

BANCO DE DADOS NÃO RELACIONAL - Prof. Me. Anderson Vanin**Atividade: Visualização de Pontos Turísticos em um Mapa****Objetivo:**

Armazenar dados de pontos turísticos com suas coordenadas GPS no MongoDB Atlas.

Utilizar o Google Colab para acessar o banco de dados e visualizar esses pontos turísticos em um mapa.

Passos:**1. Configuração do MongoDB Atlas:**

- Criar uma conta no MongoDB Atlas (se ainda não tiver).
- Criar um novo cluster.
- Criar uma base de dados e uma coleção para armazenar os dados dos pontos turísticos.
- Inserir alguns documentos na coleção, cada um contendo informações sobre um ponto turístico, incluindo suas coordenadas GPS.

2. Configuração do Google Colab:

- Abrir o Google Colab e criar um novo notebook.
- Instalar o pacote pymongo para conectar-se ao MongoDB Atlas.
- Instalar o pacote folium para criar mapas interativos.

3. Conectar-se ao MongoDB Atlas:

- Utilizar as credenciais fornecidas pelo MongoDB Atlas para conectar-se ao seu cluster.

4. Recuperar os Dados dos Pontos Turísticos:

- Executar consultas para recuperar os dados dos pontos turísticos da coleção no MongoDB Atlas.

5. Visualizar os Pontos Turísticos em um Mapa:

- Utilizar o pacote folium para criar um mapa interativo.

Faculdade de Tecnologia
Fatec Mauá

- Adicionar marcadores no mapa para cada ponto turístico, utilizando suas coordenadas GPS.
- Personalizar os marcadores com informações adicionais, como o nome do ponto turístico.

6. Conclusão:

Os alunos podem experimentar alterar as consultas no Colab para recuperar diferentes conjuntos de dados dos pontos turísticos.

Eles também podem explorar outras funcionalidades do folium para personalizar ainda mais o mapa.

Essa atividade proporciona aos alunos uma experiência prática de trabalhar com um banco de dados NoSQL (MongoDB Atlas) e de criar visualizações de dados interativas em um ambiente colaborativo (Google Colab).

Faculdade de Tecnologia
Fatec Mauá

Prática

1. Crie o documento contendo nome, latitude, longitude e descrição dos locais de interesse.
2. Configuração do MongoDB Atlas:

pontos_turisticos.locais

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

QUERY RESULTS: 1-16 OF 16

```
{
  "_id": ObjectId('662a94a3ab6efe86664a4fe8'),
  "nome": "Cristo Redentor",
  "latitude": -22.9519,
  "longitude": -43.2105,
  "descricao": "O Cristo Redentor é uma estátua art déco localizada no topo do morro d..."
}
```

```
{
  "_id": ObjectId('662a94a3ab6efe86664a4fe9'),
  "nome": "Pão de Açúcar",
  "latitude": -22.9503,
  "longitude": -43.1564,
  "descricao": "O Pão de Açúcar é um complexo de montanhas localizado no Rio de Janeir..."
}
```

3. Configuração do Google Colab:

```
# Instalar o pacote pymongo para se conectar ao MongoDB Atlas
!python -m pip install pymongo

# Instalar o pacote folium para criar mapas interativos
!python -m pip install folium

import folium
from pymongo import MongoClient
```

Collecting pymongo
Downloading pymongo-4.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
670.1/670.1 kB 4.5 MB/s eta 0:00:00

Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
307.7/307.7 kB 29.5 MB/s eta 0:00:00

Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.6.1 pymongo-4.7.0
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from folium) (0.7.1)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from folium) (3.1.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from folium) (1.25.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from folium) (2.31.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->folium) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2024.2.2)

4. Conectar-se ao MongoDB Atlas

Faculdade de Tecnologia
Fatec Mauá

```
[2] # Conectar ao MongoDB Atlas
cliente = MongoClient("mongodb+srv://[REDACTED]@cluster0.temjb4o.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0")

[3] cliente

MongoClient(host=['ac-x2dgwyy-shard-00-01.temjb4o.mongodb.net:27017', 'ac-x2dgwyy-shard-00-02.temjb4o.mongodb.net:27017', 'ac-x2dgwyy-shard-00-00.temjb4o.mongodb.net:27017'],
document_class=dict, tz_aware=False, connect=True, retrywrites=True, w='majority', appname='Cluster0', authsource='admin', replicaset='atlas-hpmi8-shard-0', tls=True)
```

5. Recuperar os Dados dos Pontos Turísticos

```
[93] # Selecionar a coleção de pontos turísticos
db = cliente.pontos_turisticos
collection = db.locais

# Recuperar os dados dos pontos turísticos
# pontos = collection.find()
pontos = list(collection.find())

[95] list(pontos)

{'_id': ObjectId('662a94a3abefe86664a4fee'),
 'nome': 'Cataratas do Iguaçu',
 'latitude': -25.6953,
 'longitude': -54.4367,
 'descricao': "As Cataratas do Iguaçu são um conjunto de cachoeiras localizadas na fronteira entre o Brasil e o Paraguai, sendo uma das maiores atrações turísticas."},
{'_id': ObjectId('662a94a3abefe86664a4fef'),
 'nome': 'Parque Nacional do Iguaçu',
 'latitude': -25.6841,
 'longitude': -54.4523,
```

6. Visualizar os Pontos Turísticos em um Mapa

```
[25] # Estabelecendo um ponto central do mapa (Centro do Brasil)
center_lat = -14.24007
center_lon = -53.18050

[26] # Cria um mapa a partir do ponto central com 800x600
mapa = folium.Map(location=[center_lat, center_lon], zoom_start=5, width=800, height=600)

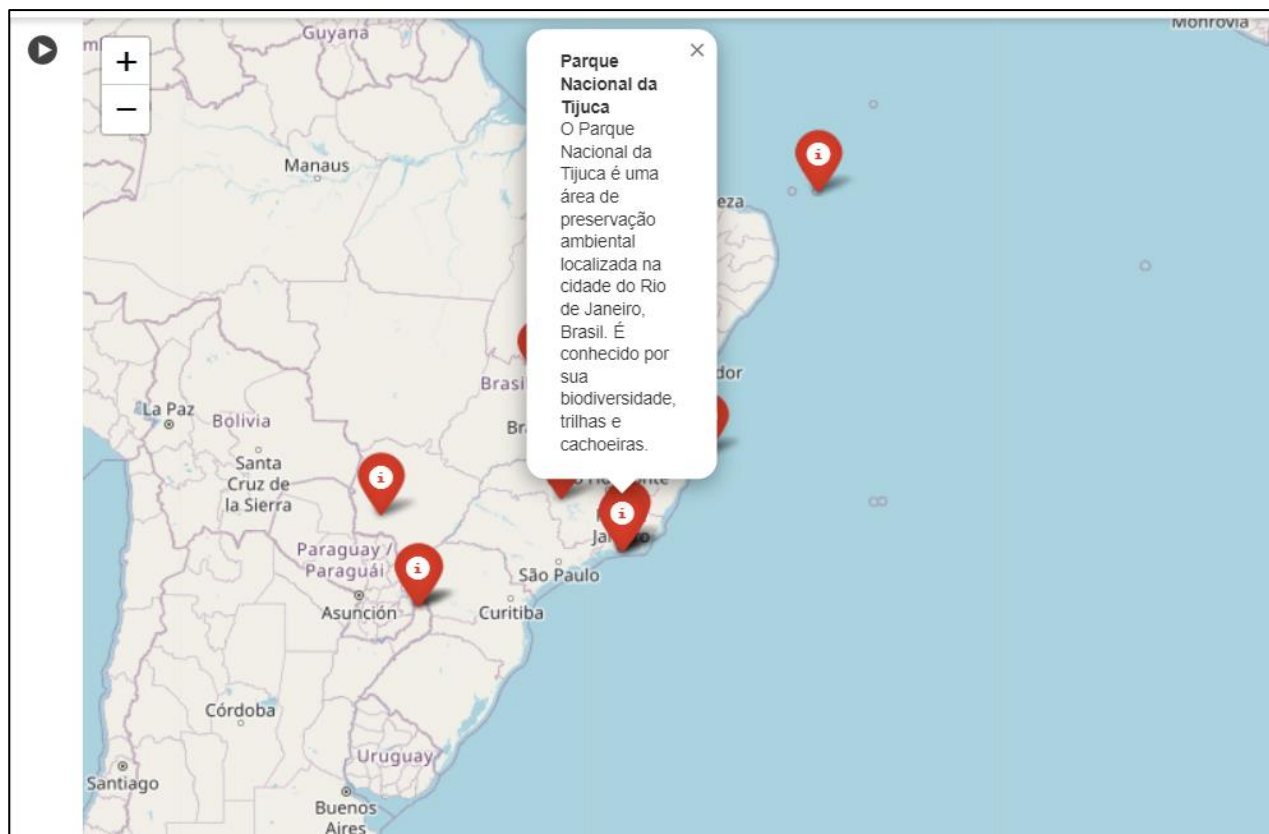
[7] # Criar um mapa folium
mapa = folium.Map(location=[-23.55052, -46.633308], zoom_start=10)
```

```
# Adicionar marcadores para cada ponto turístico
for ponto in pontos:
    nome = ponto['nome']
    latitude = ponto['latitude']
    longitude = ponto['longitude']
    descricao = ponto['descricao']

    # Adicionar marcador com informações adicionais
    folium.Marker([latitude, longitude], popup=f"<b>{nome}</b><br>{descricao}", icon=folium.Icon(color='red')).add_to(mapa)

# Exibir o mapa
mapa
```

Faculdade de Tecnologia
Fatec Mauá



Algumas sugestões para melhoria da atividade

1. Criar um filtro para que seja possível escolher um dos locais e marcá-lo no mapa individualmente.
2. Criar opções para inserir, alterar ou remover um destes pontos.

```
✓ [40] # Recuperar os nomes dos pontos turísticos
0s nomes_pontos_turisticos = [ponto['nome'] for ponto in collection.find({}, {'nome': 1, '_id': 0})]
nomes_pontos_turisticos

['Cristo Redentor',
 'Pão de Açúcar',
 'Parque Nacional da Tijuca',
 'Fernando de Noronha',
 'Bonito',
 'Lençóis Maranhenses',
 'Cataratas do Iguaçu',
 'Parque Nacional do Iguaçu',
 'Parque Nacional da Chapada Diamantina',
 'Parque Nacional dos Lençóis Maranhenses',
 'Parque Nacional da Serra da Canastra',
 'Parque Nacional da Serra dos Órgãos',
 'Parque Nacional da Chapada dos Veadeiros',
 'Parque Nacional da Serra do Cipó',
 'Parque Nacional de Jericoacoara',
 'Parque Nacional Marinho dos Abrolhos']
```

```
✓ [42] # Criar um filtro para selecionar um ponto turístico
4s print("Escolha um ponto turístico:")
for i, nome in enumerate(nomes_pontos_turisticos, 1):
    print(f"{i}. {nome}")

opcao = int(input("Digite o número do ponto turístico desejado: "))

Escolha um ponto turístico:
1. Cristo Redentor
2. Pão de Açúcar
3. Parque Nacional da Tijuca
4. Fernando de Noronha
5. Bonito
6. Lençóis Maranhenses
7. Cataratas do Iguaçu
8. Parque Nacional do Iguaçu
9. Parque Nacional da Chapada Diamantina
10. Parque Nacional dos Lençóis Maranhenses
11. Parque Nacional da Serra da Canastra
12. Parque Nacional da Serra dos Órgãos
13. Parque Nacional da Chapada dos Veadeiros
14. Parque Nacional da Serra do Cipó
15. Parque Nacional de Jericoacoara
16. Parque Nacional Marinho dos Abrolhos
Digite o número do ponto turístico desejado: 10
```

Faculdade de Tecnologia
Fatec Mauá

```

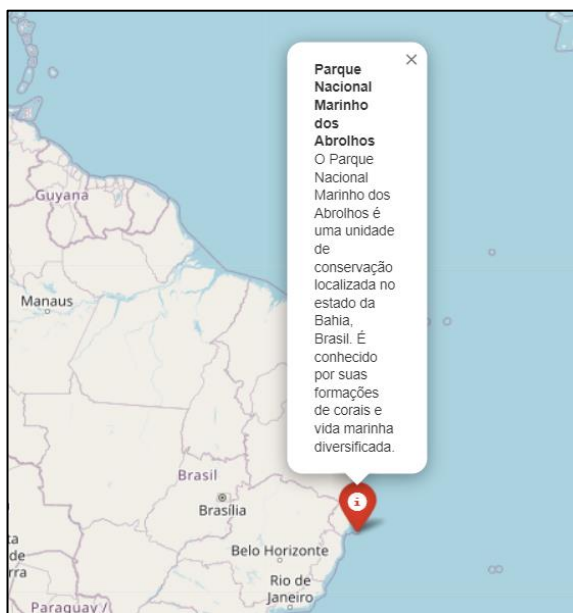
0s [ ] # Recuperar as coordenadas do ponto turístico selecionado
ponto_selecionado = collection.find_one({"nome": nomes_pontos_turisticos[opcao - 1]})
nome_escolhido = ponto_selecionado['nome']
latitude_escolhido = ponto_selecionado['latitude']
longitude_escolhido = ponto_selecionado['longitude']
descricao_escolhido = ponto_selecionado['descricao']

# Criar um mapa folium
mapa2 = folium.Map(location=[-23.55052, -46.633308], zoom_start=10)

# Adicionar marcador para o ponto turístico selecionado
folium.Marker([latitude_escolhido, longitude_escolhido], popup=f"<b>{nome_escolhido}</b><br>{descricao_escolhido}", icon=folium.Icon(color='red')).add_to(mapa2)

# Exibir o mapa
mapa2

```



```

0s [111] from bson import ObjectId

0s [ ] # Delete e Update de pontos turísticos

# Função para listar os pontos
def listar_pontos():
    print("Pontos turísticos:")
    for ponto in collection.find({}, {'_id': 1, 'nome': 1}):
        print("-", ponto['_id'], ponto['nome'])

```

Faculdade de Tecnologia
Fatec Mauá

```
✓ [0s] [▶] # Função para inserir um novo ponto
def inserir_ponto():
    nome = input("Nome do ponto turístico: ")
    latitude = float(input("Latitude: "))
    longitude = float(input("Longitude: "))
    descricao = input("Descrição: ")

    novo_ponto = {
        "nome": nome,
        "latitude": latitude,
        "longitude": longitude,
        "descricao": descricao
    }

    collection.insert_one(novo_ponto)
    print("Ponto turístico inserido com sucesso.")
```

```
✓ [0s] [116] # Função para alterar um ponto pelo ID
def alterar_ponto():
    listar_pontos()
    id_ponto = input("Digite o ID do ponto turístico que deseja alterar: ")
    novo_nome = input("Novo nome (deixe em branco para manter o mesmo): ")
    nova_latitude = input("Nova latitude (deixe em branco para manter a mesma): ")
    nova_longitude = input("Nova longitude (deixe em branco para manter a mesma): ")
    nova_descricao = input("Nova descrição (deixe em branco para manter a mesma): ")

    update_fields = {}
    if novo_nome:
        update_fields["nome"] = novo_nome
    if nova_latitude:
        update_fields["latitude"] = float(nova_latitude)
    if nova_longitude:
        update_fields["longitude"] = float(nova_longitude)
    if nova_descricao:
        update_fields["descricao"] = nova_descricao

    if update_fields:
        collection.update_one({"_id": ObjectId(id_ponto)}, {"$set": update_fields})
        print("Ponto turístico alterado com sucesso.")
    else:
        print("Nenhuma alteração realizada.")
```

```
✓ [0s] [117] # Função para Remover um ponto
def remover_ponto():
    listar_pontos()
    id_ponto = input("Digite o ID do ponto turístico que deseja remover: ")
    collection.delete_one({"_id": ObjectId(id_ponto)})
    print("Ponto turístico removido com sucesso.")
```


Faculdade de Tecnologia
Fatec Mauá

✓
2min

▶

```
# Menu de opções
while True:
    print("\nOpções:")
    print("1. Listar pontos turísticos")
    print("2. Inserir ponto turístico")
    print("3. Alterar ponto turístico")
    print("4. Remover ponto turístico")
    print("5. Sair")

    opcao = int(input("Escolha uma opção: "))

    if opcao == 1:
        listar_pontos()
    elif opcao == 2:
        inserir_ponto()
    elif opcao == 3:
        alterar_ponto()
    elif opcao == 4:
        remover_ponto()
    elif opcao == 5:
        break
    else:
        print("Opção inválida.")

# Encerrar a conexão com o MongoDB Atlas
#cliente.close()
print('Saiu do menu')
```

Faculdade de Tecnologia
Fatec Mauá

Execuções:

Opções:

1. Listar pontos turísticos
2. Inserir ponto turístico
3. Alterar ponto turístico
4. Remover ponto turístico
5. Sair

Escolha uma opção: 2

Nome do ponto turístico: Parque Ibirapuera

Latitude: -23.587646

Longitude: -46.658822

Descrição: O Parque Ibirapuera é um dos principais parques urbanos da cidade de São Paulo, Brasil.

Ponto turístico inserido com sucesso.

Opções:

1. Listar pontos turísticos
2. Inserir ponto turístico
3. Alterar ponto turístico
4. Remover ponto turístico
5. Sair

Escolha uma opção: 1

Pontos turísticos:

- 662a94a3ab6efe86664a4fe8 Cristo Redentor
- 662a94a3ab6efe86664a4fe9 Pão de Açúcar
- 662a94a3ab6efe86664a4fea Parque Nacional da Tijuca
- 662a94a3ab6efe86664a4feb Fernando de Noronha
- 662a94a3ab6efe86664a4fec Bonito - Alterado
- 662a94a3ab6efe86664a4fed Lençóis Maranhenses
- 662a94a3ab6efe86664a4fee Cataratas do Iguaçu
- 662a94a3ab6efe86664a4fef Parque Nacional do Iguaçu
- 662a94a3ab6efe86664a4ff0 Parque Nacional da Chapada Diamantina
- 662a94a3ab6efe86664a4ff1 Parque Nacional dos Lençóis Maranhenses
- 662a94a3ab6efe86664a4ff2 Parque Nacional da Serra da Canastra
- 662a94a3ab6efe86664a4ff3 Parque Nacional da Serra dos Órgãos
- 662a94a3ab6efe86664a4ff4 Parque Nacional da Chapada dos Veadeiros
- 662a94a3ab6efe86664a4ff5 Parque Nacional da Serra do Cipó
- 662a94a3ab6efe86664a4ff6 Parque Nacional de Jericoacoara
- 662a94a3ab6efe86664a4ff7 Parque Nacional Marinho dos Abrolhos
- 662aab967f1f56f410710405 Parque Ibirapuera

Opções:

1. Listar pontos turísticos
2. Inserir ponto turístico
3. Alterar ponto turístico
4. Remover ponto turístico
5. Sair

Escolha uma opção: 5

Saiu do menu