**Problema:**

Nesta última parte do trabalho, tinha como problema a ser resolvido realizar o armazenamento do pipeline dos dados e armazenar esses dados utilizando o mongoDb.

**Solução:**

Para executar essas partes do trabalho teve a sua divisão das tarefas em etapas.

**Primeira Etapa:**

Na primeira etapa a atividade que precisava ser feita era a criação de uma no mongoDb Atlas, para criar essa conta precisou colocar o email, nome, sobrenome, senha e aceitar os termos e por último clicar em Create Account.

A imagem 1 mostrar o resultado obtido na criação da conta no mongoDb.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 1**

**Segunda etapa:**

Depois de criar a conta precisa selecionar a opção gratuita para iniciar a Clusters. A imagem 2 mostra o resultado obtido.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 2**

**Terceira etapa:**

A atividade desta etapa era escolher o provedor e a região , e foram escolhidos o AWS e o estado da Virginia. A imagem 3 demostrar como ocorreu a escolha.

**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

**imagem 3**

**Quarta etapa:**

A imagem 4 mostrar como ocorreu a escolha da cluster com a memória de 512Mb.

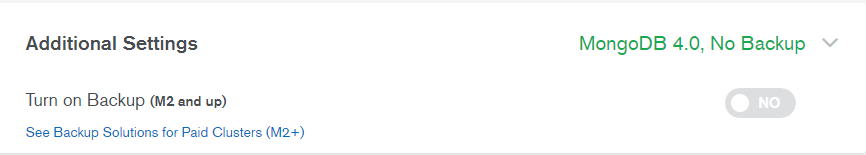
**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

**imagem 4**

**Quinta etapa:**

A imagem 5 mostra a opção de não escolher fazer o backup no mongoDb.



**imagem 5**

**Sexta etapa:**

A imagem 6 mostrar como realizou a definição do nome para a cluster, o nome escolhido foi Nuforc.

**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

**imagem 6**

**Sétima etapa:**

A imagem 7 mostrar a criação da cluster.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**Oitava etapa:**

**imagem 7**

A imagem 8 demostrar a inicialização da cluster.

**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

**imagem 8**

**A nona etapa:**

Na nona etapa precisava configurar a conexão do Database Acess. Para configurar o banco de dados com nome do usuário e senha. O resultado desta criação está sendo demostrada nas imagens 9,10 e 11.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 9**

**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

**imagem 10**

Mostra o nome do usuário e a senha

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 11**

**Decima etapa:**

Fazer a configuração do acesso a internet, e nesta opção precisa permitir conexão para todos os ips. O resultado está sendo exibido na imagens 12,13 e 14.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 12**

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 13**

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 14**

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 15**

Uma imagem contendo captura de tela

Descrição gerada automaticamente

**imagem 16**

**Decima primeira etapa:**

Nesta etapa realizou a conexão com driver e depois selecionar a conexão string only, copiou essa conexão e colou no código do colab.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

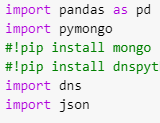
**imagem 17**

**Decima segunda etapa:**

A tarefa realizada definida nesta etapa foi a execução do tutorial do python mongoDb para conhecer como funciona o pymongo e o mongoDb.Para utilizar o módulo PyMongo, precisar fazer a instalação deste módulo usando o comando !pip install dnspython e depois de fazer essa instalação precisa reiniciar o colab, e restart runtime, para estabelecer conexão de dados com o banco de dados.

**Decima terceira etapa:**

Para fazer essa tarefa teve com operação a ser realizada foi a importação do pymongo, e json. E a instalação do mongo e do dnspyt.



**Decima quarta etapa:**

Para realizar a conexão definiu uma variável o comando ocorreu a conexão, mais o nome do banco de dados e a senha.



**Imagem 18**



**Decima quinta etapa:**

A imagem 19 mostra como realizou a criação do banco de dados denominado ovni.

Uma imagem contendo interior

Descrição gerada automaticamente

**Imagem 19**



**Decima sexta etapa:**

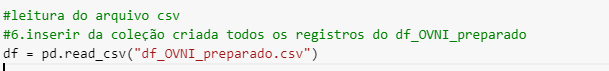
A imagem 20 mostra como realizou a criação da coleção denominado ovnis.



**Imagem 20**

**Decima sétima etapa:**

A imagem 21 mostra a inserção da coleção usando o arquivo df\_OVNI\_preparado.



**Imagem 21**

**Decima oitava etapa:**

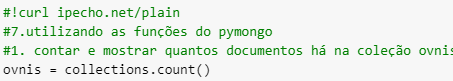
A imagem 22 mostra como ocorreu a transformação em json.

Uma imagem contendo objeto

Descrição gerada automaticamente

**Imagem 22**

A imagem 23 mostrar como ocorreu a contagem das ocorrências em ovnis, e retornou a quantidade de 77902.



**Imagem 23**



**Imagem 24**

A imagem 25 demostrar como realizou o resgate dos registros e a ordenação por Shape.



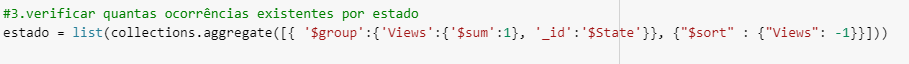
**Imagem 25**

Uma imagem contendo texto

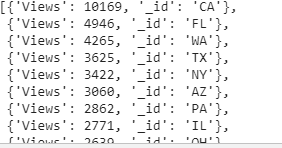
Descrição gerada automaticamente

**Imagem 26**

A imagem 27 demostra o resultado obtido para mostra as ocorrências por estados.



**Imagem 27**



**Imagem 28**

A imagem 29 mostra como ocorreu a busca de todas as ocorrências por Phoenix.



**Imagem 29**

Uma imagem contendo texto

Descrição gerada automaticamente

Uma imagem contendo texto

Descrição gerada automaticamente

**Imagem 30**

**A imagem 31 demostra como ocorreu aa ocorrência por estados da Califórnia.**



**Imagem 31**

Uma imagem contendo texto

Descrição gerada automaticamente

**Imagem 32**

**Link Para o Github:**

<https://github.com/vanessaformiga/Nuforc/tree/master/dadosMongoDB>

**Referências:**

<https://www.w3schools.com/python/python_mongodb_getstarted.asp>

<https://colab.research.google.com/>

<https://www.mongodb.com/>

<https://docs.mongodb.com/>

<https://api.mongodb.com/python/current/index.html>