

**TECNOLOGIA EM SISTEMAS PARA INTERNET**

**Matheus Nícollas de Souza Mota  
Thiago Marinho da Silva Campos**

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

**Brasília - DF  
17/10/2020**

# Sumário

1. Objetivos	3
2. Descrição do problema	4
3. Desenvolvimento	5
4 Código implementado	6
5 Considerações Finais	10
6 Referencias	11

# 1. Objetivos

O objetivo é trabalhar com a quarta etapa de um pipeline de dados ou seja, o armazenamento usando MongoDB.

## 2. Descrição do problema

O projeto consiste em reunir fatos interessantes relacionados a OVINIs, a partir de relatos realizados dentro de um período de vinte anos usando o site Nuforc. O desafio consiste em fazer a extração de dados de forma tabular, afinal, para que os dados possam ser analisados eles acabam se tornando tabelas.

O WebScraping consiste em extrair os dados formatados com tag's da linguagem HTML. Como iremos extrair vinte anos de dados, consultaremos 240 páginas web, uma por cada mês, por vinte anos, entre setembro 1997 e agosto de 2017.

Para fazer análises é necessário que os dados estejam apresentados de forma ordenada e de uma maneira funcional para que facilite a realização de pesquisas.

Após o tratamento de dados precisamos armazená-los, e é isto que será feito nesta etapa do projeto utilizando o MongoDB.

### 3. Desenvolvimento

O desenvolvimento do algoritmo foi feito na plataforma Google Collab, esta plataforma foi escolhida pois ao iniciar um notebook na mesma, as bibliotecas e dependências do Python são todas da nuvem.

Criamos uma conta no mLab, fizemos login no MongoDB Atlas, configuramos novo cluster e nova conexão, fizemos acesso ao string de conexão e configuramos o menu security para acesso via rede.

Usamos o módulo PYMONGO para estabelecer conexão com o banco, criamos o banco ovini, uma coleção com nome óvnis e inserimos os dados preparados.

Usamos as funções do pymongo para contar e mostrar quantos documentos há na coleção óvnis, resgatar todos os documentos da coleção ovini e ordenar por tipo. Verificamos quantas ocorrências por estado. Buscamos todas as ocorrências da cidade Phoenix e todas as ocorrências do estado da California ocultando o id de cada documento.

## 4 Código implementado

```
#Instala as dependencias necessarias para conexao com o banco de dados
!curl ipecho.net/plain
!pip install pymongo
!pip install dnspython

35.196.20.36Requirement already satisfied: pymongo in
/usr/local/lib/python3.6/dist-packages (3.11.0)
Collecting dnspython
  Downloading
https://files.pythonhosted.org/packages/90/49/cb426577c28ca3e35332815b795a99e467523843fc83cc85ca0d6be2515a/dnspython-2.0.0-py3-none-any.whl
(208kB)
  |████████████████████████████████████████| 215kB 2.9MB/s
Installing collected packages: dnspython
Successfully installed dnspython-2.0.0
```

```
import pandas as pd
import pymongo
from pymongo import MongoClient
import pprint

#abrindo conexão com Atlas
client = pymongo.MongoClient("mongodb://matheus:123@cluster0-shard-00-00.ja2pz.mongodb.net:27017,cluster0-shard-00-01.ja2pz.mongodb.net:27017,cluster0-shard-00-02.ja2pz.mongodb.net:27017/?ssl=true&replicaSet=atlas-13rhks-shard-0&authSource=admin&retryWrites=true&w=majority")
#criando um Banco de dados chamado ovni
db = client.ovni
#Criando uma coleção chamada ovnis.
ovnis = db.ovnis
#Leitura do arquivo df_OVNI_preparado.csv
df_ovnis = pd.read_csv('df_OVNI_preparado.csv')
ovnis.insert_many(df_OVNI_preparado.to_dict('records'))

#Contar e mostrar quantos documentos há na coleção ovnis.
print("Número de Documentos na coleção ovnis:", ovnis.count_documents({}))
```

Número de Documentos na coleção ovnis: 351420

```

#Resgatar todos os documentos (registros) da coleção ovnis e ordenar p
or tipo (shape).
shapes = ovnis.find({}).sort('Shape')

for shape in shapes:
    pprint.pprint(shape)

# Verificar quantas ocorrências existem por estado.
# Usa a função aggregate() para manipular as collections e adiciona o
valor 1 para dizer ao $sum para somar todos os registros.
results = db.ovnis.aggregate([
    {
        "$group": {
            "_id": "$State",
            "Views": {"$sum": 1}
        }
    })

for result in results:
    print(result)
{ '_id': 'NE', 'Views': 1385}
{ '_id': 'MO', 'Views': 5825}
{ '_id': 'ND', 'Views': 495}
{ '_id': 'CT', 'Views': 4095}
{ '_id': 'NY', 'Views': 11955}
{ '_id': 'WY', 'Views': 750}
{ '_id': 'NJ', 'Views': 6025}
{ '_id': 'MD', 'Views': 4165}
{ '_id': 'KS', 'Views': 2395}
{ '_id': 'CA', 'Views': 33640}
{ '_id': 'IN', 'Views': 5235}
{ '_id': 'ME', 'Views': 2540}
{ '_id': 'MT', 'Views': 2045}
{ '_id': 'VT', 'Views': 1480}
{ '_id': 'WI', 'Views': 5240}
{ '_id': 'AL', 'Views': 2880}
{ '_id': 'MN', 'Views': 4600}
{ '_id': 'GA', 'Views': 6015}
{ '_id': 'PA', 'Views': 10940}
{ '_id': 'OR', 'Views': 7095}
{ '_id': 'SC', 'Views': 5790}
{ '_id': 'NM', 'Views': 3260}
{ '_id': 'WA', 'Views': 13535}
{ '_id': 'LA', 'Views': 2470}
{ '_id': 'OH', 'Views': 9405}
{ '_id': 'FL', 'Views': 18470}
{ '_id': 'IL', 'Views': 9085}
{ '_id': 'RI', 'Views': 1365}
{ '_id': 'NC', 'Views': 8605}
{ '_id': 'NH', 'Views': 2435}
{ '_id': 'WV', 'Views': 1850}
{ '_id': 'MI', 'Views': 7885}
{ '_id': 'TX', 'Views': 12000}
{ '_id': 'IA', 'Views': 2655}

```

```
# Buscar todas as ocorrências da cidade Phoenix.
phoenix = ovnis.find({"City": "Phoenix"}).sort("shape")
for p in phoenix:
    pprint.pprint(p)
```

**A saída de streaming foi truncada nas últimas 5000 linhas.**

```
'Shape': 'Circle',
'Sight_Date': '2010-10-22',
'Sight_Day': 22,
'Sight_Month': 10,
'Sight_Time': '21:30',
'Sight_Weekday': 'Sexta-feira',
'State': 'AZ',
'Unnamed: 0': 24587,
'_id': ObjectId('5f88a6d3bbf26006c021f1a6')}}
{'City': 'Phoenix',
'Shape': 'Light',
'Sight_Date': '2010-10-17',
'Sight_Day': 17,
'Sight_Month': 10,
'Sight_Time': '04:53',
'Sight_Weekday': 'Domingo',
'State': 'AZ',
'Unnamed: 0': 24656,
'_id': ObjectId('5f88a6d3bbf26006c021f1de')}}
{'City': 'Phoenix',
'Shape': 'Formation',
'Sight_Date': '2011-04-03',
'Sight_Day': 3,
'Sight_Month': 4,
'Sight_Time': '22:30',
'Sight_Weekday': 'Domingo',
'State': 'AZ',
```



```
# Buscar as ocorrências do estado da Califórnia e ocultar o id de cada
# documento (registro).
# para ocultar o id basta passar o valor 0 na collection.
california = ovnis.find({"State": "CA"}, {"_id": 0}).sort("Shape")
for ca in california:
    pprint.pprint(ca)
```

**A saída de streaming foi truncada nas últimas 5000 linhas.**

```
'Sight_Time': '21:30',
'Sight_Weekday': 'Segunda-feira',
'State': 'CA',
'Unnamed: 0': 7533}
{'City': 'Castro Valley',
'Shape': 'Triangle',
'Sight_Date': '2007-06-16',
'Sight_Day': 16,
'Sight_Month': 6,
'Sight_Time': '22:00',
'Sight_Weekday': 'Sábado',
'State': 'CA',
'Unnamed: 0': 7713}
{'City': 'Fresno',
'Shape': 'Triangle',
'Sight_Date': '2007-06-07',
'Sight_Day': 7,
'Sight_Month': 6,
'Sight_Time': '22:00',
'Sight_Weekday': 'Quinta-feira',
'State': 'CA',
'Unnamed: 0': 7830}
{'City': 'Fresno',
'Shape': 'Triangle',
```

Github: <https://github.com/Prof-Fabio-Henrique/pratica-integrada-icd-e-ia-2020-1-g13-mmt>

## 5 Considerações Finais

Com esta etapa do projeto podemos fechar o ciclo no tratamento dos dados, depois de preparados, armazenamos os mesmos em bancos de dados para futuras consultas e análises.

## 6 Referencias

MONGO DB. The mongoDB 4.4 Manual. Disponível em:  
<https://docs.mongodb.com/manual/>. Acesso em : 15/10/2020