

Trabalho de implementação : SIG (Sistemas de Informações Geográfica)

O seguinte trabalho consiste na implementação de uma análise geográfica usando bibliotecas de geoinformações, como GeoPandas e Folium , além de implementações em Bancos de Dados PostgreSQL e PostGis

Dupla:

- Luís Miguel Alves Meireles
- Evanildo Felipe Costa Macedo

Contexto e Problematização:

Desde o início da ocupação Portuguesa das terras brasileira (até então desconhecidas) a partir de 1530 a relação entre indígenas e europeus fora conturbada, e com o passar dos séculos se tornou cada vez mais complexa. Com a expansão das fronteiras brasileiras durante a União Ibérica (1580 a 1640) cada vez mais os povos indígenas de diversas etnias foram perdendo espaço em meio ao território da recém formada colônia brasileira (e posteriormente estado nação Brasil)

A demarcação de terras Indígenas tornou-se cada vez mais um "problema" recorrentes , principalmente com grandes apoios de movimentos sociais denunciando constantes invasões e exploração mineral nessas regiões

Aliado a isso , outro problema vem aparecendo mais recentemente , alertado muito por Ecologistas e ONG's ambientais , que se trata das Áreas de Proteção Ambiental , que hoje no Brasil somam mais de 294 pontos. Muitas dessas APA's (como também são conhecidas) representam o modo de sobrevivência de muitas populações no interior do brasil, visto que o turismo ecológico tem se tornado uma indústria cada vez maior no País.

Em decorrência disso , muito tem se debatido acerca da exploração mineral dentros de APA's e as consequências disso para as populações locais

Perguntas:

O seguinte trabalho tem por objetivo responder questões como:

- Como estão distribuídas as Terras Indígenas e APA's pelo território brasileiro?
- Como estão distribuídos os pontos de mineração pelo território brasileiro?
- Os pontos de mineração tem representado alguma "ameaça" aos citados acima?

Matérias de referência

- Mineração em terras indígenas da Amazônia aumentou 1.217% nos últimos 35 anos
- Deputas de MG tentam pela terceira vez alterar limites de área ambiental protegida
- Suspensa, mineradora opina sobre manejo de área de proteção
- Mineradora Sigma tem licença suspensa em Araçuaí; ainda há povos tradicionais em risco

Importação dos dados:

Grande parte dos dados utilizados nessa análise foram obtidos pela VER FONTES DOS DADOS

```
In [1]: # Import das bibliotecas necessárias:  
import numpy as np  
import pandas as pd  
import folium  
from folium import GeoJson  
import geopandas as gpd  
from shapely.wkb import loads  
import seaborn as sns  
import random  
from folium.plugins import FloatImage  
import psycopg2
```

```
from shapely.geometry import Point
from shapely import wkt
from branca.element import Figure
from shapely.wkb import loads
import math
```

```
In [2]: # Funções utilizadas durante a Análise:
def cria_df_sql(consulta_sql):
    host = "localhost"
    database = "trabalho"
    user = "postgres"
    password = "postgres"

    # Estabeleça a conexão
    conn = psycopg2.connect(
        host=host,
        database=database,
        user=user,
        password=password
    )

    sql_query = consulta_sql

    # Use o método read_sql da biblioteca pandas para ler os dados em um DataFrame
    df = pd.read_sql(sql_query, conn)

    conn.close()

    return df

def converte_wkb(df , coluna):
    df[coluna] = df[coluna].apply(lambda x: loads(bytes.fromhex(x)))
    df = gpd.GeoDataFrame(df, geometry=coluna)
    return df

def random_hex_color():
    # Gera três componentes de cor (R, G, B) aleatórios
    r = random.randint(0, 255)
    g = random.randint(0, 255)
    b = random.randint(0, 255)

    # Converte os componentes para formato hexadecimal
    hex_color = "#{:02X}{:02X}{:02X}".format(r, g, b)

    return hex_color

def min_max_normalization(column):
    return (column - column.min()) / (column.max() - column.min())

def intensidade_cor(valor):
    cod_hex = "#ffffff"

    # Atribua a cor com base no percentil
    if valor <= 0.0006197349193691231:
        cod_hex = '#00FF00'
    elif valor <= 0.0017029010524196597:
        cod_hex = '#7FFF00'
    elif valor <= 0.003743574918667695:
        cod_hex = '#FFFF00'
    elif valor <= 0.004397833229341628:
        cod_hex = '#FFD700'
    elif valor <= 0.008014400848862324:
        cod_hex = '#FA500'
    elif valor <= 0.01170739204400649:
        cod_hex = '#FF8C00'
    elif valor <= 0.03390639633713141:
        cod_hex = '#FF4500'
    elif valor <= 0.046603964769111016:
        cod_hex = '#FF0000'
    elif valor <= 0.1039151549088595:
        cod_hex = '#FF0000'
    else:
        cod_hex = '#990000'

    style = {
        'fillColor': cod_hex, # cor de preenchimento
        'color': 'red', # cor da linha de contorno
        'weight': 2, # espessura da linha
    }

    return style
```

Visualizações Gráficas e Análise Exploratória

Visualização geral (vendo todos os dados juntos)

```
In [3]: df = cria_df_sql("SELECT * FROM estados;")

# converte o DataFrame para um GeoDataFrame
df = converte_wkb(df , "geom")

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [4]: df_APAs = cria_df_sql("SELECT * FROM apas;")

# converte o DataFrame para um GeoDataFrame
df_APAs = converte_wkb(df_APAs , "geom")

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [5]: df_terras_indigenas = cria_df_sql("SELECT * FROM terrasindigenas;")

# converte o DataFrame para um GeoDataFrame
df_terras_indigenas = converte_wkb(df_terras_indigenas , "geom")

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [6]: df_mineracao = cria_df_sql("SELECT * FROM mineracao;")

# converte o DataFrame para um GeoDataFrame
df_mineracao = converte_wkb(df_mineracao , "geom")

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [7]: df_mineracao_apas_25km = cria_df_sql("SELECT m.* FROM mineracao m, apas WHERE ST_Contains(ST_Buffer(apas.geom, 0.25), m.geom) = 1")
df_mineracao_indigenas_25km = cria_df_sql("SELECT m.* FROM mineracao m, terrasindigenas t WHERE ST_Contains(ST_Buffer(t.geom, 0.25), m.geom) = 1")

df_mineracao_25km = pd.concat([df_mineracao_apas_25km, df_mineracao_indigenas_25km], ignore_index=True)

# converte o DataFrame para um GeoDataFrame
df_mineracao_25km = converte_wkb(df_mineracao_25km , "geom")

# Remover linhas duplicadas com base em todas as colunas
df_mineracao_25km = df_mineracao_25km.drop_duplicates()

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [8]: # Exiba os estados no mapa
gjson_estados = df.__geo_interface__
gjson_apas = df_APAs.__geo_interface__
gjson_terras_indigenas = df_terras_indigenas.__geo_interface__

# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                  zoom_start=4,
                  width=1280, height=720,
                  tiles = "cartodbpositron",
                  name = 'Positron'
                 )

#####
# Estilos no mapa de estados do Brasil
style_uf = {'fillColor': 'orange', #cor de preenchimento
            'color': 'orange', #cor da linha de contorno
            'weight': 2, #espessura da linha
           }

# plotando no mapa do brasil
folium.GeoJson(gjson_estados, name='Brasil' , style_function=lambda x:style_uf).add_to(mapa)

#####
# Estilos no mapa de APA:
style_apas = {'fillColor': 'green', #cor de preenchimento
              'color': 'green', #cor da linha de contorno
```

```

        'weight': 1, #espessura da linha
    }

# plotando no mapa as APA
folium.GeoJson(gjason_apas, name='Áreas de Proteção Ambiental', style_function=lambda x:style_apas).add_to(mapa)
#####
# Estilos no mapa de Terras Indígenas
style_terras_indigenas = {'fillColor': 'red', #cor de preenchimento
                           'color': 'red',#cor da linha de contorno
                           'weight': 1, #espessura da linha
                           }

# plotando Terras Indígenas
folium.GeoJson(gjason_terras_indigenas, name='Terras indígenas', style_function=lambda x:style_terras_indigenas).add_to(mapa)
#####

# Adiciona os pontos de mineração ao mapa
for index, row in df_mineracao.iterrows():
    folium.Circle(location=[row['geom'].y, row['geom'].x], radius=1000, color="#990073", fill=True).add_to(mapa)
#####

# Adiciona os pontos de mineração há 25km ao mapa
for index, row in df_mineracao_25km.iterrows():
    folium.Circle(location=[row['geom'].y, row['geom'].x], radius=1000, color="#ff4db5", fill=True).add_to(mapa)
#####

folium.LayerControl().add_to(mapa)

#####

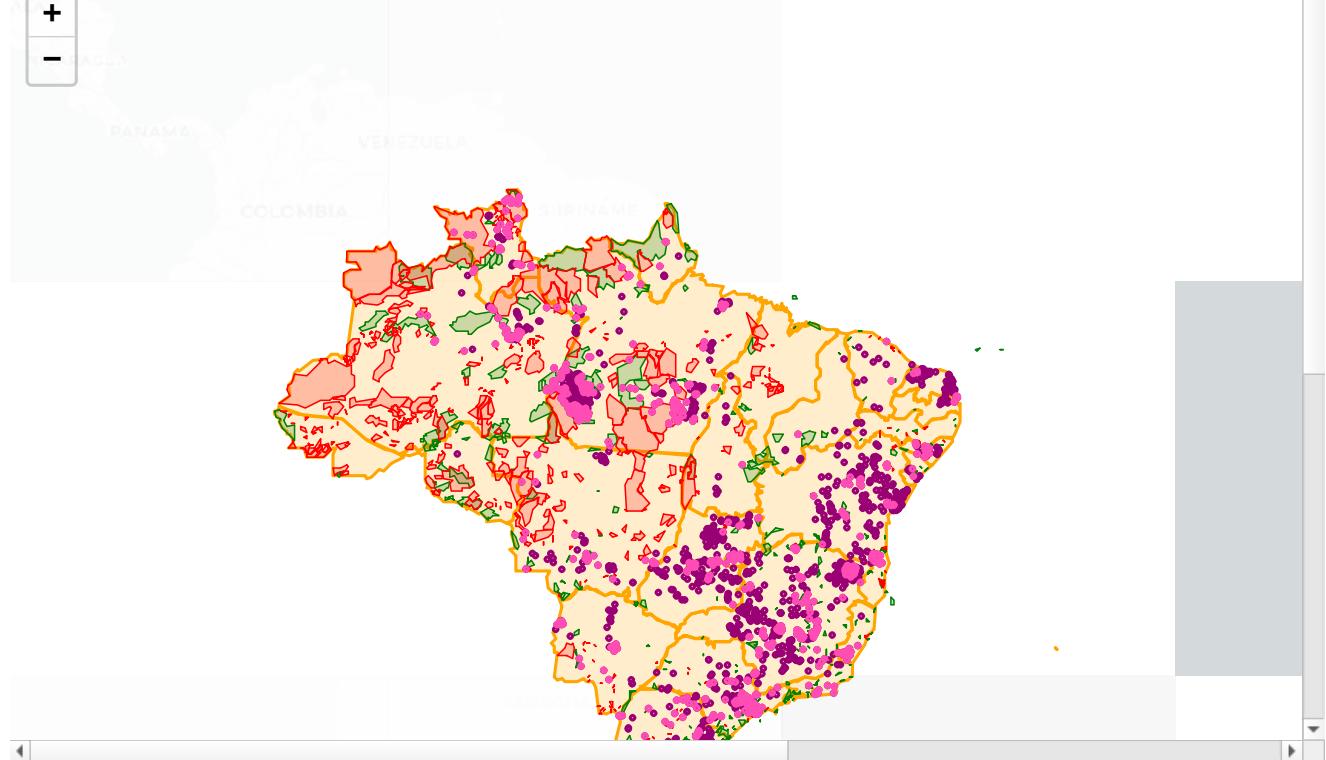
# Adicionar uma legenda com rótulos personalizados
legend_html = """
    <div style="position: fixed; bottom: 50px; left: 50px; width: 200px; height: 120px; border:2px solid grey; font-size:14px; background-color: white; ">
        <span>&ampnbsp Legenda:<br>
        </span>
        <span>&ampnbsp Brasil &ampnbsp <i class="fa fa-circle fa-1x" style="color:orange"></i><br>
        </span>
        <span>&ampnbsp Terras Indígenas &ampnbsp <i class="fa fa-circle fa-1x" style="color:red"></i><br>
        </span>
        <span>&ampnbsp Áreas de Proteção Ambiental &ampnbsp <i class="fa fa-circle fa-1x" style="color:green"></i><br>
        </span>
        <span>&ampnbsp Pontos de Mineração &ampnbsp <i class="fa fa-circle fa-1x" style="color:#990073"></i><br>
        </span>
        <span>&ampnbsp Pontos de Mineração há 25km de distância de uma APA ou Terra Indígena &ampnbsp <i class="fa fa-circle fa-1x" style="color:#ff4db5"></i>
    </div>
"""
#####
FloatImage(legend_html, bottom=100, left=50).add_to(mapa)

#####

# Salvando o mapa em html
mapa.save('mapa_completo.html')
mapa

```

Out[8]: Make this Notebook Trusted to load map: File -> Trust Notebook



Análise das Terras Indígenas

```
In [9]: df_terras_indigenas_uf = cria_df_sql("SELECT ti.* , uf.sigla_uf , uf.nm_regiao , uf.area_km2 FROM terrasindigenas ti JOIN unidade_federativa uf ON ti.sigla_uf = uf.sigla_uf")  
df_terras_indigenas_uf = converte_wkb(df_terras_indigenas_uf , "geom")
```

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)

```
In [10]: df_terras_indigenas_uf = pd.DataFrame(df_terras_indigenas_uf.groupby('sigla_uf')['geom'].apply(lambda x: x.area))  
df_terras_indigenas_uf['geom'] = (df_terras_indigenas_uf['geom'] * 10000).round(2)  
df_terras_indigenas_uf = df_terras_indigenas_uf.rename(columns={'geom': 'Área em km²'})  
df_terras_indigenas_uf = df_terras_indigenas_uf.sort_values(by='Área em km²', ascending=False)
```

```
In [11]: df_terras_indigenas_uf
```

```
Out[11]: Área em km²
```

sigla_uf	Área em km²
AM	140668.21
PA	111792.19
MT	87148.17
RO	27282.30
MA	14556.12
AC	10106.99
TO	8942.84
AP	4209.17
RR	2899.10
MS	1413.99
BA	1181.36
RS	979.96
PE	972.49
PR	852.58
SC	833.66
MG	620.08
SP	376.16
GO	346.55
ES	163.30
CE	112.15
AL	108.46
PB	103.54
SE	33.69
RJ	20.05

Nota-se que dos 5 estados da federação que mais tem terras indígenas , 3 encontram-se na região norte! Sendo que os outros 2 estados (MA e MT) constantemente tem crises de disputas de terras , além do MT ser um estado com grandes extensões de terras para utlização no agronegócio brasileiro

```
In [12]: df_estados = cria_df_sql("SELECT * FROM estados;")  
  
# converte o DataFrame para um GeoDataFrame  
df_estados = converte_wkb(df_estados , "geom")
```

```
C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy  
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n  
ot tested. Please consider using SQLAlchemy.  
df = pd.read_sql(sql_query, conn)
```

```
In [13]: # Proporção das Terras Indígenas  
print("Proporção das Terras Indígenas:")  
for _, row in df_estados.iterrows():  
    uf = (row.at['sigla_uf']).strip()  
    regiao = (row.at['nm_regiao']).strip()  
    area = round(float(row.at['area_km2']))  
  
    for index, row in df_terras_indigenas_uf.iterrows():  
        uf_apa = index  
        area_apa = round(float(row.at['Área em km²']))  
  
        if(uf_apa == uf):  
            proporcao_apa = round((area_apa/area)*100 , 2)  
            print(f"{uf} - {proporcao_apa}%")
```

Proporção das Terras Indígenas:

```
AC - 6.16%
AM - 9.02%
PA - 8.97%
AP - 2.95%
TO - 3.22%
MA - 4.42%
CE - 0.08%
PB - 0.18%
PE - 0.99%
AL - 0.39%
SE - 0.15%
BA - 0.21%
MG - 0.11%
ES - 0.35%
RJ - 0.05%
SP - 0.15%
PR - 0.43%
SC - 0.87%
RS - 0.35%
MS - 0.4%
MT - 9.65%
GO - 0.1%
RO - 11.47%
RR - 1.3%
```

Nos dados acima , temos o percentual de que as Terras indígenas fazem do todo da área de cada estado. Percebe-se que RO , AM e MT são os estados que tem mais percentual de áreas de terras indígenas em todo o seu território. Observe como apenas RO tem quase 11,5% de seu território composto de áreas indígenas

Análise das Áreas de Proteção Ambiental (APA):

```
In [14]: df_apa_uf = cria_df_sql("SELECT apa.* , uf.sigla_uf , uf.nm_regiao , uf.area_km2 FROM apa , estados uf WHERE ST_
df_apa_uf = converte_wkb(df_apa_uf , "geom")
```

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)

```
In [15]: df_apa_uf = pd.DataFrame(df_apa_uf.groupby('sigla_uf')['geom'].apply(lambda x: x.area.sum()))
df_apa_uf['geom'] = (df_apa_uf['geom'] * 10000).round(2)
df_apa_uf = df_apa_uf.rename(columns={'geom': 'Área em km²'})
df_apa_uf = df_apa_uf.sort_values(by='Área em km²', ascending=False)
```

```
In [16]: df_apa_uf
```

Out[16]: Área em km²

sigla_uf	Área em km ²
AM	61017.05
PA	50406.33
RO	10720.48
MG	7801.29
PI	6781.82
RR	5619.90
MT	5476.77
BA	5097.48
MS	2085.76
SP	1860.78
TO	1563.98
MA	1329.26
SC	1150.44
GO	956.59
RJ	878.05
PR	745.66
PE	670.43
RS	575.74
ES	568.26
CE	330.80
SE	113.00
DF	107.78
RN	78.65
AL	59.40
PB	23.38
AP	3.01

Agora visualizando os 5 maiores estados com Áreas de Proteção Ambiental , nota-se que 3 estão na região norte (e inclusive são os mesmos de terras indígenas) , sendo que os outros 2 estados (MG e PI) são estados com forte tradição de eco-turismo , PI com grandes sítios arqueológicos e MG com diversas cidades históricas muitos dependentes do turismo (Em MG se encontra um dos trechos de eco-turismo mais importantes do país: a Rota Imperial)

In [17]:

```
# Proporção das Áreas de Proteção Ambiental
print("Proporção das Áreas de Proteção Ambiental")
for _, row in df_estados.iterrows():
    uf = (row.at['sigla_uf']).strip()
    regiao = (row.at['nm_regiao']).strip()
    area = round(float(row.at['area_km2']))

    for index, row in df_apas_uf.iterrows():
        uf_apa = index
        area_apa = round(float(row.at['Área em km²']))

        if(uf_apa == uf):
            proporcao_apa = round((area_apa/area)*100 , 2)
            print(f"{uf} - {proporcao_apa}%")
```

Proporção das Áreas de Proteção Ambiental

AM	- 3.91%
PA	- 4.05%
AP	- 0.0%
TO	- 0.56%
MA	- 0.4%
PI	- 2.69%
CE	- 0.22%
RN	- 0.15%
PB	- 0.04%
PE	- 0.68%
AL	- 0.21%
SE	- 0.52%
BA	- 0.9%
MG	- 1.33%
ES	- 1.23%
RJ	- 2.01%
SP	- 0.75%
PR	- 0.37%
SC	- 1.2%
RS	- 0.2%
MS	- 0.58%
MT	- 0.61%
GO	- 0.28%
DF	- 1.87%
RO	- 4.51%
RR	- 2.51%

Seguindo os mesmos dados acima , podemos observar que RO , AM e PA se destancam com suas APA's.

Apesar de serem inferiores aos dados de Terras Indígenas, se somados representam um valor consideravelmente alto de territórios dos estados (por exemplo: RO sozinha tem quase 16% de todo o seu território composto por APA's e Terras Indígenas)

Visualizando as APA's e Pontos de Mineração:

In [18]:

```
# Exiba os estados no mapa
gjson_estados = df.__geo_interface__
gjson_apas = df_APAs.__geo_interface__
gjson_terrass_indigenas = df_terrass_indigenas.__geo_interface__

# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                  zoom_start=4,
                  width=1280, height=720,
                  tiles = "cartodbpositron",
                  name = 'Positron'
                 )

#####
# Estilos no mapa de estados do Brasil
style_uf = {'fillColor': 'orange', #cor de preenchimento
            'color': 'orange',#cor da linha de contorno
            'weight': 2, #espessura da linha
           }

# plotando no mapa do brasil
folium.GeoJson(gjson_estados, name='Brasil' , style_function=lambda x:style_uf).add_to(mapa)

#####
# Estilos no mapa de APA:
style_apas = {'fillColor': 'green', #cor de preenchimento
              'color': 'green',#cor da linha de contorno
              'weight': 1, #espessura da linha
             }

# plotando no mapa as APA
folium.GeoJson(gjson_apas, name='Áreas de Proteção Ambiental' , style_function=lambda x:style_apas).add_to(mapa)

#####
# Estilos no mapa de Terras Indígenas
style_terrass_indigenas = {'fillColor': 'red', #cor de preenchimento
                           'color': 'red',#cor da linha de contorno
                           'weight': 1, #espessura da linha
                          }

# plotando Terras Indígenas
folium.GeoJson(gjson_terrass_indigenas, name='Terrass indígenas' , style_function=lambda x:style_terrass_indigenas)
```

```

#####
folium.LayerControl().add_to(mapa)
#####

# Adicionar uma legenda com rótulos personalizados
legend_html = """
    <div style="
        position: fixed;
        bottom: 50px; left: 50px; width: 200px; height: 120px;
        border:2px solid grey; font-size:14px;
        background-color: white;
    ">

        <span>
            &nbsp; Legenda:<br>
        </span>

        <span>
            &nbsp; Brasil &nbsp; <i class="fa fa-circle fa-1x" style="color:orange"></i><br>
        </span>

        <span>
            &nbsp; Terras Indígenas &nbsp; <i class="fa fa-circle fa-1x" style="color:red"></i><br>
        </span>

        <span>
            &nbsp; Áreas de Proteção Ambiental &nbsp; <i class="fa fa-circle fa-1x" style="color:green"></i><br>
        </span>

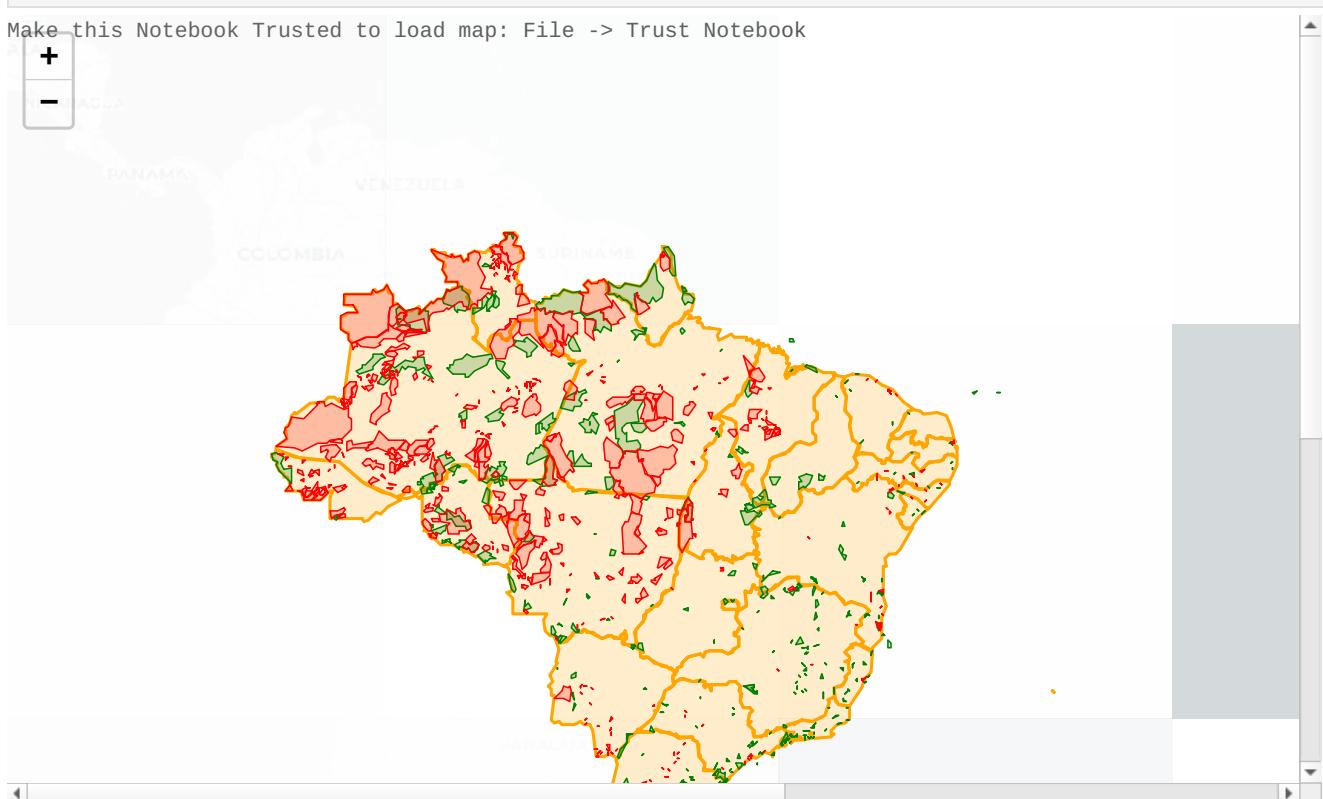
    </div>
"""
#####

FloatImage(legend_html, bottom=100, left=50).add_to(mapa)
#####

# Salvando o mapa em html
mapa.save('mapa_terras_indigenas_apa.html')
mapa

```

Out[18]: Make this Notebook Trusted to load map: File -> Trust Notebook



Análise dos Pontos de Mineração:

```
In [19]: df_mineracao = cria_df_sql("SELECT * FROM mineracao;")

# converte o DataFrame para um GeoDataFrame
df_mineracao = converte_wkb(df_mineracao, "geom")
```

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.

```
df = pd.read_sql(sql_query, conn)
```

```
In [20]: # Exiba os estados no mapa
#gjson_mineracao = df_mineracao.__geo_interface__

# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                   zoom_start=4,
                   width=1280, height=720,
                   tiles = "cartodbpositron",
                   name = 'Positron'
                  )

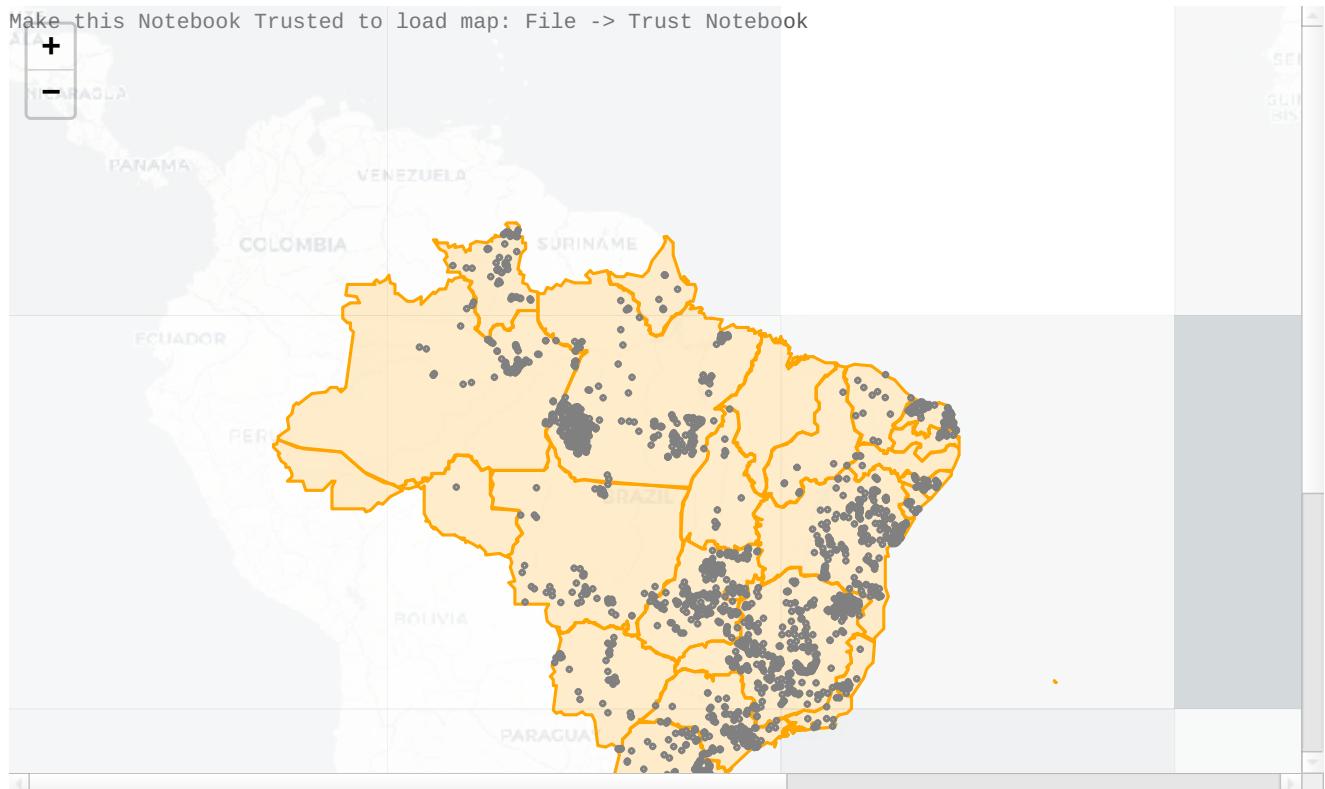
#####
# Estilos no mapa de estados do Brasil
style_uf = {'fillColor': 'orange', #cor de preenchimento
            'color': 'orange', #cor da linha de contorno
            'weight': 2, #espessura da linha
           }

# plotando no mapa do brasil
folium.GeoJson(gjson_estados, name='Brasil' , style_function=lambda x:style_uf).add_to(mapa)

#####
# Adiciona os pontos ao mapa
for index, row in df_mineracao.iterrows():
    folium.Circle(location=[row['geom'].y, row['geom'].x], radius=1000, color='gray', fill=True).add_to(mapa)

# Salvando o mapa em html
mapa.save('pontos_de_mineracao.html')
mapa
```

Out[20]: Make this Notebook Trusted to load map: File -> Trust Notebook



Quantidade pontos de mineração por Estados e Região

```
In [21]: df_mineracao_uf = cria_df_sql("SELECT m.* , uf.sigla_uf , uf.nm_regiao , uf.area_km2 FROM mineracao m , estados uf
df_mineracao_uf = converte_wkb(df_mineracao_uf , "geom")

C:\Users\Luis\AppData\Local\Temp\ipykernel_23064\368159002.py:19: UserWarning: pandas only supports SQLAlchemy
connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are n
ot tested. Please consider using SQLAlchemy.
df = pd.read_sql(sql_query, conn)
```

```
In [22]: df_mineracao_uf_agrupado = df_mineracao_uf.groupby('sigla_uf').size().reset_index(name='Pontos de Mineração').so
```

Out[22]:

	sigla_uf	Pontos de Mineração
8	MG	594
11	PA	540
20	RS	420
3	BA	337
7	GO	241
1	AM	182
23	SP	151
15	PR	136
17	RN	86
19	RR	79
10	MT	78
9	MS	42
0	AL	30
21	SC	29
6	ES	19
12	PB	17
22	SE	14
4	CE	12
16	RJ	11
5	DF	11
14	PI	10
24	TO	10
2	AP	8
13	PE	3
18	RO	1

Pode-se notar que MG tem a maior quantidade de pontos de mineração do País , visto que é de fato um grande "polo" de extração mineral no país , junto com o estado do PA

In [23]:

```
df_mineracao_regiao_agrupado = df_mineracao_uf.groupby('nm_regiao').size().reset_index(name='Pontos de Mineração')
df_mineracao_regiao_agrupado
```

Out[23]:

	nm_regiao	Pontos de Mineração
2	Norte	820
3	Sudeste\n	775
4	Sul\n	585
1	Nordeste\n	509
0	Centro-oeste\n	372

No entanto , quando se observa a visualização por região do País, nota-se que a região Norte do País é a que detém maior quantidade de pontos de mineração. E como já observado acima, muitos de seus estados detém grandes área de terras indígenas e APA's. Tal conflito ainda é fortemente presente na região, como mostrado no início da análise, com constantes brigas entre ambientalistas e empresas de mineração

Pontos de mineração por tipo de produto extraído:

In [24]:

```
# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                  zoom_start=4,
                  width=1280, height=720,
                  tiles = "cartodbpositron",
                  name = 'Positron'
                 )
#####
# Estilos no mapa de estados do Brasil
style_uf = {'fillColor': 'orange', #cor de preenchimento
            'color': 'orange', #cor da linha de contorno
            'weight': 2, #espessura da linha
            }
```

```

# plotando no mapa do brasil
folium.GeoJson(gjson_estados, name='Brasil' , style_function=lambda x:style_uf).add_to(mapa)
#####
# gerando as cores dos pontos no dataframe
valores_unicos = df_mineracao['tipoextmin'].unique()
cores = {}
for i in valores_unicos:
    cores[i] = random_hex_color()

# Adiciona os pontos ao mapa
# Iterar sobre as linhas do GeoDataFrame e adicionar marcadores coloridos ao mapa
for _, row in df_mineracao.iterrows():
    tipo = row['tipoextmin']
    cor = cores[tipo]
    folium.CircleMarker(
        location=[row['geom'].y, row['geom'].x],
        radius=5,
        color=cor,
        fill=True,
        fill_color=cor,
        popup=f'{row['nome']} - {row['tipoextmin']}'
    ).add_to(mapa)

codigo_html = ["""
    <div style="
        position: fixed;
        bottom: 50px; left: 50px; width: 200px; height: 120px;
        border:2px solid grey; font-size:14px;
        background-color: white;
    ">
        <span>
            &nbsp; Tipo de Extração Mineral:<br>
        </span>
    """
]
for i in cores.keys():
    str_html = f"""
        <span>
            &nbsp; {i} &nbsp; <i class="fa fa-circle fa-1x" style="color:{cores[i]}></i><br>
        </span>
    """
    codigo_html.append(str_html)
codigo_html.append("</div>")

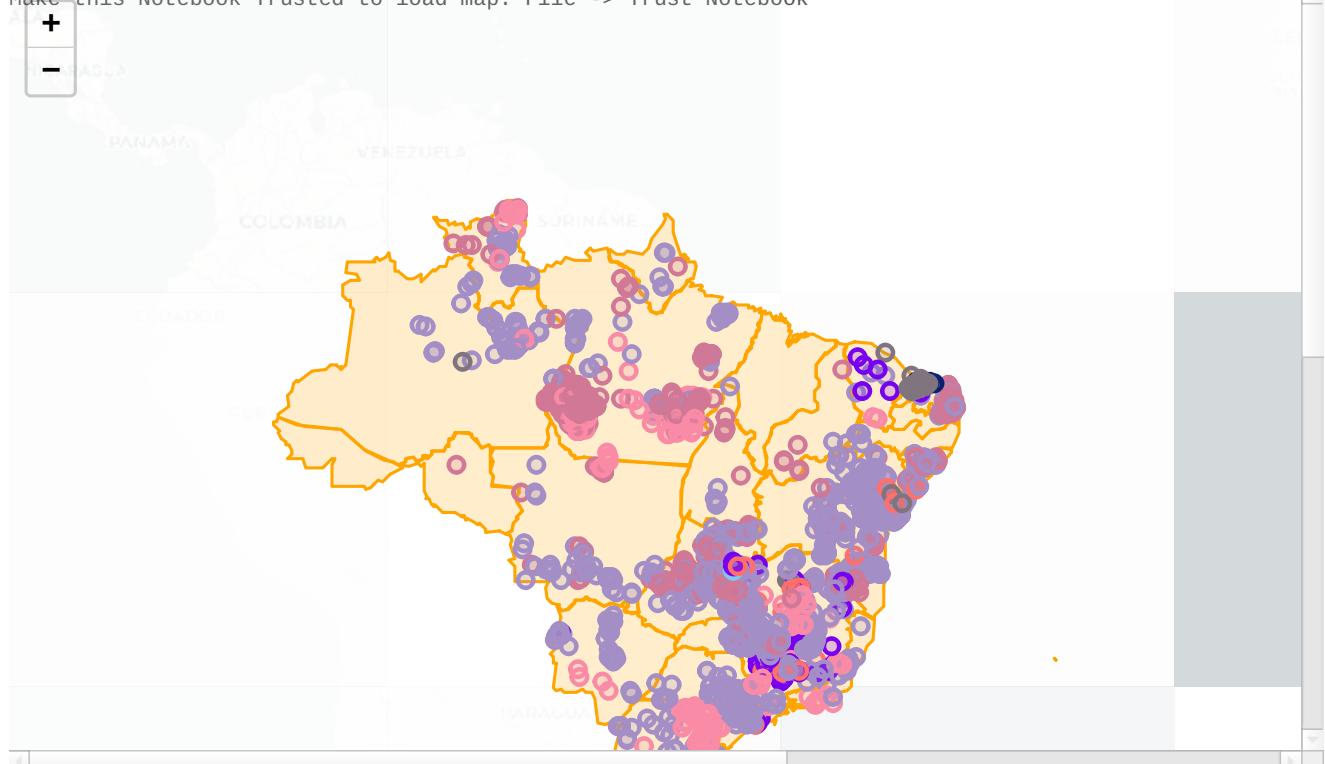
html_final = "\n".join(codigo_html)

FloatImage(html_final, bottom=100, left=50).add_to(mapa)

# Salvando o mapa em html
mapa.save('pontos_de_mineracao_por_tipo.html')
mapa

```

Out[24]: Make this Notebook Trusted to load map: File -> Trust Notebook



Com o mapa acima , podemos notar que o estado de MG tem uma matriz de mineração bem diversa, enquanto estados como o PA tem pravalecência de Garimpos , principalmente no Sul do estado. Na região Norte como um todo a forte presença de Minas de extração de Minérios.

Análise de Pontos de Mineração a pelo menos 25km de APA ou Terras indígenas:

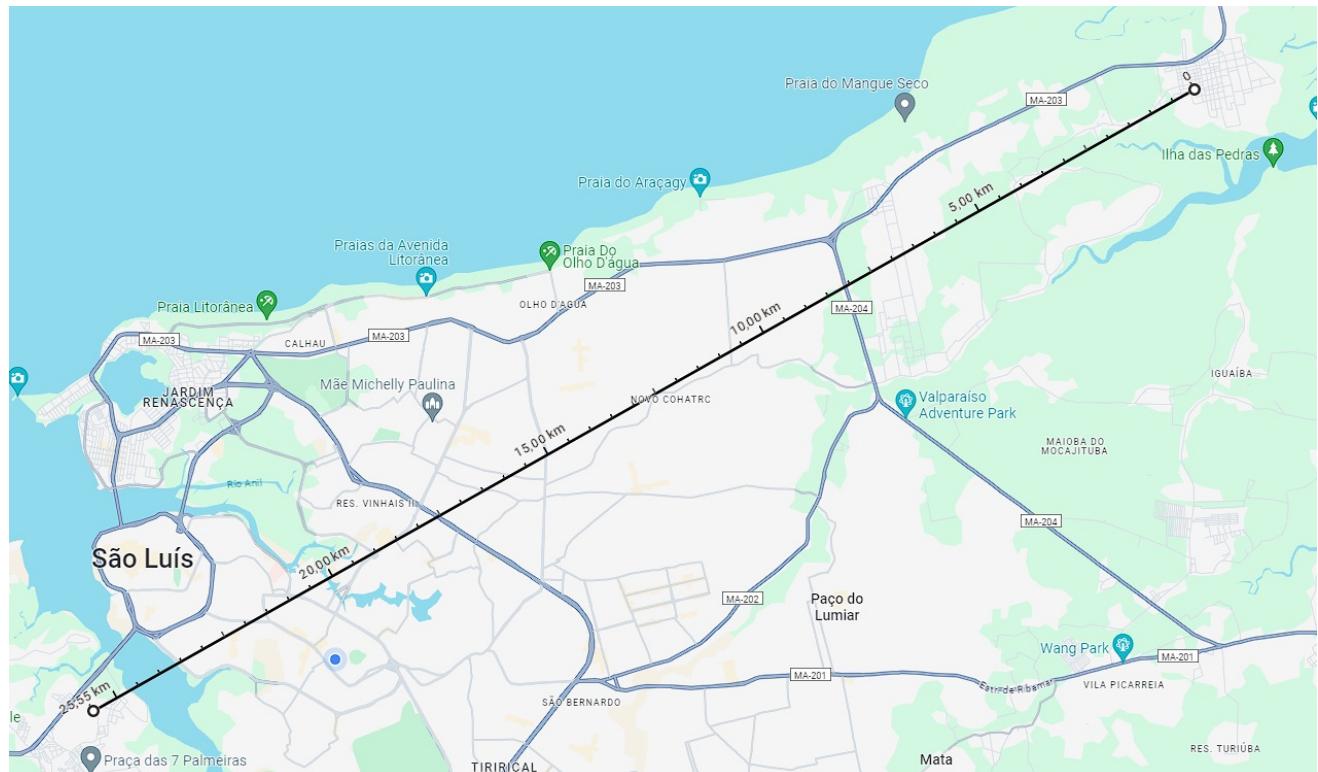
Anteriormente visualizamos os pontos que se localizam há pelo menos 25km de distância de uma APA ou Terra Indígena. Agora vamos analisar mais a fundo esses pontos de mineração

```
In [25]: qtde_mineracao_25km = df_mineracao_25km['id'].value_counts().sum()
qtde_mineracao_total = df_mineracao['id'].value_counts().sum()

proporcao = qtde_mineracao_25km/qtde_mineracao_total
print(f"Cerca de {proporcao * 100:.2f}% dos pontos de mineração conhecidos estão a 25 km ou menos de Áreas de Proteção Ambiental ou Terras Indígenas")
```

Cerca de 34.19% dos pontos de mineração conhecidos estão a 25 km ou menos de Áreas de Proteção Ambiental ou Terras Indígenas

Uma análise importante a ser feita é essa, podemos perceber que cerca de 35% de TODOS os pontos de mineração no Brasil estão a 25km ou menos. A nível de comparação , 25km é a distância aproximada entre o CCET na UFMA e o município da Raposa. Uma distância relativamente pequena quando vista de forma geral!



```
In [26]: # Exiba os estados no mapa
gjson_estados = df.__geo_interface__
gjson_apa = df_APA.__geo_interface__
gjson_terras_indigenas = df_terras_indigenas.__geo_interface__

# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                  zoom_start=4,
                  width=1280, height=720,
                  tiles = "cartodbpositron",
                  name = 'Positron'
                 )

#####
# Estilos no mapa de estados do Brasil
style_uf = {'fillColor': 'orange', #cor de preenchimento
            'color': 'orange', #cor da linha de contorno
            'weight': 2, #espessura da linha
           }

# plotando no mapa do brasil
folium.GeoJson(gjson_estados, name='Brasil' , style_function=lambda x:style_uf).add_to(mapa)

#####
```

```

# Adiciona os pontos de mineração há 25km ao mapa
for index, row in df_mineracao_25km.iterrows():
    folium.Circle(location=[row['geom'].y, row['geom'].x], radius=1000, color="#6666ff", fill=True).add_to(mapa)

#####
folium.LayerControl().add_to(mapa)
#####

# Adicionar uma legenda com rótulos personalizados
legend_html = """


<br>
    <span>&ampnbsp Legenda:<br>
    </span>
    <span>&ampnbsp Brasil &ampnbsp <i class="fa fa-circle fa-1x" style="color: orange"></i><br>
    </span>
    <span>&ampnbsp Terras Indígenas &ampnbsp <i class="fa fa-circle fa-1x" style="color: red"></i><br>
    </span>
    <span>&ampnbsp Áreas de Proteção Ambiental &ampnbsp <i class="fa fa-circle fa-1x" style="color: green"></i><br>
    </span>
    <span>&ampnbsp Pontos de Mineração há 25km de distância de uma APA ou Terra Indígena &ampnbsp <i class="fa fa-circle fa-1x" style="color: blue"></i><br>
    </span>

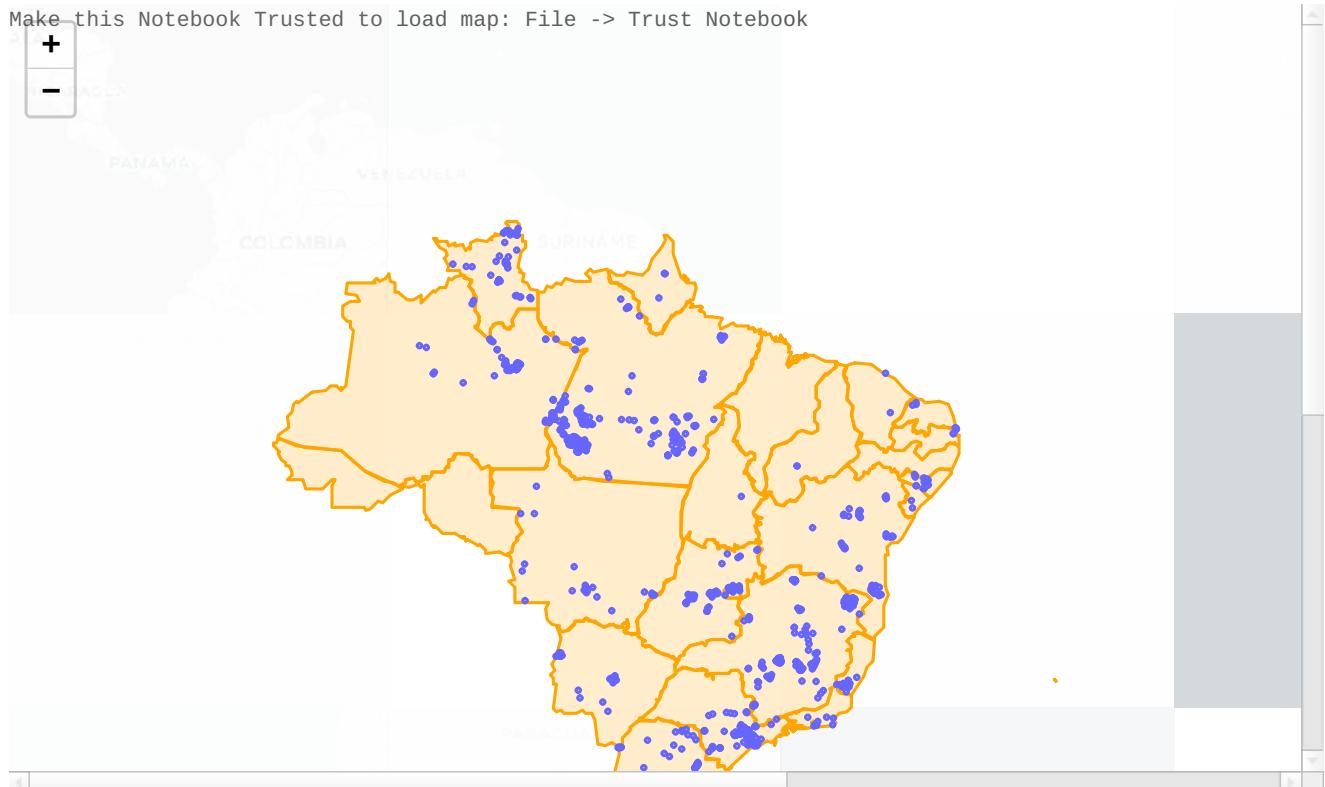

"""

FloatImage(legend_html, bottom=100, left=50).add_to(mapa)
#####

# Salvando o mapa em html
mapa.save('mapaPontosProximos_25km.html')
mapa

```

Out[26]: Make this Notebook Trusted to load map: File -> Trust Notebook



Análise de densidade de Pontos de Mineração e Área do estado

Outra análise interessante é visualizar a densidade de pontos de mineração por estados. Um

questionamento relevante é saber o quão presente são os pontos de minerações no estados:

```
In [27]: df_merged = pd.merge(df_estados, df_mineracao_uf_agrupado, on='sigla_uf', how='inner')
df_merged = df_merged.rename(columns={'Pontos de Mineração': 'qtdePontos'})
df_merged['1_ponto_mineracao_por_km2'] = round(df_merged['area_km2'] / df_merged['qtdePontos'], 2)
df_merged = df_merged.sort_values(by='1_ponto_mineracao_por_km2', ascending=True)
df_merged[['sigla_uf', '1_ponto_mineracao_por_km2']]
```

```
Out[27]:   sigla_uf  1_ponto_mineracao_por_km2
22      DF        523.71
6       RN        614.07
18      RS        670.73
9       AL        927.69
12      MG        987.40
21      GO       1411.80
16      PR       1465.43
10      SE       1567.01
15      SP       1643.84
11      BA       1675.85
1       PA       2307.17
13      ES       2424.97
24      RR       2830.94
17      SC       3301.06
7       PB       3321.60
14      RJ       3977.31
19      MS       8503.38
0       AM       8567.34
20      MT       11579.59
5       CE      12407.87
2       AP      17808.85
4       PI      25175.55
3       TO      27742.36
8       PE      32689.29
23      RO     237754.17
```

Na tabela acima temos um dado interessante , basicamente vemos a densidade de pontos de mineração por km². Portanto, podemos observar que o estado do DF que 1 ponto de mineração a cada 523,71 km², sendo esse o estado com mais pontos de mineração proporcionais a sua área (apesar de ser o menor estado da federação). Por outro lado , RO é o estado com maior valor, tendo 1 ponto de mineração a cada 237754,17 km². Portando tendo uma menor densidade

Visualizando o dado acima em um gráfico , teríamos o seguinte:

```
In [28]: df_merged['1_ponto_mineracao_por_km2_normalizado'] = min_max_normalization(df_merged['1_ponto_mineracao_por_km2'])
```

```
In [29]: # Exiba os estados no mapa
gjason_mineracao_intesidade = df_merged.__geo_interface__

# Inicializa o mapa:
loc_cords = [-13.813647, -32.839676]
mapa = folium.Map(location = loc_cords,
                  zoom_start=4,
                  width=1280, height=720,
                  tiles = "cartodbpositron",
                  name = 'Positron'
                 )

#####
# plotando no mapa do brasil
folium.GeoJson(gjason_mineracao_intesidade, name='Brasil', style_function=lambda x:intensidade_cor(x['properties']))

#####
# Adicione uma legenda com rótulos personalizados à sua variável legend_html
legend_html = """
<div style="
```

```

position: fixed;
bottom: 50px; left: 50px; width: 200px; height: 120px;
border:2px solid grey; font-size:14px;
background-color: white;
">


    Legenda:<br>


<br>

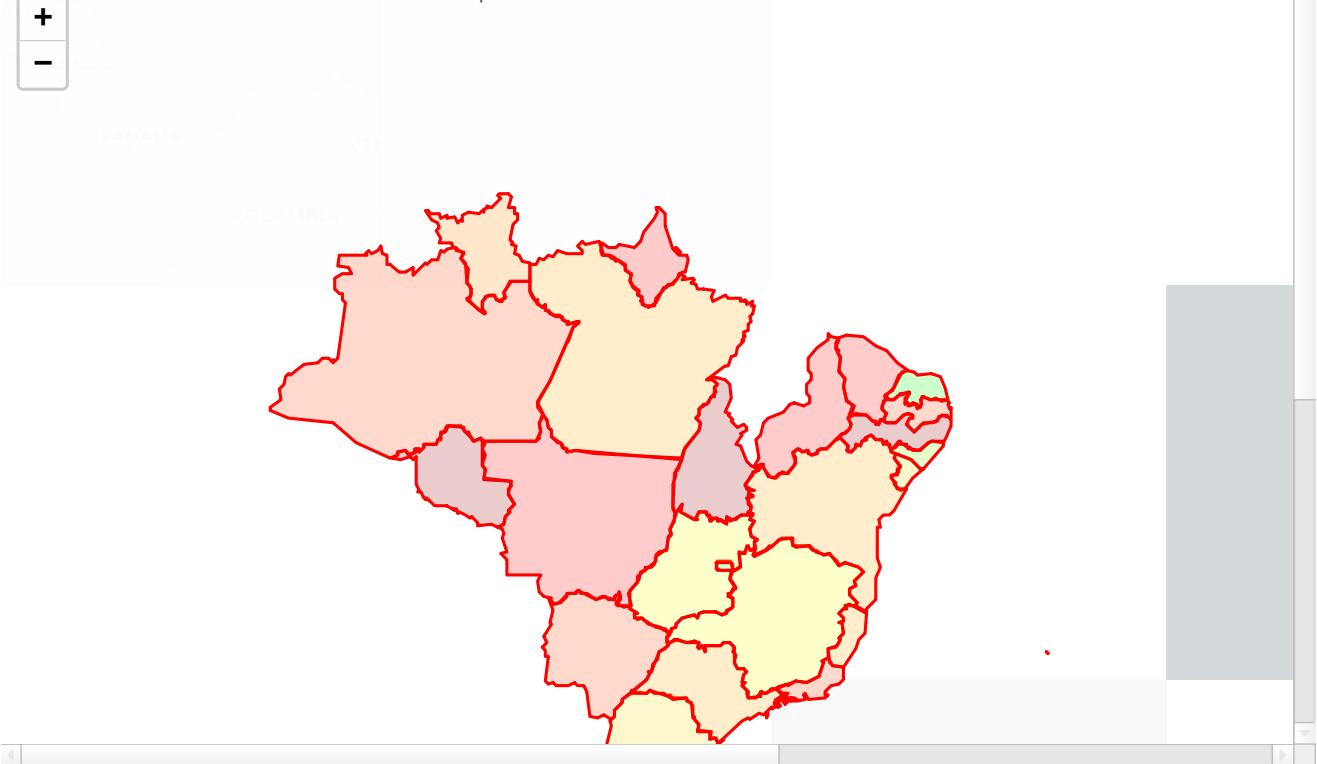

    Menor Valor     523.71<br>



    Maior Valor     237754.17<br>


```

Out[29]: Make this Notebook Trusted to load map: File -> Trust Notebook



No mapa acima podemos ver a distribuição da intensidade. Verde são os valores com densidade mais baixa, amarelo densidade média e vermelho densidades mais altas.

OBS: Nem o estado do Maranhão e nem do Acre tem pontos de mineração

Conclusões

Tendo em visto os problemas anteriormente citados, percebe-se que a questão de exploração mineral no Brasil (principalmente na região Norte do País) ainda é uma situação complexa, tendo em vista as contantes disputas nos territórios.

No que tange as perguntas do início do trabalho:

Como estão distribuídas as Terras Indígenas e APA's pelo território brasileiro?

- Percebe-se que as Terras Indígenas e APA's esão mais distribuidas nas Regiões Norte e Centro-Oeste do País

Como estão distribuídos os pontos de mineração pelo território brasileiro?

- Percebe-se que as áreas do Leste do Brasil (Nordeste , Sul e Sudeste) detém mais pontos de mineração , no entanto estados do Norte em certas regiões (Sul do Pará) detém uma alta concentração de pontos de mineração

Os pontos de mineração tem representado alguma "ameaça" aos citados acima?

- Representar uma ameaça é um conceito amplo , o fato de 34% dos pontos de mineração estarem próximos a Terras Indígenas e APA's é de fato algo preocupante , não só pela agravante da questão das terras , mas as consequências ambientais que estes podem acarretar para os mesmos. Outro agravante são ainda garimpos ilegais que constantemente estão em conflitos com ambientalistas e indígenas , principalmente na região Norte e Centro-Oeste