



INSTITUTO FEDERAL DE BRASÍLIA

CAMPUS BRASÍLIA

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

Gabriel Santos

Marcos Vasconcellos

Pedro L. B. G. de Araujo

**RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E INTELIGÊNCIA
ARTIFICIAL: TAREFA 5.5 – EXPLORAÇÃO COM GRÁFICOS E MAPAS**

Brasília

2020

SUMÁRIO

1. Objetivos	3
2. Descrição do problema	4
3. Desenvolvimento	5
3.1 Código implementado	5
4. Considerações Finais	14
5. Referências	15

1. Objetivos

Partindo pelo ponto da grande importância dos campos da Inteligência Artificial e da Ciência de Dados na área da Tecnologia da Informação, possuir no mínimo um breve conhecimento e contato com tais tecnologias, mostra-se imprescindível para um verdadeiro profissional de TI.

Sobre o campo da Inteligência Artificial, de acordo com Stuart Russell; Peter Norvig (2013), a IA define-se por um comportamento que se relaciona a processos de pensamento e raciocínio. E para seu estudo e trabalho, devemos considerar quatro definições principais sobre a Inteligência Artificial, sendo estas: pensar como um humano; agir como humano; pensar racionalmente, e agir racionalmente.

Passando para a área da Ciência de Dados, Joel Grus, (2016), diz que a quantidade de dados espalhadas por todos os lados em nosso cotidiano, é extratossférica. E um cientista de dados é alguém que trabalha com dados dos mais diversos tipos e tamanhos, geralmente desorganizados, extraindo-os com o objetivo de transformá-los em conhecimento.

Apresentada uma abordagem geral sobre conceitos importantes relativos à ambas às áreas temas deste projeto, o objetivo deste documento é introduzir gradativamente o leitor, ao desafio a nós proposto pelo Instituto Federal de Brasília. Tal desafio, consiste na aplicação de diversas tarefas menores, gradativamente aplicadas ao longo do projeto, visando um objetivo final. Para alcançarmos este objetivo, utilizaremos conhecimentos e ferramentas ligados à IA e Ciência de Dados, e metodologias ágéis como o Scrum.

2. Descrição do Problema

Para a terceira etapa do projeto, após realização da coleta de dados relacionados à OVNIS e a filtragem destes dados para que se apliquem apenas à estados dos Estados Unidos, o próximo passo será exibir os dados em forma de gráfico. Além disso, enriquecer o resultado destas informações é importante. Para tanto, transformaremos os dados coletados, limpadados e filtrados, em mapas.

Por fim, o resultado da aplicação de todos estes procedimentos, deverá resultar em um arquivo .CSV.

3. Desenvolvimento

A nossa atuação na resolução do problema proposto, inicia-se com a utilização da linguagem Python e algumas de suas bibliotecas. Segundo OLIVEIRA, Marcos (2019) “As bibliotecas e pacotes Python são um conjunto de módulos e funções úteis que minimizam o uso de código em nossa vida cotidiana. Essas bibliotecas e pacotes destinam-se a uma variedade de soluções modernas.”.

Algumas bibliotecas específicas foram utilizadas como base para a construção e desenvolvimento do projeto. Estas, respectivamente denominam-se e possuem como objetivo:

- **Numpy**: É o mais popular pacote de processamento de Arrays do Python. Não apenas oferece arrays e matrizes, também as gerencia.
- **Requests**: Uma grande biblioteca HTTP que funciona sob a licença Apache 2.0. Seu objetivo é tornar as solicitações HTTP mais responsáveis e fáceis de usar.
- **BeautifulSoup**: Consiste em uma biblioteca para extração e análise de documentos HTML e XML.
- **Pandas**: Uma das diversas bibliotecas Python voltadas para a área da Ciência de Dados. É um pacote de software Python voltado para a manipulação de estruturas de dados de forma intuitiva.
- **Matplotlib**: Biblioteca que utiliza Python Script para criar gráficos e plotagens bidimensionais, permitindo a criação de múltiplos eixos simultâneos.
- **pprint (Pretty Printer)**: É uma biblioteca que apresenta um módulo para exibição de estruturas em várias formas diferentes.
- **Zipcodes**: Conforme o próprio nome já diz, é uma biblioteca Python utilizada para o trabalho com códigos postais. Essa biblioteca nos permite filtrar e exibir dados relacionados a códigos postais. Todavia, é importante ressaltar que só possui informações de códigos postais relacionados aos Estados Unidos.
- **Folium**: Biblioteca que utiliza uma biblioteca JavaScript para juntas, permitirem a manipulação e plotagem de informações em mapas através do Python.

3.1 Código implementado

Abaixo, serão apresentados as estruturas de códigos com a utilização das bibliotecas citadas durante o desenvolvimento e plotagem dos mapas de dados.

A Figura 1, exibe à seguir a importação de todas as bibliotecas utilizadas até esta etapa do projeto.

```
1 import pandas as pd
2 import numpy as np
3 import plotly.graph_objects as go
4 import matplotlib.pyplot as plt
```

Figura 1 - Importação das Bibliotecas Python Utilizadas

A imagem a seguir, conta com o trecho do código responsável por carregar o arquivo CSV com o resultado da extração e organização dos dados de OVNIS, filtrando-os para estados dos EUA. Além disso, também é exibido um resumo do resultado gerado até o momento.

```
1 OVNIS_EUA = pd.read_csv('../5.4-Exploracao-dos-dados-com-SQL/OVNIS_EUA.csv')
2 OVNIS_EUA.head()
```

	date_time	city	state	shape	duration	summary	posted
0	9/22/97 20:00	Solomons Island	MD	Disk	10 minutes	Close up at twilight, Stationary UFO.	8/5/09
1	9/19/97	Garden Grove	CA	Rectangle	4 mins.	Around 6:30 PM I was walking through a Vons Pa...	12/1/19
2	9/18/97 20:15	Panama City	FL	Unknown	30 seconds	Looked like stars in the sky so far up/moveing...	3/13/12
3	9/15/97 00:00	Houston	TX	Disk	5 minutes	Beautiful silver-colored flying saucer about t...	7/19/10
4	9/15/97 20:00	Santa Fe	NM	Light	2-3 minutes	Saw white dot of light moving in zig-zag motio...	11/9/17

Figura 2 – Importando o CSV de Relatos de OVNIS nos Estados Unidos

A título de curiosidade, para que possamos tirar mais algumas informações importantes dessa análise de dados, decidimos encontrar os quatro estados dos Estados Unidos com a maior frequência de relatos de OVNIS. Para isso, as imagens abaixo apresentam o código que nos ajuda a realizar essa filtragem, encontrando e exibindo o resultado. Como vemos na Figura 3, nos é afirmado que os quatro estados com o maior número de ocorrências, são: CA (Califórnia); FL (Flórida); WA (Washington), e TX (Texas).

```
1 # Quantidade de relatos por estado em ordem decrescente
2 qtd_relatos_estados = OVNIS_EUA.state.value_counts().to_frame().reset_index()
3 qtd_relatos_estados.columns = ['estado', 'relatos']
4
5 # 4(quatro) estados com a maior frequência de relatos
6 GBA_ESTADOS = qtd_relatos_estados.head(4)['estado'].values
7 GBA_ESTADOS
```

```
array(['CA', 'FL', 'WA', 'TX'], dtype=object)
```

Figura 3 – Relatos Por Estado e os Quatro Estados com Maior Frequência de Relatos

Logo após, decidimos encontrar também quais são os formatos mais populares para os OVNIS destes relatos nos EUA. A Figura 4 exibe o código e o resultado, nos mostrando que os formatos “*Light*” e “*Circle*”, são os dois mais populares.

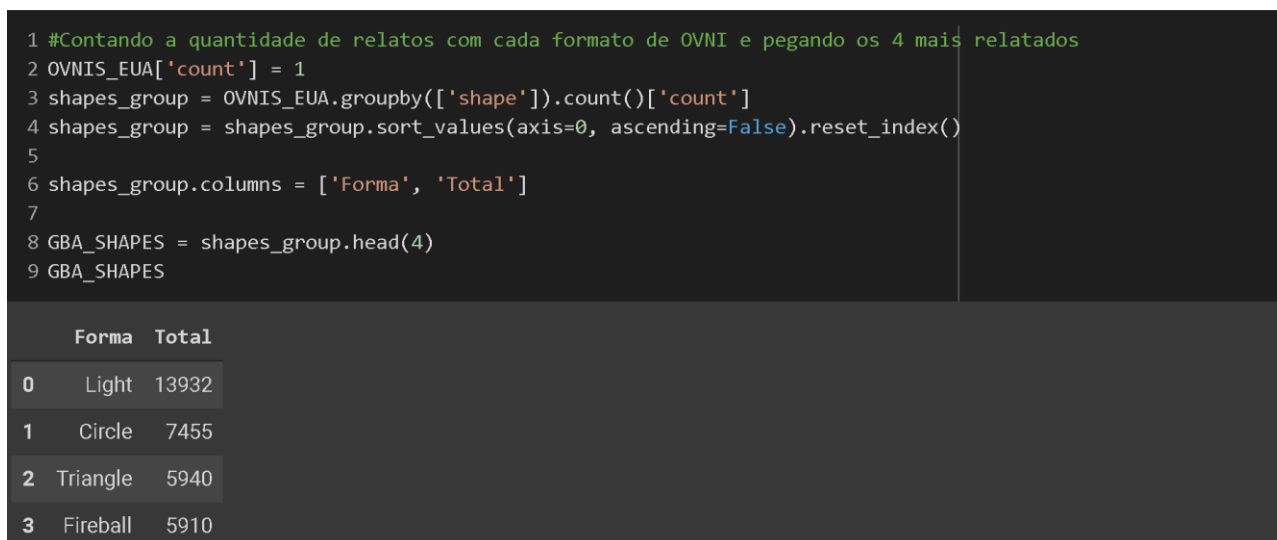


Figura 4 – Formatos de OVNIS Mais Populares de Acordo com os Relatos

Descobertas essas informações através da análise dos dados, agora faremos a criação de gráficos para que possamos exibir tais informações de uma maneira mais prática, e de fácil compreensão. Para tanto, primeiramente é necessário criar um dataframe com os OVNIS de formatos mais populares. Logo após, precisamos criar uma lista para cada formato, para que então estejamos aptos a exibir essas informações através de dois tipos de gráficos: gráfico de grupos de barras, e gráfico em barras empilhadas. A imagem abaixo exhibe visualmente o processo citado.

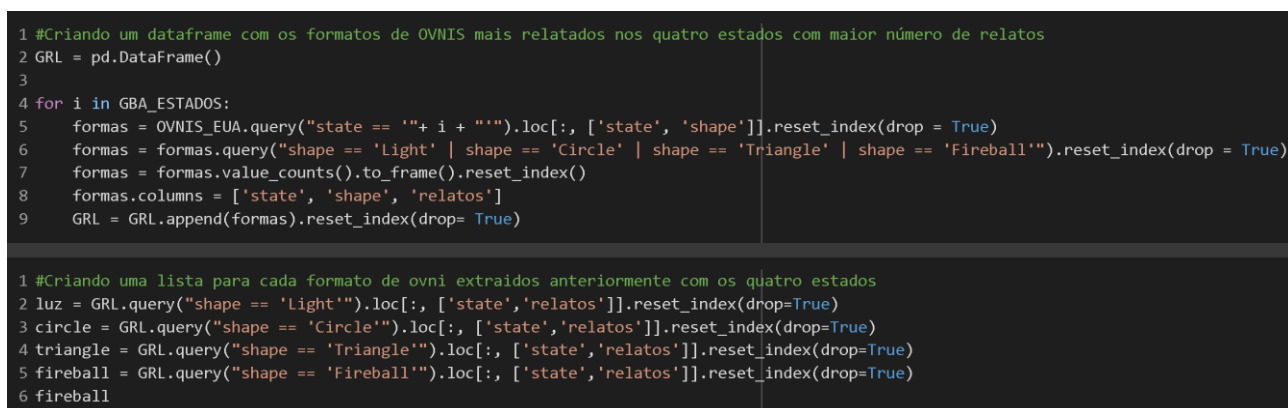


Figura 5 – Criação do Dataframe e da Lista Para os Formatos

Logo após, temos respectivamente nas figuras Figura 6 e Figura 7, o código utilizado para a criação do gráfico de barras agrupadas, e o resultado final do gráfico após sua criação.

```

2 grupos = [1,3,5,7]
3 plt.rcParams['figure.figsize'] = (11,7)
4 fig, ax = plt.subplots()
5 bar_larg = 0.4
6
7 for i in grupos:
8     p1 = ax.bar(i, luz['relatos'], bar_larg, color='blue')
9     p2 = ax.bar(i + bar_larg, circle['relatos'], bar_larg, color='orange')
10    p3 = ax.bar(i + bar_larg*2, triangle['relatos'], bar_larg, color='green')
11    p4 = ax.bar(i + bar_larg*3, fireball['relatos'], bar_larg, color='red', label='Fireball')
12
13 plt.xlabel('State')
14 plt.ylabel('Views')
15 ax.set_title('Group of bars')
16 ax.set_xticks([r + bar_larg for r in grupos])
17 ax.set_xticklabels(('CA', 'FL', 'WA', 'TX'))
18 ax.legend((p1[0], p2[0], p3[0], p4[0]), ('Light', 'Circle', 'Triangle', 'Fireball'))
19 plt.show()

```

Figura 6 – Código Relacionado a Criação do Mapa em Barras Agrupadas

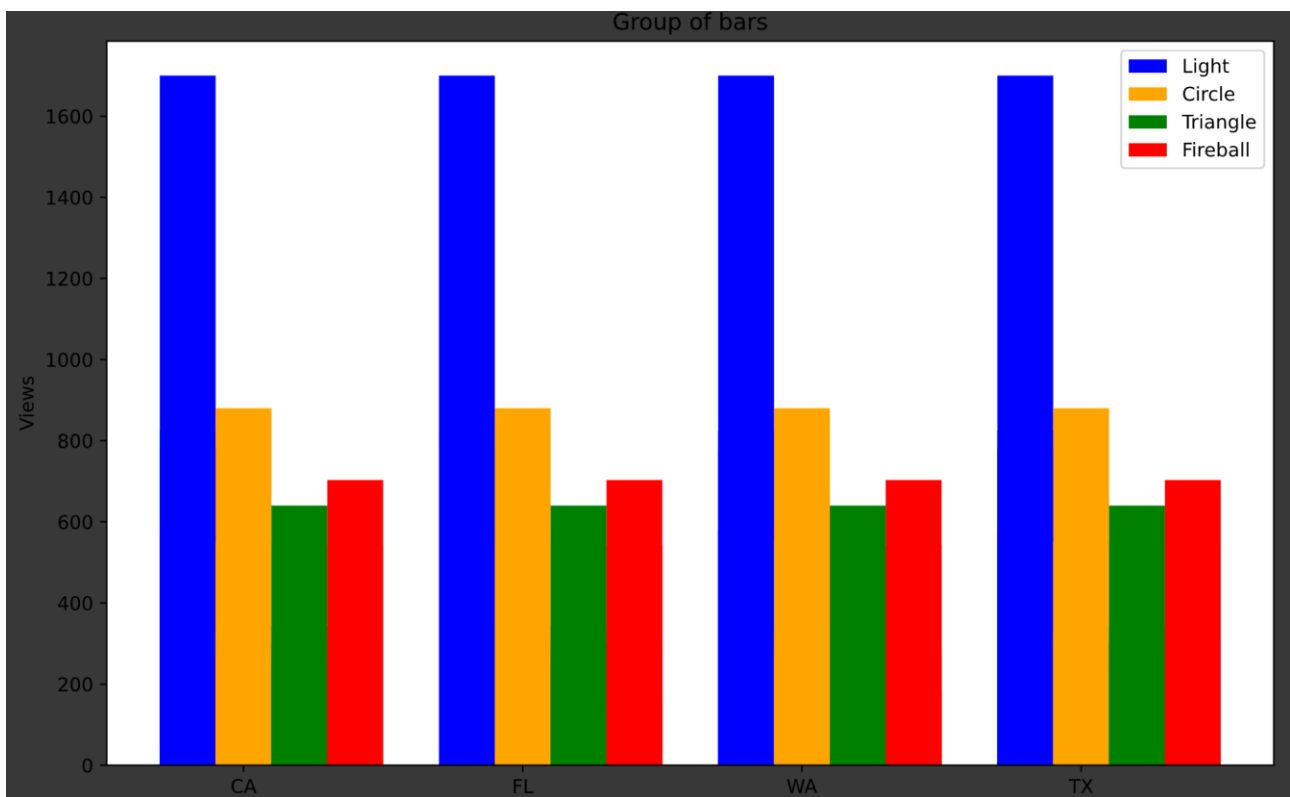


Figura 7 – Gráfico de Barras Agrupadas Relativo aos Formatos de OVNIS

Uma vez criado e exibido o gráfico de barras agrupadas, faremos o mesmo processo com relação ao gráfico de barras empilhadas. A Figura 8 exibe o código implementado para a criação do mapa, enquanto a Figura 9 exibe o resultado final do mesmo.


```

2 fig, ax = plt.subplots()
3 grupo2 = 4
4 bar_larg = 0.4
5
6 index = np.arange(grupo2)
7
8 p1 = ax.bar(index, luz['relatos'], bar_larg, color='blue')
9 p2 = ax.bar(index, circle['relatos'], bar_larg, color='orange', bottom=luz['relatos'])
10 p3 = ax.bar(index, triangle['relatos'], bar_larg, color='green', bottom=luz['relatos']+circle['relatos'])
11 p4 = ax.bar(index, fireball['relatos'], bar_larg, color='red', label='Fireball', bottom=luz['relatos']+circle['relatos']+triangle['relatos'])
12
13 plt.xlabel('State')
14 plt.ylabel('Views')
15 ax.set_xticks(index)
16 ax.set_xticklabels(('CA', 'FL', 'WA', 'TX'))
17 ax.legend((p1[0], p2[0], p3[0], p4[0]), ('Light', 'Circle', 'Triangle', 'Fireball'))
18 plt.show()

```

Figura 8 – Código Relacionado a Criação do Mapa de Barras Empilhadas

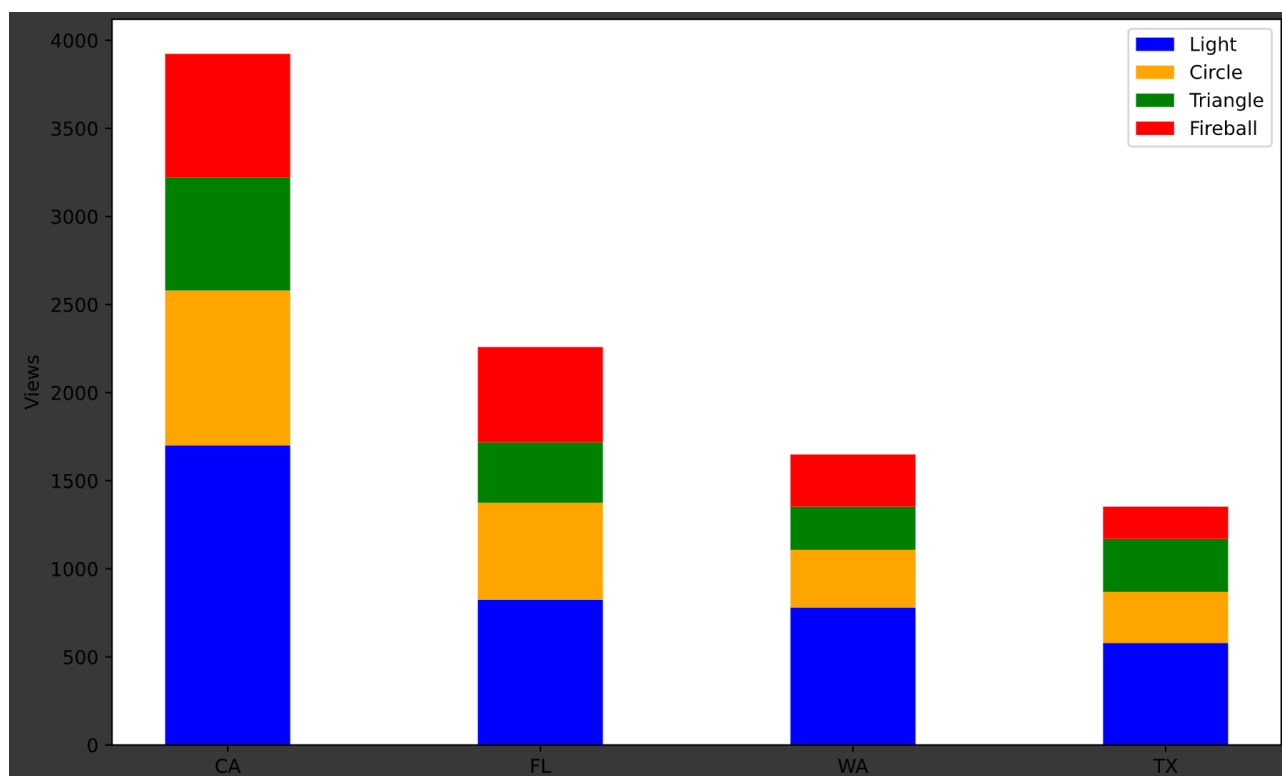


Figura 9 – Gráfico de Barras Empilhadas Relativo aos Formatos de OVNIS

Realizada a criação dos gráficos, nosso próximo desafio foi criar os mapas para exibirmos informações semelhantes, de uma forma diferente e mais representativa. Isto, pois, estaremos utilizando informações como latitude e longitude dos registros, para que possamos plotar os locais exatos dos relatos nos mapas.

Iniciando esse processo, para atingirmos o nosso objetivo, precisaremos passar por mais algumas etapas de desenvolvimento. Conforme exibido abaixo, primeiramente faremos a importação das bibliotecas pprint e zipcodes. Após a importação das bibliotecas, filtraremos os estados e cidades dos relatos para colocá-los em uma lista, para então criarmos um dataframe dessa lista.

```

1 #Bibliotecas para pegar latitude e longitude
2 from pprint import pprint
3 import zipcodes

1 #Pegando as cidades e estados dos relatos
2 cidades_eua = OVNIS_EUA.loc[:, ['city', 'state']]
3 cidades = cidades_eua.values

1 #Lista com todos os dados em relação a localidade de cada cidade
2 registros = []
3 for city in cidades:
4     registros.append(zipcodes.filter_by(city=city[0], state=city[1]))

1 #Criando um Dataframe com todos os dados da lista registros
2 df = pd.DataFrame()
3 for citys in registros:
4     for r in citys:
5         df = df.append(r, ignore_index=True)

```

Figura 10 – Importação das Bibliotecas e Criação do Dataframe com a Lista de Estados e Cidades

Logo após, conforme exibido na imagem abaixo, criaremos um dataframe apenas com as informações relacionadas aos estados, cidades, latitudes e longitudes, para que possamos trabalhar com esses dados na criação dos mapas.

```

2 df_latitude_long = df.loc[:, ['state', 'city', 'lat', 'long']]
3 df_latitude_long.head(30)

```

Figura 11 – Criação do Dataframe com Estado, Cidade, Latitude e Longitude

Na próxima etapa, exibido na Figura 11, faremos a exportação das informações obtidas até o momento para um arquivo CSV chamado “LatiLongTheCitys”. Depois disso, utilizando esse arquivo, criaremos uma lista contendo as informações de latitude e longitude de cada um desses relatos.

```

1 #Salvando esses dados em um csv
2 df_latitude_long.to_csv('LatiLongTheCitys.csv')

1 #Pegando os dados do csv e obtendo uma lista com a latitude e longitude de cada relato no EUA
2 df_latitude_long = pd.read_csv('LatiLongTheCitys.csv')
3 coordenadas = df_latitude_long.loc[:, ['lat', 'long']].values
4 coordenadas

```

Figura 12 – Exportação para o CSV e Criação da Lista com Latitude e Longitude dos Relatos

Finalmente iniciando a criação dos mapas, na imagem à seguir temos a importação da biblioteca Folium, responsável por nos permitir realizar a criação e plotagem dos mapas, através de seus *plugins*.

```

1 #Importando bibliotecas do folium para gerar os mapas
2 import folium
3 from folium import plugins
4 from folium.plugins import HeatMap
5
6 #Adicionando informações para gerar o mapa do EUA
7 eua = folium.Map(
8     location=[37.0902, -95.7129], # Coordenadas retiradas do Google Maps
9     zoom_start=4
10 )

```

Figura 13 – Importação do *Folium* e Utilização das Informações para Geração do Mapa dos EUA

Respectivamente, nas figuras a seguir, temos o código solicitando a plotagem do mapa utilizando as coordenadas informadas relacionadas aos Estados Unidos da América, e a plotagem do resultado final do mapa.

```

1 #Gerando o mapa do EUA com as coordenadas de cada relato
2 eua.add_child(plugins.HeatMap(coordenadas))
3 eua

```

Figura 14 - Solicitação do Mapa dos EUA

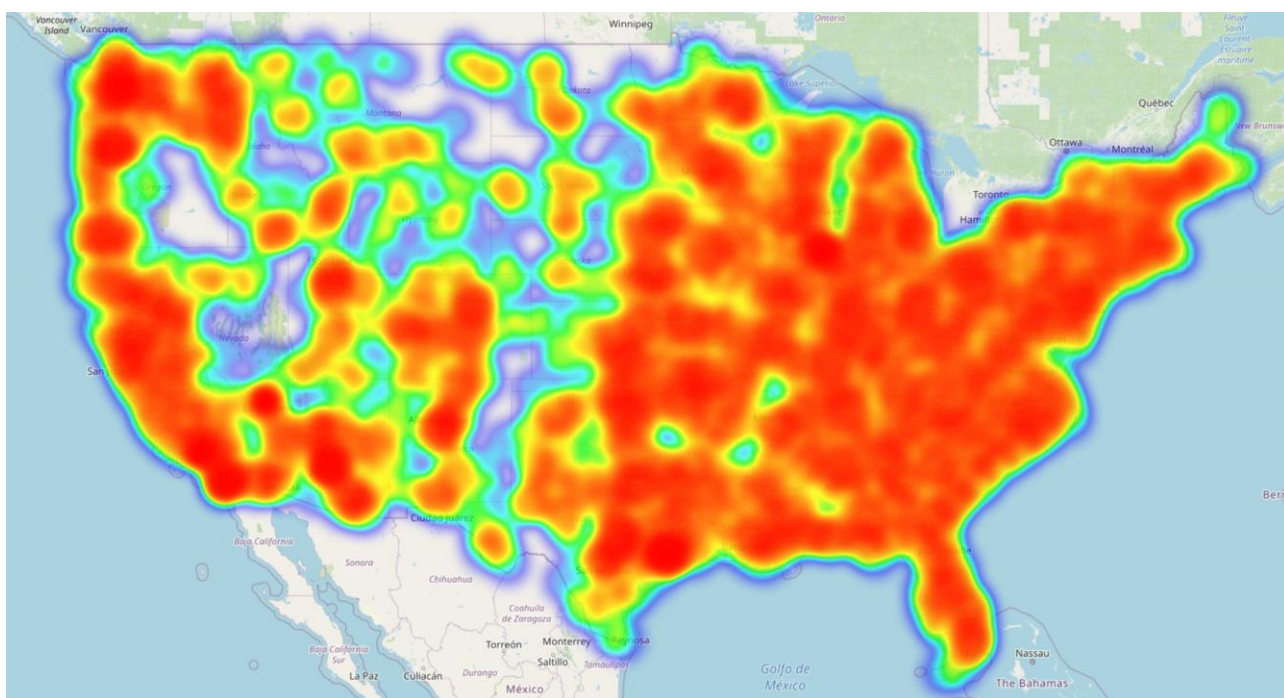


Figura 15 – Mapa de Exibição dos Relatos de OVNI nos EUA

Feita a criação e exibição do mapa de relatos de OVNI nos Estados Unidos, decidimos filtrar os resultados apresentados ainda mais. Agora, faremos a criação e plotagem dos mesmos relatos, porém com foco no estado da Califórnia, o terceiro maior estado desse país.

De início, filtraremos as informações geográficas dos relatos, ou seja, latitude e longitude, ao estado da Califórnia.

```

1 #Pegando as coordenadas de todos os relatos que ocorreram na Califórnia
2 coordenadas_CA = df_latitude_long.query("state == 'CA'").loc[:, ['lat', 'long']].values
3 coordenadas_CA

```

Figura 16 – Coordenadas da Califórnia

Em seguida, passamos as informações de latitude e longitude do estado da Califórnia ao *Folium* para que possamos realizar a criação do mapa.

```

1 #Colocando a latitude e longitude da california para a criação do mapa
2 ca = folium.Map(
3     location=[35.1258000, -117.9859000], # Coordenadas retiradas do Google Maps
4     zoom_start=6
5 )

```

Figura 17 – Passando as Informações de Latitude e Longitude ao Folium

Por fim, passadas as informações necessárias à biblioteca, solicitamos a plotagem do mapa criado e fazemos a sua exibição na Figura 19.

```

1 #Gerando o mapa da califórnia com as coordenadas de todos os relatos que aconteceram naquele estado
2 ca.add_child(plugins.HeatMap(coordenadas_CA))
3 ca

```

Figura 18 – Solicitação da Plotagem do Mapa de Relatos na Califórnia

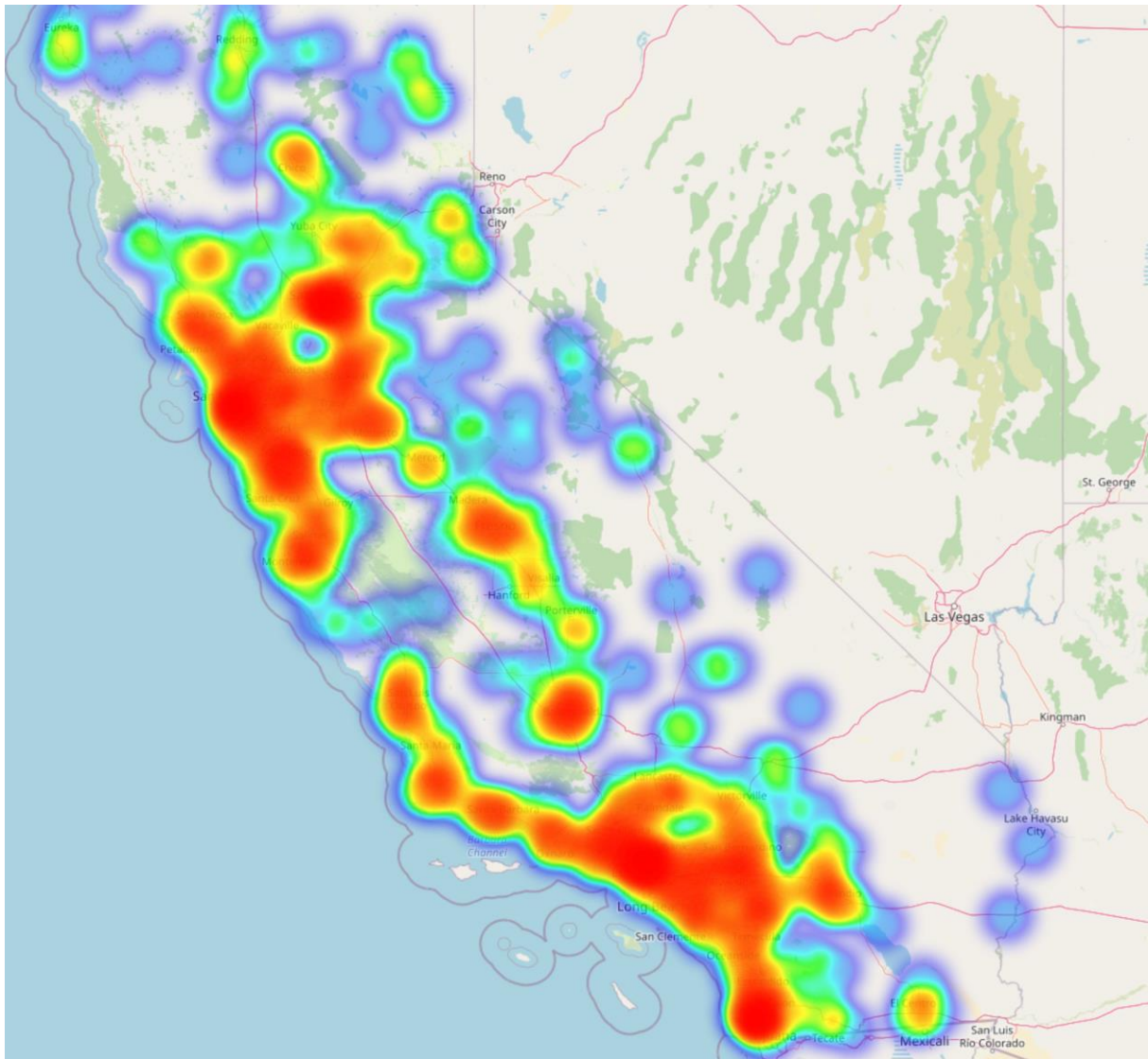


Figura 19 – Mapas de Relatos de OVNIS no Estado da Califórnia

4. Considerações Finais

Com a conclusão de todas as etapas de trabalho, o resultado obtido nos gerou um arquivo do tipo CSV, apresentando dois gráficos: um de barras agrupadas, e um de barras empilhadas. Além disso, realizamos a criação de dois mapas, para que pudéssemos facilitar ainda mais a visualização e análise desses dados.

Para tanto, o uso da linguagem Python e suas bibliotecas foi essencial. Através da integração de ambos, foi possível gerar a tabela e organizar as informações coletadas com o uso de colunas, separando as diferentes informações por data e hora, tipo, descrição do ocorrido, entre outros. Logo após, com a geração dos gráficos e mapas, temos ainda mais precisão e facilidade para analisar as informações, e de uma maneira diferente.

Com isso, é facilmente notada a importância do campo da Ciência de Dados, que cumpre exatamente com o que promete: coletar dados espalhados em grandes quantidades, e gerar informações úteis com estes dados. E isso, em uma visão generalizada, consegue mudar a forma com que o mundo e suas informações são vistas. Trazendo uma maior facilidade na visualização e entendimento dos mais diversos dados, assim como um maior valor para os mesmos.

Projetos completos e de real valor para o mercado como este que se inicia (que por enquanto se conclui), tendem apenas à agregar positivamente no conhecimento e experiência prática dos estudantes dos cursos de Tecnologia da Informação do Instituto Federal de Brasília. Tais conhecimentos práticos, servem tanto para dar uma base de como um profissional atua no mercado, com os mais diversos tipos de projetos e demandas, assim como para prepará-los para o tal mercado, quiçá para a montagem de um portfólio.

5. Referências

RUSSELL Stuart; NORVIG Peter, **Inteligência Artificial**, 3ª edição. Rio de Janeiro: Elsevier, 2013.

GRUS, Joel. *Data Science do Zero: Primeiras Regras com Python*. Rio de Janeiro: Altabooks, 2016.

OLIVEIRA, Marcos. TERMINAL ROOT. **As 30 Melhores bibliotecas e pacotes Python para Iniciantes**. 2019. Disponível em: <https://terminalroot.com.br/2019/12/as-30-melhores-bibliotecas-e-pacotes-python-para-iniciantes.html>. Acesso em: 18 de Setembro de 2020.