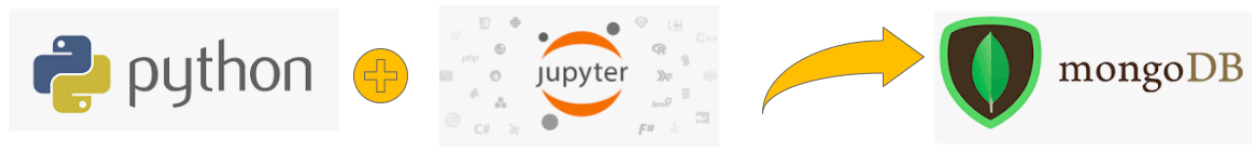


## Extração de dados - mongoDB via *python*



### Pacotes necessário do python

Para instalar os pacotes(bibliotecas) no python, basta usar o comando `!pip3 install nome_pacote` ou `!pip3 install requirements.txt` em que *requirements.txt* é um o arquivo texto com os nomes dos pacotes a serem instalados.

```
#!pip3 install -r requirements.txt # Basta rodar uma vez apenas
```

```
import numpy as np #tarefas matemáticas e machine learning
import pandas as pd #manipulação de data frames e machine learning
from pandas import Series, DataFrame
from datetime import datetime #manipulação de variáveis em formato de data
from pymongo import DESCENDING #util para acessar o mongoDB
from collections import OrderedDict
pd.set_option('display.max_columns', None) # Não limita o número de colunas para visualização
```

### Abaixo um exemplo de como acessar o mongoDB

```
from pymongo import MongoClient
def db_connect(ambiente):

    if ambiente == "prod":
        db_uri = "mongodb://credenciais_de_acesso_ao_mongoDB"
        db_name = "nome_da_base_de_dados"

        client = MongoClient(db_uri)
        db = client[db_name]

    if not db:
        print("ERROR: Couldn't connect to database")
        return None

    return db

db = db_connect("prod")
```

```
# Define o período da coleta dos dados
```

```
period_start = datetime(2020, 8, 1, 0, 0, 0)
period_end = datetime(2020, 8, 2, 0, 0, 0)
entity_id = 12
```

### Documentos na base de dados

```
# Distinct nos 'document_types' do banco
document_types = db.history_people.distinct('document_type', {'entidade_id': entity_id})
```

```
document_types
```

```
['medical_record',  
 'vital_signs',  
 'exams',  
 'medicines',  
 'pharmacy',  
 'movements',  
 'alert']
```

## Consulta para extrair as informações de sinais vitais

```
query = {  
    'entidade_id': entity_id,  
    'document_type': 'vital_signs',  
    'date': {'$gte': period_start,  
             '$lte': period_end}  
}  
  
limiters = {  
    '_id': 0,  
    'date': 1,  
    'document.data_coleta': 1,  
    'document.data_liberacao': 1,  
    'document.freq_cardiaca': 1,  
    'document.freq_respiratoria': 1,  
    'document.glicemia_capilar': 1,  
    'document.nivel_consciencia': 1,  
    'document.pa_diastolica': 1,  
    'document.pa_sistolica': 1,  
    'document.sat_o2': 1,  
    'document.temperatura': 1,  
    'document.paciente_id': 1  
}  
  
vs = list(db.history_people.find(query, limiters))
```

## Tamano da amostra

```
len(vs)
```

```
7469
```

## Formatando os dados para data frame

### Dados de Sinais vitais

```
from pandas.io.json import json_normalize  
  
### Transformando de formato JSON para data frame  
df_vs = json_normalize(vs)  
### Ordenando por data  
df_vs.sort_values(by=['date'], inplace=True)
```

```
### Visualizando as primiras linhas do data frame
df_vs.head(5)
```

```
<ipython-input-22-51dfdbf79ed6>:2: FutureWarning: pandas.io.json.json_normalize is deprecated, use pandas.io.json._normalize
df_vs = json_normalize(vs)
```

```
date
document.data__coleta
document.data__liberacao
document.freq__cardiaca
document.freq__respiratoria
document.glicemia__capilar
document.nivel__consciencia
document.pa__diastolica
document.pa__sistolica
document.paciente__id
document.sat__o2
document.temperatura
0
2020-08-01
2020-08-01
2020-08-01 00:03:45
131.0
22.0
NaN
None
62.0
135.0
731032
95.0
38.1
75
2020-08-01
2020-08-01
2020-08-01 03:02:58
86.0
16.0
NaN
```

None  
54.0  
93.0  
2782882  
100.0  
35.9  
74  
2020-08-01  
2020-08-01  
2020-08-01 03:02:05  
117.0  
16.0  
87.0  
None  
66.0  
122.0  
2802042  
100.0  
36.2  
73  
2020-08-01  
2020-08-01  
2020-08-01 02:54:08  
100.0  
24.0  
NaN  
None  
62.0  
103.0  
750156  
97.0  
38.2  
72  
2020-08-01  
2020-08-01  
2020-08-01 02:52:44

82.0

28.0

NaN

None

67.0

128.0

1958456

92.0

36.7

```
len(df_vs)
```

7469

### Nome das variáveis

```
df_vs.columns
```

```
Index(['entidade_id', 'atendimento_id', 'date', 'document.data_coleta', 'document.data_liberacao', 'document.freq_cardiaca', 'document.freq_respiratoria', 'document.glicemia_capilar', 'document.nivel_consciencia', 'document.pa_diastolica', 'document.pa_sistolica', 'document.paciente_id', 'document.sat_o2', 'document.temperatura'], dtype='object')
```

### Exportando os dados localmente em formato csv

```
df_vs.to_csv('nome.csv', sep = ',', index = False)
```