



# Juliane Andressa Camatti Turma T07

A atividade busca aprimorar os conhecimentos de Python oferecidos em aula, como uma das etapas importantes para atender a necessidade atual de indústrias em processar seus dados em tempo real em busca de mais assertividade nas tomadas de decisão.

## Objetivo Geral:

Elaborar uma análise descritiva de dados oriundos de máquinas CNC portadoras de sistemas legados, com o intuito de conhecer o real desempenho de máquina.







## Objetivo Específico do projeto:

Fazer a junção dos dados fornecidos pela área em uma mesma tabela para identificar:

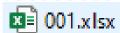
- Qualidade dos dados;
- Oportunidades de análise dos mesmos;
- \*\*Tempo total de funcionamento da máquina;

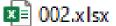


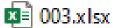
Execução do Projeto

- Quais bases de dados foram usadas?
- Quais dados elas continham?
- Qual foram os critérios de seleção do Modelo.









**⊠**≣ 004.xIsx

**□** 005.xlsx

**図** 006.xIsx

**☑** 007.xlsx

**□** 008.xlsx

**⊠** 009.xlsx

010.xlsx

011.xlsx

**図** 012.xIsx

013.xlsx

**▼** 014.xlsx



1º Passo: salvar os dados na pasta no drive

2º Passo: importar os dados



```
from google.colab import drive
drive.mount('/content/drive')
%cd drive/MyDrive/
%pwd
```



3º Passo: Importar todas as planilhas para análise, elegendo a coluna horário como fixa.

```
[3] import pandas as pd

df1 = pd.read_excel('001.xlsx',index_col = 0)

df2 = pd.read_excel('002.xlsx',index_col = 0)

df3 = pd.read_excel('003.xlsx',index_col = 0)

df4 = pd.read_excel('004.xlsx',index_col = 0)

df5 = pd.read_excel('005.xlsx',index_col = 0)

df6 = pd.read_excel('006.xlsx',index_col = 0)

df7 = pd.read_excel('007.xlsx',index_col = 0)

df8 = pd.read_excel('008.xlsx',index_col = 0)

df9 = pd.read_excel('009.xlsx',index_col = 0)

df10 = pd.read_excel('010.xlsx',index_col = 0)

df11 = pd.read_excel('011.xlsx',index_col = 0)

df12 = pd.read_excel('012.xlsx',index_col = 0)

df13 = pd.read_excel('013.xlsx',index_col = 0)

df14 = pd.read_excel('014.xlsx',index_col = 0)
```



4º Passo: Concat. Usar essa função para colocar todas as planilhas em uma mesma coluna de acordo com a coluna Index (representa hora e data)

```
[4] df= pd.concat([df1, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11, df12, df13, df14],axis = 1)
```

```
[5] df1.shape #tamanho planilha 1
```

[6] df.shape #tamanho planilha toda

(544, 14)

Divergências nas colunas de dados



```
dftotal.isnull().sum() #dados faltantes
dftotal.info()
<class 'pandas.core.frame.DataFrame'>
                                                              index
RangeIndex: 544 entries, 0 to 543
                                                              MODO AUTO
                                                                                                      438
Data columns (total 15 columns):
                                                              MODO MDI
                                                                                                      444
    Column
                                     Non-Null Count Dtype
                                                              MODO JOG
                                                                                                      438
                                                              FEED DISABLE
                                                                                                      444
                                                    object
    index
                                     544 non-null
    MODO AUTO
                                     106 non-null
                                                    float64
                                                              FEEDRATE OVERRIDE
                                                                                                      438
                                                    float64
                                     100 non-null
    MODO MDI
                                                              FEED STOP GEOAXIS 1
                                                                                                      444
                                     106 non-null
                                                    float64
    MODO JOG
                                                              HORAS TOTAIS DE FUNCIONAMENTO
                                                                                                      358
    FEED DISABLE
                                     100 non-null
                                                    float64
                                                              MINUTOS TOTAIS DE FUNCIONAMENTO
                                                                                                      346
                                                    float64
    FEEDRATE OVERRIDE
                                     106 non-null
                                                              CONTADOR TROCA DE PALETES
                                                                                                      377
                                                    float64
    FEED STOP GEOAXIS 1
                                     100 non-null
                                                              CONTADOR DE PECAS
                                                                                                      444
    HORAS TOTAIS DE FUNCIONAMENTO
                                     186 non-null
                                                    float64
                                                    float64
    MINUTOS TOTAIS DE FUNCIONAMENTO
                                     198 non-null
                                                              SPINDLE OVERRIDE
                                                                                                      443
                                                    float64
    CONTADOR TROCA DE PALETES
                                     167 non-null
                                                              MOTOR STATUS
                                                                                                      437
    CONTADOR DE PECAS
                                     100 non-null
                                                    float64
                                                              ALARME TEMPERATURA MOTOR
                                                                                                      444
    SPINDLE OVERRIDE
                                     101 non-null
                                                    float64
                                                              pinça spimdle
                                                                                                      345
                                     107 non-null
                                                    float64
    MOTOR STATUS
                                                              dtype: int64
    ALARME TEMPERATURA MOTOR
                                                    float64
                                     100 non-null
    pinça spimdle
                                     199 non-null
                                                    float64
```

dtypes: float64(14), object(1)

memory usage: 63.9+ KB

```
[138] df_analise = pd.concat([df1, df3],axis = 1)
```

```
[139] df_novo = df_analise.reset_index(drop=False)
df_novo.head(10)
```

	hora	MODO AUTO	MODO JOG
0	09/03/2021 10:57:01	1	0
1	09/03/2021 16:59:30	1	0
2	09/03/2021 17:59:33	1	0
3	09/03/2021 18:59:37	1	0
4	09/03/2021 19:59:39	1	0
5	09/03/2021 20:59:41	1	0
6	09/03/2021 21:59:43	1	0
7	09/03/2021 22:59:46	1	0
8	09/03/2021 23:59:48	1	0
9	10/03/2021 00:59:50	1	0



#### Formatar hora

[17] df\_novo['data'] = pd.to\_datetime(df\_novo['hora'], dayfirst= True)
 df\_novo.head(10)

	hora	MODO AUTO	MODO JOG	data
0	09/03/2021 10:57:01	1	0	2021-03-09 10:57:01
1	09/03/2021 16:59:30	1	0	2021-03-09 16:59:30
2	09/03/2021 17:59:33	1	0	2021-03-09 17:59:33
3	09/03/2021 18:59:37	1	0	2021-03-09 18:59:37
4	09/03/2021 19:59:39	1	0	2021-03-09 19:59:39
5	09/03/2021 20:59:41	1	0	2021-03-09 20:59:41
6	09/03/2021 21:59:43	1	0	2021-03-09 21:59:43
7	09/03/2021 22:59:46	1	0	2021-03-09 22:59:46
8	09/03/2021 23:59:48	1	0	2021-03-09 23:59:48
9	10/03/2021 00:59:50	1	0	2021-03-10 00:59:50



#### Selecionar data

df\_novo['hora\_novo\_agrupamento'] = df\_novo['data'].dt.to\_period('D') df\_novo.head(10)

	hora	MODO AUTO	MODO JOG	data	hora_novo_agrupamento
0	09/03/2021 10:57:01	1	0	2021-03-09 10:57:01	2021-03-09
1	09/03/2021 16:59:30	1	0	2021-03-09 16:59:30	2021-03-09
2	09/03/2021 17:59:33	1	0	2021-03-09 17:59:33	2021-03-09
3	09/03/2021 18:59:37	1	0	2021-03-09 18:59:37	2021-03-09
4	09/03/2021 19:59:39	1	0	2021-03-09 19:59:39	2021-03-09
5	09/03/2021 20:59:41	1	0	2021-03-09 20:59:41	2021-03-09
6	09/03/2021 21:59:43	1	0	2021-03-09 21:59:43	2021-03-09
7	09/03/2021 22:59:46	1	0	2021-03-09 22:59:46	2021-03-09
8	09/03/2021 23:59:48	1	0	2021-03-09 23:59:48	2021-03-09
9	10/03/2021 00:59:50	1	0	2021-03-10 00:59:50	2021-03-10



Agrupamento baseado na hora e na soma do status ligado

```
[22] df_novo.groupby('hora_novo_agrupamento')['MODO AUTO'].sum()

hora_novo_agrupamento
2021-03-09 9
2021-03-10 24
2021-03-11 24
2021-03-12 24
2021-03-13 19
Freq: D, Name: MODO AUTO, dtype: int64
```



Criar uma coluna nova sobre o status da máquina

```
def faixa(x):
    if x == 0:
        return('desligada')
    elif x == 1:
        return('ligada')
    else:
        return('Valor não encontrado!')

df_novo['status'] = df_novo['MODO AUTO'].apply(faixa)
    df_novo.head(20)
```

	hora	MODO AUTO	MODO JOG	data	hora_novo_agrupamento	status
0	09/03/2021 10:57:01	1	0	2021-03-09 10:57:01	2021-03-09	ligada
1	09/03/2021 16:59:30	1	0	2021-03-09 16:59:30	2021-03-09	ligada
2	09/03/2021 17:59:33	1	0	2021-03-09 17:59:33	2021-03-09	ligada
3	09/03/2021 18:59:37	1	0	2021-03-09 18:59:37	2021-03-09	ligada
4	09/03/2021 19:59:39	1	0	2021-03-09 19:59:39	2021-03-09	ligada
5	09/03/2021 20:59:41	1	0	2021-03-09 20:59:41	2021-03-09	ligada
6	09/03/2021 21:59:43	1	0	2021-03-09 21:59:43	2021-03-09	ligada



Desenvolver uma tabela dinâmica com o pivot\_table

table = pd.pivot\_table(df\_novo, index =['status', 'hora\_novo\_agrupamento'], values = ['MODO AUTO']) table.head(30)

Mα			

desligada	2021-03-10	0
	2021-03-11	0
	2021-03-12	0
ligada	2021-03-09	1
	2021-03-10	1
	2021-03-11	1
	2021-03-12	1
	2021-03-13	1



## Conclusões

## Relembrando:

Fazer a junção dos dados fornecidos pela área em uma mesma planilha variation para identificar:

- Qualidade dos dados;
- Oportunidades de análise dos mesmos; 🗸
- Tempo total de funcionamento da máquina;



df.head(10) # para observar as 10 primeiras linhas da planilha toda

<b>□</b>													
	09/03/2021 10:57:01	1.0	0.0	0.0	0.0	26.0	0.0	7290.0	437400.0	85200.0	0.0	14.0	1.0
	09/03/2021 16:59:30	1.0	0.0	0.0	0.0	26.0	0.0	17642.0	1058559.0	3399.0	0.0	14.0	1.0
	09/03/2021 17:59:33	1.0	0.0	0.0	0.0	26.0	0.0	17643.0	1058618.0	3401.0	0.0	14.0	1.0
	09/03/2021 18:59:37	1.0	0.0	0.0	0.0	26.0	0.0	17644.0	1058678.0	3401.0	0.0	14.0	1.0
	09/03/2021 19:59:39	1.0	0.0	0.0	0.0	26.0	0.0	17645.0	1058738.0	3403.0	0.0	14.0	1.0



### Trabalhos futuros

Quais passos podem ser necessários para um futuro desenvolvimento do projeto.

- ✓ Fazer uma limpeza dos dados;
- Padronizar os dado em relação a coluna tempo;
- ✓ Fazer mais análises mais detalhadas como a correlação, média entre outros;
- ✓ Gerar gráficos e dashboard no powerbi (talvez);
- ✓ Processo seja automático toda vez que for liberado relatório novo no servidor da empresa;



## Referências

Informe os links de notebooks, blogs e livros que foram utilizados:

- Análise descritiva de dados; Python para ciência e dados;
- **>** Orientação: Prof. Bruna;
- > Help: Dani e do Carlos;
- Programe Python YouTube





## **VOCÊ EM CONEXÃO** com o mercado digital