



Tweepy

Componentes do grupo

- Cassiano Kunsch das Neves
- Hélio Schimitt



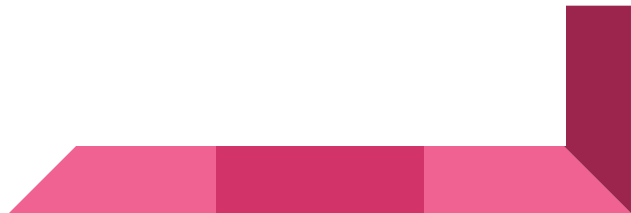
Objetivo

O trabalho terá como objetivo usar a API Tweepy, para mineração de dados da rede social Twitter. Os dados minerados são do mesmo tema do trabalho 1, esses dados serão armazenados no banco de dados para uma futura análise.



Tweepy, o que é?

Tweepy é uma API open-source, hospedado no GitHub e usa a linguagem de programação Python para se comunicar com a plataforma do Twitter e usar sua API.



Captura dos dados

- Para se capturar os twitters foi necessário criar uma conta e criar uma aplicação para poder ter acesso a obtenção dos dados;
- Após criar a conta foi criado o código para obter os dados(próximo slide);



```
class StdOutListener(tweepy.StreamListener):  
    ''' Handles data received from the stream. '''  
  
    def on_status(self, status):  
        global cont  
        global limit  
        cont = cont + 1  
        print(cont)  
  
        # Captura os twitters e coloca em uma lista  
        p_palavras.append(status.text)  
  
        if (cont == limit):  
            write_arq()  
  
    def on_error(self, status_code):  
        if status_code == 420:  
            # Retornando falso em on_data desliga  
            print("ERRO! Fechar feed.")  
            write_arq()
```

```

if __name__ == '__main__':
    # Chaves de autenticação para se ter acesso a conta
    consumer_key = 'ORMqhNG7HUwajEuorEgeGWTCL'
    consumer_secret = '3v2CNlc85hE55XiDhtUNwzssnyhrMSYFBWk2bwJokaSyK8XvhS'
    access_token = '808615970378055680-ActX61hReGiFAxkdlyDn018MMY6E03n'
    access_token_secret = '4m1u1UOV0UaKSZB2Viky7gViZeJMQMeInQzUIGaVxaM6v'

    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)

    listener = StdOutListener()

    # passando autenticação e o "escutador" como parametros
    stream = tweepy.Stream(auth, listener)

    # filtrando as frases pegas para somente aquelas que possuem a palavras abaixo
    stream.filter(track=['temperatura', 'transporte', 'caixa termica', 'orgao', 'doação', 'transplante'])

    # Chamando a funcao para pegar os twitters
    listener.on_status(tweepy.API(auth))

```

Nesse código é colocado a identificação da conta. Logo após instanciamos a classe responsável por fazer o controle de pegar o Twitters. Depois passamos a regra de Twitter que queremos, com determinadas palavras. Esses Twitters são guardados em uma lista para depois serem inseridos no banco.

Conexão com o banco?

Psycopg2

Para a conexão com o banco de dados, foi usada a biblioteca do Python, o Psycopg. Ele é o mais popular adaptador de banco de dados para PostgreSQL, com a linguagem de programação Python.



Gravação dos dados no banco

- Para a gravação foi feito um código em Python usando a biblioteca psycopg2 (próximo slide);



```
def salvar_bd():  
    # Conecta com o BD criado no postgres  
    DB = psycopg2.connect(host='localhost', dbname='SMO', user='postgres', password='Cc98576036')  
    # O cursor serve para fazer as operação, tipo: CREATE TABLE, SELECT e etc  
    cursor = DB.cursor()  
    # Utilizando o cursor pra criar a tabela  
    cursor.execute('CREATE TABLE IF NOT EXISTS table_twitter (id serial PRIMARY KEY, twitter varchar(144));')  
    for elemento in p_palavras:  
        # Utilizando o cursor pra inserir os dados na tabela  
        cursor.execute('INSERT INTO twitter (table_twitter) VALUES (%s);', (elemento))  
    # Fechando/finalizando o BD como um todo  
    DB.commit()  
    cursor.close()  
    DB.close()  
  
    # Tudo já foi analisado e salvo, então informar isso e fechar o sistema  
    print("Salvo com sucesso!")  
    sys.exit()
```

