

**TECNOLOGIA EM SISTEMAS PARA INTERNET**

**Rosana da Silva Soares  
Kennedy Viana Aguiar  
Aurélio Vinícius França dos Santos**

**RELATÓRIO 3 DE PRÁTICA INTEGRADA  
DE  
CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL**

**Brasília - DF**

**21 de Março de 2021**

# Sumário

<b>1. Objetivos</b>	<b>3</b>
<b>2. Descrição do problema</b>	<b>4</b>
<b>3. Desenvolvimento</b>	<b>5</b>
3.1 Código implementado	5
<b>4. Considerações Finais</b>	<b>8</b>
<b>Referências</b>	<b>9</b>

# 1. Objetivos

Nesta etapa do projeto realizamos uma limpeza dos dados obtidos até este momento, podendo assim trabalhar apenas com dados necessários, excluindo informações irrelevantes como, tabelas e registros vazios. Além de organizar melhor esses dados, ordenando-os e adicionando variáveis para otimizar a estrutura.

## 2. Descrição do problema

Muitos dos dados coletados não serão utilizados, e também temos colunas e outras partes da tabela que não utilizaremos, o propósito dessa sprint é justamente organizar os dados e fazer uma limpa em tudo aquilo que não será utilizado e trazendo apenas o que for relevante.

## 3. Desenvolvimento

Para organizar as colunas, utilizamos algumas bibliotecas como: *datetime*, *pandasql* e *pandas*

### 3.1 Código implementado

#### Parte 1: Limpeza de dados

##### Importando as bibliotecas e módulos necessários:

```
#pip install zipcodes
#!pip install -U pandasql
from folium.plugins import HeatMap
import folium
import zipcodes
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from collections import Counter
import pandasql
from datetime import datetime
```

##### Lendo o OVNIS.csv, apagando algumas colunas, e criando variáveis de controle

```
ovnis_df = pd.read_csv('OVNIS.csv')
ovnis_df = ovnis_df.drop(columns=['ID', 'duracao', 'resumo',
'data_postagem'])
estados = pd.read_excel('states.xlsx')['Abbreviation'].tolist()
formatos_relevantes = []
qr = "SELECT formato, COUNT(*) AS views FROM ovnis_df GROUP BY formato"
```

## Verificando quais são os formatos com mais de 1000 ocorrências

```
r = pd.DataFrame(pandasql.sqldf(qr, locals()))
for i, row in r.iterrows():
    if row.views >= 1000:
        formatos_relevantes.append(row.formato)
```

## Percorrendo o dataframe e apagando os registros que:

- Não estão dentre os estados dos Estados Unidos;
- Possuem registros Unknown
- Não pertencem ao grupo de formatos com mais de 1000 ocorrências

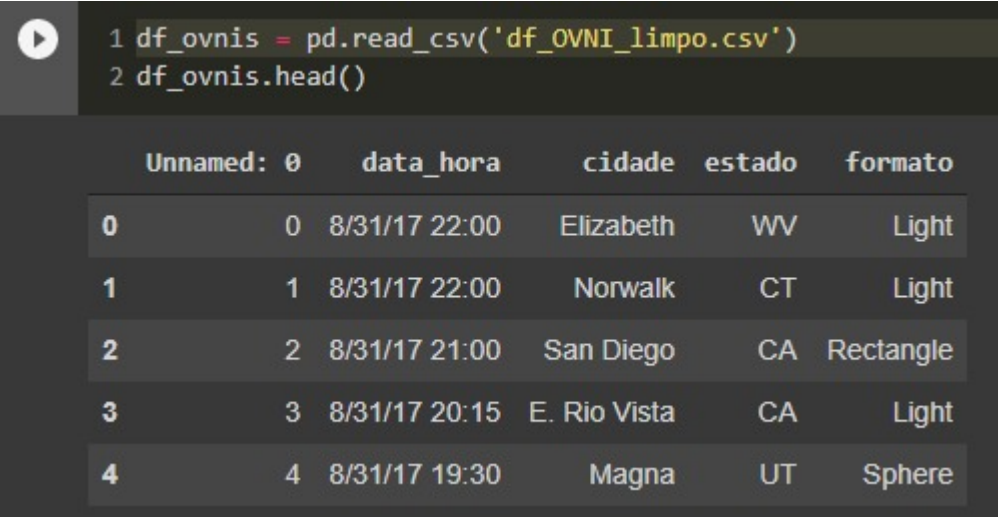
```
for i, row in ovnis_df.iterrows():
    if row.estado not in estados:
        ovnis_df.drop(index=i, inplace=True)
    elif row.estado == 'Unknown' or row.cidade == 'Unknown' or
row.formato == 'Unknown':
        ovnis_df.drop(index=i, inplace=True)
    elif row.formato not in formatos_relevantes:
        ovnis_df.drop(index=i, inplace=True)
```

## Percorrendo o dataframe e apagando linhas com registros nulos

```
for i, row in ovnis_df.isna().iterrows():
    if row.estado == True or row.cidade == True or row.formato == True:
        ovnis_df.drop(index=i, inplace=True)
```

## Ajustando e salvando o arquivo

```
ovnis_df = ovnis_df.reset_index()
ovnis_df = ovnis_df.drop(columns=['index'])
ovnis_df.to_csv('df_OVNI_limpo.csv')
```



```
1 df_ovnis = pd.read_csv('df_OVNI_limpo.csv')
2 df_ovnis.head()
```

	Unnamed: 0	data_hora	cidade	estado	formato
0	0	8/31/17 22:00	Elizabeth	WV	Light
1	1	8/31/17 22:00	Norwalk	CT	Light
2	2	8/31/17 21:00	San Diego	CA	Rectangle
3	3	8/31/17 20:15	E. Rio Vista	CA	Light
4	4	8/31/17 19:30	Magna	UT	Sphere

Dataframe após limpeza

## Parte 2: Acréscimo de variáveis

### Lendo o `df_OVNI_limpo.csv`, criando listas e variáveis de controle

```
df_limpo = pd.read_csv('df_OVNI_limpo.csv')
df_limpo = df_limpo.drop(columns=['Unnamed: 0'])
dias = []
meses = []
weekdays = []
horas = []
datas = []
DIAS = ['Segunda-feira', 'Terça-feira', 'Quarta-feira', 'Quinta-Feira',
'Sexta-feira', 'Sábado', 'Domingo']
```

### Percorrendo o dataframe e adicionando dados às listas criadas

```
for i, row in df_limpo.iterrows():
    try:
        horas.append(row.data_hora.split(' ')[1])
    except IndexError:
        horas.append(None)

    dt = datetime.strptime(row.data_hora.split(' ')[0], '%m/%d/%y')
    meses.append(dt.month)
    dias.append(dt.day)
    weekdays.append(DIAS[dt.weekday()])
    datas.append(row.data_hora.split(' ')[0])
```

### Atualizando o dataframe com novas colunas e salvando o arquivo

```
df_limpo = df_limpo.assign(data=datas, hora=horas, dia_semana=weekdays,
dia=dias, mes=meses)
df_limpo.drop(columns=['data_hora'], inplace=True)
df_limpo.to_csv('df_OVNI_preparado.csv')
```

<pre> 1 df_ovnis = pd.read_csv('df_OVNI_preparado.csv') 2 df_ovnis.head() </pre>										
	Unnamed: 0	cidade	estado	formato	data	hora	dia_semana	dia	mes	ano
0	0	Elizabeth	WV	Light	8/31/17	22:00	Quinta-Feira	31	8	2017
1	1	Norwalk	CT	Light	8/31/17	22:00	Quinta-Feira	31	8	2017
2	2	San Diego	CA	Rectangle	8/31/17	21:00	Quinta-Feira	31	8	2017
3	3	E. Rio Vista	CA	Light	8/31/17	20:15	Quinta-Feira	31	8	2017
4	4	Magna	UT	Sphere	8/31/17	19:30	Quinta-Feira	31	8	2017

Dataframe após o acréscimo de variáveis

## 4. Considerações Finais

Diante do que foi realizado nesta etapa, as tentativas de encurtamento e otimização do código foram os obstáculos encontrados. Como forma de aperfeiçoamento, é favorável modificar a lógica do script para otimizar o desempenho, e tornar o código mais compreensível e organizado.



# Referências

**The National UFO Reporting Center.** Nuforc, 2021. Disponível em: <<http://www.nuforc.org/>>. Acesso em: 20 de Março. 2021.

**Python 3.9.2 documentation.** Python, 2021. Disponível em: <<https://docs.python.org/pt-br/3/library/datetime.html>>. Acesso em: 20 de Março. 2021.

**Python package index.** Python, 2021. Disponível em: <<https://pypi.org/project/pandasql/>>. Acesso em: 20 de Março. 2021.