

# Trabalho de Conclusão do Curso **Ciência da Computação**



## **MINERAÇÃO DE DADOS APLICADO À REDE SOCIAL *TWITTER* UTILIZANDO A LINGUAGEM DE PROGRAMAÇÃO PYTHON**

Acadêmico: **Thiago Medeiros de Souza**  
Orientador: Valmei Abreu Júnior  
Co-Orientador: João Paulo de Lima Barbosa

Foz do Iguaçu - PR, Julho de 2016

# RESUMO

- Estudo e implementação de técnicas de *data mining*;
- Dados provenientes da rede social *Twitter* no dia 17 de abril de 2016;
- Utilização a linguagem de programação Python como ferramenta;
- Apresentação dos resultados obtidos através de gráficos, figuras e mapa.

# AGENDA

1. Introdução
  - 1.1. Justificativa
  - 1.2. Objetivo Geral
  - 1.3. Objetivos Específicos
2. Revisão Bibliográfica
3. Fundamentação Teórica
  - 3.1. KDD
  - 3.2. *Data Mining*
  - 3.3. Linguagem Python
4. Materiais
  - 4.1. Bibliotecas Python
  - 4.2. Protocolos de Acesso e Autenticação
  - 4.3. Rede Social *Twitter* e sua API
5. Metodologia
6. Implementação das Técnicas
  - 6.1. Coleta de Dados
  - 6.2. Análise de Dados
7. Análise dos Resultados
8. Conclusões e Trabalhos Futuros
9. Referências

# INTRODUÇÃO

- Popularização das redes sociais permite o aumento da quantidade de dados gerados;
- Estes dados podem ser estruturados ou não estruturados;
- Necessidade de minerar os dados para encontrar informação útil;
- Visualizar informações de diferentes maneiras para resultar em possibilidades ou probabilidades;
- Utilizar a linguagem Python como ferramenta para a mineração de dados.

# INTRODUÇÃO

## JUSTIFICATIVA

- A rede social *Twitter* é uma excelente escolha para coleta de dados;
- 9.100 *tweets* publicados a cada segundo;
- 5 bilhões de *tweets* por dia;
- Atual conjuntura do Brasil mostra-se propícia à mineração de dados de *tweets* publicados.

# INTRODUÇÃO

## OBJETIVO GERAL

Este trabalho tem como objetivo principal utilizar técnicas e algoritmos de *data mining*, para a análise e mineração de dados provenientes da rede social *Twitter*, utilizando os recursos e bibliotecas que a linguagem de programação Python possui.

# INTRODUÇÃO

## OBJETIVOS ESPECÍFICOS

- Identificar os conceitos sobre KDD e *data mining*;
- Descrever as técnicas de *data mining*;
- Explorar as funcionalidades das bibliotecas de mineração e visualização da linguagem Python;
- Examinar e utilizar a API da rede social *Twitter* para a coleta de dados;
- Encontrar padrões em dados provenientes do *Twitter*;
- Compreender e aplicar técnicas para apresentação e visualização de informações geográficas encontradas nos dados coletados;
- Apresentar testes e resultados obtidos da análise e mineração dos dados.

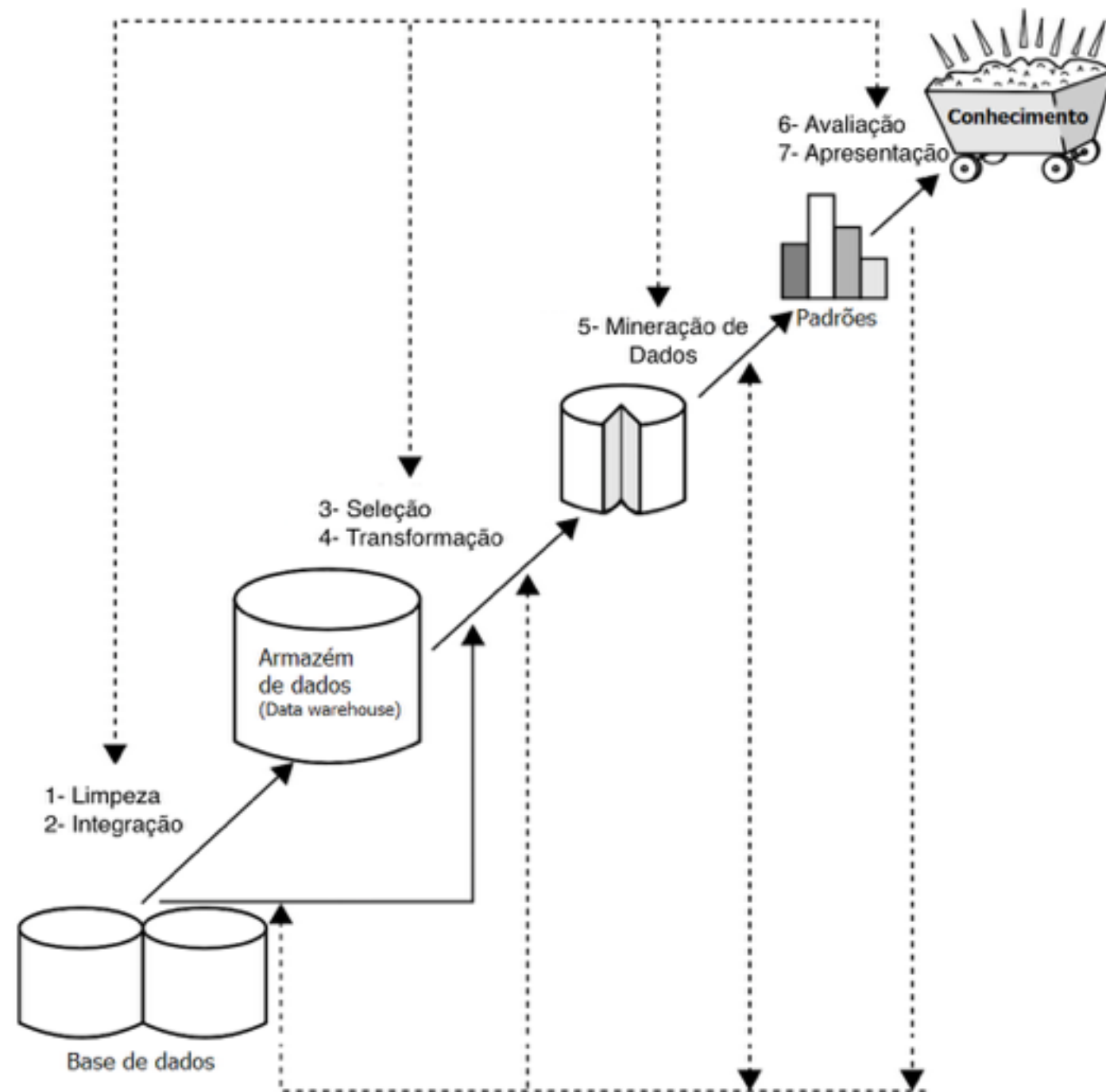
# REVISÃO BIBLIOGRÁFICA

- Análise de crédito bancário, Lemos (2003);
- Frequência de acesso em determinadas seções de páginas *web*, Silva, Boscarioli e Peres (2003).



# FUNDAMENTAÇÃO TEÓRICA

## DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS - KDD



Etapas do processo de KDD  
Fonte: Adaptado de Han et al. (2012)

*Data mining* é o processo de extração de informações de algum conjunto de dados para tomada de decisões.

---

- Classificação;
- Modelos de Relacionamento entre Variáveis;
- Análise de Agrupamento (*Cluster*);
- Sumarização;
- Modelo de Dependência;
- Regras de Associação;
- Análise de Séries Temporais.

# FUNDAMENTAÇÃO TEÓRICA

## LINGUAGEM PYTHON

- Sintaxe simples e clara;
- Extensível para outras linguagens de programação;
- É portátil a diversos sistemas operacionais;
- Solução de apenas uma única linguagem;
- Bibliotecas para a mineração de dados.

# MATERIAIS

## BIBLIOTECAS PYTHON

- *NumPy*;
- ***pandas***;
- ***matplotlib***;
- *SciPy*;
- ***IPython***;
- *Folium*;
- *NLTK*;
- *Word Cloud*;
- ***tweepy***.

# MATERIAIS

## BIBLIOTECAS PYTHON - *PANDAS*

```
[In [5]: obj = Series([4, 7, -5, 3])
```

```
[In [6]: obj
```

```
Out[6]:
```

```
0      4
1      7
2     -5
3      3
dtype: int64
```

Exemplo de uma *Series*  
*Fonte: McKinney (2013)*

```
[In [9]: frame
```

```
Out[9]:
```

	pop	state	year
0	1.5	Ohio	2000
1	1.7	Ohio	2001
2	3.6	Ohio	2002
3	2.4	Nevada	2001
4	2.9	Nevada	2002

Conteúdo de um *DataFrame* pelo interpretador *IPython*  
*Fonte: McKinney (2013)*

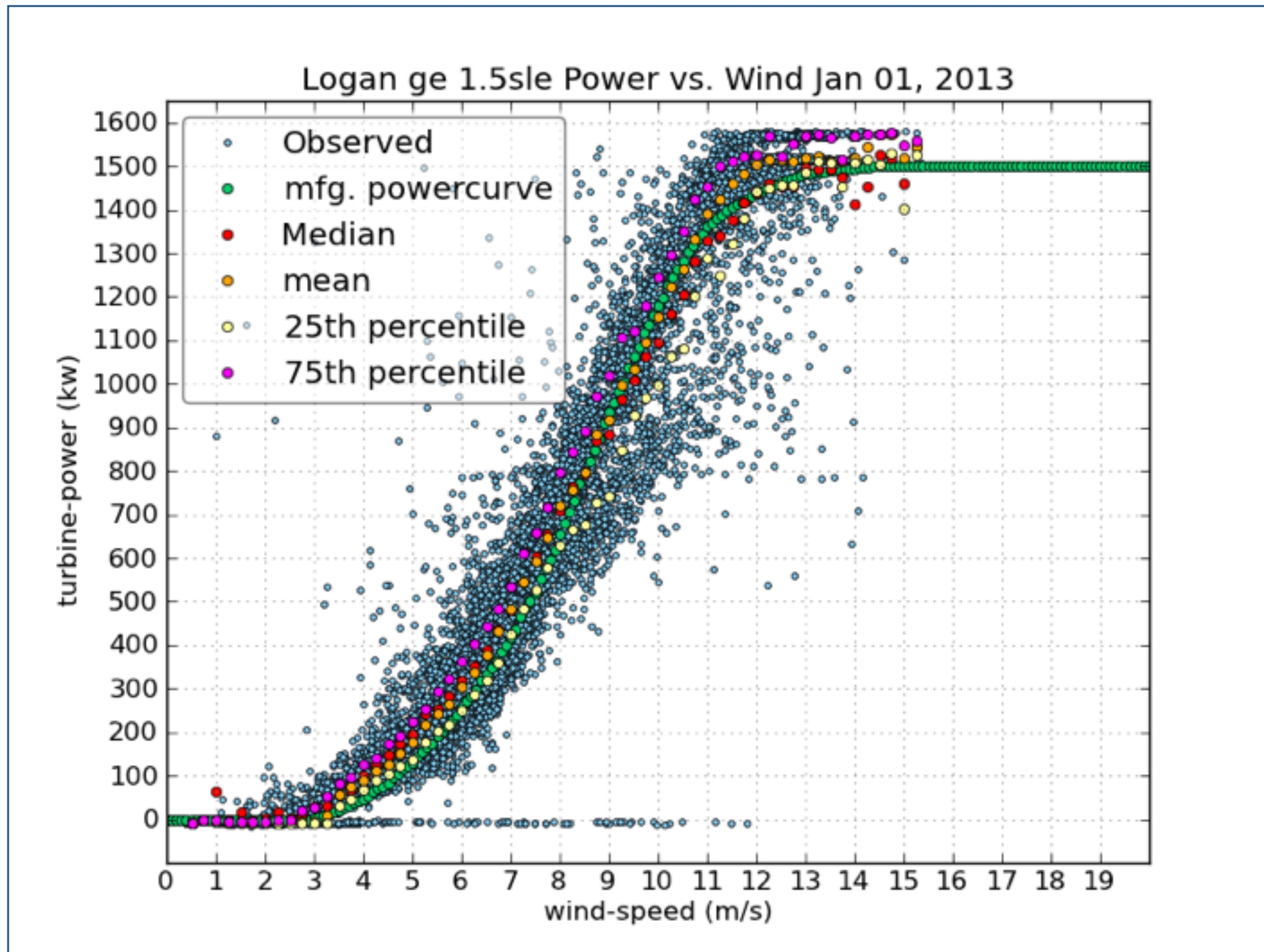
```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
        'year': [2000, 2001, 2002, 2001, 2002],
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
```

```
frame = DataFrame(data)
```

Criação de um *DataFrame*  
*Fonte: McKinney (2013)*

# MATERIAIS

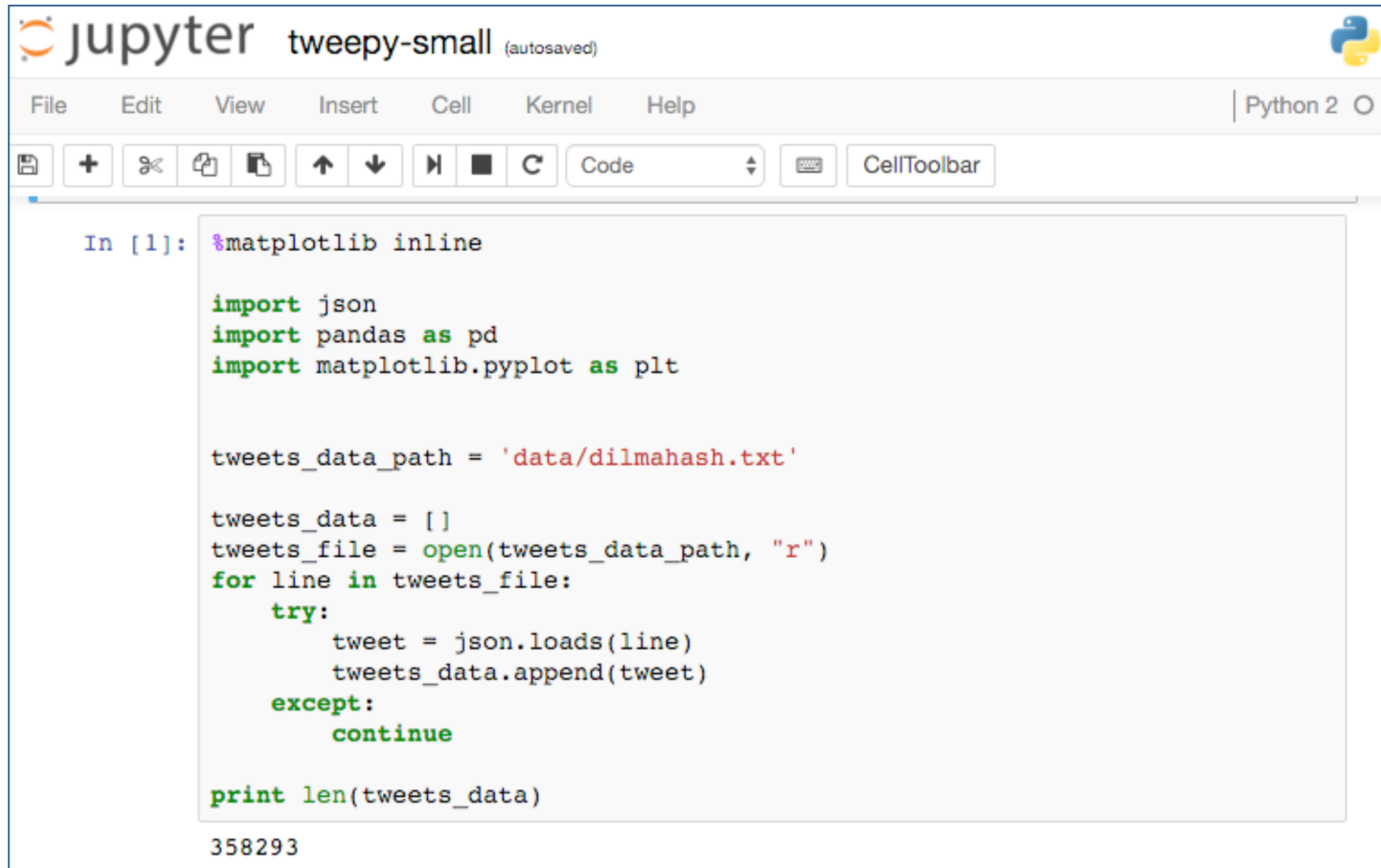
## BIBLIOTECAS PYTHON - *MATPLOTLIB*



Exemplo de um gráfico gerado pelo *matplotlib*  
Fonte: Wiener (2014)

# MATERIAIS

## BIBLIOTECAS PYTHON - INTERPRETADOR IPYTHON



The screenshot shows a Jupyter Notebook window titled "tweepy-small (autosaved)". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". Below the menu is a toolbar with icons for saving, adding, deleting, and running code. The main area displays a code cell with the following Python code:

```
In [1]: %matplotlib inline

import json
import pandas as pd
import matplotlib.pyplot as plt

tweets_data_path = 'data/dilmahash.txt'

tweets_data = []
tweets_file = open(tweets_data_path, "r")
for line in tweets_file:
    try:
        tweet = json.loads(line)
        tweets_data.append(tweet)
    except:
        continue

print len(tweets_data)
```

The output of the code cell is the number 358293.

Exemplo de uma página web do IPython Notebook  
Fonte: Wiener (2014)

# MATERIAIS

## PROTOCOLOS DE ACESSO E AUTENTICAÇÃO

- Interface de Programação de Aplicações - API:
    - Conectar componentes de um *software* com componentes de outros;
  - Arquitetura REST:
    - Requisição HTTP que recebe dados como resposta.
- 
- Protocolo OAuth:
    - Permite que um cliente *web* tenha acesso a um recurso protegido pelo seu dono em um servidor;
    - OAuth 1.0a;
    - OAuth 2.0.



# MATERIAIS

## REDE SOCIAL *TWITTER*

- Compartilhar ideias e experiências;
- Microblog que permite breve comunicação entre pessoas;
- Mensagens de no máximo 140 caracteres;
- Não requer aceitação mútua de conexão entre usuários.

# MATERIAIS

## REDE SOCIAL *TWITTER* - API

- *Tweets*;
- *Timelines* (linhas de tempo);
- Entidades:
  - *Hashtags*;
  - URLs;
  - Mídias;
- Lugares.

# MATERIAIS

## REDE SOCIAL TWITTER - API



Exemplo de um *tweet*  
Fonte: Twitter (2016a)

# MATERIAIS

## REDE SOCIAL *TWITTER* - API

- *Twitter's Search API* (API de Busca do *Twitter*);
- *Twitter's Streaming API* (API de *streaming* do *Twitter*);
- *Twitter Firehose*.

# METODOLOGIA

- Qualidade política:
    - Possibilidade de utilização de Python para a mineração de dados provenientes do *Twitter*, através das APIs disponibilizadas por este;
  - Qualidade formal:
    - Meios e formas usados na produção do trabalho.
- 

1. A existência de uma pergunta que se deseja responder;
2. A elaboração de um conjunto de passos que permitam chegar à resposta;
3. A indicação do grau de confiabilidade na resposta obtida.

# IMPLEMENTAÇÃO DAS TÉCNICAS

## COLETA DE DADOS

- Disponibilidade de informações de eventos em tempo real;
- Votação no Congresso Brasileiro sobre o Impeachment;
- Coleta dos dados referente a *hashtag* #ImpeachmentDay;
- Execução de um *script* Python por 12 horas;
- 358.293 *tweets* coletados.

# IMPLEMENTAÇÃO DAS TÉCNICAS

## COLETA DE DADOS

```
(tcc-py2)  
scripts git:(master) x  
> █
```

# IMPLEMENTAÇÃO DAS TÉCNICAS

## COLETA DE DADOS

```
scripts git:(master) x  
> python coletar-hashtags.py > ../data/coleta-impeachment.json
```

Execução do *script* para coleta de dados  
Fonte: Elaborado pelo autor



# IMPLEMENTAÇÃO DAS TÉCNICAS

## ANÁLISE DE DADOS

```
{"created_at": "Mon Apr 18 02:46:16 +0000 2016", "id": 721892}
{"created_at": "Mon Apr 18 02:46:16 +0000 2016", "id": 721892}
{"created_at": "Mon Apr 18 02:46:17 +0000 2016", "id": 721892}
{"limit": {"track": 11138, "timestamp_ms": "1460947577319"}}
{"created_at": "Mon Apr 18 02:46:17 +0000 2016", "id": 721892}
{"created_at": "Mon Apr 18 02:46:17 +0000 2016", "id": 721892}
{"created_at": "Mon Apr 18 02:46:17 +0000 2016", "id": 721892}
```

*Dirty Data* presente no arquivo coletado  
Fonte: Elaborado pelo autor

# IMPLEMENTAÇÃO DAS TÉCNICAS

## ANÁLISE DE DADOS

```
tcc git:(master) x  
> grep --invert-match '{"limit":' coleta-impeachment.json > small-data.json
```

Utilizando o comando *grep* para gerar um novo arquivo sem *dirty data*  
Fonte: Elaborado pelo autor

# IMPLEMENTAÇÃO DAS TÉCNICAS

## ANÁLISE DE DADOS

```
1 tweets_data_path = 'data/small-data.json'
2
3 tweets_data = []
4 tweets_file = open(tweets_data_path, "r")
5 for line in tweets_file:
6     try:
7         tweet = json.loads(line)
8         tweets_data.append(tweet)
9     except:
10        continue
11
12 tweets = pd.DataFrame()
13 print len(tweets_data)
```

Leitura do arquivo JSON  
Fonte: Elaborado pelo autor

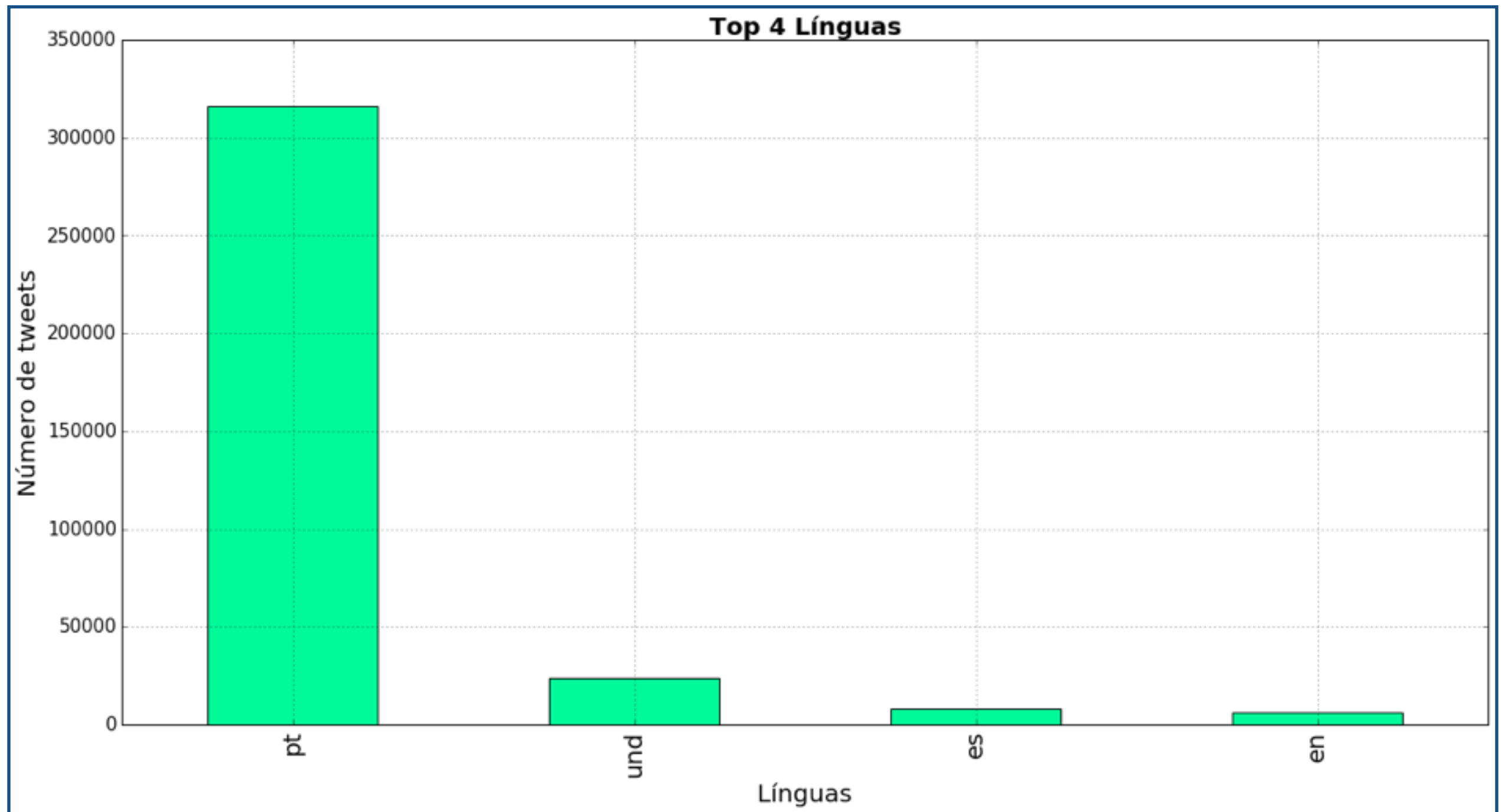
# IMPLEMENTAÇÃO DAS TÉCNICAS

## ANÁLISE DE DADOS

```
1 tweets['text'] = map(lambda tweet: tweet['text'], tweets_data)
2 tweets['lang'] = map(lambda tweet: tweet['lang'], tweets_data)
3 tweets['country'] = map(lambda tweet: tweet['place']['country']
4                         if tweet['place'] != None else None, tweets_data)
5
6 tweets_by_lang = tweets['lang'].value_counts()
7
8 fig, ax = plt.subplots(figsize=(20,10))
9 ax.tick_params(axis='x', labelsize=20)
10 ax.tick_params(axis='y', labelsize=15)
11 ax.set_xlabel('Idiomas'.decode('utf-8'), fontsize=20)
12 ax.set_ylabel('Numero de tweets'.decode('utf-8'), fontsize=20)
13 ax.set_title('Top 4 Idiomas'.decode('utf-8'),
14             fontsize=20, fontweight='bold')
15 tweets_by_lang[:4].plot(ax=ax, kind='bar', color='mediumspringgreen')
16 plt.grid()
```

Mapeamento de variáveis para um *DataFrame*  
Fonte: Elaborado pelo autor

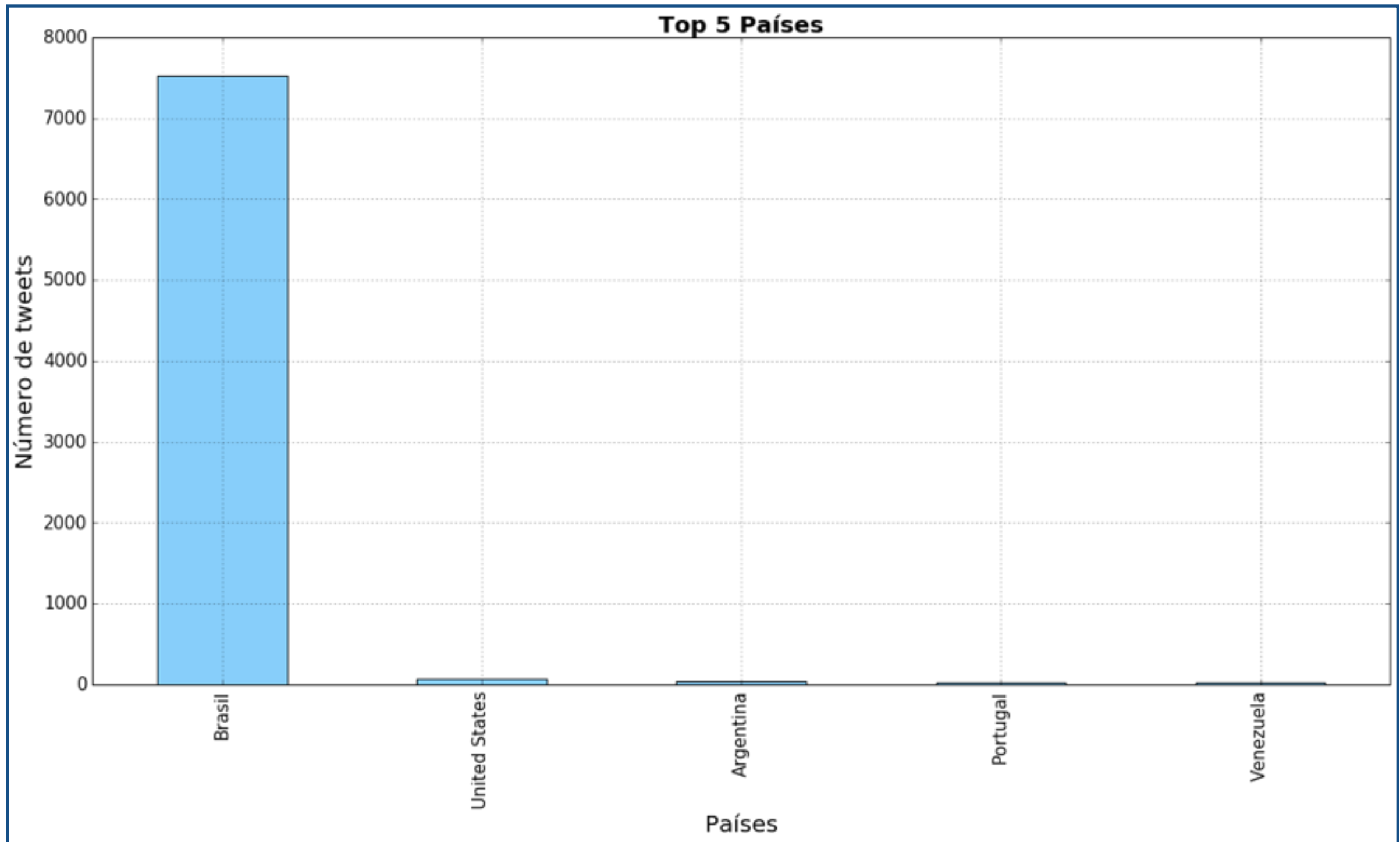
# ANÁLISE DOS RESULTADOS



Idiomas que mais realizaram *tweets*

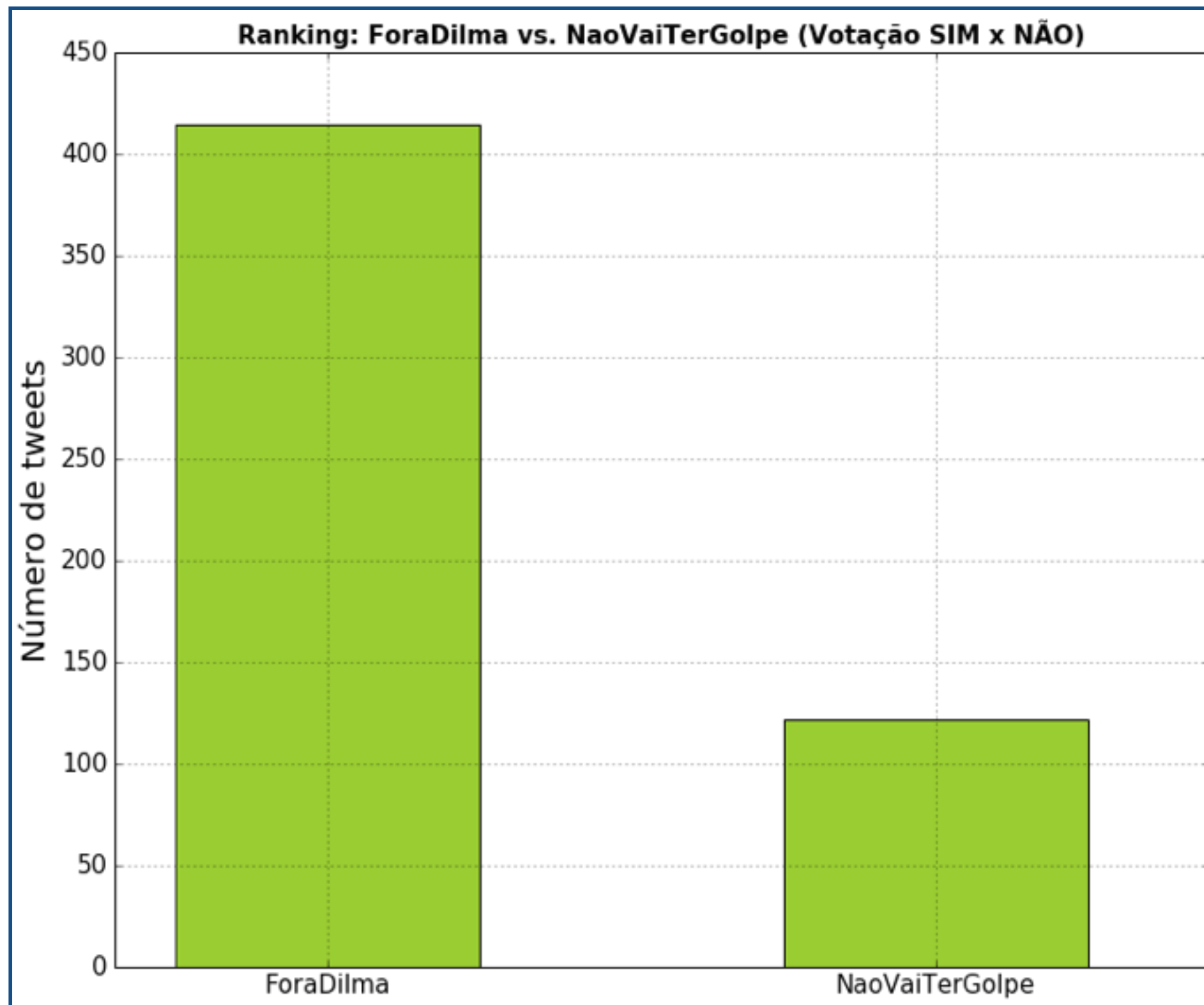
Fonte: Elaborado pelo autor

# ANÁLISE DOS RESULTADOS



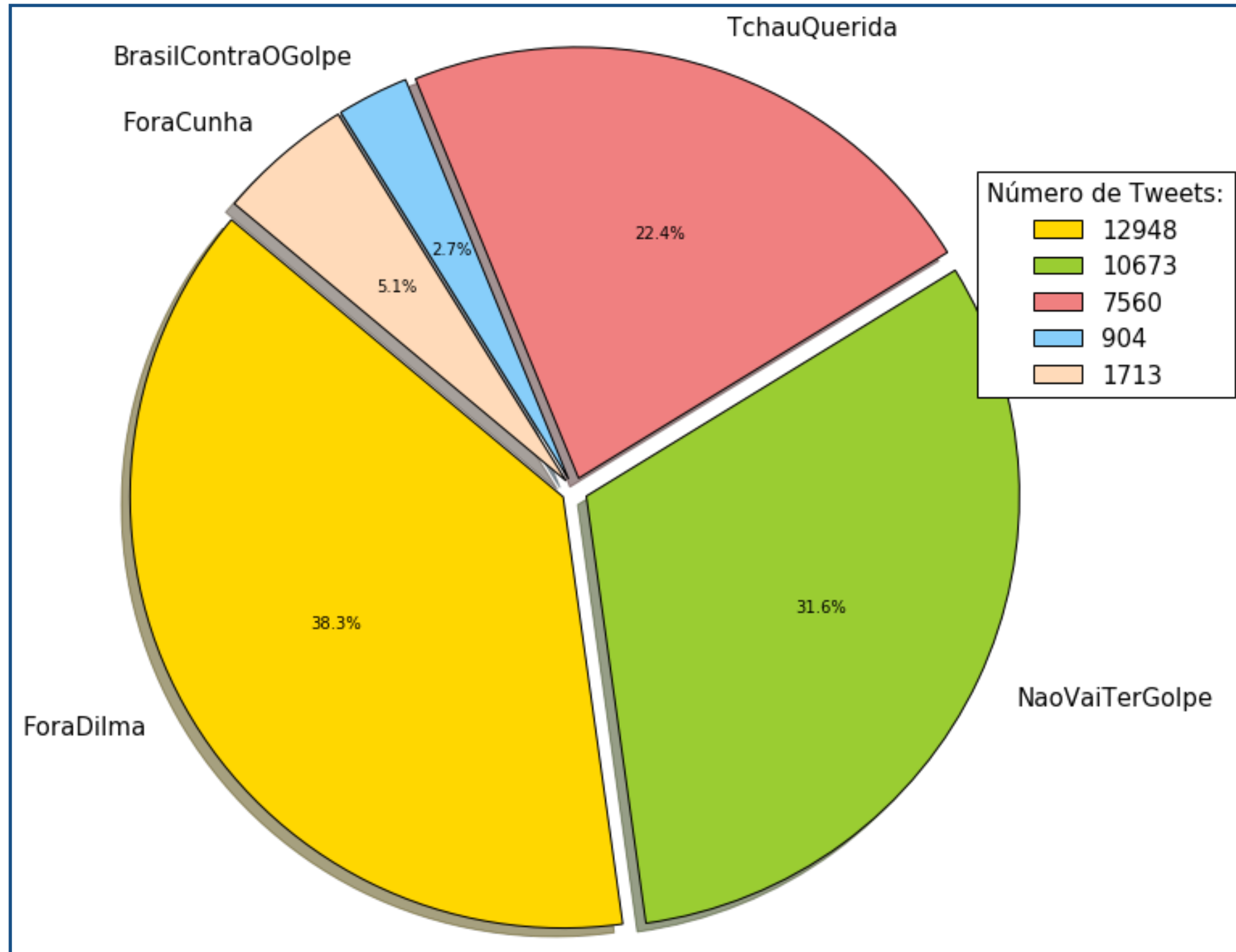
Países que mais realizaram *tweets*  
Fonte: Elaborado pelo autor

# ANÁLISE DOS RESULTADOS



Número de menções SIM e NÃO em *tweets*  
Fonte: Elaborado pelo autor

# ANÁLISE DOS RESULTADOS



*Hashtags com o maior número de tweets*

Fonte: Elaborado pelo autor



# ANÁLISE DOS RESULTADOS

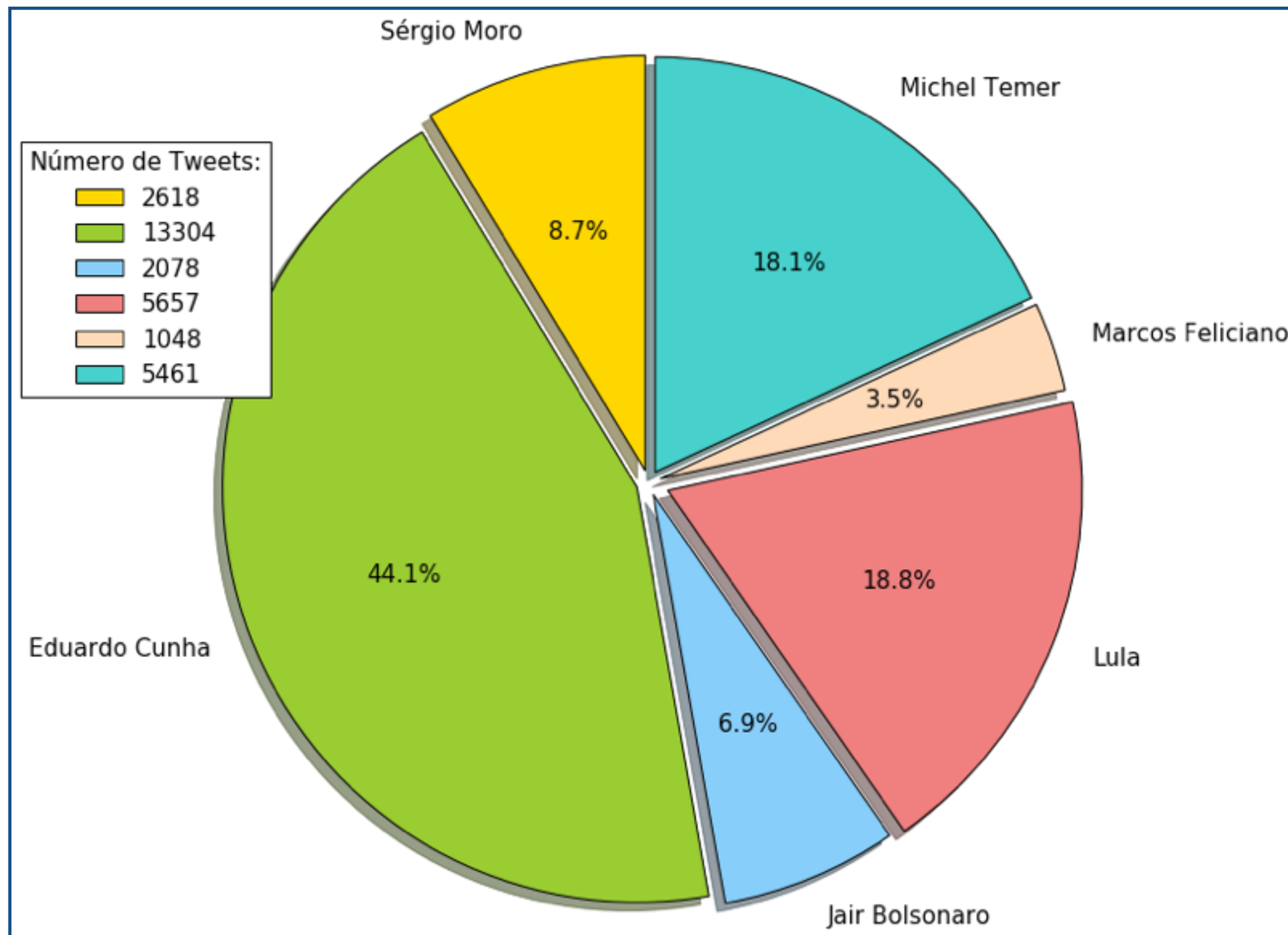


Gráfico em setores para figuras importantes  
Fonte: Elaborado pelo autor

## ANÁLISE DOS RESULTADOS



Word Cloud das 500 palavras mais mencionadas  
Fonte: Elaborado pelo autor

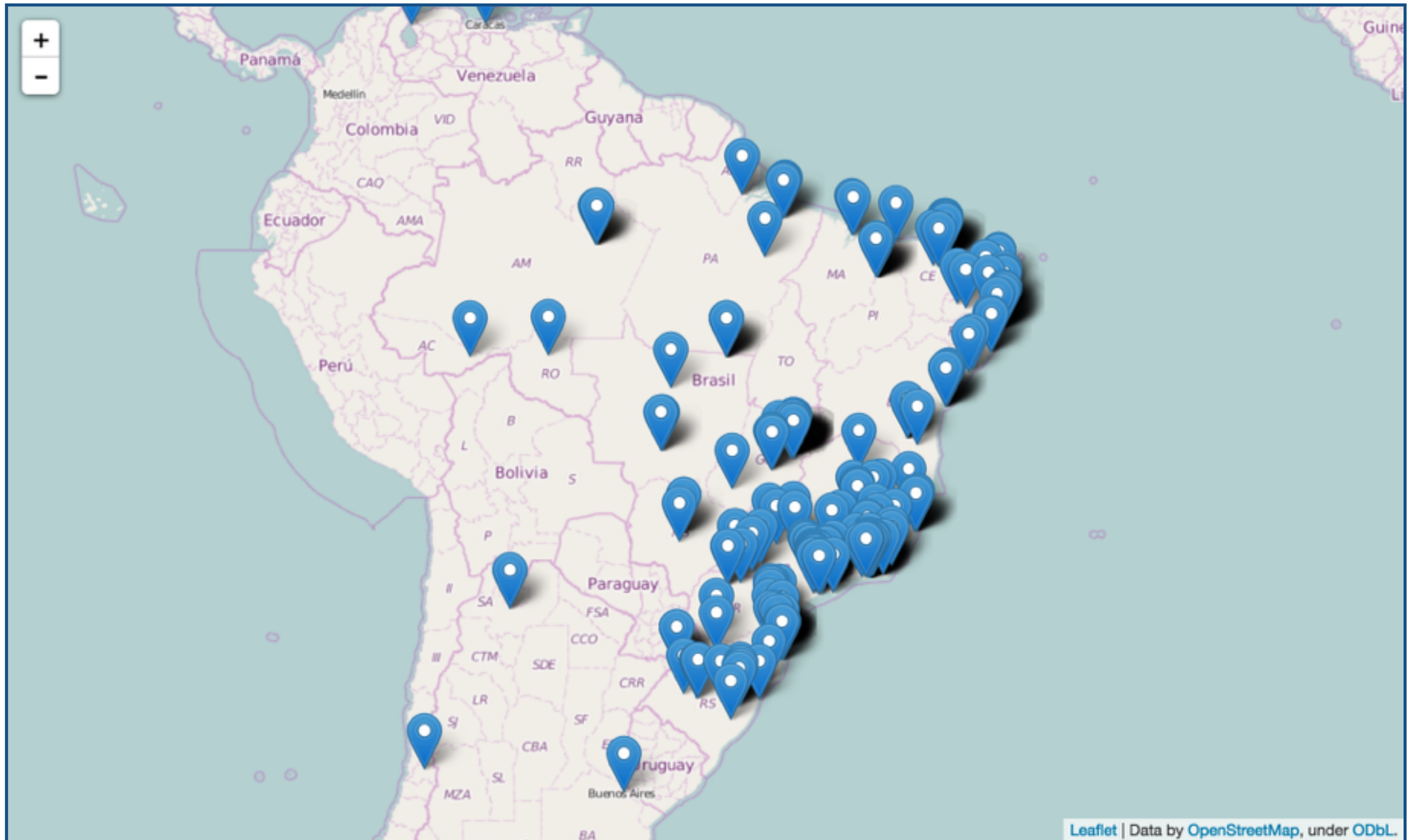


## ANÁLISE DOS RESULTADOS



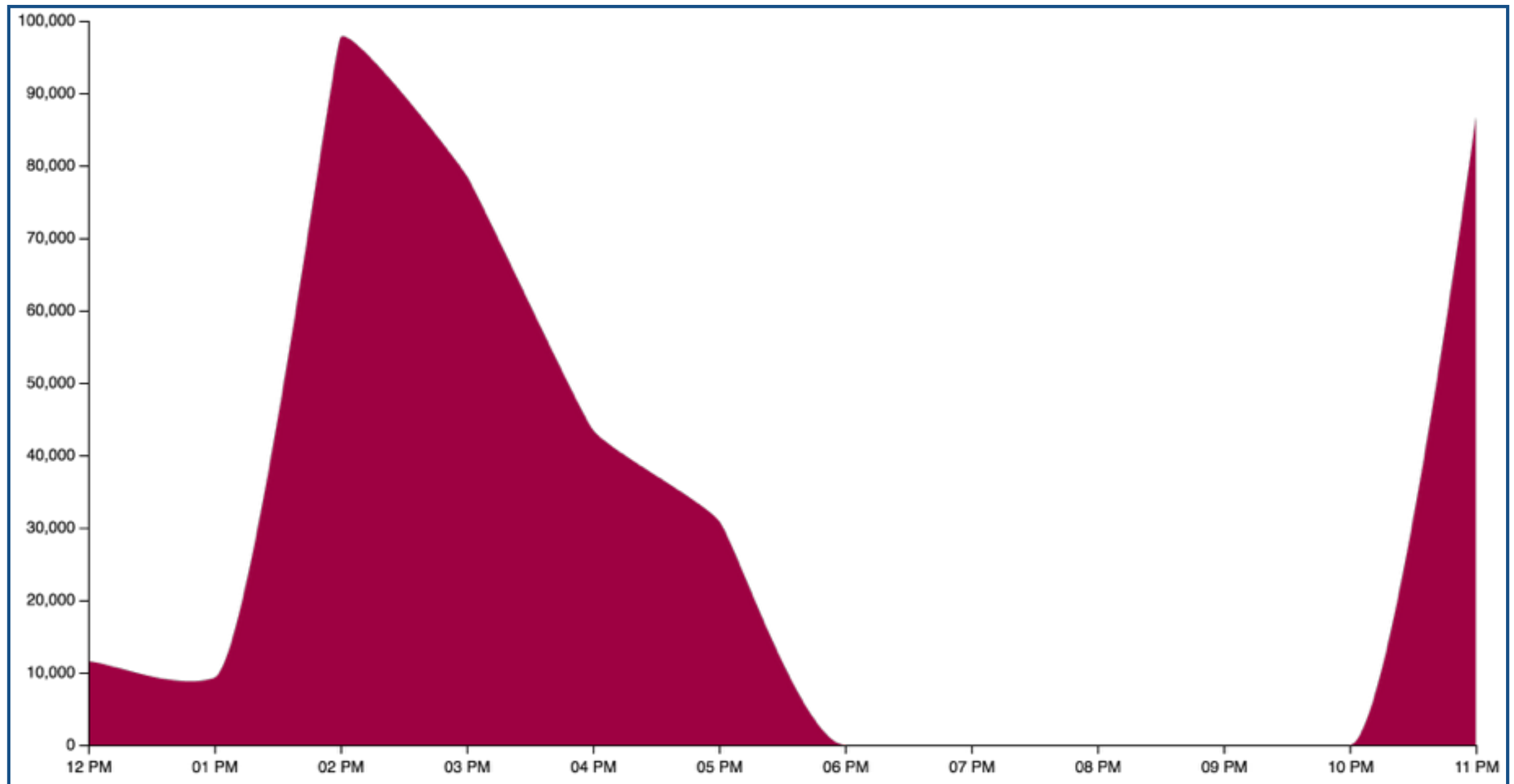
Word Cloud utilizando imagem como modelo  
Fonte: Elaborado pelo autor

# ANÁLISE DOS RESULTADOS



Distribuição geográfica de *tweets*  
Fonte: Elaborado pelo autor

# ANÁLISE DOS RESULTADOS



Publicação de *tweets* por hora  
Fonte: Elaborado pelo autor

# CONCLUSÕES E TRABALHOS FUTUROS

- Utilizar técnicas e algoritmos de *data mining* para analisar e minerar os dados provenientes do *Twitter*;
- Conhecimento dos conceitos de KDD e *data mining*;
- Linguagem Python útil para a coleta e limpeza dos dados;
- API limitada;
- Biblioteca *pandas* para a manipulação dos dados;
- Toda a implementação utilizou *IPython Notebook*;
- Apresentação de gráficos e mapa como resultados;

# CONCLUSÕES E TRABALHOS FUTUROS

- Dificuldade quanto a utilização da API do *LinkedIn*;
- Efetuar a mineração de dados em outras redes sociais;
- Estudo e implementação de técnicas de *machine learning* (aprendizado de máquina) para melhores resultados.



## REFERÊNCIAS

BRAIN, S. **Twitter Statistics**. 2016. Acesso em 20 de abril de 2016. Disponível em: <<http://www.statisticbrain.com/twitter-statistics/>>.

GOLDENBERG, M. **A arte de pesquisar**. [S.l.]: Editora Record, 2002.

HAN, J. *et al.* **Data Mining: Concepts and Techniques**. Elsevier, 2012.

KALDERO, N. **Why is Python a language of choice for data scientists?** 2015. Acesso em 27 de outubro de 2015. Disponível em: <<http://qr.ae/RkleiB>>.

LEMOS, E. P. **Análise de crédito bancário com o uso de data mining: redes neurais e árvores de decisão**. Tese (Doutorado) — Universidade Federal do Paraná, 2003.

MCKINNEY, W. **Python for Data Analysis**. O'Reilly, 2013



## REFERÊNCIAS

RUSSEL, M. A. **Mining the Social Web**. O'Reilly Media, 2013. ISBN 978-1-449-36761-9.

SILVA, M. P. da; BOSCARIOLI, C.; PERES, S. M. **Análise de logs da web por meio de técnicas de data mining**. 2003.

TWITTER. **Tweet Example**. 2016. Acesso em 24 de abril de 2016. Disponível em: <<https://twitter.com/unixstickers/status/724338974043574272>>.

TWITTER. **Twitter Documentation**. 2016. Acesso em 21 de abril de 2016. Disponível em: <<https://dev.twitter.com/overview/documentation>>.

WIENER, E. **Matplotlib tools**. 2014. Acesso em 27 de novembro de 2015. Disponível em: <<https://wiki.ucar.edu/display/ral/Matplotlib+Tools>>.