AVALIAÇÃO DATAOPS

Sumário

[Avaliação DataOps 3](#_Toc126251820)

[1. Desenvolvimento 4](#_Toc126251821)

[1.1. Criação MongoDB Local 4](#_Toc126251822)

[1.2. Criação do Pandas Dataframe 4](#_Toc126251823)

[1.3. Salvar Pandas Dataframe no MongoDB 5](#_Toc126251824)

[1.4. Criando Agregação no MongoDB 5](#_Toc126251825)

[1.5. Agrupando informações 5](#_Toc126251826)

[2. Salvando os scripts 6](#_Toc126251827)

[3. Finalização 6](#_Toc126251828)

[4. Desempenho 7](#_Toc126251829)

# Autoavaliação DataOps

O intuito deste teste é avaliar o desempenho em relação as ferramentas que serão utilizadas dentro do time de DataOps. Para este teste, iremos utilizar as seguintes ferramentas, MongoDB e Python.

Apenas para critério de conhecimento, responda as questões abaixo de acordo com sua percepção:

Como você considera seu nível de conhecimento em Python?

**Avançado**

Como você considera seu nível de conhecimento em MongoDB?

**Intermediário**

Como você considera seu nível de conhecimento em ETL?

**Intermediário**

Como você considera seu nível de conhecimento em Pentaho?

**Nenhum**

# O Desenvolvimento

Para o teste prático, será necessário criar um script em python, onde irá conter dois pandas dataframe populados com os dados pré-definidos, esses dataframes deverão ser enviados e salvos em um banco de dados MongoDB, na sequência, deverá ser desenvolvido uma agregação onde unira essas duas collection de acordo com o campo informado.

Para realizar estes testes, deverá seguir os seguintes passos descritos abaixo:

# Criação MongoDB Local

Instalar o MongoDB local em sua máquina, e criar duas collection “Carros” e “Montadoras”, por enquanto estas collection ficaram vazias, e será populada nos próximos passos.

Obs. para facilitar a utilização do MongoDB, poderá utilizar o MongoDB Compass (Ou outro de sua preferência) para interface dos dados.

# Criação do Pandas Dataframe

O início de teste se baseia em criar um arquivo em python contendo dois pandas dataframe, o primeiro onde será armazenado as informações do carro, e o segundo onde será armazenado as informações da montadora.

Para o primeiro dataframe, o resultado esperado é:

‘P

Para o segundo dataframe, o resultado esperado é:



# Salvar Pandas Dataframe no MongoDB

Após criar os pandas dataframes, definidos no passo anterior, devera salva-los nas collection correspondentes no MongoDB.

Para isso, será necessário criar uma conexão entre o Python e o MongoDB para persistir estas informações no Banco de dados.

Sendo o primeiro dataframe salvo na collection “Carros” e o segundo dataframe salvo na collection “Montadoras”.

# Criando Agregação no MongoDB

Com os dados salvos no MongoDB, o objetivo atual, é fazer o relacionamento entre as duas collection “Carros” e “Montadoras”, para esta relação vamos utilizar a coluna “Montadora” disponibilizadas nas duas collection.

Com o resultado deste relacionamento, precisamos extrair o campo País, de modo a obter o seguinte resultado:



# Agrupando informações

Ainda dentro da mesma agregação criada no passo anterior, será necessário, agrupar os dados pelo campo de “País”, de modo que as informações sobre o carro, fique inserida dentro de um elemento chamado “Carros”, como representado na imagem abaixo:



Após finalizar a agregação, salve-a em um arquivo .js

# Salvando os scripts

Ao finalizar os passos anteriores, salve todos os scripts desenvolvidos, exporte as duas collection criadas no MongoDB, e faça o upload em um repositório GIT com o nome de sua preferência, certifique-se que o repositório esteja visível para o público. Finalize também o questionário inserido no início e fim deste documento, e salve-o no GIT.

Lembre-se, o repositório deverá conter os seguintes arquivos:

* Arquivo Python com os pandas dataframes salvando as informações no banco de dados.
* Agregação criada no MongoDB com os passos descritos neste documento.
* Collections do MongoDB exportadas.
* Arquivo World contendo os questionários respondidos.

# Finalização

Assim que finalizar, informe a pessoa responsável pelo processo seletivo, que as atividades desenvolvidas foram concluídas, e envie também, o link do repositório GIT onde as informações foram salvas.

# Desempenho

Descreva abaixo quais foram os pontos mais fáceis, e mais difíceis encontrados nesta atividade, caso não tenha conseguido executar algum passo nesta atividade, especifique quais as dificuldades enfrentadas que o impediu de realizá-las:

**Pontos fáceis:**

As etapas iniciais: 1.1, 1.2, 1.3, foram executadas sem nenhuma dificuldade.

**Pontos medianos:**

A etapa: 1.4, onde foi apresentado um grau maior de complexidade, mais em relação a assertividade da sintaxe específica da agregação do MongoDB. Embora os conceitos tanto de lookup e agregação já fossem conhecidos, demandou um tempo adicional maior do que esperava. O ponto crítico foi a parte do "$arrayElemAt" para que eu pudesse acessar corretamente os dados retornados pelo $lookup.

Uma vez que a etapa 1.4 foi vencida, a etapa 1.5, já não foi mais um problema, a partir da 1.4 funcionando corretamente, foi somente mudar o parâmetro para conseguir com mais facilidade concluir também a próxima etapa.

**Pontos Difíceis:**

Com certeza, onde tive maior dificuldade foi para identificar para o meu caso de uso a sintaxe da agregação do MongoDB, já que minha experiência com MongoDB não é comparável com o Python, no Python possuo bastante experiência em diversos projetos, MongoDB ainda não possuo muitos para se tornar comparável aao Python, mas apesar das dificuldades. O teste foi concluído com êxito.