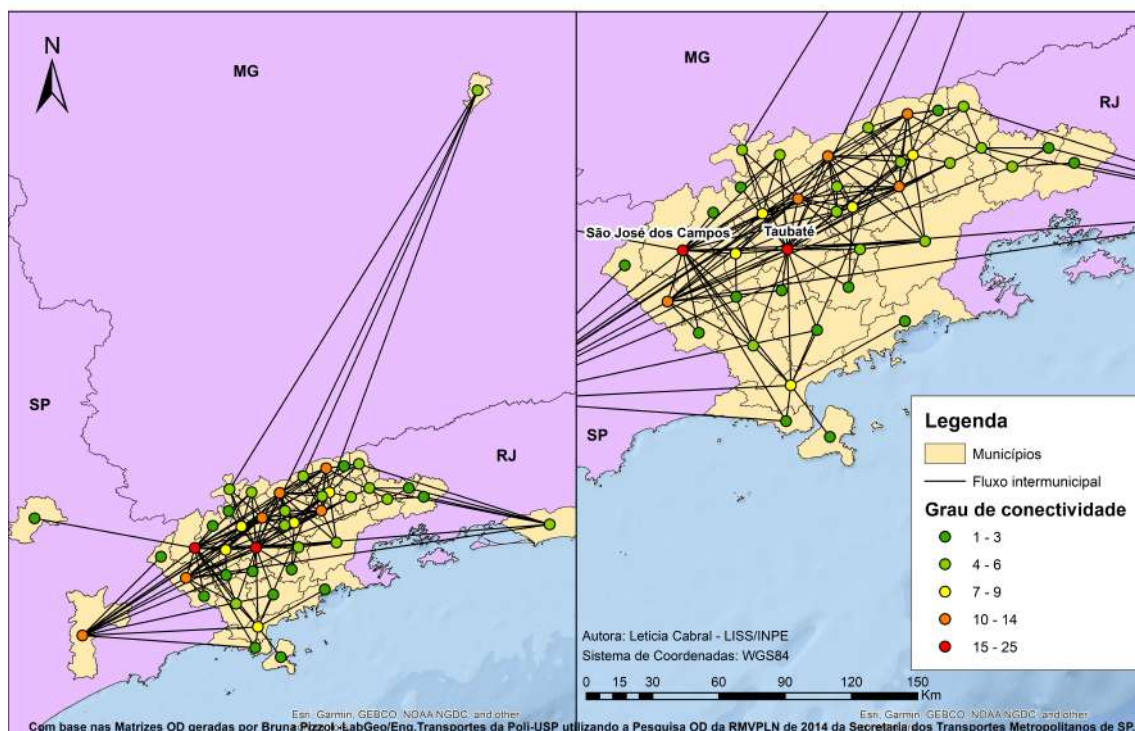


Figura 4.4 - (geo)grafo RMVPLN - Indicador degree

O outro indicador usado, foi o peso das arestas, ou seja, a quantidade de fluxo ou de número de viagens contida em cada uma delas. Também aplicou-se neste caso o uso da escala de cor, representando as arestas de menor fluxo em verde escuro e as de maior fluxo em vermelho. Esta é uma outra forma de elucidar a conexão entre os municípios, mas diferente do outro indicador que mostra o número de conexões, este mostra a intensidade das mesmas.

Através da observação desses dados, é possível notar a forte conectividade entre os municípios, levando à reflexão sobre os impactos de um intenso fluxo intermunicipal no espalhamento de vírus contagiosos como o COVID-19. O isolamento social pode ser fator importante para o controle e ajudar a manter um baixo  $R(t)$ , que em um modelo matemático epidemiológico, significa quantas pessoas um infectado irá infectar. Para a contenção de uma epidemia, é essencial que o  $R(t)$  fique abaixo de 1, estima-se que hoje o maior  $R(t)$  entre os estados brasileiro seja 1,4, mas esse valor já esteve próximo de 8<sup>(4)</sup>.

<sup>(4)</sup>Dados estimados por Flavio Figueiredo. Atualizados em: 07/07/2020. Disponível em: <<https://github.com/flaviofdf/covid19>>



Modo de deslocamento: **Individual** - Motivo do deslocamento: **Trabalho**, RMVPLN

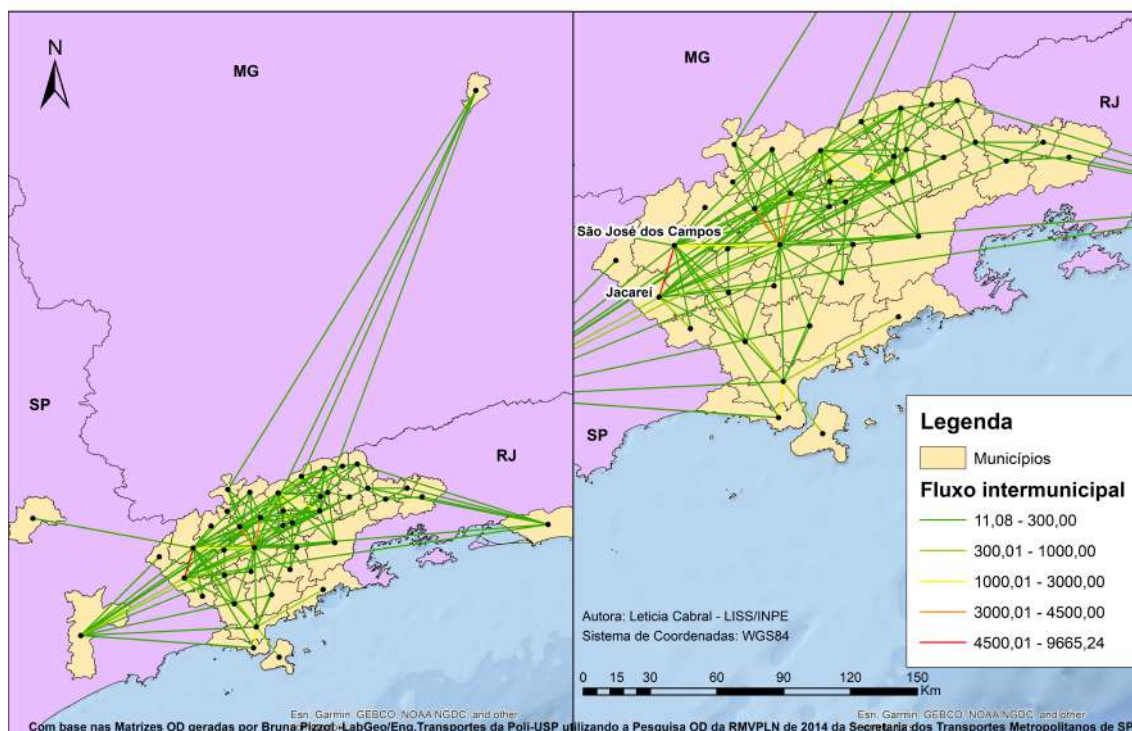


Figura 4.5 - (geo)grafo RMVPLN - Indicador peso das arestas

## 5 CONCLUSÃO

A disseminação desse vírus foi marcada pelo adensamento populacional e pela mobilidade, começando no centro da China e se alastrando por 180 países depois de apenas 3 meses, marcando bem a globalização em que vivemos. No Brasil, é possível notar que o modelo de difusão se deu de forma hierárquica, das cidades maiores para as menores em efeito cascata, saindo das grandes metrópoles e chegando nas capitais regionais, como por exemplo alguns municípios da RMVPLN, que são grandes cidades do interior de São Paulo e tem uma forte interação social com a rede urbana paulista.

Diante desses fatos e do atual cenário pandêmico, faz-se necessário observar e analisar a mobilidade urbana que nos cerca, bem como a forma com a qual ela vem acontecendo e como pode contribuir para diferentes resultados da epidemia de COVID-19 no Brasil.

## REFERÊNCIAS BIBLIOGRÁFICAS

EMPLASA. **Região Metropolitana do Vale do Paraíba e Litoral Norte**. 2016. <https://emplasa.sp.gov.br/RMVPLN: :text=Extensa>. Acesso em: 07 jul. 2020. 2

IBGE. **REGIC 2018: Campinas/SP, Florianópolis/SC e Vitória/ES passam a estar entre as 15 Metrópoles do país**. 2020. <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/28042-regic-2018-campinas-sp-florianopolis-sc-e-vitoria-es-passam-a-estar-entre-as-15-metropoles-do-pais>. Acesso em: 07 jul. 2020. 2

JARDIM, A. P. Reflexões sobre os deslocamentos populacionais no Brasil. In: \_\_\_\_\_. [S.l.]: IBGE, 2011. cap. Movimento pendular, p. 58. 2

MONTEIRO, M. A. V.; ANAZAWA, T. M.; OLIVEIRA, G. C. Covid-19 - crises entremeadas no contexto de pandemia (antecedentes, cenários e recomendações). In: \_\_\_\_\_. [S.l.]: CPOI, 2020. cap. 14, p. 219–235. 2

PERES, R. B. Covid-19 - crises entremeadas no contexto de pandemia (antecedentes, cenários e recomendações). In: \_\_\_\_\_. [S.l.]: CPOI, 2020. cap. 16, p. 247. 2

SANTOS, L.; JORGE, A.; ROSSATO, M.; SANTOS, J.; CANDIDO, O.; CERON, W.; SANTANA, C. D. (geo)graphs - complex networks as a shapefile of nodes and a shapefile of edges for different applications. 11 2017. 2



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



Laboratório de investigação  
em Sistemas Sócio-Ambientais

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO A PARTIR DA EXECUÇÃO DE CÓDIGO EM PYTHON NO GOOGLE COLAB

LISS - INPE

2020

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Este tutorial é produto de iniciação científica.

Elaboração do estudo: **LISS** – Laboratório de Investigação em Sistemas **S**ócio-Ambientais

Autor: Leticia da Silva Cabral

Orientador: Dr. Antonio Miguel Vieira Monteiro – LISS/INPE

Coorientador: Leonardo Bacelar de Lima Santos – CEMADEN

Colaboradores: Cátia Souza do Nascimento Sepetauskas e Jeferson Feitosa Mendes

CABRAL, L. S. ; MONTEIRO, A. M. V. ; SANTOS, L. B. L. ; MENDES, J. F. ; NASCIMENTO, C. S.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## APRESENTAÇÃO

Neste tutorial você aprenderá a:

Preparar os  
dados de  
entrada do  
código

Executar código  
no google colab

Entender o dado  
de saída

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Esse tutorial trata da elaboração de (geo)grafos, que são grafos, conjunto de vértices/elementos, que possuem localização espacial fixa e conhecida, e arestas/ conexões, que apresentam as dependências espaciais (fluxo).
- O Google Colaboratory é um ambiente de notebooks Jupyter que não requer configuração e é executado na nuvem. Através do Google Colab é possível executar códigos em Python de forma gratuita, bem como compartilhá-lo e salvar as saídas geradas.
- Para acessar o código usado neste tutorial clique aqui:  
[https://colab.research.google.com/drive/1\\_eqoUxyhVWrvWdWnnvoXV3ScfNKi9If8?usp=sharing](https://colab.research.google.com/drive/1_eqoUxyhVWrvWdWnnvoXV3ScfNKi9If8?usp=sharing)

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## PARTE 1 – Preparação do dado de entrada:

- Usaremos como dado de entrada dois arquivos .txt:
  - 1) Para o primeiro arquivo, deve-se gerar os centroides a partir dos polígonos (regiões de estudo. Ex: Municípios.) A partir desses centroides, calcula-se suas respectivas latitude e longitude. Essas informações devem ser dispostas da seguinte forma em um documento do bloco de notas: 1ª Coluna: ID ( identificação do centroide com um numero inteiro); 2ª Coluna: Latitude (respectiva latitude do centroide identificado); 3ª Coluna: Longitude (respectiva longitude do centroide identificado).
  - 2) Para o segundo arquivo, precisa-se dos dados de fluxo entre esses centroides, ou seja, o fluxo entre as regiões de estudo em questão. Esses dados devem ser dispostos no formato de matriz com dimensão proporcional ao número de centroides.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## PARTE 1 – Observações importantes:

```
id_long_lat - Bloco de ...
Arquivo Editar Formatar Exibir Ajuda
id long lat
1 1 5
2 2 5
3 2 3
4 3 2
5 5 1
Ln 100% Windows (CRLF) UTF-8
```

```
matriz_teste - Bloco de ...
Arquivo Editar Formatar Exibir Ajuda
5 9 8 0 0
0 6 8 0 0
0 0 7 7 0
0 0 0 8 6
0 0 0 0 9
Ln 100% Windows (CRLF) UTF-8
```

Os elementos  $a_{i,j}$  em que  $i = j$  devem conter o fluxo que existe dentro daquele centroide, ou seja, o fluxo interno da região.

À exemplo, usou-se números inteiros. Para coordenadas reais usar "." para separar as casas decimais. **O uso de "," dá erro ao executar o programa!**

Os elementos  $a_{1,2}$  e  $a_{2,1}$  representam a mesma informação, ou seja, o fluxo entre o centroide número 1 e o centroide número 2. Sendo assim, só se faz necessário preencher um desses dois elementos, por isso, teremos uma matriz triangular, com zeros abaixo da diagonal principal.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Execução do código no google colab:

- Ao clicar no link disponibilizado no slide 4, aparecerá a seguinte tela:

A screenshot of a Google Colab notebook interface. The top bar shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and the date 'Last edited on May 18'. The left sidebar contains a 'Table of contents' with sections like 'Simetrizando matriz' and 'Section'. The main area displays two code cells. The first cell runs 'pip install python-igraph' and shows output indicating that requirements are already satisfied. The second cell runs 'pip install geopandas' and shows a long list of requirements already satisfied, including shapely, pandas, fiona, pyproj, numpy, pytz, python-dateutil, cligj, click, attrs, six, click-plugins, and munch. The third cell shows the beginning of a code block for plotting with matplotlib, including imports for geopandas, pandas, igraph, and numpy.

```
pip install python-igraph

Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)

[ ] pip install geopandas

Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)

[ ] %matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
import pandas as pd
from igraph import Graph
import numpy as np
```

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

CO SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on May 18

RAM [ ] Disk [ ] Editing [ ]

Files [ ] Upload Refresh Mount Drive

sample\_data

```
[ ] pip install python-igraph
```

```
Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)
```

```
[ ] pip install geopandas
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)  
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)  
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)  
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)  
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)  
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.  
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201  
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop  
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)  
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)  
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)  
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)  
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)  
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)
```

```
[ ] %matplotlib inline  
## It is to avoiding to open an extra window showing maps  
import geopandas as gpd  
import pandas as pd  
from igraph import Graph  
import numpy as np
```

Disk [ ] 76.72 GB available

Ao clicar neste ícone, aparecerá as opções acima. Clique em "Upload" para carregar as entradas do código.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Upload > arquivo .txt matriz de fluxo + arquivo .txt lat e long dos pontos.

The screenshot displays a Jupyter Notebook environment. The top bar shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and standard menu options. On the left, a file browser shows a folder named 'sample\_data'. The main area is divided into a code editor and a terminal window. The code editor contains the following Python code:

```
[ ] %matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
```

The terminal window shows the output of a file upload operation, listing the files 'matriz\_teste' and 'id\_long\_lat' with their modification dates and types. Below the terminal, a file explorer window is open, showing the 'Downloads' folder with the same two files selected. The file explorer window has a search bar and a list of files with columns for 'Nome', 'Data de modificação', and 'Tipo'. The file names 'id\_long\_lat' and 'matriz\_teste' are visible in the search bar and the file list.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Clicar no símbolo de “play” que está indicado neste slide com o quadrado vermelho.

SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on May 18

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

```
pip install python-igraph
```

```
Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)
```

```
pip install geopandas
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)  
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)  
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)  
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)  
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)  
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.  
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201  
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop  
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)  
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)  
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)  
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)  
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)  
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)
```

```
%matplotlib inline  
## It is to avoiding to open an extra window showing maps  
import geopandas as gpd
```

Disk 76.72 GB available

matriz\_teste.txt id\_long\_lat.txt

Exibir todos

Caso o upload seja feito  
com sucesso, os arquivos  
aparecerão aqui.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Clicar em “run anyway”.

A screenshot of a Jupyter Notebook interface. The notebook title is 'SaveShapefilesFromLatLongFile.ipynb'. The left sidebar shows a file explorer with folders and files like 'sample\_data', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area shows a code cell with the command 'pip install python-igraph'. A warning dialog box is overlaid on the code cell, with the text: 'Warning: This notebook was not authored by Google. This notebook was authored by geographsmail@gmail.com. It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook. Please contact the creator of this notebook at geographsmail@gmail.com with any additional questions.' The dialog box has two buttons: 'CANCEL' and 'RUN ANYWAY'. The background shows the output of the pip command, listing various installed packages and their versions.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Cada etapa levará alguns poucos segundos para ser executada.

A screenshot of a Jupyter Notebook interface. The top bar shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and various icons for file management and sharing. The left sidebar displays a file explorer with folders like 'sample\_data' and files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area shows a code cell with the command 'pip install python-igraph' highlighted by a red box. Below it, the terminal output shows the installation progress for 'python-igraph' and 'texttable', including download URLs and progress bars. A second code cell shows 'pip install geopandas', followed by a list of requirements already satisfied, such as 'geopandas', 'shapely', 'pandas', 'fiona', 'pyproj', 'numpy', 'pytz', 'python-dateutil', 'cligj', 'attrs', 'six', 'click-plugins', and 'munch'. The bottom status bar indicates '76.72 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Assim que finalizar o símbolo de “play” aparecerá novamente.

The screenshot shows a Jupyter Notebook interface. The top bar includes the Colab logo, the notebook title 'SaveShapefilesFromLatLongFile.ipynb', and a star icon. Below the title is a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the top bar are icons for 'Comment', 'Share', and a user profile. The left sidebar shows a file explorer with 'sample\_data', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area contains two code cells. The first cell has a play button icon and contains the command 'pip install python-igraph'. The output shows the installation progress for 'python-igraph' and 'texttable'. The second cell has a square icon with brackets and contains the command 'pip install geopandas'. The output lists various requirements for 'geopandas' that are already satisfied. At the bottom left, a disk usage indicator shows '76.71 GB available'.

Files

- Upload
- Refresh
- Mount Drive
- ..
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

```
pip install python-igraph
```

```
Collecting python-igraph
  Downloading https://files.pythonhosted.org/packages/8b/74/24a1afbf3abaf1d5f393b668192888d04091d1a6d106319661cd4af05406/py
  3.2MB 9.5MB/s
Collecting texttable>=1.6.2
  Downloading https://files.pythonhosted.org/packages/ec/b1/8a1c659ce288bf771d5b1c7cae318ada466f73bd0e16df8d86f27a2a3ee7/te
Installing collected packages: texttable, python-igraph
Successfully installed python-igraph-0.8.2 texttable-1.6.2
```

```
[ ] pip install geopandas
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)
```

Disk 76.71 GB available

Quando a primeira etapa for concluída, clicar no próximo colchetes.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook titled "SaveShapefilesFromLatLongFile.ipynb". The left sidebar shows a file explorer with a folder named "sample\_data" containing "id\_long\_lat.txt" and "matriz\_teste.txt". The main area shows two code cells. The first cell, labeled "[1]", contains the command "pip install python-igraph" and shows the output of the installation process, including package collection, downloading, and successful installation of python-igraph-0.8.2 and texttable-1.6.2. The second cell, labeled "...", contains the command "pip install geopandas" and shows the output of its installation, including the collection and downloading of geopandas, pyproj, and fiona, along with their respective versions and requirements.

A red-bordered box contains the following text: "Ao clicar no próximo colchetes, aparecerá um número no colchetes anterior em que a execução foi finalizada. Esses números aparecerão em sequência até o fim do código." A blue arrow points from this box to the "[1]" label in the first code cell.

RAM: [Progress Bar] Disk: [Progress Bar] Editing [Icon]

Disk [Progress Bar] 76.61 GB available



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook titled "SaveShapefilesFromLatLongFile.ipynb". The left sidebar shows a file explorer with a folder named "sample\_data" containing "id\_long\_lat.txt" and "matriz\_teste.txt". The main area displays a code cell with the following code:

```
Successfully installed click-plugins-1.1.1 cligj-0.5.0 fiona-1.8.13.post1 geopandas-0.7.0 munch-2.5.0 pyproj-2.6.1.post1

%matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
import pandas as pd
from igragh import Graph
import numpy as np
from statistics import mean
import math

[ ] filename = 'id_long_lat'
    sep = ','
    try:
        codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
    except:
        codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
    codes
```

The output of the code cell is a table:

	id	long	lat
0	1	-46.446433	-23.666145
1	2	-46.566063	-23.626141
2	3	-46.350597	-23.761635

A red box highlights the instruction: "Clicar no colchetes seguinte e repetir isso até o fim do código." A blue arrow points from this box to the play button above the code cell.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer shows a folder named 'sample\_data' containing files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The code editor shows the following code:

```
[3] %matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
import pandas as pd
from igraph import Graph
import numpy as np
from statistics import mean
import math

filename = 'id_long_lat'
sep = ','
try:
    codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
except:
    codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
codes
```

The output of the code is a table with 3 rows and 4 columns:

	id	long	lat
0	1	1	5
1	2	2	5
2	3	2	3

A red box highlights the text: "Perceba que o número 3 apareceu no terceiro colchetes. Atente-se para não pular nenhum colchetes." A blue arrow points from this box to the third code cell in the notebook.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook titled "SaveShapefilesFromLatLongFile.ipynb". The interface includes a file explorer on the left, a code editor in the center, and a console at the bottom. The code in the cell is as follows:

```
[3] from statistics import mean
import math

filename = 'id_long_lat'
sep = ','
try:
    codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
except:
    codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
codes
```

The output of the code is a table with 5 rows and 4 columns:

	id	long	lat
0	1	1	5
1	2	2	5
2	3	2	3
3	4	3	2
4	5	5	1

The console below the code shows the next cell's code:

```
[ ] from shapely.geometry import Point
geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

A blue arrow points from the "play" button in the code editor to a red-bordered box containing the following text:

Ao executar o quarto colchetes, será possível visualizar a entrada com os pontos e suas respectivas latitude e longitude.

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

+ Code + Text

```
[4] 0 1 1 5
    1 2 2 5
    2 3 2 3
    3 4 3 2
    4 5 5 1
```

```
from shapely.geometry import Point
geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

```
[ ] crs = {'proj': 'latlong', 'ellps': 'WGS84', 'datum': 'WGS84', 'no_defs': True}

geo_data = gpd.GeoDataFrame(codes, crs = crs, geometry = geometry)
geo_data
```

	id	long	lat	geometry
0	1	-46.446433	-23.666145	POINT (-46.44643 -23.66615)
1	2	-46.566063	-23.626141	POINT (-46.56606 -23.62614)
2	3	-46.350597	-23.761635	POINT (-46.35060 -23.76164)
3	4	-46.550792	-23.812989	POINT (-46.55079 -23.81299)

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Saving...

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

```
[5] from shapely.geometry import Point
     geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

▶ crs = {'proj': 'latlong', 'ellps': 'WGS84', 'datum': 'WGS84', 'no\_defs': True}

```
geo_data = gpd.GeoDataFrame(codes, crs = crs, geometry = geometry)
geo_data
```

	id	long	lat	geometry
0	1	1	5	POINT (1.00000 5.00000)
1	2	2	5	POINT (2.00000 5.00000)
2	3	2	3	POINT (2.00000 3.00000)
3	4	3	2	POINT (3.00000 2.00000)
4	5	5	1	POINT (5.00000 1.00000)

```
[ ] import os

     dir = './'
     if not os.path.exists(dir):
         os.makedirs(dir)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. The top bar shows the notebook title "SaveShapefilesFromLatLongFile.ipynb" and various utility icons. The left sidebar contains a file explorer with a folder named "sample\_data" and files "id\_long\_lat.txt" and "matriz\_teste.txt". The main area shows a code cell with the following Python code:

```
import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)

geo_data.to_file(dir + 'CENTROIDES.shp')

geo_data.plot(figsize=(15,8), alpha=0.5)
```

Below the code, the output shows a scatter plot with two blue circular points. The y-axis is labeled from 3.5 to 5.0. The first point is at approximately (3.0, 5.0) and the second is at approximately (5.0, 5.0). The plot title is "<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4569942898>".

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface. On the left, a file browser displays a folder named 'sample\_data' containing two files: 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area shows a code cell with a play button icon and the text: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4569942898>`. Below the code is a scatter plot with five blue circular points. The x-axis ranges from 1.0 to 5.0, and the y-axis ranges from 1.0 to 5.0. The points are located at approximately (1.0, 5.0), (2.0, 5.0), (2.0, 3.0), (3.0, 2.0), and (5.0, 1.0). A blue arrow points from the plot to a red-bordered text box on the right.

Ao executar o sétimo colchetes, será possível visualizar o shape com os pontos.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface. The top bar includes the Colab logo, the file name 'SaveShapefilesFromLatLongFile.ipynb', and a star icon. Below the top bar is a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the top bar, there are icons for 'Comment', 'Share', and a user profile picture. Below the top bar, there are indicators for 'RAM' and 'Disk' usage, and a status 'Editing'. The left sidebar shows a file explorer with a folder named 'sample\_data' containing files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area contains a code cell with a play button icon on the left. The code in the cell is:

```
filename = 'matriz_teste'  
sep=','  
try:  
    mtx=pd.read_csv(filename+'.txt', delim_whitespace=True, header= None)  
except:  
    mtx=pd.read_csv(filename+'.csv', delimiter=sep, header= None)  
mtx
```

The output of the code cell is a 5x5 matrix:

	0	1	2	3	4
0	5	9	8	0	0
1	0	6	8	0	0
2	0	0	7	7	0
3	0	0	0	8	6
4	0	0	0	0	9

Below the matrix, the output shows the type of the variable 'mtx' as 'pandas.core.frame.DataFrame'. At the bottom left, there is a disk usage indicator showing '76.61 GB available'. A red box with a blue border contains the text: 'Ao executar o oitavo colchetes, será possível visualizar a matriz de fluxo de entrada.' A blue arrow points from this box to the play button icon in the code cell.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

Upload Refresh Mount Drive

sample\_data  
id\_long\_lat.txt  
matriz\_teste.txt

+ Code + Text

```
[8] 1 0 6 8 0 0  
    2 0 0 7 7 0  
    3 0 0 0 8 6  
    4 0 0 0 0 9
```

type(mtx)

```
pandas.core.frame.DataFrame
```

## Simetrizando matriz

```
[ ] def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help



Comment



Share



Editing



Files

Upload Refresh Mount Drive

<>

..

sample\_data

CENTROIDES.cpg

CENTROIDES.dbf

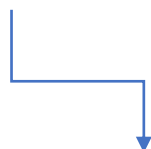
CENTROIDES.prj

CENTROIDES.shp

CENTROIDES.shx

id\_long\_lat.txt

matriz\_teste.txt



Perceba que o shape  
dos centroides passa  
a aparecer aqui.

Disk 76.61 GB available

+ Code + Text

pandas.core.frame.DataFrame

## Simetrizando matriz

```
def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)  
    mtx
```

```
0 1 2 3 4  
0 5 9 8 0 0  
1 9 6 8 0 0  
2 8 8 7 7 0  
3 0 0 7 8 6
```

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

SaveShapefilesFromLatLongFile.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Última edição em 1 de julho

Comentário Compartilhar

Índice

- Symmetrizing the matrix
  - Creating graph from adj matrix
    - Seção

```
[ ] #Symmetrizing by sum
def simetric_matrix_sum(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = original.iloc[i,j] + original.iloc[j,i]
    return dados

#Symmetrizing by average
def simetric_matrix_avg(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = (original.iloc[i,j] + original.iloc[j,i])/2
    return dados

[ ] #You have to choose which function you would like to apply

#mtx = simetric_matrix_higher(mtx)
#mtx = simetric_matrix_lower(mtx)
mtx = simetric_matrix_sum(mtx)
#mtx = simetric_matrix_avg(mtx)
```

Nesta parte, será definido a forma com que a matriz será simetrizada. As opções são simetrização pelo maior, menor, soma ou média dos dois valores. É necessário “descomentar”, ou seja, tirar o símbolo ‘#’ da linha que deve ser executada, mantendo as demais com o símbolo ‘#’.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** A sidebar on the left showing a file explorer with a folder named 'sample\_data' containing files: 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'.
- Code Editor:** The main area contains a code cell with the following Python code:

```
[10] def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```
- Execution:** Below the code, a play button is followed by the code: `mtx = simetric_matrix(mtx)` and `mtx`.
- Output:** The execution result is a 5x5 matrix displayed as follows:

	0	1	2	3	4
0	5	9	8	0	0
1	9	6	8	0	0
2	8	8	7	7	0
3	0	0	7	8	6
4	0	0	0	6	9
- System Status:** At the bottom left, a disk usage indicator shows '76.61 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

A screenshot of a Jupyter Notebook interface. The top bar shows the file name 'SaveShapefilesFromLatLongFile.ipynb' and various utility icons. The left sidebar displays a file explorer with a folder named 'sample\_data' containing several files: 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area is titled 'Creating graph from adj matrix' and contains three code cells. The first cell starts with 'from igraph import ADJ\_MAX' and creates a graph object. The second cell defines a function 'get\_lat\_long\_from\_city(id)'. The third cell creates a 'limits\_edges' DataFrame and populates it with data from the graph. A 'limits edges' prompt is visible at the bottom of the code area.

```
from igraph import ADJ_MAX

A=np.matrix(mtx.values)
graph = Graph.Weighted_Adjacency(A.tolist(), mode=ADJ_MAX, attr="weight", loops=False)
graph = graph.as_undirected()

[ ] def get_lat_long_from_city(id): #For each code return its geometry
    cidade = codes.iloc[id][0]
    idx = geo_data[geo_data['id']==cidade].index
    row = geo_data.iloc[idx]['geometry']

    return (row.total_bounds)

[ ] limits_edges = pd.DataFrame(columns=['source', 'target'])

for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
    'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
    'weight': edge['weight']},
    ignore_index=True)

limits edges
```

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer shows a folder named 'sample\_data' containing files: 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The code editor displays the following code:

```
[12] from igraph import ADJ_MAX

A=np.matrix(mtx.values)
graph = Graph.Weighted_Adjacency(A.tolist(), mode=ADJ_MAX, attr="weight", loops=False)
graph = graph.as_undirected()

def get_lat_long_from_city(id): #For each code return its geometry
    cidade = codes.iloc[id][0]
    idx = geo_data[geo_data['id']==cidade].index
    row = geo_data.iloc[idx]['geometry']

    return (row.total_bounds)

[ ] limits_edges = pd.DataFrame(columns=['source', 'target'])

for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
    'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
    'weight': edge['weight']},
    ignore_index=True)

limits_edges
```

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Top Bar:** Includes the Colab logo, the notebook title "SaveShapefilesFromLatLongFile.ipynb", and a star icon. On the right, there are buttons for "Comment", "Share", and a user profile picture. Below these are indicators for "RAM" and "Disk" usage, and a status "Editing".
- Files Panel (Left):** Shows a file explorer with a folder named "sample\_data" containing files: "CENTROIDES.cpg", "CENTROIDES.dbf", "CENTROIDES.prj", "CENTROIDES.shp", "CENTROIDES.shx", "id\_long\_lat.txt", and "matriz\_teste.txt".
- Code Editor (Center):** Contains two code cells under the heading "Creating graph from adj matrix".
  - Cell [12]:** Imports `igraph` and `ADJ_MAX`, then creates a graph from a matrix `A` using `Graph.Weighted_Adjacency` and `graph.as_undirected()`.
  - Cell [13]:** Defines a function `get_lat_long_from_city(id)` that returns the geometry of a city based on its ID.
- Execution:** A play button is visible on the left of the third code cell, which begins with `limits_edges = pd.DataFrame(columns=['source', 'target'])`.
- Bottom Bar:** Shows a "Disk" usage indicator with "76.61 GB available".



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[13] return (row.total_bounds)
```

limits\_edges = pd.DataFrame(columns=['source', 'target'])

```
for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
                                        'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
                                        'weight': edge['weight']},
                                       ignore_index=True)
```

limits\_edges

	source	target	weight
0	(1.0, 5.0)	(2.0, 5.0)	9.0
1	(1.0, 5.0)	(2.0, 3.0)	8.0
2	(2.0, 5.0)	(2.0, 3.0)	8.0
3	(2.0, 3.0)	(3.0, 2.0)	7.0
4	(3.0, 2.0)	(5.0, 1.0)	6.0

```
[ ] from shapely.geometry import Point, LineString

lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

Disk 76.61 GB available



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)



Comment



Share



Files



Upload Refresh Mount Drive



- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

+ Code + Text

RAM  Disk

Editing

[14]

```
4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
from shapely.geometry import Point, LineString
```

```
lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

```
[ ] edges_graph = gpd.GeoDataFrame(limits_edges, crs=geo_data.crs, geometry = lines)
```

```
[ ] edges_graph.crs
```

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>
```

Name: unknown

Axis Info [ellipsoidal]:

- lon[east]: Longitude (degree)

- lat[north]: Latitude (degree)

Area of Use:

- undefined

Datum: World Geodetic System 1984

- Ellipsoid: WGS 84

- Prime Meridian: Greenwich

```
[ ] edges_graph
```



source

target weight

geometry

Disk  76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings User

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[14] 4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
[15] from shapely.geometry import Point, LineString

lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

edges\_graph = gpd.GeoDataFrame(limits\_edges, crs=geo\_data.crs, geometry = lines)

```
[ ] edges_graph.crs
```

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>
Name: unknown
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- undefined
Datum: World Geodetic System 1984
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

```
[ ] edges_graph
```

source	target	weight	geometry
--------	--------	--------	----------

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment

Share



RAM  
Disk

Editing

Files

Upload Refresh Mount Drive

..  
sample\_data  
CENTROIDES.cpg  
CENTROIDES.dbf  
CENTROIDES.prj  
CENTROIDES.shp  
CENTROIDES.shx  
id\_long\_lat.txt  
matriz\_teste.txt

+ Code + Text

```
[14] 4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
[15] from shapely.geometry import Point, LineString  
lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

```
[16] edges_graph = gpd.GeoDataFrame(limits_edges, crs=geo_data.crs, geometry = lines)
```

edges\_graph.crs

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>  
Name: unknown  
Axis Info [ellipsoidal]:  
- lon[east]: Longitude (degree)  
- lat[north]: Latitude (degree)  
Area of Use:  
- undefined  
Datum: World Geodetic System 1984  
- Ellipsoid: WGS 84  
- Prime Meridian: Greenwich
```

edges\_graph



source

target weight

geometry

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** A sidebar on the left showing a file explorer with a folder named 'sample\_data' containing files: CENTROIDES.cpg, CENTROIDES.dbf, CENTROIDES.prj, CENTROIDES.shp, CENTROIDES.shx, id\_long\_lat.txt, and matriz\_teste.txt.
- Code Editor:** The main area contains code cells. The first cell shows metadata for a coordinate system: `[17] - undefined`, `Datum: World Geodetic System 1984`, `- Ellipsoid: WGS 84`, and `- Prime Meridian: Greenwich`. The second cell displays a graph structure as a table with columns 'source', 'target', 'weight', and 'geometry'. The third cell shows the execution of `edges_graph = edges_graph[['geometry']]` and `type(edges_graph)`, resulting in `geopandas.geodataframe.GeoDataFrame`. The fourth cell shows `import os` and `dir = './'`.
- UI Elements:** The top right includes 'Comment', 'Share', and 'Editing' buttons. A status bar at the bottom left shows 'Disk 76.61 GB available'.

	source	target	weight	geometry
0	(1.0, 5.0)	(2.0, 5.0)	9.0	LINSTRING (1.00000 5.00000, 2.00000 5.00000)
1	(1.0, 5.0)	(2.0, 3.0)	8.0	LINSTRING (1.00000 5.00000, 2.00000 3.00000)
2	(2.0, 5.0)	(2.0, 3.0)	8.0	LINSTRING (2.00000 5.00000, 2.00000 3.00000)
3	(2.0, 3.0)	(3.0, 2.0)	7.0	LINSTRING (2.00000 3.00000, 3.00000 2.00000)
4	(3.0, 2.0)	(5.0, 1.0)	6.0	LINSTRING (3.00000 2.00000, 5.00000 1.00000)

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

+ Code + Text

- Prime Meridian: Greenwich

[18] edges\_graph

	source	target	weight	geometry
0	(1.0, 5.0)	(2.0, 5.0)	9.0	LINESTRING (1.00000 5.00000, 2.00000 5.00000)
1	(1.0, 5.0)	(2.0, 3.0)	8.0	LINESTRING (1.00000 5.00000, 2.00000 3.00000)
2	(2.0, 5.0)	(2.0, 3.0)	8.0	LINESTRING (2.00000 5.00000, 2.00000 3.00000)
3	(2.0, 3.0)	(3.0, 2.0)	7.0	LINESTRING (2.00000 3.00000, 3.00000 2.00000)
4	(3.0, 2.0)	(5.0, 1.0)	6.0	LINESTRING (3.00000 2.00000, 5.00000 1.00000)

edges\_graph = edges\_graph[['geometry']]  
type(edges\_graph)

geopandas.geodataframe.GeoDataFrame

```
[ ] import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

co SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[19] edges_graph = edges_graph[['geometry']]
      type(edges_graph)

geopandas.geodataframe.GeoDataFrame
```

```
import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)

edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')
```

```
[ ] centroides = gpd.read_file('CENTROIDES.shp')
      edges = gpd.read_file('EDGES_SHAPEFILE.shp')

base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f8e49acb198>

A small plot window showing a blue dot at the top left and a red line segment extending downwards and to the right. The y-axis is labeled with values -23.625 and -23.650.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb

File Edit View Insert Runtime Tools Help

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[20] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')
```

```
centroides = gpd.read_file('CENTROIDES.shp')  
edges = gpd.read_file('EDGES_SHAPEFILE.shp')
```

```
base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)  
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4566b59128>

Aqui será possível visualizar o shape com as arestas que representam o fluxo entre os pontos.

Disk 76.61 GB available



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 7:00 PM

Comment Share Settings Profile

Files

Upload Refresh Mount Drive

<>

- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - EDGES\_SHAPEFILE.cpg
  - EDGES\_SHAPEFILE.dbf
  - EDGES\_SHAPEFILE.prj
  - EDGES\_SHAPEFILE.shp
  - EDGES\_SHAPEFILE.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

+ Code + Text

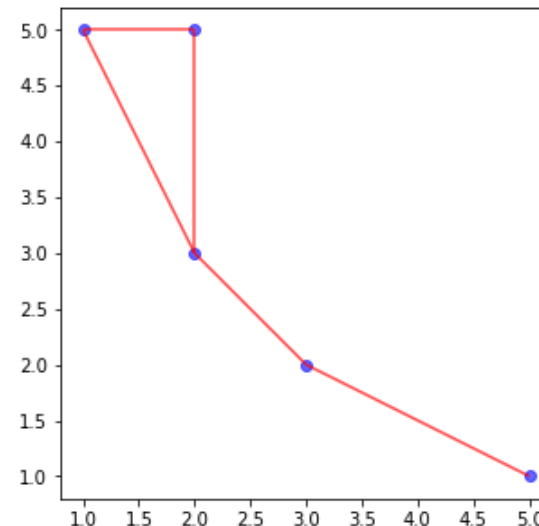
RAM Disk Editing

```
[ ] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')

[ ] centroides = gpd.read_file('CENTROIDES.shp')
edges = gpd.read_file('EDGES_SHAPEFILE.shp')

base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4566b59128>



Após alguns segundos os arquivos do shape de fluxo aparecerão abaixo dos arquivos do shape de pontos. Caso isso não aconteça automaticamente, basta atualizar a página!

Disk 76.61 GB available

Navigation icons



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO

- Após finalizar a execução do código, basta fazer o download dos arquivos.



The screenshot shows a Jupyter Notebook interface with a file browser on the left and a code editor on the right. The file browser displays a directory named 'sample\_data' containing various files including 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'EDGES\_SHAPEFILE.cpg', 'EDGES\_SHAPEFILE.dbf', 'EDGES\_SHAPEFILE.prj', 'EDGES\_SHAPEFILE.shp', 'EDGES\_SHAPEFILE.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. A context menu is open over the 'CENTROIDES.cpg' file, showing options: Download, Delete file, Rename file, Copy path, and Refresh. The code editor shows the following code:

```
[ ] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')

[ ] centroides = gpd.read_file('CENTROIDES.shp')
edges = gpd.read_file('EDGES_SHAPEFILE.shp')

edges.plot(color='blue', figsize=(10,5), alpha=0.6)
edges.plot(color='red', figsize=(10,5), color='red', alpha=0.7)
```

Below the code, a plot is displayed showing a red line connecting four points. The x-axis ranges from 1.0 to 5.0, and the y-axis ranges from 1.0 to 4.0. The points are approximately at (1.5, 4.0), (2.0, 3.0), (3.0, 2.0), and (5.0, 1.0). The status bar at the bottom indicates 'Disk 76.61 GB available'.

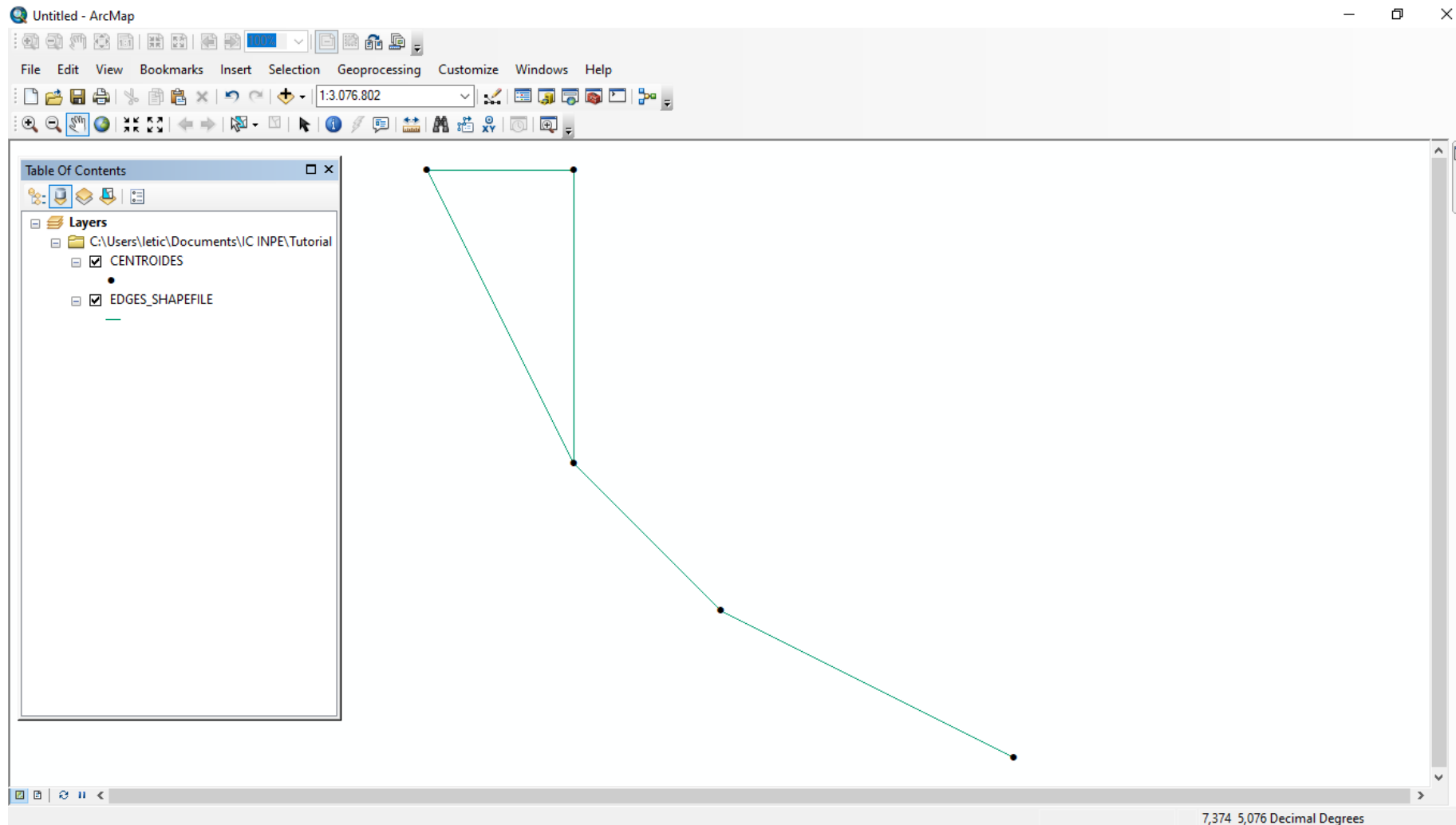
Para fazer o download dos arquivos, basta sobrepor o cursor em cima de cada um deles, clicar nos três pontinhos que aparecerão e escolher a opção de realizar o download.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 3 – Entendendo o dado de saída:



Após realizar o download dos shapes e adicioná-los no Arcgis, será possível observar os centroides e as arestas que os ligam baseado no fluxo de mobilidade. Assim, tem-se o (geo)grafo.



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



Laboratório de investigação  
em Sistemas Sócio-Ambientais

# TUTORIAL PARA GERAÇÃO DE DADOS PARA ANÁLISE DE (GEO)GRAFO A PARTIR DE CÓDIGO EM PYTHON NO GOOGLE COLAB

LISS - INPE

2020

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Este tutorial é produto de iniciação científica.

Elaboração do estudo: **LISS** – Laboratório de Investigação em Sistemas **S**ócio-Ambientais

Autor: Leticia da Silva Cabral

Orientador: Dr. Antonio Miguel Vieira Monteiro – LISS/INPE

Coorientador: Leonardo Bacelar de Lima Santos – CEMADEN

Colaboradores: Cátia Souza do Nascimento Sepetauskas e Jeferson Feitosa Mendes

CABRAL, L. S. ; MONTEIRO, A. M. V. ; SANTOS, L. B. L. ; MENDES, J. F. ; NASCIMENTO, C. S.



## APRESENTAÇÃO

Neste tutorial você aprenderá a:

Preparar os  
dados de  
entrada do  
código

Executar código  
no google colab

Entender o dado  
de saída

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- O Google Colaboratory é um ambiente de notebooks Jupyter que não requer configuração e é executado na nuvem. Através do Google Colab é possível executar códigos em Python de forma gratuita, bem como compartilhá-lo e salvar as saídas geradas.
- Para acessar o código usado neste tutorial clique aqui:  
<https://colab.research.google.com/drive/11g2Er1Oj1xvndbS3QadH93QvGyUBqog8?usp=sharing>

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Preparação do dado de entrada:

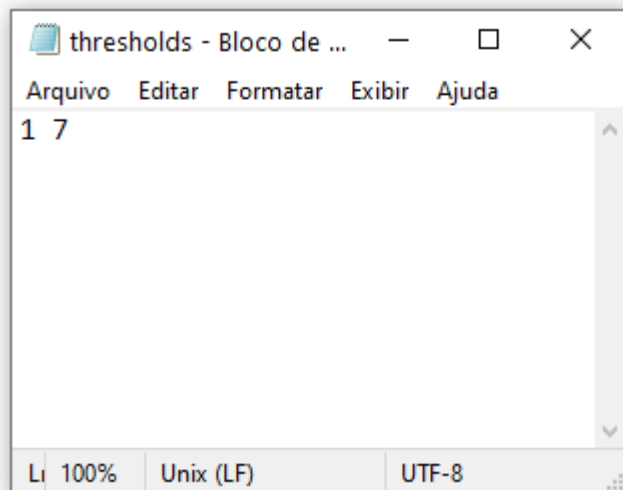
- Usaremos um arquivo .txt e um arquivo .csv:
  1. O arquivo .txt deve conter os limiares, que serão usados como um filtro para a matriz de fluxo. Esse limiar será o parâmetro usado para ver se um determinado valor de fluxo é forte o bastante para aparecer no grafo. Se o valor for menor que o limiar, ele será mostrado como 0 e se for maior continuará com o seu valor normal no dado de saída.
  2. O arquivo .cvs deve conter a matriz de fluxo, com os valores de fluxo entre as regiões de estudo.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Preparação do dado de entrada:



Perceba que colocou-se 2 limiares, portanto  
teremos dados de saída diferentes para  
cada um deles.

flow\_matrix - Excel

	A	B	C	D	E	F
1	0	9	8	0	0	
2	0	0	8	0	0	
3	0	0	0	7	0	
4	0	0	0	0	6	
5	0	0	0	0	0	
6						
7						
8						
9						
10						
11						
12						
13						



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO

## PARTE 2 – Execução do código no google colab:

- Ao clicar no link, aparecerá a seguinte tela:



Laboratório de investigação  
em Sistemas Sócio-Ambientais

The screenshot shows a Google Colab notebook titled "Leonardo.ipynb". The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help" menus. A "Table of contents" sidebar on the left lists sections: "Imports", "Lendo arquivo de limiares", "Lendo matriz de fluxo quadrada", "Simetrizando matriz e zerando a diagonal", "Criando matrizes de adjacencia por limiares", "Criando a classe GraphMaking e calculando suas metricas", and a "Section" button. The main workspace shows a code cell with the command `pip install python-igraph` and its output: "Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)" and "Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)". Below this, the notebook content is organized into sections: "Imports" with code for `import numpy as np`, `import pandas as pd`, `from igraph import Graph`, `from statistics import mean`, and `import math`; and "Lendo arquivo de limiares" with code for `filename = 'thresholds'`, `sep = ';'` , and `pd.read_csv(filename+'.txt', delim_whitespace=True, header=None, encoding="ISO-8859-1")`.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Leonardo.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 25

Comment Share

RAM Disk

Editing

Files

Upload Refresh Mount Drive

sample\_data

```
pip install python-igraph
```

Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)

Imports

```
[ ] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```

Lendo arquivo de limiares

```
[ ] filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header=None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")
thresholds
```

Disk 76.72 GB available

Ao clicar neste ícone, aparecerá as opções acima. Clique em "Upload" para carregar as entradas do código.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



- Upload > arquivo .txt + arquivo excel

The image is a screenshot of a Jupyter Notebook interface. In the background, the notebook's file browser shows a folder named 'sample\_data'. A Windows-style file dialog box titled 'Abrir' is open, displaying the 'Downloads' folder. It lists two files: 'thresholds' (a document) and 'flow\_matrix' (an Excel file), both dated 27/05/2020 23:48. The 'flow\_matrix' file is selected. Below the dialog, a code cell contains Python code for reading CSV files. The code uses 'pd.read\_csv' with 'delim\_whitespace=True' for the 'thresholds' file and a custom separator for the 'flow\_matrix' file. The code is as follows:

```
[ ] filename = 'thresholds'  
sep = ';'   
try:   
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")   
except:   
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")   
thresholds
```

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



- Clicar no símbolo de “play” que está indicado neste slide com o quadrado vermelho.

The screenshot shows a Jupyter Notebook interface. On the left, a 'Files' sidebar displays a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. A blue arrow points from this directory to a red-bordered text box. The main notebook area shows a code cell with a red square around the play button icon. The output of the code cell shows that 'python-igraph' and 'texttable' are already installed. Below this, there are sections for 'Imports' and 'Lendo arquivo de limiares' with their respective code blocks.

Caso o upload seja feito com sucesso, os arquivos aparecerão aqui.

```
pip install python-igraph
```

Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)

### Imports

```
[ ] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```

### Lendo arquivo de limiares

```
[ ] filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")
thresholds
```

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Clicar em “run anyway”.

A screenshot of a Jupyter Notebook interface. The top bar shows the user 'Leonardo.ipynb' and various menu options like 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The left sidebar displays a file explorer with a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. The main area shows a code cell with the command 'pip install python-igraph'. Below the code, a message indicates that requirements are already satisfied. A white warning dialog box is overlaid on the notebook, stating: 'Warning: This notebook was not authored by Google. This notebook was authored by souz.kti@gmail.com. It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook. Please contact the creator of this notebook at souz.kti@gmail.com with any additional questions.' The dialog box has two buttons: 'CANCEL' and 'RUN ANYWAY'. The bottom of the notebook shows a code cell with Python code for reading a CSV file.

```
[ ] filename = 'thresholds'  
    sep = ';'   
    try:   
        thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")   
    except:   
        thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")   
    thresholds
```



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Assim que finalizar o símbolo de “play” aparecerá novamente.

The screenshot shows a Jupyter Notebook interface. On the left, a file browser displays a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. A blue arrow points from the 'sample\_data' folder to a text box. The main area shows a code cell with the command `pip install python-igraph` and its output, which includes the installation of 'python-igraph-0.8.2' and 'texttable-1.6.2'. Below this, an 'Imports' section shows the following code: `import numpy as np`, `import pandas as pd`, `from igraph import Graph`, `from statistics import mean`, and `import math`. At the bottom, a code cell shows `filename = 'thresholds'` and `sep = ';'` . The interface includes a top menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A status bar at the bottom indicates 'Disk 76.71 GB available'.

Ao clicar no próximo colchetes, aparecerá um número no colchetes anterior em que a execução foi finalizada. Esses números aparecerão em sequência até o fim do código.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** Shows a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'.
- Imports:** A code cell with the following imports:

```
[2] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```
- Lendo arquivo de limiares:** A code cell with the following code:

```
filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")
thresholds
```

The output of this cell is a 2x2 matrix:

```
0 1
0 1 7
```
- Lendo matriz de fluxo quadrada:** The start of a new code cell.

A red box highlights the 'play' button above the second code cell, with a blue arrow pointing to it. The text inside the box reads: "Clicar no colchetes seguinte e repetir isso até o fim do código."



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** Shows a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. A blue arrow points from the 'thresholds.txt' file to the code cell below.
- Code Cell:** Contains Python code for reading CSV files with specific encoding and delimiter settings. The code is: 

```
thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")  
except:  
thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")  
thresholds
```
- Output:** Displays a 2x2 matrix: 

```
0 1  
0 1 7
```
- Code Cell:** Contains Python code for reading a square flow matrix. The code is: 

```
filename = 'flow_matrix'  
  
sep = ';'   
try:  
    mtx = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")  
except:  
    mtx = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")  
mtx
```
- Output:** Displays a 3x5 matrix: 

```
0 1 2 3 4  
0 0 9 8 0 0  
1 0 0 8 0 0  
2 0 0 0 7 0
```
- Annotations:** A red box highlights the text: "Perceba que o número 3 apareceu no terceiro colchetes. Atente-se para não pular nenhum colchetes." A blue arrow points from this box to the third closing bracket in the first code cell.
- UI Elements:** Includes a top navigation bar with 'File Edit View Insert Runtime Tools Help', a 'RAM Disk' indicator, and a bottom status bar showing '76.71 GB available'.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** 'Leonardo.ipynb' with a star icon, and a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are 'Comment', 'Share', and 'Editing' options, along with RAM and Disk usage indicators.
- Files Panel:** Shows a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'.
- Code Cell:** Titled 'Lendo matriz de fluxo quadrada', containing the following Python code:

```
filename = 'flow_matrix'  
  
sep = ';'   
try:  
    mtx = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")  
except:  
    mtx = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")  
mtx
```
- Output:** A table representing the flow matrix:

	0	1	2	3	4	
0	0	0	9	8	0	0
1	0	0	0	8	0	0
2	0	0	0	0	7	0
3	0	0	0	0	0	6
4	0	0	0	0	0	0
- Next Cell:** Titled 'Simetrizando matriz e zerando a diagonal'.
- Footer:** A disk usage indicator showing '76.71 GB available'.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
- flow\_matrix.csv
- thresholds.txt

+ Code + Text

```
4 0 0 0 0 0
```

Simetrizando matriz e zerando a diagonal

```
def simetric_matrix(dados):
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            if(i==j):
                dados.iloc[j,i]=0
            elif(dados.iloc[i,j]>dados.iloc[j,i]):
                dados.iloc[j,i] = dados.iloc[i,j]
            elif (dados.iloc[j,i]>dados.iloc[i,j]):
                dados.iloc[i,j] = dados.iloc[j,i]
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)
mtx
```

	0	1	2	3	4	
0	0	0	9	8	0	0
1	9	0	8	0	0	0
2	8	8	0	7	0	0
3	0	0	7	0	6	0

Disk 76.71 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

SaveShapefilesFromLatLongFile.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Última edição em 1 de julho

Comentário Compartilhar Configurações Perfil

Índice

- Symmetrizing the matrix
  - Creating graph from adj matrix
    - Seção

```
[ ] #Symmetrizing by sum
def simetric_matrix_sum(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = original.iloc[i,j] + original.iloc[j,i]
    return dados

#Symmetrizing by average
def simetric_matrix_avg(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = (original.iloc[i,j] + original.iloc[j,i])/2
    return dados

[ ] #You have to choose which function you would like to apply

#mtx = simetric_matrix_higher(mtx)
#mtx = simetric_matrix_lower(mtx)
mtx = simetric_matrix_sum(mtx)
#mtx = simetric_matrix_avg(mtx)
```

Nesta parte, será definido a forma com que a matriz será simetrizada. As opções são simetrização pelo maior, menor, soma ou média dos dois valores. É necessário “descomentar”, ou seja, tirar o símbolo ‘#’ da linha que deve ser executada, mantendo as demais com o símbolo ‘#’.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆  
File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - flow\_matrix.csv
  - thresholds.txt

+ Code + Text

### Simetrizando matriz e zerando a diagonal

```
[5] def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(i==j):  
                dados.iloc[j,i]=0  
            elif(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

▶ mtx = simetric\_matrix(mtx)  
mtx

	0	1	2	3	4	
0	0	0	9	8	0	0
1	9	0	8	0	0	
2	8	8	0	7	0	
3	0	0	7	0	6	
4	0	0	0	6	0	

Disk 76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings User Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - flow\_matrix.csv
  - thresholds.txt

### ▼ Criando matrizes de adjacencia por limiares

```
for index in thresholds:  
    limiar = thresholds[index][0]  
  
    for i in range(0, len(mtx.columns)):  
        mtx[i] = np.where((mtx[i] <= limiar), 0, mtx[i])  
  
    mtx.to_csv('adj_matrix'+str(limiar)+'.csv')
```

### ▼ Criando a classe GraphMaking e calculando suas metricas

```
[ ] class GraphMaking:  
  
    COLUMNS = [  
        'limiar',  
        'avg_betweenness_eccentricity',  
        'avg_path_length',  
        'avg_degree',  
        'diameter',  
        'transitivity',  
        'number_of_edges'  
    ]  
  
    COLUMNS_NODE = ['degree', 'betweenness', 'transitivity', 'shortest paths',
```

Disk 76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** Leonardo.ipynb, File Edit View Insert Runtime Tools Help, All changes saved, Comment, Share, Editing, RAM Disk, and a user profile icon.
- Files Panel:** A file explorer showing a directory named 'sample\_data' containing files: 'adj\_matrix1.csv', 'adj\_matrix7.csv', 'flow\_matrix.csv', and 'thresholds.txt'. A blue arrow points from this panel to a red-bordered text box.
- Code Cells:**
  - Cell 1:** Titled "Criando matrizes de adjacencia por limiares", containing a code block that iterates through thresholds and updates a matrix 'mtx' based on a 'limiar' value.
  - Cell 2:** Titled "Criando a classe GraphMaking e calculando suas metricas", containing a code block that defines a 'GraphMaking' class with a list of metrics: 'limiar', 'avg\_betweenness\_eccentricity', 'avg\_path\_length', 'avg\_degree', 'diameter', 'transitivity', and 'number\_of\_edges'.
- Footer:** A disk usage indicator showing 76.71 GB available.

Perceba que a primeira saída do código passa a aparecer aqui.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆  
File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - adj\_matrix1.csv
  - adj\_matrix7.csv
  - flow\_matrix.csv
  - thresholds.txt

+ Code + Text

▶ Criando a classe GraphMaking e calculando suas metricas

```
class GraphMaking:

    COLUMNS = [
        'limiar',
        'avg_betweenness_eccentricity',
        'avg_path_length',
        'avg_degree',
        'diameter',
        'transitivity',
        'number_of_edges'
    ]

    COLUMNS_NODE = ['degree', 'betweenness', 'transitivity', 'shortest_paths',
                    'limiar', 'diameter_graph', 'number_of_edges_graph',
                    'avg_degree', 'avg_betweenness_eccentricity',
                    'transitivity', 'avg_path_length']

    def __init__(self):
        self.graph = None
        self.flow = []
        self.contador = 0
        self.order = 0
        self.infinities = []
        self.vuln = []
        self.weighted_vuln = []
```

Disk 76.71 GB available



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - adj\_matrix1.csv
  - adj\_matrix7.csv
  - flow\_matrix.csv
  - thresholds.txt

```
[8] node_metrics = pd.concat([node_metrics, df9], axis=1, sort=False)
node_metrics = pd.concat([node_metrics, df10], axis=1, sort=False)
node_metrics = pd.concat([node_metrics, df11], axis=1, sort=False)

node_metrics.columns = self.COLUMNS_NODE

node_metrics.to_csv('node_metrics'+str(limiar)+'.csv')

def main(self, filename, limiar):
    # print (self.create_graph.__doc__)

    self.create_graph(filename)
    self.metrics_of_graph(limiar)
    self.metrics_of_nodes(limiar)

▶ for index in thresholds:
    limiar = thresholds[index][0]
    grp = GraphMaking()
    grp.main('adj_matrix'+str(limiar)+'.csv', limiar)
```

Disk 76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO

- Após finalizar a execução do código, basta fazer o download dos arquivos.



The screenshot shows a Jupyter Notebook interface. On the left, a file explorer displays a directory structure with files: sample\_data, adj\_matrix1.csv, adj\_matrix7.csv, flow\_matrix.csv, metrics\_1.csv, metrics\_7.csv, node\_metrics1.csv, node\_metrics7.csv, and thresholds.txt. A blue arrow points from the file explorer to the first text box. The main area shows a code cell with the following Python code:

```
for index in thresholds:  
    limiar = thresholds[index][0]  
    grp = GraphMaking()  
    grp.main('adj_matrix'+str(limiar)+'.csv', limiar)
```

Below the code cell, a second text box is connected to the first by a blue arrow. The interface also shows a top menu (File, Edit, View, Insert, Runtime, Tools, Help), a status bar (Last saved at 12:17 AM), and a RAM/Disk usage indicator (76.71 GB available).

Após alguns segundos os outros arquivos de saída aparecerão abaixo do primeiro arquivo de saída. Caso isso não aconteça automaticamente, basta atualizar a página!

Para fazer o download dos arquivos, basta sobrepor o cursor em cima de cada um deles, clicar nos três pontinhos que aparecerão e escolher a opção de realizar o download.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



## PARTE 3 – Entendendo o dado de saída:

	A	B	C	D	E	
1	,0,1,2,3,4					
2	0,0,9,8,0,0					
3	1,9,0,8,0,0					
4	2,8,8,0,7,0					
5	3,0,0,7,0,6					
6	4,0,0,0,6,0					
7						

	A	B	C	D	E	
1	,0,1,2,3,4					
2	0,0,9,8,0,0					
3	1,9,0,8,0,0					
4	2,8,8,0,0,0					
5	3,0,0,0,0,0					
6	4,0,0,0,0,0					
7						



Será gerado uma matriz para cada limiar inserido no dado de entrada. O nome da matriz estará de acordo com o seu limiar, ou seja, "adj\_matrix7.csv" tem limiar 7.

Perceba que a segunda matriz não possui os valores menores ou iguais a 7, pois seu limiar é 7.





# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



## PARTE 3 – Entendendo o dado de saída:

The screenshot shows the Microsoft Excel interface with the following data in the spreadsheet:

	A	B	C	D	E	F	G	H	I	J	K
1	,limiar,avg_betweenness_eccentricity,avg_path_length,avg_degree,diame										
2	0,1,1.4,1.7,2,3,0.5,5										
3											

The formula bar at the top shows the formula: `=,limiar,avg_betweenness_eccentricity,avg_path_length,avg_degree,diame`. The status bar at the bottom indicates the active sheet is 'metrics\_1' and the zoom level is 100%.





MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



Laboratório de investigação  
em Sistemas Sócio-Ambientais

# TUTORIAL DE USO DOS DADOS GERADOS POR CÓDIGOS PARA ELABORAÇÃO DE (GEO)GRAFOS

LISS - INPE

2020



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Este tutorial é produto de iniciação científica.

Elaboração do estudo: **LISS** – Laboratório de Investigação em **Sistemas Sócio-Ambientais**

Autor: Leticia da Silva Cabral

Orientador: Dr. Antonio Miguel Vieira Monteiro – LISS/INPE

Coorientador: Leonardo Bacelar de Lima Santos – CEMADEN

Colaboradores: Cátia Souza do Nascimento Sepetauskas e Jeferson Feitosa Mendes

CABRAL, L. S. ; MONTEIRO, A. M. V. ; SANTOS, L. B. L. ; MENDES, J. F. ; NASCIMENTO, C. S.



## APRESENTAÇÃO

Neste tutorial você aprenderá a:

Adicionar os  
dados do código 2  
na tabela de  
atributos

Entender a  
utilidade dos dados  
na produção do  
(geo)grafo

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Adicionar os dados na tabela de atributos

The screenshot shows the ArcMap interface. On the left, the 'Table of Contents' panel displays a list of layers under the path 'C:\Users\letic\Documents\IC INPE\Tutorial'. The layers listed are 'CENTROIDES' and 'EDGES\_SHAPEFILE'. In the center, the 'Add Data' dialog box is open, showing a file explorer view of the same directory. The files 'node\_metrics1.csv' and 'node\_metrics7.csv' are selected and highlighted in blue. The 'Name' field at the bottom of the dialog contains the text 'node\_metrics1.csv; node\_metrics7.csv'. To the right of the dialog, a red-bordered text box contains the following text: 'Adiciona-se as métricas dos nós, que estão identificadas com os seus respectivos limiares.'

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Adicionar os dados na tabela de atributos

Use-se a função “join” para incluir as métricas dos nós na tabela de atributos do shape de centroides. Neste exemplo use-se as métricas de limiar 1.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Adicionar os dados na tabela de atributos

Join Data

Join lets you append additional data to this layer's attribute table so you can, for example, symbolize the layer's features using this data.

What do you want to join to this layer?  
Join attributes from a table

1. Choose the field in this layer that the join will be based on:  
FID
2. Choose the table to join to this layer, or load the table from disk:  
node\_metrics1.csv  
 Show the attribute tables of layers in this list
3. Choose the field in the table to base the join on:  
Field1

Join Options

Keep all records  
All records in the target table are shown in the resulting table. Unmatched records will contain null values for all fields being appended into the target table from the join table.

Keep only matching records  
If a record in the target table doesn't have a match in the join table, that record is removed from the resulting target table.

Validate Join

About joining data

OK Cancel

Selecione as opções  
conforme esta  
imagem.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Adicionar os dados na tabela de atributos

Após realizar a ação “join”, os dados aparecerão na tabela de atributos do shape de centroides.

FID	Shape	id	long	lat	Field1	degree	betweenness	shortest paths	limiar	diameter graph	number of edges graph	avg degree	avg betweenness	eccentricity	transitivity	transitivity
0	Point	1	1	5	0	2	0	1,75	1	3	5	2	1,4	0,5	0,0	0,0
1	Point	2	2	5	1	2	0	1,75	1	3	5	2	1,4	0,5	0,0	0,0
2	Point	3	2	3	2	3	4	1,25	1	3	5	2	1,4	0,5	0,0	0,0
3	Point	4	3	2	3	2	3	1,5	1	3	5	2	1,4	0,5	0,0	0,0
4	Point	5	5	1	4	1	0	2,25	1	3	5	2	1,4	0,5	0,0	0,0

3,641 4,226 Decimal Degrees

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Uso dos dados

Os dados que foram inseridos podem ser explorados de diversas formas. Cada elemento das métricas é uma propriedade do grafo, que pode ser analisada e usada para representar as características do (geo)grafo, ou seja, podem ser usadas para melhor exemplificar os fenômenos de mobilidade daquela região. Seguem alguns exemplos desse uso:

1. Uso da Centralidade de grau (Degree) representada através do tamanho dos centroides.
2. Uso da Centralidade de Intermediação (Betweenness) representada através da coloração dos centroides.
3. Uso do peso das arestas para simbolizar a intensidade do fluxo.
4. Uso da coloração gradativa das arestas para simbolizar a intensidade do fluxo.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 1

Table of Contents

- Layers
  - C:\Users\letic\Documents\IC INPE\Tutorial
    - CENTROIDES**
    - EDGES\_SHAPEFILE
  - C:\Users\letic\Documents\IC INPE\Tutorial
    - node\_metrics1.csv
    - node\_metrics7.csv

Layer Properties

General Source Selection Display Symbology Fields Definition Query Labels Joins & Relates Time HTML Popup

Show:

**Features**

**Categories**

**Quantities**

- Graduated colors
- Graduated symbols
- Proportional symbols

**Charts**

**Multiple Attributes**

Draw quantities using symbol size to show relative values. Import...

Fields: Value: degree Normalization: none Classification: Natural Breaks (Jenks) Classes: 3 Classify...

Symbol Size from: 4 to: 10

Symbol	Range	Label
◆	1	1
●	2	2
●	3	3

Template

Show class ranges using feature values Advanced

OK Cancelar Aplicar

5,786 3,655 Decimal Degrees

A centralidade de grau é a coluna “degree” da tabela de atributos. Isso representa, de forma simplificada, o numero de contatos diretos que o vértice possui.

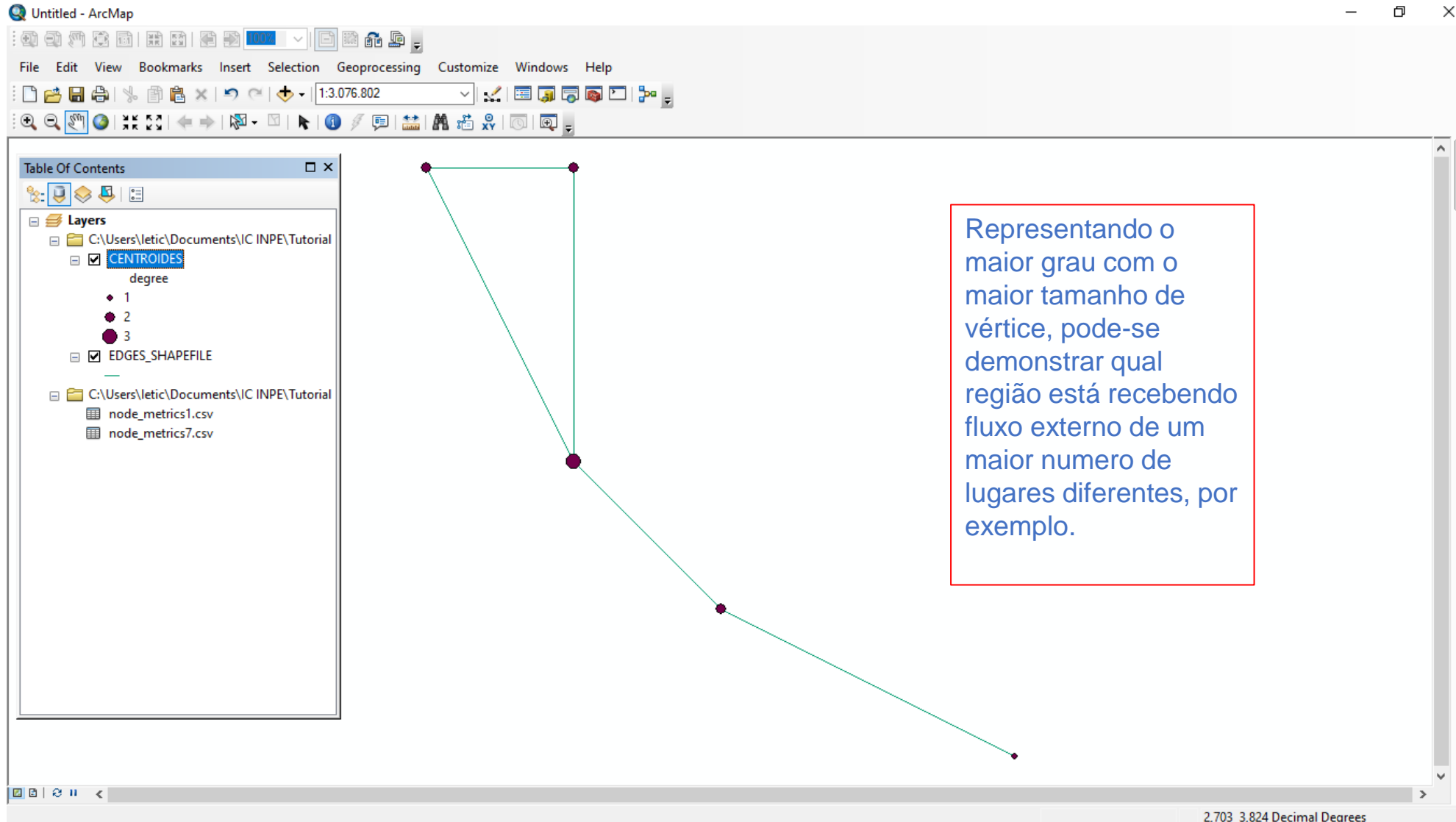


# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 1



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 2

Table Of Contents

- Layers
  - C:\Users\Ietic\Documents\IC INPE\Tutorial
    - CENTROIDES**
      - betweenness
        - 0,000000
        - 0,000001 - 3,000000
        - 3,000001 - 4,000000
      - EDGES\_SHAPEFILE
    - C:\Users\Ietic\Documents\IC INPE\Tutorial
      - node\_metrics1.csv
      - node\_metrics7.csv

Layer Properties

General Source Selection Display Symbology Fields Definition Query Labels Joins & Relates Time HTML Popup

Show:

Features

Categories

Quantities

- Graduated colors
- Graduated symbols
- Proportional symbols

Charts

Multiple Attributes

Draw quantities using color to show values. Import...

Fields

Value: betweenness

Classification

Natural Breaks (Jenks)

Classes: 3 Classify...

Normalization: none

Color Ramp:

Symbol	Range	Label
●	0,000000	0,000000
●	0,000001 - 3,000000	0,000001 - 3,000000
●	3,000001 - 4,000000	3,000001 - 4,000000

Show class ranges using feature values Advanced

OK Cancelar Aplicar

A centralidade de intermediação é a coluna “betweenness” da tabela de atributos. Isso representa, de forma simplificada, a influência daquele vértice, observando quantas vezes ele age como intermediário da ligação de outros dois vértices.

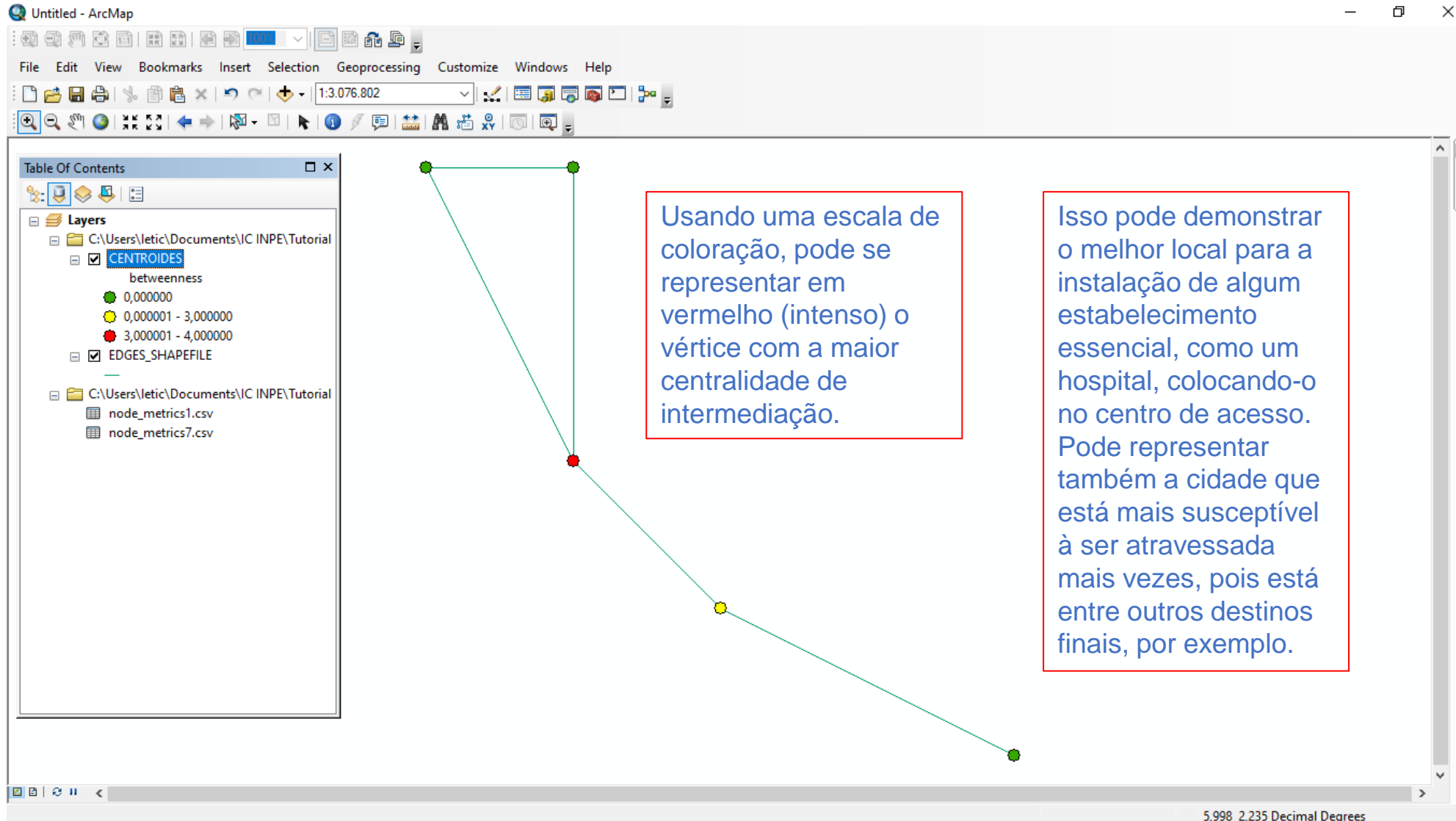
0,514 3,194 Decimal Degrees

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 2



Usando uma escala de coloração, pode se representar em vermelho (intenso) o vértice com a maior centralidade de intermediação.

Isso pode demonstrar o melhor local para a instalação de algum estabelecimento essencial, como um hospital, colocando-o no centro de acesso. Pode representar também a cidade que está mais susceptível à ser atravessada mais vezes, pois está entre outros destinos finais, por exemplo.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 3

Table Of Contents

- Layers
  - C:\Users\letic\Documents\IC INPE\Tutorial\...
    - CENTROIDES
    - EDGES\_SHAPEFILE
      - weight
        - 6,000000
        - 6,000001 - 7,000000
        - 7,000001 - 8,000000
        - 8,000001 - 9,000000

Layer Properties

General Source Selection Display Symbology Fields Definition Query Labels Joins & Relates Time HTML Popup

Show:

Features

Categories

Quantities

- Graduated colors
- Graduated symbols**
- Proportional symbols

Charts

Multiple Attributes

Draw quantities using symbol size to show relative values. Import...

Fields

Value: **weight**

Classification

Natural Breaks (Jenks)

Classes: 4 Classify...

Normalization: none

Symbol Size from: 0,5 to: 2

Symbol	Range	Label
	6,000000	6,000000
	6,000001 - 7,000000	6,000001 - 7,000000
	7,000001 - 8,000000	7,000001 - 8,000000
	8,000001 - 9,000000	8,000001 - 9,000000

Show class ranges using feature values Advanced

OK Cancel Aplicar

Escolhe-se o "weight" como valor de entrada.

0,771 3,15 Decimal Degrees

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 3

Table Of Contents

- Layers
  - C:\Users\letic\Documents\IC INPE\Tutorial
    - CENTROIDES
    - EDGES\_SHAPEFILE
      - weight
        - 6,000000
        - 6,000001 - 7,000000
        - 7,000001 - 8,000000
        - 8,000001 - 9,000000

Using the flow on the thickness of the edges, one can observe the routes with more intense traffic.

3,311 3,04 Decimal Degrees

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 4

Table Of Contents

- Layers
  - C:\Users\letic\Documents\IC INPE\Tutorial\...
    - CENTROIDES
    - EDGES\_SHAPEFILE
      - weight
        - 6,000000
        - 6,000001 - 7,000000
        - 7,000001 - 8,000000
        - 8,000001 - 9,000000

Layer Properties

General Source Selection Display Symbology Fields Definition Query Labels Joins & Relates Time HTML Popup

Show:

Features

Categories

Quantities

Graduated colors

Graduated symbols

Proportional symbols

Charts

Multiple Attributes

Draw quantities using color to show values. Import...

Fields

Value: weight

Classification

Natural Breaks (Jenks)

Classes: 4 Classify...

Normalization: none

Color Ramp: [Color Ramp]

Symbol	Range	Label
[Green]	6,000000	6,000000
[Light Green]	6,000001 - 7,000000	6,000001 - 7,000000
[Yellow]	7,000001 - 8,000000	7,000001 - 8,000000
[Red]	8,000001 - 9,000000	8,000001 - 9,000000

Show class ranges using feature values Advanced

OK Cancel Aplicar

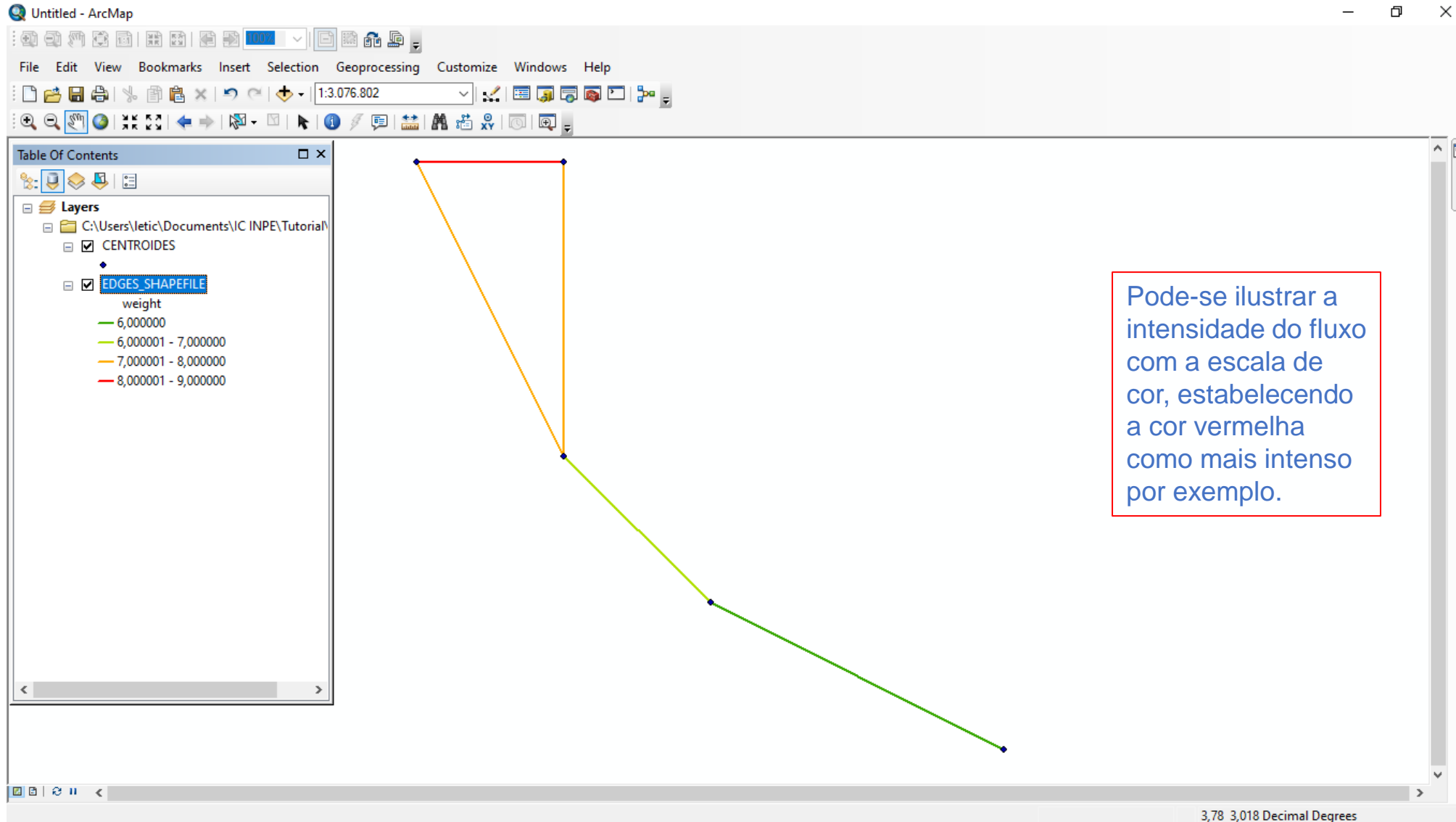
Escolhe-se o "weight" como valor de entrada.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo 4





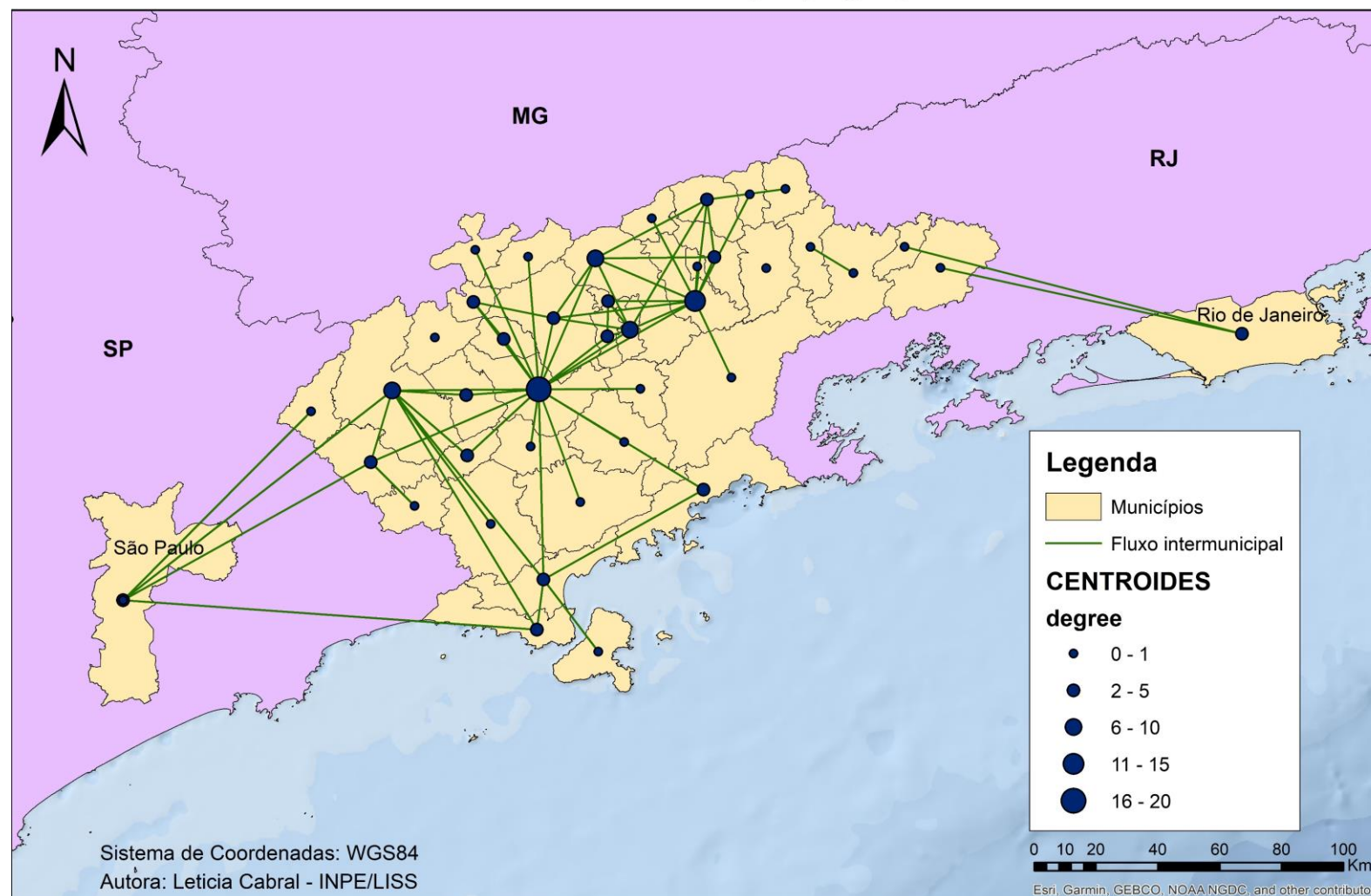
# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo prático 1

### Coletivo Escolar na RMVPLN, Rio de Janeiro e São Paulo - (geo)grafo Grau de Intermediação (degree)





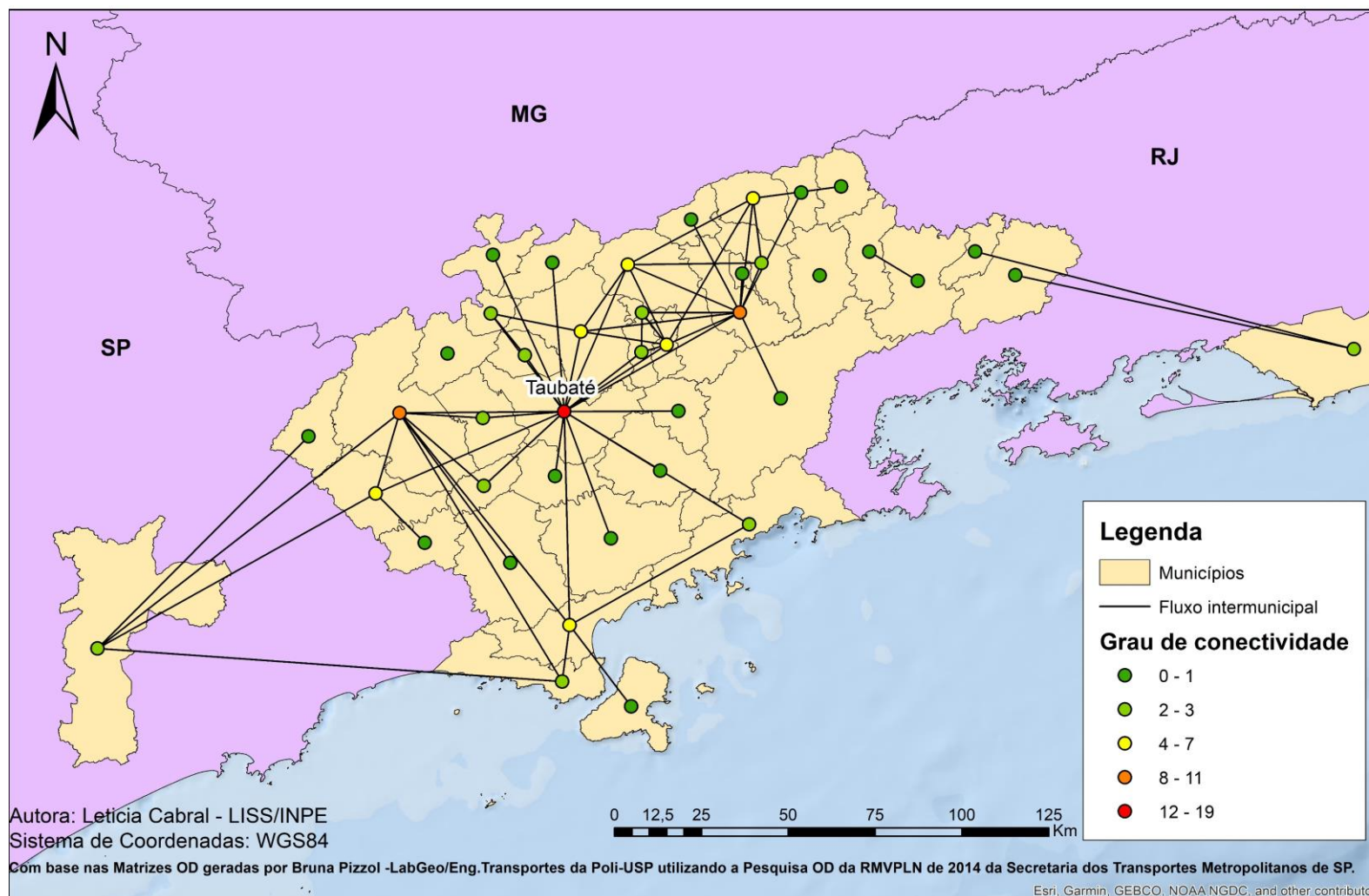
# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo prático 2

Modo de deslocamento: **Coletivo** - Motivo do deslocamento: **Escolar, RMVPLN**



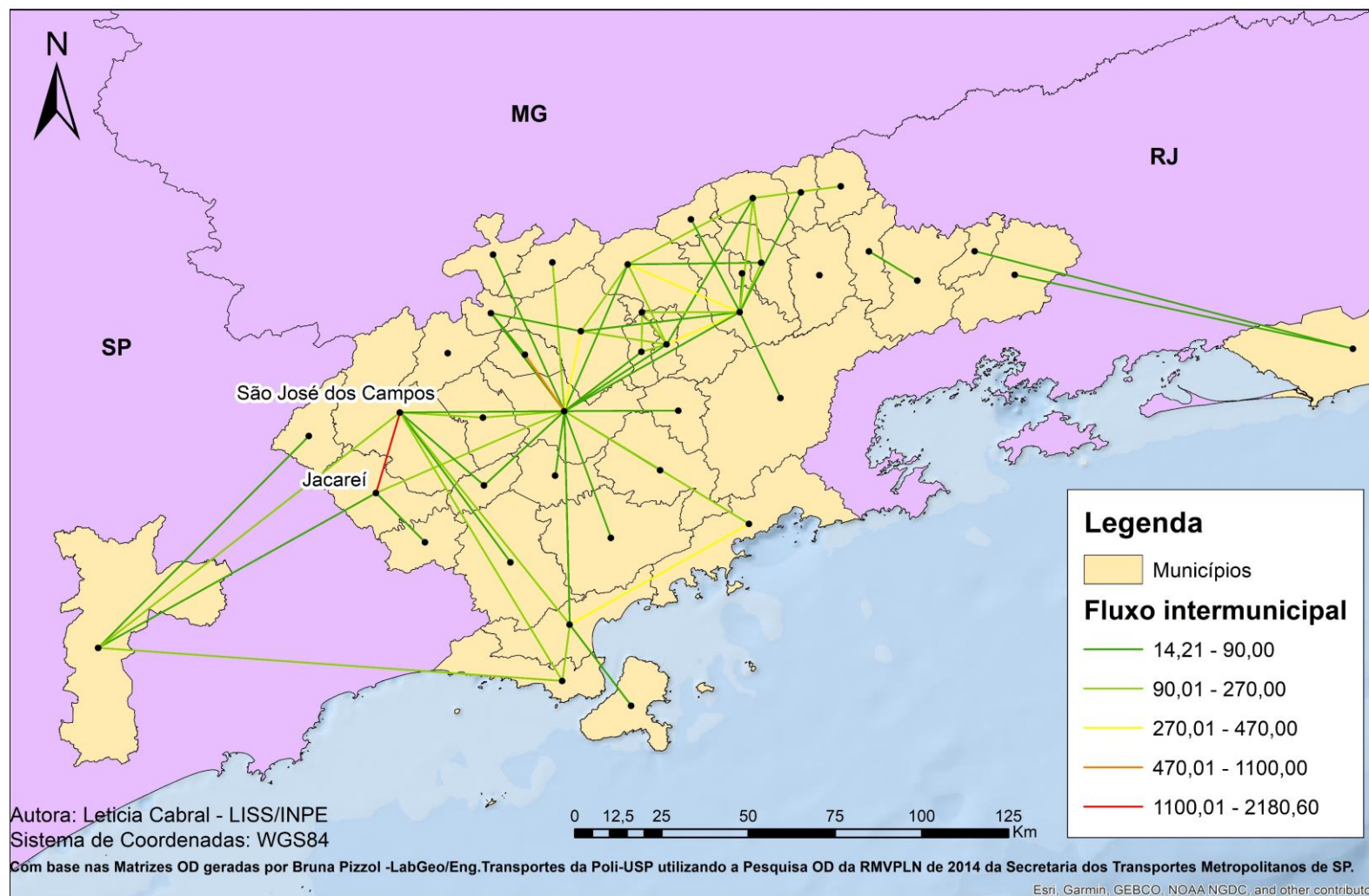
# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 2 – Exemplo prático 3

Modo de deslocamento: **Coletivo** - Motivo do deslocamento: **Escolar, RMVPLN**





MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



Laboratório de investigação  
em Sistemas Sócio-Ambientais

# TUTORIAL PARA GERAÇÃO DE DADOS PARA ANÁLISE DE (GEO)GRAFO A PARTIR DE CÓDIGO EM PYTHON NO GOOGLE COLAB

LISS - INPE

2020

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Este tutorial é produto de iniciação científica.

Elaboração do estudo: **LISS** – Laboratório de Investigação em Sistemas **S**ócio-Ambientais

Autor: Leticia da Silva Cabral

Orientador: Dr. Antonio Miguel Vieira Monteiro – LISS/INPE

Coorientador: Leonardo Bacelar de Lima Santos – CEMADEN

Colaboradores: Cátia Souza do Nascimento Sepetauskas e Jeferson Feitosa Mendes

CABRAL, L. S. ; MONTEIRO, A. M. V. ; SANTOS, L. B. L. ; MENDES, J. F. ; NASCIMENTO, C. S.



## APRESENTAÇÃO

Neste tutorial você aprenderá a:

Preparar os  
dados de  
entrada do  
código

Executar código  
no google colab

Entender o dado  
de saída

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



- O Google Colaboratory é um ambiente de notebooks Jupyter que não requer configuração e é executado na nuvem. Através do Google Colab é possível executar códigos em Python de forma gratuita, bem como compartilhá-lo e salvar as saídas geradas.
- Para acessar o código usado neste tutorial clique aqui:  
<https://colab.research.google.com/drive/11g2Er1Oj1xvndbS3QadH93QvGyUBqog8?usp=sharing>

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Preparação do dado de entrada:

- Usaremos um arquivo .txt e um arquivo .csv:
  1. O arquivo .txt deve conter os limiares, que serão usados como um filtro para a matriz de fluxo. Esse limiar será o parâmetro usado para ver se um determinado valor de fluxo é forte o bastante para aparecer no grafo. Se o valor for menor que o limiar, ele será mostrado como 0 e se for maior continuará com o seu valor normal no dado de saída.
  2. O arquivo .cvs deve conter a matriz de fluxo, com os valores de fluxo entre as regiões de estudo.

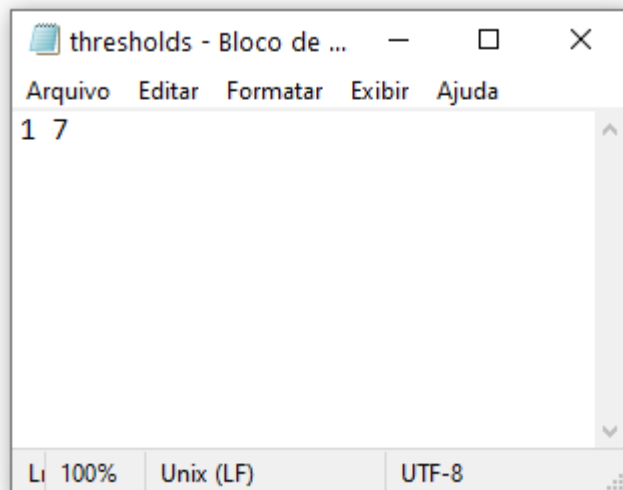


# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 1 – Preparação do dado de entrada:



Perceba que colocou-se 2 limiares, portanto  
teremos dados de saída diferentes para  
cada um deles.

flow\_matrix - Excel

	A	B	C	D	E	F
1	0	9	8	0	0	
2	0	0	8	0	0	
3	0	0	0	7	0	
4	0	0	0	0	6	
5	0	0	0	0	0	
6						
7						
8						
9						
10						
11						
12						
13						



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO

## PARTE 2 – Execução do código no google colab:

- Ao clicar no link, aparecerá a seguinte tela:



Laboratório de investigação  
em Sistemas Sócio-Ambientais

The screenshot shows a Google Colab notebook titled "Leonardo.ipynb". The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help" menus. A "Table of contents" sidebar on the left lists sections: "Imports", "Lendo arquivo de limiares", "Lendo matriz de fluxo quadrada", "Simetrizando matriz e zerando a diagonal", "Criando matrizes de adjacencia por limiares", "Criando a classe GraphMaking e calculando suas metricas", and a "Section" button. The main workspace shows a code cell with the command `pip install python-igraph` and its output: "Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)" and "Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)". Below this, the notebook content is organized into sections: "Imports" with code for `import numpy as np`, `import pandas as pd`, `from igraph import Graph`, `from statistics import mean`, and `import math`; and "Lendo arquivo de limiares" with code for `filename = 'thresholds'`, `sep = ';'` , and `thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")`.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Leonardo.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 25

Comment Share

RAM Disk

Editing

Files

Upload Refresh Mount Drive

sample\_data

```
pip install python-igraph
```

Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)

Imports

```
[ ] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```

Lendo arquivo de limiares

```
[ ] filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header=None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")
thresholds
```

Disk 76.72 GB available

Ao clicar neste ícone, aparecerá as opções acima. Clique em "Upload" para carregar as entradas do código.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Upload > arquivo .txt + arquivo excel

The screenshot shows a Jupyter Notebook interface with a file upload dialog box open. The dialog box is titled "Abrir" and shows the "Downloads" folder. It contains a table of files:

Nome	Data de modificação	Tipo
Ontem (2)		
thresholds	27/05/2020 23:48	Documento de Texto
flow_matrix	27/05/2020 23:48	Arquivo de Valor

The code in the notebook is as follows:

```
[ ] filename = 'thresholds'  
sep = ';'   
try:   
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")   
except:   
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")   
thresholds
```

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



- Clicar no símbolo de “play” que está indicado neste slide com o quadrado vermelho.

A screenshot of a Jupyter Notebook interface. On the left, a 'Files' sidebar shows a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. A blue arrow points from this folder to a red-bordered text box. The main area shows a code cell with a red square around the play button icon. The output of the code cell shows that 'python-igraph' and 'texttable' are already installed. Below the code cell, there are sections for 'Imports' and 'Lendo arquivo de limiares' with their respective code snippets. The bottom of the interface shows a disk usage indicator with 76.72 GB available.

Leonardo.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on May 25

RAM ✓ Disk ✓ Editing

Files  
Upload Refresh Mount Drive  
..  
sample\_data  
flow\_matrix.csv  
thresholds.txt

Code + Text

pip install python-igraph

Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)

Imports

```
[ ] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```

Lendo arquivo de limiares

```
[ ] filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")
thresholds
```

Disk 76.72 GB available

Caso o upload seja feito com sucesso, os arquivos aparecerão aqui.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Clicar em “run anyway”.

A screenshot of a Jupyter Notebook interface. The top bar shows the user 'Leonardo.ipynb' and various menu options like 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. The left sidebar displays a file explorer with a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. The main area shows a code cell with the command 'pip install python-igraph'. Below the code, a message indicates that requirements are already satisfied. A white warning dialog box is overlaid on the notebook, with the text: 'Warning: This notebook was not authored by Google. This notebook was authored by souz.kti@gmail.com. It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook. Please contact the creator of this notebook at souz.kti@gmail.com with any additional questions.' The dialog box has two buttons: 'CANCEL' and 'RUN ANYWAY'. The bottom of the notebook shows a code cell with Python code for reading a CSV file.

```
[ ] filename = 'thresholds'  
    sep = ';'   
    try:   
        thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")   
    except:   
        thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")   
    thresholds
```

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Cada etapa levará alguns poucos segundos para ser executada.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** Shows a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'.
- Code Cell:** Contains the command `pip install python-igraph`. The output shows the successful installation of 'python-igraph-0.8.2' and 'texttable-1.6.2'.
- Imports Cell:** Contains the following code:

```
[ ] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```
- Execution Cell:** Contains the following code:

```
[ ] filename = 'thresholds'
sep = ';'


```

A red box highlights the play button icon in the code cell, and a blue arrow points from a text box to the square execution markers in the subsequent code cells.

Quando a primeira etapa for concluída, clicar no próximo colchetes.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Assim que finalizar o símbolo de “play” aparecerá novamente.

The screenshot shows a Jupyter Notebook interface. On the left, a file browser shows a folder named 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'. A blue arrow points from the 'sample\_data' folder to a text box. The main area shows a code cell with the command `[1] pip install python-igraph` and its output, which includes downloading and installing 'python-igraph' and 'texttable'. Below this, an 'Imports' section shows the code `import numpy as np`, `import pandas as pd`, `from igraph import Graph`, `from statistics import mean`, and `import math`. At the bottom, a code cell shows `[ ] filename = 'thresholds'` and `sep = ';'` . The interface includes a top menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A status bar at the bottom indicates 'Disk 76.71 GB available'.

Ao clicar no próximo colchetes, aparecerá um número no colchetes anterior em que a execução foi finalizada. Esses números aparecerão em sequência até o fim do código.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** Shows a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'.
- Imports Cell:** Contains the following code:

```
[2] import numpy as np
import pandas as pd
from igraph import Graph
from statistics import mean
import math
```
- Lendo arquivo de limiares Cell:** Contains the following code:

```
filename = 'thresholds'
sep = ';'
try:
    thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
except:
    thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")
thresholds
```
- Output:** Shows a table with two rows:

0	1	
0	1	7
- Annotations:** A red box highlights the 'play' button in the 'Lendo arquivo de limiares' cell, with a blue arrow pointing to it. A text box next to it says: "Clicar no colchetes seguinte e repetir isso até o fim do código."



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

Leonardo.ipynb

File Edit View Insert Runtime Tools Help

RAM  Disk  Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - flow\_matrix.csv
  - thresholds.txt

```
[3] thresholds = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
      except:
          thresholds = pd.read_csv(filename+'.csv', delimiter=sep, header= None, encoding = "ISO-8859-1")
      thresholds
```

	0	1
0	1	7

Perceba que o número 3 apareceu no terceiro colchetes. Atente-se para não pular nenhum colchetes.

▼ Lendo matriz de fluxo quadrada

```
filename = 'flow_matrix'

sep = ';'
try:
    mtx = pd.read_csv(filename+'.txt', delim_whitespace=True, header= None, encoding = "ISO-8859-1")
except:
    mtx = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")
mtx
```

	0	1	2	3	4
0	0	9	8	0	0
1	0	0	8	0	0
2	0	0	0	7	0

Disk  76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** 'Leonardo.ipynb' with a star icon, and menu options: File, Edit, View, Insert, Runtime, Tools, Help. On the right, there are 'Comment', 'Share', and 'Editing' buttons, along with RAM and Disk usage indicators.
- Files Panel:** Shows a directory structure with 'sample\_data' containing 'flow\_matrix.csv' and 'thresholds.txt'.
- Code Cell:** Contains Python code to read a CSV file into a pandas DataFrame. The code uses a try-except block to handle different file extensions (.txt or .csv).
- Output:** A text representation of a 5x5 matrix with columns labeled 0, 1, 2, 3, 4.
- Next Cell:** The title 'Simetrizando matriz e zerando a diagonal' is visible at the bottom.

```
filename = 'flow_matrix'

sep = ';'
try:
    mtx = pd.read_csv(filename+'.txt', delim_whitespace=True, header=None, encoding = "ISO-8859-1")
except:
    mtx = pd.read_csv(filename+'.csv', delimiter=sep, header=None, encoding = "ISO-8859-1")
mtx
```

	0	1	2	3	4	
0	0	0	9	8	0	0
1	0	0	0	8	0	0
2	0	0	0	0	7	0
3	0	0	0	0	0	6
4	0	0	0	0	0	0

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
- flow\_matrix.csv
- thresholds.txt

+ Code + Text

```
4 0 0 0 0 0
```

Simetrizando matriz e zerando a diagonal

```
def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(i==j):  
                dados.iloc[j,i]=0  
            elif(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)  
mtx
```

	0	1	2	3	4	
0	0	0	9	8	0	0
1	9	0	8	0	0	0
2	8	8	0	7	0	0
3	0	0	7	0	6	0

Disk 76.71 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

SaveShapefilesFromLatLongFile.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Última edição em 1 de julho

Comentário Compartilhar

Índice

- Symmetrizing the matrix
  - Creating graph from adj matrix
    - Seção

```
[ ] #Symmetrizing by sum
def simetric_matrix_sum(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = original.iloc[i,j] + original.iloc[j,i]
    return dados

#Symmetrizing by average
def simetric_matrix_avg(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = (original.iloc[i,j] + original.iloc[j,i])/2
    return dados

[ ] #You have to choose which function you would like to apply

#mtx = simetric_matrix_higher(mtx)
#mtx = simetric_matrix_lower(mtx)
mtx = simetric_matrix_sum(mtx)
#mtx = simetric_matrix_avg(mtx)
```

Nesta parte, será definido a forma com que a matriz será simetrizada. As opções são simetrização pelo maior, menor, soma ou média dos dois valores. É necessário “descomentar”, ou seja, tirar o símbolo ‘#’ da linha que deve ser executada, mantendo as demais com o símbolo ‘#’.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆  
File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

### Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - flow\_matrix.csv
  - thresholds.txt

### Code

#### Simetrizando matriz e zerando a diagonal

```
[5] def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(i==j):  
                dados.iloc[j,i]=0  
            elif(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

▶ mtx = simetric\_matrix(mtx)  
mtx

	0	1	2	3	4	
0	0	0	9	8	0	0
1	9	0	8	0	0	
2	8	8	0	7	0	
3	0	0	7	0	6	
4	0	0	0	6	0	

Disk 76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings User Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - flow\_matrix.csv
  - thresholds.txt

### ▼ Criando matrizes de adjacencia por limiares

```
for index in thresholds:  
    limiar = thresholds[index][0]  
  
    for i in range(0, len(mtx.columns)):  
        mtx[i] = np.where((mtx[i] <= limiar), 0, mtx[i])  
  
    mtx.to_csv('adj_matrix'+str(limiar)+'.csv')
```

### ▼ Criando a classe GraphMaking e calculando suas metricas

```
[ ] class GraphMaking:  
  
    COLUMNS = [  
        'limiar',  
        'avg_betweenness_eccentricity',  
        'avg_path_length',  
        'avg_degree',  
        'diameter',  
        'transitivity',  
        'number_of_edges'  
    ]  
  
    COLUMNS_NODE = ['degree', 'betweenness', 'transitivity', 'shortest paths',
```

Disk 76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Header:** Leonardo.ipynb, File Edit View Insert Runtime Tools Help, Comment Share, RAM Disk, Editing.
- Files Panel:** sample\_data folder containing adj\_matrix1.csv, adj\_matrix7.csv, flow\_matrix.csv, and thresholds.txt.
- Code Cell 1:** "Criando matrizes de adjacencia por limiares" with Python code for processing thresholds.
- Code Cell 2:** "Criando a classe GraphMaking e calculando suas metricas" with a class definition.

```
[7] for index in thresholds:
    limiar = thresholds[index][0]

    for i in range(0, len(mtx.columns)):
        mtx[i] = np.where((mtx[i] <= limiar), 0, mtx[i])

    mtx.to_csv('adj_matrix'+str(limiar)+'.csv')
```

```
class GraphMaking:

    COLUMNS = [
        'limiar',
        'avg_betweenness_eccentricity',
        'avg_path_length',
        'avg_degree',
        'diameter',
        'transitivity',
        'number_of_edges'
    ]

    COLUMNS_NODE = ['degree', 'betweenness', 'transitivity', 'shortest paths',
```

Perceba que a primeira saída do código passa a aparecer aqui.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

Leonardo.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings Profile

RAM Disk Editing

### Files

Upload Refresh Mount Drive

- ..
- sample\_data
  - adj\_matrix1.csv
  - adj\_matrix7.csv
  - flow\_matrix.csv
  - thresholds.txt

### Code

#### Criando a classe GraphMaking e calculando suas metricas

```
class GraphMaking:

    COLUMNS = [
        'limiar',
        'avg_betweenness_eccentricity',
        'avg_path_length',
        'avg_degree',
        'diameter',
        'transitivity',
        'number_of_edges'
    ]

    COLUMNS_NODE = ['degree', 'betweenness', 'transitivity', 'shortest_paths',
                    'limiar', 'diameter_graph', 'number_of_edges_graph',
                    'avg_degree', 'avg_betweenness_eccentricity',
                    'transitivity', 'avg_path_length']

    def __init__(self):
        self.graph = None
        self.flow = []
        self.contador = 0
        self.order = 0
        self.infinities = []
        self.vuln = []
        self.weighted_vuln = []
```

Disk 76.71 GB available



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchete.

The screenshot shows a Jupyter Notebook interface. At the top, the user 'Leonardo.ipynb' is logged in. The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options like Comment, Share, and Editing. On the left, a file explorer shows a directory named 'sample\_data' containing files: 'adj\_matrix1.csv', 'adj\_matrix7.csv', 'flow\_matrix.csv', and 'thresholds.txt'. The main area contains a code cell with the following Python code:

```
[8] node_metrics = pd.concat([node_metrics, df9], axis=1, sort=False)
node_metrics = pd.concat([node_metrics, df10], axis=1, sort=False)
node_metrics = pd.concat([node_metrics, df11], axis=1, sort=False)

node_metrics.columns = self.COLUMNS_NODE

node_metrics.to_csv('node_metrics'+str(limiar)+'.csv')

def main(self, filename, limiar):
    # print (self.create_graph.__doc__)

    self.create_graph(filename)
    self.metrics_of_graph(limiar)
    self.metrics_of_nodes(limiar)
```

Below the code cell, a 'Run' button (play icon) is visible, followed by a code block:

```
for index in thresholds:
    limiar = thresholds[index][0]
    grp = GraphMaking()
    grp.main('adj_matrix'+str(limiar)+'.csv', limiar)
```

At the bottom left, a disk usage indicator shows '76.71 GB available'.

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO

- Após finalizar a execução do código, basta fazer o download dos arquivos.



Leonardo.ipynb ☆  
File Edit View Insert Runtime Tools Help Last saved at 12:17 AM

RAM  Disk  Editing

```
for index in thresholds:  
    limiar = thresholds[index][0]  
    grp = GraphMaking()  
    grp.main('adj_matrix'+str(limiar)+'.csv', limiar)
```

Files  
Upload Refresh Mount Drive  
sample\_data  
adj\_matrix1.csv  
adj\_matrix7.csv  
flow\_matrix.csv  
metrics\_1.csv  
metrics\_7.csv  
node\_metrics1.csv  
node\_metrics7.csv  
thresholds.txt

Após alguns segundos os outros arquivos de saída aparecerão abaixo do primeiro arquivo de saída. Caso isso não aconteça automaticamente, basta atualizar a página!

Para fazer o download dos arquivos, basta sobrepor o cursor em cima de cada um deles, clicar nos três pontinhos que aparecerão e escolher a opção de realizar o download.

Disk  76.71 GB available

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



## PARTE 3 – Entendendo o dado de saída:

	A	B	C	D	E
1	,0,1,2,3,4				
2	0,0,9,8,0,0				
3	1,9,0,8,0,0				
4	2,8,8,0,7,0				
5	3,0,0,7,0,6				
6	4,0,0,0,6,0				
7					

	A	B	C	D	E
1	,0,1,2,3,4				
2	0,0,9,8,0,0				
3	1,9,0,8,0,0				
4	2,8,8,0,0,0				
5	3,0,0,0,0,0				
6	4,0,0,0,0,0				
7					



Será gerado uma matriz para cada limiar inserido no dado de entrada. O nome da matriz estará de acordo com o seu limiar, ou seja, "adj\_matriz7.csv" tem limiar 7.

Perceba que a segunda matriz não possui os valores menores ou iguais a 7, pois seu limiar é 7.



# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 3 – Entendendo o dado de saída:

node\_metrics7 - Excel

Arquivo | Página Inicial | Inserir | Layout da Página | Fórmulas | Dados | Revisão | Exibir | Ajuda | Diga-me o que você deseja fazer | Compartilhar

Área de Transferência | Fonte | Alinhamento | Número | Estilos | Células | Edição

A1 : ,degree,betweenness,transitivity,shortest\_paths,limiar,diameter\_graph,number\_of\_edges\_graph,avg\_degree,avg\_betweenness\_eccentr

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	,degree,betweenness,transitivity,shortest_paths,limiar,diameter_graph,number_of_edges_graph,avg_degree,avg_betweenness_eccentricity,transitivity,avg_path_length																
2	0,2,0.0,1.0,0.5,7,1,3,1.2,0.0,1.0,1.0																
3	1,2,0.0,1.0,0.5,7,1,3,1.2,0.0,1.0,1.0																
4	2,2,0.0,1.0,0.5,7,1,3,1.2,0.0,1.0,1.0																
5	3,0,0.0,-1.0,0.0,7,1,3,1.2,0.0,1.0,1.0																
6	4,0,0.0,-1.0,0.0,7,1,3,1.2,0.0,1.0,1.0																
7																	

node\_metrics7

100%

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 3 – Entendendo o dado de saída:

The screenshot shows the Microsoft Excel interface with the following data in the spreadsheet:

	A	B	C	D	E	F	G	H	I	J	K
1	,limiar,avg_betweenness_eccentricity,avg_path_length,avg_degree,diame										
2	0,1,1.4,1.7,2,3,0.5,5										
3											

The formula bar shows the formula: `=,limiar,avg_betweenness_eccentricity,avg_path_length,avg_degree,diame`. The status bar at the bottom indicates the active sheet is 'metrics\_1' and the zoom level is 100%.









MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



Laboratório de investigação  
em Sistemas Sócio-Ambientais

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO A PARTIR DA EXECUÇÃO DE CÓDIGO EM PYTHON NO GOOGLE COLAB

LISS - INPE

2020

# TUTORIAL PARA ANÁLISE DE (GEO)GRAFO



Este tutorial é produto de iniciação científica.

Elaboração do estudo: **LISS** – Laboratório de Investigação em Sistemas **S**ócio-Ambientais

Autor: Leticia da Silva Cabral

Orientador: Dr. Antonio Miguel Vieira Monteiro – LISS/INPE

Coorientador: Leonardo Bacelar de Lima Santos – CEMADEN

Colaboradores: Cátia Souza do Nascimento Sepetauskas e Jeferson Feitosa Mendes

CABRAL, L. S. ; MONTEIRO, A. M. V. ; SANTOS, L. B. L. ; MENDES, J. F. ; NASCIMENTO, C. S.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## APRESENTAÇÃO

Neste tutorial você aprenderá a:

Preparar os  
dados de  
entrada do  
código

Executar código  
no google colab

Entender o dado  
de saída

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Esse tutorial trata da elaboração de (geo)grafos, que são grafos, conjunto de vértices/elementos, que possuem localização espacial fixa e conhecida, e arestas/ conexões, que apresentam as dependências espaciais (fluxo).
- O Google Colaboratory é um ambiente de notebooks Jupyter que não requer configuração e é executado na nuvem. Através do Google Colab é possível executar códigos em Python de forma gratuita, bem como compartilhá-lo e salvar as saídas geradas.
- Para acessar o código usado neste tutorial clique aqui:  
[https://colab.research.google.com/drive/1\\_eqoUxyhVWrvWdWnnvoXV3ScfNKi9If8?usp=sharing](https://colab.research.google.com/drive/1_eqoUxyhVWrvWdWnnvoXV3ScfNKi9If8?usp=sharing)

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## PARTE 1 – Preparação do dado de entrada:

- Usaremos como dado de entrada dois arquivos .txt:
  - 1) Para o primeiro arquivo, deve-se gerar os centroides a partir dos polígonos (regiões de estudo. Ex: Municípios.) A partir desses centroides, calcula-se suas respectivas latitude e longitude. Essas informações devem ser dispostas da seguinte forma em um documento do bloco de notas: 1ª Coluna: ID ( identificação do centroide com um numero inteiro); 2ª Coluna: Latitude (respectiva latitude do centroide identificado); 3ª Coluna: Longitude (respectiva longitude do centroide identificado).
  - 2) Para o segundo arquivo, precisa-se dos dados de fluxo entre esses centroides, ou seja, o fluxo entre as regiões de estudo em questão. Esses dados devem ser dispostos no formato de matriz com dimensão proporcional ao número de centroides.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## PARTE 1 – Observações importantes:

```
id_long_lat - Bloco de ...
Arquivo Editar Formatar Exibir Ajuda
id long lat
1 1 5
2 2 5
3 2 3
4 3 2
5 5 1
Ln 100% Windows (CRLF) UTF-8
```

```
matriz_teste - Bloco de ...
Arquivo Editar Formatar Exibir Ajuda
5 9 8 0 0
0 6 8 0 0
0 0 7 7 0
0 0 0 8 6
0 0 0 0 9
Ln 100% Windows (CRLF) UTF-8
```

Os elementos  $a_{i,j}$  em que  $i = j$  devem conter o fluxo que existe dentro daquele centroide, ou seja, o fluxo interno da região.

À exemplo, usou-se números inteiros. Para coordenadas reais usar "." para separar as casas decimais. **O uso de "," dá erro ao executar o programa!**

Os elementos  $a_{1,2}$  e  $a_{2,1}$  representam a mesma informação, ou seja, o fluxo entre o centroide número 1 e o centroide número 2. Sendo assim, só se faz necessário preencher um desses dois elementos, por isso, teremos uma matriz triangular, com zeros abaixo da diagonal principal.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



## PARTE 2 – Execução do código no google colab:

- Ao clicar no link disponibilizado no slide 4, aparecerá a seguinte tela:

A screenshot of a Google Colab notebook interface. The top bar shows the file name 'SaveShapefilesFromLatLongFile.ipynb' and the date 'Last edited on May 18'. The left sidebar contains a 'Table of contents' with sections like 'Simetrizando matriz' and 'Section'. The main area shows two code cells. The first cell contains the command 'pip install python-igraph' and its output: 'Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)' and 'Requirement already satisfied: texttable&gt;=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)'. The second cell contains 'pip install geopandas' and its output, listing various dependencies like shapely, pandas, fiona, and numpy. The third cell shows the start of a code block for plotting: '%matplotlib inline', '## It is to avoiding to open an extra window showing maps', 'import geopandas as gpd', 'import pandas as pd', 'from igraph import Graph', and 'import numpy as np'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

CO SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on May 18

RAM [ ] Disk [ ] Editing [ ]

Files [X] Upload Refresh Mount Drive

sample\_data

```
[ ] pip install python-igraph
```

```
Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)
```

```
[ ] pip install geopandas
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)  
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)  
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)  
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)  
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)  
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.  
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201  
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop  
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)  
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)  
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)  
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)  
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)  
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)
```

```
[ ] %matplotlib inline  
## It is to avoiding to open an extra window showing maps  
import geopandas as gpd  
import pandas as pd  
from igraph import Graph  
import numpy as np
```

Disk [ ] 76.72 GB available

Ao clicar neste ícone, aparecerá as opções acima. Clique em "Upload" para carregar as entradas do código.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Upload > arquivo .txt matriz de fluxo + arquivo .txt lat e long dos pontos.

The screenshot displays a Jupyter Notebook environment. The top bar shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and standard menu options. On the left, a file browser shows a folder named 'sample\_data'. The main area is divided into a code editor and a terminal window. The code editor contains the following Python code:

```
[ ] %matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
```

The terminal window shows the output of a file upload operation, listing the files 'matriz\_teste' and 'id\_long\_lat' with their modification dates and types. Below the terminal, a file explorer window is open, showing the 'Downloads' folder with the same two files selected. The file explorer window has a search bar and a list of files with columns for 'Nome', 'Data de modificação', and 'Tipo'. The file names 'id\_long\_lat' and 'matriz\_teste' are visible in the search bar.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Clicar no símbolo de “play” que está indicado neste slide com o quadrado vermelho.

SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on May 18

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

```
pip install python-igraph
```

```
Requirement already satisfied: python-igraph in /usr/local/lib/python3.6/dist-packages (0.8.2)  
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.6/dist-packages (from python-igraph) (1.6.2)
```

```
pip install geopandas
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.6/dist-packages (0.7.0)  
Requirement already satisfied: shapely in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.7.0)  
Requirement already satisfied: pandas>=0.23.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.0.3)  
Requirement already satisfied: fiona in /usr/local/lib/python3.6/dist-packages (from geopandas) (1.8.13.post1)  
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.6/dist-packages (from geopandas) (2.6.1.post1)  
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (1.  
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geopandas) (201  
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.6/dist-packages (from pandas>=0.23.0->geop  
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (0.5.0)  
Requirement already satisfied: click<8,>=4.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (7.1.2)  
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (19.3.0)  
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.12.0)  
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (1.1.1)  
Requirement already satisfied: munch in /usr/local/lib/python3.6/dist-packages (from fiona->geopandas) (2.5.0)
```

```
%matplotlib inline  
## It is to avoiding to open an extra window showing maps  
import geopandas as gpd
```

Disk 76.72 GB available

matriz\_teste.txt id\_long\_lat.txt Exibir todos

Caso o upload seja feito com sucesso, os arquivos aparecerão aqui.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Clicar em “run anyway”.

A screenshot of a Jupyter Notebook interface. The notebook title is 'SaveShapefilesFromLatLongFile.ipynb'. The left sidebar shows a file explorer with folders and files like 'sample\_data', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area shows a code cell with the command 'pip install python-igraph'. A warning dialog box is overlaid on the notebook, stating: 'Warning: This notebook was not authored by Google. This notebook was authored by geographsmail@gmail.com. It may request access to your data stored with Google, or read data and credentials from other sessions. Please review the source code before executing this notebook. Please contact the creator of this notebook at geographsmail@gmail.com with any additional questions.' The dialog box has two buttons: 'CANCEL' and 'RUN ANYWAY'. The background shows the notebook's output area with various package installation logs for geopandas, fiona, and munch.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Cada etapa levará alguns poucos segundos para ser executada.

A screenshot of a Jupyter Notebook interface. The top bar shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and various icons for file management and sharing. The left sidebar displays a file explorer with folders like 'sample\_data' and files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area shows a code cell with the command 'pip install python-igraph' highlighted by a red box. Below it, the terminal output shows the installation progress for 'python-igraph' and 'texttable', including download URLs and progress bars. A second code cell shows 'pip install geopandas', followed by a list of requirements already satisfied, such as 'geopandas', 'shapely', 'pandas', 'fiona', 'pyproj', 'numpy', 'pytz', 'python-dateutil', 'cligj', 'click', 'attrs', 'six', 'click-plugins', and 'munch'. The bottom status bar indicates 'Disk 76.72 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Assim que finalizar o símbolo de “play” aparecerá novamente.

The screenshot shows a Jupyter Notebook interface. On the left, a file explorer shows a folder named 'sample\_data' containing 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area displays two code cells. The first cell, which is currently active, contains the command 'pip install python-igraph'. Below the code, the terminal output shows the installation progress for 'python-igraph' and 'texttable'. The second cell contains the command 'pip install geopandas', and its output shows that all dependencies for 'geopandas' are already satisfied. A red box on the left contains the text 'Quando a primeira etapa for concluída, clicar no próximo colchetes.' with a blue arrow pointing to the square bracket at the start of the second code cell. The top of the interface shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb' and various utility icons like RAM and Disk usage.

Quando a primeira etapa for concluída, clicar no próximo colchetes.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. On the left, a file browser shows a directory named 'sample\_data' containing files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. A blue arrow points from the 'sample\_data' folder to a text box. The main area shows two code cells. The first cell, labeled '[1]', contains the command 'pip install python-igraph'. The output shows the installation of 'python-igraph-0.8.2' and 'texttable-1.6.2'. The second cell, labeled '...', contains the command 'pip install geopandas'. The output shows the installation of 'geopandas', 'pyproj', 'fiona', and 'click-plugins'. A red-bordered text box at the bottom left contains the following text:

Ao clicar no próximo colchetes, aparecerá um número no colchetes anterior em que a execução foi finalizada. Esses números aparecerão em sequência até o fim do código.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook titled "SaveShapefilesFromLatLongFile.ipynb". The left sidebar shows a file explorer with a folder named "sample\_data" containing "id\_long\_lat.txt" and "matriz\_teste.txt". The main area displays code cells. The first cell is a comment: "Successfully installed click-plugins-1.1.1 cligj-0.5.0 fiona-1.8.13.post1 geopandas-0.7.0 munch-2.5.0 pyproj-2.6.1.post1". The second cell contains Python code for imports and file reading. A blue arrow points from a red-bordered text box to the play button above the second code cell. The third cell shows the output of the code as a table.

Clicar no colchetes seguinte e repetir isso até o fim do código.

```
[ ] filename = 'id_long_lat'
sep = ','
try:
    codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
except:
    codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
codes
```

	id	long	lat
0	1	-46.446433	-23.666145
1	2	-46.566063	-23.626141
2	3	-46.350597	-23.761635

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel:** Shows a directory structure with 'sample\_data' containing 'id\_long\_lat.txt' and 'matriz\_teste.txt'. A blue arrow points from the 'id\_long\_lat.txt' file to the code cell below.
- Code Cell [3]:** Contains the following code:

```
%matplotlib inline
## It is to avoiding to open an extra window showing maps
import geopandas as gpd
import pandas as pd
from igraph import Graph
import numpy as np
from statistics import mean
import math
```
- Code Cell [4]:** Contains the following code:

```
filename = 'id_long_lat'
sep = ','
try:
    codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
except:
    codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
codes
```
- Output:** A table with 3 rows and 4 columns (id, long, lat):

	id	long	lat
0	1	1	5
1	2	2	5
2	3	2	3
- RAM/Disk:** Shows 76.61 GB available.

Perceba que o número 3 apareceu no terceiro colchetes. Atente-se para não pular nenhum colchetes.



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook titled "SaveShapefilesFromLatLongFile.ipynb". The interface includes a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the notebook is as follows:

```
[3] from statistics import mean
import math

filename = 'id_long_lat'
sep = ','
try:
    codes = pd.read_csv(filename+'.txt', delim_whitespace=True, header= 0, encoding = "ISO-8859-1")
except:
    codes = pd.read_csv(filename+'.csv', delimiter=sep, header= 0, encoding = "ISO-8859-1")
codes
```

The output of the code is a table with 5 rows and 4 columns:

	id	long	lat
0	1	1	5
1	2	2	5
2	3	2	3
3	4	3	2
4	5	5	1

The notebook also shows the following code in the next cell:

```
[ ] from shapely.geometry import Point
geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

A red box highlights the text: "Ao executar o quarto colchetes, será possível visualizar a entrada com os pontos e suas respectivas latitude e longitude." A blue arrow points from this box to the play button above the code cell.

Ao executar o quarto colchetes, será possível visualizar a entrada com os pontos e suas respectivas latitude e longitude.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk

Editing

Files

Upload Refresh Mount Drive

sample\_data

id\_long\_lat.txt

matriz\_teste.txt

```
[4] 0 1 1 5
    1 2 2 5
    2 3 2 3
    3 4 3 2
    4 5 5 1
```

```
from shapely.geometry import Point
geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

```
[ ] crs = {'proj': 'latlong', 'ellps': 'WGS84', 'datum': 'WGS84', 'no_defs': True}

geo_data = gpd.GeoDataFrame(codes, crs = crs, geometry = geometry)
geo_data
```

	id	long	lat	geometry
0	1	-46.446433	-23.666145	POINT (-46.44643 -23.66615)
1	2	-46.566063	-23.626141	POINT (-46.56606 -23.62614)
2	3	-46.350597	-23.761635	POINT (-46.35060 -23.76164)
3	4	-46.550792	-23.812989	POINT (-46.55079 -23.81299)

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help Saving...

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
- id\_long\_lat.txt
- matriz\_teste.txt

```
[5] from shapely.geometry import Point
     geometry = [Point(x) for x in zip(codes.long, codes.lat)]
```

▶ crs = {'proj': 'latlong', 'ellps': 'WGS84', 'datum': 'WGS84', 'no\_defs': True}

```
geo_data = gpd.GeoDataFrame(codes, crs = crs, geometry = geometry)
geo_data
```

	id	long	lat	geometry
0	1	1	5	POINT (1.00000 5.00000)
1	2	2	5	POINT (2.00000 5.00000)
2	3	2	3	POINT (2.00000 3.00000)
3	4	3	2	POINT (3.00000 2.00000)
4	5	5	1	POINT (5.00000 1.00000)

```
[ ] import os

     dir = './'
     if not os.path.exists(dir):
         os.makedirs(dir)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. The top bar shows the notebook title "SaveShapefilesFromLatLongFile.ipynb" and various utility icons. The left sidebar contains a file explorer with a folder named "sample\_data" and files "id\_long\_lat.txt" and "matriz\_teste.txt". The main area shows a code cell with the following Python code:

```
import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)

geo_data.to_file(dir + 'CENTROIDES.shp')

geo_data.plot(figsize=(15,8), alpha=0.5)
```

Below the code, the output shows a scatter plot with two blue circular points. The y-axis is labeled from 3.5 to 5.0. The first point is at approximately (3.0, 5.0) and the second is at approximately (5.0, 5.0). The plot title is "<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4569942898>".

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. On the left, a file explorer shows a directory named 'sample\_data' containing two files: 'id\_long\_lat.txt' and 'matriz\_teste.txt'. The main area shows a code cell with a play button icon and a scatter plot. The plot has a white background with a black border. The x-axis is labeled from 1.0 to 5.0 in increments of 0.5. The y-axis is labeled from 1.0 to 5.0 in increments of 0.5. Five blue circular data points are plotted at the following coordinates: (1.0, 5.0), (2.0, 5.0), (2.0, 3.0), (3.0, 2.0), and (5.0, 1.0). Above the plot, the text '<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4569942898>' is visible. To the right of the plot, a blue arrow points to a red-bordered text box containing the instruction: 'Ao executar o sétimo colchetes, será possível visualizar o shape com os pontos.' The top of the interface shows the notebook title 'SaveShapefilesFromLatLongFile.ipynb', a star icon, and navigation menus. On the right side, there are icons for 'Comment', 'Share', and 'Editing', along with RAM and Disk usage indicators.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface. The top bar includes the Colab logo, the file name 'SaveShapefilesFromLatLongFile.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', along with the text 'All changes saved'. On the right side of the top bar are icons for 'Comment', 'Share', a settings gear, and a user profile picture. Below the top bar is a toolbar with 'RAM' and 'Disk' usage indicators, and an 'Editing' mode indicator.

The left sidebar shows a 'Files' panel with a tree view containing a folder 'sample\_data' and files 'id\_long\_lat.txt' and 'matriz\_teste.txt'. A blue arrow points from the 'matriz\_teste.txt' file to the play button in the code cell.

The main area contains a code cell with a play button (a black circle with a white right-pointing triangle) and the following Python code:

```
filename = 'matriz_teste'  
sep=','  
try:  
    mtx=pd.read_csv(filename+'.txt', delim_whitespace=True, header= None)  
except:  
    mtx=pd.read_csv(filename+'.csv', delimiter=sep, header= None)  
mtx
```

Below the code, the output is displayed as a 5x5 matrix:

	0	1	2	3	4
0	5	9	8	0	0
1	0	6	8	0	0
2	0	0	7	7	0
3	0	0	0	8	6
4	0	0	0	0	9

Below the matrix, the output of the command `type(mtx)` is shown as `pandas.core.frame.DataFrame`.

A red-bordered box on the left contains the text: "Ao executar o oitavo colchetes, será possível visualizar a matriz de fluxo de entrada."

At the bottom left, a 'Disk' usage indicator shows '76.61 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

Upload Refresh Mount Drive

sample\_data  
id\_long\_lat.txt  
matriz\_teste.txt

+ Code + Text

```
[8] 1 0 6 8 0 0  
    2 0 0 7 7 0  
    3 0 0 0 8 6  
    4 0 0 0 0 9
```

type(mtx)

```
pandas.core.frame.DataFrame
```

## Simetrizando matriz

```
[ ] def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help



Comment



Share



Editing



Files

Upload Refresh Mount Drive

<>

..

sample\_data

CENTROIDES.cpg

CENTROIDES.dbf

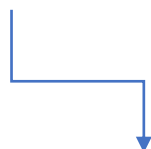
CENTROIDES.prj

CENTROIDES.shp

CENTROIDES.shx

id\_long\_lat.txt

matriz\_teste.txt



Perceba que o shape  
dos centroides passa  
a aparecer aqui.

Disk 76.61 GB available

+ Code + Text

pandas.core.frame.DataFrame

## Simetrizando matriz

```
def simetric_matrix(dados):  
    for i in range(len(dados.columns)):  
        for j in range(len(dados.columns)):  
            if(dados.iloc[i,j]>dados.iloc[j,i]):  
                dados.iloc[j,i] = dados.iloc[i,j]  
            elif (dados.iloc[j,i]>dados.iloc[i,j]):  
                dados.iloc[i,j] = dados.iloc[j,i]  
    return dados
```

```
[ ] mtx = simetric_matrix(mtx)  
    mtx
```

```
0 1 2 3 4  
0 5 9 8 0 0  
1 9 6 8 0 0  
2 8 8 7 7 0  
3 0 0 7 8 6
```



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

SaveShapefilesFromLatLongFile.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Última edição em 1 de julho

Comentário Compartilhar

Índice

- Symmetrizing the matrix
  - Creating graph from adj matrix
  - Seção

```
[ ] #Symmetrizing by sum
def simetric_matrix_sum(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = original.iloc[i,j] + original.iloc[j,i]
    return dados

#Symmetrizing by average
def simetric_matrix_avg(dados):
    original=dados.copy()
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            dados.iloc[i,j] = (original.iloc[i,j] + original.iloc[j,i])/2
    return dados

[ ] #You have to choose which function you would like to apply

#mtx = simetric_matrix_higher(mtx)
#mtx = simetric_matrix_lower(mtx)
mtx = simetric_matrix_sum(mtx)
#mtx = simetric_matrix_avg(mtx)
```

Nesta parte, será definido a forma com que a matriz será simetrizada. As opções são simetrização pelo maior, menor, soma ou média dos dois valores. É necessário “descomentar”, ou seja, tirar o símbolo ‘#’ da linha que deve ser executada, mantendo as demais com o símbolo ‘#’.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[10] def simetric_matrix(dados):
    for i in range(len(dados.columns)):
        for j in range(len(dados.columns)):
            if(dados.iloc[i,j]>dados.iloc[j,i]):
                dados.iloc[j,i] = dados.iloc[i,j]
            elif (dados.iloc[j,i]>dados.iloc[i,j]):
                dados.iloc[i,j] = dados.iloc[j,i]
    return dados
```

mtx = simetric\_matrix(mtx)

mtx

	0	1	2	3	4
0	5	9	8	0	0
1	9	6	8	0	0
2	8	8	7	7	0
3	0	0	7	8	6
4	0	0	0	6	9

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. The top bar includes the 'co' logo, the file name 'SaveShapefilesFromLatLongFile.ipynb', and navigation options like 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are buttons for 'Comment', 'Share', and a user profile icon. Below the top bar, the left sidebar shows a file explorer with a folder named 'sample\_data' containing files like 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area is titled 'Creating graph from adj matrix' and contains three code cells. The first cell starts with 'from igraph import ADJ\_MAX' and defines a graph. The second cell defines a function 'get\_lat\_long\_from\_city(id)'. The third cell creates a 'limits\_edges' DataFrame and populates it with graph edge data. A 'limits edges' prompt is visible at the bottom of the code area.

```
from igraph import ADJ_MAX

A=np.matrix(mtx.values)
graph = Graph.Weighted_Adjacency(A.tolist(), mode=ADJ_MAX, attr="weight", loops=False)
graph = graph.as_undirected()

[ ] def get_lat_long_from_city(id): #For each code return its geometry
    cidade = codes.iloc[id][0]
    idx = geo_data[geo_data['id']==cidade].index
    row = geo_data.iloc[idx]['geometry']

    return (row.total_bounds)

[ ] limits_edges = pd.DataFrame(columns=['source', 'target'])

for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
    'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
    'weight': edge['weight']},
    ignore_index=True)

limits edges
```

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. The top bar includes the file name 'SaveShapefilesFromLatLongFile.ipynb' and navigation options like 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are icons for 'Comment', 'Share', and a user profile. Below the top bar, a 'Files' sidebar on the left shows a directory structure with a folder named 'sample\_data' containing files like 'CENTROIDES.cpg', 'CENTROIDES.dbf', 'CENTROIDES.prj', 'CENTROIDES.shp', 'CENTROIDES.shx', 'id\_long\_lat.txt', and 'matriz\_teste.txt'. The main area of the notebook is titled 'Creating graph from adj matrix' and contains the following Python code:

```
[12] from igraph import ADJ_MAX

A=np.matrix(mtx.values)
graph = Graph.Weighted_Adjacency(A.tolist(), mode=ADJ_MAX, attr="weight", loops=False)
graph = graph.as_undirected()

def get_lat_long_from_city(id): #For each code return its geometry
    cidade = codes.iloc[id][0]
    idx = geo_data[geo_data['id']==cidade].index
    row = geo_data.iloc[idx]['geometry']

    return (row.total_bounds)

[ ] limits_edges = pd.DataFrame(columns=['source', 'target'])

for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
    'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
    'weight': edge['weight']},
    ignore_index=True)

limits_edges
```

At the bottom left, a disk usage indicator shows '76.61 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot displays a Jupyter Notebook interface. At the top, the title bar reads "SaveShapefilesFromLatLongFile.ipynb" with a star icon. Below the title bar is a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the title bar, there are icons for "Comment", "Share", a settings gear, and a user profile picture. Below the title bar, the interface is divided into three main sections: a file explorer on the left, a code editor in the center, and a status bar at the bottom.

The file explorer on the left shows a directory structure with a folder named "sample\_data" containing several files: "CENTROIDES.cpg", "CENTROIDES.dbf", "CENTROIDES.prj", "CENTROIDES.shp", "CENTROIDES.shx", "id\_long\_lat.txt", and "matriz\_teste.txt".

The code editor in the center is titled "Creating graph from adj matrix" and contains the following Python code:

```
[12] from igraph import ADJ_MAX

A=np.matrix(mtx.values)
graph = Graph.Weighted_Adjacency(A.tolist(), mode=ADJ_MAX, attr="weight", loops=False)
graph = graph.as_undirected()

[13] def get_lat_long_from_city(id): #For each code return its geometry
    cidade = codes.iloc[id][0]
    idx = geo_data[geo_data['id']==cidade].index
    row = geo_data.iloc[idx]['geometry']

    return (row.total_bounds)

limits_edges = pd.DataFrame(columns=['source', 'target'])

for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
    'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
    'weight': edge['weight']},
    ignore_index=True)

limits_edges
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Editing

RAM Disk

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[13] return (row.total_bounds)
```

limits\_edges = pd.DataFrame(columns=['source', 'target'])

```
for edge in graph.es: #For each edge in the graph, add coordinates of source and target nodes as a list
    limits_edges = limits_edges.append({'source': (get_lat_long_from_city(edge.source)[0], get_lat_long_from_city(edge.source)[1]),
                                        'target': (get_lat_long_from_city(edge.target)[0], get_lat_long_from_city(edge.target)[1]),
                                        'weight': edge['weight']},
                                        ignore_index=True)
```

limits\_edges

	source	target	weight
0	(1.0, 5.0)	(2.0, 5.0)	9.0
1	(1.0, 5.0)	(2.0, 3.0)	8.0
2	(2.0, 5.0)	(2.0, 3.0)	8.0
3	(2.0, 3.0)	(3.0, 2.0)	7.0
4	(3.0, 2.0)	(5.0, 1.0)	6.0

```
[ ] from shapely.geometry import Point, LineString

lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment

Share



Files

Upload Refresh Mount Drive

sample\_data  
CENTROIDES.cpg  
CENTROIDES.dbf  
CENTROIDES.prj  
CENTROIDES.shp  
CENTROIDES.shx  
id\_long\_lat.txt  
matriz\_teste.txt

+ Code + Text

RAM  
Disk

Editing

[14]

```
4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
from shapely.geometry import Point, LineString
```

```
lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

```
[ ] edges_graph = gpd.GeoDataFrame(limits_edges, crs=geo_data.crs, geometry = lines)
```

```
[ ] edges_graph.crs
```

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>
```

Name: unknown

Axis Info [ellipsoidal]:

- lon[east]: Longitude (degree)

- lat[north]: Latitude (degree)

Area of Use:

- undefined

Datum: World Geodetic System 1984

- Ellipsoid: WGS 84

- Prime Meridian: Greenwich

```
[ ] edges_graph
```



source

target weight

geometry

Disk 76.61 GB available



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[14] 4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
[15] from shapely.geometry import Point, LineString
```

```
lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

```
edges_graph = gpd.GeoDataFrame(limits_edges, crs=geo_data.crs, geometry = lines)
```

```
[ ] edges_graph.crs
```

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>
```

```
Name: unknown
```

```
Axis Info [ellipsoidal]:
```

```
- lon[east]: Longitude (degree)
```

```
- lat[north]: Latitude (degree)
```

```
Area of Use:
```

```
- undefined
```

```
Datum: World Geodetic System 1984
```

```
- Ellipsoid: WGS 84
```

```
- Prime Meridian: Greenwich
```

```
[ ] edges_graph
```

source	target	weight	geometry
--------	--------	--------	----------

Disk 76.61 GB available



# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment

Share



RAM  
Disk

Editing

Files

Upload Refresh Mount Drive

..  
sample\_data  
CENTROIDES.cpg  
CENTROIDES.dbf  
CENTROIDES.prj  
CENTROIDES.shp  
CENTROIDES.shx  
id\_long\_lat.txt  
matriz\_teste.txt

+ Code + Text

```
[14] 4 (3.0, 2.0) (5.0, 1.0) 6.0
```

```
[15] from shapely.geometry import Point, LineString  
lines = [LineString(xy) for xy in zip(limits_edges.source, limits_edges.target)]
```

```
[16] edges_graph = gpd.GeoDataFrame(limits_edges, crs=geo_data.crs, geometry = lines)
```

edges\_graph.crs

```
<Geographic 2D CRS: +proj=latlong +ellps=WGS84 +datum=WGS84 +no_defs + ...>  
Name: unknown  
Axis Info [ellipsoidal]:  
- lon[east]: Longitude (degree)  
- lat[north]: Latitude (degree)  
Area of Use:  
- undefined  
Datum: World Geodetic System 1984  
- Ellipsoid: WGS 84  
- Prime Meridian: Greenwich
```

edges\_graph



source

target weight

geometry

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

- Continuar clicando no “play” em cima de cada colchetes.

The screenshot shows a Jupyter Notebook interface with the following elements:

- Files Panel (Left):** Shows a directory structure with a folder named 'sample\_data' containing files: CENTROIDES.cpg, CENTROIDES.dbf, CENTROIDES.prj, CENTROIDES.shp, CENTROIDES.shx, id\_long\_lat.txt, and matriz\_teste.txt.
- Code Editor (Center):** Contains the following code:

```
[17] - undefined
Datum: World Geodetic System 1984
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich

edges_graph

source target weight geometry
0 (1.0, 5.0) (2.0, 5.0) 9.0 LINESTRING (1.00000 5.00000, 2.00000 5.00000)
1 (1.0, 5.0) (2.0, 3.0) 8.0 LINESTRING (1.00000 5.00000, 2.00000 3.00000)
2 (2.0, 5.0) (2.0, 3.0) 8.0 LINESTRING (2.00000 5.00000, 2.00000 3.00000)
3 (2.0, 3.0) (3.0, 2.0) 7.0 LINESTRING (2.00000 3.00000, 3.00000 2.00000)
4 (3.0, 2.0) (5.0, 1.0) 6.0 LINESTRING (3.00000 2.00000, 5.00000 1.00000)

[ ] edges_graph = edges_graph[['geometry']]
type(edges_graph)

geopandas.geodataframe.GeoDataFrame

[ ] import os

dir = './'
```
- Output (Right):** Shows the output of the code execution, including the table of graph edges and the type of the resulting object: `geopandas.geodataframe.GeoDataFrame`.
- Interface Elements:** Includes a top navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A right sidebar contains 'Comment', 'Share', and 'Editing' options. A status bar at the bottom indicates 'Disk 76.61 GB available'.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

+ Code + Text

- Prime Meridian: Greenwich

[18] edges\_graph

	source	target	weight	geometry
0	(1.0, 5.0)	(2.0, 5.0)	9.0	LINESTRING (1.00000 5.00000, 2.00000 5.00000)
1	(1.0, 5.0)	(2.0, 3.0)	8.0	LINESTRING (1.00000 5.00000, 2.00000 3.00000)
2	(2.0, 5.0)	(2.0, 3.0)	8.0	LINESTRING (2.00000 5.00000, 2.00000 3.00000)
3	(2.0, 3.0)	(3.0, 2.0)	7.0	LINESTRING (2.00000 3.00000, 3.00000 2.00000)
4	(3.0, 2.0)	(5.0, 1.0)	6.0	LINESTRING (3.00000 2.00000, 5.00000 1.00000)

edges\_graph = edges\_graph[['geometry']]  
type(edges\_graph)

geopandas.geodataframe.GeoDataFrame

```
[ ] import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)
```

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share Settings Profile

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[19] edges_graph = edges_graph[['geometry']]
type(edges_graph)

geopandas.geodataframe.GeoDataFrame

import os

dir = './'
if not os.path.exists(dir):
    os.makedirs(dir)

edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')

[ ] centroides = gpd.read_file('CENTROIDES.shp')
edges = gpd.read_file('EDGES_SHAPEFILE.shp')

base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)

<matplotlib.axes._subplots.AxesSubplot at 0x7f8e49acb198>
```

A small plot window showing a blue dot at approximately (-23.625, -23.625) and two red lines extending from it to the right and down-right. The y-axis is labeled with -23.625 and -23.650.

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



- Continuar clicando no “play” em cima de cada colchetes.

SaveShapefilesFromLatLongFile.ipynb

File Edit View Insert Runtime Tools Help

RAM Disk Editing

Files

- Upload Refresh Mount Drive
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

```
[20] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')
```

```
centroides = gpd.read_file('CENTROIDES.shp')  
edges = gpd.read_file('EDGES_SHAPEFILE.shp')
```

```
base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)  
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4566b59128>

Aqui será possível visualizar o shape com as arestas que representam o fluxo entre os pontos.

Disk 76.61 GB available

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais



SaveShapefilesFromLatLongFile.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 7:00 PM

Comment Share Settings Profile

Files

Upload Refresh Mount Drive

<>

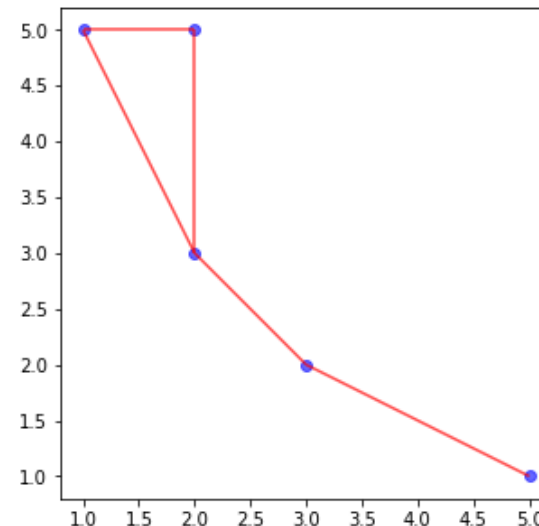
- ..
- sample\_data
  - CENTROIDES.cpg
  - CENTROIDES.dbf
  - CENTROIDES.prj
  - CENTROIDES.shp
  - CENTROIDES.shx
  - EDGES\_SHAPEFILE.cpg
  - EDGES\_SHAPEFILE.dbf
  - EDGES\_SHAPEFILE.prj
  - EDGES\_SHAPEFILE.shp
  - EDGES\_SHAPEFILE.shx
  - id\_long\_lat.txt
  - matriz\_teste.txt

+ Code + Text

RAM Disk Editing

```
[ ] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')  
  
[ ] centroides = gpd.read_file('CENTROIDES.shp')  
edges = gpd.read_file('EDGES_SHAPEFILE.shp')  
  
base = centroides.plot(color='blue', figsize=(10,5), alpha=0.6)  
edges.plot(ax=base, figsize=(10,5), color='red', alpha=0.7)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4566b59128>



Após alguns segundos os arquivos do shape de fluxo aparecerão abaixo dos arquivos do shape de pontos. Caso isso não aconteça automaticamente, basta atualizar a página!

Disk 76.61 GB available

Navigation icons

# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO

- Após finalizar a execução do código, basta fazer o download dos arquivos.



CO SaveShapefilesFromLatLongFile.ipynb ☆  
File Edit View Insert Runtime Tools Help Last saved at 7:00 PM

RAM  Disk  Editing

```
[ ] edges_graph.to_file(dir + 'EDGES_SHAPEFILE.shp')  
  
[ ] centroides = gpd.read_file('CENTROIDES.shp')  
edges = gpd.read_file('EDGES_SHAPEFILE.shp')  
  
edges.plot(color='blue', figsize=(10,5), alpha=0.6)  
centroides.plot(color='red', figsize=(10,5), color='red', alpha=0.7)
```

Files  
Upload Refresh Mount Drive  
sample\_data  
CENTROIDES.cpg  
CENTROIDES.dbf  
CENTROIDES.prj  
CENTROIDES.shp  
CENTROIDES.shx  
EDGES\_SHAPEFILE.cpg  
EDGES\_SHAPEFILE.dbf  
EDGES\_SHAPEFILE.prj  
EDGES\_SHAPEFILE.shp  
EDGES\_SHAPEFILE.shx  
id\_long\_lat.txt  
matriz\_teste.txt

Download  
Delete file  
Rename file  
Copy path  
Refresh

A plot showing a network of edges (blue lines) and centroids (red dots). The x-axis ranges from 1.0 to 5.0, and the y-axis ranges from 1.0 to 4.0. The plot shows a path of edges connecting four centroids. The first centroid is at approximately (1.5, 4.0), the second at (2.0, 3.0), the third at (3.0, 2.0), and the fourth at (5.0, 1.0). The edges are blue lines connecting these points in sequence.

Disk 76.61 GB available

Para fazer o download dos arquivos, basta sobrepor o cursor em cima de cada um deles, clicar nos três pontinhos que aparecerão e escolher a opção de realizar o download.

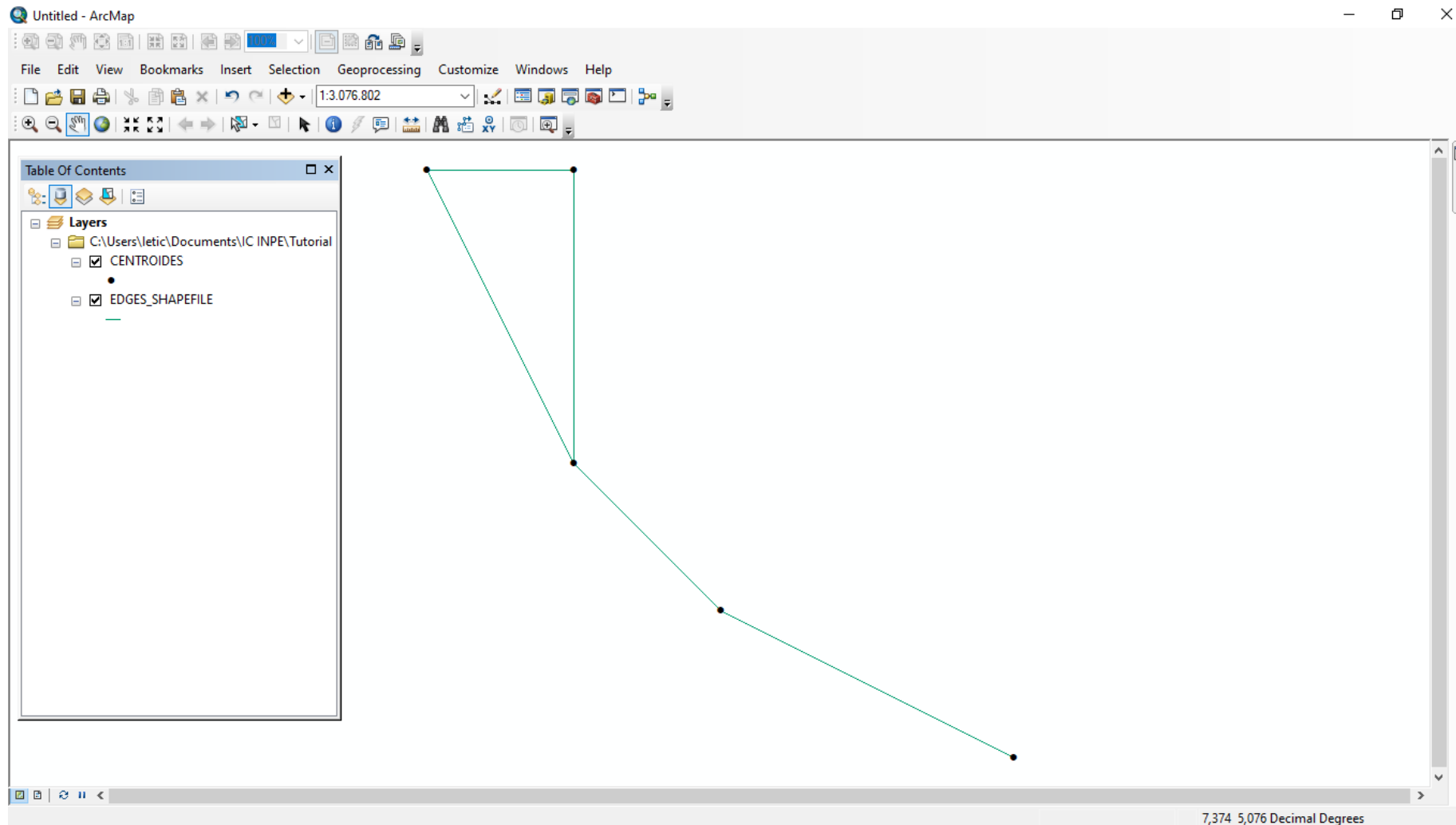


# TUTORIAL PARA ELABORAÇÃO DE (GEO)GRAFO



Laboratório de investigação  
em Sistemas Sócio-Ambientais

## PARTE 3 – Entendendo o dado de saída:



Após realizar o download dos shapes e adicioná-los no Arcgis, será possível observar os centroides e as arestas que os ligam baseado no fluxo de mobilidade. Assim, tem-se o (geo)grafo.