
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 5.10 – DADOS NO MONGODB

Brasília

2020

# SUMÁRIO

## 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 extratosférica. 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 agéis como o Scrum.

## 2. Descrição do Problema

Nas etapas anteriores do projeto, realizamos a coleta, exploração e preparação dos dados. Nesta etapa, o objetivo é fazer o armazenamento de toda a informação e trabalho reunido em um banco de dados.

Visto que os dados já foram preparados e estão prontos para serem analisados, a utilização de um banco de dados nos permitirá tornar muito mais prático o processo de acesso e utilização dos dados. Além disso, nos proporciona uma maior organização e segurança. Sendo assim, o banco selecionado para nos auxiliar nesta etapa, será o MongoDB.

## 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 minizam 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 funcionca sob a licença Apache 2.0. Seu objetivo é tornar as solicitações HTTP mais responsíveis e fáceis de usar.
- **BeatifulSoup**: 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 bibliotecas 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.
- **PyMongo**: É um biblioteca Python que nos permite fazer a utilização do banco de dados MongoDB, junto ao desenvolvimento Python.
- **Urllib**: Um pacote Python com a coleção de módulos internos para a utilização do Python com urls.

### 3.1 Código implementado

Abaixo, serão apresentados as estruturas de códigos com a utilização de uma, ou algumas das bibliotecas acima citadas, que nos auxiliaram no desenvolvimento desta etapa.

A Figura 1, exibe à seguir a importação das bibliotecas nesta esta etapa do projeto.

```
import pymongo
from pymongo import MongoClient
import csv
import urllib
from bson.objectid import ObjectId
```

Figura 1 - Importação das Bibliotecas Python Utilizadas

Uma vez importadas as bibliotecas que nos permitirão vincular as informações que temos em um banco de dados, para de fato iniciarmos esse vínculo, primeiro precisaremos criar uma conexão com o banco.

Abaixo, na Figura 2 encontramos o código solicitando à biblioteca PyMongo que faça a conexão com o banco de dados MongoDB. Além disso, para que o PyMongo faça a conexão com o *client* do banco, utilizaremos a biblioteca Urllib para passarmos a url de endereço do banco. Apenas após estes dois processos, poderemos finalmente armazenar as informações do banco, através do uso das *collections* (tabelas).

```
#criando a conexão através da string de conexão do mongodb atlas
client = MongoClient("mongodb+srv://ifbg5:ifbg5@cluster0.fqk19.mongodb.net/ovni?retryWrites=true&w=majority")
#armazenando a conexão com o banco em db
db = client['ovni']
#armazenando a conexão com as collections (tabelas)
db_c=db['ovnis']
```

Figura 2 – Criando a Conexão com o Banco de Dados MongoDB

À seguir, iniciamos o processo de importação das informações pelo banco de dados. Nosso último arquivo CSV onde preparamos todos os dados para que possam ser importados (item 5.8), agora será importado pelo banco através de uma lista. O código abaixo exibe esse processo.

```
#criando uma lista com os dados do arquivo "df_OVNI_preparado.csv" para exportar os dados para mongodb atlas
with open('../5.8-Acrescimo-de-variaveis/df_OVNI_preparado.csv') as data:
    records=csv.reader(data,delimiter=',')
    lista = []
    i=0
    for record in records:
        dict={}
        if(i>0) :
            cidade=record[0]
            estado=record[1]
            forma=record[2]
            dt=record[3]
            dt_hora=record[4]
            dia_semana=record[5]
            dia=record[6]
            mes=record[7]
            dict={"city":cidade,"state":estado,"shape":forma,"Sight_Date":dt,"Sight_Time":dt_hora,
"Sight_Weekday":dia_semana,"Sight_Day":dia,"Sight_Month":mes}
            lista.append(dict)
        i=i+1
```

Figura 3 – Importação do CSV “df_OVNI_preparado” Pelo Banco de Dados

Agora que o nosso arquivo CSV “df_OVNI_preparado” foi importado pelo banco, daremos inicio à uma série de tarefas propostas. Essas tarefas servem para que possamos consultar e exibir algumas informações interessantes, através do banco de dados.

O primeiro desafio, é contar e exibir todos os documentos (número de registros) que o nosso arquivo CSV passou para o banco de dados. A Figura 5 apresenta o processo, exibindo também um pequeno trecho de exemplo do resultado, e a Figura 6 apresenta o número total de registros importados.

```
n=0
for i in db_c.find():
    if(n>0):
        print(i)
    n+=1
```

```
0:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '9', 'Sight_Month': '8'}
{'_id': ObjectId('5f7f216adb9a691e5859e44e'), 'city': 'Davis', 'state': 'CA', 'shape': 'Light', 'Sight_Date': '2017-08-09', 'Sight_
Time': '21:30:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '9', 'Sight_Month': '8'}
{'_id': ObjectId('5f7f216adb9a691e5859e44f'), 'city': 'Corona', 'state': 'CA', 'shape': 'Changing', 'Sight_Date': '2017-08-09', 'Si
ght_Time': '21:25:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '9', 'Sight_Month': '8'}
{'_id': ObjectId('5f7f216adb9a691e5859e450'), 'city': 'Grand Junction', 'state': 'CO', 'shape': 'Light', 'Sight_Date': '2017-08-0
9', 'Sight_Time': '21:03:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '9', 'Sight_Month': '8'}
```

Figura 4 – Contagem de Número de Registros Importados pelo Banco de Dados

```
#Imprimindo o total de registros
print('O total de registros é:' + ' ' + str(n))
```

```
O total de registros é: 54880
```

Figura 5 – Contagem do Número Total de Registros

Em seguida, faremos a exibição de todos os registros de acordo com o formato de OVNI relatado. Ou seja, baseado na coluna “*shape*”. Seguido também de um resumo do resultado gerado.

```
#fazendo um laço for para exibir todos os documentos na tela ordenando por tipo
for i in db_c.find().sort('shape'):
   print(i)
```

```
rday', 'Sight_Day': '27', 'Sight_Month': '5'}
{'_id': ObjectId('5f7f216adb9a691e5859df98'), 'city': 'Venice', 'state': 'FL', 'shape': 'Triangle', 'Sight_Date': '2017-05-27', 'Si
ght_Time': '03:35:00', 'Sight_Weekday': 'Saturday', 'Sight_Day': '27', 'Sight_Month': '5'}
{'_id': ObjectId('5f7f216adb9a691e5859dfc3'), 'city': 'Silver Lake', 'state': 'KS', 'shape': 'Triangle', 'Sight_Date': '2017-05-2
0', 'Sight_Time': '22:30:00', 'Sight_Weekday': 'Saturday', 'Sight_Day': '20', 'Sight_Month': '5'}
{'_id': ObjectId('5f7f216adb9a691e5859dfce'), 'city': 'Quincy', 'state': 'WA', 'shape': 'Triangle', 'Sight_Date': '2017-05-19', 'Si
ght_Time': '17:49:00', 'Sight_Weekday': 'Friday', 'Sight_Day': '19', 'Sight_Month': '5'}
```

Figura 6 – Exibição de Relatos Com Base Tipo “*Shape*”

Na Figura 7, apresentaremos os registros de relatos de ONVIS, agrupando-os por estado.

```
#usando uma função de agregação para agrupar por estado
def most_states():
    result = db_c.aggregate([
                            { "$group" : {"_id" : {"Estado" : "$state"},
                                                   "count" : { "$sum" : 1 } } },
                            { "$sort" : { "count" : -1 }}
                        ])
    return result

if __name__ == '__main__':
    results = most_states()
    for result in results:
        print(result)
```

```
{'_id': {'Estado': 'CA'}, 'count': 6728}
{'_id': {'Estado': 'FL'}, 'count': 3694}
{'_id': {'Estado': 'WA'}, 'count': 2707}
{'_id': {'Estado': 'TX'}, 'count': 2400}
{'_id': {'Estado': 'NY'}, 'count': 2391}
{'_id': {'Estado': 'PA'}, 'count': 2187}
{'_id': {'Estado': 'AZ'}, 'count': 2012}
{'_id': {'Estado': 'OH'}, 'count': 1881}
{'_id': {'Estado': 'IL'}, 'count': 1817}
```

Figura 7 – Registros de Relatos Agrupados Por Estado

Para tornar a consulta ainda mais específica, criaremos uma *query* para consultar todos os relatos da cidade de Phoenix. A Figura 8 apresenta o código e um resumo do resultado obtido.

```
#Criando uma query para pegar os relatos da cidade phoenix e imprimindo todos os resultados
city_phoneix = {'city': 'Phoenix'}

query_city = db_c.find(city_phoneix)
for x in query_city:
    print(x)
```

```
859b577'), 'city': 'Phoenix', 'state': 'AZ', 'shape': 'Fireball', 'Sight_Date': '2015-02-18', 'Sight_Time': '19:21:00', 'Sight_Week
day': 'Wednesday', 'Sight_Day': '18', 'Sight_Month': '2'}
{'_id': ObjectId('5f7f2169db9a691e5859b5b3'), 'city': 'Phoenix', 'state': 'AZ', 'shape': 'Sphere', 'Sight_Date': '2015-02-11', 'Sig
ht_Time': '21:30:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '11', 'Sight_Month': '2'}
{'_id': ObjectId('5f7f2169db9a691e5859b5b7'), 'city': 'Phoenix', 'state': 'AZ', 'shape': 'Formation', 'Sight_Date': '2015-02-11',
'Sight_Time': '19:30:00', 'Sight_Weekday': 'Wednesday', 'Sight_Day': '11', 'Sight_Month': '2'}
{'_id': ObjectId('5f7f2169db9a691e5859b6ac'), 'city': 'Phoenix', 'state': 'AZ', 'shape': 'Disk', 'Sight_Date': '2015-03-20', 'Sight
_Time': '14:30:00', 'Sight_Weekday': 'Friday', 'Sight_Day': '20', 'Sight_Month': '3'}
{'_id': ObjectId('5f7f2169db9a691e5859b701'), 'city': 'Phoenix', 'state': 'AZ', 'shape': 'Circle', 'Sight_Date': '2015-03-14', 'Sig
ht_Time': '20:00:00', 'Sight_Weekday': 'Saturday', 'Sight_Day': '14', 'Sight_Month': '3'}
```

Figura 8 – Registros de Relatos do Estado de Phoenix

Por fim, com um propósito semelhante ao acima, através de uma consulta apresentaremos todos os registros de relatos ligados ao estado da Califórnia, com exceção do campo “ID” relacionado a esses registros.

```
#Imprimindo todos os relatos da Califórnia e excluindo o campo id do resultado
for x in db_c.find({'state': 'CA'},{ "_id": 0}):
  print(x)
```

```
:50:00', 'Sight_Weekday': 'Thursday', 'Sight_Day': '29', 'Sight_Month': '6'}
{'city': 'Seaside', 'state': 'CA', 'shape': 'Sphere', 'Sight_Date': '2017-06-27', 'Sight_Time': '21:00:00', 'Sight_Weekday': 'Tuesd
ay', 'Sight_Day': '27', 'Sight_Month': '6'}
{'city': 'Palm Springs', 'state': 'CA', 'shape': 'Rectangle', 'Sight_Date': '2017-06-27', 'Sight_Time': '02:31:00', 'Sight_Weekda
y': 'Tuesday', 'Sight_Day': '27', 'Sight_Month': '6'}
{'city': 'Lincoln', 'state': 'CA', 'shape': 'Sphere', 'Sight_Date': '2017-06-26', 'Sight_Time': '22:00:00', 'Sight_Weekday': 'Monda
y', 'Sight_Day': '26', 'Sight_Month': '6'}
{'city': 'Korbel', 'state': 'CA', 'shape': 'Flash', 'Sight_Date': '2017-06-26', 'Sight_Time': '00:00:00', 'Sight_Weekday': 'Monda
y', 'Sight_Day': '26', 'Sight_Month': '6'}
{'city': 'Long Beach', 'state': 'CA', 'shape': 'Light', 'Sight_Date': '2017-06-25', 'Sight_Time': '22:15:00', 'Sight_Weekday': 'Sun
day', 'Sight_Day': '25', 'Sight_Month': '6'}
```

Figura 9 – Registros de Relatos do Estado da Califórnia com a Exclusão do Campo “ID”

## 4. Considerações Finais

Agora que as informações relacionadas à OVNIS foram coletadas, exploradas e preparadas para análise, transferir essas informações a um banco de dados nos trouxe diversas vantagens. Vantagens estas já apresentadas ao longo deste documento, todavia também apresentadas na prática, com a realização de consultas e a organização de informações específicas.

Na etapa final que está por vir, finalizaremos o projeto realizando uma análise mais detalhada desses dados registrados no nosso banco de dados, com o objetivo de extrairmos valor dessas informações dispostas.

Os códigos e o arquivos CSV desenvolvidos no decorrer desta etapa, encontram-se no no repositório Git Hub do grupo, no endereço https://github.com/Prof-Fabio-Henrique/pratica-integrada-icd-e-iia-2020-1-g5-gmp/tree/master/5.10-Dados-no-mongodb.

Para tanto, o uso da linguagem Python e suas bibliotecas foi essencial. Através da integração de ambos, o campo da Ciência de Dados, 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.