# Extração de dados - mongoBD 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(ambient):  
  
 if ambient == "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.json\_normalize instead  
 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)